Security and Efficiency of Delegated Quantum Computing

Dominik Leichtle



Sorbonne Université – École Doctorale de Sciences Mathématiques de Paris Centre
LIP6 – Quantum Information
Institut de Mathématiques de Jussieu – Paris Rive Gauche

Security and Efficiency of Delegated Quantum Computing

Dominik Leichtle

Doctoral thesis

Advisory team: Elham Kashefi & Antoine Joux

Presented and publicly defended in February 2024, in front of the following jury:

- Dunjko Vedran,
  Leiden University (The Netherlands),
  Reviewer
- Fawzi Omar,
  ENS Lyon (France), *Examiner*
- Joux Antoine,
  CISPA Helmholtz Center for
  Information Security (Germany),
  Advisor
- Kashefi Elham,
  CNRS, Sorbonne University (France),
  University of Edinburgh (United
  Kingdom), *Advisor*
- Murao Mio,
  University of Tokyo (Japan), *President*
- Raussendorf Robert,
  Leibniz University Hannover
  (Germany), *Reviewer*

Abstract

Quantum information promises to revolutionize our world, from the way in which we communicate to the way in which we compute, deriving its power directly from the laws that govern the behavior of nature on extremely small scales – quantum mechanics. In the near future, the hardware of possibly useful quantum computers is expected to remain very expensive and thus out of reach for most interested end users. In such a world, it is an important problem to provide security guarantees for customers who wish to remotely instruct quantum servers, by keeping their data private (*blindness*) and checking the correctness of the results (*verification*). This functionality of *secure delegated quantum computing* received a lot of attention during recent years, but still admits many open questions.

In this thesis, we explore the (im)possibility of securing delegated quantum computations in different settings: what is the hardware that the client needs trusted access to, what is the minimum hardware required by the server, and how must the parties communicate? This work is driven by the motivation to break down the barriers that keep us from securing and verifying quantum computations in practice, by identifying and removing unnecessary overheads.

We set out by questioning the necessity of quantum communication between the client and the server, and find that, while in specific situations classical communication is entirely sufficient, most generally the security of delegation protocols relies irreplaceably on the very quantumness of the information exchanged between the parties. This proves that quantum communication is indeed an essential asset in our cryptographic toolbox.

We then shift our focus to the server that was suffering from impractical overheads in previous attempts at quantum verification. We show that for a large class of interesting quantum computations, there is no fundamental need to reserve extra hardware for any cryptographic techniques. Indeed, we give concrete constructions of secure protocols that achieve blindness and verification on hardware of the same size that would be

required to perform the original, unsecured computation, and provide a systematic way of optimizing their efficiency in customized settings.

Our journey then takes us to the problem of *quantum secure multi-party computation*, a generalization of the previous functionality to more than two participating, mutually distrusting parties. We explore how the improvements that we obtained in the two-party setting can be transferred to the multi-party case, and finish with the presentation of two actual experiments that demonstrate the practical impact and real-world feasibility of the results obtained during the course of this thesis.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Dominik Leichtle

Sorbonne Université, February 2024

Signed: ..... Date: .....

Contents

List of Figures

List of Tables

Chapter 1

Introduction

If quantum machines eventually turn out to eclipse their classical counterparts with respect to computational power, how will we, caught within the limitations of our restricted means, be able to possibly witness this miracle? How will we be able to develop trust in the results that more and more powerful quantum computers present us once we are out of methods to obtain them using devices familiar to us?

In this thesis, we offer a concrete approach to this epistemological conundrum through the study of protocols for the verification of quantum computation. In particular, we ask:

> *What minimal assumptions or trust in hardware do we require to secure quantum computations performed on an untrusted and possibly maliciously acting device?*

Quantum information and computing. At first (human) glance, our world seems classical. However, at very small scales, our world behaves fundamentally differently from what humankind believed for a long time. The discovery of these strange phenomena and the proposal of the theory of quantum mechanics at the beginning of the 20th century eventually paved the way towards a new paradigm in the theory of information. While classical computers operate on bits that are restricted to the binary values of 0 and 1, their quantum analogue, *qubits*, can exist in *superpositions* of these two poles. Even though a qubit can take infinitely many different states, this does not mean that an infinite amount of information can be stored in it, due to limited accessibility of the full description of a quantum state. According to the laws of quantum mechanics, information about a quantum system can only be accessed by the means of *measurements*, a process

that can be thought of as the extraction of a limited amount of classical information about the quantum system while irreversibly changing the system itself. This inherent restriction leads to other principles of quantum information that might appear odd to our classical intuition. As one example, the *no-cloning principle* states that there is no general way of copying an unknown quantum state. Quantum information can therefore not be distributed just as classical information and requires special treatment in communication. While this might sound like a disadvantage, this property can also be incredibly powerful in cryptographic contexts, as we will soon see. One final non-classical feature of quantum information that deserves to be mentioned at this point is its ability to retain *entanglement*. At the core, this property allows the state of one qubit to be inherently connected to the state of another, even when separated by physical distances.

This behavior of quantum systems opens up new possibilities for computation and communication. As one of the most famous examples, Shor's algorithm [Sho99] manages to efficiently factor large numbers on a quantum computer. It is currently not known whether it is possible to achieve this on a classical computer. But also beyond tackling abstract mathematical problems, scalable quantum computers are believed to have a drastic impact on our world.

Cryptography. In some sense, cryptography can be seen as a tool in the hands of the weak in the struggle for justice and equality. Cryptography is the science and practice of secure communication, seeking to protect information from unauthorized access or alteration. Through the use of mathematical algorithms, it strives to equilibrate the imbalance between weak parties and much more powerful, potentially malicious actors.

Maybe surprisingly, classical cryptography has been very successful in the design of algorithms that achieve a wide array of functionalities, from basic primitives such as private and authenticated communication, to advanced tools like secure multi-party computation, that allows multiple, each other distrusting parties to jointly evaluate a function over their secret inputs.

With the advent of quantum computing, cryptography needs to be rethought to adapt to a quantum world. Quantum adversaries will be able to outperform classical adversaries and stage even more powerful attacks against our cryptosystems. We will hence need to update them to keep secure. This is the goal of *post-quantum cryptography*. At the same time, the distinct properties of quantum information can also be used to our advantage, to create new cryptographic protocols in which also the honest parties wield the quantum power, giving rise to truly *quantum cryptography*.

This thesis is drawing from both of these fields, and employs post-quantum and quantum primitives to construct better quantum cryptography.

Secure delegated quantum computing. Any first useful quantum computers to be available will be very expensive and will most likely require frequent calibration, making them inaccessible to most potential users. It seems therefore likely that quantum computing will be offered as a cloud service, to which customers could submit their tasks on demand. Indeed, this development can already be observed with currently available experimental quantum devices. In such a scenario, clients will want to ensure confidentiality of their data and algorithms, and the integrity of their computations.

Several protocols have been proposed for this purpose, one of them being the *Universal Blind Quantum Computing* (UBQC) protocol [BFK09]. It allows a client to delegate a universal computation to a more powerful quantum server while keeping their input, output, and algorithm private, or *blind*. To this end, the client is required to operate on single qubits only, all other of the client's operations being classical.

More protocols have been developed to additionally provide *verification*, which empowers the client to check the obtained results for correctness [FK17; Aha+17; Bro18; GKK19]. Further works in this line of research [KW17b; Mah18b; XTH20] are aimed at reducing the resources consumed by the protocols, in terms of (quantum) computing power of client and server, and their (quantum) communication.

However, no proposal has so far managed to optimize these schemes sufficiently to reach the regime of current implementability, because of impermissible computation or communication overheads. It is the goal of this thesis to finally close the gap between our theoretical protocols and our practical quantum abilities, and to present the first readily realizable schemes for secure delegated quantum computing.

Contributions

The contributions of this work are split into the following six main chapters.

Chapter 3: Classical-client delegated quantum computing. Many protocols for the secure delegation of quantum computing require the client and server to establish the necessary correlations using communication via quantum channels, rendering them unfeasible before the widespread deployment of a reliable quantum network.

This naturally leads to the question of whether it is possible to securely delegate quantum computations in a setting where client and server are restricted to classical communication and do not have access to other kind of shared quantum resources. More generally, we are interested in the fundamental question:

> *Can quantum channels generally be securely replaced by protocols relying only on classical communication?*

We formalize this problem by asking about the possibility of securely implementing a functionality called *classical-client remote state preparation* ($\mathsf{RSP}_{\mathsf{CC}}$) that allows a client to remotely prepare quantum states over a classical channel. We investigate this question in the Abstract Cryptography framework which guarantees a strong sense of *general-purpose* security, context-insensitive and robust to arbitrary composition.

In Chapter 3 which is based on publication [Bad+20] we answer this question in the negative. Even for computational security only, any classically constructed ideal RSP resource must leak the full classical description of the prepared quantum state to the server. We further study the possibility of using $\mathsf{RSP}_{\mathsf{CC}}$ in the restricted context of Universal Blind Quantum Computing (UBQC), and find that while generally composable security still remains impossible, a concrete implementation can be found that achieves security in a weaker, but yet useful game-based notion.

Chapter 4: Quantum verification with minimal overhead. Previously existing protocols for the blind and verifiable delegation of quantum computation were suffering from high overheads and over-sensitivity to noise: device imperfections would trigger the same detection mechanisms as malicious attacks, resulting in perpetually aborted computations. This leads to the fundamental question of whether these overheads are inherent and necessary for quantum verification, or whether they could, in principle, be avoided.

> *Do protocols for the verified delegation of quantum computations necessarily require more powerful quantum hardware than would be needed to perform the target computation in an unsecured manner?*

This problem can also be seen as asking whether all available quantum hardware can be used to perform the algorithm of interest, or whether at least a part of it *must be wasted* to secure its execution.

In Chapter 4 which is based on publications [Kas+21; Lei+21] we provide an answer to this question by introducing the first blind and verifiable protocol for delegating BQP computations to a quantum server with repetition as the only overhead. The protocol achieves composable, statistical security with only negligible soundness error and can tolerate a constant amount of global noise. This represents an important step towards bringing the verification of quantum computations closer to near-term feasibility.

Chapter 5: A framework for quantum verification. In an attempt to further optimize protocols for verifiable blind quantum computing, in Chapter 5, which is based on publication [Kap+22], we search for sufficient conditions for generally composable security and uncover a fundamental correspondence between error-detection and verification. As a direct application, we demonstrate how to systematize the search for new efficient and robust blind verification protocols and give a concrete construction for the verification of BQP computations beating previously known protocols in terms of efficiency.

Chapter 6: Quantum secure multi-party computation. Secure multi-party computation (SMPC) protocols allow several mutually distrusting parties to collectively compute a function over their joint inputs.

In Chapter 6, based on publication [Kap+23], we introduce a protocol that lifts classical SMPC to quantum SMPC in a composable and statistically secure way, even for a single honest party. Our proposal is a generalization of efficient single-client verification protocols to the multi-client case and, hence, unlike previous SMPC protocols, requires only a very limited overhead compared to the unsecured target computation.

As a building block in the construction of the new protocol, we find two cryptographic primitives of independent interest. The first is a modular way to turn single-client remote state preparation (RSP) into multi-client collaborative remote state preparation (CRSP). We further present a new technique for quantum verification that requires the client to prepare state only in a single plane of the Bloch sphere. Previous comparable verification protocols required the preparation of states from more than a single plane. In the course of proving the security of this new verification protocol, we uncover a fundamental invariance inherent to measurement-based quantum computing.

Chapters 7 & 8: Experimental realizations of delegated quantum computing. While the core part of this thesis is of a theoretical nature, we also present two novel

experimental realizations of single-client and multi-client secure delegated quantum computing, respectively.

Chapter 7 is based on publication [Drm+23b] and presents the first hybrid matter-photon implementation of verifiable blind quantum computing with a single client. The experiment uses a trapped-ion quantum server and a photonic client that are connected by a fibre-optic quantum network link. The implementation admits all main features necessary for scalability, including the avoidance of post-selection, rendered possible by the availability of memory qubits.

Chapter 8, which is based on publication [Pol+23], generalizes this setting to the case of multiple weak clients that collaboratively delegate a quantum computation to a more powerful quantum server, thereby implementing multi-client blind quantum computing. While this entirely photonic experiment does not feature the verifiability of the target computation, it is based on a novel quantum network architecture, the Qline, designed with a focus on scalability and ease of deployment. In particular, the clients only need to be able to perform single-qubit operations and do not require access to trusted state preparations or measurements, making this the first protocol of its kind.

Publications. This thesis is based on the results presented in the following papers, listed in chronological order:

- [Bad+20] rules out the existence of composably secure protocols that could replace a quantum channel using only classical communication. However, it proves that in the restricted scenario of UBQC, classical communication suffices to achieve weaker non-composable security. This work was published in the conference proceedings of ASIACRYPT 2020.

- [Kas+21] presents a new, highly efficient, and noise-robust protocol for the verified and blind delegation of pseudodeterministic quantum polynomial-time computations. [Lei+21] generalizes these results to the larger class of bounded-error quantum polynomial-time computations (BQP). This work was published in PRX Quantum.

- [Kap+22] introduces a framework for the design of blind verification protocols which uncovers a deep connection between the fields of quantum verification, error detection, and error correction. This framework allows for the construction of even more efficient and customizable protocols for the secure delegation of quantum computation. This work is currently under review.

- [Kap+23] constructs a quantum secure multi-party computation protocol optimized for resource efficiency, using the tools from the antecedent works on quantum verification. This work is currently under review.
- [Drm+23b] demonstrates that quantum verification has reached practical implementability by presenting an experimental realization of a single-client secure delegation of quantum computation, achieving both blindness and verifiability. This work was accepted for publication in PRL.
- [Pol+23] pushes the boundaries of the practically possible further by experimentally implementing a multi-client blind delegation of quantum computation on an entirely photonic setup. This work was published in Nature Communications.

How to read this thesis

Chapter 2 gives an overview of the preliminary knowledge required to read this thesis. It focuses on the basic principles of quantum computing and quantum information, and explains fundamental concepts of cryptography that are of subsequent importance.

In Chapters 3-8, the main research contributions of this thesis are presented. Each of the chapters is based on a different publication and attempts to be as self-contained as possible. It is, therefore, entirely possible and up to the reader to directly jump to the chapter of interest. Every chapter starts with its own abstract, which briefly summarizes its main research questions and contributions.

The final Chapter 9 wraps up this work by recalling the main milestones of this journey through the realm of delegated quantum computing. We conclude by presenting a few interesting open questions that hopefully the invested reader will feel challenged to answer.

Chapter 2

Preliminaries

In this chapter, we present a selection of concepts important for the understanding of this thesis: Abstract Cryptography, measurement-based quantum computing, and basic protocols for delegated quantum computing. This overview is not meant to replace full courses on these topics, but rather to gather some of the basic concepts required to read this thesis.

We assume basic familiarity with quantum information and computing; for a detailed introduction, see [NC00] (note that henceforth all Hilbert spaces are assumed to have a finite dimension).

2.1 (Abstract) Cryptography

The definition of game-based security is pretty straightforward: we define a *game* between a challenger and an (arbitrary) adversary: a protocol is secure if no adversary can win this game with "good" probability. The problem with this approach is that one game describes only one possible attack, and it might be hard to list all the possible attacks against a protocol. Therefore, a protocol that proves to be secure in a specific game might not be secure in an arbitrary environment (composed with other protocols in parallel or in series).

Composable security on the other hand takes a different approach to phrasing the guarantees achieved by a protocol. Loosely speaking, a protocol is composable when it is shown to be secure in an arbitrarily adversarial environment[1], and where secure

[1]Of course, the environment may still be limited to "efficient" computations.

means that it achieves a well-defined ideal (secure-by-definition) resource. This means the protocol retains the desired functionality even if it is composed of other instances of its own or a completely different protocol. There are several approaches which provide a general framework to study this cryptographic definitions [Can01; BPW03; MR11], but we will focus in this thesis on Abstract Cryptography (AC) (also known under the term Constructive Cryptography (CC)). In this section, we provide relevant terminologies required to analyze composable security in this framework, introduced by Maurer and Renner in [MR11]. For more details, we refer readers to some of the previous works [Mau11; MR11; Dun+14; DK16].

Note, that the AC framework is equivalent to the Quantum Universal Composability (Q-UC) Model of [Unr10] if a single adversary controls all corrupted parties – which is the case in this work. Therefore, any protocol which is secure in the Q-UC model is also secure in the AC model considered here.

The basic elements of AC are systems: objects with well-distinguished and labeled interfaces. The system uses interfaces to exchange information with the outside world and/or other systems. Systems are grouped into distinct classes: resources, converters, filters, and distinguishers.

In this framework, the purpose of a secure protocol π is, given a number of available resources $\mathcal{R}$, to construct a new resource – written as $\pi\mathcal{R}$. This new resource can be itself reused in a future protocol.

The actions of all honest players in a given protocol are represented as a sequence of efficient CPTP maps acting on their internal quantum registers – which may contain communication registers, both classical and quantum. An n-party quantum protocol is therefore described by $\pi = (\pi_1, \ldots, \pi_n)$ where π_j is the aforementioned sequence of efficient CPTP maps executed by party j, called the *converter* of party j.

A *resource* $\mathcal{R}$ is described as a sequence of CPTP maps with an internal state. It has input and output interfaces describing which party may exchange states with it. Some interfaces may be *filtered*, meaning that they are only accessible to a corrupted party.[2] Resources work by having the participating parties sending it states at its input interfaces, applying the specified CPTP map after all input interfaces have been initialised and then outputting the resulting state at its output interfaces in a specified order. Classical resources are modelled by considering that the input state is measured in the computational basis upon reception and the output is a measurement result of its

[2]In this thesis, filtered input interfaces consist of single bits, set to 0 in the default, honest case.

internal state.

In order to define the security of a protocol, we need to give a pseudo-metric on the space of resources. The security analysis then consists of considering a special type of converters called *distinguishers*. The distinguisher's aim is to discriminate between resources $\mathcal{R}_0$ and $\mathcal{R}_1$ which have the same input and output interfaces. It attaches to the inputs and outputs of one of the resources, interacting with it according to its own – possibly adaptive – strategy, and outputs a single bit indicating its guess as to which resource it had access to. Two resources are said to be indistinguishable if no distinguisher can make this guess with good probability.

Definition 2.1.1 (Indistinguishability of Resources)**.** *Let $\epsilon(\eta)$ be a function of security parameter η and $\mathcal{R}_0$ and $\mathcal{R}_1$ be two resources with the same input and output interfaces. Then, these resources are ϵ-statistically-indistinguishable, denoted $\mathcal{R}_0 \underset{stat,\epsilon}{\approx} \mathcal{R}_1$, if for all (unbounded) distinguishers $\mathcal{D}$, we have:*

$$|\Pr[b = 1 \mid b \leftarrow \mathcal{D}\mathcal{R}_0] - \Pr[b = 1 \mid b \leftarrow \mathcal{D}\mathcal{R}_1]| \leq \epsilon. \tag{2.1}$$

Analogously, $\mathcal{R}_0$ and $\mathcal{R}_1$ are said to be computationally indistinguishable if this holds for all quantum polynomial-time distinguishers.

The correctness of a protocol π applied to resource $\mathcal{R}$ can be expressed as the indistinguishability between the resource $\pi\mathcal{R}$ and a desired target resource $\mathcal{S}$.

The security of the protocol is captured by the fact that the resources remain indistinguishable if we allow some parties to deviate, in the sense that they are no longer forced to use the converters defined in the protocol but can use any other CPTP maps instead. This is done by removing the converters for those parties in Equation 2.1, keeping only $\pi_{M^c} = \{\pi_j\}_{j \notin M}$ where M is the set of corrupted parties. On the other side, there must exist a converter called a *simulator* which attaches to the interfaces of $\mathcal{S}$ for corrupted parties $j \in M$ and aims to reproduce the transcript of honest players interacting with the corrupted ones. The security is formalised as follows in Definition 2.1.2.

Definition 2.1.2 (Construction of Resources)**.** *Let $\epsilon(\eta)$ be a function of security parameter η. We say that an n-party protocol π ϵ-statistically-*constructs *(or* realizes*) resource $\mathcal{S}$ from resource $\mathcal{R}$ against adversarial patterns $\mathsf{P} \subseteq \wp([N])$, denoted $\mathcal{R} \xrightarrow[\epsilon]{\pi} \mathcal{S}$, if:*

1. *It is correct: $\pi\mathcal{R} \underset{stat,\epsilon}{\approx} \mathcal{S}\perp$, where $\perp$ filters the malicious interfaces;*

2. *It is secure for all subsets of corrupted parties in the pattern* $M \in \mathsf{P}$*: there exists a converter called simulator* σ_M *such that* $\pi_{M^c}\mathcal{R} \underset{stat,\epsilon}{\approx} \mathcal{S}\sigma_M$.

Analogously, computational correctness and security are given for computationally bounded distinguishers as in Definition 2.1.1, and with a quantum polynomial-time simulator σ_M.

The intuition behind this definition is that if no distinguisher can know whether he is interacting with an ideal resource or with the real protocol, then it means that any attack done in the "real world" can also be done in the "ideal world". Because the ideal world is secure by definition, so is the real world. Using such a definition is particularly useful to capture the "leakage" of information to the server. This is quite subtle to capture in the real world, but very natural in the ideal world.

In this thesis, we instantiate a general model of computation to capture general quantum computations within converters which ensures that they follow the laws of quantum physics (e.g., excluding that the input-output behavior is signaling). Indeed, without such a restriction, we could not base our statements on results from quantum physics, because an arbitrary physical reality must not respect them, such as cloning of quantum states, signaling, and more. More specifically, in this work, we assume that any converter that interacts classically on its inner interface and outputs a single quantum message on its outer interface can be represented as a sequence of quantum instruments (which is a generalization of CPTP maps taking into account both quantum and classical outputs, a concept introduced by [DL70], see Definition 2.1.3) and constitutes the most general expression of allowed quantum operations. As one example, the representation of a protocol as a sequence of quantum instruments is depicted in Figure 3.4.

Definition 2.1.3 (Quantum Instrument)**.** *A map* $\Lambda : \mathbb{C}^{n\times n} \to \{0,1\}^{m_1} \times \mathbb{C}^{m_2\times m_2}$ *is said to be a quantum instrument if there exists a collection* $\{\mathcal{E}_y\}_{y\in\{0,1\}^{m_1}}$ *of trace-non-increasing completely positive maps such that the sum is trace-preserving (i.e. for any positive operator* ρ*,* $\sum_y \mathcal{E}_y(\rho) = \mathrm{Tr}(\rho)$*), and, if we define* $\rho_y = \frac{\mathcal{E}_y(\rho)}{\mathrm{Tr}(\mathcal{E}_y(\rho))}$*, then* $\Pr[\,\Lambda(\rho) = (y, \rho_y)\,] = \mathrm{Tr}(\mathcal{E}_y(\rho))$.

More precisely, this model takes into account interactive converters (and models the computation in sequential dependent stages). This is similar to if one would in the classical world instantiate the converter by a sequence of classical Turing machines (passing state to each other) [Gol01].

We can now present the General Composition Theorem (Theorem 1 from [MR11]).

Theorem 2.1.4 (General Composition of Resources)**.** *Let $\mathcal{R}$, $\mathcal{S}$ and $\mathcal{T}$ be resources, α, β and id be protocols (where protocol id does not modify the resource it is applied to). Let $\circ$ and $\mid$ denote respectively the sequential and parallel composition of protocols and resources. Then the following implications hold:*

- *The protocols are* sequentially composable*: if $\alpha\mathcal{R} \underset{stat,\epsilon_\alpha}{\approx} \mathcal{S}$ and $\beta\mathcal{S} \underset{stat,\epsilon_\beta}{\approx} \mathcal{T}$ then $(\beta \circ \alpha)\mathcal{R} \underset{stat,\epsilon_\alpha+\epsilon_\beta}{\approx} \mathcal{T}$*
- *The protocols are* context-insensitive*: if $\alpha\mathcal{R} \underset{stat,\epsilon_\alpha}{\approx} \mathcal{S}$ then $(\alpha \mid \mathsf{id})(\mathcal{R} \mid \mathcal{T}) \underset{stat,\epsilon_\alpha}{\approx} (\mathcal{S} \mid \mathcal{T})$*

Combining the two properties presented above yields concurrent composability (the distinguishing advantage cumulates additively as well).

The computational versions of these definitions are obtained by quantifying over quantum polynomial time parties. Composing a statistically-secure protocol with a computationally-secure protocol is possible provided that the simulator for the statistically-secure one runs in expected polynomial time. The resulting protocol is of course only computationally secure.

Comments on the Security Framework. First, we always consider in this work a single Adversary controlling all the corrupted parties. As explained above, it is therefore possible to instantiate all purely classical Resources using any classical protocol which is secure in the Q-UC framework of [Unr10] with the same security guarantees. It is also possible to instantiate them with any classical UC-secure protocol whose security relies on a quantum-hard problem thanks to Theorem 18 (Quantum Lifting Theorem – Computational) from [Unr10].

Also, it is impossible to have fairness of output distribution in the case of a dishonest majority, the malicious parties can always choose to receive their output before the honest players. This is modelled in the resources by a filtered bit f_j at each player's interface, indicating that it receives the output before others. The corrupted players can then decide to make the honest players abort before receiving their output.

2.2 Measurement-Based Quantum Computing

The Measurement-Based Quantum Computing (MBQC) model of computation emerged from the gate teleportation principle. It was shown in [RB01] that any quantum

computation can be implemented by performing single-qubit measurements on a type of entangled states called graph states.

Given a graph $G = (V, E)$, and input and output vertices $I, O \subseteq V$, the corresponding graph state is generated by initialising a qubit in state $|+\rangle$ for each vertex in V and performing entangling operator CZ between qubits whose vertices are linked by an edge in E. The qubits are measured according to an order given by a function $f : O^c \to I^c$ called the flow of the computation.

We define the rotation operator around the Z axis of the Bloch sphere by an angle θ as $\mathsf{Z}(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$ and $|+_\theta\rangle = \mathsf{Z}(\theta)\,|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}\,|1\rangle)$. For approximate universality, we can restrict the set of angles to $\Theta = \left\{\frac{k\pi}{4}\right\}_{k\in\{0,\ldots,7\}}$ [BFK09]. The measurement associated to an angle $\phi \in \Theta$ is given by the basis $|\pm_\phi\rangle$. We consider from now on that this measurement is performed by rotating the state to be measured using the operation $\mathsf{Z}(-\phi)$ and then measuring in the X-basis.

Later measurements may depend on the outcomes of previous measurements. Let $\{\phi(v)\}_{v\in O^c}$ be a set of default measurement angles for non-output qubits. Let $S_X(v)$ and $S_Z(v)$ be respectively the X and Z dependency-sets for qubit v.[3] The measurement result $s(w)$ for qubit $w \in S_X(v) \cup S_Z(v)$ induces Pauli corrections on qubit v which are equivalent to measuring qubit v with corrected angle $\phi'(v) = (-1)^{s_X(v)}\phi(v) + \pi s_Z(v)$, where $s_X(v) = \bigoplus_{w\in S_X(v)} s(w)$ and $s_Z(v) = \bigoplus_{w\in S_Z(v)} s(w)$.

The special case of classical inputs is handled by adding an angle $x(v)\pi$ to the measurement angle $\phi(v)$ of input qubit $v \in I$. Classical outputs correspond to the case where all qubits are measured.

The classical input-output computation is defined by a graph $G = (V, E)$, input and output vertices $I, O \subseteq V$, a set of default measurement angles $\{\phi(v)\}_{v\in V}$ and a flow function $f : O^c \to I^c$. To perform the computation, one generates the graph state associated with G, and performs the measurements with angles $\phi'(v)$ using the default angles and the flow. The outcome is defined by bit-string $\{s(v)\}_{v\in O}$.

Formally, an MBQC computation is defined by a *measurement pattern* as follows.

Definition 2.2.1 (Measurement Pattern)**.** *A* pattern *in the Measurement-Based Quantum Computation model is given by a graph $G = (V, E)$, input and output vertex sets I and O, a flow f which induces a partial ordering of the qubits V, and a set of measurement angles $\{\phi(i)\}_{i\in O^c}$ in the $X - Y$ plane of the Bloch sphere.*

[3] These sets are also given by the flow, see [HEB03; DK06] for details.

Execution of MBQC patterns can be delegated to servers using Protocol 1, thus alleviating the need for the client to own a quantum machine.

Protocol 1 Delegated MBQC Protocol

Client's Inputs: A measurement pattern $(G, I, O, \{\phi(i)\}_{i\in O^c}, f)$ and a quantum register containing the input qubits $i \in I$.
Protocol:

1. The Client sends the graph's description (G, I, O) to the Server;
2. The Client sends its input qubits for positions I to the Server;
3. The Server prepares $|+\rangle$ states for qubits $i \in I^c$;
4. The Server applies a CZ gate between qubits i and j if (i, j) is an edge of G;
5. The Client sends the measurement angles $\{\phi(i)\}_{i\in O^c}$ along with the description of f to the Server;
6. The Server measures the qubits $i \in O^c$ in the order defined by f in the rotated basis $\left|\pm_{\phi'(i)}\right\rangle$ where

$$s_X(i) = \bigoplus_{j\in S_X(i)} b(j),\ s_Z(i) = \bigoplus_{j\in S_Z(i)} b(j), \tag{2.2}$$

$$\phi'_i = (-1)^{s_X(i)}\phi(i) + s_Z(i)\pi, \tag{2.3}$$

   where $b(j) \in \{0, 1\}$ is the measurement outcome for qubit j, with 0 being associated to $\left|+_{\phi'(j)}\right\rangle$, and $S_X(i)$ (resp. $S_Z(i)$) is the X (resp. Z) dependency set for qubit i defined by $S_X(i) = f^{-1}(i)$ (resp. $S_Z(i) = \{j : i \in N_G(f(j))\}$);
7. The Server performs the correction $\mathsf{Z}^{s_Z(i)}\mathsf{X}^{s_X(i)}$ for output qubits $i \in O$, which it sends back to the Client.

2.3 Universal Blind Quantum Computing

If the client is able to perform single qubit preparations and use quantum communication, it can delegate an MBQC pattern blindly [BFK09], meaning that the Server does not learn anything about the computation besides the prepared graph G, the set of outputs O and the order of measurements. This is done using Protocol 2.

Note that if the output of the client's computation is classical, the set O is empty and the client only receives measurement outcomes. The output measurement outcomes $b(i)$ sent by the Server need to be decrypted by the Client according to the equation

Protocol 2 UBQC Protocol

Client's Inputs: A measurement pattern $(G, I, O, \{\phi(i)\}_{i\in O^c}, f)$ and a quantum register containing the input state ρ_C on qubits $i \in I$.

Protocol:

1. The Client sends the graph's description (G, I, O) and the measurement order to the Server;
2. The Client prepares and sends all the qubits in V to the Server:[4]
   (a) For $i \in I$, it chooses a random bit $a(i)$. For $i \in I^c$, it sets $a(i) = 0$.
   (b) For $i \in O$, it chooses a random bit $r(i)$ and sets $\theta(i) = (r(v) + a_N(v))\pi$ where $a_N(i) = \sum_{j\in N_G(i)} a(j)$. For $i \in O^c$, it samples a random $\theta(i) \in \Theta$.
   (c) For $i \in I$, it sends $\bigotimes_{i\in I} \mathsf{Z}_i(\theta(i))\mathsf{X}_i^{a(i)}(\rho_C)$. For $i \in I^c$ it sends $\left|+_{\theta(i)}\right\rangle$.
3. The Server applies a CZ gate between qubits i and j if (i, j) is an edge of G;
4. For all $i \in O^c$, in the order specified by the flow f, the Client computes the measurement angle $\delta(i)$ and sends it to the Server, receiving in return the corresponding measurement outcome $b(i)$:

$$s_X(i) = \bigoplus_{j\in S_X(i)} b(i) \oplus r(i), \; s_Z(i) = \bigoplus_{j\in S_Z(i)} b(i) \oplus r(i), \tag{2.4}$$

$$\delta(i) = (-1)^{a(i)}\phi'(i) + \theta(i) + (r(i) + a_N(i))\pi, \tag{2.5}$$

   where $\phi'(i)$ is computed using Equation (2.3) with the new values of $s_X(i)$ and $s_Z(i)$;
5. The Server sends back the output qubits $i \in O$;
6. The Client applies $\mathsf{Z}_i^{s_Z(i)+r(i)}\mathsf{X}_i^{s_X(i)+a(i)}$ to the received qubits $i \in O$.

$s(j) = b(j) \oplus r(j)$, thus preserving the confidentiality of the output of the computation.

To analyze the security of our protocol later, we will require the following Pauli Twirling Lemma as a way to decompose the actions of an adversary in the blind protocol above. A Pauli twirl occurs when a random Pauli operator is applied (such as an encryption and decryption). The result from the point of view of someone who does not know which Pauli has been used is a state or channel that is averaged over all possible Pauli operators. This has the effect of removing all off-diagonal factors from the operation sandwiched between the two applications of the random Pauli, thus making it

[4] In the original UBQC Protocol from [BFK09], the outputs are prepared by the Server in the $|+\rangle$ state and are encrypted by the computation flow. In the verification protocol in which we will use the UBQC Protocol later, some inputs to auxiliary trap computations may be included in the global output, meaning that all output qubits must also be prepared by the Client. This does not change the security properties of the UBQC Protocol.

a convex combination of Pauli operators.

Lemma 2.3.1 (Pauli Twirling). *Let ρ be an n-qubit mixed state and $\mathsf{Q}, \mathsf{Q}' \in \mathcal{P}_n$ two n qubit Pauli operators. Then, if $\mathsf{Q} \neq \mathsf{Q}'$, we have:*

$$\sum_{\mathsf{P} \in \mathcal{P}_n} \mathsf{P}^\dagger \mathsf{Q} \mathsf{P} \rho \mathsf{P}^\dagger \mathsf{Q}'^\dagger \mathsf{P} = 0. \tag{2.6}$$

On the other hand, the following Resource 1 models the abstract security of the UBQC Protocol 2. It leaks no information to the Server beyond a controlled leak, but allows the Server to modify the output by deviating from the Client's desired computation.

Resource 1 Blind Delegated Quantum Computation

Public Information: Nature of the leakage l_{ρ_C}.
Inputs:

- The Client inputs the classical description of a computation C from subspace $\Pi_{I,C}$ to subspace $\Pi_{O,C}$ and a quantum state ρ_C in $\Pi_{I,C}$.
- The Server chooses whether or not to deviate. This interface is filtered by two control bits (e, c) (set to 0 by default for honest behaviour). If $c = 1$, the Server has an additional input CPTP map F and state ρ_S.

Computation by the Resource:

1. If $e = 1$, the Resource sends the leakage l_{ρ_C} to the Server's interface.
2. If $c = 0$, it outputs $\mathsf{C}(\rho_C)$ at the Client's output interface. Otherwise, it waits for the additional input and outputs $\mathrm{Tr}_S(\mathsf{F}(\rho_{CS}))$ at the Client's interface.

The following Theorem 2.3.2 captures the security guarantees of the UBQC Protocol 2 in the Abstract Cryptography Framework, as expressed in [Dun+14].

Theorem 2.3.2 (Security of Universal Blind Quantum Computation). *The UBQC Protocol 2 perfectly constructs the Blind Delegated Quantum Computation Resource 1 for leak $l_{\rho_C} = (G, O, \preceq_G)$, where $\preceq_G$ is the ordering induced by the flow of the computation.*

2.4 Quantum Verification

Verifiable protocols allow the Client to check that its computation has been done correctly. One way to achieve this is by enlarging the graph used for the computation and to

insert traps. These traps are made from qubits randomly prepared in $|+_\theta\rangle$ states and disconnected from the sub-graph used for performing the desired computation with the help of *dummy qubits* – i.e. randomly initialised qubits sent by the Client in states $\{|0\rangle, |1\rangle\}$. The first verification protocol via trappification was introduced in [FK17]. It was further optimised in the *Verifiable Blind Quantum Computation Protocol* (or VBQC) of [KW17b; XTH20], achieving a linear overhead.

Resource 2 captures the security properties of a blind and verifiable delegated protocol for a given class of computations. It allows a single Client to run a quantum computation on a Server so that the Server cannot corrupt the computation and does not learn anything besides a given leakage l_ρ. We recall the original definition from [Dun+14, Definition 4.2].

Resource 2 Secure Delegated Quantum Computation

Public Information: Nature of the leakage l_{ρ_C}.
Inputs:

- The Client inputs the classical description of a computation C from subspace $\Pi_{I,C}$ to subspace $\Pi_{O,C}$ and a quantum state ρ_C in $\Pi_{I,C}$.
- The Server chooses whether or not to deviate. This interface is filtered by two control bits (e, c) (set to 0 by default for honest behaviour).

Computation by the Resource:

1. If $e = 1$, the Resource sends the leakage l_ρ to the Server's interface; if it receives $c = 1$, the Resource outputs $|\perp\rangle\langle\perp| \otimes |\mathsf{Rej}\rangle\langle\mathsf{Rej}|$ at the Client's output interface.
2. Otherwise, it outputs $\mathsf{C}(\rho_C) \otimes |\mathsf{Acc}\rangle\langle\mathsf{Acc}|$ at the Client's output interface.

Note, that Resource 2 can be seen as a strengthening of Resource 1 by adding verifiability to the blindness guarantees, thereby restricting the power of possibly adversarial servers.

Trappification is not the only known way to construct verification protocols, although it is the approach that will be used for large parts of this thesis. For other quantum verification schemes and concrete protocols, we refer to the survey in [GKK19].

2.5 Notation

Throughout this thesis, we will use the following notations. We denote by $\mathsf{Z}_{\frac{\pi}{2}}$ the set of the 4 angles $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$, and $\mathsf{Z}_{\frac{\pi}{4}} = \{0, \frac{\pi}{4}, ..., \frac{7\pi}{4}\}$ the analogous set of 8 angles. If ρ is

a quantum state, $[\rho]$ is the *classical* representation (as a density matrix) of this state. For a protocol $\mathcal{P} = (P_1, P_2)$ with two interacting algorithms P_1 and P_2 denoting the two participating parties, let $r \leftarrow \langle P_1, P_2 \rangle$ denote the execution of the two algorithms, exchanging messages, with output r. We use the notation $\mathfrak{C}$ to denote the *classical channel* resource, that just forwards classical messages between the two parties. For a set V, $\wp(V)$ is the powerset of V, the set of all subsets of V. For a set $B \subseteq A$, we denote by B^c the complement of B in A, where A will often be the vertex set of a graph and B a subset of vertices, usually input or output locations. For $n \in \mathbb{N}$, the set of all integers from 0 to n included is denoted $[n]$. For a real function $\epsilon(\eta)$, we say that $\epsilon(\eta)$ is *negligible in* η if, for all polynomials $p(\eta)$ and η sufficiently large, we have $\epsilon(\eta) \leq \frac{1}{p(\eta)}$. For a real function $\mu(\eta)$, we say that $\mu(\eta)$ is *overwhelming in* η if there exists a negligible $\epsilon(\eta)$ such that $\mu(\eta) = 1 - \epsilon(\eta)$.

Chapter 3

Classical-Client Delegated Quantum Computing

Secure delegated quantum computing is a two-party cryptographic primitive, where a computationally weak client wishes to delegate an arbitrary quantum computation to an untrusted quantum server in a privacy-preserving manner. Communication via quantum channels is typically assumed such that the client can establish the necessary correlations with the server to securely perform the given task. This has the downside that all these protocols cannot be put to work for the average user unless a reliable quantum network is deployed.

Therefore the question becomes relevant whether it is possible to rely solely on classical channels between client and server and yet benefit from its quantum capabilities while retaining privacy. Classical-client remote state preparation ($\mathsf{RSP}_{\mathsf{CC}}$*) is one of the promising candidates to achieve this because it enables a client, using only classical communication resources, to remotely prepare a quantum state. However, the privacy loss incurred by employing* $\mathsf{RSP}_{\mathsf{CC}}$ *as sub-module to avoid quantum channels is unclear.*

In this work, we investigate this question using the Abstract Cryptography framework by Maurer and Renner [MR11]. We first identify the goal of $\mathsf{RSP}_{\mathsf{CC}}$ *as the construction of ideal* RSP *resources from classical channels and then reveal the security limitations of using* $\mathsf{RSP}_{\mathsf{CC}}$ *in general and in specific contexts:*

1. *We uncover a fundamental relationship between constructing ideal* RSP *resources (from classical channels) and the task of cloning quantum states with auxiliary information. Any classically constructed ideal* RSP *resource must leak to the server the full classical description (possibly in an encoded form) of the generated quantum*

state, even if we target computational security only. As a consequence, we find that the realization of common RSP *resources, without weakening their guarantees drastically, is impossible due to the no-cloning theorem.*

2. *The above result does not rule out that a specific* $\mathsf{RSP}_{\mathsf{CC}}$ *protocol can replace the quantum channel at least in some contexts, such as the Universal Blind Quantum Computing (*UBQC*) protocol of Broadbent et al. [BFK09]. However, we show that the resulting* UBQC *protocol cannot maintain its proven composable security as soon as* $\mathsf{RSP}_{\mathsf{CC}}$ *is used as a subroutine.*

3. *We show that replacing the quantum channel of the above* UBQC *protocol by the* $\mathsf{RSP}_{\mathsf{CC}}$ *protocol QFactory of Cojocaru et al. [Coj+19], preserves the weaker, game-based, security of* UBQC*.*

This chapter is based on the paper "Security limitations of classical-client delegated quantum computing" [Bad+20] which is joint work with Christian Badertscher, Alexandru Cojocaru, Léo Colisson, Elham Kashefi, Atul Mantri, and Petros Wallden, and has been published in the conference proceedings of ASIACRYPT 2020.

3.1 Introduction

The expected rapid advances in quantum technologies in the decades to come are likely to further disrupt the field of computing. To fully realize the technological potential, remote access, and manipulation of data must offer strong privacy and integrity guarantees and currently available quantum cloud platform designs have still a lot of room for improvement.

There is a large body of research that exploits the client-server setting defined in [Chi05] to offer different functionalities, including secure delegated quantum computation [BFK09; MF12; Dun+14; Bro15; Mah18a] [1], verifiable delegated quantum computation [ABE10; RUV12; FK17; HM15; Bro18; FHM18; Tak+18; Mah18b] [2], secure multiparty quantum computation [KP17; KMW17; KW17a], quantum fully homomorphic encryption [BJ15; DSS16]. It turns out that one of the central building blocks is secure *remote state preparation* (RSP) that was first defined in [DKL12]. At a high level, RSP resources enable a client to remotely prepare a quantum state on the

[1] For more details see review of this field in [Fit17]
[2] For more details see recent reviews in [GKK19; Vid20]

server and are, therefore, the natural candidate to replace quantum channel resources in a modular fashion. These resources further appear to enable a large ecosystem of composable protocols [DKL12; Dun+14], including in particular the important *Universal Blind Quantum Computation* (UBQC) [BFK09] protocol used to delegate a computation to a remote quantum server who has no knowledge of the ongoing computation.

However, in most of the above-mentioned works, the users and providers do have access to quantum resources to achieve their goals, in particular to quantum channels in addition to classical communication channels. This might prove to be challenging for some quantum devices, e.g. those with superconducting qubits, and in general, it also restricts the use of these quantum cloud services to users with suitable quantum technology. Motivated by this practical constrain, [Coj+21] introduced a protocol mimicking this remote state preparation resource over a purely *classical* channel (under the assumption that learning with error problem is computationally hard for quantum servers). This is a cryptographic primitive between a fully classical client and a server (with a quantum computer). By the end of the interactive protocol the client has "prepared" remotely on the server's lab, a quantum state (typically a single qubit $|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle)$). This protocol further enjoys some important privacy guarantees with respect to the prepared state.

The important role of such a classical RSP primitive as part of larger protocols – most notably in their role in replacing quantum channels between client and server – stems from their ability to make the aforementioned protocols available to classical users, in particular clients without quantum-capable infrastructure on their end. It is therefore of utmost importance to develop an understanding of this primitive, notably its security guarantees when composed in larger contexts such as in [GV19].

In this chapter, we initiate the study of analyzing classical remote state-preparation from first principles. We thereby follow the Abstract Cryptography (AC) framework [MR11; Mau11] to provide a clean treatment of the RSP primitive from a composable perspective. Armed with such a definition, we then investigate the limitations and possibilities of using classical RSP both in general and in more specific contexts. Using AC is a common approach to analyze classical as well as quantum primitives and their composable security guarantees in general and in related works including [Dun+14; DK16; MK13].

3.1.1 Overview of our Contributions

We present an informal overview of our main results. In this work, we cover the security of $\mathsf{RSP}_{\mathsf{CC}}$, the class of remote state preparation protocols which only use a classical channel, and the use-case that corresponds to its arguably most important application: Universal Blind Quantum Computing (UBQC) protocols with a completely classical client. More specifically, we analyze the security of $\mathsf{UBQC}_{\mathsf{CC}}$, the family of protocols where a protocol in $\mathsf{RSP}_{\mathsf{CC}}$ is used to replace the quantum channel from the original quantum-client UBQC protocol. An example of an RSP resource is the $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$ [3] resource (depicted in Figure 3.1) outputting the quantum state $|+_\theta\rangle$ on its right interface, and the classical description of this state, θ, on its left interface.

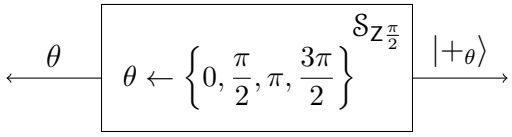


Figure 3.1: Ideal resource $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$

We show in Section 3.2 a wide-ranging limitation to the universally composable guarantees that any protocol in the family $\mathsf{RSP}_{\mathsf{CC}}$ can achieve. The limitation follows just from the relation between (i) the notion of classical realization and (ii) a property we call describability – which roughly speaking measures how leaky an RSP resource is. The limitation directly affects the amount of additional leakage on the classical description of the quantum state. In this way, it rules out a wide set of desirable resources, even against computationally bounded distinguishers.

Theorem 3.2.6 (Security Limitations of $\mathsf{RSP}_{\mathsf{CC}}$). *Any* RSP *resource, realizable by an* $\mathsf{RSP}_{\mathsf{CC}}$ *protocol with security against quantum polynomial-time distinguishers, must leak an encoded, but complete description of the generated quantum state to the server.*

The importance of Theorem 3.2.6 lies in the fact that it is drawing a connection between the composability of an $\mathsf{RSP}_{\mathsf{CC}}$ protocol – a *computational* notion – with the statistical leakage of the ideal functionality it is constructing – an *information-theoretic* notion. This allows us to use fundamental physical principles such as no-cloning or no-signaling in the security analysis of *computationally* secure $\mathsf{RSP}_{\mathsf{CC}}$ protocols. As one

[3] The notation $\mathsf{Z}_{\frac{\pi}{2}}$ denotes the set of the 4 angles $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$.

direct application of this powerful tool, we show that secure implementations of the ideal resource in Figure 3.1 give rise to the construction of a quantum cloner, and are hence impossible.

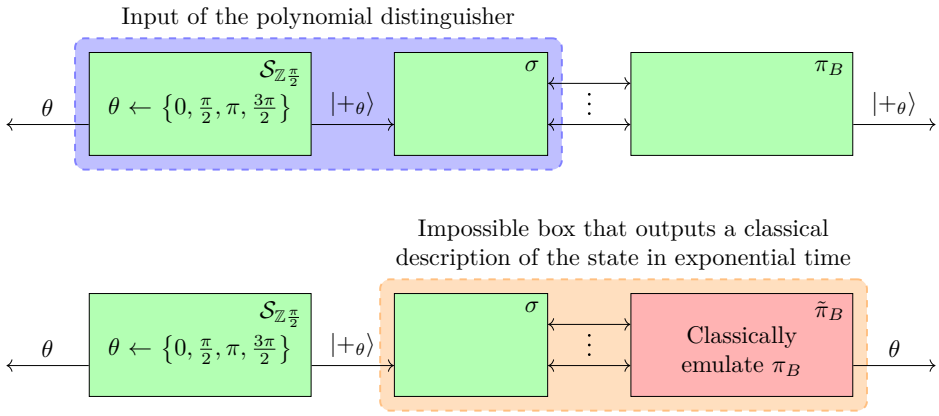


Figure 3.2: Idea of the proof of impossibility of composable $\mathsf{RSP}_{\mathsf{CC}}$, exemplified by the $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$ primitive from Figure 3.5. The green boxes run in polynomial time, while the red box runs in exponential time. $\tilde{\pi}_B$ runs the same computations as π_B by emulating it. In this way, the classical description of the quantum state can be extracted.

Proof sketch. While Theorem 3.2.6 applies to much more general RSP resources having arbitrary behavior at its interfaces and targeting any output quantum state, for simplicity we exemplify the main ideas of our proof for the ideal resource $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$.

The composable security of a protocol realizing $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$ implies, by definition, the existence of a simulator σ which turns the right interface of the ideal resource into a completely classical interface as depicted in Figure 3.2. Running the protocol of the honest server with access to this classical interface allows the distinguisher to reconstruct the quantum state $|+_\theta\rangle$ the simulator received from the ideal resource. Since the distinguisher also has access to θ via the left interface of the ideal resource, he can perform a simple measurement to verify the consistency of the state obtained after interacting with the simulator. By the correctness of the protocol, the obtained quantum state $|+_\theta\rangle$ must therefore indeed comply with θ. We emphasize that this consistency check can be performed efficiently, i.e. by *polynomially-bounded* quantum distinguishers.

Since the quantum state $|+_\theta\rangle$ is transmitted from σ to the distinguisher over a classical channel, the ensemble of exchanged classical messages must contain a complete encoding of the description of the state, θ. A (possibly unbounded) algorithm can hence extract the actual description of the state by means of a classical emulation of the honest

server. This property of the ideal resource is central to our proof technique, we call it *describability*. □

Having a full description of the quantum state produced by $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$ would allow us to clone it, a procedure prohibited by the no-cloning theorem. We conclude that the resource $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$ cannot be constructed from a classical channel only.

One could attempt to modify the ideal resource, to incorporate such an extensive leakage, which is necessary as the above proof implies. However, this yields an ideal resource that is actually not a useful idealization or abstraction of the real world (because it is fully leaky) which puts in question whether they are at all useful in a composable analysis. Consider for example constructions of composite protocols that utilize the (non-leaky) ideal resource as a sub-module. These constructions require a fresh security analysis if the sub-module is replaced by any leaky version of it, but since the modified resource is very specific and must mimic its implementation (in terms of leakage) it appears that this replacement does not give any benefit compared to directly using the implementation as a subroutine and then examining the compsoable security of the combined protocol as a whole. This latter way is therefore examined next. More precisely, we might still be able to use $\mathsf{RSP}_{\mathsf{CC}}$ protocols as a subroutine in other, specific protocols, and expect the overall protocol to still construct a useful ideal functionality. The protocol family $\mathsf{UBQC}_{\mathsf{CC}}$ is such an application. Unfortunately, as we show in Section 3.3, $\mathsf{UBQC}_{\mathsf{CC}}$ fails to provide the expected composable security guarantees once classical remote state preparation is used to replace the quantum channel from client to server (where composable security for UBQC refers to the goal of achieving the established ideal functionality of [Dun+14] which we recall in Section 3.3). This holds even if the distinguisher is computationally bounded.

Theorem 3.3.10 (Impossibility of $\mathsf{UBQC}_{\mathsf{CC}}$). *No* $\mathsf{RSP}_{\mathsf{CC}}$ *protocol can replace the quantum channel in the* UBQC *protocol while preserving composable security.*

Proof sketch. We first show that the existence of any composable $\mathsf{UBQC}_{\mathsf{CC}}$ protocol (in the sense of achieving the ideal UBQC resource) implies the existence of a composable *single-qubit* $\mathsf{UBQC}_{\mathsf{CC}}$ protocol. In turn, the impossibility of composable single-qubit $\mathsf{UBQC}_{\mathsf{CC}}$ protocols is then proven in two steps. First, we show that single-qubit $\mathsf{UBQC}_{\mathsf{CC}}$ protocols can, in fact, be turned into RSP protocols. This allows us to employ the toolbox we developed before on RSP protocols. As a second step, we deduce that an RSP protocol of this specific kind (that leaks the classical description, even in the form

of an encoded message) would violate the no-signaling principle, thereby showing that a composable $\mathsf{UBQC}_{\mathsf{CC}}$ protocol could not have existed in the first place. □

Finally in Section 3.4, we show that the protocol family $\mathsf{RSP}_{\mathsf{CC}}$ is not trivial with respect to privacy guarantees. It contains protocols with reasonably restricted leakage that can be used as subroutines in specific applications resulting in combined protocols that offer a decent level of security. Specifically, we prove the blindness property of QF-UBQC, a concrete $\mathsf{UBQC}_{\mathsf{CC}}$ protocol that consists of the universal blind quantum computation (UBQC) protocol of [BFK09] and the specific LWE-based remote state preparation ($\mathsf{RSP}_{\mathsf{CC}}$) protocol from [Coj+19]. This yields the first provably secure $\mathsf{UBQC}_{\mathsf{CC}}$ protocol from standard assumptions with a classical RSP protocol as a subroutine.

Theorem 3.4.9 (Game-Based Security of QF-UBQC). *The universal blind quantum computation protocol with a classical client* $\mathsf{UBQC}_{\mathsf{CC}}$ *that combines the* $\mathsf{RSP}_{\mathsf{CC}}$ *protocol of* [Coj+19] *and the* UBQC *protocol of* [BFK09] *is adaptively blind in the game-based setting. We call this protocol* QF-UBQC. *This protocol is secure under standard assumptions.*

The statement of Theorem 3.4.9 can be summarized as follows: No malicious (but computationally bounded) server in the QF-UBQC protocol could distinguish between two runs of the protocol performing different computations. This holds even when it is the adversary that chooses the two computations that he will be asked to distinguish. The security is achieved in the plain model, i.e., without relying on additional setup such as a measurement buffer. The protocol itself is a combination of UBQC with the QFactory protocol. For every qubit that the client would transmit to the server in the original UBQC protocol, QFactory is invoked as a subprocedure to the end of remotely preparing the respective qubit state on the server over a classical channel.

Proof sketch. By a series of games, we show that the real protocol on a single qubit is indistinguishable from a game where the adversary guesses the outcome of a hidden coin flip. We generalize this special case to the full protocol on graphs with a polynomial number of qubits by induction over the size of the graph. □

3.1.2 Related Work

While $\mathsf{RSP}_{\mathsf{CC}}$ was first introduced in [Coj+21] (under a different terminology), (game-based) security was only proven against weak (honest-but-curious) adversaries. Security against malicious adversaries was proven for a modified protocol in [Coj+19][4], this protocol, called *QFactory*, is the basis of the positive results in this work. In parallel [GV19] gave another protocol that offers a stronger notion of *verifiable* $\mathsf{RSP}_{\mathsf{CC}}$ and proved the security of their primitive in the AC framework. The security analysis, however, requires an assumption of *measurement buffer* resource in addition to the classical channel to construct a verifiable $\mathsf{RSP}_{\mathsf{CC}}$. Our result confirms that the measurement buffer resource is a strictly non-classical assumption.

In the information-theoretic setting with perfect security[5], the question of secure delegation of quantum computation with a completely classical client was first considered in [MK14]. The authors showed a negative result by presenting a *scheme-dependent* impossibility proof. This was further studied in [DK16; Aar+19] which showed that such a classical delegation would have implications in computational complexity theory. To be precise, [Aar+19] conjecture that such a result is unlikely by presenting an oracle separation between BQP and the class of problems that can be classically delegated with perfect security (which is equivalent to the complexity class $\mathrm{NP}/\mathrm{poly} \cap \mathrm{coNP}/\mathrm{poly}$ as proven by [AFK87]). On the other hand, a different approach to secure delegated quantum computation with a completely classical client, without going via the route of $\mathsf{RSP}_{\mathsf{CC}}$, was also developed in [Man+17] where the server is unbounded and in [Mah18a; Bra18] with the bounded server. The security was analysed for the overall protocol (rather than using a module to replace quantum communication). It is worth noting that [Man+17] is known to be not composable secure in the Abstract Cryptography framework [Man19].

3.2 Impossibility of Composable Classical RSP

In this section, we first define the general notion of what RSP tries to achieve in terms of resources and subsequently quantify information that an ideal RSP resource must leak at its interface to the server even if the distinguisher is computationally bounded.

[4] In [Coj+19] a verifiable version of $\mathsf{RSP}_{\mathsf{CC}}$ was also given, but security was not proven in full generality.

[5] By perfect security we mean at most input size is allowed to be leaked

One would expect, that against bounded distinguisher, the resource can express clear privacy guarantees, which we prove cannot be the case.

The reason is roughly as follows: assuming that there exists a simulator making the ideal resource indistinguishable from the real protocol, we can exploit this fact to construct an algorithm that can classically describe the quantum state given by the ideal resource. It is not difficult to verify that there could exist an inefficient algorithm (i.e. with exponential run-time) that achieves such a task. We show that even a computationally bounded distinguisher can distinguish the real protocol from the ideal protocol whenever a simulator's strategy is independent of the classical description of the quantum state. This would mean that for an RSP protocol to be composable there must exist a simulator that possesses at least a classical transcript encoding the description of a quantum state. This fact coupled with the quantum no-cloning theorem implies that the most meaningful and natural RSP resources cannot be realized from a classical channel alone. We finally conclude the section by looking at the class of imperfect (describable) RSP resources which avoid the no-go result at the price of being "fully-leaky", not standard, and having an unfortunately unclear composable security.

3.2.1 Remote State Preparation and Describable Resources

We first introduce, based on the standard definition in the Abstract Cryptography framework, the notion of *correctness* and *security* of a two-party protocol which constructs (realizes) a resource from a *classical* channel $\mathcal{C}$.

Definition 3.2.1 (Classically-Realizable Resource)**.** *An ideal resource $\mathcal{S}$ is said to be* ε-classically-realizable *if it is realizable (in the sense of Definition 2.1.2) from a* classical *channel, i.e. if there exists a protocol $\boldsymbol{\pi} = (\pi_A, \pi_B)$ between two parties (interacting classically) such that:*

$$\mathcal{C} \xrightarrow[\varepsilon]{\boldsymbol{\pi}} \mathcal{S} \tag{3.1}$$

We would like to point out that since Alice is honest, this definition incorporates already the case when Alice and Bob share purely classical resources that are achievable by Alice emulating the resource and sending Bob's output over a classical channel.
A simple ideal prototype that captures the goal of a RSP protocol could be phrased as follows: the resource outputs a quantum state (chosen from a set of states) on one interface and classical description of that state on the other interface to the client. For

our purposes, this view is too narrow and we want to generalize this notion. For instance, a resource could accept some inputs from the client or interact with the server and be powerful enough to comply with the above basic behavior if both follow the protocol. We would like to capture that any resource can be seen as an RSP resource as soon as we fix a way to efficiently convert the client and server interfaces to comply with the basic prototype. To make this formal, we need to introduce some converters that will witness this:

1. A converter $\mathcal{A}$ will output, after interacting with the ideal resource[6], a classical description $[\rho]$ which is one of the following:
   (a) A density matrix (positive and with trace 1) corresponding to a quantum state ρ.
   (b) The null matrix, which is useful to denote the fact that we detected some deviation that should not happen in an honest run.
2. A converter $\mathcal{Q}$, whose goal is to output a quantum state ρ' as close as possible to the state ρ output by $\mathcal{A}$.
3. A converter $\mathcal{P}$, whose goal is to output a classical description $[\rho']$ of a quantum state ρ' which is on average "close" to ρ.

An RSP must meet two central criteria:

1. Accuracy of the classical description of the obtained quantum state: We require that the quantum state ρ described by $\mathcal{A}$'s output is close to $\mathcal{Q}$'s output ρ'. This is to be understood in terms of the trace distance.
2. Purity of the obtained quantum state: Since the RSP resource aims to replace a noise-free quantum channel, it is desirable that the quantum state output by $\mathcal{Q}$ admit a high degree of purity, i.e. more formally, that $\mathrm{Tr}\left(\rho'^2\right)$ be close to one. Since ρ' is required to be close to ρ, this implies a high purity of ρ as well.

It turns out that these two conditions can be unified and equivalently captured requiring that the quantity $\mathrm{Tr}(\rho\rho')$ is close to one. A rigorous formulation of this claim and its proof is provided by Lemma 3.5.3.

[6] $\mathcal{A}$ is allowed to interact with the (ideal) resource in a non-trivial manner. However, $\mathcal{A}$ will often be the trivial converter in the sense that it simply forwards the output of the ideal resource, or – when the resource waits for a simple activation input – picks some admissible value as input to the ideal resource and forwards the obtained description to its outer interface.

We can also gain a more operational intuition of the notion of RSP by considering that an RSP resource (together with $\mathcal{A}$ and $\mathcal{Q}$) can be seen, not only as a box that produces a quantum state together with its description but also as a box whose accuracy can be easily *tested*[7]. For example, if such a box produces a state ρ', and pretends that the description of that state corresponds to $|\phi\rangle$ (i.e. $[\rho] = [|\phi\rangle\langle\phi|]$), then the natural way to test it would be to measure ρ' by doing a projection on $|\phi\rangle$. This test would pass with probability $p_s := \langle\phi|\rho'|\phi|\phi|\rho'|\phi\rangle$, and therefore if the box is perfectly accurate (i.e. if $\rho' = |\phi\rangle\langle\phi|$), the test will always succeed. However, when ρ' is far from $|\phi\rangle\langle\phi|$, this test is unlikely to pass, and we will have $p_s < 1$. We can then generalise this same idea for arbitrary (eventually not pure) states by remarking that $p_s = \langle\phi|\rho'|\phi|\phi|\rho'|\phi\rangle = \mathrm{Tr}(|\phi\rangle\langle\phi|\,\rho') = \mathrm{Tr}(\rho\rho')$. Indeed, this last expression corresponds[8] exactly to the probability of outputting E_0 when measuring the state ρ' according to the POVM $\{E_0 := \rho, E_1 := I - \rho\}$, and since the classical description of ρ is known, it is possible to perform this POVM and test the (average) accuracy of our box. This motivates the following definition for general RSP resources.

Definition 3.2.2 (RSP resources)**.** *A resource $\mathcal{S}$ is said to be a* remote state preparation resource *within ε with respect to converters $\mathcal{A}$ and $\mathcal{Q}$ if the following three conditions hold: (1) both converters output a single message at the outer interface, where the output $[\rho]$ of $\mathcal{A}$ is classical and is either a density matrix or the null matrix, and the output ρ' of $\mathcal{Q}$ is a quantum state; (2) the equation:*

$$\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{A}\mathcal{S}\vdash\mathcal{Q}}\left[\,\mathrm{Tr}(\rho\rho')\,\right] \geq 1-\varepsilon \tag{3.2}$$

is satisfied, where the probability is taken over the randomness of $\mathcal{A}$, $\mathcal{S}$ and $\mathcal{Q}$, and finally, (3) for all the possible outputs $[\rho]$ of $([\rho],\rho') \leftarrow \mathcal{A}\mathcal{S} \vdash \mathcal{Q}$, if we define $E_0 = \rho$, $E_1 = I - \rho$, then the POVM $\{E_0, E_1\}$ must be efficiently implementable[9] by any distinguisher.

Whenever we informally speak of a resource $\mathcal{S}$ as being an RSP resource, this has to be understood always in a context where the converters $\mathcal{A}$ and $\mathcal{Q}$ are fixed.

Describable resources. So far, we have specified that a resource qualifies as an RSP resource if, when all parties follow the protocol, we know how to compute a quantum

[7] This testable property will be of great importance in our argument later.

[8] Note that it also turns out to be equal to the (squared) fidelity between ρ and ρ' when ρ is pure.

[9] We could also define a similar definition when this POVM can only be approximated (for example because the distinguishers can only perform quantum circuits using a finite set of gates) and the theorems would be similar, up to this approximation, but for simplicity we will stick to that setting.

state on the right interface and classical description of a "close" state on the other interface. A security-related question now is, if it is also possible to extract (possibly inefficiently) from the right interface a *classical* description of a quantum state that is close to the state described by the client. If we find a converter $\mathcal{P}$ doing this, we would call the (RSP) resource *describable*. The following definition captures this.

Definition 3.2.3 (Describable Resource)**.** *Let $\mathcal{S}$ be a resource and $\mathcal{A}$ a converter outputting a single classical message $[\rho]$ on its outer interface (either equal to a density matrix or the null matrix). Then we say that $(\mathcal{S}, \mathcal{A})$ is ε-*describable *(or, equivalently, that $\mathcal{S}$ is describable within ε with respect to $\mathcal{A}$) if there exists a (possibly unbounded) converter $\mathcal{P}$ (outputting a single classical message $[\rho']$ on its outer interface representing a density matrix) such that:*

$$\mathop{\mathbb{E}}_{([\rho],[\rho'])\leftarrow\mathcal{A}\mathcal{S}\mathcal{P}} [\,\mathrm{Tr}(\rho\rho')\,] \geq 1-\varepsilon \tag{3.3}$$

(the expectation is taken over the randomness of $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{P}$).

Reproducible converters. In the proof of our first result, we will encounter a crucial decoding step. Roughly speaking, the core of this decoding step is the ability to convert the classical interaction with a client, which can be seen as an arbitrary encoding of a quantum state, back into an explicit representation of the state prepared by the server. The ability of such a conversion can be phrased by the following definition.

Definition 3.2.4 (Reproducible Converter)**.** *A converter π that outputs (on the right interface) a quantum state ρ is said to be reproducible if there exists a (possibly inefficient) converter $\tilde{\pi}$ such that:*

1. *the outer interface of $\tilde{\pi}$ outputs only a classical message $[\rho']$*
2. *the converter π is perfectly indistinguishable from $\tilde{\pi}$ against any unbounded distinguisher $D \in \mathcal{D}^u$, up to the conversion of the classical messages $[\rho']$ into a quantum state ρ'. More precisely, if we denote by $\mathcal{T}$ the converter that takes as input on its inner interface a classical description $[\rho']$ of a quantum state and outputs that quantum state ρ' (as depicted in Figure 3.3), we have:*

$$\mathcal{C}\pi \approx_0^{\mathcal{D}^u} \mathcal{C}\tilde{\pi}\mathcal{T} \tag{3.4}$$

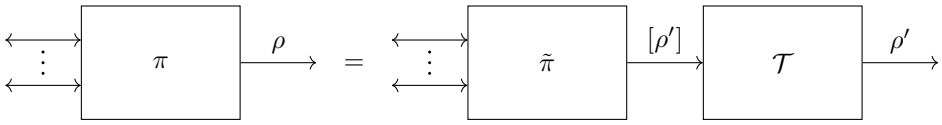


Figure 3.3: Reproducible converter.

Classical communication and reproducibility. We see that in general, being reproducible is a property that stands in conflict with the quantum no-cloning theorem. More precisely, the ability to reproduce implies that there is a way to extract knowledge of a state sufficient to clone it. However, whenever communication is classical, quite the opposite is true. This is formalized in the following lemma. Intuitively, it says that in the principle it is always possible to compute the exact description of the state from the classical transcript and the *quantum instruments* (circuit) used to implement the action of the converter, where an instrument is a generalized CPTP map which allows a party to output both a quantum and a classical state and is formalized more precisely in Definition 2.1.3. Recall that this is the most general way of representing a quantum operation.

In the proof, we just need to assume that π interacts (classically) with the inner interface first, and finally outputs a quantum state on the outer interface, so for simplicity we will stick to that setting. In this way we can decompose π as depicted in Figure 3.4 using the following notation:

$$\pi := (\pi_i)_i \tag{3.5}$$

Each π_i represents a round, and we denote with $(y_i, \rho_{i+1}) \leftarrow \pi_i(x_i, \rho_i)$ the output of the i-th round, assuming that $x_i \in \{0,1\}^{l_i}$ is a classical input message sent from the inner interface, ρ_i is the internal quantum state (density matrix) after round $i-1$, ρ_{i+1} is the internal state after round i, and $y_i \in \{0,1\}^{l'_i} \cup \perp$ is a classical message, sent to the inner interface when $y_i \neq \perp$. For the first protocol, we set $\rho_0 = (1)$, which is the trivial density matrix of dimension 1. Moreover, when $y_i = \perp$, we do not send any message to the inner interface and instead we send ρ_{i+1} to the outer interface and we stop the protocol. Note that if we want to let π send the first message instead of receiving it, we can set $x_0 = \perp$, and similarly, if the last message is sent instead of received, we can add one more round where we set $x_{n+1} = \perp$.

Now, we can prove that a party, that produces a quantum state at the end of a protocol with exclusively classical communication, is reproducible:

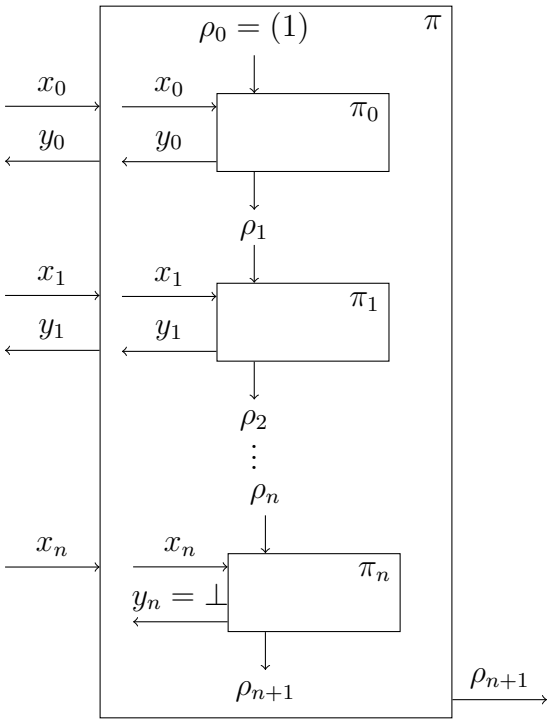


Figure 3.4: Representation of an interactive protocol π as a sequence of quantum instruments.

Lemma 3.2.5. *Let $\pi = (\pi_i)_i$ (using the notation introduced (3.5)) be a converter such that:*

1. *it receives and sends only classical messages from the inner interfaces*
2. *it outputs at the end a quantum state on the outer interface*
3. *each π_i is a quantum instrument*

then π is reproducible.

Proof. The intuition behind the proof is to argue that because the only interactions with the outside world are classical as seen from Figure 3.4, the internal state of π can always be computed (in exponential time) manually.
More precisely, for all i, because π_i is a quantum instrument, there exists a set $\{\mathcal{E}_{y_i}\}$ of maps having the properties defined in Definition 2.1.3. And because for all y_i, $\mathcal{E}_{y_i}$ is completely positive, there exists a finite set of matrices $\{B_k^{(i,y_i)}\}_k$, known as Kraus operators, such that we have for all ρ (and in particular for $\rho = |x_i\rangle\langle x_i| \otimes \rho_i$):

$$\mathcal{E}_{y_i}(\rho) = \sum_k B_k^{(i,y_i)} \rho B_k^{(i,y_i)\dagger} \tag{3.6}$$

Therefore, for all x_i, ρ_i and y_i, we have with probability $p_{y_i} := \mathrm{Tr}(\mathcal{E}_{y_i}(|x_i\rangle\langle x_i| \otimes \rho_i))$:

$$\pi_i(x_i, \rho_i) = (y_i, \mathcal{E}_{y_i}(|x_i\rangle\langle x_i| \otimes \rho_i)) \tag{3.7}$$

$$= (y_i, \underbrace{\sum_k B_k^{(i,y_i)}(|x_i\rangle\langle x_i| \otimes \rho_i)B_k^{(i,y_i)\dagger}}_{\rho_{i+1}}) \tag{3.8}$$

We remark that if we know $[\rho_i]$, the coefficients of the matrix ρ_i, then for all y_i we can compute the probability p_{y_i} of outputting y_i, and the corresponding $[\rho_{i+1}]$, (the coefficients of the matrix ρ_{i+1}) by just doing the above computation. So to construct $\tilde{\pi}$ (using notations from Definition 3.2.4) we do as follows:

- first, for all i we construct $\tilde{\pi}_i$, which on input $(x_i, [\rho_i])$ outputs $(y_i, [\rho_{i+1}])$ with probability p_{y_i} using the formula (3.8).
- then, we define $\tilde{\pi}$ as $(\tilde{\pi}_i)$ with $[\rho_0] = (1)$.

Then, we trivially have $\mathcal{C}\pi \approx_0 \mathcal{C}\tilde{\pi}\mathcal{T}$, even for unbounded distinguishers, because $\tilde{\pi}$ is exactly the same as π, except that the representations of the quantum states in $\tilde{\pi}$ are matrices, while they are actual quantum states in π. Therefore, adding $\mathcal{T}$ (which turns any $[\rho_i]$ into ρ_i) on the outer interface (which is the only interface that sends a classical state $[\rho_i]$) gives us $\pi \approx_0 \mathcal{C}\tilde{\pi}\mathcal{T}$. □

3.2.2 Classically-Realizable RSP are Describable

In this section we show our main result about remote state preparation resources, which interestingly links a constructive notion (*composability*) with respect to a computational notion with an information theoretic property (*describability*).

This implies directly the *impossibility result* regarding the existence of non-describable $\mathsf{RSP}_{\mathsf{CC}}$ composable protocols (secure against *bounded* BQP distinguishers). While this theorem does not rule out all the possible RSP resources, it shows that most "*useful*" RSP resources are impossible. Indeed, the describable property is usually not a desirable property, as it means that an unbounded adversary could learn the description of the state he received from an ideal resource. To illustrate this theorem, we will see in the Section 3.2.3 some examples showing how this result can be used to prove the impossibility of classical protocols implementing some specific resources, and in Section 3.2.4 we will see some example of "imperfect" resources escaping the impossibility result.

Theorem 3.2.6 (Classically-Realizable RSP are Describable)**.** *If an ideal resource $\mathcal{S}$ is both an ε_1-remote state preparation with respect to some $\mathcal{A}$ and $\mathcal{Q}$ and ε_2-classically-realizable (including against only polynomially bounded distinguishers), then it is $(\varepsilon_1 + 2\varepsilon_2)$-describable with respect to $\mathcal{A}$. In particular, if $\varepsilon_1 = \mathsf{negl}(n)$ and $\varepsilon_2 = \mathsf{negl}(n)$, then $\mathcal{S}$ is describable within a negligible error $\varepsilon_1 + 2\varepsilon_2 = \mathsf{negl}(n)$.*

Proof. Let $\mathcal{S}$ be an ε_1-remote state preparation resource with respect to $(\mathcal{A}, \mathcal{Q})$ which is ε_2-classically-realizable. Then there exist π_A, π_B, σ, such that:

$$\mathop{\mathbb{E}}_{([\rho],\rho') \leftarrow \mathcal{A}\mathcal{S}\vdash\mathcal{Q}} [\,\mathrm{Tr}(\rho\rho')\,] \geq 1 - \varepsilon_1 \tag{3.9}$$

$$\pi_A \mathcal{C} \pi_B \approx_{\varepsilon_2} \mathcal{S} \vdash \tag{3.10}$$

and

$$\pi_A \mathcal{C} \approx_{\varepsilon_2} \mathcal{S}\sigma \tag{3.11}$$

Now, using (3.10), we get:

$$\mathcal{A}\pi_A \mathcal{C} \pi_B \mathcal{Q} \approx_{\varepsilon_2} \mathcal{A}\mathcal{S} \vdash \mathcal{Q} \tag{3.12}$$

So it means that we can't distinguish between $\mathcal{A}\mathcal{S} \vdash \mathcal{Q}$ and $\mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}$ with an advantage better than ε_2 (i.e. with probability better than $\frac{1}{2}(1 + \varepsilon_2)$). But, if we construct the following distinguisher, that runs $([\rho], \rho') \leftarrow \mathcal{A}\mathcal{S} \vdash \mathcal{Q}$, and then measures ρ' using the POVM $\{E_0, E_1\}$ (possible because this POVM is assumed to be efficiently implementable by distinguishers in $\mathcal{D}$), with $E_0 = [\rho]$ and $E_1 = I - [\rho]$ (which is possible because we know the classical description of ρ, which is positive and smaller than I, even when $[\rho] = 0$), we will measure E_0 with probability $1 - \varepsilon_1$. So it means that by replacing $\mathcal{A}\mathcal{S} \vdash \mathcal{Q}$ with $\mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}$, the overall probability of measuring E_0 needs to be close to $1 - \varepsilon_1$. More precisely, we need to have:

$$\mathop{\mathbb{E}}_{([\rho],\rho') \leftarrow \mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}} [\,\mathrm{Tr}(\rho\rho')\,] \geq 1 - \varepsilon_1 - \varepsilon_2 \tag{3.13}$$

Indeed, if the above probability is smaller than $1 - \varepsilon_1 - \varepsilon_2$, then we can define a distinguisher that outputs 0 if he measures E_0, and 1 if he measures E_1, and his

probability of distinguishing the two distributions would be equal to:

$$\frac{1}{2}\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{AS}\vdash\mathcal{Q}}[\,\mathrm{Tr}(\rho\rho')\,]+\frac{1}{2}\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}}[\,\mathrm{Tr}((I-\rho)\rho')\,] \tag{3.14}$$

$$>\frac{1}{2}\left((1-\varepsilon_1)+1-(1-\varepsilon_1-\varepsilon_2)\right) \tag{3.15}$$

$$=\frac{1}{2}(1+\varepsilon_2) \tag{3.16}$$

So this distinguisher would have an advantage greater than ε_2, which is in contradiction with (3.12).

Using a similar argument and (3.10), we have:

$$\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{AS}\sigma\pi_B\mathcal{Q}}[\,\mathrm{Tr}(\rho\rho')\,]\geq 1-\varepsilon_1-2\varepsilon_2 \tag{3.17}$$

We will now use $\pi_B\mathcal{Q}$ to construct a $\mathcal{B}$ that can describe the state given by the ideal resource. To do that, because $\pi_B\mathcal{Q}$ interacts only classically with the inner interface and outputs a single quantum state on the outer interface, then according to Lemma 3.2.5, $\pi_B\mathcal{Q}$ is reproducible, i.e. there exists[10] $\mathcal{B}$ such that $\mathcal{C}\pi_B\mathcal{Q}\approx_0\mathcal{CBT}$. Therefore[11], we have:

$$\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{AS}\sigma\mathcal{BT}}[\,\mathrm{Tr}(\rho\rho')\,]\geq 1-\varepsilon_1-2\varepsilon_2 \tag{3.18}$$

But because $\mathcal{T}$ simply converts the classical description $[\rho']$ into ρ', we also have:

$$\mathop{\mathbb{E}}_{([\rho],[\rho'])\leftarrow\mathcal{AS}\sigma\mathcal{B}}[\,\mathrm{Tr}(\rho\rho')\,]\geq 1-\varepsilon_1-2\varepsilon_2 \tag{3.19}$$

After defining $\mathcal{P}=\sigma\mathcal{B}$, we have that $\mathcal{S}$ is $(\varepsilon_1+2\varepsilon_2)$-describable, which ends the proof. □

3.2.3 RSP Resources Impossible to Realize Classically

In the last section we proved that if an RSP functionality is classically-realizable (secure against polynomial quantum distinguishers), then this resource is describable by an unbounded adversary having access to the right interface of that resource.

Our main result in the previous section directly implies that as soon as there exists *no unbounded* adversary that, given access to the right interface, can find the classical

[10] Note that here $\mathcal{B}$ is not efficient anymore, so that's why in the describable definition we don't put any bound on $\mathcal{B}$, but of course the proof does apply when the distinguisher is polynomially bounded.

[11] Indeed, we also have in particular $\mathcal{AS}\sigma\mathcal{C}\pi_B\mathcal{Q}\approx_0\mathcal{AS}\sigma\mathcal{CBT}$, and because $\mathcal{C}$ is a neutral resource [MR11, Sec. C.2] we can remove $\mathcal{C}$.

description given on the left interface, then the RSP resource is *impossible* to classically realize (against *bounded* BQP distinguishers). Very importantly, this no-go result shows that the *only* type of RSP resources that can be classically realized are the ones that *leak* on the right interface enough information to allow an (possibly unbounded) adversary to determine the classical description given on the left interface. From a security point of view, this property is highly non-desirable, as the resource must leak the *secret description* of the state at least in *some representation*.

In this section we present some of these RSP resources that are impossible to classically realize.

Definition 3.2.7 (Ideal Resource $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$)**.** *$\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$ is the verifiable* RSP *resource (*RSP *which does not allow any deviation from the server), that receives no input, that internally picks a random $\theta \leftarrow \mathsf{Z}_{\frac{\pi}{2}}$, and that sends θ on the left interface, and $|+_\theta\rangle$ on the right interface as shown in Figure 3.5.*

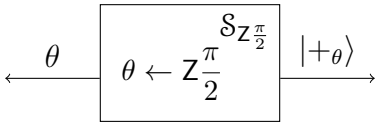


Figure 3.5: Ideal resource $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$.

Lemma 3.2.8. *There exists a universal constant $\eta > 0$, such that for all $0 \leq \varepsilon < \eta$ the resource $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$ is not ε-classically-realizable.*

Proof. This proof is at its core a direct consequence of quantum no-cloning: If we define $\mathcal{A}(\theta) := [|+_\theta\rangle\langle+_\theta|]$ ($\mathcal{A}$ just converts θ into its classical density matrix representation) and $\mathcal{Q}$ the trivial converter that just forwards any message, then $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$ is a 0-remote state preparation resource with respect to $\mathcal{A}$ and $\mathcal{Q}$ because:

$$\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{A}\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}\vdash\mathcal{Q}}\left[\,\mathrm{Tr}(\rho\rho')\,\right] = \frac{1}{4}\sum_{\theta\in\mathsf{Z}_{\frac{\pi}{2}}}\mathrm{Tr}(|+_\theta\rangle\langle+_\theta|\,|+_\theta\rangle\langle+_\theta|) = 1 \geq 1 - 0 \tag{3.20}$$

Then, we remark also that there exists a constant $\eta > 0$ such that for all $\delta < \eta$, $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$ is not δ-describable with respect to $\mathcal{A}$.

Indeed, it is first easy to see that $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$ is not 0-describable with respect to $\mathcal{A}$. Indeed, we can assume by contradiction that there exists $\mathcal{P}$ such that:

$$\mathop{\mathbb{E}}_{([\rho],[\rho'])\leftarrow\mathcal{A}\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}\mathcal{P}}\left[\,\mathrm{Tr}(\rho\rho')\,\right] = 1 \tag{3.21}$$

Then, because $\rho = |+_\theta\rangle\langle+_\theta|$ is a pure state, $\mathrm{Tr}(\rho\rho')$ corresponds to the fidelity of ρ and ρ', so $\mathrm{Tr}(\rho\rho') = 1 \Leftrightarrow \rho = \rho'$. But this is impossible because $\mathcal{P}$ just has a quantum state ρ as input, and if he can completely describe this quantum state then he can actually clone perfectly the input state with probability 1. But because the different possible values of ρ are not orthogonal, this is impossible due to the no-cloning theorem.

Moreover, it is also not possible to find a sequence $(\mathcal{P}^{(n)})_{n\in N}$ of CPTP maps that produces two copies of ρ with a fidelity arbitrary close to 1 (when $n \to \infty$), because CPTP maps are compact and the fidelity is continuous.
Therefore, there exists a constant $\eta > 0$,[12] such that:

$$\mathop{\mathbb{E}}_{([\rho],[\rho'])\leftarrow\mathcal{A}\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}\mathcal{P}} [\,\mathrm{Tr}(\rho\rho')\,] < 1 - \eta \tag{3.22}$$

Now, by contradiction, we assume that $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$ is ε-classically-realizable. Because $\lim_{n\to\infty} \varepsilon(n) = 0$, there exists $N \in \mathbb{N}$ such that $\varepsilon(N) < \eta/2$. So, using Theorem 3.2.6, $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$ is $2\varepsilon(N)$-describable with respect to $\mathcal{A}$, which contradicts $2\varepsilon(N) < \eta$. □

Next, we describe RSP_V, a variant of $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$ introduced in [GV19]. In the latter, RSP_V, the adversary can make the resource abort, that the set of output states is bigger, and that the client can partially choose the basis of the output state. Similar to the $\mathcal{S}_{\mathsf{Z}\frac{\pi}{2}}$, we prove that classically-realizable RSP_V is not possible. Before going into the details of the no-go result, we formalize the ideal resource for a verifiable remote state preparation, RSP_V, below.

Definition 3.2.9 (Ideal Resource RSP_V, See [GV19])**.** *The ideal verifiable remote state preparation resource,* RSP_V*, takes an input* $W \in \{X, Z\}$ *on the left interface, but no honest input on the right interface. The right interface has a filtered functionality that corresponds to a bit* $c \in \{0, 1\}$*. When* $c = 1$*,* RSP_V *outputs error message* ERR *on both the interfaces, otherwise:*

1. *if* $W = Z$ *the resource picks a random bit* b *and outputs* $b \in \mathsf{Z}_2$ *to the left interface and a computational basis state* $|b\rangle\langle b|$ *to the right interface;*

2. *if* $W = X$ *the resource picks a random angle* $\theta \in \mathsf{Z}_{\frac{\pi}{4}}$ *and outputs* θ *to the left interface and a quantum state* $|+_\theta\rangle\langle+_\theta|$ *to the right interface.*

[12] Note that for finding a more precise bound for η, it is possible to use Semidefinite Programming (SDP), or the method presented in [KRK12, p. 2]. However in our case it is enough to say that $\varepsilon > 0$ as we are interested only in asymptotic security.

Corollary 3.2.10. *There exists a universal constant $\eta > 0$, such that for all $0 \leq \varepsilon < \eta$ the resource $\mathsf{RSP}_{\mathsf{V}}$ is not ε-classically-realizable.*

Proof. The proof is quite similar to the proof of impossibility of $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$. The main difference is that we need to address properly the abort case when $c = 1$. The main idea is to define $\mathcal{A}$ a bit differently: $\mathcal{A}$ picks always $W = X$, and outputs as ρ the classical density matrix corresponding to s when $s \neq \text{ERR}$, and when $s = \text{ERR}$, $\mathcal{A}$ outputs the null matrix $\rho = 0$ (Ω is still the trivial converter). It is easy to see again that this resource is a 0-remote state preparation resource, and it is also impossible to describe it with arbitrary small probability: indeed, when $c = 1$, $\rho = 0$, so the trace $\text{Tr}(\rho\rho')$ (that appears in (3.3)) is equal to 0. Therefore, from a converter $\mathcal{P}$ that (sometimes) inputs $c = 1$, we can always increase the value of $\text{Tr}(\rho\rho')$ by creating a new converter $\mathcal{P}'$ turning c into 0. And we are basically back to the same picture as $\mathcal{S}_{\mathsf{Z}_{\frac{\pi}{2}}}$, where we have a set of states that is impossible to clone with arbitrary small probability, which finishes the impossibility proof. □

Remark 3.2.11. *Note that our impossibility of classically-realizing $\mathsf{RSP}_{\mathsf{V}}$ does not contradict the result of [GV19]. Specifically, in their work they make use of an additional assumption (the so called "Measurement Buffer" resource), which "externalizes" the measurement done by the distinguisher onto the simulator. In practice, this allows the simulator to change the state on the distinguisher side without letting him know. However, what our result shows is that it is impossible to realize this Measurement Buffer resource with a protocol interacting purely classically. Intuitively, the Measurement Buffer re-creates a quantum channel between the simulator and the server: when the simulator is not testing that the server is honest, the simulator replaces the state of the server with the quantum state sent by the ideal resource. This method has however a second drawback: it is possible for the server to put a known state as the input of the Measurement Buffer, and if he is not tested on that run (occurring with probability $\frac{1}{n}$), then he can check that the state has not been changed, leading to polynomial security (a polynomially bounded distinguisher can distinguish between the ideal and the real world). And because in AC, the security of the whole protocol is the sum of the security of the inner protocols, any protocol using this inner protocol will not be secure against polynomial distinguishers.*

3.2.4 Accepting the Limitations: Fully Leaky RSP resources

As explained in the previous section, Theorem 3.2.6 rules out all resources that are impossible to be *describable* with unbounded power, and that the only type of classically-realizable RSP resources would be the one leaking the full classical description of the output quantum state to an unbounded adversary, which we will refer to as being *fully-leaky* RSP. Fully-leaky RSP resources can be separated into two categories:

1. If the RSP is describable in quantum polynomial time, then the adversary can get the secret in polynomial time. This is obviously not an interesting case as the useful properties that we know from quantum computations (such as UBQC) cannot be preserved if such a resource is employed to prepare the quantum states.

2. If the RSP are only describable using unbounded power, then these *fully-leaky* RSP resources are not trivially insecure, but their universally composable security remains unclear. Indeed, it defeats the purpose of aiming at a nice ideal resource where the provided security should be clear "by definition" and it becomes hard to quantify how the additional leakage could be used when composed with other protocols. A possible remedy would be to show restricted composition following [JM17] which we discuss at the end of this paragraph.

For completeness, we present an example of a resource that stands in this second category when assuming that post-quantum encryption schemes exist (e.g. based on the hardness of the LWE problem). As explained before, this resource needs to completely leak the description of the classical state, which in our case, is done by leaking an encryption of the description of the output state. The security guarantees therefore rely on the properties of the encryption scheme, and not on an ideal privacy guarantee as one would wish for, which is an obvious limitation.

A concrete example. In this section we focus on the second category of fully-leaky RSP and we show an example of resource that belongs to this class and a protocol realizing this resource. The fully-leaky RSP resource that we will implement, produces a BB84 state (corresponding to the set of states produced by the simpler QFactory protocol) and is described below:

Definition 3.2.12 (Ideal Resource $\mathsf{RSP}_{\mathsf{CC}}^{4-\mathsf{states},\mathcal{F}}$)**.** *Let* $\mathcal{F} = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *be a family of public-key encryption functions. Then, we define* $\mathsf{RSP}_{\mathsf{CC}}^{4-\mathsf{states},\mathcal{F}}$ *as pictured in Figure 3.6.*

B_1 represents the basis of the output state, and is guaranteed to be random even if the right interface is malicious. B_2 represents the value bit of the output state when encoded in the basis B_1, and in the worst case it can be chosen by the right interface in a malicious scenario[13]*. Note however that in a malicious run, the adversary does not have access (at least not directly from the ideal resource) to the quantum state whose classical description is known by the classical client.*

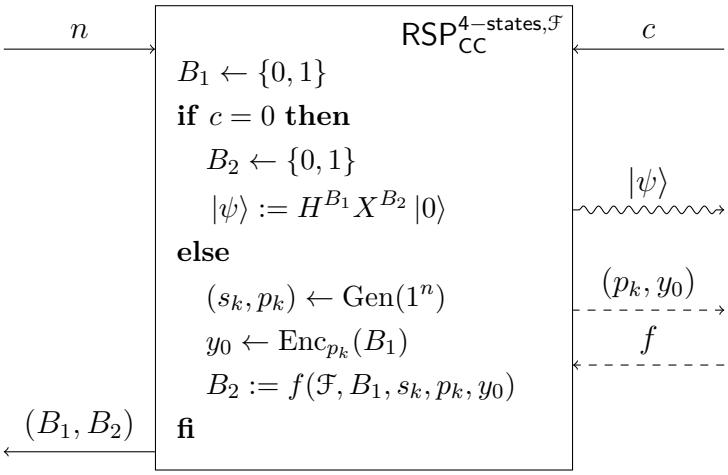


Figure 3.6: Ideal resource RSP$^{4-\text{states},\mathcal{F}}_{\text{CC}}$, which prepares one of the four BB84 states. The "snake" arrow is sent only in the honest case ($c = 0$), and the dashed arrows are send/received only in the malicious case ($c = 1$).

Lemma 3.2.13. *The 4-states QFactory protocol* [Coj+19] *(Protocol 2) securely constructs* RSP$^{4-\text{states},\mathcal{F}}_{\text{CC}}$ *from a classical channel, where $\mathcal{F}$ is defined as follows:*

1. $(t_K, K) \leftarrow \text{Gen}(1^n)$ *outputs two matrices: public K (used to describe the function) and secret t_K (a trapdoor used to invert the function) as defined in* [Coj+19; Coj+21] *(which is itself based on the learning with errors problem and the construction presented in* [MP12]*);*

2. $y_0 \leftarrow \text{Enc}_K(B_1)$ *, where* $y_0 = Ks_0 + e_0 + B_1 \begin{pmatrix} q/2 & 0 & \dots & 0 \end{pmatrix}^T$*, s_0 and e_0 being sampled accordingly to some distribution presented in* [Coj+19; Coj+21]

3. $B_1 \leftarrow \text{Dec}_{t_K}(y)$ *- using t_K we can efficiently obtain B_1 from y_0.*

[13] Note that here the right interface can have (in a malicious scenario) full control over B_2, but in the QFactory Protocol 2 it is not clear what an adversary can do concerning B_2.

Proof. We already know that the protocol of QFactory (π_A, π_B) is correct with super-polynomial probability if the parameters are chosen accordingly (Theorem 3.4.1), therefore

$$\pi_A \mathcal{C} \pi_B \approx_\varepsilon \mathsf{RSP}_{\mathsf{CC}}^{4-\text{states},\mathcal{F}} \vdash \tag{3.23}$$

for some negligible ε. We now need to find a simulator σ such that

$$\pi_A \mathcal{C} \approx_{\varepsilon'} \mathsf{RSP}_{\mathsf{CC}}^{4\text{-states},\mathcal{F}} \sigma \tag{3.24}$$

The simulator is trivial here: it sends $c = 1$ to ideal resource then, it just forwards the (K, y_0) given by the resource to its outer interface, and when it receives the (y, b) corresponding to the measurements performed by the server, it just sets the deviation f to be the same function as the one computed by π_A. Therefore, $\pi_A \mathcal{C} \approx_0 \mathsf{RSP}_{\mathsf{CC}}^{4\text{-states},\mathcal{F}} \sigma$, which ends the proof. □

Concluding remarks. We see that using this kind of leaky resource is not desirable: the resources are non-standard and it seems hard to write a modular protocol with this resource as an assumed resource. The resource is very specific and mimics its implementation. As such, we cannot really judge its security.

On the other hand however, if a higher-level protocol did guarantee that the value B_2 always remains hidden, i.e., a higher level protocol's output does not depend on on B_2 (e.g., by blinding it all the time), it is easy to see that we could simulate y_0 without knowledge about B_1 thanks to the semantic security of the encryption scheme. If we fix this restricted context, the ideal resource in Figure 3.6 could be re-designed to not produce the output (p_k, y_0) at all and therefore, by definition, leak nothing extra about the quantum state (note that in such a restricted context, the simulator can simply come up with a fake encryption that is indistinguishable). This can be made formal following [JM17]. We note in passing that this particular example quite severely restricts applicability unfortunately. Indeed, it is interesting future research whether it is possible to come up with restricted yet useful contexts that admit nice ideal resources for RSP following the framework in [JM17].

3.3 Impossibility of Composable Classical-Client UBQC

In the previous section, we showed that it was impossible to get a (useful) composable $\mathsf{RSP}_{\mathsf{CC}}$ protocol. A (weaker) RSP protocol, however, could still be used internally in other protocols, hoping for the overall protocol to be composably secure. To this end, we analyze the composable security of a well-known delegated quantum computing protocol, universal blind quantum computation (UBQC), proposed in [BFK09]. The UBQC protocol allows a semi-quantum client, Alice, to delegate an arbitrary quantum computation to a (universal) quantum server Bob, in such a way that her input, the quantum computation and the output of the computation are information-theoretically hidden from Bob. The protocol requires Alice to be able to prepare single qubits of the form $|+_\theta\rangle$, where $\theta \in \mathsf{Z}_{\frac{\pi}{4}}$ and send these states to Bob at the beginning of the protocol, the rest of the communication between the two parties being classical. We define the family of protocols $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$ as the RSP protocols that classically delegate the preparation of an output state $|+_\theta\rangle$, where $\theta \in \mathsf{Z}_{\frac{\pi}{4}}$. That is, without loss of generality, we assume a pair of converters P_A, P_B such that the resource $R := P_A \mathcal{C} P_B$ has the behavior of the prototype RSP resource except with negligible probability. Put differently, we assume we have an (except with negligible error) *correct* RSP protocol, but we make *no assumption about the security* of this protocol. Therefore, one can directly instantiate the quantum interaction with the $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$ at the first step as shown in Protocol 1. While UBQC allows for both quantum and classical outputs and inputs, given that we want to remove the quantum interaction in favor of a completely classical interaction, we only focus on the classical input and classical output functionality of UBQC in the remaining of the chapter.

Protocol 1 UBQC with $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$ (See [BFK09])

- **Client's classical input:** An n-qubit unitary U that is represented as set of angles $\{\phi\}_{i,j}$ of a one-way quantum computation over a brickwork state/cluster state [MDF17], of the size $n \times m$, along with the dependencies X and Z obtained via flow construction [DK06].
- **Client's classical output:** The measurement outcome $\bar{s}$ corresponding to the n-qubit quantum state, where $\bar{s} = \langle 0| U |0\rangle$.

1. Client and Server runs $n \times m$ different instances of $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$ (in parallel) to obtain $\theta_{i,j}$ on client's side and $\left|+_{\theta_{i,j}}\right\rangle$ on server's side, where $\theta_{i,j} \leftarrow \mathsf{Z}_{\frac{\pi}{4}}$, $i \in \{1, \cdots, n\}$, $j \in \{1, \cdots, m\}$

2. Server entangles all the qubits, $n \times (m-1)$ received from $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$, by applying controlled-Z gates between them in order to create a graph state $\mathcal{G}_{n\times m}$

3. For $j \in [1,m]$ and $i \in [1,n]$

   (a) Client computes $\delta_{i,j} = \phi'_{i,j} + \theta_{i,j} + r_{i,j}\pi$, $r_{i,j} \leftarrow \{0,1\}$, where $\phi'_{i,j} = (-1)^{s^X_{i,j}}\phi_{i,j} + s^Z_{i,j}\pi$ and $s^X_{i,j}$ and $s^Z_{i,j}$ are computed using the previous measurement outcomes and the X and Z dependency sets. Client then sends the measurement angle $\delta_{i,j}$ to the Server.

   (b) Server measures the qubit $\left|+_{\theta_{i,j}}\right\rangle$ in the basis $\{\left|+_{\delta_{i,j}}\right\rangle, \left|-_{\delta_{i,j}}\right\rangle\}$ and obtains a measurement outcome $s_{i,j} \in \{0,1\}$. Server sends the measurement result to the client.

   (c) Client computes $\bar{s}_{i,j} = s_{i,j} \oplus r_{i,j}$.

4. The measurement outcome corresponding to the last layer of the graph state ($j = m$) is the outcome of the computation.

Note that Protocol 1 is based on measurement-based model of quantum computing (MBQC). This model is known to be equivalent to the quantum circuit (up to polynomial overhead in resources) and does not require one to perform quantum gates on their side to realize arbitrary quantum computation. Instead, the computation is performed by an (adaptive) sequence of single-qubit projective measurements that steer the information flow across a highly entangled resource state. Intuitively, UBQC can be seen as a distributed MBQC where the measurements are performed by the server whereas the classical update of measurement bases is perfomed by the client. Since the projective measurements in quantum physics, in general, are probabilistic in nature and therefore, the client needs to update the measurement bases (and classically inform the server about the update) based on the outcomes of the earlier measurements to ensure the correctness of the computation. Roughly speaking, this information flow is captured by the X and Z dependencies. For more details, we refer the reader to [RB01; Nie06].

Next, we show that the Universal Blind Quantum Computing protocol [BFK09], which is proven to be secure in the Abstract Cryptography framework [Dun+14], cannot be proven composably secure (for the same ideal resource) when the quantum interaction is replaced with $\mathsf{RSP}_{\mathsf{CC}}$ (this class of protocol is denoted as $\mathsf{UBQC}_{\mathsf{CC}}$). We also give an outlook that the impossibility proof also rules out weaker ideal resources.

3.3.1 Impossibility of Composable $\mathsf{UBQC}_{\mathsf{CC}}$ on 1 Qubit

In order to prove that there exists no $\mathsf{UBQC}_{\mathsf{CC}}$ protocol, we will first focus on the simpler case when the computation is described by a single measurement angle. The resource that performs a blind quantum computation on one qubit ($\mathcal{S}_{UBQC1}$) is defined as below:

Definition 3.3.1 (Ideal resource of single-qubit UBQC (See [Dun+14]))**.** *The definition of the ideal resource $\mathcal{S}_{UBQC1}$, depicted in Figure 3.7, achieves blind quantum computation specified by a single angle ϕ. The input (ξ, ρ) is filtered when $c = 0$. The ξ can be any deviation (specified for example using the classical description of a CPTP map) that outputs a classical bit, and which can depend on the computation angle ϕ and on some arbitrary quantum state ρ.*

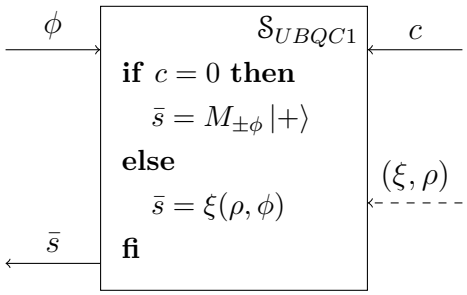


Figure 3.7: Ideal resource $\mathcal{S}_{UBQC1}$ for UBQC with one angle, with a filtered (dashed) input. In the case of honest server the output $\bar{s} \in \{0, 1\}$ is computed by measuring the qubits $|+\rangle$ in the $\{|+_\phi\rangle, |-_\phi\rangle\}$ basis. On the other hand if $c = 1$ any malicious behaviour of server can be captured by (ξ, ρ), i.e. the output $\bar{s}$ is computed by applying the CPTP map ξ on the input ϕ and on another auxiliary state ρ chosen by the server.

Theorem 3.3.2 (No-go composable classical-client single-qubit UBQC)**.** *Let (P_A, P_B) be a protocol interacting only through a classical channel $\mathcal{C}$, such that $(\theta, \rho_B) \leftarrow (P_A \mathcal{C} P_B)$ with $\theta \in \mathsf{Z}\frac{\pi}{4}$, and such that (by correctness) the trace distance between ρ_B and $|+_\theta\rangle\langle+_\theta|$ is negligible with overwhelming probability[14] with overwhelming probability[15]. Then, if we define π_A and π_B as the UBQC protocol on one qubit that makes use of (P_A, P_B) as a sub-protocol to replace the quantum channel (as pictured in Figure 3.8), (π_A, π_B) is*

[14] In the following, the parties P_A and P_B (and therefore π_A and π_B) and the simulator σ depend on some security parameter n, but, in order to simplify the notations and the proof, this dependence will be implicit. We are as usual interested only in the asymptotic security, when $n \to \infty$.

[15] Note that here ρ_B is different at every run: it corresponds to the density matrix of the state obtained after running P_B, when tracing out the environment and the internal registers of P_B and P_A.

not composable, i.e. there exists no simulator σ such that:

$$\pi_A \mathcal{C} \pi_B \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^{c=0} \tag{3.25}$$

$$\pi_A \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma \tag{3.26}$$

for some negligible $\varepsilon = \mathsf{negl}(n)$.

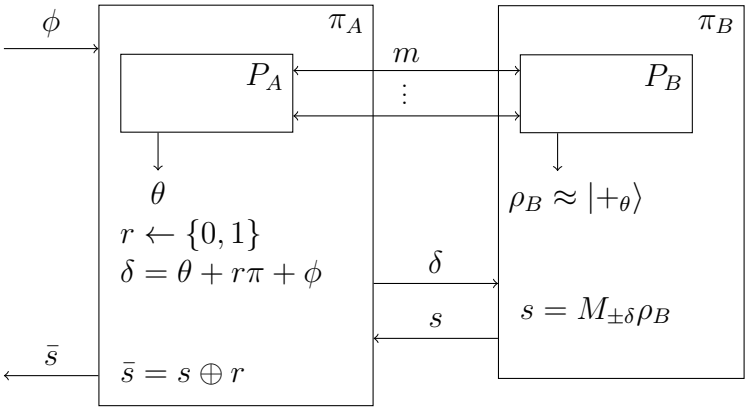


Figure 3.8: UBQC with one qubit when both Alice and Bob follows the protocol honestly (see Protocol 1)

Proof. In order to prove this theorem, we will proceed by contradiction. Let us assume that there exists (P_A, P_B), and a simulator σ having the above properties.
Then, for the same resource $\mathcal{S}_{UBQC1}$ we consider a different protocol $\boldsymbol{\pi}' = (\pi'_A, \pi'_B)$ that realizes it, but using a different filter[16] $\vdash^\sigma$ and a different simulator σ':

$$\pi'_A \mathcal{C} \pi'_B \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^\sigma \tag{3.27}$$

$$\pi'_A \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma' \tag{3.28}$$

More specifically, the new filter $\vdash^\sigma_{UBQC1}$ will depend on σ defined in (3.26). Then our main proof can be described in the following steps:

1. We first show in Lemma 3.3.4 that $\mathcal{S}_{UBQC1}$ is also ε-classically-realizable by (π'_A, π'_B) with the filter $\vdash^\sigma$.

2. We then prove in Lemma 3.3.5 that the resource $\mathcal{S}_{UBQC1}$ is an RSP within $\mathsf{negl}(n)$,

[16] Note that we could include this new filter inside $\mathcal{S}_{UBQC1}$ and use a more traditional filter $\vdash^{c=0}$ but for simplicity we will just use a different filter.

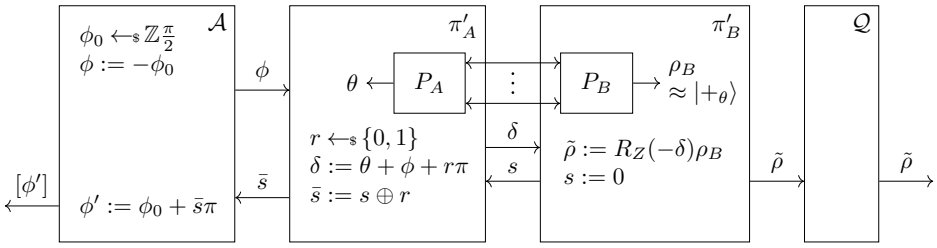


Figure 3.9: Definition of $\mathcal{A}$, π'_A, π'_B and $\mathcal{Q}$.

with respect to some well chosen converters $\mathcal{A}$ and $\mathcal{Q}$ (see Figure 3.9) and this new filter $\vdash^\sigma$.

3. Then, we use the main result about RSP (Theorem 3.2.6) to show that $\mathcal{S}_{UBQC1}$ is describable within $\mathsf{negl}(n)$ with respect to $\mathcal{A}$ (Corollary 3.3.6).

4. Finally, in Lemma 3.3.8 we prove that if $\mathcal{S}_{UBQC1}$ is describable then we could achieve *superluminal signaling*, which concludes the contradiction proof. □

Definition 3.3.3. *Let $\pi' = (\pi'_A, \pi'_B)$ the protocol realizing $\mathcal{S}_{UBQC1}$ described in the following way (as pictured Figure 3.9):*

- *$\pi'_A = \pi_A$ (Figure 3.8)*
- *π'_B: runs P_B, obtains a state ρ_B, then uses the angle δ received from its inner interface to compute $\tilde{\rho} := R_Z(-\delta)\rho_B$, and finally outputs $\tilde{\rho}$ on its outer interface and $s := 0$ on its inner interface.*

Then we define $\vdash^\sigma = \sigma\pi'_B$ depicted in Figure 3.10 (with σ the simulator defined in (3.26) as explained before).
We define the converters $\mathcal{A}$ and $\mathcal{Q}$ as seen in:

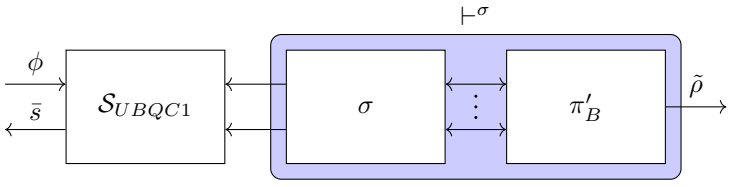


Figure 3.10: Description of $\vdash^\sigma$.

Lemma 3.3.4. *If $\mathcal{S}_{UBQC1}$ is ε-classically-realizable by (π_A, π_B) with the filter $\vdash^{c=0}$ then $\mathcal{S}_{UBQC1}$ is also ε-classically-realizable by (π'_A, π'_B) with the filter $\vdash^\sigma$.*

Proof. If S_{UBQC1} is ε-classically-realizable with $\vdash^{c=0}$, then as seen in Theorem 3.3.2, we have:

$$\pi_A \mathcal{C} \pi_B \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^{c=0} \tag{3.29}$$

$$\pi_A \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma \tag{3.30}$$

Now we can show that $\mathcal{S}_{UBQC1}$ is ε-classically-realizable by (π'_A, π'_B) with $\vdash^\sigma$, i.e. that there exists a simulator σ' such that:

$$\pi'_A \mathcal{C} \pi'_B \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^\sigma \tag{3.31}$$

$$\pi'_A \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma' \tag{3.32}$$

For the correctness condition, we have:

$$\pi'_A \mathcal{C} \pi'_B = (\pi_A \mathcal{C}) \pi'_B \tag{3.33}$$

$$\approx_\varepsilon (\mathcal{S}_{UBQC1} \sigma) \pi'_B \tag{3.34}$$

$$= \mathcal{S}_{UBQC1} \vdash^\sigma \tag{3.35}$$

For the security condition, we define $\sigma' = \sigma$. Then, we have:

$$\pi'_A \mathcal{C} = \pi_A \mathcal{C} \tag{3.36}$$

$$\approx_\varepsilon \mathcal{S}_{UBQC1} \sigma \tag{3.37}$$

Which concludes our proof. □

Lemma 3.3.5. *If $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(n)$-classically-realizable with $\vdash^{c=0}$ then $\mathcal{S}_{UBQC1}$ is an $\mathsf{negl}(n)$-*remote state preparation resource *with respect the converters $\mathcal{A}$ and Ω and filter $\vdash^\sigma$ defined in Figure 3.9.*

Proof. We need to prove that:

$$\mathop{\mathbb{E}}_{([\rho], \rho_B) \leftarrow \mathcal{A} \mathcal{S}_{UBQC1} \vdash^\sigma \Omega} [\, \mathrm{Tr}(\rho \rho_B) \,] \geq 1 - \varepsilon \tag{3.38}$$

First, we remark that due to Lemma 3.3.4:

$$\mathcal{A} \mathcal{S}_{UBQC1} \vdash^\sigma \Omega \approx_\varepsilon \mathcal{A} \pi'_A \mathcal{C} \pi'_B \Omega \tag{3.39}$$

However, from the protocol description it is easy to check that in the real world $\bar{s} = 0 \oplus r = r$, and therefore $\phi' := \phi_0 + \bar{s}\pi = \phi_0 + r\pi$ and $\rho = |+_{\phi'}\rangle\langle+_{\phi'}|$. And because the trace distance between ρ_B and $|+_\theta\rangle\langle+_\theta|$ is negligible with overwhelming probability (by the correctness of (P_A, P_B)), then we also have that $\tilde{\rho} = R_Z(-\delta)\rho_B R(-\delta)^\dagger$ is negligibly close in trace distance to $|+_{\theta-\delta}\rangle\langle+_{\theta-\delta}| = |+_{-\phi_0+r\pi}\rangle\langle+_{-\phi_0+r\pi}| = |+_{\phi'}\rangle\langle+_{\phi'}|$. Therefore, we have:

$$\mathop{\mathbb{E}}_{([\rho],\tilde{\rho})\leftarrow\mathcal{A}\pi'_A\mathcal{C}\pi'_B\mathcal{Q}}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1 - \mathsf{negl}(n) \tag{3.40}$$

Then it also means that:

$$\mathop{\mathbb{E}}_{([\rho],\tilde{\rho})\leftarrow\mathcal{A}\mathcal{S}_{UBQC1}\vdash^\sigma\mathcal{Q}}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1 - \mathsf{negl}(n) \tag{3.41}$$

otherwise we could (using a similar argument to the one given in the proof of Theorem 3.2.6) distinguish between the ideal and the real world, contradicting (3.39), which concludes the proof. □

Now, using our main Theorem 3.2.6 we obtain directly that if $\mathcal{S}_{UBQC1}$ is classically-realizable and RSP with respect to filter $\vdash^\sigma$, then it is also describable:

Corollary 3.3.6. *If $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(n)$-classically-realizable with respect to filter $\vdash^{c=0}$ then $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(n)$-describable with respect to the converter $\mathcal{A}$ described above.*

Lemma 3.3.7. *Let $\Omega = \{[\rho_i]\}$ be a set of (classical descriptions of) density matrices, such that $\forall i \neq j$, $\mathrm{Tr}(\rho_i\rho_j) \leq 1-\eta$. Then let $([\rho],[\tilde{\rho}])$ be two random variables (representing classical description of density matrices), such that $[\rho] \in \Omega$ and $\mathop{\mathbb{E}}_{([\rho],[\tilde{\rho}])}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1-\varepsilon$, with $\eta > 6\sqrt{\varepsilon}$. Then, if we define the following "rounding" operation that rounds $\tilde{\rho}$ to the closest $\tilde{\rho}_r \in \Omega$:*

$$[\tilde{\rho}_r] := \mathrm{Round}_\Omega([\tilde{\rho}]) := \mathop{\arg\max}_{[\tilde{\rho}_r]\in\Omega} \mathrm{Tr}(\tilde{\rho}_r\tilde{\rho}) \tag{3.42}$$

Then we have:

$$\Pr_{([\rho],[\tilde{\rho}])}[\,\mathrm{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1 - \sqrt{\varepsilon} \tag{3.43}$$

In particular, if $\varepsilon = \mathsf{negl}(n)$, and $\eta \neq 0$ is a constant, $\Pr[\,\mathrm{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1 - \mathsf{negl}(n)$.

Proof. We know that $\mathop{\mathbb{E}}_{([\rho],[\tilde{\rho}])}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1-\varepsilon$. Therefore, using Markov inequality we get

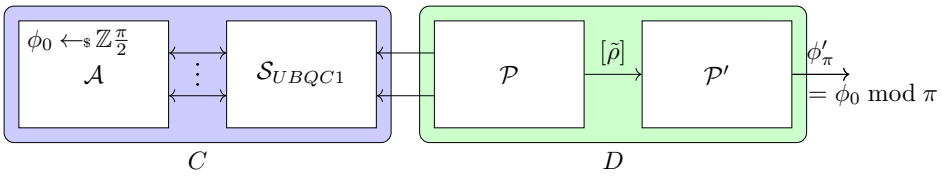


Figure 3.11: Illustration of the no-signaling argument

that:

$$\Pr_{([\rho],[\tilde{\rho}])}[\,1 - \mathrm{Tr}(\rho\tilde{\rho}) \geq \sqrt{\varepsilon}\,] \leq \frac{\mathbb{E}[\,1 - \mathrm{Tr}(\rho\tilde{\rho})\,]}{\varepsilon} \tag{3.44}$$

$$\Pr_{([\rho],[\tilde{\rho}])}[\,\mathrm{Tr}(\rho\tilde{\rho}) \leq 1 - \sqrt{\varepsilon}\,] \leq \frac{\varepsilon}{\sqrt{\varepsilon}} \tag{3.45}$$

$$\Pr_{([\rho],[\tilde{\rho}])}[\,\mathrm{Tr}(\rho\tilde{\rho}) \geq 1 - \sqrt{\varepsilon}\,] \geq 1 - \sqrt{\varepsilon} \tag{3.46}$$

But when $\mathrm{Tr}(\rho\tilde{\rho}) \geq 1 - \sqrt{\varepsilon}$, we have $\mathrm{Round}_\Omega([\tilde{\rho}]) = \rho$.

We will indeed show that $\forall \rho_i \in \Omega$, $\mathrm{Tr}(\rho_i\tilde{\rho}) \leq \mathrm{Tr}(\rho\tilde{\rho})$. By contradiction, we assume there exists $\rho_i \in \Omega$ such that $\rho_i \neq \rho$ and $\mathrm{Tr}(\rho_i\tilde{\rho}) > \mathrm{Tr}(\rho\tilde{\rho}) \geq 1 - \sqrt{\varepsilon}$. But due to Lemma 3.5.4 we have:

$$\mathrm{Tr}(\rho_i\rho) \geq 1 - 3(\sqrt{\varepsilon} + \sqrt{\varepsilon}) = 1 - 6\sqrt{\varepsilon} \tag{3.47}$$

However, because both ρ_i and ρ belong to Ω, we also have $\mathrm{Tr}(\rho_i\rho) \leq 1 - \eta < 1 - 6\sqrt{\varepsilon}$, which is absurd.

Therefore, using (3.46) we have

$$\Pr_{([\rho],[\tilde{\rho}])}[\,\mathrm{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1 - \sqrt{\varepsilon} \tag{3.48}$$

which concludes the proof. □

Lemma 3.3.8. *$\mathcal{S}_{UBQC1}$ cannot be $\mathsf{negl}(n)$-describable with respect to converter $\mathcal{A}$.*

Proof. If we assume that $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(n)$-describable, then there exists a converter $\mathcal{P}$ (outputting $[\tilde{\rho}]$) such that:

$$\mathop{\mathbb{E}}_{([\rho],[\tilde{\rho}]) \leftarrow \mathcal{A}\mathcal{S}_{UBQC1}\mathcal{P}}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1 - \mathsf{negl}(n) \tag{3.49}$$

We define the set $\Omega := \{[|+_{\theta'}\rangle\langle+_{\theta'}|] \mid \theta' \in \{0, \pi/4, ..., 7\pi/4\}\}$. For simplicity, we will denote in the following $[\theta] = [|+_\theta\rangle\langle+_\theta|]$.

In the remaining of the proof, we are going to use the converters $\mathcal{A}$ and $\mathcal{P}$ together with the ideal resource $\mathcal{S}_{UBQC1}$, to construct a 2-party setting that would achieve signaling, which would end our contradiction proof. More specifically, we will define a converter D running on the right interface of $\mathcal{S}_{UBQC1}$ which will manage to recover the ϕ_0 chosen randomly by $\mathcal{A}$.
As shown in Figure 3.11, if we define C as $C := \mathcal{A}\mathcal{S}_{UBQC1}$ and D the converter described above, then the setting can be seen equivalently as: C chooses as random ϕ_0 and D needs to output $\phi_0 \bmod \pi$. This is however impossible, as no message is sent from $\mathcal{S}_{UBQC1}$ to its right interface (as seen in Figure 3.11) (and thus no message from C to D), and therefore guessing ϕ_0 is forbidden by the no-signaling principle [GRW80].

We define $\mathcal{P}'$ as the converter that, given $[\tilde{\rho}]$ from the outer interface of $\mathcal{P}$ computes $[\tilde{\phi}] = \text{Round}_\Omega([\tilde{\rho}])$ and outputs $\tilde{\phi}_\pi = \tilde{\phi} \bmod \pi$ (as depicted in Figure 3.11). We will now prove that $\tilde{\phi}_\pi = \phi_0 \bmod \pi$ with overwhelming probability.

All elements in Ω are different pure states, and in finite number, so there exist a constant $\eta > 0$ respecting the first condition of Lemma 3.3.7. Moreover from (3.49) we have that $\mathcal{S}_{UBQC1}$ is ε-describable with $\varepsilon = \mathsf{negl}(n)$, so we also have (for large enough n), $\eta > 6\sqrt{\varepsilon}$. Therefore, from Lemma 3.3.7, we have that:

$$\Pr_{([\rho],[\tilde{\rho}])\leftarrow\mathcal{A}\mathcal{S}_{UBQC1}\mathcal{P}}[\,\text{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1 - \mathsf{negl}(n) \tag{3.50}$$

But using the definition of converter $\mathcal{A}$, we have: $[\rho] = [\phi']$, where $\phi' = \phi_0 + \bar{s}\pi$, and hence $\phi' \bmod \pi = \phi_0 \bmod \pi$. Then, using the definition of $\mathcal{P}'$, the (3.50) is equivalent to:

$$\Pr_{([\phi'],\tilde{\phi}_\pi)\leftarrow\mathcal{A}\mathcal{S}_{UBQC1}\mathcal{P}\mathcal{P}'}[\,\tilde{\phi}_\pi = \phi_0 \bmod \pi\,] \geq 1 - \mathsf{negl}(n) \tag{3.51}$$

However, as pictured in Figure 3.11, this can be seen as a game between $C = \mathcal{A}\mathcal{S}_{UBQC1}$ and $D = \mathcal{P}\mathcal{P}'$, where, as explained before, C picks a $\phi_0 \in \mathsf{Z}_{\frac{\pi}{2}}$ randomly, and D needs to output $\phi_0 \bmod \pi$. From (3.51) D wins with overwhelming probability, however, we know that since there is no information transfer from C to D, the probability of winning this game better than 1/4 (guessing both the bits at random) would imply signalling.

□

Remark 3.3.9. *The guessing game described at the end of the preceding proof can be generalized to the case when some (partial) information transfer from C to D takes place.*

More precisely, whenever we consider a new resource together with some converters $\mathcal{A}$ and $\mathcal{Q}$, it is enough to show that this resource is not describable to prove that it is impossible to classically realize. To that purpose, it may as above be practical to define a guessing game similar to the above one, but without the nice property that no information flows from C to D. Here, the connections with the non-local games [Bru+14] and information causality [Paw+09] could provide an upper bound on the winning probability (e.g., as a function of the conditional mutual information conditioned on the information exchanged). We leave the quantitative analysis for future work.

3.3.2 Impossibility of Composable $\mathsf{UBQC}_{\mathsf{CC}}$ on Any Number of Qubits

We saw in Theorem 3.3.2 that it is not possible to implement a composable classical-client UBQC protocol performing a computation on a single qubit. In this section, we prove that this result generalizes to the impossibility of $\mathsf{UBQC}_{\mathsf{CC}}$ on computations using an arbitrary number of qubits. The proof works by reducing the general case to the single-qubit case from the previous section.

Theorem 3.3.10 (No-go Composable Classical-Client UBQC)**.** *Let (P_A, P_B) be a protocol interacting only through a classical channel $\mathcal{C}$, such that $(\theta, \rho_B) \leftarrow (P_A \mathcal{C} P_B)$ with $\theta \in \mathsf{Z}_{\frac{\pi}{4}}$, and such that the trace distance between ρ_B and $|+_\theta\rangle\langle+_\theta|$ is negligible with overwhelming probability. Then, if we define (π_A^G, π_B^G) as the UBQC protocol on any fixed graph G (with at least one output qubit[17]), that uses (P_A, P_B) as a sub-protocol to replace the quantum channel, (π_A^G, π_B^G) is not composable, i.e. there exists no simulator σ such that:*

$$\pi_A^G \mathcal{C} \pi_B^G \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^{c=0} \tag{3.52}$$

$$\pi_A^G \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma \tag{3.53}$$

for some negligible $\varepsilon = \mathsf{negl}(n)$.

Proof. To prove this statement, we just need to prove that we can come back to the setting with a single qubit, where we want to perform a computation with angle ϕ, and output one angle close to ϕ as in the proof of Theorem 3.3.2. Because the graph has at

[17] Note, that in $\mathsf{UBQC}_{\mathsf{CC}}$ with zero output qubits the client does not receive any results. Hence, the protocol is trivially implementable for this degenerated case.

least one output qubit, we will denote by ω the index of the last output qubit. So the idea is to let the distinguisher choose the client input such that for any node $i \neq \omega$ in the graph, $\phi_i = 0$, and for the output qubit, $\phi_\omega = \phi$. Moreover, on the server side, the distinguisher will behave like the honest protocol π_B^G, except that it will not entangle the qubits provided by P_A, and it will deviate on the output qubit ω by not measuring it and sending $s := 0$, the qubit being rotated again with angle $-\delta_\omega$, and outputed on the outer interface, like in the one-qubit case. It is now easy to see by induction (over the index of the qubit, following the order chosen on G) that, in the real world, for all $i \neq \omega$, we always have $s_i = r_i$, therefore $\bar{s}_i = 0$. So for all nodes i, (including ω), $s_i^X = \oplus_{i \in D_i} \bar{s}_i = 0$ and $s_i^Z = \oplus_{i \in D'_i} \bar{s}_i = 0$. Thus we have on the last node:

$$\begin{aligned} \delta_\omega &= \theta_\omega + (-1)^{s_\omega^X} \phi_\omega + s_\omega^Z \pi + r_\omega \pi \\ &= \theta_\omega + \phi + r_\omega \pi \end{aligned}$$

which corresponds exactly to the single-qubit setting, shown to be impossible. □

3.4 Game-Based Security of QF-UBQC

While we know from Theorem 3.3.10 that classical-client UBQC (henceforth simply $\mathsf{UBQC}_{\mathsf{CC}}$) cannot be proven secure in a fully composable setting, there is hope that it remains possible with a weaker definition of security. And indeed, in this section we show that $\mathsf{UBQC}_{\mathsf{CC}}$ is possible in the *game-based setting* by implementing it using a combination of the known quantum-client UBQC Protocol 1 [BFK09] and 8-states QFactory Protocol 3 [Coj+19].

3.4.1 QFactory: Remote State Preparation, Revisited

The construction of the QFactory protocol relies on a family of functions with certain cryptographic properties, specifically, a 2-regular homomorphic-hardcore family of functions. For the formal definition of these properties, see [Coj+19].

We first begin by recalling the formal description of the protocol in Section 3.4.1.1 and then in Section 3.4.1.2 and Section 3.4.1.3 we present the results concerning the correctness and security of QFactory.

3.4.1.1 4-states and 8-states QFactory protocol

Protocol 2 4-states QFactory: classical delegation of the BB84 states ([Coj+19])

Requirements: Public: A 2-regular homomorphic-hardcore family $\mathcal{F}$ with respect to $\{h_k\}$ and d_0. For simplicity, we will represent the sets $\mathcal{D}'$ (respectively $\mathcal{R}$) using n (respectively m) bits strings: $\mathcal{D}' = \{0,1\}^n$, $\mathcal{R} = \{0,1\}^m$.

Stage 1: Preimages superposition

1. Client runs the algorithm $(k, t_k) \leftarrow \mathrm{Gen}_{\mathcal{F}}(1^n)$.
2. Client instructs Server to prepare one register at $\otimes^n H |0\rangle$ and second register initiated at $|0\rangle^m$.
3. Server receives k from the client and applies U_{f_k} using the first register as control and the second as target.
4. Server measures the second register in the computational basis, obtains the outcome y. The combined state is given by $(|x\rangle + |x'\rangle) \otimes |y\rangle$ with $f_k(x) = f_k(x') = y$ and $y \in \mathrm{Im}\, f_k$.

Stage 2: Output preparation

1. Server applies U_{h_k} on the preimage register $|x\rangle + |x'\rangle$ as control and another qubit initiated at $|0\rangle$ as target. Then, measures all the qubits, but the target in the $\{\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)\}$ basis, obtaining the outcome $b = (b_1, ..., b_n)$. Now, the Server returns both y and b to the Client.
2. Client using the trapdoor t_k computes the preimages of y:
   - if y does not have exactly two preimages x, x' (the server is cheating with overwhelming probability), defines $B_1 = d_0(t_k)$, and chooses $B_2 \in \{0,1\}$ uniformly at random
   - if y has exactly two preimages x, x', defines $B_1 = h_k(x) \oplus h_k(x') = d_0(t_k)$, and B_2.

Output: The quantum state that the Server has generated is (with overwhelming probability [18]) the BB84 state $|\mathtt{out}\rangle = H^{B_1} X^{B_2} |0\rangle$ (see (3.55) and (3.56) for the exact value of B_1 and B_2). The output of the Server is a quantum state $|\mathtt{out}\rangle$ and the output of the Client is given by (B_1, B_2) (2 bits).

[18] As for the previous protocol, the probability comes from the probability of $\mathcal{F}$ being a 2-regular homomorphic-hardcore family of functions

Protocol 3 8-states QFactory: classical delegation of the $|+_\theta\rangle$ states ([Coj+19])

Requirements: Same as in Protocol 2

Input: Client runs twice the algorithm $Gen_{\mathcal{F}}(1^n)$, obtaining $(k^1, t_k^1), (k^2, t_k^2)$. Client keeps t_k^1, t_k^2 private.

Protocol Steps:

1. Client runs 4-states QFactory Protocol 2 to obtain a state $|\mathtt{in}_1\rangle$ and a "rotated" 4-states QFactory to obtain a state $|\mathtt{in}_2\rangle$ (by rotated 4-states QFactory we mean a 4-states QFactory, but where the last set of measurements in the $|\pm\rangle$ basis is replaced by measurements in the $\left|\pm_{\frac{\pi}{2}}\right\rangle$ basis).

2. Client records measurement outcomes (y^1, b^1), (y^2, b^2) and computes and stores the corresponding indices of the output states of the 2 runs of 4-states QFactory protocol: (B_1, B_2) for $|\mathtt{in}_1\rangle$ and (B_1', B_2') for $|\mathtt{in}_2\rangle$.

3. Client instructs Server to apply the Merge Gadget in Fig. 3.12 ([Coj+19]) on the states $|\mathtt{in}_1\rangle$, $|\mathtt{in}_2\rangle$.

4. Server returns the 2 measurement results s_1, s_2.

5. Client using (B_1, B_2), (B_1', B_2'), s_1, s_2 computes the index $L = L_1L_2L_3 \in \{0,1\}^3$ of the output state (see (3.57), (3.58), and (3.59) for the exact value of L_1, L_2, and L_3, respectively.)

Output: The output of the Server is (with overwhelming probability) a quantum state $|\mathtt{out}\rangle := \left|+_{L\frac{\pi}{4}}\right\rangle$ and the output of the Client is given by L (3 bits).

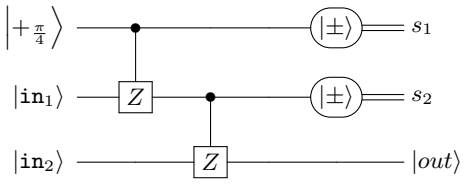


Figure 3.12: Merge Gadget (Taken from [Coj+19])

3.4.1.2 Correctness of QFactory

In an honest run, the description of the output state of the protocol depends on measurement results $y \in \text{Im}\, f_k$ and b, but also on the 2 preimages x and x' of y.

The output state of 4-states QFactory belongs to the set of states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ and its exact description is the following:

Theorem 3.4.1 (4-states QFactory is correct ([Coj+19])). *In an honest run, with overwhelming probability the output state $|\texttt{out}\rangle$ of the 4-states QFactory Protocol 2 is a BB84 state whose basis is $B_1 = h_k(x) \oplus h_k(x') = d_0$, and:*

- *if $d_0 = 0$, then the state is $|h_k(x)\rangle$ (computational basis, also equal to $|h_k(x')\rangle$)*
- *if $d_0 = 1$, then if $\sum_i b_i \cdot (x_i \oplus x'_i) = 0 \bmod 2$, the state is $|+\rangle$, otherwise the state is $|-\rangle$ (Hadamard basis).*

i.e.

$$|\texttt{out}\rangle = H^{B_1} X^{B_2} |0\rangle \tag{3.54}$$

with

$$B_1 = h_k(x) \oplus h_k(x') = d_0 \tag{3.55}$$

$$B_2 = (d_0 \times (b \cdot (x \oplus x'))) \oplus h(x)h(x') \tag{3.56}$$

(the inner product is taken modulo 2, and $x \oplus x'$ is a bitwise xor)

Theorem 3.4.2 (8-states QFactory is correct ([Coj+19])). *In an honest run, the Output state of the 8-states QFactory Protocol is of the form $\left|+_{L\cdot\frac{\pi}{4}}\right\rangle$, where $L = L_1L_2L_3 \in \{0,1\}^3$, defined as:*

$$L_1 = B'_2 \oplus B_2 \oplus [B_1 \cdot (s_1 \oplus s_2)] \tag{3.57}$$

$$L_2 = B'_1 \oplus [(B_2 \oplus s_2) \cdot B_1] \tag{3.58}$$

$$L_3 = B_1 \tag{3.59}$$

3.4.1.3 Security of QFactory

In any run of the protocol, honest or malicious, the state that the client believes that the server has is given by Theorem 3.4.1. Therefore, the task that a malicious server wants

to achieve, is to be able to guess, as good as he can, the description of the output state that the client (based on the public communication) thinks the server has produced. In particular, in our case, the server needs to guess the bit B_1 (corresponding to the basis) of the (honest) output state.

Definition 3.4.3 (4-states basis blindness)**.** *We say that a protocol (π_A, π_B) achieves **basis-blindness** with respect to an ideal list of 4 states $S = \{S_{B_1,B_2}\}_{(B_1,B_2)\in\{0,1\}^2}$ if*

1. *S is the set of states that the protocol outputs, i.e.,*

$$\Pr[\,|\phi\rangle = S_{B_1B_2} \in S \mid ((B_1,B_2),|\phi\rangle) \leftarrow (\pi_A\|\pi_B)\,] \geq 1 - \mathsf{negl}(n), \tag{3.60}$$

2. *and no information is leaked about the index bit B_1 of the output state of the protocol, i.e for all QPT adversary $\mathcal{A}$, it holds that*

$$\Pr[\,B_1 = \tilde{B}_1 \mid ((B_1,B_2),\tilde{B}_1) \leftarrow (\pi_A\|\mathcal{A})\,] \leq 1/2 + \mathsf{negl}(n). \tag{3.61}$$

Theorem 3.4.4 (4-states QFactory is secure ([Coj+19]))**.** *Protocol 2 satisfies 4-states basis blindness with respect to the ideal list of states*

$$S = \{H^{B_1}X^{B_2}\,|0\rangle\}_{B_1,B_2} = \{|0\rangle\,,|1\rangle\,,|+\rangle\,,|-\rangle\}. \tag{3.62}$$

Definition 3.4.5 (8-states basis blindness)**.** *Similarly, we say that a protocol (π_A, π_B) achieves **basis-blindness** with respect to an ideal list of 8 states $S = \{S_{L_1,L_2,L_3}\}_{(L_1,L_2,L_3)\in\{0,1\}^3}$ if:*

1. *S is the set of states that the protocol outputs, i.e.,*

$$\Pr\,[|\phi\rangle = S_{L_1,L_2,L_3} \in S \mid ((L_1,L_2,L_3),|\phi\rangle) \leftarrow (\pi_A\|\pi_B)] = 1, \tag{3.63}$$

2. *and if no information is leaked about the "basis" bits (L_2, L_3) of the output state of the protocol, i.e for all QPT adversary $\mathcal{A}$, it holds that*

$$\Pr[\,L_2 = \tilde{L_2} \text{ and } L_3 = \tilde{L_3} \mid ((L_1,L_2,L_3),(\tilde{L_2},\tilde{L_3})) \leftarrow (\pi_A\|\mathcal{A})\,] \leq 1/4 + \mathsf{negl}(n). \tag{3.64}$$

Theorem 3.4.6 (8-states QFactory is secure ([Coj+19])). *Protocol 3 satisfies 8-state basis blindness with respect to the ideal set of states*

$$S = \left\{ \left| +_{\pi L/4} \right\rangle \right\}_{L \in \{0,\ldots,7\}} = \left\{ |+\rangle, \left| +_{\frac{\pi}{4}} \right\rangle, \ldots, \left| +_{\frac{7\pi}{4}} \right\rangle \right\}. \tag{3.65}$$

3.4.2 Game-Based Blindness

We start with giving a formal definition of the game-based security of $\mathsf{UBQC}_{\mathsf{CC}}$.

Definition 3.4.7 (Blindness of $\mathsf{UBQC}_{\mathsf{CC}}$). *A $\mathsf{UBQC}_{\mathsf{CC}}$ protocol $\mathcal{P} = (P_C, P_S)$ is said to be (computationally) adaptively blind if no computationally bounded malicious server can distinguish between runs of the protocol with adversarially chosen measurement patterns on the same MBQC graph.*

In formal terms, $\mathcal{P}$ is said to be (computationally) adaptively blind if and only if for any quantum-polynomial-time adversary A it holds that

$$\begin{aligned} &\Pr\left[c' = c \;\middle|\; (\phi^{(1)}, \phi^{(2)}) \leftarrow A,\, c \leftarrow\!\$\, \{0,1\},\, \left\langle P_C(\phi^{(c)}), A \right\rangle,\, c' \leftarrow A\right] \\ &\qquad \leq \frac{1}{2} + \mathrm{negl}(\lambda), \end{aligned} \tag{3.66}$$

where λ is the security parameter, and $\left\langle P_C(\phi^{(c)}), A \right\rangle$ denotes the interaction of the two algorithms $P_C(\phi^{(c)})$ and A.

Remark 3.4.8. *Although, Definition 3.4.7 is written using the terminology of measurement-based model, it doesn't compromise the generality, as the model is universal and can be easily translated into a circuit model, because the measurement pattern and unitary operator have a one-to-one mapping.*

3.4.3 Implementing Classical-Client UBQC with QFactory

The UBQC protocol from [BFK09], where the quantum interaction is replaced by a $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$ protocol, is shown in Protocol 1. In this section, we replace the $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$ protocol with the concrete protocol proposed in [Coj+19]. This protocol, known by the name of 8-states QFactory[19] and described in Protocol 3, exactly emulates the capability of $\mathsf{RSP}_{\mathsf{CC}}^{8-\mathsf{states}}$. The resulting protocol contains a QFactory instance for each qubit that

[19] We refer here to the 8-states QFactory implementation with negligible abort probability, and superpolynomial parameters. This is necessary since our proof does not take the abort case into account for now.

would have been generated on the client's side. The keys to all QFactory instances are generated entirely independently by the client.

Unfortunately, considering the results from Section 3.3 there is no hope that the composable security of any $\mathsf{UBQC}_{\mathsf{CC}}$ may be achieved. Nonetheless, letting go of composability, we are able to prove the game-based security for this specific combination of protocols. This leads us to the main theorem of this section.

Theorem 3.4.9 (Game-based Blindness of QF-UBQC)**.** *The protocol resulting from combining the quantum-client* UBQC *protocol with QFactory is a (computationally) adaptively blind implementation of* $\mathsf{UBQC}_{\mathsf{CC}}$ *in the game-based model according to Definition 3.4.7. We call this protocol* QF-UBQC*.*

The proof of Theorem 3.4.9 which will be given in the remainder of this section follows two main ideas:

1. Every angle used in the UBQC protocol has only eight possible values, and can, therefore, be described by three bits. In the protocol, the first bit is the one for which QFactory *cannot* guarantee blindness. Fortunately, the additional one-time padding in UBQC allows analyzing the blindness of the protocol independently of the blindness of exactly this first bit. Therefore, it suffices to rely on the blindness of the last two bits which is conveniently guaranteed by QFactory and the hardness of LWE.

2. To analyze the leakage about the last two bits during a QFactory run, it is sufficient to notice that the leakage is equal to a ciphertext under an LWE-based encryption scheme. The semantic security of this encryption scheme and the hardness assumption for LWE guarantee that this leakage is negligible and can be omitted.

In more detail, the 8-states QFactory protocol which is used here consists of two combined runs of 4-states QFactory, each contributing with a single blind bit to the three-bit angles used in the UBQC protocol. Recall from Theorem 3.4.2 and Theorem 3.4.6 the formulae for how these angles from the 4-states protocol are combined in the 8-states protocol. If B_1 is the hidden bit of the first 4-states QFactory instance and B_1' the hidden bit of the second instance, then we obtain

$$L_1 = B_2' \oplus B_2 \oplus [B_1 \cdot (s_1 \oplus s_2)],\ L_2 = B_1' \oplus [(B_2 \oplus s_2) \cdot B_1],\ L_3 = B_1, \tag{3.67}$$

where $L = L_1L_2L_3 \in \{0,1\}^3$ is the description of the output state $\left|+_{L\frac{\pi}{4}}\right\rangle$, s_1, s_2 are computed by the server, and

$$B_2 = f(\mathsf{sk}, B_1, y, b), \qquad B_2' = f(\mathsf{sk}', B_1', y', b') \tag{3.68}$$

for some function f, QFactory secret keys $\mathsf{sk}, \mathsf{sk}'$, and server-chosen values y, b, y', b'.

The two 4-states QFactory instances now leak the ciphertext of B_1 and B_1', respectively. Given the semantic security of the encryption, after a run of 8-states QFactory, L_2 and L_3 remain hidden, while the blindness of L_1 cannot be guaranteed by QFactory. This fact is going to be useful in the following proof.

3.4.4 Single-Qubit QF-UBQC

We first prove the security of combining QFactory with UBQC on a single qubit.

Lemma 3.4.10 (Blindness in the single-qubit case)**.** *The protocol resulting from combining the quantum-client* UBQC *protocol with (8-states) QFactory is a (computationally) adaptively blind implementation of* $\mathsf{UBQC}_{\mathsf{CC}}$ *in the game-based model for MBQC computations on a single qubit.*

Proof. We start with the real protocol, describing the adaptive blindness of QFactory combined with single-qubit UBQC. In the following, we denote the set of possible angles by $M = \{j\pi/4, j = 0, \ldots, 7\}$. The encryption scheme that appears in Game 1 is the semantically secure public-key encryption scheme from [Reg09]. Note that the two key pairs are generated completely independently on the challenger's side.

GAME 1:

	Adversary		**Challenger**
1 :	Choose $\phi^{(1)}, \phi^{(2)} \in M$	$\xrightarrow{\phi^{(1)}, \phi^{(2)}}$	$c \leftarrow\$ \{0,1\}$
2 :			$B_1, B_1' \leftarrow\$ \{0,1\}$
3 :		$\xleftarrow{\mathsf{pk}, \mathsf{pk}', \mathrm{Enc}^{\mathsf{pk}}(B_1), \mathrm{Enc}^{\mathsf{pk}'}(B_1')}$	Generate key pairs $(\mathsf{sk}, \mathsf{pk}), (\mathsf{sk}', \mathsf{pk}')$
4 :		$\xrightarrow{y, b, y', b', s_1, s_2}$	$B_2 = f(\mathsf{sk}, B_1, y, b)$, $B_2' = f(\mathsf{sk}', B_1', y', b')$
5 :			$L_1 = B_2' \oplus B_2 \oplus [B_1 \cdot (s_1 \oplus s_2)]$
6 :			$L_2 = B_1' \oplus [(B_2 \oplus s_2) \cdot B_1]$
7 :			$L_3 = B_1$
8 :			$r \leftarrow\$ \{0,1\}$
9 :		$\xleftarrow{\delta}$	$\delta = \phi^{(c)} + L_3\pi/4 + L_2\pi/2 + L_1\pi + r\pi$
10 :		$\xrightarrow{s}$	
11 :	Compute guess $c' \in \{0,1\}$	$\xrightarrow{c'}$	Check $c' = c$?

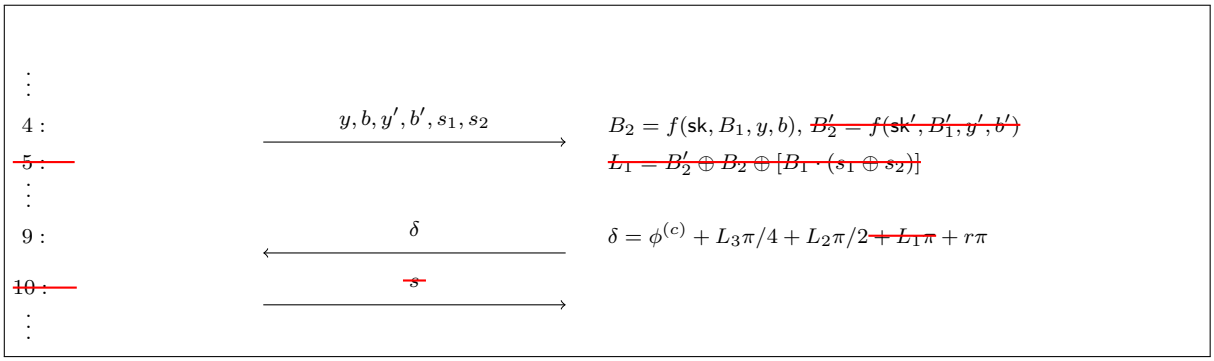
In the following, instead of repeating the redundant parts of subsequent games, we only present incremental modifications to Game 1. Every not explicitly written line is assumed to be identical to the previous game.

Clearly, since s is never used by the challenger, we can remove it from the protocol without distorting the success probability of the adversary. Next, we remove L_1 from the protocol and from the calculation of δ. L_1 is only used in the calculation of δ, which can be rewritten as

$$\delta = \phi^{(c)} + L_3\pi/4 + L_2\pi/2 + (L_1 + r)\pi. \tag{3.69}$$

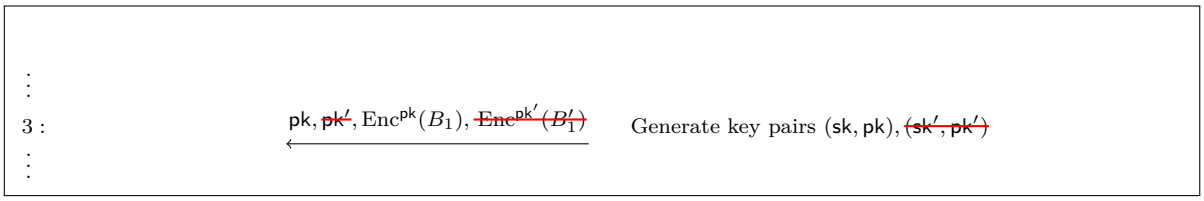
Since r is a uniform binary random variable with unique use in this line, $(L_1 + r)$ is still uniform over $\{0,1\}$. Therefore, removing L_1 leaves the distribution of the protocol outcome unchanged.

GAME 2:



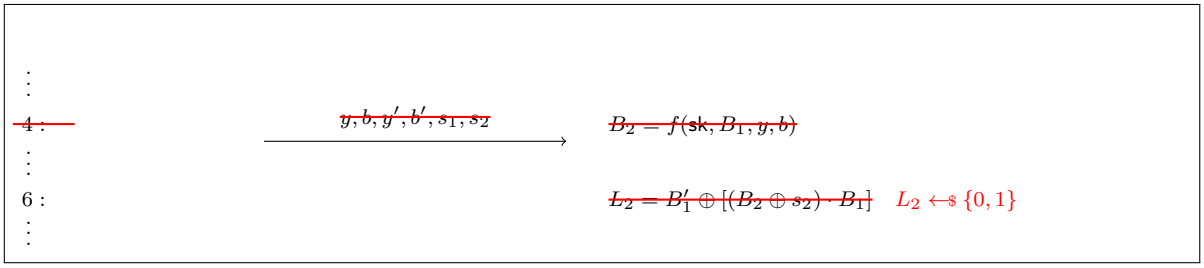
The next step introduces a (negligible) distortion to the success probability of the adversary. By the semantic security of the employed encryption scheme, no quantum-polynomial-time adversary can notice if the plaintext is replaced by pure randomness except with negligible probability, even if information about the original plaintext is leaked on the side. Therefore, replacing B'_1 in the encryption by independent randomness cannot lead to a significant change of the adversary's success probability. Further, since ciphertexts of independent randomness can be equally generated by the adversary herself (being in possession of the public key), we can remove the encryption of B'_1 from the protocol altogether.

GAME 3:



Next, note that B'_1 perfectly one-time pads the value of L_2. This breaks the dependency of L_2 on B_2, s_2 and B_1. It does not change the distribution of L_2, if L_2 is instead directly sampled uniformly from $\{0, 1\}$. Since B_2 is unused, we remove it in the following game, and y, b, y', b', s_1, s_2 can be ignored.

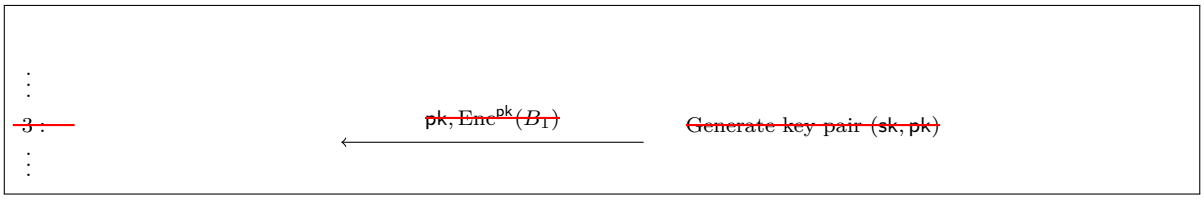
GAME 4:



By the same argument as for the transition from Game 2 to Game 3, we remove the encryption of B_1 from the following game. This introduces at most a negligible change in the success probability of the adversary.

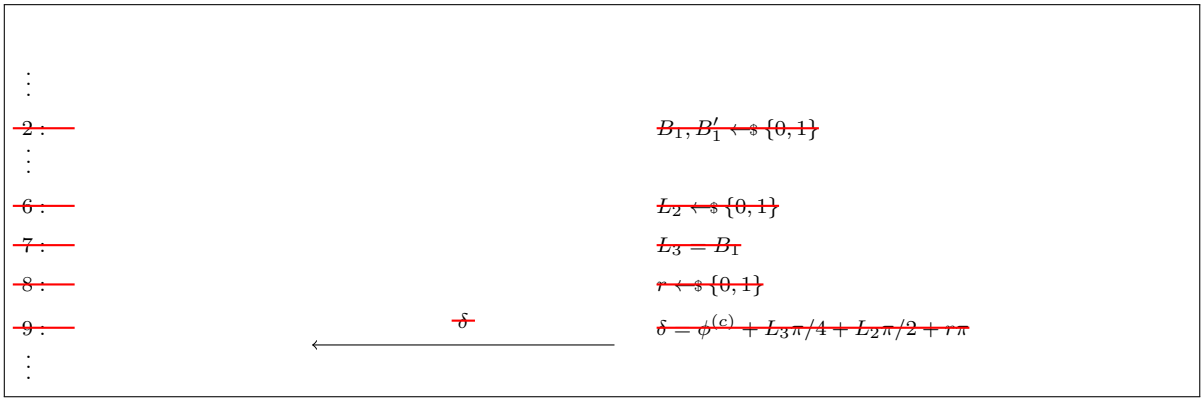
Finally, since the encryption scheme is not in use anymore, we can also remove the key generation and the message containing the public key without affecting the adversary's success probability.

GAME 5:



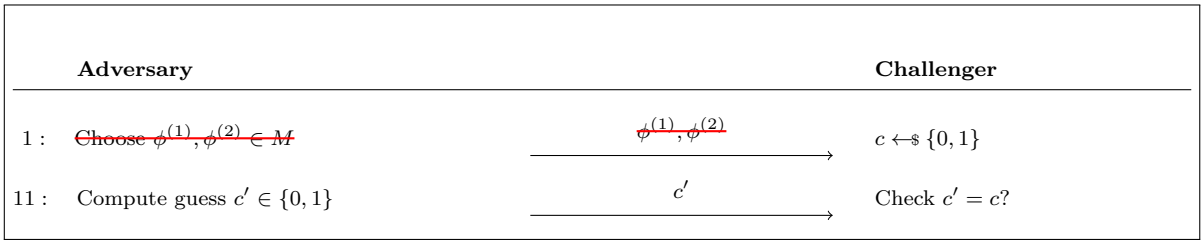
We now see that δ is a uniformly random number, L_2, L_3, and r being i.i.d. uniform bits. Therefore, the calculation and the message containing δ can be removed from the protocol without affecting the adversary.

GAME 6:



In Game 6, the inputs of the adversary are ignored by the challenger. Therefore, the computation angles $\phi^{(1)}$, $\phi^{(2)}$ can equally be removed from the protocol, leaving us with the final Game 7.

GAME 7:



Game 7 exactly describes the adversary's uninformed guess of the outcome of an

independent bit flip. Therefore, by a simple information-theoretic argument, any strategy for the adversary will lead to a success probability of exactly 1/2.

We summarize:

$$\text{Succ-Pr}_{\text{Game1}} = \text{Succ-Pr}_{\text{Game2}}, \quad |\text{Succ-Pr}_{\text{Game2}} - \text{Succ-Pr}_{\text{Game3}}| \leq \text{negl}(\lambda), \tag{3.70}$$

$$\text{Succ-Pr}_{\text{Game3}} = \text{Succ-Pr}_{\text{Game4}}, \quad |\text{Succ-Pr}_{\text{Game4}} - \text{Succ-Pr}_{\text{Game5}}| \leq \text{negl}(\lambda), \tag{3.71}$$

$$\text{Succ-Pr}_{\text{Game5}} = \text{Succ-Pr}_{\text{Game6}} = \text{Succ-Pr}_{\text{Game7}} = \frac{1}{2}, \tag{3.72}$$

and therefore we have $\left|\text{Succ-Pr}_{\text{Game1}} - \frac{1}{2}\right| \leq \text{negl}(\lambda)$ concluding the proof. □

3.4.5 General QF-UBQC

We extend the security proof from Section 3.4.4 to UBQC on polynomially-sized graphs, i.e. MBQC computations on a polynomial number of qubits. The proof works by induction over the number n of qubits in the graph. Lemma 3.4.10 with $n = 1$ serves as start of the induction. We continue with proving the induction step, assuming the security of QF-UBQC on graphs of size n and showing its security for any graph of size $n + 1$. The induction step works analogously to the proof of Lemma 3.4.10. In this way, the security of QF-UBQC on n qubits is reduced to the security of QF-UBQC on $n - 1$ qubits, which can be reduced to the security of QF-UBQC on even one qubit less. This chain continues down to the single-qubit case whose security was already established in Lemma 3.4.10. Every step in this chain adds at most a negligible probability to the adversary's advantage. Therefore, also any such chain of polynomial length adds no more than a negligible probability to the adversary's advantage in the single-qubit case, thereby showing the security of the protocol on n qubits. We now provide the full details of the induction step.

Details of the proof of Theorem 3.4.9. The proof works by induction over the number n of qubits in the graph. Lemma 3.4.10 with $n = 1$ serves as start of the induction. We continue with proving the induction step, assuming the security of QF-UBQC on graphs of size n and showing its security for any graph of size $n + 1$.

We first state some useful observations for the proof:

1. The existence of a *flow* on the MBQC graph induces a total order of all qubits in the graph, the order in which the qubits are measured. We subsequently assume

that in the protocol the qubits are processed in exactly this order.

2. Given this order on the qubits, the dependence of the computation angles δ_i on outcomes of measurement of other qubits takes a specific form, they solely depend on previous (corrected) measurement outcomes $\{\bar{s}_j, j < i\}$, i.e. outcomes of measurements of qubits smaller in the order induced by the flow. Since the exact form of this dependence does not matter for the following proof, we denote the update of the angles in the following general way:

$$\begin{aligned}\delta_i =&(-1)^{f_1(s_1,r_1,\ldots,s_{i-1},r_{i-1})}\phi_i + \theta_1\pi/4 + \theta_2\pi/2 + \theta_3\pi + r_i\pi \\ &+ f_2(s_1, r_1, \ldots, s_{i-1}, r_{i-1})\pi,\end{aligned} \tag{3.73}$$

with (deterministic families of) functions f_1 and f_2.

3. Given the previous observation, one can generalize the statement of the theorem to a family of protocols for any functions f_1 and f_2. For the remainder of the proof, we do hence not assume anything about these two functions, but simply take them as given. The actual statement of the theorem then follows as a special case, imposing that f_1 and f_2 describe the MBQC correction terms.

Given these observations, the rest of the proof works analogously to the proof of Lemma 3.4.10, removing one-by-one the ciphertexts of the two basis bits B_1, B_1' of the last QFactory instance, before removing the last measurement angle δ and reducing the protocol on $n+1$ qubits to the protocol on one qubit less. □

By the inductive nature of this proof, every qubit – and hence every QFactory instance – adds some negligible value to the success probability of the malicious adversary. This explains that the security only holds for polynomially-sized graphs. For an MBQC graph on a superpolynomial number of qubits, there are no guarantees anymore that these small errors don't add up to something constant. Having in mind that QFactory is trivially broken by exponential adversaries, it is clear that this is the best we can expect.

3.5 Appendix: Distance Measures for Quantum States

In this section, we give some distance measures for density matrices that are useful for the formal definition of RSP resources and their describability.

Lemma 3.5.1. *For any two self-adjoint trace-class operators ρ, σ it holds that*

$$\mathrm{Tr}(\rho\sigma) = \frac{1}{2}\left[\mathrm{Tr}\left(\rho^2\right) + \mathrm{Tr}\left(\sigma^2\right)\right] - \frac{1}{2}\left\|\rho - \sigma\right\|_{HS}^2, \tag{3.74}$$

where the Hilbert-Schmidt norm is defined as

$$\|A\|_{HS} = \sqrt{\mathrm{Tr}(A^* A)}. \tag{3.75}$$

Proof. This follows directly from the relation

$$(\rho - \sigma)^2 = \rho^2 - \rho\sigma - \sigma\rho + \sigma^2 \tag{3.76}$$

and the fact that ρ and σ are self-adjoint operators. □

The following lemma formalizes the following statement: If $\mathrm{Tr}(\rho\sigma)$ is close to 1, then both ρ and σ must be almost pure, and ρ and σ must be close. Note that Lemma 3.5.2 holds in particular for density matrices ρ and σ, despite being stated for a more general class of operators.

Lemma 3.5.2. *Let $\varepsilon \geq 0$ and $\mathrm{Tr}\left(\rho\sigma\right) \geq 1 - \varepsilon$ for two self-adjoint, positive semi-definite operators ρ, σ with trace less than 1. Then, it holds that*

1. $\mathrm{Tr}\left(\rho^2\right) \geq 1 - 2\varepsilon$,
2. $\mathrm{Tr}\left(\sigma^2\right) \geq 1 - 2\varepsilon$, *and*
3. $\left\|\rho - \sigma\right\|_{HS} \leq \sqrt{2\varepsilon}$.

Proof. 1. With the formula from Lemma 3.5.1, we infer that

$$\mathrm{Tr}(\rho\sigma) \leq \frac{1}{2}\left[\mathrm{Tr}\left(\rho^2\right) + \mathrm{Tr}\left(\sigma^2\right)\right] \leq \frac{1}{2}\left[\mathrm{Tr}\left(\rho^2\right) + 1\right], \tag{3.77}$$

using the non-negativity of the Hilbert-Schmidt norm and the fact that $\mathrm{Tr}\left(\sigma^2\right) \leq 1$. Hence,

$$\mathrm{Tr}\left(\rho^2\right) \geq 2\,\mathrm{Tr}\left(\rho\sigma\right) - 1 \geq 1 - 2\varepsilon. \tag{3.78}$$

2. Analogously to 1.

3. Using $\mathrm{Tr}\left(\rho^2\right) \leq 1$ and $\mathrm{Tr}\left(\sigma^2\right) \leq 1$, we obtain

$$\mathrm{Tr}\left(\rho\sigma\right) \leq 1 - \frac{1}{2}\left\|\rho - \sigma\right\|_{\mathrm{HS}}^2 \tag{3.79}$$

$$\Rightarrow \left\|\rho - \sigma\right\|_{\mathrm{HS}}^2 \leq 2\left(1 - \mathrm{Tr}\left(\rho\sigma\right)\right) \leq 2\varepsilon, \tag{3.80}$$

which implies the claim. □

Lemma 3.5.3. *Let λ be a security parameter and let ρ, σ be two density matrices of finite and fixed dimension. Then, the following statements are equivalent:*

1. $\mathrm{Tr}\left(\rho^2\right) \geq 1 - \mathrm{negl}(\lambda)$, $\mathrm{Tr}\left(\sigma^2\right) \geq 1 - \mathrm{negl}(\lambda)$, *and* $\mathrm{TD}\left(\rho - \sigma\right) \leq \mathrm{negl}(\lambda)$,

2. $\mathrm{Tr}\left(\rho\sigma\right) \geq 1 - \mathrm{negl}(\lambda)$,

where TD *denotes the trace distance.*

Proof. One direction of the equivalence follows directly from Lemma 3.5.2. The other direction follows from the formula in Lemma 3.5.1 and the fact that in finite-dimensional spaces the trace norm is equivalent to the Hilbert-Schmidt norm. □

Lemma 3.5.4. *Let $\varepsilon_1, \varepsilon_2 \geq 0$. Let further $\mathrm{Tr}\left(\rho_1\rho_2\right) \geq 1 - \varepsilon_1$ and $\mathrm{Tr}\left(\rho_2\rho_3\right) \geq 1 - \varepsilon_2$ for self-adjoint, positive semi-definite operators ρ_1, ρ_2, ρ_3 with trace less than 1. Then it holds that $\mathrm{Tr}\left(\rho_1\rho_3\right) \geq 1 - 3\left(\varepsilon_1 + \varepsilon_2\right)$.*

Proof. From Lemma 3.5.2 we know that $\mathrm{Tr}\left(\rho_1^2\right) \geq 1 - 2\varepsilon_1$, $\mathrm{Tr}\left(\rho_3^2\right) \geq 1 - 2\varepsilon_2$, and

$$\left\|\rho_1 - \rho_2\right\|_{\mathrm{HS}} \leq \sqrt{2\varepsilon_1}, \quad \left\|\rho_2 - \rho_3\right\|_{\mathrm{HS}} \leq \sqrt{2\varepsilon_2}. \tag{3.81}$$

By the triangle inequality for the Hilbert-Schmidt norm, it follows readily that

$$\left\|\rho_1 - \rho_3\right\|_{\mathrm{HS}} \leq \sqrt{2\varepsilon_1} + \sqrt{2\varepsilon_2} \tag{3.82}$$

and therefore

$$\left\|\rho_1 - \rho_3\right\|_{\mathrm{HS}}^2 \leq \left(\sqrt{2\varepsilon_1} + \sqrt{2\varepsilon_2}\right)^2 = 2\varepsilon_1 + 2\varepsilon_2 + 4\sqrt{\varepsilon_1}\sqrt{\varepsilon_2} \leq 4\left(\varepsilon_1 + \varepsilon_2\right) \tag{3.83}$$

where we applied the inequality of the geometric mean to obtain the last bound. Using

the formula from Lemma 3.5.1, we then conclude that

$$\begin{aligned}\mathrm{Tr}\left(\rho_1\rho_3\right) &= \frac{1}{2}\left[\mathrm{Tr}\left(\rho_1^2\right)+\mathrm{Tr}\left(\rho_3^2\right)\right]-\frac{1}{2}\left\|\rho_1-\rho_3\right\|_{\mathrm{HS}}^2 \\ &\geq \frac{1}{2}\left[1-2\varepsilon_1+1-2\varepsilon_2\right]-\frac{1}{2}4\left(\varepsilon_1+\varepsilon_2\right) \geq 1-3\left(\varepsilon_1+\varepsilon_2\right), \end{aligned} \tag{3.84}$$

which implies the claim. □

Chapter 4

Verifying BQP Computations with Minimal Overhead

With the development of delegated quantum computation, clients will want to ensure confidentiality of their data and algorithms, and the integrity of their computations. While protocols for blind and verifiable quantum computation exist, they suffer from high overheads and from over-sensitivity: When running on noisy devices, imperfections trigger the same detection mechanisms as malicious attacks, resulting in perpetually aborted computations. We introduce the first blind and verifiable protocol for delegating BQP computations to a powerful server with repetition as the only overhead. It is composable and statistically secure with exponentially-low bounds and can tolerate a constant amount of global noise.

This chapter is based on the papers "Securing Quantum Computations in the NISQ Era" [Kas+21] and "Verifying BQP Computations on Noisy Devices with Minimal Overhead" [Lei+21], published in PRX Quantum, which are joint work with Elham Kashefi, Luka Music, and Harold Ollivier.

4.1 Introduction

Remotely accessible quantum computing platforms free clients from the burden of maintaining complex physical devices in house. Yet, when delegating computations, they want their data and algorithms to remain private, and that these computations are executed as specified. Several methods have been devised to achieve this (e.g. [BFK10;

FK17], see [GKK19] for a review). Nonetheless, a practical solution remains to be found as all known protocols are too sensitive to noise. Indeed, they have been designed for perfect devices, thus aborting as soon as the smallest deviation is detected. Unfortunately, replacing such machines by even slightly noisy ones would make the verification procedure abort constantly, mistaking plain imperfections for the signature of malicious behaviour.

For dealing with this over-sensitivity, previous research either gave up on blindness [GHK18], imposed restrictions on the noise model [KD19], switched to a setting with two non-communicating servers and classical clients [MF13b], or introduced computational assumptions [Mah18b]. Yet, these protocols either only achieve inverse-polynomial security or obtain exponential security by requiring an additional fault-tolerant encoding of the computation on top of the one used to suppress device noise.

We tackle this problem for BQP computations – i.e. the class of decision problems that quantum computers can solve efficiently – by introducing a protocol that provides noise-robustness, verification, blindness and delegation. The protocol repeats the client's computation framed in the Measurement-Based Quantum Computation (MBQC) model – a natural choice for delegating computations – several times in a blind fashion while interleaving these executions with test rounds which aim at detecting a dishonest behaviour of the server. A final majority vote over the computation rounds mitigates possible errors, thus providing the desired robustness.

Combined with blindness, this forces the server to attack at least a constant fraction of the rounds to corrupt the computation, hence increasing its chances of getting caught by the tests. Information theoretic security is proven in the composable framework of Abstract Cryptography [MR11], ensuring security is not jeopardised by sequential or simultaneous instantiations with other protocols.

Crucially, our protocol has *no space overhead* for each round when compared to the insecure computation in the MBQC model: the only price to pay for exponential security and correctness is a *polynomial number of repetitions* of computations similar to the unprotected one. This lets the client use the full extent of the available hardware for its computational tasks, and any increase in the capabilities of the quantum devices can be used entirely to scale-up these computations. These properties make it, to our knowledge, the first experimentally realisable solution for verification of BQP computations, thus going beyond experimental feasibility demonstrations of verifiable building blocks [Bar+12; Bar+13; Gre+16; McC+16] and potentially serving as a blueprint for the development of future quantum network applications.

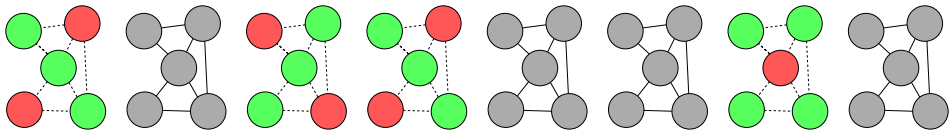


Figure 4.1: An example of rounds of the proposed protocol. Graphs in grey denote computation rounds while graphs containing red nodes (traps) and green nodes (dummies) are test rounds. Each qubit is always included in one type of test round. The Server remains completely oblivious of the differences between the rounds, which are solely known to the Client.

4.2 Noise-Robust Verifiable Protocol

Our Noise-Robust VBQC Protocol is formally defined in Protocol 4 where test rounds are used in conjunction with computation rounds to provide verifiability. We introduce it more intuitively in the next paragraphs and discuss the features that make it suitable for practical purposes.

BQP Computations. The complexity class BQP contains the decisions problems that can be solved with bounded error probability using a polynomial size quantum cricuit. More formally, a language L is in BQP if there is a family of polynomial size quantum circuits which decides the language with an error probability of at most p. The chosen value for p is arbitrary as long as it is fixed and smaller than $1/2$, and is usually taken to be $1/3$. Hence, a BQP computation for L will have output $F(x) = 1$ for $x \in L$ with probability at least $1 - p$, while it will have output $F(x) = 0$ for $x \notin L$ with probability at least $1 - p$. In the following, for a given BQP computation, p will be referred to as the *inherent error probability* to distinguish it from errors due to external causes such as the use of noisy devices.

Trap Insertion for BQP Computations. Because BQP computations have classical inputs and classical outputs, there exists a more economical trap insertion than what is available for quantum input and quantum output computations. More concretely, it does not require any enlargement of the graph to insert traps alongside the computation. Rather, the idea is to interleave pure *computation rounds* (i.e. without inserted traps) and *pure test rounds* (i.e. only made up of traps).

Given a UBQC computation defined by a graph G, we construct test rounds based on a k-colouring $\{V_i\}_{i\in[k]}$ of G. A partition of a graph in k sets – called colours – is a valid k-colouring if all adjacent vertices in the graph have different colours. Therefore,

Algorithm 4 Noise-Robust VBDQC for BQP Computations

Client's Inputs: Angles $\{\phi_v\}_{v\in V}$ and flow f on graph G, classical input to the computation $x \in \{0,1\}^{\#I}$ (where $\#X$ is the size of X).

Protocol:

1. The Client chooses uniformly at random a partition (C, T) of $[n]$ ($C \cap T = \emptyset$) with $\#C = d$, the sets of indices of the computation and test rounds respectively.
2. For $j \in [n]$, the Client and the Server perform the following sub-protocol (the Client may send message Redo_j to the Server before step 2.c while the Server may send it to the Client at any time, both parties then restart round j with fresh randomness):
   (a) If $j \in T$ (test), the Client chooses uniformly at random a colour $\mathsf{V}_j \in_R \{V_k\}_{k\in[K]}$ (this is the set of traps for this test round).
   (b) The Client sends $\#V$ qubits to the Server. If $j \in T$ and the destination qubit $v \notin \mathsf{V}_j$ is a non-trap qubit (therefore a dummy), then the Client chooses uniformly at random $d_v \in_R \{0,1\}$ and sends the state $|d_v\rangle$. Otherwise, the Client chooses at random $\theta_v \in_R \Theta$ and sends the state $|+_{\theta_v}\rangle$.
   (c) The Server performs a CZ gate between all its qubits corresponding to an edge in the set E.
   (d) For $v \in V$, the Client sends a measurement angle δ_v, the Server measures the appropriate corresponding qubit in the δ_v-basis, returning outcome b_v to the Client. The angle δ_v is defined as follows:
      - If $j \in C$ (computation), it is the same as in UBQC, computed using the flow and the computation angles $\{\phi_v\}_{v\in V}$. For $v \in I$ (input qubit) the Client uses $\tilde{\theta}_v = \theta_v + x_v\pi$ in the computation of δ_v.
      - If $j \in T$ (test): if $v \notin \mathsf{V}_j$ (dummy qubit), the Client chooses it uniformly at random from Θ; if $v \in \mathsf{V}_j$ (trap qubit), it chooses uniformly at random $r_v \in_R \{0,1\}$ and sets $\delta_v = \theta_v + r_v\pi$.
3. For all $j \in T$ (test round) and $v \in \mathsf{V}_j$ (traps), the Client verifies that $b_v = r_v \oplus d_v$, where $d_v = \bigoplus_{i\in N_G(v)} d_i$ is the sum over the values of neighbouring dummies of qubit v. Let c_{fail} be the number of failed test rounds (where at least one trap qubit does not satisfy the relation above), if $c_{fail} \geq w$ then the Client aborts by sending message Abort to the Server.
4. Otherwise, let y_j for $j \in C$ be the classical output of computation round j (after corrections from measurement results). The Client checks whether there exists some output value y such that $\#\{y_j \,|\, j \in C,\, y_j = y\} > \frac{d}{2}$. If such a value y exists (this is then the majority output), it sets it as its output and sends message Ok to the Server. Otherwise it sends message Abort to the Server.

by definition, a k-colouring satisfies $\bigcup_{i=1}^{k} V_i = V$, and $\forall i \in [k]$, $\forall v \in V_i : N_G(v) \cap V_i = \emptyset$, where $N_G(v)$ are the neighbours of v in G. Hence, for each colour i, the Client can decide to insert traps for all vertices of V_i and dummies in all other positions. This defines the test round associated to colour i. These tests require the same sequence of operations for the Server as regular UBQC computations, making them undetectable.

Informal Presentation of the Protocol. Suppose the Client wishes to delegate a BQP computation corresponding to a measurement pattern on a graph G to the Server. The Client chooses a colouring $\{V_i\}_{i\in[k]}$ of G, and two integers d and t. All these parameters are fixed for a given instantiation of the protocol and are publicly available to both parties.

The Client runs the UBQC Protocol $n := t + d$ times successively. For d of the rounds chosen at random (computation rounds), the Client updates the measurement angles according to the measurement pattern of its desired computation. The remaining t rounds are test rounds. For each such test round, the Client secretly chooses a colour at random and sends traps for vertices of that colour and dummies everywhere else. The Client instructs the Server to measure all qubits as in computation rounds, but with the measurement angle of trap qubits corresponding to the basis they were prepared in and a random measurement basis for the dummies. Because the trap qubits are isolated from each other, they should remain in their initial state. A test round is said to have *passed* if all the traps yield the expected measurement results, and *failed* otherwise. Figure 4.1 depicts such possible succession of rounds.

At the end of the protocol, the Client counts the number of failed test rounds. If this number is higher than a given threshold w, it aborts the protocol by sending the message Abort to the Server [1]. Otherwise it sets the majority outcome of the computation rounds as its output and sends message Ok to the Server.

In this construction all rounds share the same underlying graph G, the same order for the measurements of qubits, and all angles are chosen from the same uniform distribution. We prove formally later that this implies blindness – i.e. the Server cannot distinguish computation and test rounds, nor tell which qubits are traps – which in turn makes this trap insertion strategy efficient to obtain verifiability. The parameters' range and

[1] w would typically be set by the Client given its *a priori* understanding of the quality of the Server. As explained in the Discussion, this does not affect security: a higher value would induce more rounds than necessary to achieve a given confidence level, while a lower value would risk aborting with high probability.

influence on verifiability and noise-robustness bounds are detailed in the next section.

Redo Feature. Because the Client or the Server may experience unintentional devices failures, they might wish to discard and redo a round $j \in [n]$. In this case, our protocol allows each party to send a Redo_j request to the other, in which case both parties simply repeat the exact same round albeit with fresh randomness. Redo_j requests are allowed only so long as the party asking for it is still supposed to be manipulating the qubits of round j. We show that this does not impact the blindness nor verifiability of the scheme. This means that a dishonest Server cannot use Redo requests to trick the Client into accepting an incorrect result. Such capability of our protocol is crucial in practice: without it, detected honest failures of devices happening during a test round would be counted as a failed test round, thus decreasing drastically the likelihood of successfully completing the protocol. Since concerned rounds can be safely repeated, the only consequence of experimental failures caught during an execution is an increase in the expected number of rounds.

Exponential Security Amplification. The above approach to trap insertion is efficient as the only overhead is the repetition of the same sub-protocol. Yet, using a single computation round and $n-1$ test rounds would leave at least $1/n$ chance for the Server to corrupt the computation. The only previously-known method to obtain an exponentially-low cheating probability was to insert traps into a single computation round at the expense of drastically increasing the graph's complexity and then using fault-tolerant encoding on top to amplify the security. By restricting the computation to BQP computations, we prove that a classical repetition error-correcting code is sufficient to achieve exponentially-low cheating probability. This amplification technique is common in purely classical scenarios where attacks can be classically correlated across various rounds. Although this claim has been made as well in the quantum case in previous works [FK17; KW17b; KD19], it remained up to now unproven. The difficulty, which we address below, is that quantum attacks entangled across rounds are much more powerful than what classical correlations allow.

4.3 Security Results and Noise Robustness

This section presents the protocol's security properties in the Abstract Cryptography Framework of [MR11] (AC) and its noise-robustness on honest devices.

4.3.1 Overview of Security Analysis

Security Analysis. In AC, security is defined as indistinguishability between an Ideal Resource, which is secure by definition, and its real-world implementation, i.e. the protocol. This framework ensures a higher standard of security than in other approaches (see e.g. [Kön+07] and Section 5.1 of [PR14]) and is inherently composable, meaning that security holds when the protocol is repeated sequentially or in parallel with others. This property is crucial as delegated protocols are important stepping stones towards more complex functionalities (e.g. subroutine for building Multi-Party Quantum Computation protocols [Kap+21]).

Our security proof uses the results of [Dun+14] that reduce the composable security of a Verifiable Delegated Quantum Computation Protocol to four *stand-alone criteria*:

- ϵ_{cor}-local-correctness: the protocol with honest players produces the expected output;
- ϵ_{bl}-local-blindness: the Server's state at the end of the protocol is indistinguishable from the one which it could have generated on its own;
- ϵ_{ver}-local-verifiability: either the Client accepts a correct computation or aborts the protocol.
- ϵ_{ind}-independent-verification: the Server can determine on its own, using the transcript of the protocol and its internal registers, whether the Client will decide to abort or not.

Then, the Local-Reduction Theorem (Corollary 6.9 from [Dun+14]) states that if a protocol implements a unitary transformation on classical inputs and is ϵ_{cor}-locally-correct, ϵ_{bl}-locally-blind and ϵ_{ver}-locally-verifiable with ϵ_{ind}-independent verification, then it is ϵ-composably-secure with:

$$\epsilon = max\{\epsilon_{sec}, \epsilon_{cor}\} \text{ and } \epsilon_{sec} := 4\sqrt{2\epsilon_{ver}} + 2\epsilon_{bl} + 2\epsilon_{ind}. \tag{4.1}$$

With this at hand, we can state our main result:

Theorem 4.3.1 (Security of Protocol 4)**.** *For $n = d + t$ such that d/n and t/n are fixed in $(0, 1)$ and w such that w/t is fixed in $(0, \frac{1}{k} \cdot \frac{2p-1}{2p-2})$, where p is the inherent error probability of the BQP computation, Protocol 4 with d computation rounds, t test rounds,*

and a maximum number of tolerated failed test rounds of w is ϵ-composably-secure with ϵ exponentially small in n.

Simple Upper-Bound on the Probability of Failure. The ϵ_{ver}-local-verifiability amounts to upper bound the probability that an erroneous result is accepted by ϵ_{ver}. Given a BQP computation that decides whether x belongs or not to the language L, our protocol would yield the correct result after the majority vote whenever less than $d/2$ computation rounds yield $F(x) \oplus 1$. These erroneous results can be due to malicious behaviours of the server, to its use of noisy devices or to inherent errors of the BQP algorithm. It is expected that, in pd computation rounds, the BQP computation will give an inherently erroneous result, and that this will happen for a fraction greater than p only with negligible probability. Therefore, the result obtained by running our protocol will be correct whenever it is possible to guarantee that there is a negligible probability that the server corrupts more than $\left(\frac{1}{2} - p - \varphi\right) d$ computation runs for some $\varphi > 0$. To this end, we use the trapification paradigm. First, it ensures that each non-trivial deviation to the computation will be detected by at least one of the k possible types of test rounds. Second, because the deviations are distributed equally among test and computation runs, we can conclude that if less than $(\frac{1}{2} - p - \varphi - \varepsilon_1)t$ test runs are corrupted for some $\varepsilon_1 > 0$, then less than $(\frac{1}{2} - p - \varphi)d$ computations are corrupted with overwhelming probability. This implies that setting $w = (\frac{1}{k} - \varepsilon_2)(\frac{1}{2} - p - \varphi - \varepsilon_1)t$ for $\varepsilon_2 > 0$ yields an exponentially low probability of failure. Since $\varphi, \varepsilon_1, \varepsilon_2$ can be chosen arbitrarily small, we conclude that ϵ_{ver} can be made negligible for $0 < w/t < \frac{1}{k}(\frac{1}{2} - p)$.

Improved Upper-Bound on the Probability of Failure. The former bound can be improved by realising that some situations leading to incorrect results were double counted. Indeed, we need to consider inherent errors from the BQP computation solely for the computation rounds that where unaffected by the Server's malicious behaviour. This is due to the blindness of the scheme ensuring that the Server's deviation will be distributed equally among computation rounds with or without inherent errors. Denoting by m the total number of rounds affected by the Server's deviation, we expect $(md + (n - m)pd)/n$ computation rounds to be erroneous. The first term comes from deviations of the Server, while the second comes from inherent errors in the BQP computation when the Server has not deviated on these rounds. Requiring this quantity to be below $d/2$ amounts to guarantee that $m < \frac{2p-1}{2p-2}n$, which can be obtained following the line of arguments given in the previous paragraph whenever w satisfies

$0 < w/t < \frac{1}{k} \cdot \frac{2p-1}{2p-2}$.

Local-Correctness on Honest-but-Noisy Devices. None of the stand-alone criteria introduced above consider device imperfections. In fact, the analysis of correctness, blindness and verification makes no distinction between device imperfections and potentially malicious behaviours. Although satisfactory – these properties make our protocol a concrete implementation of the Ideal Resource for Verifiable Delegated Quantum Computation –, it could still fall short of expectations in terms of usability because non malicious device imperfections could cause unintentional aborts. Fortunately, for a class of realistic imperfections, our protocol is capable of correcting their impact and accepts with high probability. In such case, the final outcome is the same as that obtained on noiseless devices with honest participants.

This additional *noise-robustness* property, the main innovation of this chapter, means that Protocol 4 also satisfies the local-correctness property with negligible ϵ_{cor} for noisy but honest Client and/or Server. This property holds under the following restrictions:

- The noise can be modelled by round-dependent Markovian processes – i.e. a possibly different arbitrary CPTP map acting on each round.
- The probability that at least one of the trap measurements fails in any single test round is upper-bounded by some constant $p_{max} < \frac{1}{k} \cdot \frac{2p-1}{2p-2}$ and lower-bounded by $p_{min} \leq p_{max}$.

Theorem 4.3.2 states that, in order for the protocol to terminate correctly with overwhelming probability on these noisy devices, w should be chosen such that $w/t > p_{max}$. Conversely, for any choice of $w/t < p_{min}$, we show that the protocol aborts with overwhelming probability.

Theorem 4.3.2 (Local-Correctness of VDQC Protocol on Noisy Devices, Informal)**.** *As before, p denotes the inherent error probability for the BQP computation. Assume a Markovian round-dependent model for the noise on Client and Server devices and let $p_{min} \leq p_{max} < \frac{1}{k} \cdot \frac{2p-1}{2p-2}$ be respectively a lower and an upper-bound on the probability that at least one of the trap measurement outcomes in a single test round is incorrect. If $w/t > p_{max}$, Protocol 4 is ϵ_{cor}-locally-correct with exponentially low ϵ_{cor}. On the other hand, if $w/t < p_{min}$, then the probability that Protocol 4 terminates without aborting is exponentially low.*

Using again the Local-Reduction Theorem from [Dun+14], this new bound concerning local-correctness on noisy devices can be combined with noise-independent blindness, input-independent verification and verifiability, to yield a composably secure protocol for $\epsilon = max\{\epsilon_{sec}, \epsilon_{cor}\}$. Here, ϵ might depend on the noise level of the devices through ϵ_{cor}.

4.3.2 Formal Security Definitions

We model N-round two party protocols between players A (the honest Client) and B (the potentially dishonest Server) as a succession of $2N$-CPTP maps $\{\mathcal{E}_i\}_{i\in[1,N]}$ and $\{\mathcal{F}_j\}_{j\in[1,N]}$. The maps $\{\mathcal{E}_i\}_i$ act on $\mathcal{A}$, A's register, and $\mathcal{C}$, a shared communication register between A and B. Similarly, the maps $\{F_j\}_j$ act on $\mathcal{B}$ and $\mathcal{C}$. Note that $\mathcal{B}$ and the maps $\{\mathcal{F}_j\}_j$ can be chosen arbitrarily by B and thus, unless B is specified to be behaving honestly, there is no guarantee that they are those implied by our protocol. Since we are only interested in protocols where A is providing a classical input x, we will equivalently write the input as the corresponding computational basis state $|x\rangle$ used to initialize $\mathcal{A}$, whereas $\mathcal{B}$ and $\mathcal{C}$ are initialized in a fixed state $|0\rangle$.

Below, we denote by $\Delta(\rho,\sigma) = \frac{1}{2}\|\rho - \sigma\|$, the distance on the set of density matrices induced by the trace norm $\|\rho\| = \mathrm{Tr}\sqrt{\rho^\dagger\rho}$. We first define $\mathcal{S}$ the ideal resource for verifiable delegated quantum computation and then the local-properties from [Dun+14].

Ideal Resource for Verifiable Delegated Quantum Computation. The ideal resource $\mathcal{S}$ has interfaces for two parties, A and B. The A-interface takes two inputs: a classical input string x and the description of $\mathcal{U}$, the computation to perform. The B-interface is filtered by a bit b. When $b = 0$, there is no further legitimate input from B, while for $b = 1$, it is allowed to send a bit c that determines the output of the computation available at A's interface. When $b = 0$ or $c = 0$, the output at A's interface is equal to $\mathcal{M}_{Comp} \circ \mathcal{U}(|x\rangle)$, where $\mathcal{M}_{Comp}$ is the computational basis measurement. This corresponds to a "no cheating" behaviour. When $c = 1$, B decided to cheat and A receives the Abort message which can be given as a quantum state of $\mathcal{A}$ which is taken orthogonal to any other possible output state. At B's interface, $\mathcal{S}$ outputs nothing for $b = 0$ while for $b = 1$, B receives $l(\mathcal{U}, x)$, the permitted leakage. For generic MBQC computations, the permitted leakage is set to G, the graph used in the computation. When G is a universal graph for MBQC computation, the permitted leakage reduces to an upper-bound on the size of the computation $\#\mathcal{U}$.

For this ideal resource, the blindness is an immediate consequence of the server receiving at most the permitted leak, while verifiability is a consequence of the computation being correct when the server is not cheating while being aborted otherwise.

ϵ_{cor}-Local-Correctness. Let $\mathcal{P}_{AB}$ be a two-party protocol as defined above with the honest CPTP maps for players A and B. We say that such a protocol implementing $\mathcal{U}$ is ϵ_{cor}-*locally-correct* if for all possible inputs x for A we have:

$$\Delta\left(\mathrm{Tr}_B \circ \mathcal{P}_{AB}(|x\rangle), \mathcal{U}(|x\rangle)\right) \leq \epsilon_{cor} \tag{4.2}$$

ϵ_{bl}-Local-Blindness. Let $\mathcal{P}_{AB}$ be a two-party protocol as defined above, and where the maps $\{\mathcal{E}_i\}_i$ are the honest maps. We say that such protocol is ϵ_{bl}-*locally-blind* if, for each choice of $\{\mathcal{F}_i\}_i$ there exists a CPTP map $\mathcal{F}' : L(\mathcal{B}) \to L(\mathcal{B})$ such that, for all inputs x for A, we have:

$$\Delta\left(\mathrm{Tr}_A \circ \mathcal{P}_{AB}(\rho), \mathcal{F}' \circ \mathrm{Tr}_A(|x\rangle)\right) \leq \epsilon_{bl} \tag{4.3}$$

ϵ_{ind}-Independent Verification. Let $\mathcal{P}_{AB}$ be a verifiable 2-party protocol as defined above, where the maps $\{\mathcal{E}_i\}_i$ are the honest maps. Let $\bar{B}$ be a qubit extending B's register and initialized in $|0\rangle$. Let $\mathcal{Q}_{A\bar{B}} : L(\mathcal{A} \otimes \bar{\mathcal{B}}) \to L(\mathcal{A} \otimes \bar{\mathcal{B}})$ be a CPTP map which, conditioned on $\mathcal{A}$ containing the state $|\mathsf{Abort}\rangle$, switches the state in $\bar{\mathcal{B}}$ from $|0\rangle$ to $|1\rangle$ and does nothing in the other cases.

We say that such a protocol's verification procedure is ϵ_{ind}-*independent* from player A's input if there exists CPTP maps $\mathcal{F}'_i : L(\mathcal{C} \otimes \mathcal{B} \otimes \bar{\mathcal{B}}) \to L(\mathcal{C} \otimes \mathcal{B} \otimes \bar{\mathcal{B}})$ such that:

$$\Delta\left(\mathrm{Tr}_A \circ \mathcal{Q}_{A\bar{B}} \circ \mathcal{P}_{AB}(\rho), \mathrm{Tr}_A \circ \mathcal{P}'_{AB\bar{B}}(\rho)\right) \leq \epsilon_{ind} \tag{4.4}$$

where

$$\mathcal{P}'_{AB\bar{B}} := \mathcal{E}_1 \circ \mathcal{F}'_1 \circ \ldots \circ \mathcal{E}_n \circ \mathcal{F}'_n \tag{4.5}$$

ϵ_{ver}-Local-Verifiability. Let $\mathcal{P}_{AB}$ be 2-party protocols as defined above where the maps for A are the honest maps, while the maps $\{\mathcal{F}_j\}_j$ for B are not necessarily corresponding to the ideal (honest) ones. Let x be the input given by A in the form of a computational state $|x\rangle$ and $\mathcal{U}$ the computation it wants to perform. The protocols $\mathcal{P}_{AB}$ are ϵ_{ver}-*locally-verifiable* for A if for each choice of CPTP maps $\{\mathcal{F}_j\}_j$, there exists

$p \in [0, 1]$ such that we have:

$$\Delta\Big(\mathrm{tr}_B\, \mathcal{P}_{AB}(|x\rangle),\, p\mathcal{U}(|x\rangle) + (1-p)\,|\mathsf{Abort}\rangle\langle\mathsf{Abort}|\Big) \leq \epsilon_{ver}. \tag{4.6}$$

4.3.3 Composable Security

In the paragraphs below, we show that our protocol satisfies each of the stand-alone criteria before combining them to get composable security.

Perfect Local-Correctness. On perfect (non-noisy) devices, local-correctness is implied by the correctness of the underlying UBQC Protocol. This is because all the completed computation rounds correspond to the same deterministic UBQC computation, and that on such devices, general UBQC Protocols have been proven to be perfectly correct [BFK10; Dun+14]. Thus $\epsilon_{cor} = 0$.

Perfect Local-Blindness. In case the computation is accepted, each round looks exactly like a UBQC computation to the Server. Therefore the blindness comes directly from the composability of the various UBQC rounds that make our protocol [Dun+14]. In case the computation is aborted, we need to take into account the fact that a possibly malicious Server could deduce the position of a trap qubit. That could be the case if it attacked a single position in the test rounds and got caught. Yet, as the position of the traps is not correlated to the input nor to the computation itself, knowing it does not grant additional attack capabilities to the Server, and blindness is recovered again as a consequence of the blindness of UBQC. Subsequently, we give more detailed arguments and show that Redo requests have no effect on the local-blindness of the scheme.

Proof. To prove that Equation 4.3 holds for $\epsilon_{bl} = 0$, first note that at the end of our protocol, the Client A reveals to the Server B whether the computation was accepted or aborted. Hence, each case can be analyzed separately. Second, we show that the interrupted rounds that have triggered a Redo can be safely ignored. Indeed, each one of them is the begining of an interrupted UBQC computation, and, because UBQC is composable and perfectly blind [Dun+14], no information can leak to the Server through the transmitted qubits. In addition, our protocol restricts the honest party A in its ability to emit Redo requests, so that no correlations are created between the index of the interrupted rounds and $\mathcal{U}$ or the secret random parameters used in the rounds (angle and measurement padding, and trap preparations). As a consequence, from the

point of view of B, the state of the interrupted rounds is completely independent of the state of the non-interrupted ones and does not contain information regarding the input, computation or secret parameters. That is, its partial trace over A can be generated by B alone.

For the non-interrupted rounds, we can invoke the same kind independence argument between the computation rounds and the test rounds. As a result blindness of our protocol stems from the blindness of the underlying computation rounds. In case the full protocol is a success, we can rely on the composability of the perfect blindness of each UBQC computation round to have perfect local-blindness. For an abort, we can consider a situation that is more advantageous for B by supposing that alongside the Abort message sent by A, it also gives away the location of the trap qubits. In this modified situation, the knowledge of the computation being aborted does not bring additional information to B as it only reveals that one of the attacked position was a trap qubit, which B now already knows. Using our independence argument between trap location on the one hand and the inputs, computation and other secret parameters, we conclude that revealing the location of the trap qubits does not affect the blindness of the computation rounds. Hence, using composability again and combining the abort and accept cases, we arrive at Equation 4.3 with $\epsilon_{bl} = 0$. □

Perfect Local-Independent-Verification. Because in our protocol, the Client shares with the Server whether the computation was a success or an abort, this is trivially verified.

Exponential Local-Verifiability. Local-verifiability is satisfied if any deviation by the possibly malicious Server yields a state that is ϵ_{ver}-close to a mixture of the correct output and the Abort message. Equivalently, the probability that the Server makes the Client accept an incorrect outcome is bounded by ϵ_{ver}. Let d/n, t/n and w/t be the ratios of test, computation and tolerated failed test rounds. Our protocol's local-verifiability is given by Theorem 4.3.3 and proven subsequently.

Theorem 4.3.3 (Local Verifiability of Protocol 4)**.** *Let* $0 < w/t < \frac{1}{k} \cdot \frac{2p-1}{2p-2}$ *and* $0 < d/n < 1$ *be fixed ratios, for* k *different test rounds and where* p *is the inherent error probability of the BQP computation. Then, Protocol 4 is* ϵ_{ver}*-locally-verifiable for exponentially-low* ϵ_{ver}*.*

Proof. Proving verifiability of a computation amounts to upper-bounding the probability

of yielding a wrong output while not aborting. This could be the result of the inherent randomness of the BQP computation that gives the wrong outcome with probability p, or of the server deviating from the instructed computation. In the following, although rounds are expected to be run sequentially, the proof will examine the state of the *combined computation*. This state corresponds to the server having simulaneous unrestricted access to all quantum systems sent by the client and possibly operating on them as a whole irrespectively of the underlying rounds they belong to. In particular, the server could decide to perform some action on a qubit given measurements in one or several of the underlying runs, or to entangle the various underlying runs together.

Note that, because the parties can only ask for redoing a run independently of the input, of the computation, of used randomness and of the output of the computation itself (comprising the result of trap measurements), interrupted runs can be safely ignored in the verification analysis as the state corresponding to these runs is uncorrelated to that of the completed runs.

Output of the combined computation. First, consider the output density operator $B(\{\mathcal{F}_j\}_j, \nu)$ representing all the classical messages the Client A receives during its interaction with the Server B, comprising the final message containing the encrypted measurement outcomes. Below, the CPTP maps $\{\mathcal{F}_j\}_j$ represent the chosen deviation of B on the combined computation. By encoding the classical messages as quantum states in the computational basis, the output density operator satisfies:

$$B(\{\mathcal{F}_j\}_j, \nu) = \mathrm{Tr}_B \left\{ \sum_b |b + c_r\rangle\langle b| \, \mathcal{F}\mathcal{P} \times \left(|0\rangle\langle 0|_B \otimes \left|\Psi^{\nu,b}\right\rangle\left\langle\Psi^{\nu,b}\right| \right) \times \mathcal{P}^\dagger \mathcal{F}^\dagger \, |b\rangle\langle b + c_r| \right\} \tag{4.7}$$

where b is the list of measurement outcomes defining the computation branch; ν is a composite index relative to the secret parameters chosen by A, i.e. the type of each underlying run, the padding of the measurement angles and measurements outcomes and the trap setup; $|b + c_r\rangle\langle b|$ ensures that only the part corresponding to the current computation branch is taken into account and removes the One-Time-Pad encryption on non-output and non-trap qubits while leaving output and trap qubits unaffected, i.e. encrypted; $|0\rangle\langle 0|_B$ is some internal register for B in a fixed initial state; and $\left|\Psi^{\nu,b}\right\rangle$ is the state of the qubits sent by A to B at the beginning of the protocol tensored with quantum states representing the measurement angles of the computation branch b.

To obtain this result, the line of proof of [FK17] can be applied to the combined computation. This works by noting that for a given computation branch b and given random parameters ν, all the measurement angles are fully determined. Therefore, provided that the computation branch is b, the measurement angles can be included into the initial state. This defines $\left|\Psi^{\nu,b}\right\rangle$. Then, each $\mathcal{F}_j$ is decomposed into an honest part and a pure deviation. All the deviations are commuted and collected into $\mathcal{F}$ applied after $\mathcal{P}$, the unitary part of honest protocol, is applied. The projections onto $|b\rangle$ then ensures that, after the deviation induced by B, the perceived computation branch is b. This, together with the decrypting of non-output non-trap qubits, gives Equation 4.7.

Probability of failure. Recall that a failure for the combined computation on input x occurs when the result after decrypting the outputs and performing the majority vote differs from $F(x)$ while the computation is accepted.

For the combined computation to be accepted, no more than w test runs should have a trap qubit measurement outcome opposite to what was expected. Let $\mathtt{T}$ denote the set of trap qubits which is determined by T, the set of test runs, and the type of each test run. In absence of any deviation on the combined computation, their expected value is $|r_{\mathtt{T}}\rangle = \bigotimes_{\mathtt{t}\in\mathtt{T}} |r_{\mathtt{t}}\rangle$ where $r_{\mathtt{T}} = (r_{\mathtt{t}})_{\mathtt{t}\in\mathtt{T}}$ denotes the measurement outcome padding values restricted to trap qubits. Therefore, the projector onto the states of the trap qubits yielding to an accepted combined computation can be written as $Q_\perp = \sum_{\mathtt{w}\in\mathtt{W}} X^{\mathtt{w}}_{\mathtt{T}} |r_{\mathtt{T}}\rangle\langle r_{\mathtt{T}}| X^{\mathtt{w}}_{\mathtt{T}}$ with $X^{\mathtt{w}}_{\mathtt{T}} = \bigotimes_{\mathtt{t}\in\mathtt{T}} X^{\mathtt{w}_{\mathtt{t}}}_{\mathtt{t}}$, and where $\mathtt{W}$ is the set of length $|\mathtt{T}|$ binary vectors $\mathtt{w}$ that have at least a one in no more than w underlying (test) runs.

Similarly, define by $\mathtt{O}$ the set of output qubits. The correct value for these output qubit is $|F(x)_{\mathtt{O}} + r_{\mathtt{O}}\rangle$. Then, for $\mathtt{V}$ the set of length $|\mathtt{O}|$ binary vectors $\mathtt{v}$ that have at least $d/2$ ones in the underlying (computation) runs, the operator $P_\perp = \sum_{\mathtt{v}\in\mathtt{V}} X^{\mathtt{v}}_{\mathtt{O}} |F(x) + r_{\mathtt{O}}\rangle\langle F(x) + r_{\mathtt{O}}| X^{\mathtt{v}}_{\mathtt{O}}$ with $X^{\mathtt{v}}_{\mathtt{O}} = \bigotimes_{\mathtt{o}\in\mathtt{O}} X^{\mathtt{v}_{\mathtt{o}}}_{\mathtt{o}}$ is the projector onto the subspace of states that yield an incorrect result for the whole computation. This is because when each output has been decrypted by the Client – the one-time-padding $r_{\mathtt{O}}$ is removed – the majority vote will output $F(x)+1$ because more than half of the outputs are equal to $F(x)+1$.

Combining these two projectors allows to write the probability of failure:

$$\Pr[\text{fail}] = \sum_{\nu} \sum_{b,k,\sigma,\sigma'} \Pr[\nu] \operatorname{Tr}\Big\{ (P_\perp \otimes Q_\perp) \times \Big(\alpha_{k\sigma}\alpha^*_{k\sigma'} \left|b + c_r\right\rangle\!\left\langle b\right| \sigma \mathcal{P} \left|\Psi^{\nu,b}\right\rangle\!\left\langle\Psi^{\nu,b}\right| \mathcal{P}^\dagger \sigma' \left|b\right\rangle\!\left\langle b + c_r\right|\Big) \Big\} \tag{4.8}$$

where $\mathcal{F}$ has been decomposed into Kraus operators indexed by k, that were in turn decomposed onto the Pauli basis through the coefficients $\alpha_{k\sigma}$ and $\alpha_{k\sigma'}$. Consequently, σ and σ' are Pauli matrices.

Using the explicit expressions for $P_\perp$ and $Q_\perp$, the above formula can be simplified:

$$\Pr[\text{fail}] = \sum_{\nu} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} \sum_{b',k,\sigma,\sigma'} \Pr[\nu] \Big\{ \langle F(x)_{\mathtt{O}} + r_{\mathtt{O}}| \otimes \langle r_{\mathtt{T}}| \otimes \langle b'| \left(X^{\mathtt{v}}_{\mathtt{O}} \otimes X^{\mathtt{w}}_{\mathtt{T}}\right) \times \Big(\alpha_{k\sigma}\alpha^*_{k\sigma'} \mathcal{P} \left|\Psi^{\nu,b}\right\rangle\!\left\langle\Psi^{\nu,b}\right| \mathcal{P}^\dagger \sigma'\Big) \left(X^{\mathtt{v}}_{\mathtt{O}} \otimes X^{\mathtt{w}}_{\mathtt{T}}\right) |F(x)_{\mathtt{O}} + r_{\mathtt{O}}\rangle \otimes |r_{\mathtt{T}}\rangle \otimes |b'\rangle \Big\} \tag{4.9}$$

where b' is the binary vector obtained from b by restricting it to non-output and non-trap qubits. This was obtained using the circularity of the trace and the fact that $\sum_b \langle F(x)_{\mathtt{O}} + r_{\mathtt{O}}| \otimes \langle r_{\mathtt{T}}| \left(X^{\mathtt{v}}_{\mathtt{O}} \otimes X^{\mathtt{w}}_{\mathtt{T}}\right) \left|b + c_r\right\rangle\!\left\langle b\right| = \sum_{b'} \langle F(x)_{\mathtt{O}} + r_{\mathtt{O}}| \otimes \langle r_{\mathtt{T}}| \otimes \left|b' + c_r\right\rangle\!\left\langle b'\right| \left(X^{\mathtt{v}}_{\mathtt{O}} \otimes X^{\mathtt{w}}_{\mathtt{T}}\right)$ since there is no decoding for output and trap qubits – i.e. c_r is 0.

Using blindness of the scheme. At this point, standard proofs of verifiability sum over the secret parameters defining the encryption to twirl the deviation of the Server and trace out non-trap qubits. Here, because it is necessary to assess the probability of having more than half of the output qubits yielding the wrong measurement output $F(x) + 1$, the trace shall be taken on non-trap and non-output qubits only.

The design of the protocol yielding the combined computation ensures blindness. This implies that the resulting state of any set of qubits after applying $\mathcal{P}$ and taking the average over their possible random preparations parameters is a completely mixed state. This can be applied in the above equation for the set of non-output and non-trap qubits. For output and trap qubits, the inner products must be computed before taking the sum over their random preparation parameters $\nu_{\mathtt{O}}$ and $\nu_{\mathtt{T}}$ respectively.

This gives:

$$\Pr[\text{fail}] = \sum_{\nu_0,\nu_T,\mathtt{u}} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} \sum_{b',k,\sigma,\sigma'} \Pr[\nu_0,\nu_T]\alpha_{k\sigma}\alpha^*_{k\sigma'} \times \Bigg\{ \langle F(x)_0 + r_0| \otimes \\ \langle r_T| \otimes \langle b'| \left(X_0^{\mathtt{v}} \otimes X_T^{\mathtt{w}}\right) \times \sigma \left(|s_0 + r_0\rangle\langle s_0 + r_0| \otimes |r_T\rangle\langle r_T| \otimes \frac{\mathbb{I}}{\operatorname{Tr}\mathbb{I}} \right) \sigma' \times \\ \left(X_0^{\mathtt{v}} \otimes X_T^{\mathtt{w}}\right) |F(x)_0 + r_0\rangle \otimes |r_T\rangle \otimes |b'\rangle \Bigg\} \tag{4.10}$$

where $|s_\mathtt{o}\rangle$ is the state of the output qubit $\mathtt{o} \in \mathtt{O}$ when no deviation is applied by the Server.

In the above equation, the contribution of each qubit factorizes. For $l \notin \mathtt{O} \cup \mathtt{T}$, because the Pauli matrices are traceless save for the identity, the only non-vanishing terms are obtained for $\sigma_l = \sigma'_l$, where subscript l is used to select the action of σ and σ' on qubit l. In such case, the corresponding multiplicative factor equals 1. A direct calculation shows that, for an output qubit $\mathtt{o} \in \mathtt{O}$,

$$\sum_{r_\mathtt{o}} \langle F(x)_\mathtt{o} + r_\mathtt{o}| X_\mathtt{o}^{\mathtt{v}_\mathtt{o}} \sigma_\mathtt{o} |s_\mathtt{o} + r_\mathtt{o}\rangle \langle s_\mathtt{o} + r_\mathtt{o}| \sigma'_\mathtt{o} X_\mathtt{o}^{\mathtt{v}_\mathtt{o}} |F(x)_\mathtt{o} + r_\mathtt{o}\rangle = 0 \tag{4.11}$$

for $\sigma_\mathtt{o} \neq \sigma'_\mathtt{o}$ Similarly, for a trap qubit $\mathtt{t} \in \mathtt{T}$, $\sum_{r_\mathtt{t}} \langle r_\mathtt{t}| X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} \sigma_\mathtt{t} |r_\mathtt{t}\rangle\langle r_\mathtt{t}| \sigma'_\mathtt{t} X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} |r_\mathtt{t}\rangle$ vanishes for $\sigma_\mathtt{t} \neq \sigma'_\mathtt{t}$. Combining these yields:

$$\begin{aligned} \Pr[\text{fail}] &= \sum_{\nu_0,\nu_T} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} \sum_{k,\sigma} \Pr[\nu_0,\nu_T]|\alpha_{k\sigma}|^2 \times \\ &\qquad \prod_{\mathtt{o}\in\mathtt{O}} |\langle F(x)_\mathtt{o} + r_\mathtt{o}| X_\mathtt{o}^{\mathtt{v}_\mathtt{o}} \sigma_\mathtt{o} |s_\mathtt{o} + r_\mathtt{o}\rangle|^2 \times \prod_{\mathtt{t}\in\mathtt{T}} |\langle r_\mathtt{t}| X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} \sigma_\mathtt{t} |r_\mathtt{t}\rangle|^2 \\ &= \sum_k \sum_\sigma |\alpha_{k\sigma}|^2 f(\sigma) \end{aligned} \tag{4.12}$$

with

$$f(\sigma) = \sum_{\nu_0,\nu_T} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} \Pr[\nu_0,\nu_T] \times \prod_{\mathtt{o}\in\mathtt{O}} |\langle F(x)_\mathtt{o} + r_\mathtt{o}| X_\mathtt{o}^{v_\mathtt{o}} \sigma_\mathtt{o} |s_\mathtt{o} + r_\mathtt{o}\rangle|^2 \times \prod_{\mathtt{t}\in\mathtt{T}} |\langle r_\mathtt{t}| X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} \sigma_\mathtt{t} |r_\mathtt{t}\rangle|^2 \tag{4.13}$$

In short, this proves that the overall deviation $\mathcal{F}$ has the same effect as a convex combination of Pauli deviations σ each occuring with probability $\sum_k |\alpha_{k,\sigma}|^2$.

Implicit upper bound. Because, $\sum_{k,\sigma} |\alpha_{k\sigma}|^2 = 1$, the worst case scenario for the bound in Equation 4.13 is when $\alpha_{k\sigma} = 1$ for σ such that $f(\sigma)$ is maximum. Hence, the probability of failure is upper-bounded as follows:

$$\Pr[\text{fail}] \leq \max_{\sigma} f(\sigma). \tag{4.14}$$

Protocol 4 defines trap and output qubit configuration $\nu_{\mathtt{O}}, \nu_{\mathtt{T}}$ by (i) the set $\mathtt{T}$ of trap qubits, itself entirely determined by the position and kind of test runs within the sequence of runs, and (ii) the preparation parameters θ_l and r_l of each trap and output qubits. Each parameter of (i) and (ii) being chosen independently, the probability of a given configuration $\nu_{\mathtt{O}}, \nu_{\mathtt{T}}$ can be decomposed into the probability $\Pr[\mathtt{T}]$ for a given configuration of trap locations multiplied by the probability of a given configuration for the prepared state of the trap and output qubits, $\prod_{l \in \mathtt{O} \cup \mathtt{T}} \sum_{\theta_l, r_l} \Pr[\theta_l, r_l]$. Using this, one can rewrite $f(\sigma)$:

$$\begin{aligned} f(\sigma) = & \sum_{\mathtt{T}} \sum_{\mathtt{v} \in \mathtt{V}, \mathtt{w} \in \mathtt{W}} \Pr[\mathtt{T}] \times \\ & \prod_{\mathtt{o} \in \mathtt{O}} \sum_{\theta_\mathtt{o}, r_\mathtt{o}} \Pr[\theta_\mathtt{o}, r_\mathtt{o}] \left| \langle F(x)_\mathtt{o} + r_\mathtt{o} | X_\mathtt{o}^{\mathtt{v}_\mathtt{o}} \sigma_\mathtt{o} | s_\mathtt{o} + r_\mathtt{o} \rangle \right|^2 \times \\ & \prod_{\mathtt{t} \in \mathtt{T}} \sum_{\theta_\mathtt{t}, r_\mathtt{t}} \Pr[\theta_\mathtt{t}, r_\mathtt{t}] \left| \langle r_\mathtt{t} | X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} \sigma_\mathtt{t} | r_\mathtt{t} \rangle \right|^2 \end{aligned} \tag{4.15}$$

For σ a Pauli deviation, denote by $\sigma_{|X}$ the binary vector indexed by qubit positions of the combined computation where ones mark qubit positions for which σ acts as X or Y. Abusing notation, in the following, $\mathtt{O}$ will denote the binary vector over qubit positions i of the combined computation where ones are positioned for qubits in O – that is the vector $(\mathbb{1}_{i \in \mathtt{O}})_i$ for i a qubit location. Similarly, $\mathtt{T}$ will also denote $(\mathbb{1}_{i \in \mathtt{T}})_i$.

Using the fact that $\left| \langle r_\mathtt{t} | X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} \sigma_\mathtt{t} | r_\mathtt{t} \rangle \right|^2$ is 1 for $X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} \sigma_\mathtt{t} \in \{I, Z\}$ and 0 otherwise, the product over the trap qubits can be writen as:

$$\prod_{\mathtt{t} \in \mathtt{T}} \sum_{\theta_\mathtt{t}, r_\mathtt{t}} \Pr[\theta_\mathtt{t}, r_\mathtt{t}] \left| \langle r_\mathtt{t} | X_\mathtt{t}^{\mathtt{w}_\mathtt{t}} \sigma_\mathtt{t} | r_\mathtt{t} \rangle \right|^2 = \begin{cases} 1 & \text{for } \mathtt{T}.\sigma_{|X} = \mathtt{w} \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

where, for a and b binary vectors, $a.b$ is the bit-wise binary product vector.

For output qubits, before attempting the same computation, it is important to point out a important dependency of the deviation for the output qubits. Failing to take it into account would yield an overly optimistic bound. This dependency is due to the

fact that, contrarily to trap qubits where the perfect protocol performs the identity, the output qubits are the result of more complex computation. More precisely, the guarantee given by the protocol at this stage is only blindness. Following the definition of local blindness – Equation 4.3 – the Server is able to choose a deviation $\mathcal{E}$ and have it applied to the unprotected input of the protocol x, while himself not getting either x nor $\mathcal{E}(x)$. While $\mathcal{E}$ has been reduced here to a convex sum of Pauli deviations applied after the perfect protocol, nothing prevents these Pauli deviations to incorporate a dependency on the input x or on the unencrypted output of the perfect protocol. In short, this means that the Server could craft a deviation in such a way that only outputs equal to $F(x)$ are flipped, leaving those yielding $F(x)+1$ unaffected.

Going forward with the computation of factors for output qubits in Equation 4.15, it is thus necessary to distinguish output qubits that belong to computation rounds where no non-trivial deviation take place, and those that don't. Define $\mathtt{u}$ to be the random binary vector of length $|\mathtt{O}|$ such that $s_\mathtt{o} = F(x)+\mathtt{u}_\mathtt{o}$. For an output qubit that is part of a computation round without a non-trivial deviation,

$$\begin{aligned}
&\sum_{\theta_\mathtt{o},r_\mathtt{o}} \Pr[\theta_\mathtt{o},r_\mathtt{o}] \left| \langle F(x)_\mathtt{o}+r_\mathtt{o}| X_\mathtt{o}^{\mathtt{v}_\mathtt{o}}\sigma_\mathtt{o} |s_\mathtt{o}+r_\mathtt{o}\rangle \right|^2 \\
&\quad = \sum_{\theta_\mathtt{o},r_\mathtt{o}} \Pr[\theta_\mathtt{o},r_\mathtt{o},\mathtt{u}_\mathtt{o}] \times \left| \langle F(x)_\mathtt{o}+r_\mathtt{o}| X_\mathtt{o}^{\mathtt{v}_\mathtt{o}}\sigma_\mathtt{o}X_\mathtt{o}^{\mathtt{u}_\mathtt{o}} |F(x)+r_\mathtt{o}\rangle \right|^2 \\
&\quad = \begin{cases} \Pr[\mathtt{u}_\mathtt{o}] & \text{for } \sigma_{|X,\mathtt{o}}+\mathtt{u}_\mathtt{o}=\mathtt{v}_\mathtt{o} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{4.17}$$

When the output qubit is part of a computation round with a non-trivial deviation, the dependency argument given above yields:

$$\sum_{\theta_\mathtt{o},r_\mathtt{o}} \Pr[\theta_\mathtt{o},r_\mathtt{o}] \times \left| \langle F(x)_\mathtt{o}+r_\mathtt{o}| X_\mathtt{o}^{\mathtt{v}_\mathtt{o}}\sigma_\mathtt{o}X_\mathtt{o}^{\mathtt{u}_\mathtt{o}} |F(x)+r_\mathtt{o}\rangle \right|^2 \leq \Pr[\mathtt{u}_\mathtt{o}]. \tag{4.18}$$

Hence, for a fixed σ, a necessary condition on $\mathtt{u}$ and $\mathtt{T}$ for having a non zero contribution to $f(\sigma)$ is thus:

$$wt(\mathtt{T}.\sigma_{|X}) \leq w \text{ and } wt(\mathtt{u}.\neg\mathtt{S}) \geq d/2 - |\mathtt{S}|$$

where $wt(.)$ is the Hamming weight of a binary vector, $\mathtt{S}$ is a length $|\mathtt{O}|$ binary vector where the ones are located on output qubits where at least one non-trivial deviation was performed in the corresponding computation round, and $\neg\mathtt{S}$ is the bitwise negation of $\mathtt{S}$.

Combining the corresponding bounds and summarizing the necessary condition with $(\mathtt{T},\mathtt{u}) \in \Upsilon_\sigma$, one obtains:

$$f(\sigma) \leq \sum_{(\mathtt{T},\mathtt{u})\in\Upsilon_\sigma} \Pr[\mathtt{T},\mathtt{u}]. \tag{4.19}$$

Otherwise said, to record a failure of the protocol, the number of incorrect trap rounds need to be below the threshold w, while the number of non-trivially attacked computation rounds need to be greater than $d/2$ reduced by the amount of incorrect outcomes on non-attacked rounds due to the inherent randomness of the algorithm.

Explicit upper bound. Now, assume that the maximum of the bound above is attained for some σ that happens to non-trivially affect one of the round, say k, on more than one qubit. Consider σ' with the sole difference to σ that σ' restricted to one of these two qubits is equal to the identity. Then, σ' still affects the round k non-trivially, which implies that all configurations $(\mathtt{T},\mathtt{u})$ in Υ_σ are also in $\Upsilon_{\sigma'}$. Therefore

$$\Pr[\text{fail}] \leq \max_m \max_{\sigma\in E_m} \sum_{(\mathtt{T},\mathtt{u})\in\Upsilon_\sigma} \Pr[\mathtt{T},\mathtt{u}]. \tag{4.20}$$

where E_m denotes the set of Pauli operators with m single qubit non-trivial deviations all in distinct rounds.

Because the bound above depends on $\mathtt{u}$ only through $wt(\mathtt{u}.\neg\mathtt{S})$ and because for any such subset the random variable $wt(\mathtt{u}.\neg\mathtt{S})$ is less than $B(wt(\neg\mathtt{S}),p)$ in the usual stochastic order, one obtains:

$$\Pr[\text{fail}] \leq \max_m \max_{\sigma\in E_m} \sum_{(\mathtt{T},\mathtt{u})\in\Upsilon_\sigma} \Pr[\mathtt{T}] \times \Pr[\tilde{\mathtt{u}} = \mathtt{u}], \tag{4.21}$$

where $\tilde{\mathtt{u}}$ is a random binary vector where each coordinate follows a Bernouilli law with probability p, and where $B(n,p)$ is the binomial distribution for n draws and probability p. Using the fact that the random choice of test runs is completely uniform, the right hand side is invariant under permutations of the test and computation runs. It is thus possible to restrict the range of the maximum to the specific Pauli operators σ_m with a deviation on a single qubit in each of the first m runs:

$$\Pr[\text{fail}] \leq \max_m \sum_{T\in\Upsilon_{\sigma_m}} \Pr[\mathtt{T}]. \tag{4.22}$$

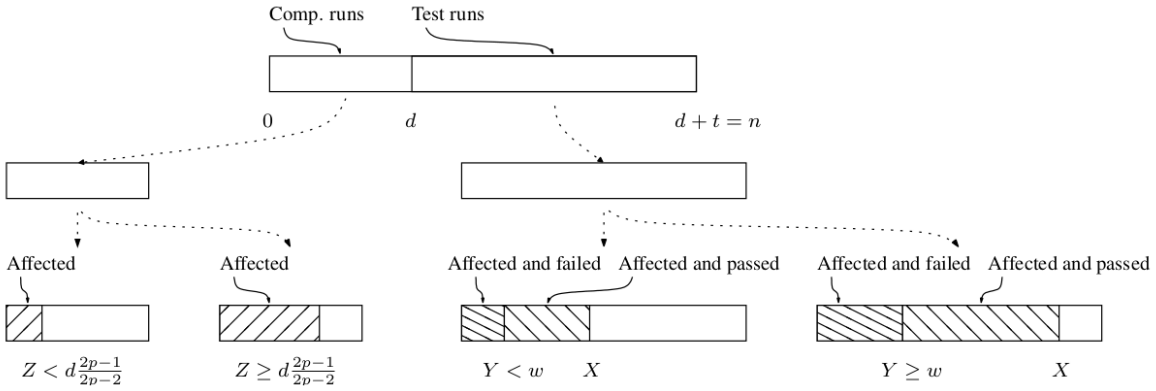


Figure 4.2: The four cases needed to determine a closed form upper bound for the probabiliity of failure. First, we determine the probability for the number of affected computation rounds. If it is low enough ($Z < d(2p-1)/(2p-2)$), no need to abort. If it is high ($Z \geq d(2p-1)/(2p-2)$), we find a bound on the probability that the number of failed test rounds Y is below or above w.

A closed from for the upper bound. To find a closed form upper bound for the soundness error, we now distinguish between two regimes for m, controlled by the parameter $\varphi > 0$:

1. For $m \leq \left(\frac{2p-1}{2p-2} - \varphi\right) n$, we find a small upper bound on the probability that the client obtains a wrong result,

2. whereas for $m \geq \left(\frac{2p-1}{2p-2} - \varphi\right) n$, we find a small upper bound on the probability that the client accepts the outcome of the protocol, i.e. that the verification passes.

In the following, we define the constant ratios of test, computation and tolerated failed test runs as $\delta := d/n$, $\tau := t/n$ and $\omega := w/t$. Let Z be a random variable counting the number affected computation runs (by the server's deviation or by inherent failure of the algorithm) and Y a random variable counting the number of failed test runs, i.e. the number of affected test runs where the deviation hits a trap. We have that:

$$\begin{aligned}\Pr[\text{fail}] &\leq \max_m \sum_{T \in \Upsilon_{\sigma_m}} \Pr[T] = \max_m \Pr\left[Z \geq \frac{d}{2} \wedge Y \leq w\right] \\ &\leq \max\left\{ \max_{m \leq \left(\frac{2p-1}{2p-2} - \varphi\right) n} \Pr\left[Z \geq \frac{d}{2}\right], \max_{m \geq \left(\frac{2p-1}{2p-2} - \varphi\right) n} \Pr[Y \leq w] \right\}. \end{aligned} \tag{4.23}$$

Since $\Pr[Z \geq d/2]$ and $\Pr[Y \leq w]$ are respectively increasing and decreasing with the

number of attacked runs, both inner maximums are attained for $m = \left(\frac{2p-1}{2p-2} - \varphi\right) n$ and we therefore focus on this case.

Analogously to the verification proof of the original protocol, the second term can be bounded from above by first determining the minimum number of affected test runs before calculating the probability that the server's attack triggers a sufficient number of traps.

Hence, with X denoting the number of test runs affected by the server's deviation, tail bounds for the hypergeometric distribution imply for all $\varepsilon_1 > 0$ that

$$\Pr\left[X \leq \left(\frac{m}{n} - \varepsilon_1\right) t\right] \leq \exp\left(-\frac{2\tau^2\epsilon_1^2}{\frac{2p-1}{2p-2} - \varphi} n\right). \tag{4.24}$$

Further, it follows by Hoeffding's bound for the binomial distribution that

$$\begin{aligned}\Pr\left[Y \leq \left(\frac{1}{k} - \varepsilon_2\right)\left(\frac{m}{n} - \epsilon_1\right) t \;\middle|\; X = \left(\frac{m}{n} - \varepsilon_1\right) t\right] \\ \leq \exp\left(-2\left(\frac{2p-1}{2p-2} - \varphi - \varepsilon_1\right)\tau\varepsilon_2^2 n\right).\end{aligned} \tag{4.25}$$

All in all, we therefore obtain

$$\Pr\left[Y \leq w\right] \leq \exp\left(-\frac{2\tau^2\epsilon_1^2}{\frac{2p-1}{2p-2} - \varphi} n\right) + \exp\left(-2\left(\frac{2p-1}{2p-2} - \varphi - \varepsilon_1\right)\tau\varepsilon_2^2 n\right), \tag{4.26}$$

where the threshold of tolerated failed test runs is set to $w = (1/k - \epsilon_2)\left(\frac{2p-1}{2p-2} - \varphi - \varepsilon_1\right) t$.

Let's now focus on the first term and introduce the hypergeometrically distributed random variable $\bar{Z}$ counting the number of computation runs that are affected by the server's deviation. Then, for $\varepsilon_3 > 0$ tail bounds on the hypergeometric distribution imply

$$\Pr\left[\bar{Z} \geq \left(\frac{m}{n} + \varepsilon_3\right) d\right] \leq \exp\left(-\frac{2\delta^2\varepsilon_3^2}{\frac{2p-1}{2p-2} - \varphi} n\right). \tag{4.27}$$

Next, let Z' be the random variable counting the number of computation runs that have not been affected by the server's deviation but which give a from $\bar{x}$ distinct result because of inherent failures of the algorithm. Note, that Z' conditioned on $\bar{Z}$ fixed to a

specific value is binomially distributed. It hence follows that

$$\begin{aligned}\Pr\left[Z' \geq (p+\varepsilon_4)\left(1-\frac{m}{n}-\varepsilon_3\right)d \;\middle|\; \bar{Z}=\left(\frac{m}{n}+\varepsilon_3\right)d\right] \\ \leq \exp\left(-2\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)\delta\varepsilon_4^2 n\right).\end{aligned} \tag{4.28}$$

Note that it holds that $Z = \bar{Z} + Z'$. Therefore, it follows that

$$\begin{aligned}\Pr\left[Z \geq \frac{d}{2}\right] &\leq \Pr\left[Z \geq \frac{d}{2} \;\middle|\; \bar{Z} \leq \left(\frac{m}{n}+\varepsilon_3\right)d\right] \\ &\quad + \Pr\left[\bar{Z} \geq \left(\frac{m}{n}+\varepsilon_3\right)d\right] \\ &\leq \Pr\left[Z' \geq \frac{d}{2} - \left(\frac{m}{n}+\varepsilon_3\right)d \;\middle|\; \bar{Z}=\left(\frac{m}{n}+\varepsilon_3\right)d\right] \\ &\quad + \Pr\left[\bar{Z} \geq \left(\frac{m}{n}+\varepsilon_3\right)d\right].\end{aligned} \tag{4.29}$$

Using the inequalities from above, we arrive at

$$\begin{aligned}\Pr\left[Z \geq \frac{d}{2}\right] \leq &\exp\left(-2\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)\delta\varepsilon_4^2 n\right) \\ &+ \exp\left(-\frac{2\delta^2\varepsilon_3^2}{\frac{2p-1}{2p-2}-\varphi}n\right)\end{aligned} \tag{4.30}$$

where we set

$$\frac{d}{2} - \left(\frac{m}{n}+\varepsilon_3\right)d = (p+\varepsilon_4)\left(1-\frac{m}{n}-\varepsilon_3\right)d. \tag{4.31}$$

This condition can be rewritten as

$$\frac{1}{2} - \frac{2p-1}{2p-2} + \varphi - \varepsilon_3 = (p+\varepsilon_4)\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right), \tag{4.32}$$

or equivalently

$$\varepsilon_4 = \left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)^{-1} \cdot \left(\frac{1}{2}-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right) - p. \tag{4.33}$$

It can be readily seen that this equation has solutions $\varepsilon_3, \varepsilon_4 > 0$ when φ is fixed.

We finally conclude that

$$\Pr[\text{fail}] \leq \max\left\{\exp\left(-2\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)\delta\varepsilon_4^2 n\right)+\exp\left(-\frac{2\delta^2\varepsilon_3^2}{\frac{2p-1}{2p-2}-\varphi}n\right),\right.$$
$$\left.\exp\left(-\frac{2\tau^2\epsilon_1^2}{\frac{2p-1}{2p-2}-\varphi}n\right)+\exp\left(-2\left(\frac{2p-1}{2p-2}-\varphi-\varepsilon_1\right)\tau\varepsilon_2^2 n\right)\right\} \quad (4.34)$$

for

$$\begin{aligned}
&w=(1/k-\epsilon_2)\left(\frac{2p-1}{2p-2}-\varphi-\varepsilon_1\right)t,\\
&0<\varphi<\frac{2p-1}{2p-2},\\
&0<\varepsilon_1<\frac{1}{2}-\varphi,\\
&0<\varepsilon_2<\frac{1}{k},\\
&0<\varepsilon_3<\varphi,\\
&\varepsilon_4=\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)^{-1}\cdot\left(\frac{1}{2}-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)-p. \qquad (4.35)
\end{aligned}$$

To obtain an optimal bound, this expression must be minimized over ε_1, ε_2, ε_3 and φ.

Irrespective of the exact form of the optimal bound, choosing φ, ε_1, ε_2, and ε_3 sufficiently small implies the existence of protocols with verification exponential in n, for any fixed $0 < w/t < \frac{1}{k}\cdot\frac{2p-1}{2p-2}$ and fixed $\frac{d}{n}, \frac{t}{n} \in (0,1)$.

Optimality of the bound. To obtain the improved bound above, Z_2 was introduced as the count of non-affected computation runs yielding the correct result – i.e. accept on yes instances, and reject on no instances. Making sure that Z_2 would be greater than $d/2$ ensures that no matter what happens on computation runs that would yield an incorrect result, there is no possibility of being mistaken and reject in place of accept, and vice versa. Yet, one might wonder if the situation is not more favorable: if the deviation by the server induces a flip of the accept / reject then could it be possible that some of the runs yielding incorrect result would be corrected by the deviation. At first sight, this could be motivated by the fact that the computation being blind, the server could not possibly craft an attack that would selectively affect the runs yielding the correct results. Unfortunately, this intuition is wrong: blindness does not rule out

attacks that have different effects depending on the result of the computation itself.

To see this, consider the following situation. Consider an algorithm solving a decision problem deterministically, so that in case of a yes instance, the algorithm outputs $|+\rangle$, and, in case of a no instance the output is $|-\rangle$. This deterministic algorithm yields a trivial randomized algorithm where a second qubit is generated in state $\alpha|0\rangle + \beta|1\rangle$, with $|\alpha|^2 > 2/3$. The new algorithm would take the output of the first one and apply a control-Z gate between both qubits so that when the second qubit is traced out, the first one yields the correct answer with probability $|\alpha|^2$. Yet, nothing could rule out an alternate implementation where after the control-Z gate, the state of the first qubit undergoes two H gates controlled by the second qubit being $|0\rangle$. Clearly this operation applies the identity to the first qubit as $H^2 = I$. However, if the server applies a X gate on the first qubit between these two control-H gates, it will amount to a deviation consisting of a Z gate applied only when the second qubit is $|0\rangle$. As a result, its attack only affects runs with the correct result. Note that the attack affects correct outcomes only because in between the two control-H gates, the computational branch for correct outcomes yields a state in the computational basis, while for incorrect ones it is the $|\pm\rangle$ basis. This property is true independently of the quantum one-time-pad encryption of the states and can hence be applied on an encrypted computation.

This example might seem excessively artificial, but such situations cannot be ruled out a priori, i.e. without an extensive understanding of the algorithm being implemented and of the proposed implementation. In fact, a similar situation [Kap+21] has already been encountered in the context of multi-party quantum computation where attacks could be crafted to evade detection when using less obvious inappropriate implementations. □

Proof of Exponential Composable-Security. Our protocol has perfect correctness (for noiseless devices), blindness and input-independent verification. In addition, it is ϵ_{ver}-locally-verifiable with ϵ_{ver} exponentially small in n. Therefore, by the Local-Reduction Theorem, it is ϵ-composably-secure with $\epsilon = \epsilon_{sec} = 4\sqrt{2\epsilon_{ver}}$ and ϵ exponentially small in n. Note that because we used the Local-Reduction Theorem to obtain fully composable security, we incurred an additional square root on our verifiability bound given by Equation 4.1 and needed to satisfy the additional independence property. This is of course not required if the protocol is only used sequentially with other schemes, which will probably be the case in early quantum computations since the machines will not be able to handle multiple protocols at the same time. In this case, the stand-alone model would be sufficient since it provides sequential composition, but would fail if parallel

composition is needed.

4.3.4 Noise Robustness

Recall that the constant ratios of test, computation and tolerated failed test rounds are given by $\delta = d/n$, $\tau = t/n$ and $\omega = w/t$. We define the acceptance of the protocol to be the probability that the Client does not abort at the end of an execution. We then bound this probability in two regimes: (i) if the maximal noise p_{max} is smaller the (ratio) threshold of failed test runs, the protocol accepts with high probability; (ii) if the noise of the device is too large, i.e. p_{min} is already too large compared to the threshold, the protoco will most certainly abort.

Lemma 4.3.4 (Acceptance on Noisy Devices)**.** *Assume a Markovian round-dependent model for the noise on the Client and Server devices and let $p_{min} \leq p_{max} < 1/2$ be respectively a lower and an upper-bound on the probability that at least one of the trap measurement outcomes in a single test round is incorrect.*

If $\omega > p_{max}$, then the probability that the Client does not accept at the end of Protocol 4 is bounded by exponentially small ϵ_{rej} where

$$\epsilon_{rej} = \exp\left(-2(\omega - p_{max})^2 \tau n\right). \tag{4.36}$$

On the other hand, if $\omega < p_{min}$, then the Client's acceptance in Protocol 4 is exponentially small and bounded by $\exp\left(-2(p_{min} - \omega)^2 \tau n\right)$.

Proof. We define the random variables Y that corresponds to the number of failed test rounds during one execution of the protocol. We call Ok the event that the Client accepts at the end of the protocol – if not too many test rounds fail, meaning that $Y < w$.

For $\omega > p_{max}$. Equivalently, we have that $w > tp_{max}$. We are looking to lower-bound the probability that an honest round does not abort:

$$\Pr\left[\mathsf{Ok}\right] = \Pr\left[Y < w\right]. \tag{4.37}$$

Note that Y describes exactly the number of test rounds in which at least one trap measurement outcome is incorrect (by definition of a failed test round). The probability that a given test round fails is therefore upper-bounded by p_{max}. Let $\hat{Y}_1$ be a random

variable following a (t, p_{max})-binomial distribution. Since we suppose that the noise is not correlated across rounds, Y is upper-bounded by $\hat{Y}_1$ in the usual stochastic order:

$$\Pr[Y < w] \geq \Pr\left[\hat{Y}_1 < w\right] = 1 - \Pr\left[\hat{Y}_1 \geq w\right]. \tag{4.38}$$

Further, since $\mathbb{E}\left[\hat{Y}_1\right] = tp_{max} < w$, applying Lemma 4.5.6 yields:

$$\Pr\left[\hat{Y}_1 \geq w\right] \leq \exp\left(-2\frac{(tp_{max} - w)^2}{t}\right) = \exp\left(-2(\omega - p_{max})^2\tau n\right) = \epsilon_{rej}. \tag{4.39}$$

For $\omega < p_{min}$. In that case, we have that $w < tp_{min}$. We show that the probability of accepting is upper-bounded by a negligible function. Let $\hat{Y}_2$ be a random variable following a (t, p_{min})-binomial distribution, Y then is lower-bounded by $\hat{Y}_2$ in the usual stochastic order:

$$\Pr[Y < w] \leq \Pr\left[\hat{Y}_2 < w\right]. \tag{4.40}$$

Since $w < tp_{min}$, using Lemma 4.5.6 directly and with the same simplifications as above, we get:

$$\Pr\left[\hat{Y}_2 < w\right] \leq \exp\left(-2(p_{min} - \omega)^2\tau n\right), \tag{4.41}$$

concluding the proof. □

Theorem 4.3.5 (Local-Correctness of VDQC Protocol on Noisy Devices)**.** *Assume a Markovian round-dependent model for the noise on Client and Server devices and let p_{max} be an upper-bound on the probability that at least one of the trap measurement outcomes in a single test round is incorrect.*

If $p_{max} < \omega < \frac{1}{k} \cdot \frac{2p-1}{2p-2}$, then the protocol is ϵ_{cor}-locally-correct with exponentially small $\epsilon_{cor} = \epsilon_{rej} + \epsilon_{ver}$, with ϵ_{rej} from Lemma 4.3.4 and ϵ_{ver} from Theorem 4.3.3.

Proof. We call Ok the event that the Client accepts at the end of the protocol – if not too many test rounds fail – and $\mathsf{Correct}$ the event corresponding to a correct output – if only few of the computation rounds have their output bits flipped.

We are looking to lower-bound the probability of an honest round producing the correct outcome and not aborting:

$$\Pr[\mathsf{Correct} \wedge \mathsf{Ok}] = \Pr[\mathsf{Ok}] - \Pr[\neg\mathsf{Correct} \wedge \mathsf{Ok}]. \tag{4.42}$$

As $p_{max} < \frac{1}{k} \cdot \frac{2p-1}{2p-2} < 1/2$, from Lemma 4.3.4 we have

$$\Pr[\mathsf{Ok}] \geq 1 - \epsilon_{rej}. \tag{4.43}$$

Since $\omega < \frac{1}{k} \cdot \frac{2p-1}{2p-2}$, the parameters of Protocol 4 comply with Theorem 4.3.3, from which we get that

$$\Pr[\neg\mathsf{Correct} \wedge \mathsf{Ok}] \leq \epsilon_{ver}. \tag{4.44}$$

It follows that

$$\Pr[\mathsf{Correct} \wedge \mathsf{Ok}] \geq 1 - \epsilon_{rej} - \epsilon_{ver}, \tag{4.45}$$

which concludes the proof. □

4.4 Discussion

Role of Noise Assumptions in Correctness Analysis. Our security proof does not rely on any assumption regarding the form or amplitude of the noise: it considers any deviation as potentially malicious and shows that the protocol provides information-theoretic verification and blindness. The assumptions on the noise – limited strength and markovianity – are used only to show that correctness holds not only in the honest and noiseless case, but also when the imperfections of the devices are mild. In such cases, their impact on the computation can be mitigated and the protocol will accept with high probability.

Fine-Tuning the Number of Repetitions. For specific computations with fixed security and correctness targets as well as noise levels, several parameters can be tuned to optimise the total runtime of our protocol. First, distributing rounds across different machines is an effective way to reduce the overall execution time while composability ensures that security is preserved. Second, for a fixed graph, a smaller value of k allows a larger value of p_{max}, since exponential verification and correctness require $p_{max} < w/t < \frac{1}{k} \cdot \frac{2p-1}{2p-2}$: finding a small k-colouring of the graph used for the computation widens the gap between the chosen threshold ratio w/t and $\frac{1}{k} \cdot \frac{2p-1}{2p-2}$, thereby reducing

the number of rounds required to get the desired security and correctness levels.[2] Third, the ratio d/t also influences the number of repetitions. Given fixed values for p, k, w/t, security and correctness levels, the optimal ratio can be determined numerically using equations 4.34 and 4.36, which explicitly relate the failure and success probabilities to these parameters.

Decoupling Verifiability and Fault-Tolerance. Because a single trap has bounded sensitivity – the probability α of not detecting an attack at a given vertex is bounded away from 0 – it must be boosted to get exponential security. Previous work resorted to fault-tolerant encoding of the computation path to ensure that r errors can be corrected (see [FK17; KW17b]). This forces attackers to corrupt at least r locations to affect the computation, which decreases the probability of not detecting such attacks to α^r. Increasing the security of these protocols simultaneously increases the minimum distance of the fault-tolerant amplification scheme, thereby reducing the number of available qubits to perform the computation.

Our protocol's repetition of test rounds and majority vote serve the same purpose but with a much lighter impact. Because our detection probability amplification relies on a classical procedure, all qubits can be devoted to useful computations irrespective of the desired security level.

Additionally, our protocol does not abort at the first failed trap while previous approaches do. This means that, in the presence of noise, other protocols always require an exponentially low global residual error level to accept with overwhelming probability. On the contrary, our protocol only needs the average ratio of failed test rounds to be upper-bounded away from $\frac{1}{k} \cdot \frac{2p-1}{2p-2}$, which requires to bring the global residual error level to a constant only. This promises to drastically ease experimental feasibility of verified quantum computations.

4.5 Appendix: Useful Inequalities from Probability Theory

The following definitions and lemmata are useful tools for our proof. We refer the reader to [Fel91] for more in-depth definitions.

[2] This can be done once by the Server for its architecture and later shared with the Client before starting the protocol as a service.

Definition 4.5.1 (Hypergeometric distribution)**.** *Let $N, K, n \in \mathbb{N}$ with $0 \leq n, K \leq N$. A random variable X is said to follow the* hypergeometric distribution*, denoted as $X \sim \text{Hypergeometric}(N, K, n)$, if its probability mass function is described by*

$$\Pr[X = k] = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}. \tag{4.46}$$

As one possible interpretation, X describes the number of drawn marked items when drawing n items from a set of size N containing K marked items, without replacement*.*

Lemma 4.5.2 (Tail bound for the hypergeometric distribution)**.** *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $0 < t < K/N$. It then holds that*

$$\Pr\left[X \leq \left(\frac{K}{N} - t\right) n\right] \leq \exp\left(-2t^2 n\right). \tag{4.47}$$

Corollary 4.5.3. *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $0 < \lambda < \frac{nK}{N}$. It then holds that*

$$\Pr[X \leq \lambda] \leq \exp\left(-2n\left(\frac{K}{N} - \frac{\lambda}{n}\right)^2\right). \tag{4.48}$$

Lemma 4.5.4 (Serfling's bound for the hypergeometric distribution [GW17; Ser74])**.** *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $\lambda > 0$. It then holds that*

$$\Pr\left[\sqrt{n}\left(\frac{X}{n} - \frac{N}{K}\right) \geq \lambda\right] \leq \exp\left(-\frac{2\lambda^2}{1 - \frac{n-1}{N}}\right). \tag{4.49}$$

Corollary 4.5.5. *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $\lambda > \frac{nK}{N}$. It then holds that*

$$\Pr[X \geq \lambda] \leq \exp\left(-2n\left(\frac{\lambda}{n} - \frac{K}{N}\right)^2\right). \tag{4.50}$$

Note the symmetry of Corollary 4.5.3 and Corollary 4.5.5.

Lemma 4.5.6 (Hoeffding's inequality for the binomial distribution)**.** *Let $X \sim \text{Binomial}(n, p)$*

be a random variable. For any $k \leq np$ it then holds that

$$\Pr\left[X \leq k\right] \leq \exp\left(-2\frac{(np-k)^2}{n}\right). \tag{4.51}$$

Similarly, for any $k \geq np$ it holds that

$$\Pr\left[X \geq k\right] \leq \exp\left(-2\frac{(np-k)^2}{n}\right). \tag{4.52}$$

Chapter 5

Unifying Quantum Verification and Error-Detection

With the recent availability of cloud quantum computing services, the question of verifying quantum computations delegated by a client to a quantum server is becoming of practical interest. While Verifiable Blind Quantum Computing (VBQC) has emerged as one of the key approaches to address this challenge, current protocols still need to be optimised before they are truly practical. To this end, we establish a fundamental correspondence between error-detection and verification and provide sufficient conditions to both achieve security in the Abstract Cryptography framework and optimise resource overheads of all known VBQC-based protocols. As a direct application, we demonstrate how to systematise the search for new efficient and robust verification protocols for BQP *computations. While we have chosen Measurement-Based Quantum Computing (MBQC) as the working model for the presentation of our results, one could expand the domain of applicability of our framework via direct known translation between the circuit model and MBQC.*

This chapter is based on the paper "Unifying Quantum Verification and Error-Detection: Theory and Tools for Optimisations" [Kap+22] which is joint work with Theodoros Kapourniotis, Elham Kashefi, Luka Music, and Harold Ollivier.

5.1 Introduction

5.1.1 Context

Secure delegation of quantum computation is a long-standing topic of research where a client wants to perform a computation on a remote server, without necessarily trusting it. In this context, a computation is deemed blind when the privacy of the data and algorithm is guaranteed, and verified whenever the integrity of the computation is guaranteed or else the computation has aborted. None of these criteria are specific to quantum computing as users have always needed to protect their data, their algorithmic know-how and ensure that no party can manipulate results beyond their ability to choose their inputs [Gen09; Gen17]. Initially, the main interest for verifying quantum computations was relative to the nature of the client (or verifier) [Aar07; Vaz07]: what quantum power is needed by the client to verify a possibly unbounded quantum server (or prover)? Yet, this topic has gained attention due to the recent development of remotely accessible quantum computers, where no cryptographic guarantee is currently provided to clients delegating their computations to service providers. This, in turn, transformed a mostly theoretical question into a more practical one.

The first line of work to tackle this question introduced protocols guaranteeing statistical security by requiring the client to perform single qubit operations – either preparations or measurements. More recent protocols provide only computational security, with the benefit of being applicable to fully classical clients. In addition to the theoretical implications raised by verification, the possible practicality of proposed protocols has always been an important aspect of research on this topic as it was anticipated that quantum computers would be mostly available remotely. Recent years have confirmed this direction. Existing end-users of quantum computing services often emphasise the importance of integrity guarantees for the computations they delegate, as well as privacy of their data and algorithms.

Several protocols have been introduced along the years with the purpose of lowering some of the resource overhead of secure delegated computations. Yet, there is a lack of theoretical understanding of the requirements to construct robust and efficient verification protocols, as well as a lack of tools to systematise their optimisation. More precisely, while there are protocols that optimise the qubit communication, the complexity of the operations on the client's side, the overhead on the server's side, or the amount of tolerable noise, none provide general methods that could be applied when designing

the protocols themselves and used to tailor their performance to specific contexts and use-cases.

In this work, we deconstruct composable and statistically secure protocols for delegated quantum computations, to both exhibit their fundamental structure and allow for their convenient optimisation. We focus on protocols framed in the Measurement Based Quantum Computation (MBQC) model [RB01]. Our results are based on the simple yet powerful ideas that *detecting deviations from the client's instructions which are potentially harmful for the computation should yield verification*, while *the ability to be insensitive to those that are not harmful should provide noise-robustness*. We formalise this intuition through the concept of trappified schemes – a set of computations containing factitious computations whose results are known only to the client –, together with a necessary condition for obtaining negligible security errors with polynomial resources. Even more importantly, this work naturally connects the field of error-detection to that of verification, opening considerably the sources of inspiration for designing new trappified schemes and thus verification protocols.

As a concrete application, we construct a generic compiler for verifying BQP computations without any overhead of physical resources compared to the unprotected computation. Its efficiency is then optimised thanks to the introduction of new traps inspired by syndrome measurements of error-correcting codes.

Related Work. The first verification protocols have relied on the client's ability to access a small constant size quantum machine. It serves to encrypt the instructions delegated to the server or to perform the necessary operations to complete the computation once a complex resource state is provided by the server [ABE10; Aha+17; FK17; Bro18; HM15]. In both cases, the behaviour of the server is checked thanks to insertion of smaller computations alongside the one of interest whose result is known to the client.

More recent protocols used the mapping of BQP computations onto the 2-local Hamiltonian problem. In [FHM18; Han+17], the necessity of encryption was removed while the client was still required to perform X and Z measurements. In the ground breaking work of [Mah18b], the client was made entirely classical at the expense of some post-quantum secure computational assumptions.

Unfortunately, all these protocols – even those with a classical client – are too resource-intensive on the server's side to be practical. Several efforts have been devoted to improve the situation, in particular for protocols using Universal Blind Quantum Computing to encrypt the instructions sent to the server. [KW17b; XTH20] seek to

reduce the connectivity of the graph supporting the computation; [KDK15] reduces the communication instead; and the objective of [FKD18] is to limit further the set of operations that the client must wield to be able to perform the protocol. Recently, [Lei+21] considers the joint optimisation of the space overhead as well as the level of honest noise that the protocol is able to withstand while still accepting.

5.1.2 Overview of results

In this chapter, we express our results in the prepare-and-send model trading generality for simplicity, whereas we rely on the equivalence with the receive-and-measure model to extend their applicability [WEP22]. In this model, the client is prepares a small subset of quantum states, performs limited single-qubit operations and sends its prepared states to a server via a quantum communication channel. The server then executes the client's instructions and possibly returns some quantum output via the same quantum channel. As we seek not only verification but also blindness, we will use extensively the simple obfuscation technique put forth in the Universal Blind Quantum Computation (UBQC) protocol (see Section 2.3 for basics about UBQC) and consisting in randomly rotating each individual qubit sent by the client to the server.

The main idea that has been put at work in previous verification protocols is that, in such case, the client can chose to insert some factitious computations alongside the one it really intends to delegate. Because the client can choose factitious computations whose results are easy to compute classically and therefore to test, and because the server does not know whether the computation is genuine or factitious, these allows to ensure that the server is non-malicious.

Analyzing Deviations with Traps (Section 5.2). Here, we lay out a series of concepts that formally define theses factitious computations, or traps, as probabilistic error-detecting schemes. More precisely, we define *trappified canvases* as subcomputations on an MBQC graph with a fixed input state and classical outputs which follow a probability distribution that is efficiently computable classically. This is paired to a decision function which, depending on the output of this subcomputation, returns whether the trap accepts or rejects. The term canvas refers to the fact that there is still empty space on the graph alongside the factitious computation for the client's computation to be "painted into". This task is left to an *embedding algorithm*, which takes a computation and a trappified canvas and fills in the missing parts so that the

output is a computation containing both the client's computation and a trap.

Because we aim at blind delegating the execution of trappified canvases to a possibly fully malicious server that can deviate adaptively, a single trappified canvas will not be enough to constraint its behaviour significantly. Instead we randomise the construction of trappified canvases, and in particular the physical location of the trap. This gives rise to the concept of *trappified schemes* (Definition 5.2.7) which are sets of trappified canvases from which the client can sample efficiently.

Additionally, for these constructs to be useful in blind protocols they need to satisfy two properties. First, no information should leak to the server when it is using one trappified canvas over another. This means that executing one trappified canvas or another must be indistinguishable to the server. If this is the case, we say that they are *blind-compatible*. Second, no information should leak to the server about the computation in spite of being embedded into a larger computation that contains a trap. This implies that the decision to accept or reject the computation should not be depending on the client's desired computation. If this is the case, we call the embedding a *proper embedding*.

Finally, we examine the effect of deviations on individual trappified canvases as well as on trappified schemes. More precisely, we categorise deviations with the help of trappified schemes as follows: (i) if the scheme rejects with probability $(1-\epsilon)$, then it ϵ-detects the deviation; (ii) if the scheme accepts with probability $1-\delta$, it is δ-insensitive to the deviation; and finally (iii) if the result of all possible computations of interest is correct with probability $1-\nu$, then the scheme is ν-correct for this deviation.

Secure Verification from Trap Based Protocols (Section 5.3). Here, our contribution is a series of theorems that give general design rules for constructing secure, efficient and robust verification protocols based on the detection, insensitivity and correctness properties of trappified schemes.

We start by constructing a natural Prepare-and-send protocol from any trappified scheme, see the informal Protocol 3.

We then address the following question: what are the conditions required for these error-detection mechanisms to provide verification? The following theorem states that the trappified scheme should detect with high probability all errors for which the computation is not correct.

Theorem 5.1.1 (Detection Implies Verifiability, Informal)**.** *Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two sets*

Protocol 3 Trappified Delegated Blind Computation Protocol (Informal)

1. The Client samples a trappified canvas from the trappified scheme and embeds its computation, yielding a trappified pattern.
2. The Client blindly delegates this trappified pattern to the Server using the UBQC Protocol, after which the Client obtains the output of the trappified pattern.
3. The Client decides whether to abort or not based on the result of the decision function of the trappified canvas.
4. If it didn't abort, the Client performs some simple classical or quantum post-processing on the output.

of Pauli deviations such that $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$*, and* $\mathbb{I} \in \mathcal{E}_2$*. If the Trappified Delegated Blind Computation Protocol uses a trappified scheme which:*

- ϵ*-detects* $\mathcal{E}_1$*,*
- *is* δ*-insensitive to* $\mathcal{E}_2$*,*
- *is* ν*-correct on* $\mathcal{G}_V \setminus \mathcal{E}_1$*,*

then the protocol is $\max(\epsilon, \delta + \nu)$*-secure.*

In other words, it is acceptable to not detect a deviation so long as it has only little effect on the result of the computation of interest. This intuitive result is proved in the framework of Abstract Cryptography [MR11]. We introduce novel techniques to derive the protocol's composable security directly, without resorting to local criteria as in [Dun+14]. We construct a simulator that is able to correctly guess whether to accept or reject its interaction with the server without ever knowing what the client's computation is, thereby reproducing the behaviour of the concrete protocol although it is accessing a secure-by-design ideal delegated quantum computation resource. As such, this provides the first direct proof of composable security of the original VBQC protocol [FK17].

We next examine the conditions under which the protocol is robust against honest noise. We show that it is sufficient for the trappified scheme to be both insensitive to and correct on likely errors generated by the noise model.

Theorem 5.1.2 (Robust Detection Implies Robust Verifiability, Informal)**.** *We assume that the server in the Trappified Delegated Blind Computation Protocol is honest-but-noisy: the error applied is in* $\mathcal{E}_2$ *with probability* $(1 - p_2)$ *and* $\mathcal{G}_V \setminus \mathcal{E}_2$ *with probability*

p_2. Then, the client accepts with probability at least $(1-p_2)(1-\delta)$, and if accepted the distance between the implemented transformation and the client's computation is bounded by $\nu + p_2 + \delta$.

We conclude this theoretical deconstruction of verification protocols by exploring the necessary conditions for obtaining a security error which is exponentially close to zero without blowing up the server's memory requirements. We show that efficient trappified schemes must incorporate some error-correction mechanism.

Theorem 5.1.3 (Error-Correction Prevents Resource Blow-up, Informal)**.** *Assume that the Trappified Delegated Blind Computation Protocol has a negligible security error with respect to a security parameter λ. If the size of the output in the trappified pattern is the same as an unprotected execution of the Client's computation for a non-negligible fraction of trappified canvases in the trappified scheme used in the protocol, then the size of the common graph state required to implement the trappified patterns scales super-polynomially in λ.*

These results reveal the strong interplay between the deviation detection properties of trappified schemes and the properties of the corresponding prepare-and-send verification protocol. As a consequence, the optimisation of verification protocols translates into tailoring the deviation detection properties of trappified schemes to specific needs, for which the rich tools of error-correction can be used. This is the focus of the rest of this chapter.

Correctness and Security Amplification for Classical Input-Output Computations (Section 5.4). Here, we construct a general compiler for obtaining trappified schemes. It interleaves separate computations and test rounds in a way inspired by [Lei+21]. As a consequence, the overhead for protocols based on such schemes is simply a repetition of operations of the same size as the client's original computation, meaning that verification comes for free so long as the client and server can run the blind protocol. Using our correspondence between error-detection and verification, we then show that this compiler's parameters can be chosen to boost the constant detection and insensitivity rates of the individual test rounds to exponential levels after compilation.

Theorem 5.1.4 (From Constant to Exponential Detection and Insensitivity Rates, Informal)**.** *Let $\boldsymbol{P}$ be a trappified scheme and $\boldsymbol{P}'$ be the compiled version described above for n rounds combining a number of tests and computations which are both linear in n.*

If $\boldsymbol{P}$ ϵ-detects error set $\mathcal{E}_1$ and is δ-insensitive to $\mathcal{E}_2$, then there exists k_1, k_2 linear in n and ϵ', δ' exponentially-low in n such that $\boldsymbol{P}'$ ϵ'-detects errors with more than k_1 errors on all rounds from set $\mathcal{E}_1$ and is δ-insensitive to errors with less than k_2 errors from set $\mathcal{E}_2$.

This however not enough to obtain negligible security and, as per Theorem 5.1.3, we must recombine the results of the computation rounds to correct for these low-weight errors which are not detected. This is done by using a simple majority vote on the computation round outcomes, so that correctness can be independently amplified to an exponential level by using polynomially many computation rounds.

Theorem 5.1.5 (Exponential Correctness from Majority Vote, Informal)**.** *There exists k linear in n and ν exponentially-low in n such that $\boldsymbol{P}'$ is ν-correct so long as there are no more than k errors.*

In doing so, we have effectively untangled what drives correctness, security and robustness, thereby considerably simplifying the task of designing and optimising new protocols. More precisely, we can now focus only on the design of the test rounds as their performance greatly influences the value of exponents in the exponentials from the two previous theorems.

New Optimised Trappified Schemes from Stabiliser Testing (Section 5.5). In this section, we design test rounds and characterise their error-detection and insensitivity properties. This allows to recover the standard traps used in several other protocols, while also uncovering new traps that correspond to syndrome measurements of stabiliser generators – hence once again fruitfully exploiting the correspondence between error-detection and verification.

Finally, we combine all of the above into an optimisation of the deviation detection capability of the obtained trappified schemes that not only beats the current state-of-the-art, but more importantly provides an end-to-end application of our theoretical results.

5.1.3 Future Work and Open Questions

First, the uncovered connection between error-detection and verification raises further questions such as the extent to which it is possible to infer from the failed traps what the server has been performing.

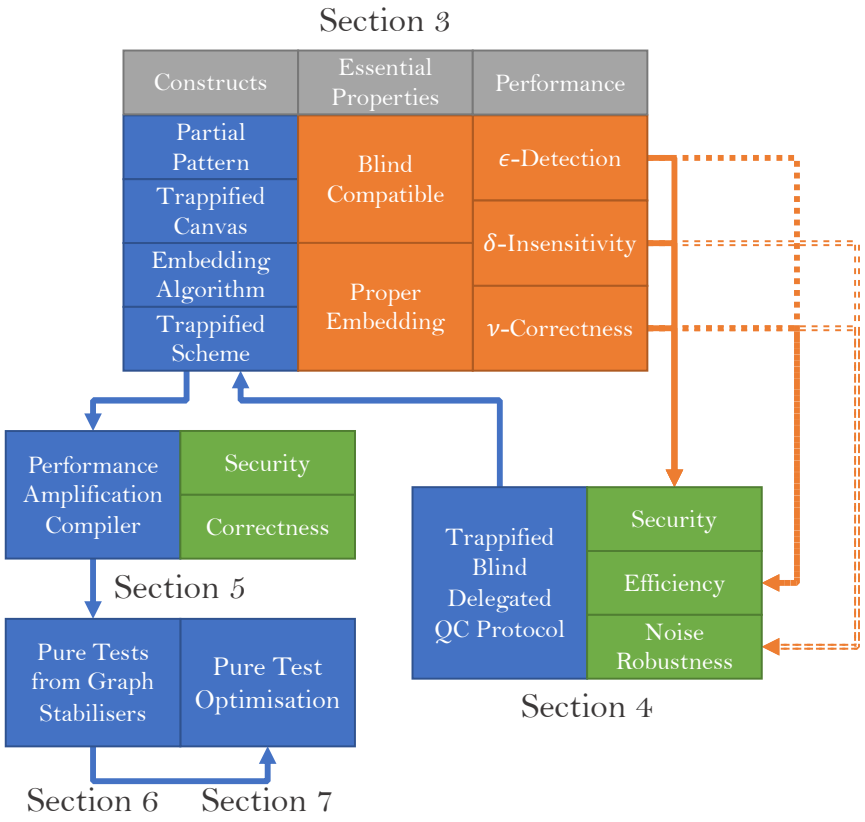


Figure 5.1: Structure of Chapter 5. The blue boxes represent the main objects which we construct, the orange ones are the main properties and the green the main theorems. The blue arrows go towards a higher level of granularity, meaning that an object can be simplified by using the next construction. The orange arrows indicate which property plays a role in the proof of each theorem.

Second, Theorem 5.1.3 implies that some form of error-correction is necessary to obtain exponential correctness. Yet, our protocol shows that sometimes classical error-correction is enough, thereby raising the question of understanding what are the optimal error-correction schemes for given classes of computation that are to be verified.

5.2 Analysing Deviations with Traps

The goal of this section is to introduce the concepts and tools for detecting deviations from a given computation. Later, in Section 5.3, we combine these techniques with blindness in order to detect malicious deviations, i.e. perform verification.

5.2.1 Abstract Definitions of Traps

We start by defining partial MBQC patterns in Definition 5.2.1, which fix only a subset of the measurement angles and flow conditions on a given graph. We constrain the flow such that the determinism of the computation is preserved on the partial pattern independently of the how the rest of the flow is specified.

Definition 5.2.1 (Partial MBQC Pattern)**.** *Given a graph $G = (V, E)$, a partial pattern P on G is defined by:*

- *$G_P = (V_P, E_P = E \cap V_P \times V_P)$, a subgraph of G;*
- *I_P and O_P, the partial input and output vertices, with subspaces $\Pi_{I,P}$ and $\Pi_{O,P}$ defined on vertices I_P and O_P through bases $\mathcal{B}_{I,P}$ and $\mathcal{B}_{O,P}$ respectively;*
- *$\{\phi(i)\}_{i \in V_P \setminus O_P}$, a set of measurement angles;*
- *$f_p : V_P \setminus O_P \to V_P \setminus I_P$, a flow inducing a partial order $\preceq_P$ on V_P.*

Example 1 (Partial Pattern for Computing)**.** *Let G be the $n \times m$ 2D-cluster graph – i.e. n-qubit high and m-qubit wide – and the ordering of the qubits starting in the upper-left corner, going down first then right. Such graph state is universal for MBQC [RB01]. There are many possible partial patterns that can be defined on such graph. For instance, consider a pattern Q that runs on a smaller $n' \times m'$ 2D-cluster graph. Then, one can define a partial pattern P on G as the top-left $(n'+1) \times (m'+1)$ subgraph. The set I_P is defined as the set I of Q together with all the qubits on the bottom row and right column. The input space corresponds to the Hilbert space of the input qubits of Q tensored with $|0\rangle$ for the qubits of the bottom row and right column. The output set O_P is the same set as in Q and $\Pi_{O,P}$ is the full Hilbert space of the output qubits. The measurement angles are the same as in Q for the corresponding qubits and set to be random for the bottom row and right column. The flow is the same as in Q, provided that the added $|0\rangle$ qubits have no dependent qubits. Because the added qubits are forced to be in the $|0\rangle$ state, this isolates a $n' \times m'$ 2D-cluster graph that can then be used to perform the same operations as in Q, thereby allowing to compute the same unitary, albeit using a larger graph, see Figure 5.2. Note that one can change the location of the $n' \times m'$ 2D-cluster graph used for the computation, as long as it is properly surrounded by qubits in the $|0\rangle$ state. This is done by defining the input subspace of the partial pattern to take that constraint into account.*

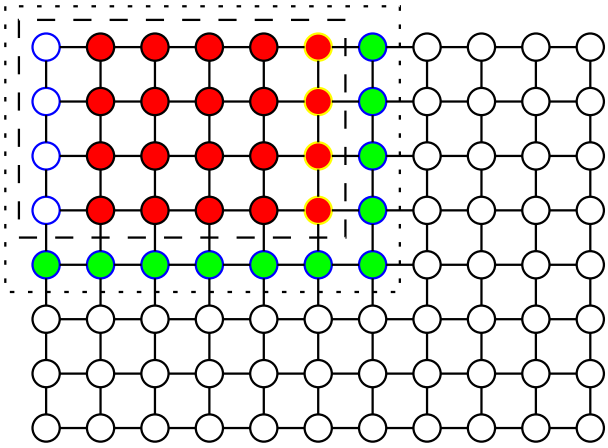


Figure 5.2: Partial pattern for computing. The partial pattern is in the dashed box. Input qubits are surrounded in blue, output qubits in yellow. Red filled qubits are prepared in $|+\rangle$ while the green ones are prepared in $|0\rangle$. The green qubits define a subspace of the Hilbert space of the input qubits that guarantees that a 4×6 cluster state computation can be run inside the long-dashed box.

We now use this notion to define trappified canvases. These contain a partial pattern whose input state is fixed such that it produces a sample from an easy to compute probability distribution when its ouput qubits are measured in the X basis. These partial patterns are called *traps* and will be used to detect deviations in the following way. Whenever a trap computation is executed, it should provide outcomes that are compatible with the trap's probability distribution. Failure to do so is a sign that the server deviated from the instructions given by the client.

Definition 5.2.2 (Trappified Canvas)**.** *A* trappified canvas $(T, \sigma, \mathcal{T}, \tau)$ *on a graph* $G = (V, E)$ *consists of:*

- T*, a partial pattern on a subset of vertices* V_T *of* G *with input and output sets* I_T *and* O_T*;*
- σ*, a tensor product of single-qubit states on* $\Pi_{I,T}$*;*
- $\mathcal{T}$*, an efficiently classically computable probability distribution over binary strings;*
- *and* τ*, an efficient classical algorithm that takes as input a sample from* $\mathcal{T}$ *and outputs a single bit;*

such that the X*-measurement outcomes of qubits in* O_T *are drawn from probability distribution* $\mathcal{T}$*. Let* t *be such a sample, the outcome of the trappified canvas is given by* $\tau(t)$*. By convention we say that it accepts whenever* $\tau(t) = 0$ *and rejects for* $\tau(t) = 1$*.*

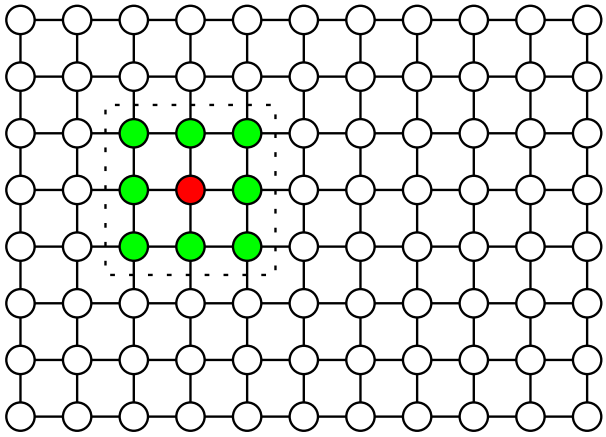


Figure 5.3: Trappified canvas. The partial pattern inside the dashed box is the trap. The central qubit (red) is prepared in $|+\rangle$ and is surrounded by $|0\rangle$'s (green) that effectively ensure that irrespective of the measurement angles on the remaining qubits the central qubit will remain in $|+\rangle$. Failure to obtain the 0-outcome when measuring X will be a proof that the server deviated from the given instructions. The preparation and measurement angles of the remaining qubits is left unspecified.

We will often abuse the notation and refer to the trappified canvas $(T, \sigma, \mathcal{T}, \tau)$ as T.

Example 2 (Canvas with a Single Standard Trap)**.** *Consider the $n \times m$ 2D-cluster graph and consider the partial pattern of Example 1 where the subgraph is a 3×3 square – i.e. a single computation qubit surrounded by 8 $|0\rangle$ states. The input state is fixed to be $\sigma = |+\rangle \otimes |0\rangle^{\otimes 8}$ where $|+\rangle$ is the state of the central qubit, the others being the aforementioned peripheral ones. Because the central qubit is measured along the X-axis $\mathcal{T}$ is deterministic – the measurement outcome 0 corresponding to the projector $|+\rangle\langle+|$ has probability 1. The accept function is defined by $\tau(t) = t$ so that the trappified canvas accepts whenever the measurement outcome of the central qubit corresponds to the expected 0 outcome. Here, the 3×3 partial pattern defines a trap(see Figure 5.3).*

Note that the input and output qubits of a partial pattern may not be included in the input and output qubits of the larger MBQC graph. This gives us more flexibility in defining trappified canvases: during the protocol presented in the next section, the server will measure all qubits in O^c – which may include some of the trap outputs –, while any measurement of qubits in O will be performed by the client. This allows the trap to catch deviations on the output qubits as well.

In order to be useful, trappified canvases must contain enough empty space – vertices which have been left unspecified – to accommodate the client's desired computation. Inserting this computation is done via an *embedding algorithm* as described in the

following Definition.

Definition 5.2.3 (Embedding Algorithm)**.** *Let $\mathfrak{C}$ be a class of quantum computations. An* embedding algorithm *$E_{\mathfrak{C}}$ for $\mathfrak{C}$ is an efficient classical probabilistic algorithm that takes as input:*

- $\mathsf{C} \in \mathfrak{C}$*, the computation to be embedded;*
- $G = (V, E)$*, a graph, and an output set O;*
- T*, a trappified canvas on graph G;*
- $\preceq_G$*, a partial order on V which is compatible with the partial order defined by T;*

and outputs:

- *a partial pattern C on $V \setminus V_T$, with*
  - *input and output vertices $I_C \subset V \setminus V_T$ and $O_C = O \setminus O_T$;*
  - *two subspaces (resp.) $\Pi_{I,C}$ and $\Pi_{O,C}$ of (resp.) I_C and O_C with bases (resp.) $\mathcal{B}_{I,C}$ and $\mathcal{B}_{O,C}$;*
- *a decoding algorithm $\mathsf{D}_{O,C}$;*

such that the flow f_C of partial pattern C induces a partial order which is compatible with $\preceq_G$. If $E_{\mathfrak{C}}$ is incapable of performing the embedding, it outputs $\perp$.

As will be come apparent in later definitions, a good embedding algorithm will yield patterns which apply a desired computation C to any input state in subspace $\Pi_{I,C}$, with the output being in subspace $\Pi_{O,C}$ after the decoding algorithm has been run. The decoding algorithm can be quantum or classical depending on the nature of the output. We will furthermore require all embedding algorithms in this chapter to have the following property.

Definition 5.2.4 (Proper Embedding)**.** *We say that an embedding algorithm $E_{\mathfrak{C}}$ is* proper *if, for any computation $\mathsf{C} \in \mathfrak{C}$ and trappified canvas T that do not result in a $\perp$ output, we have that:*

- *f_C does not induce dependencies on vertices V_T of partial pattern T;*

- *the input and output subspaces $\Pi_{I,C}$ and $\Pi_{O,C}$ do not depend on the trappified canvas T.*

Example 3 (Embedding Algorithm on a 2D-Cluster Graph Canvas with a Single Trap)**.** *Define $\mathfrak{C}$ as the class of computations that can be implemented using a $(n-3)\times m$ 2D-cluster state. An embedding algorithm for $\mathfrak{C}$ on T can be defined in the following way. Consider the trappified canvas T of Example 2 with a $n\times m$ 2D-cluster graph and a single 3×3 trap in the upper left corner. The output of the embedding algorithm would be the pattern P defined in the following way. For $C\in\mathfrak{C}$, by assumption, one can define a pattern Q on a $(n-3)\times m$ 2D-cluster graph that implements C. The angles and flow of the partial pattern P is identical to that of Q albeit applied on the lower $n-3$ rows of T. On the $3\times(m-3)$ upper right rectangular subgraph, all angles are set randomly. I_C is such that it comprises all inputs defined in Q and the last $m-3$ qubits of the third row. Choose $\Pi_{I,P}$ so that these $m-3$ qubits are set to $|0\rangle$. Then, by construction, this together with the trap isolates a $(n-3)\times m$ rectangular subgraph on which P will be defining MBQC instructions identical to those of Q, thereby implementing C. In addition, one can see that there are no dependency between measurements of P and that of the trap in T so that the embedding algorithm is proper. Note that one can change the location of the trap to any column. If in addition the 2D-cluster graph if cylindrical instead of rectangular, the trap can be moved to any location within the cylinder.*

Definition 5.2.5 (Trappified Pattern)**.** *Let $E_{\mathfrak{C}}$ be an embedding algorithm for $\mathfrak{C}$. Given a computation $\mathsf{C}\in\mathfrak{C}$ and a trappified pattern T on graph G with order $\preceq_G$, we call the completed pattern $C\cup T$ which is the first output of $E_{\mathfrak{C}}(\mathsf{C},G,T,\preceq_G)$ a* trappified pattern*.*

While embedding a computation in a graph that has enough space for it might seem simple, the hard part is to ensure that the embedding is *proper*. This property implies that no information is carried via the flow of the global pattern from the computation to the trap and it is essential for the security of the verification protocol built using trappified canvases. In Example 3 above, this is done by breaking the graph using the states initialised in $|0\rangle$. The only other known way is to separate runs for tests and computations and satisfying this condition using other methods is left as an open question.

Note that the input and output qubits of the computation C might be constrained to be in (potentially strict) subspaces $\Pi_{I,C}$ and $\Pi_{O,C}$ of I_C and O_C respectively. This allows for error-protected inputs and outputs, without having to specify any implementation

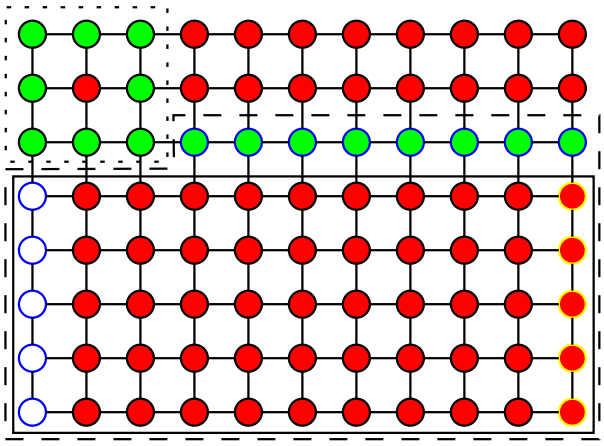


Figure 5.4: Trappified canvas. Input qubits are surrounded in blue, output qubits in yellow. Red filled qubits are prepared in $|+\rangle$ while the green ones are prepared in $|0\rangle$, white ones are left unspecified. The trap is located in the upper left corner. The actual computation takes place in the 5×11 rectangular cluster state surrounded by a solid-line while the computation pattern comprises the qubits surrounded by long-dashed line. This allows to include some dummy $|0\rangle$ qubits in the inputs so as to disentangle the lower 5 rows from the rest of the graph and perform the computation. Output qubits of the computation are surrounded by a yellow line. The partial pattern inside the dashed box is the trap. The central qubit (red) is prepared in $|+\rangle$ and is surrounded by $|0\rangle$'s (green) that effectively ensure that irrespective of the measurement angles on the remaining qubits the central qubit will remain in $|+\rangle$. Failure to obtain the 0-outcome when measuring X will be a proof that the server deviated from the given instructions. The preparation and measurement angles of the remaining qubits is left unspecified.

for the error-correction scheme. In particular, it encompasses encoding classical output data as several, possibly noisy, repetitions which will be decoded by the client through a majority vote as introduced in [Lei+21]. It also allows to take into account the case where the trappified pattern comprises a fully fault-tolerant MBQC computation scheme for computing C using topological codes as described in [RJK07].

For verification, our scheme must be able to cope with malicious behaviour: detecting deviations is useful for verification only so long as the server cannot adapt its behaviour to the traps that it executes. Otherwise, it could simply decide to deviate exclusively on non-trap qubits. This is achieved by executing the patterns in a blind way so that the server has provably no information about the location of the traps and cannot avoid them with high probability. To this end, we define *blind-compatible* patterns as those which share the same graph, output vertices and measurement order of their qubits. The UBQC Protocol (Protocol 2) leaks exactly this information to the server, meaning that it cannot distinguish the executions of two different blind-compatible patterns.

Definition 5.2.6 (Blind-Compatibility)**.** *A set of patterns $\boldsymbol{P}$ is* blind-compatible *if all patterns $P \in \boldsymbol{P}$ share the same graph G, the same output set O and there exists a partial ordering $\preceq_{\boldsymbol{P}}$ of the vertices of G which is an extension of the partial ordering defined by the flow of any $P \in \boldsymbol{P}$. This definition can be extended to a set of trappified canvases $\boldsymbol{P} = \{(T, \sigma, \mathcal{T}, \tau)\}$. The partial order $\preceq_{\boldsymbol{P}}$ is required to be an extension of the orderings $\preceq_T$ of partial patterns T.*

A single trap is usually not sufficient to catch deviations on more than a subset of positions of the graph. In order to catch all deviations, it is then necessary to randomise the blind delegated execution over multiple patterns. We therefore define a trappified scheme as a set of blind-compatible trappified canvases which can be efficiently sampled according to a given distribution, along with an algorithm for embedding computations from a given class into all the canvases.

Definition 5.2.7 (Trappified Scheme)**.** *A* trappified scheme $(\boldsymbol{P}, \preceq_G, \mathcal{P}, E_{\mathfrak{C}})$ *over a graph G for computation class $\mathfrak{C}$ consists of:*

- *$\boldsymbol{P}$, a set of blind-compatible trappified canvases over graph G with common partial order $\preceq_{\boldsymbol{P}}$;*
- *$\preceq_G$, a partial ordering of vertices V of G that is an extension of $\preceq_{\boldsymbol{P}}$;*
- *$\mathcal{P}$, a probability distribution over the set $\boldsymbol{P}$ which can be sampled efficiently;*

- *$E_{\mathfrak{C}}$, an proper embedding algorithm for $\mathfrak{C}$;*

such that for all $\mathsf{C} \in \mathfrak{C}$ and all trappified canvases $T \in \boldsymbol{P}$, $E_{\mathfrak{C}}(\mathsf{C}, G, T, \preceq_G) \neq \perp$, i.e. any computation can be embedded in any trappified canvas using the common order $\preceq_G$.

Without loss of generality, in the following, the probability distribution used to sample the trappified canvases will generally be $u(\boldsymbol{P})$, the uniform distribution over $\boldsymbol{P}$. The general case can be approximated from the uniform one with arbitrary fixed precision by having several copies of the same canvas in $\boldsymbol{P}$. We take $T \sim \boldsymbol{P}$ to mean that the trappified canvas is sampled according to the distribution $\mathcal{P}$ of trappified scheme $\boldsymbol{P}$.

Note that in Definition 5.2.7 above, while the blindness condition ensures that a completed patterns obtained after running the embedding algorithm hides the location of the traps, the existence of a partial order $\preceq_G$ compatible with that of the trappified canvases ensures that this remains true when considering the scheme as a whole, i.e the order in which the qubits are measured does not reveal information about the chosen trappified canvas itself, which would otherwise break the blindness of the scheme.

Example 4 (Trappified Scheme for a Cylindrical-Cluster Graph)**.** *Consider the set of trappified canvases together with the embedding algorithm $E_{\mathfrak{C}}$ on the cylindrical cluster-graph with a single randomly placed 3×3 trap as defined in Example 3. This defines a trapification scheme for $\mathfrak{C}$ consisting of computations that can be implemented using a $(n-3) \times m$ 2D-cluster graph (See Figure 5.5).*

5.2.2 Effect of Deviations on Traps

We can now describe the purpose of the objects described in the previous subsection, namely detecting the server's deviations from their prescribed operations during a given delegated computation. We start by recalling that the blindness of UBQC Protocol is obtained by Pauli-twirling the operations delegated to the server. This implies that any deviation can be reduced to a convex combination of Pauli operators. Then, we formally define Pauli deviation detection and insensitivity for trappified canvases and schemes. We show in the next section that these key properties are sufficient for obtaining a verifiable delegated computation by formalising the steps sketched here.

When a client delegates the execution of a pattern P to a server using Protocol 2, the server can potentially deviate in an arbitrary way from the instructions it receives. By converting into quantum states both the classical instructions sent by the client –

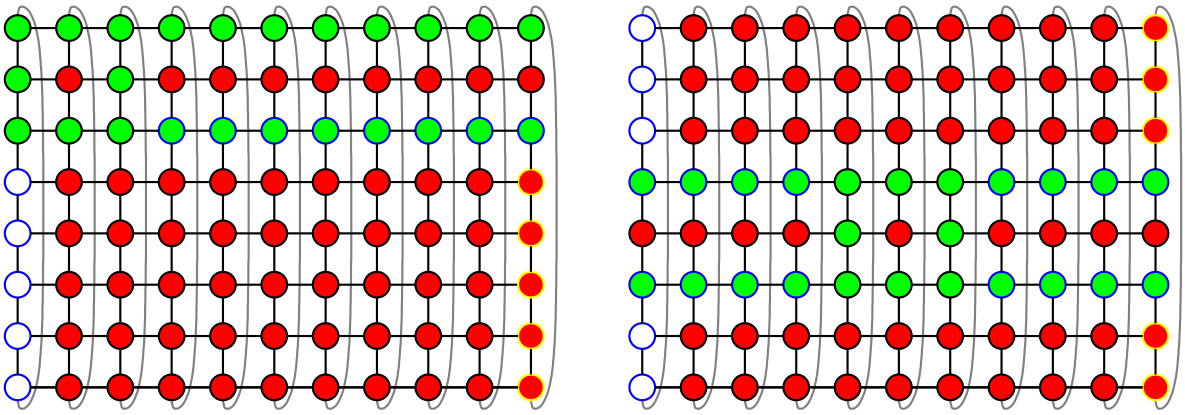


Figure 5.5: Trappified scheme. Two possible canvases extracted from a trappified scheme with a single 3×3 trap on a toric 8×11 toric cluster state. Input qubits are surrounded in blue, output qubits in yellow. Red filled qubits are prepared in $|+\rangle$ while the green ones are prepared in $|0\rangle$, white ones are left unspecified. The actual computation takes place in the 5×11 rectangular cluster state, while the trap is located at a different positions in each picture allowing to detect all possible deviations performed by a server, albeit with a low probability of success.

i.e. the measurement angles – and the measurement outcomes sent back by the server, all operations on the server's side can be modelled as a unitary F acting on all the qubits sent by the client and some ancillary states $|0\rangle_S$, before performing measurements in the computational basis to send back the outcomes $|b\rangle$ that the client expects from the server.

The instructions of the server in an honest execution of the UBQC Protocol 2 correspond to:

1. entangling the received qubits corresponding to the vertices of the computation graph with operation $E_G = \bigotimes_{(i,j)\in E} \mathsf{CZ}_{i,j}$;

2. performing rotations on non-output vertices around the Z-axis, controlled by the qubits which encode the measurement angles instructed by the client;

3. applying a Hadamard gate H on all non-output vertices;

4. measuring non-output vertices in the $\{|0\rangle, |1\rangle\}$ basis.

The steps (i-iii) correspond to a unitary transformation U_P that depends only on the public information that the server has about the pattern P – essentially the computation graph G and an order of its vertices compatible with the flow of P.

Hence, the unitary part U_P of the honestly executed protocol for delegating P can always be extracted from F, so that $\mathsf{F} = \mathsf{F}' \circ \mathsf{U}_P$. Here, F' is called a *pure deviation* and is applied right before performing the computational basis measurements for non-output qubits and right before returning the output qubits to the Client.

When the pattern is executed blindly using Protocol 2, the state in the server's registers during the execution is a mixed state over all possible secret parameters chosen by the client. It is shown in [Kap16] that the resulting summation over the secret parameters which hide the inputs, measurement angles and measurement outcomes is equivalent to applying a Pauli twirl to the pure deviation F'. This effectively transforms it into a convex combination of Pauli operations applied after U_P.

Hence, any deviation by the server can be represented without loss of generality by choosing with probability $\Pr[\mathsf{E}]$ an operator E in the Pauli group $\mathcal{G}_V$ over the vertices V of the graph used to define P, and executing $\mathsf{E} \circ \mathsf{U}_P$ instead of U_P for the unitary part of the protocol. By a slight abuse of notation, such transformation will be denoted $\mathsf{E} \circ P$. Furthermore, if $C \cup T$ is a trappified pattern obtained from a trappified canvas T that samples $t = (t_1, \ldots, t_N)$ from the distribution $\mathcal{T}$, then in the presence of deviation E, it will sample from a different distribution. For instance, whenever E applies a Z operator on a vertex, it can be viewed as an execution of a pattern where the angle δ for this vertex is changed into $\delta + \pi$. Whenever E applies a X operator on a vertex, δ is transformed into $-\delta$. We now give a lemma which will be useful throughout the rest of this chapter.

Lemma 5.2.8 (Independence of Trap and Computation)**.** *Let $C \cup T$ be a trappified pattern obtained from the trappified canvas T which samples from distribution $\mathcal{T}$ through a proper embedding algorithm of computation C. Then, for all Pauli errors E, the distribution of trap measurement outcomes is independent of the computation C and of the input state in the subspace $\Pi_{I,C}$.*

Proof. Let f_C be the flow of computation of the embedded computation C. Because the embedding is proper according to Definition 5.2.3, the dependencies induced by f_C do not affect trap qubits V_T. Furthermore, the input of the trap is fixed along with its partial pattern, independently of the computation. Therefore, the distribution of the trap measurement outcomes is also independent of the embedded computation being performed on the rest of the graph as well as the input state of such computation. □

Indeed, for a completed trappified pattern $C \cup T$ obtained by embedding a computation C onto a trappified canvas T, the action of E on the vertices outside V_T does not have

an impact on the measurement outcomes of the vertices in V_T. This allows to define the trap outcome distribution under the influence of error E solely as a fuction of E and $\mathcal{T}$. Such modified distribution is denoted $\mathsf{E} \circ \mathcal{T}$.

As an additional consequence, it is possible to define what it means for a given trappified canvas to detect and to be insensitive to Pauli errors:

Definition 5.2.9 (Pauli Detection)**.** *Let T be a trappified canvas sampling from distribution $\mathcal{T}$. Let $\mathcal{E}$ be a subset of $\mathcal{G}_V$. For $\epsilon > 0$, we say that T* ϵ-detects *$\mathcal{E}$ if:*

$$\forall \mathsf{E} \in \mathcal{E}, \ \Pr_{t \sim \mathsf{E} \circ \mathcal{T}}[\tau(t) = 1] \geq 1 - \epsilon. \tag{5.1}$$

We say that a trappified scheme $\boldsymbol{P}$ ϵ-detects *$\mathcal{E}$ if:*

$$\forall \mathsf{E} \in \mathcal{E}, \ \sum_{T \in \boldsymbol{P}} \Pr_{\substack{T \sim \mathcal{P} \\ t \sim \mathsf{E} \circ \mathcal{T}}}[\tau(t) = 1, T] \geq 1 - \epsilon. \tag{5.2}$$

Definition 5.2.10 (Pauli Insensitivity)**.** *Let T be a trappified canvas sampling from distribution $\mathcal{T}$. Let $\mathcal{E}$ be a subset of $\mathcal{G}_V$. For $\delta > 0$, we say that T is* δ-insensitive to *$\mathcal{E}$ if:*

$$\forall \mathsf{E} \in \mathcal{E}, \ \Pr_{t \sim \mathsf{E} \circ \mathcal{T}}[\tau(t) = 0] \geq 1 - \delta. \tag{5.3}$$

We say that a trappified scheme $\boldsymbol{P}$ is δ-insensitive to *$\mathcal{E}$ if:*

$$\forall \mathsf{E} \in \mathcal{E}, \ \sum_{T \in \boldsymbol{P}} \Pr_{\substack{T \sim \mathcal{P} \\ t \sim \mathsf{E} \circ \mathcal{T}}}[\tau(t) = 0, T] \geq 1 - \delta. \tag{5.4}$$

Above, the probability distribution stems both from the randomness of quantum measurements of the trap output qubits yielding the bit string t, and the potentially probabilistic nature of the decision function τ. In the case of trappified schemes, the probability distribution for obtaining a given result for τ also depends on the choice of canvas $T \in \boldsymbol{P}$, sampled according to the probability distribution $\mathcal{P}$.

In the same spirit, there are physical deviations that nonetheless produce little effect on the computations embedded into trappified canvases and trappified schemes. When they occur, the computation is still almost correct.

Definition 5.2.11 (Pauli Correctness)**.** *Let $(T, \sigma, \mathcal{T}, \tau)$ be a trappified canvas and $E_{\mathfrak{C}}$ an embedding algorithm. Let $C \cup T$ be the pattern obtained by embedding a computation $\mathsf{C} \in \mathfrak{C}$ on T using $E_{\mathfrak{C}}$ and let $|\psi\rangle$ be a state in $I_C \otimes R$, for sufficiently large auxiliary*

*system R, such that $\mathrm{Tr}_R(|\psi\rangle) \in \Pi_{I,C}$, where $\Pi_{I,C}$ is the client's input subspace. Let $\mathcal{E}$ be a subset of $\mathcal{G}_V$. For $\mathsf{E} \in \mathcal{E}$, we define $\tilde{\mathsf{C}}_{T,\mathsf{E}} = \mathsf{D}_{O,C} \circ \mathrm{Tr}_{O_C^c} \circ \mathsf{E} \circ (C \cup T)$ to be the CPTP map resulting from applying the trappified pattern $C \cup T$ followed by the decoding algorithm $\mathsf{D}_{O,C}$ on the output of the computation. For $\nu \geq 0$, we say that T is ν-*correct *on $\mathcal{E}$ if:*[1]

$$\forall \mathsf{E} \in \mathcal{E},\ \forall \mathsf{C} \in \mathfrak{C},\ \max_{\psi} \|(\tilde{\mathsf{C}}_{T,\mathsf{E}} - \mathsf{C} \otimes \mathbb{I}_T) \otimes \mathbb{I}_R(|\psi\rangle\langle\psi| \otimes \sigma)\|_{\mathrm{Tr}} \leq \nu. \tag{5.5}$$

This is extended to a trappified scheme $\boldsymbol{P}$ by requiring the bound to hold for all $T \in \boldsymbol{P}$.

In the following, sets of deviations that have little effect on the result of the computation according to diamond distance will be called *harmless*, while their complement are *possibly harmful*.

We conclude this section with some remarks regarding basic properties of trappified schemes and a simple but powerful result allowing to construct trappification schemes from simpler ones.

Remark 5.2.12 (Existence of Harmless Deviations)**.** *Why not just detect all possible deviations rather than count on the possibility that some have little impact on the actual computation? The reason is that these are plentiful in MBQC. Following our convention to view all measurements as computational basis measurements preceded by an appropriate rotation, any deviation E that acts as $\mathbb{I}$ and Z on measured qubits does not change the measurement outcomes and have no effect on the final outcome. Consequently, for classical output computations, only X and Y deviations need to be analysed. These are equivalent to flipping the measurement outcome, which propagate to the output via the flow corrections.*

Remark 5.2.13 (A Trappified Canvas is a Trappified Scheme)**.** *Any trappified canvas T can be seen as a trappified scheme $\boldsymbol{P} = \{T\}$ and the trivial distribution. If the trappified pattern ϵ-detects $\mathcal{E}_1$ and is δ-insensitive to $\mathcal{E}_2$, so is the corresponding trappified scheme.*

Remark 5.2.14 (Pure Traps)**.** *A trappified scheme $\boldsymbol{P}$ may only consist of trappified canvases that cover the whole graph $G = (V, E)$ if $V_T = V$ for all $T \in \boldsymbol{P}$. This corresponds to the special case where the trappified scheme cannot embed any computation, i.e. $\mathfrak{C} = \emptyset$ and the embedding algorithm applied to a canvas $T \in \boldsymbol{P}$ always return T. The detection,*

[1] Equation 5.5 corresponds to the diamond norm between the correct and deviated CPTP maps, but with a fixed input subspace and a fixed input for the trap qubits.

insensitivity and correctness properties also apply to this special case, although the correctness is trivially satisfied.

Lemma 5.2.15 (Simple Composition of Trappified Schemes)**.** *Let $(\boldsymbol{P}_i)_i$ be a sequence of trappified schemes with corresponding distributions $\mathcal{P}_i$ such that $\boldsymbol{P}_i$ ϵ_i-detects $\mathcal{E}_1^{(i)}$ and is δ_i-insensitive to $\mathcal{E}_2^{(i)}$. Let $(p_i)_i$ be a probability distribution.*

Let $\boldsymbol{P} = \bigcup_i \boldsymbol{P}_i$ be the trappified scheme with the following distribution $\mathcal{P}$:

1. *Sample a trappified scheme $\boldsymbol{P}_j$ from $(\boldsymbol{P}_i)_i$ according to $(p_i)_i$;*
2. *Sample a trappified canvas from $\boldsymbol{P}_j$ according to $\mathcal{P}_j$.*

Let $\mathcal{E}_1 \subseteq \bigcup_i \mathcal{E}_1^{(i)}$ and $\mathcal{E}_2 \subseteq \bigcup_i \mathcal{E}_2^{(i)}$. Then, $\boldsymbol{P}$ ϵ-detects $\mathcal{E}_1$ and is δ-insensitive to $\mathcal{E}_2$ with:

$$1 - \epsilon = \min_{\mathsf{E} \in \mathcal{E}_1} \sum_{\substack{i \\ \mathsf{E} \in \mathcal{E}_1^{(i)}}} p_i(1 - \epsilon_i), \text{ and} \tag{5.6}$$

$$1 - \delta = \min_{\mathsf{E} \in \mathcal{E}_2} \sum_{\substack{i \\ \mathsf{E} \in \mathcal{E}_2^{(i)}}} p_i(1 - \delta_i). \tag{5.7}$$

Note that we do not consider above the embedding function. If we assume that all schemes can embed the same set of computations, then it is possible to use the embedding of the one which is chosen at step 1 above. We will see later an example of how to combine trappified schemes with different computation classes in Section 5.4.

5.3 Secure Verification from Trap Based Protocols

In this section we use the properties defined above to derive various results which help break down the tasks of designing and proving the security of verification protocols into small and intuitive pieces. We start by giving a description of a general protocol using trappified schemes which encompasses all prepare-and-send MBQC-based protocol aiming to implement the SDQC funtionality (Definition 2). We then relate the security of this protocol in the Abstract Cryptography framework to the ϵ-detection, δ-insensitivity and ν-correctness of the trappified scheme used in the protocol. Consequently, we can from then on only focus on these three properties instead of looking at the full protocol, which already removes a lot of steps in future proofs.

Then we demonstrate how increasing the insensitivity set yields a protocol which is robust to situations where the server is honest-but-noisy with a contained noise

parameter. These results further simplify the design of future protocols since many complex proofs can be avoided, allowing us to concentrate on designing more efficient trappified schemes and directly plugging them into the generic protocol and compiler to yield exponentially-secure and noise-robust protocols implementing SDQC. We finally describe a consequence of these results in the case where the security of the protocol is exponentially-low in a given security parameter. We show that this automatically implies that the computation must be protected against low-weight errors if we restrict the server's resources to be polynomial in the security parameter.

5.3.1 General Verification Protocol from Trappified Schemes

Given a computation C, it is possible to delegate its trappified execution in a blind way. To do so, the Client simply chooses one trappified canvas from a scheme at random, inserts into it the computation C using an embedding algorithm and blindly delegates the execution of the resulting trappified pattern to the Server. The steps are formally described in Protocol 4.

Note that this protocol offers blindness not only at the level of the chosen trappified pattern, but also at the level of the trappified scheme itself. More precisely, by delegating the chosen pattern, the client reveals at most the graph of the pattern, a partial order of its vertices and the location of the output qubits of the pattern, if there are any, comprising computation and trap outputs. However, trappified patterns of a trappified scheme are blind-compatible, that is they share the same graph and same set of output qubits. Therefore, the above protocol also hides which trappified pattern has been executed among all possible ones, hence concealing the location of traps.

Blind Deviation Detection Implies Verifiability. We now formalise the following intuitive link between deviation detection and verification in the context of delegated computations. On one hand, if a delegated computation protocol is correct,[2] not detecting any deviation by the server from its prescribed sequence of operations should be enough to guarantee that the final result is correct. Conversely, detecting that some operations have not been performed as specified should be enough for the client to reject potentially incorrect results. Combining those two cases should therefore yield a verified delegated computation.

[2]Here we use correctness in a cryptographic setting, meaning that all parties execute as specified their part of the protocol.

Protocol 4 Trappified Delegated Blind Computation

Public Information:

- $\mathfrak{C}$, a class of quantum computations;
- $G = (V, E)$, a graph with output set O;
- $\boldsymbol{P}$, a trappified scheme on graph G;
- $\preceq_G$, a partial order on V compatible with $\boldsymbol{P}$.

Client's Inputs: A computation $\mathsf{C} \in \mathfrak{C}$ and a quantum state ρ_C compatible with C.
Protocol:

1. The Client samples a trappified canvas T from the trappified scheme $\boldsymbol{P}$.
2. The Client runs the embedding algorithm $E_{\mathfrak{C}}$ from $\boldsymbol{P}$ on its computation C, the graph G with output space O, the trappified pattern T, and the partial order $\preceq_G$. It obtains as output the trappified pattern $C \cup T$.
3. The Client and Server blindly execute the trappified pattern $C \cup T$ on input state ρ_C using the UBQC Protocol 2.
4. If the output set is non-empty (if there are quantum outputs), the Server returns the qubits in positions O to the Client.
5. The Client measures the qubits in positions $O \cap V_T$ in the X basis. It obtains the trap sample t.
6. The Client checks the trap by computing $\tau(t)$. It rejects and outputs $(\perp, \mathsf{Rej})$ if $\tau(t) = 1$.
7. Otherwise, the Client accepts the computation. It applies the decoding algorithm $\mathsf{D}_{O,C}$ to the output of Protocol 2 on vertices $O \setminus V_T$ and set the result as its output along with Acc.

To this end, we show how the deviation detection capability of trappified schemes is used to perform verification. This is done by proving that Protocol 4 above constructs the Secure Delegated Quantum Computation Resource 2 in the Abstract Cryptography framework. This resource allows a Client to input a computation and a quantum state and to either receive the correct outcome or an abort state depending on the Server's choice, whereas the Server only learns at most some well defined information contained in a leak l_ρ. More precisely, we show that any distinguisher has a bounded distinguishing advantage between the real and ideal scenarios so long as the trappified scheme $\boldsymbol{P}$ detects a large fraction of deviations that are possibly harmful to the computation.

Theorem 5.3.1 (Detection Implies Verifiability)**.** *Let $\boldsymbol{P}$ be a trappified scheme with a proper embedding. Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two sets of Pauli deviations such that $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$, and $\mathbb{I} \in \mathcal{E}_2$. If $\boldsymbol{P}$:*

- *ϵ-detects $\mathcal{E}_1$,*
- *is δ-insensitive to $\mathcal{E}_2$,*
- *is ν-correct on $\mathcal{G}_V \setminus \mathcal{E}_1$,*

for $\epsilon, \delta, \nu > 0$, then the Trappified Delegated Blind Computation Protocol 4 for computing CPTP maps in $\mathfrak{C}$ using $\boldsymbol{P}$ is $\delta+\nu$-correct and $\max(\epsilon, \nu)$-secure in the Abstract Cryptograhy framework, i.e. it $\max(\epsilon, \delta + \nu)$-constructs the Secure Delegated Quantum Computation Resource 2 where the leak is defined as $l_\rho = (\mathfrak{C}, G, \boldsymbol{P}, \preceq_G)$.

Proof of Correctness. We start by analysing the correctness of Protocol 4, i.e. the distance between the real and ideal input/output relation if both parties follow their prescribed operations. Let $\mathsf{C} \in \mathfrak{C}$ be the client's desired computation. Let ρ_C be the Client's input state and $|\psi_C\rangle$ a purification of ρ_C using the distinguisher's register D. Let $C \cup T$ be a trappified pattern obtained from sampling a trappified canvas T from the trappified scheme $\boldsymbol{P}$ using probability distribution $\mathcal{P}$ and embedding computation C into in using the embedding algorithm. We denote $\mathsf{C}(\rho_C) \otimes |\mathsf{Acc}\rangle\langle\mathsf{Acc}|$ and $\mathrm{Tr}_{O_C^c}(C \cup T(\rho_C \otimes \sigma) \otimes |\tau(t)\rangle\langle\tau(t)|)$ the final outputs of the Client in the ideal and real settings, where the trace is over all registers not containing the output of the Client's computation.[3] The distinguishing advantage is defined as:

$$\begin{aligned}\epsilon_{cor} = \Big\| &\mathsf{C} \otimes \mathbb{I}_D(|\psi_C\rangle\langle\psi_C|) \otimes |\mathsf{Acc}\rangle\langle\mathsf{Acc}| - \\ &\tilde{\mathsf{C}}_{T,\mathbb{I}} \otimes \mathbb{I}_D(|\psi_C\rangle\langle\psi_C| \otimes \sigma) \otimes |\tau(t)\rangle\langle\tau(t)|)\Big\|_{\mathrm{Tr}},\end{aligned} \tag{5.8}$$

where $\tilde{\mathsf{C}}_{T,\mathbb{I}} = \mathsf{D}_{O,C} \circ \mathrm{Tr}_{O_C^c} \circ (C \cup T)$. In the honest case, the concrete and ideal settings will output different states only in the case where the protocol wrongly rejects the computation or outputs a wrong result despite the absence of errors.

Since the trappified scheme is δ-insensitive to $\mathbb{I} \in \mathcal{E}_2$, the probability that the decision function outputs Rej is bounded by δ as per Definition 5.2.10. Furthermore, using Lemma 5.2.8, the output of the test is independent of the computation being performed. Combining these two properties yields:

$$\begin{aligned}\epsilon_{cor} \leq \|&\mathsf{C} \otimes \mathbb{I}_D(|\psi_C\rangle\langle\psi_C|) \otimes |\mathsf{Acc}\rangle\langle\mathsf{Acc}| - \\ &\tilde{\mathsf{C}}_{T,\mathbb{I}} \otimes \mathbb{I}_D(|\psi_C\rangle\langle\psi_C| \otimes \sigma)) \otimes (\delta\,|\mathsf{Rej}\rangle\langle\mathsf{Rej}| + (1-\delta)\,|\mathsf{Acc}\rangle\langle\mathsf{Acc}|)\|_{\mathrm{Tr}}.\end{aligned} \tag{5.9}$$

[3] We use here the notation $P(\rho)$ to mean the application of the trappified pattern P to the input state ρ. Also we consider here that the decision function τ outputs either Acc for acceptance or Rej for rejection instead of a binary value.

Using the convexity of the trace distance, we get:

$$\epsilon_{cor} \leq (1-\delta)\|\mathsf{C}\otimes\mathbb{I}_D(|\psi_C\rangle\langle\psi_C|) - \tilde{\mathsf{C}}_{T,\mathbb{I}}\otimes\mathbb{I}_D(|\psi_C\rangle\langle\psi_C|\otimes\sigma))\|_{\mathrm{Tr}} + \delta. \tag{5.10}$$

Finally, the trappified pattern is ν-correct on $\mathbb{I}\in\mathcal{E}_2\subseteq\mathcal{G}_V\setminus\mathcal{E}_1$. Therefore we have that $\|\mathsf{C}\otimes\mathbb{I}_D(|\psi_C\rangle\langle\psi_C|)-\tilde{\mathsf{C}}_{T,\mathbb{I}}\otimes\mathbb{I}_D(|\psi_C\rangle\langle\psi_C|\otimes\sigma))\|_{\mathrm{Tr}}\leq\nu$, meaning that $\epsilon_{cor}\leq(1-\delta)\nu+\delta$. Hence, the protocol is $(\delta+\nu)$-correct. □

Proof of Security against Malicious Server. To prove the security of the protocol, we define as per Definition 2.1.2 a Simulator σ that has access to the Server's interface of the Secure Delegated Quantum Computation Resource. The interaction involving either the Simulator or the real honest Client should be indistinguishable.

Defining the Server's Simulator. To do so, we use again the fact that when the protocol is run and a deviation is applied by the Server, the probability of accepting or rejecting the computation is dependent only on the deviation and not of the computation performed on the non-trap part of the pattern. This is a crucial property as this allows to simulate the behaviour of the concrete protocol even when the computation performed is unknown. More precisely, we define the Simulator in the following way:

Simulator 1

1. The Simulator request a leak from the Secure Delegated Quantum Computation Resource and receives in return $(\mathfrak{C}, G, \boldsymbol{P}, \preceq_G)$.
2. It chooses at random any computation $\mathsf{C}_S\in\mathfrak{C}$ and an input which is compatible with C.
3. It performs the same tasks as those described by the Client's side of the Trappified Delegated Blind Computation Protocol 4.
4. Whenever τ accepts, the Simulator sends $c=0$ to the Secure Delegated Quantum Computation Resource, indicating that the honest Client should receive its output. If it rejects, the Simulator sends $c=1$ Secure Delegated Quantum Computation Resource, indicating an abort.

We now show that the distinguisher cannot tell apart with high probability the simulation and the concrete protocol.

Applying the Pauli Twirl. We first use the twirling lemma to decompose the deviation of the Server. Here we are only concerned with the state representing the interaction of the Client or of the Simulator with the Server. Since the Simulator defined above performs the same tasks as the Client when the Protocol is run, we only need to derive the expression for the Client's interaction. The following steps are similar to the ones in [FKD18, Proof of Theorem 3] and work as can be seen here for the basic UBQC protocol and any protocol based on it.

Let C and ρ_C be the Client's computation and input, let T and σ be the trappified canvas chosen from the trappified scheme $\boldsymbol{P}$ and the associated input. Finally, let $C \cup T$ the trappified pattern resulting from embedding C into T, with base angles $\{\phi(i)\}_{i\in O^c}$.

We start by expressing the state in the simulation and the real protocol. The Server first receives quantum states which are encrypted with $\mathsf{Z}(\theta(i))\mathsf{X}^{a(i)}$ for all vertices $v \in V$. This is explicitly the case for the inputs to the computation and trap patterns, but also for the other qubits of the graph, since we have that $|+_\theta\rangle = \mathsf{Z}(\theta)\,|+\rangle = \mathsf{Z}(\theta)\mathsf{X}^a\,|+\rangle$.[4] Recall that $a_N(i) = \sum_{j\in N_G(i)} a(j)$ and the outputs qubits are only Quantum One-Time-Padded, i.e. $\theta(i) = (r(i) + a_N(v))\pi$ for $i \in O$. Then, omitting the Client's classical registers containing the secret values $\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{r}$, state from the point of view of the Client is noted $\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{in,\boldsymbol{b}+\boldsymbol{r}}$, defined as:

$$\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{in,\boldsymbol{b}+\boldsymbol{r}} = \left(\bigotimes_{i\in V} \mathsf{Z}_i(\theta(i))\mathsf{X}_i^{a(i)}\right)(\rho_C \otimes \sigma \otimes |+\rangle\langle+|^{\otimes|V|-|I|}) \bigotimes_{i\in O^c} |\delta_{\boldsymbol{b}+\boldsymbol{r}}(i)\rangle\langle\delta_{\boldsymbol{b}+\boldsymbol{r}}(i)|\,, \tag{5.11}$$

where $\boldsymbol{b}$ corresponds to the perceived branch of computation based on the outcomes returned by the Server to the Client. The values $\delta_{\boldsymbol{b}+\boldsymbol{r}}(i) = (-1)^{a(i)}\phi'_{\boldsymbol{b}+\boldsymbol{r}}(i) + \theta(i) + (r(i) + a_N(i))\pi$ are each encoded as computational basis states on three qubits from a register R with $3n$ qubits. The angle $\phi'_{\boldsymbol{b}+\boldsymbol{r}}(i)$ is obtained through the formula for $\phi'(i)$ from the UBQC Protocol 2, Equation (2.5), and includes the corrections stemming from $\boldsymbol{b}$ and $\boldsymbol{r}$, while $a_N(i)$ compensates the effect of the X encryption from a qubit on its neighbours. While this seems that the Client is sending the values of $\delta_{\boldsymbol{b}+\boldsymbol{r}}(i)$ at the beginning breaks the causal structure of the protocol, these states will indeed not be affected by any operations before they can actually be correctly computed by the Client. This will be made formal below. Finally, note that for simplicity, the qubits in the state above are not grouped in the order in which the Client sends.

[4] In the real protocol, this value is always 0. This is perfectly indistinguishable since the distribution of the values of δ are identical regardless of this choice of parameter for non-input qubits and correctness is unaffected.

We consider a purification $|\psi_S\rangle\langle\psi_S|$ of the Server's input ρ_S. Let F_{in} be a unitary such that $|\psi_S\rangle = \mathsf{F}_{in}|0\rangle^{\otimes w}$ for the appropriate work register size w. Then the operations which the Server applies before any measurement can be written as unitaries acting on all qubits which have not yet been measured and the available values of $\delta_{\boldsymbol{b}+\boldsymbol{r}}(i)$. These can be then decomposed into the correct unitary operation followed by a unitary attack of the Server's choice. The Server receives all qubits, applies the entanglement operation corresponding to the Client's desired graph, then a unitary attack F_G, then the correct rotation on the first measured qubits, followed by another attack F_1. These two last steps are repeated for all measured qubits.

The entanglement according to the graph $G = (V, E)$ is noted $\mathsf{G} = \bigotimes_{(i,j)\in E} \mathsf{CZ}_{i,j}$. The Z-axis rotations required for performing the measurement in the basis defined by $\delta_{\boldsymbol{b}+\boldsymbol{r}}(i)$ are represented by unitaries CR_v, controlled rotations around the Z-axis with the control being performed by the registers containing the corresponding value of $\delta_{\boldsymbol{b}+\boldsymbol{r}}(i)$. Figure 5.6 shows one possible implementation of this controlled operation.

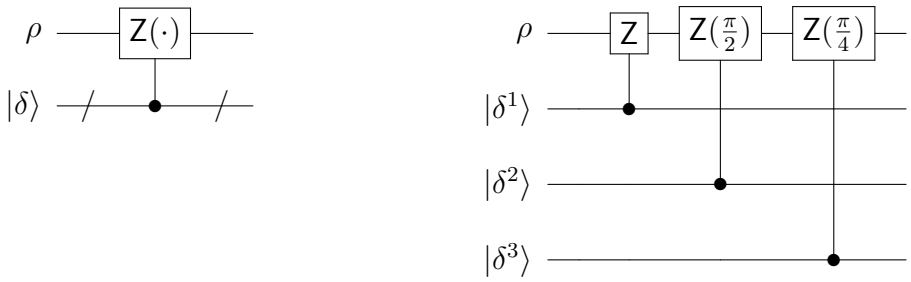


Figure 5.6: Controlled rotation used to unitarise Protocol 2. The right hand side is a possible implementation of the rotation on the left, where δ^i are the bits composing the value δ. The 3 controlling qubits are sent by the client to the server in the computational basis as they correspond to classical values.

The quantum state representing the interaction between the Client and Server implementing the protocol just before the measurements are performed, noted $\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{pre,\boldsymbol{b}+\boldsymbol{r}}$, is thus:

$$\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{pre,\boldsymbol{b}+\boldsymbol{r}} = \mathsf{F}_n\mathsf{CR}_n^\dagger \dots \mathsf{F}_1\mathsf{CR}_1^\dagger\mathsf{F}_G\mathsf{F}_{in}\mathsf{G}(\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{in,\boldsymbol{b}+\boldsymbol{r}} \otimes |0\rangle\langle 0|^{\otimes w}), \tag{5.12}$$

with $|O^c| = n$. We can move all deviations through the controlled rotations and regroup them as F'.[5] Then, it is possible to replace the (classically) controlled rotations corresponding to the honest execution of the protocol by ordinary rotations $\mathsf{Z}(\delta_{\boldsymbol{b}+\boldsymbol{r}}(i))^\dagger$,

[5]Formally, we have $\mathsf{F}' = \mathsf{F}_n\mathsf{CR}_n^\dagger \dots \mathsf{F}_1\mathsf{CR}_1^\dagger\mathsf{F}_G\mathsf{F}_{in} \circ \bigotimes_{i\in O^c} \mathsf{CR}_i$.

thus yielding:[6]

$$\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{pre,\boldsymbol{b}+\boldsymbol{r}} = \mathsf{F}'\left(\bigotimes_{i\in O^c} \mathsf{Z}_i(\delta_{\boldsymbol{b}+\boldsymbol{r}}(i))^\dagger\right)\mathsf{G}(\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{in,\boldsymbol{b}} \otimes |0\rangle\langle 0|^{\otimes w}). \tag{5.13}$$

We now apply the decryption operations performed by the Client on the output layer qubits after the Server has returned these qubits at the end of the protocol. The resulting state, noted $\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{dec,\boldsymbol{b}+\boldsymbol{r}}$, can be written as:

$$\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{dec,\boldsymbol{b}+\boldsymbol{r}} = \left(\bigotimes_{i\in O} \mathsf{Z}_i^{s_Z(i)+r(i)}\mathsf{X}_i^{s_X(i)+a(i)}\right)(\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{pre,\boldsymbol{b}+\boldsymbol{r}}), \tag{5.14}$$

where $s_X(i)$ and $s_Z(i)$ stem from the flow of the trappified pattern $C \cup T$. To finish, we enforce that the computation branch is effectively $\boldsymbol{b}$ by projecting all non-output qubits $i \in O^c$ onto $\mathsf{Z}_i^{b(i)+r(i)}|+\rangle\langle+|\mathsf{Z}_i^{b(i)}$.[7,8] Since $|+\rangle$ is a $+1$ eigenstate of X, this is equivalent to projecting onto $\mathsf{Z}_i^{b(i)+r(i)}|+\rangle\langle+|_i\mathsf{X}_i^{a(i)}\mathsf{Z}^{b(i)}$. The final state, noted $\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{out,\boldsymbol{b}+\boldsymbol{r}}$, is therefore:

$$\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{out,\boldsymbol{b}+\boldsymbol{r}} = \left(\bigotimes_{i\in O^c} \mathsf{Z}_i^{b(i)+r(i)}|+\rangle\langle+|_i\right)\left(\bigotimes_{i\in O^c}\mathsf{X}_i^{a(i)}\mathsf{Z}_i^{b(i)}\right)(\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{dec,\boldsymbol{b}+\boldsymbol{r}}). \tag{5.15}$$

We then apply the change of variable $b'(i) = b(i) + r(i)$ and then relabelling $b'(i)$ into $b(i)$. This has the effect of removing the influence of $r(i)$ in the corrected measurement angles, transforming $\phi'_{\boldsymbol{b}+\boldsymbol{r}}(i)$ into $\phi'_{\boldsymbol{b}}(i)$:[9]

$$\begin{aligned}\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{out,\boldsymbol{b}} &= \left(\bigotimes_{i\in O^c} \mathsf{Z}_i^{b(i)}|+\rangle\langle+|_i\right)\left(\bigotimes_{i\in O^c}\mathsf{X}_i^{a(i)}\mathsf{Z}_i^{b(i)+r(i)}\right)(\rho^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}_{dec,\boldsymbol{b}}) \\ &= \mathsf{P}_{\boldsymbol{b}}\tilde{\mathsf{U}}_P\left(\rho_C \otimes \sigma \otimes |+\rangle\langle+|^{\otimes|V|-|I|}\bigotimes_{i\in O^c}|\delta_{\boldsymbol{b}}(i)\rangle\langle\delta_{\boldsymbol{b}}(i)| \otimes |0\rangle\langle 0|^{\otimes w}\right).\end{aligned} \tag{5.16}$$

[6] If these operations were replaced before, the deviations would pick up a dependency on $\delta_{\boldsymbol{b}+\boldsymbol{r}}(i)$ during the commutation.

[7] These qubits can be assumed to be measured without loss of generality since (i) the Server needs to produce the values $\boldsymbol{b}$ using its internal state and the values received from the Client and (ii) the operation F' is fully general, meaning that the Server can use it to reorder the qubits before the measurement if it so desires.

[8] The difference in coefficients takes into account the corrections which the Client applies to the outputs of the measurements to account for $r(i)$.

[9] This value uses the formula for $\phi'(i)$ from the Delegated MBQC Protocol 1, Equation (2.3).

where we defined $\mathsf{P}_{\boldsymbol{b}}$ as:

$$\mathsf{P}_{\boldsymbol{b}} = \bigotimes_{i \in O^c} \mathsf{Z}_i^{b(i)} \left|+\right\rangle\!\left\langle+\right|_i , \tag{5.17}$$

and $\tilde{\mathsf{U}}_P$ as:

$$\begin{aligned} \tilde{\mathsf{U}}_P = &\left(\bigotimes_{i \in O^c} \mathsf{X}_i^{a(i)} \mathsf{Z}_i^{b(i)+r(i)} \right) \left(\bigotimes_{i \in O} \mathsf{Z}_i^{s_Z(i)+r(i)} \mathsf{X}_i^{s_X(i)+a(i)} \right) \mathsf{F}' \circ \\ &\left(\bigotimes_{i \in O^c} \mathsf{Z}_i(\delta_{\boldsymbol{b}}(i))^\dagger \right) \mathsf{G} \left(\bigotimes_{i \in V} \mathsf{Z}_i(\theta(i)) \mathsf{X}_i^{a(i)} \right). \end{aligned} \tag{5.18}$$

We now look at the state from the point of view of the Server, noted $\rho_{out,\boldsymbol{b}}$, which can be written as follows considering that in this case the secret parameters are unknown:

$$\rho_{out,\boldsymbol{b}} = \frac{1}{8^{|O^c|} \cdot 4^{|V|}} \sum_{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}} \rho_{out,\boldsymbol{b}}^{\boldsymbol{\theta},\boldsymbol{a},\boldsymbol{r}}. \tag{5.19}$$

We focus on the state before the projection $\mathsf{P}_{\boldsymbol{b}}$ is applied. The goal is to remove dependencies on $r(i), a(i)$ which appear outside the encryption and decryption procedures in order to be able to use the twirling lemma, using the fact that these parameters are chosen at random.[10] To this end we cancel out the values of $\theta(i)$ coming from the initial encryption with those which appear in the rotations by $\delta_{\boldsymbol{b}}(i) = (-1)^{a(i)}\phi'_{\boldsymbol{b}}(i) + \theta(i) + (r(i) + a_N(i))\pi$ for $i \in O^c$:

$$\mathsf{Z}_i(\delta_{\boldsymbol{b}}(i))^\dagger \mathsf{G} \mathsf{Z}_i(\theta(i)) \mathsf{X}_i^{a(i)} = \mathsf{Z}_i((-1)^{a(i)}\phi'_{\boldsymbol{b}}(i) + (r(i) + a_N(i))\pi)^\dagger \mathsf{G} \mathsf{X}_i^{a(i)}, \tag{5.20}$$

due to the fact that the entanglement operation consists of CZ operations through which the Z rotations commute. Now, the values $\theta(i)$ appear only through the definition of the angles $\delta_{\boldsymbol{b}}(i)$. Hence, they perfectly One-Time-Pad these angles and summing over $\theta(i)$ yields the perfectly mixed state in the register R. Formally:

$$\rho_{out,\boldsymbol{b}} = \frac{1}{4^{|V|}} \sum_{\boldsymbol{a},\boldsymbol{r}} \mathsf{P}_{\boldsymbol{b}} \tilde{\mathsf{U}}_P \left(\rho_C \otimes \sigma \otimes \left|+\right\rangle\!\left\langle+\right|^{\otimes |V|-|I|} \otimes \mathbb{1}_{3n} \otimes \left|0\right\rangle\!\left\langle0\right|^{\otimes w} \right), \tag{5.21}$$

where $\mathbb{1}_{3n}$ is the perfectly mixed state over the $3n$ qubits of R. This register has thus no effect on either the computation or the traps and is in tensor product with the rest

[10] These paramters must be perfectly random as using them multiple times might introduce correlations which the Server can exploit to derandomise the Pauli twirl.

of the state, it can therefore be traced out by assuming without loss of generality that the Server's deviation has no effect on it.

We can now commute the encryption on both sides of the deviation so that the deviation is exactly sandwiched between two identical random Pauli operations. We start on the right side of F' in the expression of $\tilde{\mathsf{U}}_P$. For all qubits in the graph, we need to commute $\mathsf{X}^{a(i)}$ through the entanglement operation first. Since $\mathsf{CZ}_{i,j}\mathsf{X}_i = \mathsf{Z}_j\mathsf{X}_i\mathsf{CZ}_{i,j}$ (and similarly for X_j), using $a_N(i) = \sum_{j\in N_G(i)} a(j)$ we get that:

$$\mathsf{G}\left(\bigotimes_{i\in V}\mathsf{X}_i^{a(i)}\right) = \left(\bigotimes_{i\in V}\mathsf{Z}_i^{a_N(i)}\mathsf{X}_i^{a(i)}\right)\mathsf{G}. \tag{5.22}$$

The additional $\mathsf{Z}_i^{r(i)+a_N(i)}$ encryption of the output qubits commute unchanged through the entanglement operation G. These encryptions now need to be commuted through the Z rotations for measured qubits:

$$\mathsf{Z}_i((-1)^{a(i)}\phi'_{\boldsymbol{b}}(i) + (r(i)+a_N(i))\pi)^\dagger\mathsf{Z}_i^{a_N(i)}\mathsf{X}_i^{a(i)} = \mathsf{Z}_i^{r(i)}\mathsf{X}_i^{a(i)}\mathsf{Z}_i(\phi'_{\boldsymbol{b}}(i))^\dagger \tag{5.23}$$

On the other hand, on the output qubits, the operation applied is $\mathsf{Z}_i^{a_N(i)}\mathsf{Z}_i^{r(i)+a_N(i)}\mathsf{X}_i^{a(i)} = \mathsf{Z}_i^{r(i)}\mathsf{X}_i^{a(i)}$. In total, we have that:

$$\left(\bigotimes_{i\in O^c}\mathsf{Z}_i(\delta_{\boldsymbol{b}}(i))^\dagger\right)\mathsf{G}\left(\bigotimes_{i\in V}\mathsf{Z}_i(\theta(i))\mathsf{X}_i^{a(i)}\right) = \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}\left(\bigotimes_{i\in O^c}\mathsf{Z}_i(\phi'_{\boldsymbol{b}}(i))^\dagger\right)\mathsf{G} \tag{5.24}$$

where $\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} = \bigotimes_{i\in V}\mathsf{Z}_i^{r(i)}\mathsf{X}_i^{a(i)}$. On the other side of F' in the expression of $\tilde{\mathsf{U}}_P$, we simply have that:

$$\left(\bigotimes_{i\in O^c}\mathsf{X}_i^{a(i)}\mathsf{Z}_i^{b(i)+r(i)}\right)\left(\bigotimes_{i\in O}\mathsf{Z}_i^{s_Z(i)+r(i)}\mathsf{X}_i^{s_X(i)+a(i)}\right) = \left(\bigotimes_{i\in O^c}\mathsf{Z}_i^{b(i)}\right)\left(\bigotimes_{i\in O}\mathsf{Z}_i^{s_Z(i)}\mathsf{X}_i^{s_X(i)}\right)\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger, \tag{5.25}$$

up to a global phase.

We note $\rho_{cor,\boldsymbol{b}} = \left(\bigotimes_{i\in O^c}\mathsf{Z}_i(\phi'_{\boldsymbol{b}}(i))^\dagger\right)\mathsf{G}(\rho_C\otimes\sigma\otimes|+\rangle\langle+|^{\otimes|V|-|I|})$ the correct state before the encryption-deviation-decryption, and define $\mathsf{D}_{\boldsymbol{b}} = \left(\bigotimes_{i\in O^c}\mathsf{Z}_i^{b(i)}\right)\left(\bigotimes_{i\in O}\mathsf{Z}_i^{s_Z(i)}\mathsf{X}_i^{s_X(i)}\right)$ as the measurement outcome and final plain MBQC correction operator. We then

obtain:

$$\rho_{out,\boldsymbol{b}} = \frac{1}{4^{|V|}} \mathsf{P}_{\boldsymbol{b}} \mathsf{D}_{\boldsymbol{b}} \sum_{\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \in \mathcal{G}_V} (\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \otimes \mathbb{I}_w) \mathsf{F}' (\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \otimes \mathbb{I}_w) (\rho_{cor,\boldsymbol{b}} \otimes |0\rangle\langle 0|^{\otimes w}). \tag{5.26}$$

Without loss of generality we can decompose the Server's deviation in the Pauli operator basis over the graph's vertices as $\mathsf{F}' = \sum_{\mathsf{E} \in \mathcal{G}_V} \alpha_E \mathsf{E} \otimes \mathsf{U}_\mathsf{E}$. Applying the notation $\mathsf{U}(\rho) = \mathsf{U}\rho\mathsf{U}^\dagger$, we get:

$$\rho_{out,\boldsymbol{b}} = \frac{1}{4^{|V|}} \mathsf{P}_{\boldsymbol{b}} \mathsf{D}_{\boldsymbol{b}} \sum_{\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \in \mathcal{G}_V} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{F}' \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \rho_{cor,\boldsymbol{b}} \otimes |0\rangle\langle 0|^{\otimes w} \, \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{F}'^\dagger \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \tag{5.27}$$

$$= \frac{1}{4^{|V|}} \mathsf{P}_{\boldsymbol{b}} \mathsf{D}_{\boldsymbol{b}} \sum_{\mathsf{E},\mathsf{E}' \in \mathcal{G}_V} \alpha_\mathsf{E} \alpha_{\mathsf{E}'}^* \sum_{\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \in \mathcal{G}_V} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{E} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \rho_{cor,\boldsymbol{b}} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{E}'^\dagger \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \otimes \mathsf{U}_\mathsf{E} |0\rangle\langle 0|^{\otimes w} \, \mathsf{U}_{\mathsf{E}'}^\dagger, \tag{5.28}$$

where $\alpha_{\mathsf{E}'}^*$ is the complex conjugate of $\alpha_{\mathsf{E}'}$. We now apply the Twirling Lemma 2.3.1, leading to $\sum_{\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \in \mathcal{G}_V} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{E} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \rho_{cor,\boldsymbol{b}} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{E}'^\dagger \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} = 0$ for $\mathsf{E} \neq \mathsf{E}'$. Therefore:

$$\rho_{out,\boldsymbol{b}} = \frac{1}{4^{|V|}} \mathsf{P}_{\boldsymbol{b}} \mathsf{D}_{\boldsymbol{b}} \sum_{\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}, \mathsf{E} \in \mathcal{G}_V} |\alpha_\mathsf{E}|^2 \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{E} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \rho_{cor,\boldsymbol{b}} \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}^\dagger \mathsf{E}^\dagger \mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}} \otimes \mathsf{U}_\mathsf{E} |0\rangle\langle 0|^{\otimes w} \, \mathsf{U}_\mathsf{E}^\dagger, \tag{5.29}$$

The result is a CPTP map defined by $\{\mathsf{E} \otimes \mathsf{U}_\mathsf{E}, p_\mathsf{E} = |\alpha_\mathsf{E}|^2\}_{\mathsf{E} \in \mathcal{G}_V}$, a convex combination of Paulis on the graph's vertices tensored with an operation on the Server's internal register. Overall, this shows that the effect of the Server's deviation is – when averaged over the choice of secret parameters – a probabilistic mixture of Pauli operators on the qubits of the graph.

The Pauli encryption and decryption $\mathsf{Q}_{\boldsymbol{a},\boldsymbol{r}}$ commutes up to a global phase with the Pauli deviation E. We can therefore rewrite the state as:

$$\rho_{out,\boldsymbol{b}} = \mathsf{P}_{\boldsymbol{b}} \mathsf{D}_{\boldsymbol{b}} \sum_{\mathsf{E} \in \mathcal{G}_V} p_\mathsf{E} \mathsf{E}(\rho_{cor,\boldsymbol{b}}) \otimes \mathsf{U}_\mathsf{E}(|0\rangle\langle 0|^{\otimes w}) \tag{5.30}$$

Since the distinguisher wishes to maximise its distinguishing probability, it is sufficient to consider that it applies a fixed Pauli deviation $\mathsf{E} \in \mathcal{G}_V$ for which the distinguishing probability is maximal. Furthermore, the state in the Server's register is unentangled from the rest and therefore does not contribute to the attack of the Server on the Client's state. Once this is traced out, seeing as D_b and E are Paulis and therefore commute up

to a global phase, the final state can be written as:

$$\rho_{out,\boldsymbol{b}} = \mathsf{P}_{\boldsymbol{b}}\mathsf{ED}_{\boldsymbol{b}}(\rho_{cor,\boldsymbol{b}}) = \mathsf{E} \circ (C \cup T)(\rho_C \otimes \sigma). \tag{5.31}$$

The final equality stems from the definition of the notation $\mathsf{E} \circ P$ for a pattern P (Section 5.2.1) and the fact that applying $\mathsf{D}_{\boldsymbol{b}}$ to $\rho_{cor,\boldsymbol{b}}$ performs exactly the correct unitary portion of plain MBQC pattern $C \cup T$ – up to the measurements.[11]

Applying the Composable Security of UBQC. We next show that this deviation depends on the same classical parameters in the ideal and real scenarii. To this end, we apply the composition Theorem 2.1.4 of the AC framework to replace the execution of the UBQC Protocol by the Blind Delegated Quantum Computation Resource 1 both in the simulation and the real protocol. As per the security of the UBQC Protocol as expressed in Theorem 2.3.2, the distinguishing advantage is not modified by this substitution so long as the graph, order of measurements and output set of qubits are known to the Server. The results can be seen in Figures 5.7 and 5.8. The distinguisher has access to all outward interfaces.

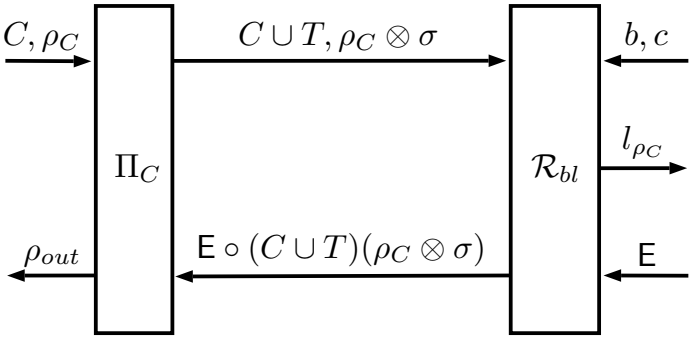


Figure 5.7: Real world hybrid interaction between the Client's protocol CPTP map Π_C and Blind Delegated QC Resource $\mathcal{R}_{bl}$.

The Simulator receives the leak $l_{\rho_C} = (\mathfrak{C}, G, \boldsymbol{P}, \preceq_G)$ from the Secure Delegated Quantum Computation Resource. In both cases, we assume that both the Client and Simulator send $(\mathfrak{C}, G, \boldsymbol{P}, \preceq_G)$ as a first message to the Server. All canvases in $\boldsymbol{P}$ are blind-compatible (Definition 5.2.6) meaning that they all share the graph G and the same output set O, and the order $\preceq_G$ is used for all patterns generated from $\boldsymbol{P}$ and the

[11] This is correct up to a relabelling of $\mathcal{G}_V$ since in the rest of the chapter we assumed that the measurements are performed in the computational basis.

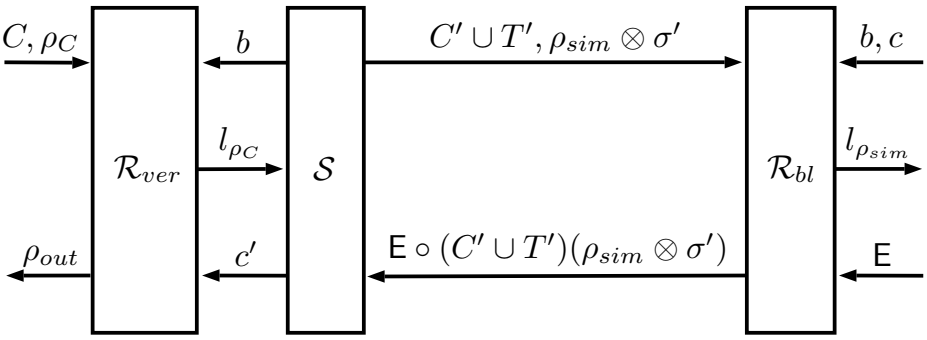


Figure 5.8: Simulator $\mathcal{S}$ interacting with the Secure and Blind DQC Resources $\mathcal{R}_{ver}$ and $\mathcal{R}_{bl}$.

embedding algorithm. Since these parameters are the same in all executions of both the real and ideal case, the leak $l_{\rho_{ideal}}$ obtained by the Server in the simulation does not yield any more information. Overall, the classical information in both cases is identical and does not help the distinguisher on its own.

Output and Abort Probability Analysis. The interactions are therefore indistinguishable before the output is sent back to the Client and we focus in the following on the output state and the abort probability in both cases, which are the only remaining elements which the distinguisher can use to decide which setup it is interacting with. The outputs $\rho^{out}_{real}, \rho^{out}_{ideal}$ of the Client in respectively the real and ideal settings can be written as follows for Pauli deviation $\mathsf{E} \in \mathcal{G}_V$ introduced by the distinguisher:

$$\rho^{out}_{real} = p^{\mathsf{Acc}}_{real} \tilde{\mathsf{C}}_{T,\mathsf{E}}(\rho_C \otimes \sigma) \otimes |\mathsf{Acc}\rangle\langle\mathsf{Acc}| + \left(1 - p^{\mathsf{Acc}}_{real}\right) |\perp\rangle\langle\perp| \otimes |\mathsf{Rej}\rangle\langle\mathsf{Rej}|, \tag{5.32}$$

$$\rho^{out}_{ideal} = p^{\mathsf{Acc}}_{ideal} \mathsf{C}(\rho_C) \otimes |\mathsf{Acc}\rangle\langle\mathsf{Acc}| + \left(1 - p^{\mathsf{Acc}}_{ideal}\right) |\perp\rangle\langle\perp| \otimes |\mathsf{Rej}\rangle\langle\mathsf{Rej}|, \tag{5.33}$$

where $\tilde{\mathsf{C}}_{T,\mathsf{E}} = \mathsf{D}_{O,C} \circ \mathrm{Tr}_{O^c_C} \circ \mathsf{E} \circ (C \cup T)$ and for all $\mathsf{C} \in \mathfrak{C}$ we have that:

$$p^{\mathsf{Acc}}_{\mathsf{C}} = \sum_{T \in \boldsymbol{P}} \Pr_{T \sim \mathcal{P}} \left[\tau\left(\mathrm{Tr}_{O^c_T}\left(\mathsf{E} \circ (C \cup T)(\rho_C \otimes \sigma)\right)\right) = 0, T\right] \tag{5.34}$$

$$= \sum_{T \in \boldsymbol{P}} \Pr_{\substack{T \sim \mathcal{P} \\ t \sim \mathsf{E} \circ \mathcal{T}}} [\tau(t) = 0, T], \tag{5.35}$$

which uses Lemma 5.2.8, namely that the acceptance probability does not depend on the input or the computation. Therefore this probability is identical in the real and ideal setting $p^{\mathsf{Acc}}_{real} = p^{\mathsf{Acc}}_{ideal} = p_{\mathsf{Acc}}$, regardless of the deviation chosen by the distinguisher.

We see that whenever the computation is rejected, the output state is identical in both setups. On the other hand, whenever the computation is accepted, the ideal resource will always output the correct state, while the concrete protocol outputs a potentially erroneous state. By convexity of the trace distance, the distinguishing probability p_d can therefore be written as:

$$p_d = \max_{\substack{\mathsf{E}\in\mathcal{G}_V \\ \mathsf{C}\in\mathfrak{C} \\ \psi_C}} \left(p_{\mathsf{Acc}} \times \|(\tilde{\mathsf{C}}_{T,\mathsf{E}} - \mathsf{C}\otimes\mathbb{I}_T)\otimes\mathbb{I}_D(|\psi_C\rangle\langle\psi_C|\otimes\sigma)\|_{\mathrm{Tr}} \right), \tag{5.36}$$

where $|\psi_C\rangle$ is a purification of the Client's input ρ_C using the distinguisher's register D. We now therefore analyse the output state in the case where the computation is accepted.

Error Influence on Distinguishing Probability. First consider the case where $\mathsf{E} \in \mathcal{E}_1$. Since $\boldsymbol{P}$ ϵ-detects such errors (Definition 5.2.9), the probability of accepting is upper-bounded by ϵ, which implies:

$$p_{d,\mathcal{E}_1} \leq \epsilon \times \max_{\substack{\mathsf{E}\in\mathcal{E}_1 \\ \mathsf{C}\in\mathfrak{C} \\ \psi_C}} \left(\|(\tilde{\mathsf{C}}_{T,\mathsf{E}} - \mathsf{C}\otimes\mathbb{I}_T)\otimes\mathbb{I}_D(|\psi_C\rangle\langle\psi_C|\otimes\sigma)\|_{\mathrm{Tr}} \right). \tag{5.37}$$

The distinguisher can freely choose the Client's input state ψ_C and the computation $\mathsf{C} \in \mathfrak{C}$ and there is no constraint on the effect of this deviation on the computation part of the trappified pattern. In the worst case the incorrect real output state is orthogonal to the ideal output state, meaning that the distinguisher can tell apart both settings with certainty and the trace distance is upper-bounded by 1. The distinguishing probability in this scenario therefore follows $p_{d,\mathcal{E}_1} \leq \epsilon$.

Second, we consider the alternate case, where $\mathsf{E} \notin \mathcal{E}_1$. Here, we assumed that the trappified scheme $\boldsymbol{P}$ is ν-correct on the set $\mathcal{G}_V \setminus \mathcal{E}_1$ (Definition 5.2.11), therefore the trace distance between the correct result of the computation and the real output of the protocol is upper-bounded by ν:

$$\|(\tilde{\mathsf{C}}_{T,\mathsf{E}} - \mathsf{C})\otimes\mathbb{I}_D(|\psi_C\rangle\langle\psi_C|\otimes\sigma)\|_{\mathrm{Tr}} \leq \nu. \tag{5.38}$$

Therefore:

$$p_{d,\mathcal{G}_V\setminus\mathcal{E}_1} \leq \nu \times \max_{\mathsf{E}\in\mathcal{G}_V\setminus\mathcal{E}_1}(p_{\mathsf{Acc}}), \tag{5.39}$$

where the maximisation is done only over the error since the acceptance probability is independent of the input and computation. In this case, the accepting probability p_{Acc} is not constrained and hence only upper bounded by 1, yielding $p_{d,\mathcal{G}_V\setminus\mathcal{E}_1} \leq \nu$.

Since the deviation chosen by the distinguisher falls in either of these two cases, we have $p_d = \max(p_{d,\mathcal{E}_1}, p_{d,\mathcal{G}_V\setminus\mathcal{E}_1})$ and the maximum distinguishing probability between the Resource together with the Simulator and the concrete Protocol is thus upper-bounded by $\max(\epsilon, \nu)$. □

Remark 5.3.2 (Using Other Blind Protocols.)**.** *In this work we use the UBQC protocol to provide blindness. This protocol is based on the* prepare-and-send *principle. The direct mirror situation, where the Server prepares states and sends them to the Client, is called the* receive-and-measure *paradigm. These are also based on MBQC and were shown to be equivalent to prepare-and-send protocol by [WEP22] using the Abstract Cryptography framework. Our techniques are therefore directly applicable to this setting as well with the same security guarantees. These two setups together cover most protocols that have been designed and which may be implemented in the near future.*

The work of [Mah18b] introduced an explicit protocol for verifying BQP *computations by relying only on classical interactions and a computational hardness assumption. Our techniques are fully applicable as well using a protocol which* ϵ_{bl}-computationally-constructs *the Blind Delegated Quantum Computation Resource 1 in the AC framework and is capable of implementing MBQC computations natively. The resulting protocol is of course computationally-secure only. A simple hybrid argument can be used first to replace any such computationally-secure protocol with Resource 1 first – at a cost of* ϵ_{bl} *– and then the UBQC protocol at no cost. The other steps of the proof remain unchanged.*

5.3.2 Insensitivity Implies Noise-Robustness

Then, we give conditions on protocols implementing SDQC so that they are able to run on noisy machines with a good acceptance probability. We show formally the following intuitive reasoning: if the errors to which the trappified scheme is insensitive do not disturb the computation too much, then a machine which mostly suffers from such errors will almost always lead to the client accepting the computation and the output will be

close to perfect.

Theorem 5.3.3 (Robust Detection Implies Robust Verifiability)**.** *Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two sets of Pauli deviations such that $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$ and $\mathbb{I} \in \mathcal{E}_2$. Let $\boldsymbol{P}$ be a trappified scheme for computation set $\mathfrak{C}$, which is δ-insensitive to $\mathcal{E}_2$ and ν-correct on $\mathcal{G}_V \setminus \mathcal{E}_1$. Let $C \cup T$ be a trappified pattern resulting from embedding computation $\mathsf{C} \in \mathfrak{C}$ in trappified canvas T sampled from $\boldsymbol{P}$. We assume an execution of Protocol 4 with an honest-but-noisy Server whose noise is modelled by sampling an error $\mathsf{E} \in \mathcal{E}_2$ with probability $(1-p_2)$ and $\mathsf{E} \in \mathcal{G}_V \setminus \mathcal{E}_2$ with probability p_2. Then, the Client in Protocol 4 accepts with probability at least $(1-p_2)(1-\delta)$, and if accepted the distance between the implemented transformation and the client's computation is bounded as follows:*

$$\forall \mathsf{C} \in \mathfrak{C},\ \max_{\psi} \|(\tilde{\mathsf{C}}_{T,\mathsf{E}} - \mathsf{C}) \otimes \mathbb{I}_R(|\psi_C\rangle\langle\psi_C| \otimes \sigma)\|_{\mathrm{Tr}} \leq \nu + p_2 + \delta, \tag{5.40}$$

where $|\psi_C\rangle$ is a purification of the Client's input ρ_C using auxiliary quantum register R, and $\tilde{\mathsf{C}}_{T,\mathsf{E}} = \mathsf{D}_{O,C} \circ \mathrm{Tr}_{O_C^c} \circ \mathsf{E} \circ (C \cup T)$.

Proof. By construction, $\boldsymbol{P}$ is δ-insensitive to $\mathcal{E}_2$. Hence, it will accept deviations in $\mathcal{E}_2$ with probability at least $1-\delta$ which yields the overall lower bound on the acceptance probability of $(1-p_2)(1-\delta)$.

There are then two cases when the computation is accepted. If the deviation E is in $\mathcal{E}_2 \subseteq \mathcal{G}_V \setminus \mathcal{E}_1$, we have by definition:

$$\forall \mathsf{C} \in \mathfrak{C},\ \max_{\psi} \|(\tilde{\mathsf{C}}_{T,\mathsf{E}} - \mathsf{C}) \otimes \mathbb{I}_R(|\psi\rangle\langle\psi| \otimes \sigma)\|_{\mathrm{Tr}} \leq \nu. \tag{5.41}$$

Otherwise, if the deviation is not in $\mathcal{E}_2$ but is accepted, the distance can always be bounded by 1.

The first case happens with probability at least $(1-p_2)(1-\delta)$, since according to Bayes' theorem:

$$\Pr[\mathsf{E} \in \mathcal{E}_2 \,|\, \mathsf{Acc}] = \frac{\Pr[\mathsf{Acc} \,|\, \mathsf{E} \in \mathcal{E}_2] \cdot \Pr[\mathsf{E} \in \mathcal{E}_2]}{\Pr[\mathsf{Acc}]} \geq (1-p_2)(1-\delta). \tag{5.42}$$

Consequently, for the second case it holds then that:

$$\Pr[\mathsf{E} \notin \mathcal{E}_2 \,|\, \mathsf{Acc}] = 1 - \Pr[\mathsf{E} \in \mathcal{E}_2 \,|\, \mathsf{Acc}] \leq 1 - (1-p_2)(1-\delta) = p_2 + \delta - p_2\delta. \tag{5.43}$$

Using the convexity of the trace norm, for the case of an accepting run of the protocol

we finally arrive at

$$\forall \mathsf{C} \in \mathfrak{C},\ \max_{\psi} \|(\tilde{\mathsf{C}}_{T,\mathsf{E}} - \mathsf{C}) \otimes \mathbb{I}_R(|\psi\rangle\langle\psi| \otimes \sigma)\|_{\mathrm{Tr}} \tag{5.44}$$

$$\leq \nu \Pr\left[\mathsf{E} \in \mathcal{E}_2 \,|\, \mathsf{Acc}\right] + 1 - \Pr\left[\mathsf{E} \in \mathcal{E}_2 \,|\, \mathsf{Acc}\right] = 1 - (1-\nu)\Pr\left[\mathsf{E} \in \mathcal{E}_2 \,|\, \mathsf{Acc}\right] \tag{5.45}$$

$$\leq 1 - (1-\nu)(1-p_2)(1-\delta) = \nu(1-p_2)(1-\delta) + p_2 + \delta - p_2\delta, \tag{5.46}$$

which concludes the proof. □

This Theorem shows that whenever (i) a noise process generates deviations that are within $\mathcal{E}_2$ with overwhelming probability, (ii) the embedding of the computation C within $\boldsymbol{P}$ adds redundancy in such a way that ν is negligible, and (iii) $\boldsymbol{P}$ is δ-insensitive to $\mathcal{E}_2$ for a negligible δ, then the protocol will accept the computation almost all the time, and the computation will be very close to C. We will see in the next section how these parameters can be amplified. The theorem above shows the importance not only of the parameters of the scheme, but also the size of the sets $\mathcal{E}_1$ and $\mathcal{E}_2$. By creating schemes which have more errors fall in set $\mathcal{E}_2$, it is possible to have a direct impact both in terms of acceptance probability and fidelity in the context of honest-but-noisy executions. We now show that this error-correction is not only necessary for the noise-robustness of the protocol but also its efficiency.

5.3.3 Efficient Verifiability Requires Error-Correction.

We now present an important consequence of Theorem 5.3.1 in the case where the correctness error $(\delta + \nu)$ and the security error $\max(\epsilon, \nu)$ are negligible with respect to a security parameter λ. We show that this correctness and security regime can only be achieved with a polynomial qubit overhead if the computation is error-protected.

More precisely, we denote $\boldsymbol{P}(\lambda)$ a sequence of trappified schemes indexed by a security parameter λ, such that it $\epsilon(\lambda)$-detects a set $\mathcal{E}_1(\lambda) \subseteq \mathcal{G}_V(\lambda)$ of Pauli deviations, is $\nu(\lambda)$-correct outside $\mathcal{E}_1$, and is $\delta(\lambda)$-insensitive to $\mathcal{E}_2(\lambda) \subseteq \mathcal{G}_V(\lambda) \setminus \mathcal{E}_1(\lambda)$, for $\epsilon(\lambda)$, $\nu(\lambda)$ and $\delta(\lambda)$ negligible in λ. Additionally, let C be a computation pattern which implements the client's desired computation CPTP map $\mathsf{C} \in \mathfrak{C}$ on some input state $|\psi\rangle$.

We are now interested in the server's memory overhead introduced by implementing C using $\boldsymbol{P}(\lambda)$ for computation class $\mathfrak{C}$ instead of the unprotected pattern C. This is expressed by the ratio $|G_{\boldsymbol{P}(\lambda)}|/|G_C|$ between the number of vertices in the graph $G_{\boldsymbol{P}(\lambda)}$ common to all canvases in $\boldsymbol{P}(\lambda)$ and the graph G_C used by the pattern C.

For a trappified pattern $C \cup T$ obtained by using the embedding algorithm on a trappified canvas from $\boldsymbol{P}(\lambda)$ we denote by $|O_{C\cup T}|$ the number of computation output qubits in $C \cup T$. Similarly, $|O_C|$ is the number of output qubits in C. Without loss of generality, we impose that $|O_C|$ is minimal, in the sense that given the set of possible inputs and C, the space spanned by all possible outputs is the whole Hilbert space of dimension $2^{|O_C|}$. This is always possible as one can add a compression phase at the end of any non-minimal pattern.

Theorem 5.3.4 (Error-Correction Prevents Resource Blow-up)**.** *Let C be a minimal MBQC pattern implementing a CPTP map C. Let $C \cup T$ denote a trappified pattern implementing C obtained from $\boldsymbol{P}(\lambda)$. Further assume that Protocol 4 using $\boldsymbol{P}(\lambda)$ has negligible security error $\max(\epsilon, \nu)$ with respect to λ.*

If $|O_{C\cup T}|/|O_C| = 1$ for a non-negligible fraction of trappified canvases $T \in \boldsymbol{P}(\lambda)$, then the overhead $|G_{\boldsymbol{P}(\lambda)}|/|G_C|$ is super-polynomial in λ.

The usefulness of this theorem comes from the contra-positive statement. Achieving exponential verifiability with a polynomial overhead imposes that $|O_{C\cup T}|/|O_C| > 1$ for an overwhelming fraction of the trappified patterns. This means that the computation is at least partially encoded into a larger physical Hilbert space, which then serves to actively perform some form of error-correction.

Proof. Consider a trappified pattern $C \cup T$ for computing C obtained from $\boldsymbol{P}(\lambda)$ such that $|O_{C\cup T}| = |O_C|$. Given $\preceq_{G_{\boldsymbol{P}(\lambda)}}$, let $o_{C\cup T} \in O_{C\cup T}$ be the first output position of the computation. By definition, a bit-flip operation applied on position $o_{C\cup T}$ cannot be detected by the trap in $C \cup T$ since the outcome of the trap is independent of the computation. Yet, because C is minimal and $|O_{C\cup T}| = |O_C|$, we get that for some input states, the bit-flip deviation on $o_{C\cup T}$ si harmful. As a consequence, there exists a λ_0 such that, for all $\lambda \geq \lambda_0$, the diamond distance between C and the bit-flipped version is greater than $\nu(\lambda)$. To obtain exponential verification it is therefore necessary for this bit flip to be in the set of ϵ-detected deviations. This means that deviating on this position without being detected can happen for at most a negligible fraction $\eta(\lambda)$ of the trappified canvases in $\boldsymbol{P}(\lambda)$. In other words, the position $o_{C\cup T}$ can only be the first output computation qubit for a negligible fraction $\eta(\lambda)$ of trappified patterns in $\boldsymbol{P}(\lambda)$ that satisfy $|O_{C\cup T}| = |O_C|$.

Then define $\tilde{\boldsymbol{P}}(\lambda) = \{P = E_{\mathfrak{C}}(\mathsf{C}, \boldsymbol{P}(\lambda)),\ |O_{C\cup T}| = |O_C|\}$ as the set of trappified patterns for $\mathfrak{C}$ that have no overhead, and $O = \{o_{C\cup T},\ T \in \tilde{\boldsymbol{P}}(\lambda)\}$ the set of vertices

corresponding to their first output location. By construction, we have $\sum_{o\in O}|\{T \in \tilde{\boldsymbol{P}},\ o_{C\cup T} = o\}| = |\tilde{\boldsymbol{P}}|$. But, we just showed that $|\{T \in \tilde{\boldsymbol{P}},\ o_{C\cup T} = o\}|/|\boldsymbol{P}(\lambda)|$ is upper-bounded by η, negligible in λ. Thus, $|O|$ is lower-bounded by $|\tilde{\boldsymbol{P}}(\lambda)|/(|\boldsymbol{P}(\lambda)|\eta)$ which is super-polynomial in λ so long as $|\tilde{\boldsymbol{P}}(\lambda)|/|\boldsymbol{P}(\lambda)|)$ is not negligible in λ. □

Note that the situation where $|O_{C\cup T}| > |O_C|$ is interesting only if the bit-flip deviation on qubit $o_{C\cup T}$ does not alter the computation. Otherwise, the same reasoning as above is still applicable. This shows that enlarging the physical Hilbert space storing the output of the computation is useful only if it allows for some error-correction which transforms low-weight harmful errors into harmless ones.

5.4 Correctness and Security Amplification for Classical Input-Output Computations

We now construct a generic compiler to boost the properties of trappified schemes in the case of classical inputs. This compiler is a direct application of the results from the previous section regarding the requirement of error-correction since it uses a classical repetition code to protect the computation from low-weight bit-flips. It works by decreasing the set of errors which are detected and increasing the set of errors to which the trappified scheme is insensitive. These errors then can be corrected via a recombination procedure, which in the classical case can be as simple as a majority vote.

5.4.1 Classical Input-Output Trappified Scheme Compiler

Theorem 5.3.1 presents a clear objective for traps: they should (i) detect harmful deviations while being insensitive to harmless ones. Yet, a trap in a trappified pattern cannot detect deviations happening on the computation part of the pattern itself. To achieve exponential verifiability, one further needs to ensure that there are sufficiently many trappified patterns so that it is unlikely that a potentially harmful deviation hits only the computation part of the pattern, and that it is detected with high probability when it hits the rest. This is best stated by Theorem 5.3.4, which imposes to (ii) error-protect the computation so that hard-to-detect deviations are harmless while remaining harmful errors are easy to detect. Additionally, one further needs to (iii) find a systematic way to insert traps alongside computation patterns to generate these exponentially many trappified patterns.

Ideally, we would like to be able to design and analyse points (i), (ii) and (iii) independently from one another as much as possible. We show here a general way of performing this decomposition given slight constraints on the client's desired computation.

It is based on the realisation that if the client has d copies of its inputs – which is always possible whenever the inputs are classical – it can run d times its desired computation by repeating d times the desired pattern C on graph G sequentially or in parallel. If the output is classical, it is then naturally protected by a repetition code of length d and the result of the computation can be obtained through a majority vote. These d executions are called *computation rounds*. To detect deviations, the client needs to run s additional rounds which contain only traps. More precisely, each of these *test rounds* is a pattern run on the same graph G so that it is blind-compatible with C (see Definition 5.2.6). The collections of these s test rounds themselves constitute trappified canvases according to Definition 5.2.2, where acceptance is conditioned to less than w test rounds failures. Now, because computation rounds and test rounds are executed using blind-compatible patterns on the graph G, the trap insertion (iii) can be achieved by interleaving at random the s test rounds with the d computation rounds.

These steps, which are a generalisation of the technique from [Lei+21], are formalised in the following definition.

Definition 5.4.1 (Amplified Trappification Compiler)**.** *Let $\boldsymbol{P}$ be trappified scheme on a graph $G = (V, E)$, and let $d, s \in \mathbb{N}$, $n = d + s$ and $w \in [s]$. Let $\mathfrak{C}$ be the class of computations with classical inputs that can be evaluated by an MBQC pattern on G using an order $\preceq_G$ which is compatible with the order $\preceq_{\boldsymbol{P}}$ induced by $\boldsymbol{P}$. We define the Amplified Trappification Compiler that turns $\boldsymbol{P}$ into a trappified scheme $\boldsymbol{P}'$ on G^n for computation class $\mathfrak{C}$ as follows:*

- *The trappified canvases $T' \in \boldsymbol{P}'$ and their distribution is given by the following sampling procedure:*

  1. *Randomly choose a set $S \subset [n]$ of size s. These will be the test rounds;*
  2. *For each $j \in S$, independently sample a trappified canvas T_j from the distribution of $\boldsymbol{P}$.*

- *For each trappified canvas T' defined above and an output $t = (t_j)_{j \in S}$, the output of the decision function τ' is obtained by thresholding over the outputs of the decision*

functions τ_j of individual trappified canvases. More precisely:

$$\tau'(t) = 0 \text{ if } \sum_{j \in S} \tau_j(t_j) < w, \text{ and } 1 \text{ otherwise} \tag{5.47}$$

- *The partial ordering of vertices of G^n in $\boldsymbol{P}'$ is given by the ordering $\preceq_G$ on each copy of G.*
- *Let $\mathsf{C} \in \mathfrak{C}$ and C the pattern on G which implements the computation C. Given a trappified canvas $T' \in \boldsymbol{P}'$, the embedding algorithm $E_{\mathfrak{C}}$ sets to C the pattern of the d graphs that are not in S.*

5.4.2 Boosting Detection and Insensitivity

The next theorem relates the parameters d, s, w with the deviation detection capability of the test rounds, thus showing that not only (i), (ii) and (iii) can be designed separately, but also analysed separately with regards to the security achieved by the protocol.

Theorem 5.4.2 (From Constant to Exponential Detection and Insensitivity Rates)**.** *Let $\boldsymbol{P}$ be a trappified scheme on graph G which ϵ-detects the error set $\mathcal{E}_1$, is δ-insensitive to $\mathcal{E}_2$ and perfectly insensitive to $\{\mathbb{I}\}$. For $d, s \in \mathbb{N}$, $n = d + s$ and $w \in [s]$, let $\boldsymbol{P}'$ be the trappified scheme resulting from the compilation defined in Definition 5.4.1.*

For $\mathsf{E} \in \mathcal{G}_{V^n}$, let $\mathrm{wt}(\mathsf{E})$ be defined as the number of copies of G on which E does not act as the identity. We define $\mathcal{E}_{\geq k,\mathcal{F}} = \{\mathsf{E} \in (\mathcal{F} \cup \{\mathbb{I}\})^n \mid \mathrm{wt}(\mathsf{E}) \geq k\}$, and $\mathcal{E}_{\leq k,\mathcal{F}}$ analogously.

Let $k_1 > nw/(s\epsilon)$ and $k_2 < nw/(s\delta)$. Then, $\boldsymbol{P}'$ ϵ'-detects $\mathcal{E}_{\geq k_1,\mathcal{E}_1}$ and is δ'-insensitive to $\mathcal{E}_{\leq k_2,\mathcal{E}_2}$ where:

$$\epsilon' = \min_{\chi \in \left[0, \frac{k_1}{n} - \frac{w}{s\epsilon}\right]} \exp\left(-2\chi^2 s\right) + \exp\left(-2\frac{\left(\left(\frac{k_1}{n} - \chi\right) s\epsilon - w\right)^2}{\left(\frac{k_1}{n} - \chi\right) s}\right), \tag{5.48}$$

$$\delta' = \min_{\chi \in \left[0, \frac{w}{s\delta} - \frac{k_2}{n}\right]} \exp\left(-2\chi^2 s\right) + \exp\left(-2\frac{\left(\left(\frac{k_2}{n} + \chi\right) s\delta - w\right)^2}{\left(\frac{k_2}{n} + \chi\right) s}\right). \tag{5.49}$$

Proof. For a given deviation E, let X be a random variable describing the number of test rounds on which the deviation's action is not identity, where the probability is taken over the choice of the trappified canvas $\boldsymbol{P}'$. Let Y be a random variable counting the number of test rounds for which the decision function rejects.

Let $x \in [s]$, we can always decompose $\Pr[Y < w]$ as:

$$\begin{aligned}\Pr[Y < w] &= \Pr[Y < w \mid X \leq x] \Pr[X \leq x] + \Pr[Y < w \mid X > x] \Pr[X > x] \\ &\leq \Pr[X \leq x] + \Pr[Y < w \mid X > x]. \end{aligned} \tag{5.50}$$

We now aim to bound both terms above.

Let $\mathsf{E} \in \mathcal{E}_{\geq k_1, \varepsilon_1}$. In this case, by definition of E and construction of $\boldsymbol{P'}$, X is lower-bounded in the usual stochastic order by a variable $\tilde{X}$ following a hypergeometric variable distribution of parameters (n, k_1, s). We fix $x = \left(\frac{k_1}{n} - \chi\right) s$ for $\chi \geq 0$ and use tail bounds for the hypergeometric distribution to get:

$$\Pr\left[X \leq \left(\frac{k_1}{n} - \chi\right) s\right] \leq \Pr\left[\tilde{X} \leq \left(\frac{k_1}{n} - \chi\right) s\right] \leq \exp\left(-2\chi^2 s\right). \tag{5.51}$$

For the other term, note that Y, conditioned on a lower bound x for X, is lower-bounded in the usual stochastic order by an (x, ϵ)-binomially distributed random variable $\tilde{Y}$. Hoeffding's inequality for the binomial distribution then implies that:

$$\Pr[Y < w \mid \ X > x] \leq \Pr\left[\tilde{Y} < w\right] \leq \exp\left(-2\frac{(x\epsilon - w)^2}{x}\right). \tag{5.52}$$

All in all, replacing the value of x above with $\left(\frac{k_1}{n} - \chi\right) s$ and combining it with the first bound, we have for $\chi \leq \frac{k_1}{n} - \frac{w}{s\epsilon}$ that:

$$\Pr[Y < w] \leq \exp\left(-2\chi^2 s\right) + \exp\left(-2\frac{\left(\left(\frac{k_1}{n} - \chi\right) s\epsilon - w\right)^2}{\left(\frac{k_1}{n} - \chi\right) s}\right). \tag{5.53}$$

This concludes the first statement.

For the second statement, we can similarly decompose $\Pr[Y \geq w]$ as:

$$\Pr[Y \geq w] \leq \Pr[Y \geq w \mid X < x] + \Pr[X \geq x]. \tag{5.54}$$

Let $\mathsf{E} \in \mathcal{E}_{\leq k_2, \varepsilon_2}$. Now X is upper-bounded in the usual stochastic order by a variable $\tilde{X}$ following a hypergeometric distribution of parameters (n, k_2, s), by definition of E. This holds here because the scheme is perfectly insensitive to $\mathbb{I}$, and therefore the identity

never triggers tests. It then holds for all $\chi \geq 0$ that

$$\Pr\left[X \geq \left(\frac{k_2}{n} + \chi\right) s\right] \leq \Pr\left[\tilde{X} \geq \left(\frac{k_2}{n} + \chi\right) s\right] \leq \exp\left(-2\chi^2 s\right), \tag{5.55}$$

using tail bounds for the hypergeometric distribution.

Similarly, here, Y (conditioned on an upper bound x for X) is upper-bounded in the usual stochastic order by an (x, δ)-binomially distributed random variable $\tilde{Y}$. This also holds because of the perfect insensitivity of tests to $\mathbb{I}$. Hoeffding's inequality yields

$$\Pr\left[Y \geq w \mid \ X \leq x\right] \leq \Pr\left[\tilde{Y} \geq w\right] \leq \exp\left(-2\frac{(x\delta - w)^2}{x}\right). \tag{5.56}$$

We then conclude that

$$\Pr\left[Y \geq w\right] \leq \exp\left(-2\chi^2 s\right) + \exp\left(-2\frac{\left(\left(\frac{k_2}{n} + \chi\right) s\delta - w\right)^2}{\left(\frac{k_2}{n} + \chi\right) s}\right) \tag{5.57}$$

for $\chi \leq \frac{w}{s\delta} - \frac{k_2}{n}$. □

The consequence of the above theorem is that whenever the trappified schemes are constructed by interleaving computation rounds with test rounds chosen at random from a given set, the performance of the resulting protocol implementing SDQC crucially depends on the ability of these test rounds to detect harmful errors. Therefore, when using the compiler, optimisation of the performance is achieved by focussing only on designing more efficient test rounds. This is addressed in Section 5.5.

Remark 5.4.3. *Note that we do not make use in Definition 5.4.1 of the embedding function or computation class associated with the trappified scheme $\boldsymbol{P}$. In fact the initial scheme can even consist of pure traps as described in Remark 5.2.14. This is the case for the schemes described in the next sections. If each trappified scheme used for tests can also embed the client's computation of interest, it is possible to use the alternative parallel repetition compiler presented in Appendix 5.8 which has no separate computation rounds.*

5.4.3 Correctness Amplification via Majority Vote

Theorem 5.4.2 has given detection and insensitivity errors that are negligible n. In order to recover exponential verifiability, we must now also make the correctness error

negligible in n. To this end, we recombine the multiple computation rounds into a single final result so that error of weight lower than k_2 are corrected.

Here, $\mathfrak{C}$ is the class of BQP computations that can be implemented on G, which implies that the failure probability for obtaining the correct result is c, below and bounded away from $1/2$. Then, we define $\boldsymbol{V}$ from the compiled $\boldsymbol{P}'$ by requiring that the input subspace is symmetric with respect to exchanging computation rounds – i.e. all computation rounds have the same inputs – and by defining the output subspace as the bitwise majority vote of computation round outputs.

Intuitively, if it is guaranteed that the fraction of all rounds affected by a possibly harmful deviation is less than $(2c-1)/(2c-2)$ then the output of $\boldsymbol{V}$ will yield the correct result of the computation. This is because, in the large n limit, out of the d computation rounds a fraction c will be incorrect due to the probabilistic nature of the computation itself. Consequently, to maintain that more than $1/2$ the computation rounds yield the correct result so that the majority vote is able to eliminate the suprious results, the fraction f of computation rounds that the deviation can affect must satisfy $(1-c)(1-f) > c + (1-c)f$, that is $f < (2c-1)/(2c-2)$. Due to the blindness of the scheme, it is enough to impose that no more than a fraction $(2c-1)/(2c-1)$ of the n rounds is affected by the deviation to obtain the desired guarantee on the computation rounds with high probability.

Theorem 5.4.4 (Exponential Correctness from Majority Vote)**.** *Let $\boldsymbol{T}$ be a trappified scheme on graph G which is perfectly correct on $\{\mathbb{I}\}$, for computations $\mathfrak{C} = \mathsf{BQP} \cap \mathfrak{G}$ where $\mathfrak{G}$ is the set of MBQC computations which can be performed on graph G. For $d, s \in \mathbb{N}$ and $n = d + s$, let $\boldsymbol{V}$ be the trappified scheme obtained through the compiler of Definition 5.4.1 and let the input subspace $\Pi_{I,C}$ be symmetric with respect to exchanging computation rounds. The output subspace $\Pi_{O,C}$ is defined as the concatenation of the (classical) outputs of all computation rounds and the decoding algorithm $\mathsf{D}_{O,C}$ is the bitwise majority vote of computation rounds outputs from the d computations.*

Let c be the bounded error of BQP *computations and $k < \frac{2c-1}{2c-2}n$. Then, $\boldsymbol{V}$ is ν-correct on $\mathcal{E}_{\leq k, \mathcal{G}_V}$ for*

$$\nu \leq \exp\left(-2\left(1 - \frac{2c-1}{2c-2} + \varphi - \epsilon_1\right) d\epsilon_2^2\right), \tag{5.58}$$

with

$$\frac{1}{2} - \left(\frac{2c-1}{2c-2} - \varphi + \epsilon_1\right) = (c + \epsilon_2)\left(1 - \frac{2c-1}{2c-2} + \varphi - \epsilon_1\right) \tag{5.59}$$

and $\varphi, \epsilon_1, \epsilon_2 > 0$. Thus ν is exponentially small in n if d/n is constant.

Proof. We will compute the bound on the correctness for finite n. First, define two random variables Z_1 and Z_2 that account for possible sources of erroneous results for individual computation rounds. More precisely, Z_1 is the number of computation rounds that are affected by a deviation containing an Y or Z for one of the qubits in the round. Z_2 is the number of computation rounds which give the wrong outcome due to the probabilistic nature of the computation itself – i.e. inherent failures for the computation in the honest and noise free case. Given that $\boldsymbol{V}$ uses a majority vote to recombine the results of each computation rounds, as long a $Z_1 + Z_2 < d/2$, then the output result will be correct.

Our goal now is to show that the probability that $Z_1 + Z_2$ is greater than $d/2$ can be made negligible. For any z_1 one has the following:

$$\Pr\left[Z_1 + Z_2 \geq \frac{d}{2}\right] = \Pr\left[Z_1 + Z_2 \geq \frac{d}{2} | Z_1 \leq z_1\right] \Pr[Z_1 \leq z_1] \tag{5.60}$$

$$+ \Pr\left[Z_1 + Z_2 \geq \frac{d}{2} | Z_1 > z_1\right] \Pr[Z_1 > z_1]. \tag{5.61}$$

Then:

$$\Pr\left[Z_1 + Z_2 \geq \frac{d}{2}\right] \leq \Pr\left[Z_1 + Z_2 \geq \frac{d}{2} | Z_1 \leq z_1\right] + \Pr[Z_1 > z_1] \tag{5.62}$$

$$\leq \Pr\left[Z_2 \geq \frac{d}{2} - z_1 | Z_1 \leq z_1\right] + \Pr[Z_1 > z_1] \tag{5.63}$$

$$\leq \Pr\left[Z_2 \geq \frac{d}{2} - z_1 | Z_1 = z_1\right] + \Pr[Z_1 > z_1]. \tag{5.64}$$

Now, consider a deviation in $\mathcal{E}_{\leq k, \mathcal{G}_V}$. Using the tail bound for the hypergeometric distribution defined by choosing independently at random and without replacement d computation rounds out of a total of n rounds, k of which at most are affected by the deviation, one finds that for $z_1 = (k/n + \epsilon_1)d$ with $0 < \epsilon_1$,

$$\Pr\left[Z_1 > \left(\frac{k}{n} + \epsilon_1\right) d\right] \leq \exp\left(-2\epsilon_1^2 d\right). \tag{5.65}$$

Additionally, once Z_1 is fixed, Z_2 is biniomally distributed with probability c. Therefore,

using tail bound for this distribution, one has for $\epsilon_2 > 0$:

$$\Pr\left[Z_2 \geq (c+\epsilon_2)\left(1-\frac{k}{n}-\epsilon_1\right)d | Z_1 = (\frac{k}{n}+\epsilon_1)d\right] \leq \exp\left(-2\left(1-\frac{k}{n}-\epsilon_1\right)d\epsilon_2^2\right). \tag{5.66}$$

Using these inequalities, we obtain that:

$$\Pr\left[Z_1+Z_2 \geq \frac{d}{2}\right] \leq \exp\left(-2\epsilon_1^2 d\right) + \exp\left(-2\left(1-\frac{k}{n}-\epsilon_1\right)d\epsilon_2^2\right), \tag{5.67}$$

where we set

$$\frac{d}{2} - \left(\frac{k}{n}+\epsilon_1\right)d = (c+\epsilon_2)\left(1-\frac{k}{n}-\epsilon_1\right)d, \tag{5.68}$$

which has solutions for $\epsilon_1, \epsilon_2 > 0$ when $k/n = (2c-1)/(2c-2) - \varphi$ with $\varphi > 0$. This shows that the correctness error $\nu = \Pr[Z_1 + Z_2 \geq d/2]$ can therefore be made negligible in n for fixed d/n.

□

To conclude this section, we obtain simultaneous negligibility for detection, insensitivity and correctness errors by combining the conditions from Theorems 5.4.2 and 5.4.4:

$$\begin{aligned} &w = \left(\frac{2c-1}{2c-2} - \varphi - \chi\right) s(1-p), \\ &0 < \varphi < \frac{2c-1}{2c-2}, \quad 0 < \chi < \frac{2c-1}{2c-2} - \varphi, \quad 0 < \epsilon_1 < \varphi, \\ &\frac{1}{2} - \frac{2c-1}{2c-2} - \epsilon_1 = (c+\epsilon_2)\left(1 - \frac{2c-1}{2c-2} - \epsilon_1\right). \end{aligned} \tag{5.69}$$

Under these conditions, Theorem 5.3.1 yields an exponentially secure verification protocol using the trappified scheme $\boldsymbol{V}$.

Finally, while a simple majority vote is sufficient to recombine the computations in the classical case, finding such a distillation procedure in the quantum case is left as an open question.

5.5 New Optimised Trappified Schemes from Stabiliser Testing

In this section we demonstrate how the various tools and techniques introduced earlier can be combined to design trappified schemes that provide efficient and robust verifiability. To achieve this, we use Remark 5.2.13 and Lemma 5.2.15 to construct a trappified scheme $\boldsymbol{T}$ based on stabiliser testing with a constant detection error. Here we again focus on classical-input classical-output computations. Theorems 5.4.2 and 5.4.4 show that it is sufficient in this case to focus on designing test rounds, with the compiler from Definition 5.4.1 and majority vote then boosting the detection, insensitivity and correctness.

In the process, we show a close correspondence between prepare-and-send protocols derived from [FK17], and protocols based on stabiliser tests following [McK16]. This broadens noticeably the possibilities for designing new types of trappified patterns beyond those which are used by existing prepare-and-send protocols. It also allows to transfer existing protocols based on stabiliser testing from the non-communicating multi-server setting to the prepare-and-send model, thus lowering the assumptions of these protocols and making them more readily implementable and practical. We show in later subsections how to use the compiler results together with these new possibilities to optimise the current state-of-the-art protocol of [Lei+21].

5.5.1 Trappified Schemes from Subset Stabiliser Testing

Given $G = (V, E)$ and a partial order $\preceq_G$ on V, the first step for constructing a verification protocol for computations on G is to detect deviations from the server. To this end, we recall that any action from the server can be always be viewed as first performing the unitary part of Protocol 2 followed by a pure deviation that is independent from the computation delegated to the server (see Section 5.3). To be constructive and build traps that can be easily computed and checked by the client, we impose in this section that the outcomes of trappified canvases are deterministic and that they accept with probability 1 for honest executions of the protocol.

We first focus on the simplest case of deterministic functions, where the decision algorithm τ for the trappified canvas is such that $\tau(t) = t_i$ where t_i is measurement outcome of qubit i. In other words the test round accepts if the outcome $t_i = 0$, which corresponds to obtaining outcome $|0\rangle$ for qubit i, while all other measurements outcomes

t_j for $j \neq i$ are ignored.[12]

For the outcome of the trappified canvas to be deterministic, qubit i must be equal to $|0\rangle$ in absence of deviations before the computational basis measurement. In other words, the state of i is an eigenstate of Z_i. By commuting Z_i towards the initialisation of the qubits – through the Hadamard gate and the entangling operations defined by the graph G, we conclude that determinism and acceptance of deviation-less test rounds implies that the initial state of the qubits before running the protocol is an eigenstate of $\mathsf{X}_i \bigotimes_{j \in N_G(i)} \mathsf{Z}_j = \mathsf{S}_i$.

The following lemma explains how to prepare a single-qubit tensor product state stabilised by such given Pauli operator.

Lemma 5.5.1 (Tensor Product Preparation of a State in a Stabiliser Subspace)**.** *Let* P *be an element of the Pauli group over* N *qubits, such that* $\mathsf{P}^2 \neq -\mathbb{I}$*. Then, there exists* $|\psi\rangle = \bigotimes_{i=1}^{N} |\psi_i\rangle$ *such that* $|\psi\rangle = \mathsf{P}|\psi\rangle$*, and* $\forall i, |\psi_i\rangle \in \{|0\rangle, |+\rangle, \left|+_{\pi/2}\right\rangle\}$.

Proof. Without loss of generality, one can write $\mathsf{P} = s \bigotimes_i \mathsf{P}(i)$ with $s = \pm 1$ and where $\mathsf{P}(i) \in \{\mathbb{I}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}\}$ is the restriction of P to qubit i. Then by construction, $\mathsf{P} \in \langle S \rangle$, where $\langle \mathcal{S} \rangle$ denotes the multiplicative group generated by the set $\mathcal{S} = \{s\mathsf{P}(i_0) \bigotimes_{j \neq i_0} \mathbb{I}\} \cup \{\mathsf{P}(i) \bigotimes_{j \neq i} \mathbb{I}\}_{i \neq i_0}$, where i_0 is the smallest index i for which $\mathsf{P}(i) \neq \mathbb{I}$. Now, consider the state that is obtained by taking the tensor product of single qubit states that are the common $+1$ eigenstates of the operators in set $\mathcal{S}$. The above shows that it is a $+1$ eigenstate of all operators in $\langle \mathcal{S} \rangle$, and in particular of P, which concludes the proof as eigenstates of single-qubit Pauli operators are precisely the desired set. □

One can further note that the above lemma also holds for a set $\mathcal{R}$ of Pauli operators if,

$$\forall \mathsf{P}, \mathsf{Q} \in \mathcal{R},\ \forall i \in V,\ \mathsf{P}(i) = \mathsf{Q}(i) \text{ or } \mathsf{P}(i) = \mathbb{I} \text{ or } \mathsf{Q}(i) = \mathbb{I}. \tag{5.70}$$

Now take $\mathcal{R}$ a set of Pauli operators generating the stabiliser group of $|G\rangle$, and $\{\mathcal{R}^{(k)}\}_j$ a collection of subsets of $\mathcal{R}$ that such that each $\mathcal{R}^{(k)}$ satisfies the condition of Equation 5.70 and $\cup_k \mathcal{R}^{(k)} = \mathcal{R}$ – note that $\mathcal{R}$ need not be a minimal set of generators. We then construct a set of trappified canvases $T^{(k)}$ which have V as their input set and for which all qubits are measured in the X basis. They only differ in the prepared input states, each being prescribed by Lemma 5.5.1 for the stabilisers in $\mathcal{R}^{(k)}$ – that is qubits are prepared in an X, Y or Z eigenstate each time one of the Pauli operator in $\mathcal{R}^{(k)}$ is

[12] Recall that throughout this chapter, our convention is to view rotated $\{|\pm_\theta\rangle\}$ measurements as Z rotations followed by a Hadamard gate and a measurement in the computational basis.

respectively X, Y or Z for this qubit, and chosen arbitrarily to be X eigenstates elsewhere. As above, the computation defined by the pattern where all qubits are measured in the X basis amounts to measuring the stabiliser generators S_i. The output distribution $\mathcal{T}^{(k)}$ can be computed given the prepared input state for $T^{(k)}$ using elementary properties of stabiliser states. But for our purposes, it is sufficient to construct the decision function $\tau^{(k)}$. This can be done by noting that for all $\mathsf{P} \in \mathcal{R}^{(k)}$, there is a unique binary vector $\{p_i\}_i$ such that $\mathsf{P} = \prod_i \mathcal{S}_i^{p_i}$. This, in turn, implies that $\mathcal{T}^{(k)}$ is such that $\bigoplus_i p_i t_i = 0$ where t_i is the outcome of the measurement of the i-th qubit in the X basis. Therefore, we define

$$\tau^{(k)}(t) = \bigwedge_{\mathsf{P} \in \mathcal{R}^{(k)}} \left(\bigoplus_i p_i t_i = 0 \right), \tag{5.71}$$

which reconstructs the measurement outcomes of stabilisers in $\mathcal{R}^{(k)}$ from the measurements outcomes of operators S_i. The function $\tau^{(k)}(t)$ will accept whenever the measurement outcomes of all stabilisers in $\mathcal{R}^{(k)}$ are zero. We denote by $\mathcal{E}_1^{(k)}$ the set of Pauli deviations that are perfectly detected by $T^{(k)}$ and $\mathcal{E}_2^{(k)} = \mathcal{G}_V \setminus \mathcal{E}_1^{(k)}$ the set of deviations to which $T^{(k)}$ is perfectly insensitive.

Now, using Remark 5.2.13 and Lemma 5.2.15, the trappified canvases $T^{(k)}$ can be composed with equal probability p to obtain a trappified scheme $\boldsymbol{T}$. We then consider the sets of all Pauli deviations $\mathcal{E}_1 = \bigcup_k \mathcal{E}_1^{(k)}$ and $\mathcal{E}_2 = \bigcup_k \mathcal{E}_2^{(k)} = \mathcal{G}_V$. We conclude that the scheme $\boldsymbol{T}$ then $(1-p)$-detects $\mathcal{E}_1$ and is $(1-p)$-insensitive to $\mathcal{G}_V$. Note that these values are upper-bounds, with equality being achieved if there is no overlap in the set of errors which each canvas can detect.

The scheme $\boldsymbol{T}$ therefore detects all possibly harmful deviations with finite probability, and is partly insensitive to all deviations – i.e. both harmless and harmful – that can affect computations in $\mathfrak{C}$.

5.5.1.1 A Linear Programming Problem for Trap Optimisation

At first glance, the main goal to optimise such schemes seems to be to lower as much as possible the number of subsets of stabilisers $\mathcal{R}^{(k)}$ which cannot be tested at the same time. Each such subset of stabilisers needs a different canvas $T^{(k)}$ to test for it, and the probability p increases with a lower number of canvases. An increase in p automatically decreases the detection and insensitivity errors. These in turn appear in the exponential bounds from Theorem 5.4.2, meaning that even a slight decrease greatly influences the total security for a given number of repetitions, or equivalently the number of repetitions

required to achieve a given security level.

However this is the case only if each test detects a set of errors disjoint from those detected by the other sets. Another way to increase the probability of detection is to increase the coverage of each canvas by increasing the number of stabiliser errors which each can detect. In this case, the sets can be made to overlap and the detection probability can be lowered below the upper-bound of $1 - p$. We explore both approaches in the next two subsections. We now give a general process for systematising this optimisation with different constraints.

In particular situations, it might be useful to have more granular control of the design and error-detecting capabilities of the test rounds. For instance, because of hardware constraints or ease of implementation, it might be favourable to restrict the set of tests one is willing to perform to only a subset of the tests resulting from generalised traps. As one example, one might desire to avoid the preparation of dummy states and therefore restrict the set of feasible tests to those requiring the preparation of quantum states in the $X - Y$-plane only. It might also not be necessary for the employed tests to detect all possible Pauli errors because of inherent robustness of the target computation.

In such cases, we can expect better error-detection rates if we (i) allow for more types of tests, or (ii) remove deviations from the set of errors that are required to be detected. To this end, we present a linear programming formulation of the search for more efficient tests in Problem 1.

Remark 5.5.2. *While efficient algorithms exist to find solutions to such real-valued constrained linear problems, in this case the number of constraints grows linearly with the number of errors that need to be detected, and therefore generally exponentially in the size of the graph.*

Remark 5.5.3. *Solutions to the dual problem of Problem 1 are distributions of deviations applied to the test rounds. An optimal solution to the dual gives therefore an optimal attack, i.e. a distribution of deviations that achieves a minimal detection rate with the tests at hand.*

5.5.2 Standard Traps

The simplest application of Lemma 5.5.1 is to prepare qubit i_0 as an eigenstate of X, while its neighbours in the graph are prepared as an eigenstate of Z. This setup can detect all deviations which do not commute with the Z_{i_0} measurements of i_0. Here, the

Problem 1 Optimisation of the Distribution of Tests

Given

- a set of errors $\mathcal{E}$ to be detected,
- a set of feasible tests $\mathcal{H}$,
- a relation between tests and errors describing whether a test detects an error, $R : \mathcal{H} \times \mathcal{E} \to \{0, 1\}$,

find an optimal distribution $p : \mathcal{H} \to [0, 1]$ **maximising** the detection rate $\epsilon \in [0, 1]$ **subject to** the following conditions:

- p describes a probability distribution, i.e. $\sum_{H \in \mathcal{H}} p(H) \leq 1$,
- all concerned errors are detected at least with the target detection rate, i.e.

$$\forall E \in \mathcal{E} : \quad \sum_{\substack{H \in \mathcal{H} \\ R(H,E)=1}} p(H) \geq \epsilon. \tag{5.72}$$

reader familiar with the line of work following [FK17] note that we have recovered their single-qubit traps: single qubits prepared in the $X - Y$ plane and surrounded by dummy $|0\rangle$ or $|1\rangle$ qubits.

Additionally, within each test round, it is possible to include several such atomic traps as long as their initial states can be prepared simultaneously – i.e. they can at most overlap on qubits that need to be prepared as eigenstates of Z. More precisely, take H to be an independent set of vertices from G (see Definition 5.7.1). We define the set of stabilisers associated to H as $\mathcal{R}_H = \{\mathsf{S}_i\}_{i \in H}$. Such sets naturally follow the condition of Equation 5.70 since H is an independent set and therefore if $i \neq j$, $\mathsf{S}_i(j) = \mathsf{S}_j(i) = \mathbb{I}$ and both stabilisers are equal to either Z or $\mathbb{I}$ for all qubits different from i or j. This is the extreme case where all stabilisers in $\mathcal{R}_H$ have a single component when decomposed in the generator set $\{\mathsf{S}_i\}$.

Following the same line of argument as above, in absence of deviation, the state of qubit i must be $|0\rangle$ for all $i \in H$ before the measurement, or equivalently, is an eigenstate of Z_i. Commuting these operators towards the initialisation of the qubits shows that the qubits in H must be prepared in the state $|+\rangle$, and $|0\rangle$ for qubits in $N_G(H)$. These qubits form the input set I_T of the trappified canvas T_H associated to the independent set H. Other qubits can be prepared in any allowed state. Its output locations O_T are the independent set H.

Using the formula from Equation 5.71 for set $\mathcal{R}_H$, we get $\tau(t) = \bigwedge_{i\in H} t_i$ for the decision algorithm. That is, the trappified canvas accepts whenever all outcomes Z measurements for qubits $i \in H$ are 0.

A trappified canvas T_H generated in this way depends only on the choice of independent set H. Such trappified canvases will be called *standard trap* in the remaining of this work.

Let $\{H^{(j)}\}_j$ be a set of independent sets. Since $\mathcal{R}_{H^{(j)}}$ contains all stabilisers S_i for $i \in H^{(j)}$, the sets $\mathcal{R}_{H^{(j)}}$ cover the generating set of stabiliser $\{\mathsf{S}_i\}_{i\in V}$ entirely if and only if each qubit $i \in V$ is in at least one of the independent sets $H^{(j)}$. Then one can conclude that all X and Y deviations have a non-zero probability of being detected, while $\mathbb{I}$ and Z deviations are never detected, but are harmless for classical output computations.

5.5.2.1 Optimising Standard Traps.

The background in graph theory and graph colourings necessary for this section can be found in Appendix 5.7.

The crucial parameter to optimise is the detection probability of individual test rounds with respect to X deviations. In other words, the performance of the scheme will vary depending on the choice of probability distribution over the independent set $\mathcal{I}(G)$ and the detection capability of each individual test round.

A test round, and therefore its corresponding trappified canvas, will detect a Pauli error if and only if at least one of the $|+\rangle$-states is hit by a local X or Y deviation.

Lemma 5.5.4 (Detection Rate)**.** *Let $G = (V, E)$ be an undirected graph. Let $\mathcal{D}$ be a probability distribution over $\mathcal{I}(G)$, giving rise to the trappified scheme $\boldsymbol{P}$ where every element of $\mathcal{I}(G)$ describes one trappified canvas. We define the* detection rate *of $\mathcal{D}$ over G as*

$$p_{det}(\mathcal{D}) = 1 - \epsilon(\mathcal{D}) = \min_{\substack{M\subseteq V \\ M\neq\emptyset}} \Pr_{H\sim\mathcal{D}} [M \cap H \neq \emptyset] \,. \tag{5.73}$$

Then $\boldsymbol{P}$ $\epsilon(\mathcal{D})$-detects the error set $\mathcal{E} = \{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}^{\otimes V} \setminus \{\mathsf{I},\mathsf{Z}\}^{\otimes V}$.

Proof. The trappified canvas induced by the independent set $H \in \mathcal{I}(G)$ detects an error E if and only if $M \cap H \neq \emptyset$, where M is the set of all vertices on which E reduces to the Pauli-X or Y. The claim is then implied by Lemma 5.2.15. □

In the definition above, H corresponds to a choice of test round, while M is the set of qubits that are affected by to-be-detected X and Y deviations.

To obtain the lowest overhead, the distribution $\mathcal{D}$ should be chosen such that it maximises the detection probability $1 - \epsilon(\mathcal{D})$ for a given graph G. The following characterisation of the detection rate is going to be useful to determine upper bounds on p_{det}.

Remark 5.5.5. *For any graph G and any distribution $\mathcal{D}$ over $\mathcal{I}(G)$ it holds that*

$$p_{det}(\mathcal{D}) = \min_{\mathcal{M}} \Pr_{\substack{M\sim\mathcal{M} \\ H\sim D}} [M \cap H \neq \emptyset] \tag{5.74}$$

where the minimum ranges over distributions $\mathcal{M}$ over $\wp(V) \setminus \{\emptyset\}$.

It can be shown that the best achievable detection rate by standard traps for a graph G lie in the interval $\left[\frac{1}{\chi(G)}, \frac{1}{\omega(G)}\right]$, where $\chi(G)$ and $\omega(G)$ are respectively the chromatic number and the clique number of G. The protocol of [Lei+21] in particular is designed with security bounds depending on the chromatic number of the underlying graph. Note that the two graph invariants $\chi(G)$ and $\omega(G)$ are dual in the sense that they are integer solutions to dual linear programs and the gap between these two values can be large (see Lemma 5.7.4). It turns out that both bounds can be improved to depend on the solutions of the relaxations of the respective linear programs. This closes the integrality gap between the chromatic number and the clique number.

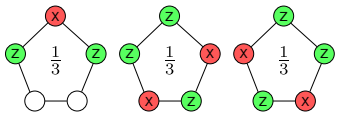
Lemma 5.5.6. *For every (non-null) graph G there exists a distribution $\mathcal{D}$ over $\mathcal{I}(G)$ such that $p_{det}(\mathcal{D}) \geq \frac{1}{\chi_f(G)}$, where $\chi_f(G)$ is the fractional chromatic number of G (see Definition 5.7.5).*

Proof. Let $\mathcal{D}$ be a distribution over $\mathcal{I}(G)$ such that for all $v \in V$ it holds that $\Pr_{H\sim\mathcal{D}}[v \in H] \geq \frac{1}{k}$. For all $M \subseteq V, M \neq \emptyset$, then $\Pr_{H\sim\mathcal{D}}[M \cap H \neq \emptyset] \geq \frac{1}{k}$ and therefore $p_{\text{det}}(\mathcal{D}) \geq \frac{1}{k}$. By Lemma 5.7.6, we can find such a distribution $\mathcal{D}$ for any $k \geq \chi_f(G)$. □

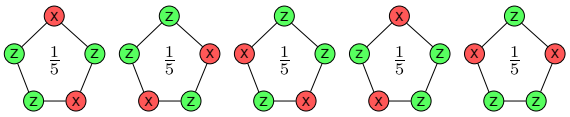
We can also improve the upper bound using fractional cliques.

Lemma 5.5.7. *For every (non-null) graph G and every distribution $\mathcal{D}$ over $\mathcal{I}(G)$ it holds that $p_{det}(\mathcal{D}) \leq \frac{1}{\omega_f(G)}$, where $\omega_f(G)$ is the fractional clique number of G (see Definition 5.7.7).*

Proof. This statement is a direct consequence of Lemma 5.7.8. □



(a) Trap distribution based on an optimal colouring of G.



(b) Trap distribution based on an optimal fractional colouring of G.

Figure 5.9: Traps on the cycle graph G with 5 nodes from Example 5.

As a consequence, this shows that the protocol described in [Lei+21], which is the current state-of-the-art, can sometimes be improved by constructing additional test rounds that would allow to have a probability of detection greater than the reported $1/\chi(G)$. In fact, this proves that the best possible detection rate by standard traps is equal to $1/\chi_f(G)$ since $\chi_f(G) = \omega_f(G)$ by Lemma 5.7.9. This is achieved precisely by choosing the set of possible tests to be a fractional colouring of the graph.

Example 5. *Let $G = (V, E)$ be the cycle graph on 5 nodes with $V = \{0, 1, 2, 3, 4\}$. An optimal proper 3-colouring of G is given by $(\{0, 2\}, \{1, 3\}, \{4\})$, which gives rise to a standard trap with detection rate $1/3$. However, this may be further improved using Lemma 5.5.6 and the fact that $\chi_f(G) = 5/2$. A standard trap with the optimal detection rate of $2/5$ is given by the uniform distribution over the set $\{\{0, 2\}, \{1, 3\}, \{2, 4\}, \{0, 3\}, \{1, 4\}\}$.*

Yet, this leaves a dependency of the protocol's efficiency on graph invariants, meaning that depending on the chosen computation, the protocol could perform poorly. The next section shows how to overcome this obstacle, as long as the client is willing to use more generalised traps.

5.5.3 General Traps

Above, the trappified canvases we obtained are a consequence of determinism, insensitivity to harmless deviations and a restriction on the subsets H, constrained to be independent. To construct *general traps*, we simply remove this last requirement and define instead $\mathcal{R}_H = \{\prod_{i \in H} \mathsf{S}_i\}$. Using Equation (5.71), τ is then the parity of measurement outcomes for qubits from H, i.e. $\tau(t) = \bigoplus_{i \in H} t_i$. This means that to accept the execution of such trappified canvas, the state of the qubits $i \in H$ needs to be in the $+1$ eigenspace of the operator $\bigotimes_{i \in H} \mathsf{Z}_i$. This is the other extreme case since there is only a single stabiliser in the set $\mathcal{R}_H$.

Commuting this operator towards the initialisation imposes to prepare a $+1$ eigenstate of $\bigotimes_{i\in H_{even}} \mathsf{X}_i \bigotimes_{j\in H_{odd}} \mathsf{Y}_j \bigotimes_{k\in N_G^{odd}(H)} \mathsf{Z}_k$, where H_{even} (resp. H_{odd}) are the qubits of even (resp. odd) degree within H, and $k \in N_G^{odd}(H)$ means k is in the odd neighbourhood of H. Again, applying Lemma 5.5.1 allows us to find in the eigenspace of this operator a state that can be obtained as a tensor product of single-qubit states, simply by looking at the individual Paulis from the operator $\prod_{i\in H} \mathsf{S}_i$.

It is easy to see that this trappified canvas detects all deviations that anti-commute with $\bigotimes_{i\in H} \mathsf{Z}_i$, that is deviations that have an odd number of X or Y for qubits in H. Varying the sets H allows to construct a trappified scheme which detects all possible deviations containing any number of X or Y with a constant probability.

5.5.3.1 Optimising General Traps.

General traps are based on test rounds defined by a set $H \subseteq V$ of qubit locations. It accepts whenever the parity of outcomes of Z-measurements on the qubits of H is even. Here the testing set H can be chosen freely and does not need to be independent as in the construction of standard traps.

Lemma 5.5.8 (General Stabiliser-Based Trappified Scheme)**.** *Let $\boldsymbol{P}$ be the trappified scheme defined by sampling uniformly at random a non-empty set $H \subseteq V$ and preparing the trappified canvas associated to $\mathcal{R}_H = \{\prod_{i\in H} \mathsf{S}_i\}$. Then $\boldsymbol{P}$ $1/2$-detects the error set $\mathcal{E} = \{\mathsf{I},\mathsf{X},\mathsf{Y}\}^{\otimes V} \setminus \{\mathsf{I}^{\otimes V}\}$.*

Proof. Looking at a given deviation E, we conclude that a test-round defined by H detects E if and only if $|E\cap H|$ is odd – here E denotes the set of qubits where E is equal to X or Y. If H is sampled uniformly at random from $\wp(G)$, then $\Pr_{H\sim\mathcal{U}(\wp(G))}[|E\cap H| \equiv 1 \mod 2] = 1/2$, and this is valid for any $\mathsf{E} \neq \mathsf{I}$. □

As a conclusion, we obtain that the probability of detection for this scheme is equal to $1/2$, which is independent of the graph G, and generally will beat the upper bound obtained in the previous section through standard traps.

5.6 Discussion and Future Work

We uncovered a profound correspondence between error-detection and verification that applies and unifies all previous trap-based blind verification schemes in the prepare-and-send MBQC model, which covers the majority of proposed protocols from the

literature. In addition, all results mentioned here also apply to receive-and-measure MBQC protocols via the recent equivalence result from [WEP22]. On the theoretical side, it provides a direct and generic composable security proof of these protocols in the AC framework, which also gives the first direct and explicit proof of composability if the original VBQC protocol [FK17]. We also formally showed that error-correction is required if one hopes to have negligible correctness and security errors with polynomial overhead when comparing unprotected and unverified computations to their secure counterparts. On a practical side, this correspondence can be used to increases the tools available to design, prove the composable security, and optimise the performance of new protocols. To exemplify these new possibilities, we described new protocols that improve the overhead of state-of-the-art verification protocols, thus making them more appealing for experimental realisation and possibly for integration into future quantum computing platforms.

The uncovered connection between error-detection and verification raises new questions such as the extent to which it is possible to infer from the failed traps what the server has been performing. Additionally, Theorem 5.3.4 implies that some form of error-correction is necessary to obtain exponential correctness. Yet, our protocol [Lei+21] shows that sometimes classical error-correction is enough, thereby raising the question of understanding what are the optimal error-correction schemes for given classes of computations that are to be verified. Finally, we strongly suspect the link between error detection and verification can be further developed and yield new trappified schemes with not only more efficient implementations but also additional capabilities.

5.7 Appendix: Graph Colourings

In this section, we introduce graph colourings and recall some known related results that are useful to our theory.

Definition 5.7.1 (Independent Set)**.** *Let $G = (V, E)$ be a graph. Then a set of vertices $t \subseteq V$ is called an* independent set *of G if*

$$\forall v_1, v_2 \in t : \{v_1, v_2\} \notin E. \tag{5.75}$$

The size of the largest independent set of G is called the independence number *of G and denoted by $\alpha(G)$. The set of all independent sets of G is denoted $I(G)$.*

Definition 5.7.2 (Graph Colouring)**.** *Let $G = (V, E)$ be a graph. Then a collection of k pairwise disjoint independent sets $H_1, \ldots, H_k \subseteq V$ such that $\bigcup_{j=1}^{k} H_j = V$ is called a* (proper) k-colouring *of G. The smallest number $k \in \mathbb{N}_0$ such that G admits a k-colouring is called the chromatic number of G and denoted by $\chi(G)$.*

Definition 5.7.3 (Clique)**.** *Let $G = (V, E)$ be a graph. Then a complete subgraph $C \subseteq V$ of size k is called a k-*clique *of G. The largest number $k \in \mathbb{N}_0$ such that G admits a k-clique is called the clique number of G and denoted by $\omega(G)$.*

Lemma 5.7.4. *For any graph G it holds that $\omega(G) \leq \chi(G)$. For any $n \in \mathbb{N}$, there exists a graph G_n such that $\chi(G_n) - \omega(G_n) \geq n$.*

Definition 5.7.5 (Fractional Graph Colouring)**.** *Let $G = (V, E)$ be a graph. For $b \in \mathbb{N}$, a collection of independent sets $H_1, \ldots, H_k \subseteq V$, such that for all $v \in V : |\{1 \leq j \leq k \mid v \in H_j\}| = b$, is called a k:b-*colouring *of G. The smallest number $k \in \mathbb{N}_0$ such that G admits a k:b-*colouring *is called the b-fold chromatic number of G and denoted by $\chi_b(G)$. Since $\chi_b(G)$ is subadditive we can define the* fractional chromatic number *of G as*

$$\chi_f(G) = \lim_{b \to \infty} \frac{\chi_b(G)}{b} = \inf_{b \in \mathbb{N}} \frac{\chi_b(G)}{b}. \tag{5.76}$$

Note that k:1-colourings are k-colourings and therefore $\chi_1(G) = \chi(G)$ which in turn implies that for all $b \in \mathbb{N}$ it holds that

$$\chi_f(G) \leq \chi_b(G) \leq \chi(G). \tag{5.77}$$

Lemma 5.7.6. *Let $G = (V, E)$ be a graph. Then $\chi_f(G)$ equals the smallest number $k \in \mathbb{R}_0^+$ such that there exists a probability distribution $\mathcal{D}$ over the independent sets $\mathcal{I}(G)$ such that for all $v \in V$ it holds that*

$$\Pr_{H \leftarrow \mathcal{D}}[v \in t] \geq \frac{1}{k}. \tag{5.78}$$

Definition 5.7.7 (Fractional Clique)**.** *Let $G = (V, E)$ be a graph. For $b \in \mathbb{N}$, a function $f : V \to \mathbb{N}_0$, such that for all $H \in \mathcal{I}(G) : \sum_{v \in H} f(v) \leq b$ and $\sum_{v \in V} = k$, is called a k:b-*clique *of G. The biggest number $k \in \mathbb{N}_0$ such that G admits a k:b-*clique *is called the b-fold clique number of G and denoted by $\omega_b(G)$. Since $\chi_b(G)$ is superadditive we*

can define the fractional clique number *of G as*

$$\omega_f(G) = \lim_{b\to\infty} \frac{\omega_b(G)}{b} = \sup_{b\in\mathbb{N}} \frac{\omega_b(G)}{b}. \tag{5.79}$$

Note that k:1-cliques are k-cliques and therefore $\omega_1(G) = \omega(G)$ which in turn implies that for all $b \in \mathbb{N}$ it holds that

$$\omega(G) \leq \omega_b(G) \leq \omega_f(G). \tag{5.80}$$

Lemma 5.7.8. *Let $G = (V, E)$ be a graph. Then $\omega_f(G)$ equals the biggest number $k \in \mathbb{R}_0^+$ such that there exists a probability distribution $\mathcal{D}$ over the vertices V such that for all $H \in \mathcal{I}(G)$ it holds that*

$$\Pr_{v\leftarrow\mathcal{D}}[v \in H] \leq \frac{1}{k}. \tag{5.81}$$

Both the fractional clique number $\omega_f(G)$ and the fractional chromatic number $\chi_f(G)$ are rational-valued solutions to dual linear programs. By the strong duality theorem, the two numbers must be equal.

Lemma 5.7.9. *For any graph G it holds that $\omega_f(G) = \chi_f(G)$.*

5.8 Appendix: General Parallel Repetition

We here show an alternative method for performing the same decomposition, by focusing solely on the error-detection amplification of classical input computations. We then recover the results above as a consequence of this generic amplification. We start as above by defining a compiler taking as input a trappified scheme and running it several times in parallel before thresholding over the outcomes of the individual decision functions.

Definition 5.8.1 (Parallel Repetition Compiler)**.** *Let $(\boldsymbol{P}, \preceq_G, \mathcal{P}, E_{\mathfrak{C}})$ be trappified scheme over a graph G for computation class $\mathfrak{C}$ with classical inputs, and let $n \in \mathbb{N}$ and $w \in [n-1]$. We define the Parallel Repetition Compiler that turns $\boldsymbol{P}$ into a trappified scheme $\boldsymbol{P}_{\|n}$ on G^n for computation class $\mathfrak{C}$ as follows:*

- *The set of trappified canvases is defined as $\{T_{\|n}\} = \boldsymbol{P}_{\|n} = \boldsymbol{P}^n$, the distribution $\mathcal{P}_{\|n}$ samples n times independently from $\mathcal{P}$;*

- *For each trappified canvas T' defined above and an output $t = (t_j)_{j\in n}$, we have:*

$$\tau'(t) = 0 \text{ if } \sum_{j=1}^{n} \tau_j(t_j) < w, \text{ and } 1 \text{ otherwise} \tag{5.82}$$

- *The partial ordering of vertices of G^n in $\boldsymbol{P}_{\|n}$ is given by the ordering $\preceq_G$ on every copy of G.*

- *Let $\mathsf{C} \in \mathfrak{C}$. Given a trappified canvas $T_{\|n} = \{T_j\}_{j\in[n]}$, the embedding algorithm $E_{\mathfrak{C},\|\mathfrak{n}}$ applies $E_{\mathfrak{C}}$ to embed C in each T_j.*

The next lemma relates the parameters above to the detection and insensitivity of the compiled scheme.

Lemma 5.8.2 (Exponential Detection and Insensitivity from Parallel Repetitions)**.** *Let $\boldsymbol{P}$ be a trappified scheme on graph G which ϵ-detects the error set $\mathcal{E}_1$, is δ-insensitive to $\mathcal{E}_2$ and perfectly insensitive to $\{\mathbb{I}\}$. For $n \in \mathbb{N}$ and $w \in [n-1]$, let $\boldsymbol{P}_{\|n}$ be the trappified scheme resulting from the compilation defined in Definition 5.8.1.*

Let $\mathcal{E}_{\geq k,\mathcal{E}}$ and $\mathcal{E}_{\leq k,\mathcal{E}}$ be defined as in Theorem 5.4.2. Let $k_1 > w/\epsilon$ and $k_2 < w/\delta$. Then, $\boldsymbol{P}_{\|n}$ $\epsilon_{\|n}$-detects $\mathcal{E}_{\geq k_1,\mathcal{E}_1}$ and is $\delta_{\|n}$-insensitive to $\mathcal{E}_{\leq k_2,\mathcal{E}_2}$ where:

$$\epsilon_{\|n} = \exp\left(-2\frac{(k_1\epsilon - w)^2}{k_1}\right), \tag{5.83}$$

$$\delta_{\|n} = \exp\left(-2\frac{(k_2\delta - w)^2}{k_2}\right). \tag{5.84}$$

Proof. As in the proof of the previous lemma, we denote Y a random variable counting the number of trappified canvases whose decision function rejects.

Let $\mathsf{E} \in \mathcal{E}_{\geq k_1,\mathcal{E}_1}$. We can lower-bound Y in the usual stochastic order by a (k_1, ϵ)-binomially distributed random variable $\tilde{Y}$. Then, Hoefffding's inequality yields directly that:

$$\Pr[Y < w] \leq \Pr\left[\tilde{Y} < w\right] \leq \exp\left(-2\frac{(k_1\epsilon - w)^2}{k_1}\right). \tag{5.85}$$

Similarly, let $\mathsf{E} \in \mathcal{E}_{\leq k_2,\mathcal{E}_2}$. Due to the perfect insensitivity of $\boldsymbol{P}$ to $\mathbb{I}$, we can now upper-bound Y in the usual stochastic order by a (k_2, δ)-binomially distributed random

variable $\tilde{Y}$. Then,

$$\Pr[Y \geq w] \leq \Pr\left[\tilde{Y} \geq w\right] \leq \exp\left(-2\frac{(k_1\epsilon - w)^2}{k_1}\right) \tag{5.86}$$

follows from a direct application of Hoefffding's inequality. □

Test and Computation Separation from Parallel Repetitions. We can now recover the case where some runs contain tests only while others consist only of the client's computation. This will be based on the following remark

Remark 5.8.3 (Pure Computation)**.** *A trappified scheme $\boldsymbol{P}$ may also only contain a single trappified canvas on graph $G = (V, E)$ such that $V_T = \emptyset$. This is the opposite case from Remark 5.2.14 above in the sense that all vertices serve to embed a computation of interest and none are devoted to detecting deviations. The decision function always accepts. Correctness is trivially (perfectly) satisfied for set $\{\mathbb{I}\}$, the detection and insensitivity are $\epsilon = 1$ and $\delta = 0$ respectively for any set.*

We then use Remarks 5.2.14 and 5.8.3, which allow us to define trappified schemes $\boldsymbol{P}_C$ and $\boldsymbol{P}_T$ on a graph G. $\boldsymbol{P}_C$ contains a single empty trappified canvas (with no trap) which can then be used to embed any computation on graph G, with 1-detection and 0-insensitivity to all Paulis. On the other hand, $\boldsymbol{P}_C$ may only contain pure traps with no space for embedding any computation, which ϵ-detects a set of errors $\mathcal{E}_1$ and is δ-insensitive to $\mathcal{E}_2$ (and perfectly insensitive to $\{\mathbb{I}\}$).

Then, Lemma 5.2.15 allows us to compose these two schemes via a probabilistic mixture noted $\boldsymbol{P}_M$. For parameters $d, s \in \mathbb{N}$ and $n = d + s$, $\boldsymbol{P}_M$ chooses schemes $\boldsymbol{P}_C$ and $\boldsymbol{P}_T$ with probabilities d/n and s/n respectively. The parameters for $\boldsymbol{P}_M$ are $\epsilon_M = (d + s\epsilon)/n = 1 - (1 - \epsilon)s/n$ and $\delta_M = s\delta/n$. It is also perfectly insensitive to $\{\mathbb{I}\}$.

Now the parallel repetition of Lemma 5.8.2 can be applied to $\boldsymbol{P}_M$ with parameters n, w to yield $\boldsymbol{P}_{\|n}$ with the following parameters:

$$\epsilon_{\|n} = \exp\left(-2\frac{(k_1(1 - (1 - \epsilon)s/n) - w)^2}{k_1}\right), \tag{5.87}$$

$$\delta_{\|n} = \exp\left(-2\frac{(k_2 s\delta/n - w)^2}{k_2}\right), \tag{5.88}$$

for values $k_1 > wn/(n - s + s\epsilon)$ and $k_2 < wn/s\delta$.

Notice that the bound on k_2 is identical to the one from Theorem 5.4.2, while the value for k_1 is smaller. The bounds obtained here are also simpler since they do not require an optimisation over the parameter χ, while still being exponential. However, they may be less sharp due to the degradation of ϵ_M (since we consider here the computation as trappified canvases which always accept). Finally, note that $\boldsymbol{P}_M$ is not strictly speaking a trappified scheme since it cannot embed computations with probability 1 as is required from Definition 5.2.7. However, all claims here are valid as they only consider the detection and insensitivity parameters, showing again the importance of separating these three properties. Consequently, the analysis in terms of correctness may slightly more complex since the number of computation runs is probabilistic, but can be bounded using tail bounds.

Chapter 6

Quantum Secure Multi-Party Computation

Secure multi-party computation (SMPC) protocols allow several parties that distrust each other to collectively compute a function on their inputs. In this chapter, we introduce a protocol that lifts classical SMPC to quantum SMPC in a composably and statistically secure way, even for a single honest party. Unlike previous quantum SMPC protocols, our proposal only requires very limited quantum resources from all but one party; it suffices that the weak parties, i.e. the clients*, are able to prepare single-qubit states in the $X - Y$ plane.*

The novel quantum SMPC protocol is constructed in a naturally modular way, and relies on a new technique for quantum verification that is of independent interest. This verification technique requires the remote preparation of states only in a single plane of the Bloch sphere. In the course of proving the security of the new verification protocol, we also uncover a fundamental invariance that is inherent to measurement-based quantum computing.

This chapter is based on the paper "Asymmetric Quantum Secure Multi-Party Computation With Weak Clients Against Dishonest Majority" [Kap+23] which is joint work with Theodoros Kapourniotis, Elham Kashefi, Luka Music, and Harold Ollivier.

6.1 Introduction

6.1.1 Motivation

Secure Multi-Party Computation (SMPC) protocols allow several parties who do not trust one another to collectively compute a function on their inputs. This question was first considered by Yao [Yao86] and has been developed extensively in various settings (see [CDN15] and references therein). Several security guarantees can be provided by such protocols depending on the setting: all parties can be on an equal footing, doing each their share of the computation, or one can handle the brunt of the computation while all others provide the data. In the first case, the security goal is to maximise the privacy of the data, while in the latter it extends to the privacy of the computation which is delegated to the server.

Practical computationally-secure protocols have been developed and implemented in commercial solutions for protecting classical multi-party computations. In the quantum case, several concrete protocols have been proposed (see § 6.1.2). In the circuit model, the state-of-the-art protocol [Dul+20] provides an information theoretic upgrade of classical SMPC that can withstand a dishonest majority. In the measurement-based model, where weakly quantum clients delegate their computation to a powerful server, the best protocol [KP17] does not provide verification of the computation and settles instead for blindness (i.e. privacy) of the data when there is no client-server collusion.

In this work, we show that this difference is not due to the asymmetry of the clients-server setting. We introduce for this specific situation a statistically secure lift of a classical SMPC protocol to a quantum one that provides blindness and verification for BQP computations. It remains secure so long as a single client is honest, thus withstanding possible collusions between dishonest clients and the server. Building on the techniques introduced in [Kap+22], its security is proved in the Abstract Cryptography (AC) framework. The protocol is highly modular and can tolerate a fixed amount of global noise during the quantum computation without aborting nor compromising statistical security. Additionally, it has no space overhead compared to an unprotected delegated computation, thereby allowing clients to use the server's full power to perform their desired computation, while security comes only at the price of a polynomial number of repetitions.

6.1.2 Related Work

Quantum SMPC is a long standing research topic in quantum cryptography, with several directions being explored in the past two decades.

The first one started with [CGS02]. Along with the introduction of the concept itself, it provided a concrete protocol for performing such computations in the quantum circuit model. It guarantees the security of the computation as long as the fraction of malicious parties does not exceed 1/6. This work has been later extended in [Ben+06], lowering the minimum number of honest players required for security to a strict majority.

The second focuses on the interesting edge case of two-party quantum computations. Several constructive results have been proposed in the circuit model. In [DNS10], a protocol was introduced and proven secure for quantum honest-but-curious adversaries. This restriction on the adversaries was removed in [DNS12] which proved security in the fully malicious setting and with negligible security bounds. The measurement-based model of quantum computation has also been considered for constructing secure two-party quantum computations as it provides a different set of tools than the circuit model. Verifiable Blind Quantum Computation (VBQC) first was introduced in [FK17] in this model, followed by optimised protocols [KW17b; KDK15]. In [KW17a] a protocol was proposed in this setting and proven secure against honest-but-curious adversaries. In [KMW17] this result was extended to fully malicious adversaries with inverse-polynomial security using the Quantum Cut-and-Choose technique. More recently, a round-optimal protocol was given in [Bar+21] based on Oblivious Transfer and LWE, showing that two-party quantum computation tasks can be performed in as little as three rounds in the CRS model, and two if quantum pre-processing is allowed.

A third set of results focuses on the composability of such protocols, as earlier results didn't satisfy this property. Bit commitment was shown to be complete in the Quantum Universal Composability framework of [Unr10], meaning that it is sufficient for constructing quantum or classical SMPC if parties have access to quantum channels and operations. This result was later extended in [Feh+13; Dup+16], which gives a full analysis of feasibility and completeness of cryptographic primitives in a composable setting.

More recently, building on these previous works, new concrete protocols have been proposed to decrease the restrictions on adversaries while also providing composable security. In the circuit model, a composably-secure protocol has been introduced in [Dul+20]. It is an extension of [DNS12] that is able to cope with a dishonest majority,

but which relies on a complete graph for quantum communication and on a large number of quantum communication rounds together with powerful quantum participants. In the MBQC model, [KP17] describes a protocol that is composable, can tolerate a dishonest majority and allows the clients to delegate the quantum computation to a powerful server. Its security is an information-theoretic upgrade of the classical SMPC primitive used for constructing the protocol. It is however limited by the absence of verifiability of outputs and the impossibility to tolerate client-server collusions. Other protocols have been proposed in alternative models or with different trust assumptions such as [Hou+18; LRW20]. Finally, recent protocols for secure delegated quantum computations can be run even by purely classical clients. These have been lifted to a multi-client setting in [Bar21] while at the same time optimising the number of classical rounds of communication. This is however at the cost of a larger computation space on the server's device, which needs to be able to perform QFHE computations of functions large enough to be computationally-secure.

A subset of the authors proposed an earlier protocol for QSMPC [Kap+21] which comprised a blind pre-computation step meant to produce a resource state that could then be used to perform VBQC. This pre-computation turned out to be vulnerable to an attack that can be applied blindly by the server while having an effect only on some specific types of qubits thereby compromising security of the whole protocol. While the present work is a complete redesign of the protocol that shows improved performance, we include in § 6.6 an in-depth analysis of the shortcommings of the previous construction. This might be a useful tool to revisit earlier work where a similar blind pre-computation step is used.

6.1.3 Overview of the Protocol and Results

In this chapter, we consider the setting where several weakly quantum clients want to securely delegate their quantum computation to a powerful server. The proposed construction turns a single-client MBQC-based protocol into a multi-party one. More precisely, we use single-client Secure Delegated Quantum Computing (SDQC) protocols obtained through the techniques presented in [Kap+22]. Such protocols interleave several computation rounds and test rounds, the latter of which correspond to stabiliser measurements of the MBQC resource graph-state used to perform the computations. In such a protocol, the client must perform two different tasks. First, it has to prepare encrypted single-qubit states and send them to the server. This prevents the server from

distinguishing computation and test rounds and also hides the client's data. Then, the client uses the classical encryption key as well as the measurement outcomes reported by the server to classically drive the computations and tests performed by the server on these encrypted qubits. Hence, turning this protocol into a multi-party one amounts to finding (i) an appropriate single-client SDQC protocol that will (ii) be composed with a secure collaborative remote state preparation for the single qubit encrypted states and that will (iii) be driven collaboratively to perform and verify the desired computation.

In § 6.2, we describe a single-client SDQC Protocol using only $|+_\theta\rangle = (|0\rangle + e^{i\theta}|1\rangle)/\sqrt{2}$ states, based on the generic single-client SDQC Protocol of [Kap+22]. This was an open question in the field as all previous SDQC protocols in the MBQC framework with a formal security analysis use computational basis states (called dummies) to isolate single qubits in the computation graph. These remain unchanged if the server is honest and can be used as traps to detect deviations. To overcome this restriction, we must ideally find a generating set of stabilisers of the graph state for the client's computation that can be written with $\mathbb{I}$, X and Y Paulis only.

However, while it is possible to construct $N-1$ independent stabilisers of this form – where N denotes the number of vertices of the graph – it seems that the stabiliser which consists of Z operators on odd-degree vertices of the graph cannot be generated. This therefore corresponds to a server's deviation which cannot be caught by our tests on graphs containing odd degree nodes. If this attack would corrupt the client's computation, the whole protocol would be insecure. Fortunately, this is not the case for classical input/output computations. Indeed, we prove that this deviation corresponds to a server which has chosen a different orientation of the Z axis compared to the client. Because inputs are prepared in the $X-Y$ plane and outputs are projected onto it, we show that this has no effect on the outcome of the computation. As a consequence, it is not necessary to detect this specific deviation by the server to verify the computation. This proves that the generic single-client SDQC Protocol of [Kap+22] can be used to produce secure dummyless protocols.[1]

Theorem (Informal)**.** *For any graph G, there exists a single-client statistically secure SDQC protocol in the Abstract Cryptography framework that requires the client to only prepare states in the $X-Y$ plane.*

We then focus on turning this new single-client protocol into a multi-party one.

[1]Note that here has been a previous protocol for dummyless verification [FKD18], whose security analysis didn't take into account the above deviation. Our proof of invariance of MBQC to this specific error shows that this deviation does not constitute a security threat to the protocol in [FKD18].

In § 6.3, we introduce a Collaborative Remote State Preparation (CRSP) protocol. We show that this gadget (Protocol 6) securely implements Remote State Preparation (Resource 4), which allows a classical party request any $|+_\theta\rangle$ state to be prepared on the server's device with the help of clients preparing single qubit states in the $X-Y$ plane.

Theorem (Informal)**.** *The CRSP gadget is a statistically secure implementation of the Remote State Preparation Resource in the Abstract Cryptography framework.*

The second set of tasks in the single-client protocol, i.e. choosing the measurement angles of the various computation and test rounds according to the states prepared using CRSP, only involve classical computations. These can be performed using a composably secure classical SMPC.[2]

In § 6.4, we compose the CRSP gadget, classical SMPC, and the dummyless SDQC protocol into a complete quantum SMPC protocol (Protocol 7). Its outline is:

1. The clients use the CRSP gadget to prepare $|+_\theta\rangle$ states on the server's side.

2. They use the classical SMPC together to drive and verify the single-client SDQC protocol.

3. Upon acceptance, the results and decryption keys are sent by the classical SMPC to each client.

The security proof relies on the composable security of all three ingredients. Because the CRSP gadget and the dummyless protocol are statistically secure, this is a direct upgrade of classical to quantum SMPC.

Theorem (Informal)**.** *Composable classical SMPC can be lifted to perform robust quantum SMPC for* BQP *computations in a statistically secure way, such that all parties but one are restricted to singe-qubit preparations.*

We note that this protocol requires no additional resources in terms of hardware or quantum communication from the client's side compared to the single-client protocol. The server only needs to be able to perform the CRSP gadget in addition to the operations required by the single-client protocol.

[2]The Abstract Cryptography framework used in this work is equivalent to the Quantum Universal Composability (Q-UC) Model of [Unr10] if a single Adversary controls all corrupted parties – which is the case here. Therefore any Classical SMPC protocol which is secure in the Q-UC model can be used to instantiate this functionality.

6.1.4 Discussion

In the course of constructing our protocol, we have built two new ingredients that we believe are of independent interest.

The first one is the Collaborative Remote State Preparation gadget. Its main feature is to provide some privacy amplification for the classical-quantum correlations that clients share with the server. Interestingly, we give evidence that it is hard to construct a generic gadget that would have similar features for correlations outside of a single plane of the Bloch sphere, while retaining its usefulness for cryptographic purposes. We leave it as an open question to prove a full no-go theorem in the Abstract Cryptography framework to further explore what seems to be a deep difference between classical and quantum input-output computations. Note also that this work supersedes a previous effort to construct a quantum SMPC protocol in the clients-server setting with quantum input and outputs. The proposed construction was similar in spirit with a collaborative remote state preparation gadget that allowed to prepare encrypted $X - Y$ plane states but also dummies. However, we give an attack on multiple approaches which were explored to perform this task, further strengthening the belief that such cryptographic protocols are hard if not impossible to construct.

The second new ingredient of our proof is the first dummyless SDQC protocol. Outside of the specific purpose of quantum SMPC, it exemplifies the usefulness of the general tests that were introduced in [Kap+22]. By reducing the requirements on the client side, it also possibly decreases a source of errors in physical implementations as it is not uncommon that rotations around one specific axis of the Bloch sphere are notably easier to perform than others. We also strongly believe that similar approaches, where traps are tailored to specific settings, will find applications in the future. Additionally, we show that while dummyless tests were not enough to detect all deviations, it is possible to nonetheless verify computations thanks to an as of now unknown invariance in MBQC. This raises the question of whether it is possible to do this on purpose, and engineer an invariance in order to lighten the constraints on the error-detection scheme that the traps implement.

Finally, note that because all SDQC protocols constructed from the generic protocol of [Kap+22] are robust to a fixed amount of global noise, so is our new multi-party protocol. While not being enough to scale to large quantum computations, it opens the possibility to implement experimental proof-of-concepts without resorting to error correction on near term devices.

6.1.5 Organisation of this chapter

In § 6.2 we construct a single-client SDQC Protocol using only $|+_\theta\rangle = (|0\rangle + e^{i\theta}|1\rangle)/\sqrt{2}$ states. §§ 6.2.1-6.2.3 construct a family of such schemes and prove their security, while § 6.2.4 provides optimised protocols for various classes of MBQC resource graph-states. In § 6.3, we introduce a Collaborative Remote State Preparation (CRSP) protocol and prove its security in the AC framework. In § 6.4, we compose the CRSP Protocol, the dummyless SDQC Protocol and a classical SMPC into a complete quantum SMPC Protocol (Protocol 7) for BQP computations. In § 6.5, we provide an in-depth comparison with other protocols, give arguments justifying the proposed construction – especially the need for a dummyless SDQC Protocol – and discuss some open questions.

Some preliminary notation and material relevant for this chapter can be found in the corresponding sections of Chapter 2: Abstract Cryptography in § 2.1, MBQC computation in § 2.2 and the Universal Blind Quantum Computation (UBQC) Protocol in § 2.3. A detailed analysis of a previous attempt at constructing quantum SMPC for weakly quantum clients is provided in § 6.6.

6.2 Verification with States in a Single Plane

6.2.1 A Framework for Verification

The goal of the protocol presented in this section is to construct the Secure Delegated Quantum Computation Resource 3 (SDQC), introduced by [Dun+14]. It allows a single Client to run a quantum computation on a Server so that the Server cannot corrupt the computation and doesn't learn anything besides a controlled leakage l_ρ about the Client's computation and input. The value of l_ρ, as a function of inputs and computation, is specified by each protocol.

Several protocols implementing this resource have been constructed in the past [GKK19]. Yet, none has the ability to provide negligible statistical security while having a client sending *states in a single plane*. To achieve this, we use the framework from [Kap+22] which neatly separates the various ingredients required to implement SDQC. We start by briefly summarising the ingredients which are relevant to this chapter.

Reduction to Pauli Deviations. Using the UBQC Protocol 2 to delegate computations from Client to Server hides the operations which the Client wishes to delegate.

Resource 3 Secure Delegated Quantum Computation with Classical Inputs and Outputs

Inputs:

- The Client inputs a bit-string x and the classical description of a unitary U.
- The Server chooses whether or not to deviate. This interface is filtered by two control bits (e, c).

Computation by the Resource:

1. If $e = 1$, the Resource sends the leakage l_ρ to the Server's interface and awaits further input from the Server; if it receives $c = 1$, the Resource outputs Abort at the Client's output interface.
2. If $c = 0$, it outputs $O = \mathcal{M}_C \circ \mathsf{U}\,|x\rangle$ at the Client's output interface, where $\mathcal{M}_C$ is a computational basis measurement.

The encoding scheme of UBQC naturally imposes a Pauli twirl on any deviation and hence any attacks by the Server can always be decomposed as a convex combination of Pauli operators acting on the qubits of the graph just before performing the X-basis measurement. Because X Pauli operators applied in this fashion have no effect on the computation, as they are absorbed by the X-basis measurement, we can focus on convex combinations of deviations of the form $\bigotimes_{v\in V} \mathsf{Z}(v)^{e(v)}$ where the values of $e(v)$ are chosen by the Server and $\mathsf{Z}(v)$ applies the Pauli Z to qubit v. Such deviation are equivalent to flipping the measurement outcome for vertices where $e(v) = 1$.

General Strategy for Robust Verification. Once all operations delegated to the Server are blind a general strategy for robust and secure computation follows from the intuition that (i) correctness is obtained by accepting with overwhelming probability in the absence of deviation, (ii) security derives from the ability of the protocol to detect with overwhelming probability all deviations that potentially affect the computation, and (iii) robustness follows from accepting additional deviations which have, with overwhelming probability, no effect on the computation.

Generic Trappified Schemes for Classical I/O. With this strategy in mind, a whole class of protocols for verifying BQP computations can be easily described. Their flexible design is able to accommodate objectives that go beyond security, e.g. for instance the absence of dummy qubits. These protocols work by performing separate rounds which are indistinguishable from the Server's point of view, some implementing tests, and others computing C, the Client's target computation. More precisely, s test rounds and d computation rounds are delegated to the Server using the UBQC Protocol 2, with

the requirement that they share the same graph G and the same order $\preceq_G$ for measuring the qubits.

Each test round is sampled uniformly at random from a set $\boldsymbol{P}$ of possible traps called a *trappified scheme*. They each consist of an input state σ which is a tensor product of single-qubit states, one for each vertex in the graph G, a measurement pattern T, and a binary decision function τ. The test round is accepted when the decision function outputs 0 when evaluated on the measurement results returned by the Server for this trap. It is rejected when the output is 1. The d computation rounds correspond to repeating d times the target computation C on the target input chosen by the Client using the graph G. The outputs of these computations are then combined through a majority vote. When all rounds have been executed, the Client accepts if less than a fixed fraction of test rounds reject. In this case, the output of the protocol is the result of the majority vote. The formal protocol is described in Protocol 5.

Security Conditions for Trappified Schemes with Classical I/O. The analysis of the security and robustness properties in the Abstract Cryptography framework for the resulting protocol depends on two sets of Pauli operators defined relatively to $\boldsymbol{P}$: the set of detectable deviations and the set of deviations to which $\boldsymbol{P}$ is insensitive. These rely on the following definitions, where we use $\mathcal{T}$ to denote the probability of the measurement outcomes for a trap T in $\boldsymbol{P}$ and $\mathsf{E} \circ \mathcal{T}$ to denote the probability distribution of measurement outcomes when the deviation E is applied to T.

Definition 6.2.1 (Pauli Insensitivity)**.** *We say that the trappified scheme $\boldsymbol{P}$ is δ-*insensitive to $\mathcal{E} \subset \mathcal{G}_V$ *if:*

$$\forall \mathsf{E} \in \mathcal{E},\ \sum_{T \in \boldsymbol{P}} \Pr_{\substack{T \sim \mathcal{P} \\ t \sim \mathsf{E} \circ \mathcal{T}}} [\tau(t) = 0, T] \geq 1 - \delta. \tag{6.1}$$

Definition 6.2.2 (Pauli Detection)**.** *We say that a trappified scheme $\boldsymbol{P}$ ϵ-*detects $\mathcal{E} \subset \mathcal{G}_V$ *if:*

$$\forall \mathsf{E} \in \mathcal{E},\ \sum_{T \in \boldsymbol{P}} \Pr_{\substack{T \sim \mathcal{P} \\ t \sim \mathsf{E} \circ \mathcal{T}}} [\tau(t) = 1, T] \geq 1 - \epsilon. \tag{6.2}$$

Definition 6.2.3 (Pauli Correctness (Informal))**.** *We say that a computation is correct on deviation E if the output distribution is the same whether the deviation is applied or not.*

Protocol 5 Trappified Delegated Blind Computation

Public Information:

- $G = (V, E, I, O)$, a graph with input and output vertices I and O respectively;
- $\boldsymbol{P}$, a trappified scheme on graph G;
- $\preceq_G$, a partial order on the set V of vertices;
- N, d, w, parameters representing the number of runs, the number of computation runs, and the number of tolerated failed tests.

Client's Inputs: A set of angles $\{\phi_i\}_{i\in V}$ and a flow f which induces an ordering compatible with $\preceq_G$.

Protocol:

1. The Client samples uniformly at random a subset $C \subset [N]$ of size d representing the runs which will be its desired computation, henceforth called computation runs.
2. For $k \in [N]$, the Client and Server perform the following:
   (a) If $k \in C$, the Client sets the computation for the run to its desired computation $(\{\phi_i\}_{i\in V}, f)$. Otherwise, the Client samples a test (T, σ, τ) from the trappified scheme $\boldsymbol{P}$.
   (b) The Client and Server blindly execute the run using the UBQC Protocol 2.
   (c) If it is a test, it uses τ on the measurement results to decide whether the test passed or not.
3. At the end of all runs, let x be the number of failed tests. If $x \geq w$, the Client rejects and outputs $(\perp, \mathsf{Rej})$.
4. Otherwise, the Client accepts the computation. It performs a majority vote on the output results of the computation runs and sets the result as its output.

The virtue of defining these properties is that the sets of deviations above can be characterised efficiently and yield correctness and security with negligible errors for the overall protocol:

Theorem 6.2.4 (Security of Protocol 5, Combining Theorems 8 and 13 from [Kap+22])**.** *Let $\mathfrak{C}$ be a set of classical* BQP *computations on graph G. Let $\boldsymbol{P}$ be a trappified scheme on graph G that ϵ-detects a set of Pauli deviations $\mathcal{E}_1$ and is δ-insensitive to $\mathcal{E}_2$ and perfectly insensitive to $\mathbb{I}$. Assume that all computations in a set $\mathfrak{C}$ are correct on $\mathcal{G}_V \setminus \mathcal{E}_1$. Let $n = s + d$ for d and s proportional to n, and c the bounded error of* BQP*. Let w be the maximum number of test rounds allowed to fail, chosen such that $w < \frac{2c-1}{2c-2} s(1-\epsilon)$.*

Then Protocol 5 $\eta(n)$-constructs the Secure Delegated Quantum Computation Resource 3 for computations in set $\mathfrak{C}$ in the Abstract Cryptography framework, where the

leak is defined as $l_\rho = (\mathfrak{C}, G, \boldsymbol{P}, \preceq_G)$*, for* $\eta(n)$ *negligible in* n*.*

Note that the value of $\eta(n)$ heavily depends on the value of δ and ϵ, in particular via the coefficient in the exponential. This means that it is crucial to minimise these detection and insensitivity errors.

Notice also that w is also reliant on ϵ, and minimising this error also allows the protocol to tolerate more honest errors before aborting. This noise-robustness of Protocol 5 can be characterised as follows.

Theorem 6.2.5 (Noise-Robustness of Protocol 5, Combining Theorems 9 and 13 from [Kap+22])**.** *For the same parameter choices as in Theorem 6.2.4, assume an execution of Protocol 5 with an honest-but-noisy Server such that* p *is the probability that less than* $\frac{w}{s\delta}n$ *rounds are affected by a Pauli error. Then the Client accepts the outcome with probability* $(1-p)(1-\delta')$*, for* δ' *negligible in* n*.*

Since the protocol is secure, we can then guarantee that, if the client accepts, the outcome is also correct up to negligible total variational distance. This means that for machines with a constant amount of global noise below a certain bound, our protocol accepts and yields the correct result with overwhelming probability.

Traps from Stabiliser Tests. As a result, the performance of Protocol 5 is governed by the choice of s, d, w defined above, together with the error detection and insensitivity capabilities of traps in $\boldsymbol{P}$. Ref. [Kap+22] § 6.1 shows how to construct general traps from subset stabiliser testing. Indeed, let S be the stabiliser group for $|G\rangle\langle G|$ the graph state associated to G, and consider $\{\mathsf{S}_v = \mathsf{X}(v)\bigotimes_{(v,w)\in E}\mathsf{Z}(w), v \in V\}$ the set of canonical generators of S. One can then associate a trap to each $\mathsf{R} \in \mathrm{S}$ by (i) having the Client prepare a $+1$ eigenspace of R as input, and (ii) delegating to the Server the computation consisting of measuring R using the UBQC Protocol 2. An accepted trap then corresponds to the measurement of R returning the $+1$ eigenvalue.

For the preparation, the client sets each qubit $v \in V$ in the $+1$ eigenstate of $\mathsf{R}(v)$ with $\mathsf{R}(v)$ being uniquely defined by:

$$\mathsf{R} = \bigotimes_{v\in V}\mathsf{R}(v), \tag{6.3}$$

$$\mathsf{R}(v_0) \in \{\pm 1\} \times \{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}, \text{ for } v_0 = \mathrm{argmin}_{v\in V}\,\mathsf{R}(v) \neq \mathsf{I}, \tag{6.4}$$

$$\mathsf{R}(v) \in \{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}, \text{ for } v \in V \setminus v_0. \tag{6.5}$$

This corresponds to preparing a $+1$ eigenstate of the group generated by $\{\mathsf{R}(v), v \in V\}$ which contains R hence satisfying (i) above.

For the delegated computation consisting of measuring R, the Client simply instructs the Server to measure each qubit in the X-basis, getting outcome $t(v)$. The motivation for these measurements is better understood by examining to which observable they correspond on the inputs provided by the Client. To this end, one can conjugate each $\mathsf{X}(v)$ by $\prod_{(v,w)\in E} \mathsf{CZ}_{(v,w)}$, the entangling operation that the Server performs prior to the measurement. A simple stabiliser computation shows that $\mathsf{X}(v)$ is mapped to S_v. That is, measuring $\mathsf{X}(v)$ after the entangling operation corresponds to measuring S_v on the inputs provided by the client. As R is uniquely defined as $\prod_{v\in\mathbb{1}_\mathsf{R}} \mathsf{S}_v$ for some set $\mathbb{1}_\mathsf{R} \subset V$, and because S is abelian, the outcome of R on the input state provided by the client is the binary sum of the outcomes of S_v. Using the above correspondence for measurements of S_v on the inputs, one concludes that $\bigoplus_{v\in\mathbb{1}_\mathsf{R}} t(v)$ determines the outcome of the measurement of R on the inputs provided by the Client. Combining the preparation and the measurement, the Client therefore expects that for an honest Server, $\bigoplus_{v\in\mathbb{1}_\mathsf{R}} t(v) = 1$, thereby fulfilling (ii) above.

The freedom in choosing which R's to include in the trappified scheme $\boldsymbol{P}$ will be at the core of constructing dummyless verification protocols.

6.2.2 A Natural Invariance of MBQC with Classical Input and Output

In MBQC, computation qubits, i.e. $v \in O^c$, are measured in the $\left|\pm_{\phi'(v)}\right\rangle$ basis, where $\phi'(v) \in \Theta = \left\{\frac{k\pi}{4}\right\}_{k\in\{0,\ldots,7\}}$ is defined by the pattern used for the computation. As a result, the computation is invariant under rotations around the $\phi'(v)$ axis in the $X-Y$ plane just before the measurement. The reason is that such rotations leave the projectors $\left|+_{\phi'(v)}\right\rangle\!\left\langle+_{\phi'(v)}\right|$ and $\left|-_{\phi'(v)}\right\rangle\!\left\langle-_{\phi'(v)}\right|$ untouched so that it does not affect the probabilities of the outcomes of a measurement in the $\left|\pm_{\phi'(v)}\right\rangle$ basis. This property is well known and is actively used in the proof of security of the UBQC protocol as it allows to fully twirl the deviation of the server on computation qubits.

If one not only considers local unitary transformations but more generally local invertible transformations, then MBQC is also invariant under reflections through the $X-Y$ plane for $v \in O^c$. The reason is similar to the one given above: such transformations do not change the projectors onto the $\left|\pm_{\phi'(v)}\right\rangle$ basis and hence do not affect probability distributions of measurements in the $\left|\pm_{\phi'(v)}\right\rangle$ basis.

We will now explore the latter invariance in the special case of classical input classical output computations where it naturally extends to the result of the computation itself, as in such case all qubits are measured in the $X-Y$ plane.

Lemma 6.2.6. *For matrices $\rho = \sum_{\mathsf{P}\in\{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}^{\otimes n}} \alpha_{\mathsf{P}}\mathsf{P}$ decomposed in the Pauli basis, let F_A be the linear map that applies the reflection through the $X-Y$ plane for all vertices in $A \subset V$, defined as*

$$F_A(\rho) = \sum_{\mathsf{P}\in\{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}^{\otimes n}} (-1)^{\mathrm{zwt}_A(\mathsf{P})}\alpha_{\mathsf{P}}\mathsf{P}, \tag{6.6}$$

where $\mathrm{zwt}_A(\mathsf{P}) = |\{v \in A | \mathsf{P}_v = \mathsf{Z}\}|$ counts the number of vertices in A on which P equals the Pauli Z. Then, MBQC is invariant under F_A when applied right before the $\left|\pm_{\phi'(v)}\right\rangle$ measurements.

Proof. The probability to obtain the all-zero outcome when measuring all qubits $v \in O^c$ of a state ρ in the $\left|+_{\phi'(v)}\right\rangle\!\left\langle+_{\phi'(v)}\right|$-bases is given by

$$\mathrm{Tr}\left(\left(\mathrm{id}_O \otimes \bigotimes_{v\in O^c} \left|+_{\phi'(v)}\right\rangle\!\left\langle+_{\phi'(v)}\right|\right)\rho\right). \tag{6.7}$$

Decomposing the above expression in the Pauli basis yields

$$\begin{aligned}
&\mathrm{Tr}\left(\left(\sum_{\mathsf{P}'\in\{\mathsf{I},\mathsf{X},\mathsf{Y}\}^{\otimes n}} \beta_{\mathsf{P}'}\mathsf{P}'\right)\left(\sum_{\mathsf{P}\in\{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}^{\otimes n}} \alpha_{\mathsf{P}}\mathsf{P}\right)\right) \\
&= \sum_{\mathsf{P}'\in\{\mathsf{I},\mathsf{X},\mathsf{Y}\}^{\otimes n}} \sum_{\mathsf{P}\in\{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}^{\otimes n}} \beta_{\mathsf{P}'}\alpha_{\mathsf{P}}\,\mathrm{Tr}\left(\mathsf{P}'\mathsf{P}\right) = \sum_{\mathsf{P}\in\{\mathsf{I},\mathsf{X},\mathsf{Y}\}^{\otimes n}} \beta_{\mathsf{P}}\alpha_{\mathsf{P}}2^{|V|}.
\end{aligned} \tag{6.8}$$

Calculating the same probability for the all-zero outcome when measuring after applying F_A yields

$$\begin{aligned}
&\mathrm{Tr}\left(\left(\mathrm{id}_O \otimes \bigotimes_{v\in O^c} \left|+_{\phi'(v)}\right\rangle\!\left\langle+_{\phi'(v)}\right|\right)F_A(\rho)\right) \\
&= \mathrm{Tr}\left(\left(\sum_{\mathsf{P}'\in\{\mathsf{I},\mathsf{X},\mathsf{Y}\}^{\otimes n}} \beta_{\mathsf{P}'}\mathsf{P}'\right)\left(\sum_{\mathsf{P}\in\{\mathsf{I},\mathsf{X},\mathsf{Y},\mathsf{Z}\}^{\otimes n}} (-1)^{\mathrm{zwt}_A(\mathsf{P})}\alpha_{\mathsf{P}}\mathsf{P}\right)\right) \\
&= \sum_{\mathsf{P}\in\{\mathsf{I},\mathsf{X},\mathsf{Y}\}^{\otimes n}} \beta_{\mathsf{P}}\alpha_{\mathsf{P}}2^{|V|},
\end{aligned} \tag{6.9}$$

and therefore the same value. By an analogous argument, the probabilities for any other

outcome coincide as well. □

Note that $F_A(\rho)$ might not always be a physical state. As a result, if $|G\rangle\langle G|$ denotes the graph state used to implement classical input classical output MBQC on G, one has:

$$|G\rangle\langle G| = \frac{1}{2^{|V|}} \sum_{\mathsf{S}\in\mathcal{S}} \mathsf{S}, \tag{6.10}$$

for $\mathcal{S}$ the stabiliser group of the graph state, so that for any $\mathsf{S}' \in \mathrm{S}$ we have:

$$\mathrm{Tr}(\mathsf{S}' |G\rangle\langle G|) = \frac{1}{2^{|V|}} \sum_{\mathsf{S}\in\mathrm{S}} \mathrm{tr}(\mathsf{S}'\mathsf{S}) = 1. \tag{6.11}$$

In turn, this implies that

$$\mathrm{Tr}(\mathsf{S}' F_A(|G\rangle\langle G|)) = \frac{1}{2^{|V|}} \sum_{\mathsf{S}\in\mathrm{S}} (-1)^{\mathrm{zwt}_A(\mathsf{S})} \mathrm{Tr}(\mathsf{S}'\mathsf{S}) = (-1)^{\mathrm{zwt}_A(\mathsf{S}')}. \tag{6.12}$$

If $F_A(|G\rangle\langle G|)$ was a physical state, Equation (6.12) would imply that it would be stabilised by $(-1)^{\mathrm{zwt}_A(\mathsf{S})}\mathsf{S}$ for all $\mathsf{S} \in \mathrm{S}$. The group structure of stabilisers would then imply that it is also stabilised by the operator $(-1)^{\mathrm{zwt}_A(\mathsf{S})+\mathrm{zwt}_A(\mathsf{S}')}\mathsf{S}\mathsf{S}'$ for all $\mathsf{S}, \mathsf{S}' \in \mathcal{S}$, and hence $\mathrm{zwt}_A(\mathsf{S}\mathsf{S}') \equiv \mathrm{zwt}_A(\mathsf{S}) + \mathrm{zwt}_A(\mathsf{S}') \pmod 2$.

However, for $A \subsetneq V$, $\mathrm{zwt}_A(\cdot)$ does not in general satisfy the above equation. More precisely, take $(v, w) \in E$, the stabiliser $\mathsf{S}_v\mathsf{S}_w$ will then satisfy $\mathrm{zwt}(\mathsf{S}_v\mathsf{S}_w) \equiv \mathrm{zwt}(\mathsf{S}_v) + \mathrm{zwt}(\mathsf{S}_w) - 1 \pmod 2$. This is because the overlap of S_v and S_w at v will always remove a single Z coming from S_w, while if the two stabilisers overlap at some other Z location in A this will remove 2 from the weight.

Conversely[3], setting $A = V$, then indeed $\mathrm{zwt}_V(\mathsf{S}\mathsf{S}') \equiv \mathrm{zwt}_V(\mathsf{S}) + \mathrm{zwt}_V(\mathsf{S}') \pmod 2$ for all $\mathsf{S}, \mathsf{S}' \in \mathcal{S}$. Moreover, it is possible to find a unitary transformation that has the same effect as F_A on $|G\rangle\langle G|$, implying that $F_A(|G\rangle\langle G|)$ is then a physical state, as witnessed by the following lemma.

Lemma 6.2.7. *For any graph $G = (V, E)$ it holds that $F_A(|G\rangle\langle G|) = \mathsf{U}\,|G\rangle\langle G|\,\mathsf{U}^\dagger$, where*

$$\mathsf{U} = \prod_{\substack{v\in V,\\ \deg v \equiv 1 \,(\mathrm{mod}\ 2)}} \mathsf{Z}_v \tag{6.13}$$

describes the application of Z's to all odd-degree vertices of G.

[3]More generally, for disconnected graphs this holds if and only if A is a connected component or a union of connected components.

Proof. It will be useful to rewrite the stabilisers of $|G\rangle\langle G|$ as follows. For every $\mathsf{S} \in \mathrm{S}$, there exists exactly one subset of vertices $V_{\mathsf{S}} \subset V$ such that

$$\mathsf{S} = \prod_{v \in V_{\mathsf{S}}} \mathsf{S}_v. \tag{6.14}$$

We start with the right side of the equation:

$$\mathsf{U} |G\rangle\langle G| \mathsf{U}^\dagger = \mathsf{U} \left(\frac{1}{2^{|V|}} \sum_{\mathsf{S} \in \mathrm{S}} \mathsf{S} \right) \mathsf{U}^\dagger = \frac{1}{2^{|V|}} \sum_{\mathsf{S} \in \mathrm{S}} \mathsf{U} \left(\prod_{v \in V_{\mathsf{S}}} \mathsf{S}_v \right) \mathsf{U}^\dagger. \tag{6.15}$$

Complementing $\mathsf{U}^\dagger\mathsf{U}$ terms, this expression gives:

$$\frac{1}{2^{|V|}} \sum_{\mathsf{S} \in \mathrm{S}} \prod_{v \in V_{\mathsf{S}}} \mathsf{U}\mathsf{S}_v\mathsf{U}^\dagger. \tag{6.16}$$

It is easy to verify that $\mathsf{U}\mathsf{S}_v\mathsf{U}^\dagger = (-1)^{\mathrm{zwt}_V(\mathsf{S}_v)}\mathsf{S}_v$ because of the particular structure of U, and hence the above expression equals

$$\frac{1}{2^{|V|}} \sum_{\mathsf{S} \in \mathcal{S}} \prod_{v \in V_{\mathsf{S}}} (-1)^{\mathrm{zwt}_V(\mathsf{S}_v)} \mathsf{S}_v. \tag{6.17}$$

Exploiting the additivity of $\mathrm{zwt}_V(\cdot)$, we arrive at

$$\frac{1}{2^{|V|}} \sum_{\mathsf{S} \in \mathcal{S}} (-1)^{\sum_{v \in V_{\mathsf{S}}} \mathrm{zwt}_V(\mathsf{S}_v)} \mathsf{S} = \frac{1}{2^{|V|}} \sum_{\mathsf{S} \in \mathcal{S}} (-1)^{\mathrm{zwt}_V(\mathsf{S})} \mathsf{S} = F_V(|G\rangle\langle G|), \tag{6.18}$$

which concludes the proof. □

Combining the statements of Lemma 6.2.6 and Lemma 6.2.7, we finally arrive at the following result, capturing the inherent invariance of classical I/O MBQC to one specific nontrivial error.

Lemma 6.2.8. *Let $G = (V, E)$ be a graph and U be the unitary operation given by*

$$\mathsf{U} = \prod_{\substack{v \in V, \\ \deg v \equiv 1 \,(\mathrm{mod}\ 2)}} \mathsf{Z}_v, \tag{6.19}$$

describing the application of Z's to all odd-degree vertices of G. For MBQC on G with classical input and output, the application of U before the measurements has no effect on the results of the computation.

Summarising the results of this section, for any classical-input classical-output MBQC there exists a non-trivial and non-stabiliser deviation that has no influence on the results of the computation. It is important to bear in mind the harmlessness of this error when constructing a verification scheme, as dummyless stabiliser tests will – by construction – not be able to detect it.

6.2.3 Dummyless Verification

We now arrive at the core of this section: designing single-round traps restricted to preparing states in the $X-Y$ plane. Using the construction of traps from Section 6.2.1, it amounts to finding a set of stabilisers of $|G\rangle\langle G|$ that are only made out of I, X, Y tensor products.

More precisely, we show that

Lemma 6.2.9. *For any $G=(V,E)$, consider the graph state $|G\rangle$ and its stabiliser group* S. *Then, it is always possible to find $|V|-1$ generators of* S *that are tensor products of* I, X *and* Y *only.*

Proof. We proceed constructively and exhibit a set of $|V|-1$ generators of R, subgroup of S, and show that $|\mathrm{R}|=2^{|V|-1}$.

We start with one such stabiliser, $\mathsf{R}_{\text{full}}=\prod_v \mathsf{S}_v$. This follows simply from

$$\mathsf{R}_{\text{full}}(v)=\mathsf{X}\mathsf{Z}^{\deg(v)}, \tag{6.20}$$

as for qubit v, S_v contributes to the X and all neighbours contribute a Z each. Additionally, this shows that for vertices v of even degree $\mathsf{R}_{\setminus v}=\prod_{w\in V\setminus v}\mathsf{S}_v=\mathsf{R}_{\text{full}}\mathsf{S}_v$ is also a tensor product of I, X, Y. This is because removing S_v from R_{full} leaves an I at v and changes by one the number of Zs on the neighbours of v. Unfortunately, removing S_v for v of odd degree leaves a Z at v. To further remove this unwanted Z, one can also remove one stabiliser S_w from a neighbouring node w of v from the product. If, in addition, w is of odd degree, then the obtained stabiliser will be a tensor product of I, X, Y only. The reason is that at w, one Z has been removed when S_v was removed from R_{full} thereby leaving an X at w, so that removing S_w leaves an I. In the general case, one can always remove from R_{full} the stabilisers S_v along a chain between u and w consisting of even degree nodes except for u and w that are odd degree. We denote by $\mathsf{R}_{\setminus(u,w)}$ such generator. Note that a given odd-degree node will always be in at least one

such stabiliser as there are always an even number of odd degree nodes in a connected component of a graph.

Now define the group R generated by R_{full}, $\mathsf{R}_{\setminus v}$ and $\mathsf{R}_{\setminus(u,w)}$ above. Notice that multiplying R_{full} with $\mathsf{R}_{\setminus v}$ gives S_v, so that S_v is in R for even $\deg(v)$. Similarly, multiplying R_{full} with $\mathsf{R}_{\setminus(u,w)}$ and S_v for v an even-degree node linking u to w shows that any $\mathsf{S}_u\mathsf{S}_w$ with u and w odd-degree nodes are also in R. Therefore, R contains all stabilisers that have an arbitrary number of even-degree node and an even number of odd-degree ones. Counting the number of such stabilisers gives $2^{|V|-1}$ while we know that the size of S is $2^{|V|}$, which concludes the proof. □

We now consider the trappified scheme $\boldsymbol{P}$ that can be obtained by sampling uniformly at random from all these traps rounds. We can characterise the errors that can be detected by $\boldsymbol{P}$ and those to which it is insensitive using properties of stabilisers. To this end, recall that if a Pauli error E is applied right before the measurement of a 2-outcome observable M, then (i) the measurement outcome probabilities are unchanged if $[\mathsf{E},\mathsf{M}] = 0$, and (ii) are swapped for $\{\mathsf{E},\mathsf{M}\} = 0$. Hence, whenever E commutes with $\bigotimes_{v\in\mathbb{1}_{\mathsf{R}}}\mathsf{X}(v)$ the trap never detects E, whereas it always detects it whenever it anticommutes. As a consequence, the set of detectable errors is the set of errors that anticommute with at least one of the $\bigotimes_{v\in\mathbb{1}_{\mathsf{R}}}\mathsf{X}(v)$ for R a dummyless trap measurement.

Hence, for an error $E = \mathsf{E}_Z\mathsf{E}_X$ we need to assess whether there exists at least one R in $\boldsymbol{P}$ such that $|\mathbb{1}_{\mathsf{R}}\cap\mathbb{1}_{\mathsf{E}_Z}| \equiv 1 \pmod 2$ – where we have implicitely defined E_Z (resp. E_X) as the operators made of Zs at location of Y or Z qubits in E (resp. X or Y qubits). To this end, consider F such that $\mathsf{U}_G\mathsf{F} = \mathsf{E}\mathsf{U}_G$ where U_G is the entangling operation for creating the graph state. Because a trap amounts to measuring the corresponding stabiliser before the entangling operation, the above question amounts to knowing whether F commutes with the stabilisers used to define the dummyless traps of $\boldsymbol{P}$. Alternatively, we can answer this question by finding out which Pauli operations commute with all stabilisers defining the dummyless traps while not being a product of them.

Using Lemma 6.2.9, there is one generator S_0 of S that is not in R and such that all errors that commute with R and are not in R are of the form $\mathsf{S}_0\mathsf{R}$. From the above description of R, S_0 can be taken as being equal to Z on all odd-degree nodes. S_0 commutes with all elements of R since they have an even number of S_v for v odd-degree, and it is not in R as R has no element with Zs only. Yet, Lemma 6.2.8 shows that while S_0 cannot be detected, it is indeed harmless for the computation.

Hence, we are led to conclude that all possibly harmful errors are detected by the

trappified scheme $\boldsymbol{P}$. Using § 6.2.1, we conclude that

Theorem 6.2.10. *Let $G = (V, E)$ be a graph, and $\boldsymbol{P}$ the trappified scheme on G defined by sampling at random from a generating set of* R *containing only stabilisers with no* Z*s. Then, $\boldsymbol{P}$ constructs the SDQC Resource 3 for* BQP *computations that can be embedded on the graph G with negligible correctness and security errors.*

This follows from the fact that Theorem 6.2.4 states that a secure verification scheme can be built from a trappified scheme that 1) detects a specific set $\mathcal{E}$ of Z-Pauli errors, and 2) correctly evaluates the target computation in the presence of any other Z-Pauli error in $\mathcal{G}_V^{\mathsf{Z}} \setminus \mathcal{E}$. Lemma 6.2.8 then shows that there is a specific error E^* which never affects the output distribution of the target computation and which therefore does not need to be detected. It hence suffices to find a dummyless trappified scheme detecting $\mathcal{E} = \mathcal{G}_V^{\mathsf{Z}} \setminus \{\mathsf{I}, \mathsf{E}^*\}$. As shown with Lemma 6.2.9, it is indeed possible to find such a trappified scheme. Therefore, this settles the question whether dummyless verification for BQP is possible by the affirmative.

6.2.4 Concrete Dummyless Tests

The previous subsection left open how to concretely construct the trappified scheme $\boldsymbol{P}$. More precisely, since the efficiency of the resulting SDQC protocol is tightly linked to the detection rate of the trappified scheme, it is important to minimise its detection, insensitivity and correctness errors. In this section, we discuss the question of optimising the detection rate. In particular, we construct concrete dummyless trappified schemes for universal BQP computations with constant detection rates, independent of the size of the computation.

[Kap+22] shows that the general optimisation problem of maximising the detection rate can be expressed in the language of linear programming. Adapted to the case of dummyless trappified schemes, we recall it in the following, as Problem 2.

For any feasible solution to Problem 2, the trappified scheme induced by the given distribution of tests gives rise to a secure dummyless SDQC protocol if and only if the detection rate satisfies $\epsilon > 0$.

Recall from Section 6.2.2 the structure of the harmless error:

$$\mathsf{E}^* = \prod_{\substack{v \in V(G), \\ \deg(v) \equiv 1 \,(\mathrm{mod}\ 2)}} \mathsf{Z}_v. \tag{6.22}$$

Problem 2 Optimisation of the Distribution of Tests

Given

- the set of errors $\mathcal{E} = \mathcal{G}_V^{\mathsf{Z}} \setminus \{\mathsf{I}, \mathsf{E}^*\}$ to be detected,
- the set of dummyless tests $\mathcal{T}_{\text{dummyless}}$,
- the relation between tests and errors describing whether a test detects an error, $R : \mathcal{T}_{\text{dummyless}} \times \mathcal{E} \to \{0, 1\}$,

find an optimal distribution $p : \mathcal{T}_{\text{dummyless}} \to [0, 1]$ **maximising** the detection rate $\epsilon \in [0, 1]$ **subject to** the following conditions:

- p describes a probability distribution, i.e. $\sum_{T \in \mathcal{T}_{\text{dummyless}}} p(T) \leq 1$,
- errors are detected at least with the target detection rate, i.e.

$$\forall E \in \mathcal{E} : \sum_{\substack{T \in \mathcal{T}_{\text{dummyless}} \\ R(T,E)=1}} p(T) \geq \epsilon. \tag{6.21}$$

Further, as described in Section 6.2.3, the set of dummyless tests can be expressed as:

$$\mathcal{T}_{\text{dummyless}} = \left\{ \prod_{v \in V_{\text{trap}}} \mathsf{X}_v \prod_{w \in N_G(v)} \mathsf{Z}_w \;\middle|\; V_{\text{trap}} \subseteq V, \forall v \notin V_{\text{trap}} : |N_G(v) \cap V_{\text{trap}}| \equiv 0 \pmod 2 \right\}. \tag{6.23}$$

The last condition ensures that there are no vertices with a single Z in the respective stabiliser. In this way, every test can be identified with the subset of vertices which act as traps, or equivalently with the complement, the subset of vertices which act as *holes*, i.e. vertices on which the respective stabiliser equals the identity and which can therefore be ignored by the decision function of the trappified scheme. In the following we will also write $V_{\text{trap}}(T)$ and $V_{\text{hole}}(T)$ as shorthands for these two sets of vertices.

Analogously, we write $V_{\text{error}}(\mathsf{E})$ for the set of vertices on which the error E is not equal to the identity (and therefore equals the Pauli Z). This makes it easy to give a short description of the relation R:

$$R : (T, \mathsf{E}) \mapsto |V_{\text{error}}(\mathsf{E}) \cap V_{\text{trap}}(T)| \pmod 2. \tag{6.24}$$

Handling Errors on Even-degree Vertices. As described in Section 6.2.3, for all even-degree vertices $v \in V$, the test T with $V_{\text{hole}}(T) = \{v\}$ is indeed dummyless.

Generalising this concept, for any independent set V^* of even-degree vertices, we can define a dummyless test T with $V_{\text{hole}}(T) = V^*$. Similarly to the construction of tests in [Kap+22], any (fractional) colouring of the vertices of a graph G gives rise to a distribution of independent sets of G, and therefore also a distribution of independent sets of even-degree vertices and tests. To this end, let $\mathcal{D}$ be a distribution of independent sets of G such that

$$\forall v \in V : \Pr_{I \leftarrow \mathcal{D}}[v \in I] \geq \frac{1}{\chi_f(G)}, \tag{6.25}$$

where $\chi_f(G)$ is the fractional chromatic number of G. This distribution exists by definition of the fractional chromatic number. Consider the test strategy given by the distribution $\mathcal{D}_{\text{even}}$ of tests in $\mathcal{T}_{\text{dummyless}}$ described as follows:

1. Sample an independent set: $V_1 \leftarrow \mathcal{D}$.
2. Restrict the set to even-degree vertices: $V_2 = V_1 \cap V_{\text{even}}(G)$, where $V_{\text{even}}(G) = \{v \in V \mid \deg(v) \equiv 0 \pmod 2\}$.
3. Choose a random subset to determine the location of holes: $V_3 \leftarrow \mathcal{U}(\wp(V_2))$.
4. Perform the dummyless test T determined by $V_{\text{hole}}(T) = V_3$.

As the following Lemma shows, this strategy allows for a detection rate of errors that affect even-degree vertices that scales inversely with the fractional chromatic number of the graph.

Lemma 6.2.11 (Even-degree Error Detection)**.** *The above-mentioned test strategy* $\left(\frac{1}{2\chi_f(G)}\right)$*-detects the error set* $\mathcal{E}_{even} = \{E \in \mathcal{G}_V^{\mathsf{Z}} \mid V_{error}(E) \cap V_{even} \neq \emptyset\}$*, i.e.*

$$\forall E \in \mathcal{E}_{even} : \quad \mathbb{E}_{T \leftarrow \mathcal{D}_{even}}[|V_{error}(E) \cap V_{trap}(T)| \equiv 1 \pmod 2] \geq \frac{1}{2\chi_f(G)}. \tag{6.26}$$

Proof. Let $E \in \mathcal{E}_{\text{even}}$. Then, by definition of the test distribution, it holds that

$$\begin{aligned}
&\mathbb{E}_{T \leftarrow \mathcal{D}_{\text{even}}}[|V_{\text{error}}(E) \cap V_{\text{trap}}(T)| \equiv 1 \pmod 2] \\
&\geq \mathbb{E}_{V_3 \leftarrow \mathcal{U}(\wp(V_2))}[|V_{\text{error}}(E) \cap V_3| \equiv 1 \pmod 2 \mid V_{\text{error}}(E) \cap V_2 \neq \emptyset] \\
&\qquad \cdot \Pr_{V_1 \leftarrow \$ \mathcal{D}}[V_{\text{error}}(E) \cap V_1 \cap V_{\text{even}}(G) \neq \emptyset] \\
&\geq \frac{1}{2} \cdot \frac{1}{\chi_f(G)},
\end{aligned} \tag{6.27}$$

which concludes the proof. □

Handling Errors on Odd-degree Vertices. Since all errors acting non-trivially on even-degree vertices are already handled in the previous case, it remains to detect errors that affect only odd-degree vertices and act as the identity on even-degree vertices.

To this end, we construct a specific type of test. For $k \geq 2$, let $(v_1, \ldots, v_k) \in V^k$ be a chain of vertices in G satisfying the following conditions:

1. The end vertices are of odd degree: $\deg(v_1) \equiv \deg(v_k) \equiv 1 \pmod 2$.
2. All intermediate vertices are of even degree: $\deg(v_2) \equiv \cdots \equiv \deg(v_{k-1}) \equiv 0 \pmod 2$.
3. Only subsequent vertices are neighbours in G:

$$\forall i, j \in \{1, \ldots, k\} : \quad \{v_i, v_j\} \in E(G) \Leftrightarrow |i - j| = 1. \tag{6.28}$$

It is easy to verify that under these conditions there exists a valid dummyless test T with $V_{\text{hole}}(T) = \{v_1, \ldots, v_k\}$. Note, that there might not be a chain of this type in G for any pair of odd-degree vertices as end points. However, it is possible to connect any two odd-degree vertices through a chain of chains that might traverse other odd-degree vertices at the end and starting points of chains. In this way, it is possible to choose a "spanning tree" of $(|V_{\text{odd}}(G)| - 1)$ chains that connects all odd-degree vertices in the graph G.

Define the set of errors on odd-degree nodes only as $\mathcal{E}_{\text{odd}} = \{E \in \mathcal{G}_V^{\mathsf{Z}} \mid V_{\text{error}}(\mathsf{E}) \cap V_{\text{even}}(G) = \emptyset \ \wedge \ V_{\text{error}}(\mathsf{E}) \cap V_{\text{odd}}(G) \neq \emptyset\}$ and let $\mathsf{E} \in \mathcal{E}_{\text{odd}} \setminus \{\mathsf{E}^*\}$. Then, there must exist two odd-degree vertices $v_1 \in V_{\text{error}}(\mathsf{E})$ and $v_2 \notin V_{\text{error}}(\mathsf{E})$. But then at least one of the chains connecting v_1 and v_2 with start in a vertex affected by the error E and end in a vertex unaffected by E. Since all intermediate vertices are of even degree and therefore unaffected by E, the test given by this chain detects E. This essentially shows the following statement.

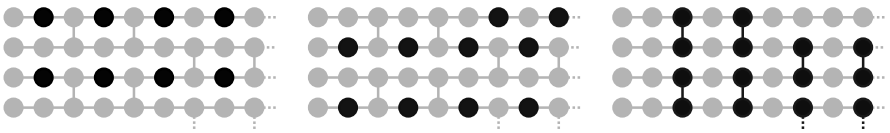
Lemma 6.2.12 (Odd-degree Error Detection)**.** *There exists an efficient testing strategy that* $\left(\frac{1}{|V_{odd}(G)|-1}\right)$*-detects errors in* $\mathcal{E}_{odd} \setminus \{\mathsf{E}^*\}$.

Combining the testing strategies from Lemma 6.2.11 and Lemma 6.2.12 immediately yields the following result for testing strategies on general graphs.

Lemma 6.2.13 (Error Detection on General Graphs)**.** *For any graph G, there exists an efficient testing strategy that ε-detects $\mathcal{E} = \mathcal{G}_V^{\mathsf{Z}} \setminus \{\mathsf{I}, \mathsf{E}^*\}$, where*

$$\begin{aligned}\varepsilon &= \frac{1}{2\chi_f(G)(|V_{odd}(G)|-1)}\left(\frac{1}{2\chi_f(G)}+\frac{1}{|V_{odd}(G)|-1}\right)^{-1} \\ &\geq \frac{1}{2}\min\left\{\frac{1}{2\chi_f(G)}, \frac{1}{|V_{odd}(G)|-1}\right\}. \end{aligned} \tag{6.29}$$

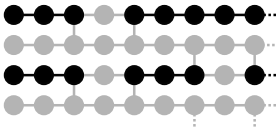
This already shows that the detection rate that is achievable on general graphs decreases at most linearly in the number of vertices of the graph. This lower bound is however far from tight in many cases. In fact, even for universal graph states a constant lower bound is possible as the following result shows.



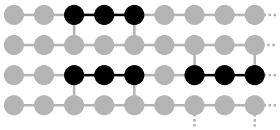
(a) Holes in even-degree vertices, type 1.

(b) Holes in even-degree vertices, type 2.

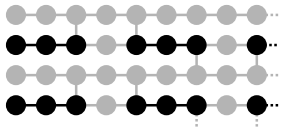
(c) Holes in odd-degree vertices, type 1.



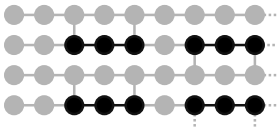
(d) Holes in odd-degree vertices, type 2.



(e) Holes in odd-degree vertices, type 3.



(f) Holes in odd-degree vertices, type 4.



(g) Holes in odd-degree vertices, type 5.

Figure 6.1: The seven types of dummyless tests for the brickwork graph. A trap configuration is sampled by randomly choosing one of the seven types, and then in cases 6.1a-6.1b sampling uniformly at random a subset of marked vertices as holes, and in cases 6.1c-6.1g sampling uniformly at random a subset of marked chains as holes.

Lemma 6.2.14 (Error Detection on the Brickwork State)**.** *Let G be a brickwork graph. Then, there exists an efficient testing strategy that $(1/14)$-detects $\mathcal{E} = \mathcal{G}_V^{\mathsf{Z}} \setminus \{\mathsf{I}, \mathsf{E}^*\}$.*

Proof Sketch. To detect errors affecting even-degree vertices, use the strategy from Lemma 6.2.11. As the brickwork graph is bipartite, this will yield a detection rate of $1/4$.

To detect errors on odd-degree vertices, follow the strategy from Lemma 6.2.12, but

use chains that can be tested in parallel to boost the detection rate. There are five classes of chains between odd-degree vertices that can each be run at the same time. One class consists of all vertical chains, and the other four of horizontal chains where every class contains chains only in every second row and only every second horizontal chain on these rows. By testing random subsets of these classes of chains, the detection rate in this case is lower bounded by 1/10.

Optimal switching between these two strategies (with probabilities 2/7 and 5/7) yields an overall detection rate of 1/14. The different types of tests on the brickwork graph are depicted in Figure 6.1. □

6.3 Collaborative State Preparation

Following the approach outlined in § 6.1.3, we now turn to the design of a composably secure protocol for implementing the preparation of the input states required by the dummyless protocols introduced in § 6.2.3. The Collaborative Remote State Preparation Protocol 6 presented here will allow n Clients to collaboratively construct an encrypted state on the Server whose encryption key is held by a purely classical party called the Orchestrator. It guarantees that no malicious coalition including up to $n-1$ Clients and the Server (but not the Orchestrator) has any knowledge about the final state.

This security property is captured formally as follows. The Remote State Preparation Resource 4 (or RSP) allows one party called the Sender to prepare a quantum state on a device held by another party called the Receiver. Its simplest instantiation requires only a direct quantum channel between the two participants but more interesting scenarios can be considered, for example using untrusted relays or additional participants. We specify this resource for our specific case, i.e. sending states in the $X-Y$ plane.

Resource 4 Remote State Preparation

Inputs: The Sender has as input an angle $\theta \in \Theta = \left\{\frac{k\pi}{4}\right\}_{k\in\{0,\ldots,7\}}$.

Computation by the Resource: The Resource prepares and sends the state $|+_\theta\rangle$ to the Receiver.

The goal of the Collaborative Remote State Preparation Protocol is then to construct this Remote State Preparation Resource 4 between the Orchestrator and the Server using one Quantum Channel Resource between each Client and the Server and one Secure Classical Channel Resource between each Client and the Orchestrator. This

latter Resource transmits faithfully and privately any classical message from the sender to the receiver, while only leaking the size of the message to an eavesdropper.

Protocol 6 Collaborative Remote State Preparation

Input: The Orchestrator has as input an angle $\theta \in \Theta$. The Server and Clients have no input.
Protocol:

- Client j samples uniformly at random $\theta_j \in_R \Theta$ and sends $\left|+_{\theta_j}\right\rangle$ to the Server.
- Client j sends θ_j to the Orchestrator using a Secure Classical Channel.
- For each $j \neq n$, the Server applies $\mathsf{CNOT}_{n,j}$ between the qubits n and j, with the first being the control and the second the target. It measures the target qubit j in the computational basis with measurement outcome t_j. It sends the vector $\boldsymbol{t}$ containing all the measurement outcomes to the Orchestrator.
- The Orchestrator computes $\theta' = \theta_n + \sum_{j\in[n-1]}(-1)^{t_j}\theta_j$ and sends a correction $(b, (-1)^b\theta - \theta')$ to the Server, who applies $\mathsf{X}^b\mathsf{Z}((-1)^b\theta - \theta')$ to the unmeasured qubit, keeping it as output.

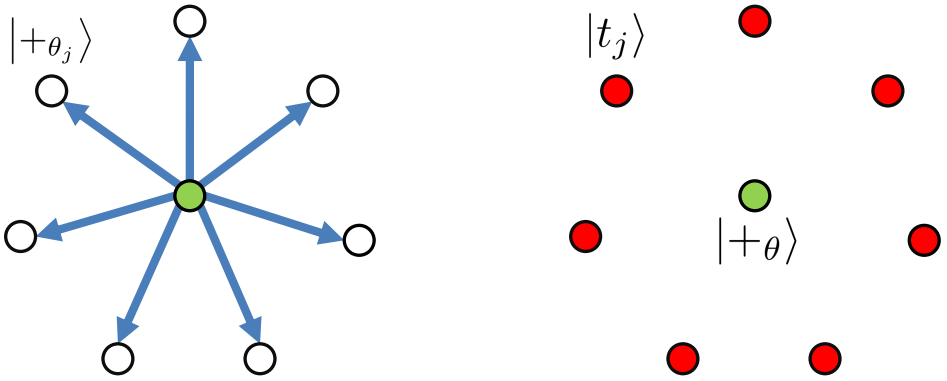
We can now state the main result of this section, namely the correctness and security of Protocol 6 in the AC framework. Both properties are proven independently below.

Theorem 6.3.1 (Security of Collaborative Remote State Preparation)**.** *Protocol 6 perfectly constructs the Remote State Preparation Resource 4 from Secure Classical Channel Resources between each Client and the Orchestrator, for malicious coalitions that include the Server and at most $n-1$ Clients.*

Proof of Correctness. The state of the central qubit after an honest execution of Protocol 6 before the correction sent by the Client is $|+_{\theta'}\rangle$ with:

$$\theta' = \theta_n + \sum_{j\in[n-1]} (-1)^{t_j}\theta_j. \tag{6.30}$$

It is sufficient to prove this for a pure state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ as control. We apply a CNOT gate with $|\phi\rangle$ as control and $|+_{\hat{\theta}}\rangle$ with $\hat{\theta} \in \Theta$ as target, followed by a measurement of this second qubit in the computational basis. Let $t \in \{0,1\}$ be the measurement result. After tracing out the second qubit post-measurement, the system is in the following



(a) The Server receives the qubits and applies CNOT gates. The central qubit n is the control, the rest are targets.

(b) The Server measures all qubits but the central one in the computational basis and gets outcomes $t_j \in \{0, 1\}$.

Figure 6.2: Collaborative Remote State Preparation for eight qubits. All qubits start in the state $|+_{\theta_j}\rangle$.

state:

$$\begin{aligned}\sqrt{2}\,\langle 0|_2\,\mathsf{X}_2^t\mathsf{CNOT}_{1,2}\,|\phi\rangle\,|+_{\hat{\theta}}\rangle &= \langle 0|_2\,\mathsf{X}_2^t\Big(\alpha\,|00\rangle + \alpha e^{i\hat{\theta}}\,|01\rangle + \beta\,|11\rangle + \beta e^{i\hat{\theta}}\,|10\rangle\Big)\\ &= \langle 0|_2\,(\alpha\,|0\rangle + \beta e^{i\hat{\theta}}\,|1\rangle)\,|t\rangle + e^{i\hat{\theta}}\,\langle 0|_2\,(\alpha\,|0\rangle + \beta e^{-i\hat{\theta}}\,|1\rangle)\,|t\oplus 1\rangle\\ &= \mathsf{Z}(\hat{\theta})\,|\phi\rangle\,\langle 0|t\rangle + e^{i\hat{\theta}}\mathsf{Z}(-\hat{\theta})\,|\phi\rangle\,\langle 0|t\oplus 1\rangle \qquad (6.31)\end{aligned}$$

Therefore, the result of this single step is $\mathsf{Z}((-1)^t\hat{\theta})\,|\phi\rangle$ up to a global phase. Replacing the result above in the sequence of CNOT's and measurements performed by the Server where the control is qubit n and the targets are qubits $j \neq n$ yields the desired value for θ'. Finally, the rotation correction $(-1)^b\theta - \theta'$ sent by the Orchestrator, along with X^b, transform the value of the final state into $|+_\theta\rangle$. □

Proof of Security. We first construct a Simulator against an adversarial Server and a coalition of $n-1$ Clients, which represents the worst case. The Server expects to receive n qubits and a final correction after transmitting the measurement results. The Simulator has single-query oracle access to the Remote State Preparation Resource 4 for state set $\{|+_\theta\rangle\}_{\theta\in\Theta}$. It receives a state from this resource, without the corresponding classical description, and must make the Server accept this state as its output at the end of the interaction. The actions of this Simulator are described in Simulator 2. Let

h be the index associated to the honest Client.

Simulator 2 Malicious Server and $n-1$ Clients

1. The Simulator calls the Remote State Preparation Resource 4 and receives a state $|+_\theta\rangle$.
2. It then emulates the behaviour of the n Quantum Channel Resources:
   - For indices $j \neq h$, it simply forwards the state from corrupted Client j to the Server;
   - For index h, it samples uniformly at random $\theta_h \in_R \Theta$ and $b_h \in_R \{0,1\}$, and sends an encrypted version $\mathsf{Z}(\theta_h)\mathsf{X}^{b_h}(|+_\theta\rangle)$ of the state received from the RSP Resource.
3. It then emulates the Secure Classical Channel Resources and receives from each corrupted Client $j \neq h$ a value θ_j
4. It receives from the Server a bit-string of measurement results $\mathbf{t} \in [n-1]$.
5. After extending the bit-string $\mathbf{t}$ with $t_n = 0$, it computes θ' using Equation 6.30 and sends the correction $(t_h \oplus b_h, -\theta')$ to the Server (by impersonating the Orchestrator) and halts.

We can now prove that no Distinguisher can tell apart the following two situations with one honest client: (i) the ideal resource interacting with the Simulator, and (ii) the real scenario.

Data and transcripts available to the Distinguisher. By construction, the Distinguisher fixes θ the angle of the desired state to be prepared at the Server output-interface. It also fixes the value of all θ_j for $j \neq h$ both in the real and ideal scenarios and has perfect knowledge of the states sent by malicious parties. It does not have access to θ_h as this is fixed by the honest client protocol.

Before sending the values for the measurement outcomes, the Distinguisher receives from the non-corrupted party the state $|+_{\theta_h}\rangle$ in the real case and the state $\left|+_{(-1)^{b_h}\theta+\theta_h}\right\rangle$ in the ideal case. After sending the bit-string $\mathbf{t}$, regardless of how it was chosen, the Distinguisher receives a bit and an angle corresponding to the corrections chosen by either the Orchestrator or the Simulator. In the first case this is equal to $(b, (-1)^b\theta - \theta')$ and in the second case $(t_h \oplus b_h, -\theta')$ with b being chosen uniformly at random and θ' being computed in the exact same way in both settings (see Equation 6.30).

The remaining parameters in the real case and ideal cases are the received honest state, the associated measurement outcome, the X-correction bit and the Z-correction angle. This gives us the following variables that are in the hands of the Distinguisher (rows are labeled by the meaning of the corresponding data in the real setting):

	Real world	**Ideal world**
Orchestrator-chosen output angle	θ	θ
Server's received quantum state	$\lvert+_{\theta_h}\rangle$	$\left\lvert+_{(-1)^{b_h}\theta+\theta_h}\right\rangle$
Measurement result bit	t_h	t_h
Orchestrator correction bit	b	$b_h \oplus t_h$
Orchestrator correction angle	$(-1)^b\theta - (-1)^{t_h}\theta_h$	$-(-1)^{t_h}\theta_h$

Indistinguishability of data and transcripts for the Distinguisher. To finish the security proof, we need to show that the distributions of the above data and transcripts are statistically indistinguishable in both scenarios. To do this, we will perform a series of row-wide operations and eliminate the parameters of the corrupted parties so that we are left with a new set of variables that will be trivially indistinguishable. The reversibility of each operation and its dependency on values that are known to the Distinguisher guarantees that it can always undo them.

First, multiply the final angle by $(-1)^{t_h}$ and use this angle to apply a rotation to the state. This transforms the above values into:

Real world	**Ideal world**
θ	θ
$\left\lvert+_{(-1)^{b\oplus t_h}\theta}\right\rangle$	$\left\lvert+_{(-1)^{b_h}\theta}\right\rangle$
t_h	t_h
b	$b_h \oplus t_h$
$(-1)^{b\oplus t_h}\theta - \theta_h$	$-\theta_h$

Note that in both cases, the value for θ_h only appears in the last row term. Since it is chosen uniformly at random both final terms follow the same distribution, meaning that they give no distinguishing advantage. We can therefore safely omit them in the rest of the process:

Real world	**Ideal world**
θ	θ
$\left\|+_{(-1)^{b\oplus t_h}\theta}\right\rangle$	$\left\|+_{(-1)^{b_h}\theta}\right\rangle$
t_h	t_h
b	$b_h \oplus t_h$

Since b in the first row is a bit sampled uniformly at random, we can substitute it with $b \oplus t_h$ without changing the distribution.[4] We arrive at

Real world	**Ideal world**
θ	θ
$\left\|+_{(-1)^{b}\theta}\right\rangle$	$\left\|+_{(-1)^{b_h}\theta}\right\rangle$
t_h	t_h
$b \oplus t_h$	$b_h \oplus t_h$

Because the b and b_h are uniformly random bits, the above two distributions are identical, which concludes the proof. □

6.4 Quantum Secure Multi-Party Computation

We present in this section an extension of the SDQC Protocol 5 from Section 6.2.3 based on the trappified schemes in the $X-Y$ plane. We consider here that n Clients want to perform a joint MBQC computation on private classical inputs, receiving at the end either the same classical output or an abort message. There are two steps in the SDQC protocol which must be modified: the preparation of a state which is compatible with the SDQC protocol and does not leak any information to coalitions of malicious parties, and the classical interaction between with the server to drive the computation and tests. If these components are available, the composable security of the SDQC protocol ensures that the multi-party version is also secure.

The second step is purely classical once the state and computation have been fixed and we will use a Classical SMPC Resource to handle it. This Resource will also sample the trappified canvas and embed the Client's desired computation into it. Hence, no malicious coalition will be able to learn where the tests are located among the blind computations. The first step will make use of the Collaborative RSP Protocol 6 from the

[4] This is the hidden reason for the additional encryption via X^b in the protocol.

previous section, replacing the Orchestrator by calls to the Classical SMPC Resource. The n Clients will use it to prepare rotated $|+\rangle$ states on the Server such that the encryption angle θ is unknown to any malicious coalition, which protects the blindness of each computation.

Our resulting Secure Delegated Quantum Secure Multi-Party Computation Protocol with Classical IO (Protocol 7) is therefore an information-theoretic upgrade of the Classical SMPC functionality. This is the best one can hope for without an honest majority since it is impossible in that case to construct an information-theoretically secure Quantum SMPC protocol. Crucially, no additional computational assumptions are used beyond what is required to construct the Classical SMPC Resource. This modularity means that we can instantiate our protocol using any post-quantum secure assumption which is capable of constructing a Classical SMPC.

Quantum Secure Multi-Party Computation Resource. Our protocol will construct the following Quantum Secure Multi-Party Computation Resource 5. It has $n+1$ interfaces, one for each Player and the last one for an Eavesdropper. It allows n Players to perform a collectively defined quantum computation C over their private classical inputs with the guarantee that their computation is either executed properly, in which case Player j receives the correct classical output, or it is aborted altogether. It is allowed to leak a known value l_ρ about the Players' computation and input on the Eavesdropper's filtered interface.

In order to construct this resource, we will make use of its classical equivalent. Our protocol will in the end be an information theoretical upgrade of the following Resource.

Classical Secure Multi-Party Computation Resource. Resource 6 allows n Players to provide their private inputs and perform a collectively defined computation C on them with the guarantee that the computation is performed properly. We assume that it keeps an internal state between calls.

Delegated QSMPC Protocol. Our final protocol will be built upon the two presented earlier. In an execution the Trappified Delegated Blind Computation Protocol 5, the Client can perform all of its classical interactions with the Server via a Classical SMPC Resource 6 if it provides this resource with its input and computation (angles and flow). This resource is then responsible for sampling all the secret parameters – angles, bits, order of test and computation runs, which tests to perform – and simply

Resource 5 Quantum Secure Multi-Party Computation with Classical IO

Inputs:

- Player j sends a classical bit-string x_j. It can also input two bits f_j and c_j as a filtered interface.
- The n Players send the classical description of a quantum polynomial-time computation C with classical inputs and outputs.
- The Eavesdropper can input two bits e and c as a filtered interface.

Computation by the Resource:

- If $e = 1$, the Resource sends the leakage l_ρ to the Eavesdropper's interface.
- If $c = 1$ or there exists j such that $c_j = 1$, the Resource sends Abort to all Players j such that $c_j = 0$.
- It computes $O = \mathsf{C}(x)$, where x is the concatenation of strings x_j.
- If there exists $j \in [n]$ such that $f_j = 1$, it sends O to Player j.
- If there has been no abort at this stage, it sends the outputs O to all other Players j in a similar fashion.

instructs the Client to prepare specific states to send to the Server. Since only rotated $|+\rangle$ states are required for this verification protocol, this step can further be replaced by an instance of the Remote State Preparation Resource 4 for states $\{|+_\theta\rangle\}_{\theta\in\Theta}$, as sending a state from this set is a perfect protocol constructing the RSP Resource. We can then finally replace this resource by the Collaborative Remote State Preparation Protocol 6, in which the Orchestrator is played by the Classical SMPC Resource.

In essence, the Classical SMPC together with the Collaborative RSP emulate the behaviour of the honest Client in an execution of the Trappified Delegated Blind Computation Protocol, whose tests – described in Section 6.2.4 – needed to be tailored specifically to require only the preparation of rotated $|+\rangle$ states. The full description is given below in Protocol 7. We continue to refer to the Classical SMPC Resource as the Orchestrator for simplicity, since in the Abstract Cryptography framework there is no formal difference between an honest party and an interactive Resource.

Extending the Functionality. The presentation above restricts how the input and output are treated for simplicity's sake and any additional efficient classical pre- and post-processing steps can be performed by the Orchestrator with no impact on the security of the protocol.

Resource 6 Classical Secure Multi-Party Computation

Inputs:

- Player j sends a classical bit-string x_j. It can also input two bits f_j and c_j as a filtered interface.
- The n Players send the description of a classical polynomial-time computation C.

Computation by the Resource:

- If there exists j such that $c_j = 1$, the Resource sends Abort to all Players j such that $c_j = 0$.
- It computes $O = C(x)$, where x is the concatenation of strings x_j.
- If there exists $j \in [n]$ such that $f_j = 1$, it sends O to Player j.
- If there has been no abort at this stage, it sends the outputs O to all other Players j in a similar fashion.

Removing the Correction in the Collaborative RSP Protocol used with UBQC. The final step of the Collaborative RSP Protocol calls for the Orchestrator to instruct the Server to apply a correction $\mathsf{X}^b\mathsf{Z}((-1)^b\theta - \theta')$ to a state which in the honest case is equal to $|+_{\theta'}\rangle$, for a random value of $b \in_R \{0,1\}$ and the Orchestrator's desired angle θ. This is required to make the protocol simulatable against a malicious coalition – otherwise, the Simulator has no way of transmitting the correct state to the Server. However, in Protocol 7 these qubits are used in an execution of the UBQC Protocol, in which the Orchestrator requests that the Server measures the qubit in the basis $\{|\pm_\delta\rangle\}$ for $\delta = \phi' + \theta + r\pi$. Together, the unitary operations on this qubit in the honest case can be written as

$$\mathsf{Z}(-\delta)\mathsf{E}\mathsf{X}^b\mathsf{Z}((-1)^b\theta - \theta')\mathsf{Z}(\theta')\,|+\rangle \otimes |\psi\rangle \tag{6.32}$$

for a state $|\psi\rangle$ representing the rest of the state and the graph entangling operation E. Then, this is equal to

$$\begin{aligned}&\mathsf{Z}(-\phi' - \theta - r\pi)\mathsf{E}\mathsf{Z}(\theta - (-1)^b\theta')\mathsf{Z}((-1)^b\theta')\,|+\rangle \otimes |\psi\rangle \\ &= \mathsf{Z}(-\phi' - (-1)^b\theta' - r\pi)\mathsf{E}\mathsf{Z}((-1)^b\theta')\,|+\rangle \otimes |\psi\rangle\,.\end{aligned} \tag{6.33}$$

By performing the change of variables $\hat{\theta} = (-1)^b\theta'$, which is drawn from the same distribution, we recover the state in the original UBQC Protocol, with no correction

Protocol 7 Secure Delegated Quantum Secure Multi-Party Computation with Classical IO

Public Information:

- $G = (V, E, I, O)$, a graph with input and output vertices I and O respectively;
- $\{I_j\}_{j\in[n]}$, a partition of the input vertices, with each I_j being associated to Client j.
- $\boldsymbol{P}$, a trappified scheme on graph G;
- $\preceq_G$, a partial order on the set V of vertices;
- N, d, w, parameters representing the number of runs, the number of computation runs, and the number of tolerated failed tests.

Clients' Inputs:

- Each Client j has as input a classical bit-string $x_j \in \{0,1\}^{|I_j|}$.
- The n Clients collaboratively have as input a set of angles $\{\phi_i\}_{i\in V}$ and a flow f which induces an ordering compatible with $\preceq_G$.

Protocol:

1. The Clients send their input x_j to the Orchestrator, together with the computation angles $\{\phi_i\}_{i\in V}$ and flow f. Let x be the concatenation of all x_j.
2. The Orchestrator and the Server perform an execution of the Trappified Delegated Blind Computation Protocol 5. Instead of having the Orchestrator send rotated states during the UBQC execution, they perform for each state an instance of the Collaborative State Preparation Protocol 6 together with the n Clients.
   (a) The Orchestrator samples uniformly at random a subset $C \subset [N]$ of size d representing the computation runs.
   (b) For $k \in [N]$:
      i. If $k \in C$, the Orchestrator sets the computation for the run to $(\{\phi_i\}_{i\in V}, f)$ with input x. Otherwise, the Orchestrator samples a test (T, σ, τ) from the trappified scheme $\boldsymbol{P}$.
      ii. The Orchestrator and Server execute the chosen run with the UBQC Protocol 2. For each qubit sent during the execution of the protocol, they instead execute the Collaborative RSP Protocol 6 together with the n Clients.
      iii. If the run is a test, the Orchestrator checks whether it passed.
   (c) If the number of failed tests is greater than w, the Orchestrator sets the output to $(\perp, \mathsf{Rej})$.
   (d) Otherwise, let O be the majority vote on the output results of the computation runs. The Orchestrator sets the output to (O, Acc).
3. The Orchestrator sends its set output to all Clients.

from the Orchestrator:

$$\mathsf{Z}(-\phi' - \hat{\theta} - r\pi)\mathsf{EZ}(\hat{\theta}) \left|+\right\rangle \otimes \left|\psi\right\rangle . \tag{6.34}$$

Therefore in the full protocol, requesting and applying the correction are unnecessary steps, either for correctness or security, since the states with or without these corrections are equal.

Security of QSMPC. We now prove the correctness and security of our QSMPC protocol using the composition of AC resources and protocols.

Theorem 6.4.1 (Security of Delegated Quantum SMPC)**.** *Suppose that the Trappified Delegated Blind Computation Protocol 5 ϵ_V-constructs the Secure Delegated Quantum Computation with Classical IO Resource 3 for leak l_ρ. Then Protocol 7 ϵ_V-constructs the Quantum Secure Multi-Party Computation with Classical IO Resource 5 from an interactive Classical Secure Multi-Party Computation Resource 6 for the same leak l_ρ, against malicious coalitions that include at most the Server and $n-1$ Clients.*

Proof. This proof is very simple and works by retracing in reverse order the high-level description of the protocol in the worst case with $n-1$ malicious Clients in collusion with a malicious Server

We first use the security of the Collaborative RSP Protocol as expressed in Theorem 6.3.1 to replace each instance of this protocol with a call to the RSP Resource 4, at no security cost. The Secure Classical Channel Resources from the Clients to the Orchestrator come for free since this party is now replaced by the Classical SMPC Resource in our protocol.

We can then replace these Resources with a direct quantum communication channel between the Orchestrator and the Server, since this protocol perfectly implements the RSP Resource. We obtain as a result exactly an execution of the UBQC Protocol 2 between the Orchestrator and the Server in step 2.b.ii of Protocol 7. The whole step 2 of Protocol 7 is then exactly an execution of Protocol 5 between the Orchestrator and the Server.

We then use the fact that this protocol ϵ_V-constructs the Secure Delegated Quantum Computation with Classical IO Resource and replace it by a call to that resource with a cost of ϵ_V.

In this final stage, the Clients send their desired computation and inputs to the Orchestrator, which only forwards the concatenated input to the SDQC Resource. This

Resource leaks the value l_ρ to the Server and returns the correct value to the Orchestrator if there has been no abort from the Server. The Orchestrator then sends back this output to the malicious Clients if they desire to receive it first. If there has been no abort at this stage, the Orchestrator finally transmits the output to the honest Clients as well. Therefore merging the Orchestrator – a Classical SMPC Resource – and the SDQC Resource yields exactly the behaviour of the desired QSMPC Resource between the n Clients and the Server. □

6.5 Discussion

6.5.1 Comparison with Other QSMPC Protocols

Table 6.1 below gives a comparison of our protocol with the peer-to-peer protocols of [Dul+20] and [LRW20], and with the more recent semi-delegated protocol of [Alo+20]. We note n is the number of parties, d the depth of the computation (MBQC in our case, circuit for [LRW20] and $\{\mathsf{T}, \mathsf{CNOT}\}$-depth for [Dul+20]), t the number of T gates, c the number of CNOT gates, C_{dist} the code distance used in [LRW20] and η a statistical security parameter. The values below correspond to the simple case where each player has a single qubit of input.

Security guarantees. Reference [Dul+20] achieve an information-theoretic upgrade of a Classical SMPC to the quantum domain, secure against an arbitrary number of corrupted parties. On the other hand, the protocol from [Alo+20] is only computationally-secure since it relies on a Fully-Homomorphic Encryption Scheme on top of the Classical SMPC, but it is also secure against arbitrary corruptions. The protocol of [LRW20] constructs an information-theoretically secure Quantum SMPC but suffers from an artificial blow-up in the number of participants and exchanged qubits.[5] The protocols of [LRW20; Alo+20] are proven secure in the Stand-Alone Model, whereas ours and that of [Dul+20] are fully composable. On top of blindness, all protocols provide verifiability with unanimous abort apart from that of [Alo+20] which achieves the stronger notion

[5] It is based on error-correcting codes and the size of the code must correspond to the number of players n. The maximum number of cheaters tolerated by the protocol is the number of correctable errors $\lfloor \frac{C_{dist}-1}{2} \rfloor$, which by the quantum Singleton bound [Rai99] is at most $\lfloor \frac{n-1}{4} \rfloor$. In their example, 7 players are required for implementing a two-party computation since the code that is used is of size 7 and corrects 1 error. This leads to a situation where 5 participants that don't have inputs nor outputs must still exchange messages and none can be malicious if one of the players with inputs is.

	Dulek et al. [Dul+20]	**Lipinska et al. [LRW20]**	**Alon et al. [Alo+20]**	**This work**
Security	Stat. upgrade of CSMPC	Stat.	Comp. (FHE + CSMPC)	Stat. upgrade of CSMPC
Abort	Unanimous	Unanimous	Identifiable	Unanimous
Composability	Composable	Stand-Alone	Stand-Alone	Composable
Max adversaries	$n-1$	$\left\lfloor \frac{C_{dist}-1}{2} \right\rfloor$	$n-1$	$n-1$
Protocol nature	Symmetric	Symmetric	Semi-Delegated	Delegated
Network topology	Q and C: Complete	Q and C: Complete	Q and C: Complete	Q: Star / C: Complete
Q operations	FTQC	FTQC	FTQC	Cl: Single Qubit S: FTQC
Classical SMPC	Clifford Computation, Operations in $\mathbb{Z}_2$, CT	CT	Clifford Computation, FHE verification	Operations in $\mathbb{Z}_8$, $\mathbb{Z}_2$, CT
Rounds (C)	$\mathcal{O}(d+\eta(N+t))$	$d+2$	$\mathcal{O}(1)$	$d+3$
Rounds (Q)	Par: $\mathcal{O}(nd)$ Seq: $\mathcal{O}(n(n+t+c))$	Par: 3 (2 if C output) Seq: $\mathcal{O}(\eta^2(n+t))$	Par: $\mathcal{O}(n^4)$	Par: 1 Seq: $\mathcal{O}(\eta nd)$
Size of Q memory	Par: $\mathcal{O}(\eta^2(n+t)))$ Seq: $\mathcal{O}(\eta^2 n)$	Par: $\mathcal{O}(\eta^2 n(n+t))$ Seq: $\mathcal{O}(n^2)$	Par: $\mathcal{O}(tn^9\eta^2)$	Cl: 0 S (par): $\mathcal{O}(\eta n^2 d)$ S (seq): $\mathcal{O}(nd)$

Table 6.1: Comparison with [Dul+20; LRW20; Alo+20]. Q stands for quantum and C for classical. The abbreviations Cl and S stand for Client and Server respectively. Stat. means statistical, FTQC stands for Fault-Tolerant Quantum Computer and CT for Coin-Toss.

of identifiable abort.[6]

Communication requirements. One key advantage of our protocol over the others lies in its delegated nature, where only one participant needs a full fault-tolerant quantum computer while the rest only perform very limited quantum operations, compared with the symmetric setup in [Dul+20; LRW20] where all participant has requires fault-tolerance. The protocol of [Alo+20] can be considered semi-delegated in the sense that the brunt of the quantum computation is performed by a single player. However, all players must have the ability to perform arbitrary Cliffords on large states and cannot do so without having at their disposal a full fault-tolerant quantum computer. This is also reflected in the network topology: whereas the best performance in [Dul+20; LRW20; Alo+20] can only be reached by using a complete quantum and classical communication graph, we only need a star graph for quantum communications. While the network topology of [Dul+20] and [LRW20] can also be star-shaped – with one player acting as a router – this would degrade their performance in terms of quantum communication

[6] A protocol satisfies the unanimous abort property if all honest players abort at the same time, as compared with selective aborts where the Adversary can choose which players will abort separately. On top of that, identifiable abort means that all honest players agree on the malicious party responsible for the failure of the protocol.

rounds.

Usage of Classical Primitives. Regarding classical primitives, [LRW20] only requires secure coin-tossing and authenticated broadcast channels (information-theoretically secure since they can rely on an honest majority). We only use our Classical SMPC to perform coin-tossing, basic string operations (array lookup) and computations in $\mathbb{Z}_8$ and $\mathbb{Z}_2$. The Classical SMPC is more complex in [Dul+20; Alo+20] since it must be able to sample uniformly at random and perform computations on the classical descriptions of arbitrary Cliffords.

Rounds of communication. We can now quantify more precisely the number of classical rounds of communication or calls to the Classical SMPC resource, quantum rounds of communication, and size of quantum memory required by each participant in the protocol. [Dul+20] calls the Classical SMPC very often: a constant number of times for each input qubit and gate in the circuit. But the most costly part is the generation of ancillary magic states (for implementing T gates via gate-teleportation), which requires $\mathcal{O}(\eta(n+t))$ invocations of the Classical SMPC. Our protocol simply uses $d+3$ calls to this Resource, 2 for setting up the state and 2 for the key-release step. This is equivalent to the classical communication requirements of [LRW20], where they only need $d+2$ classical broadcasts per participant (with one for setting up the shared randomness and another for the state preparation). If all quantum communications are done in parallel in [LRW20], it can be further parallelised to only require a constant number of classical broadcast rounds. The protocol of [Alo+20] uses FHE (classical and quantum) to perform the computation and consequently the number of calls to the Classicl SMPC is only constant. We note that using a classical primitive called functional encryption, where a party in possession of an evaluation key can recover the clear-text of a function of the encrypted values (and only that), would allow to attain the same result for our construction by allowing the Server to compute the next measurement angle as a function of the encrypted secrets and previous measurement results.

The protocol of [Dul+20] requires numerous rounds of quantum communication as they need to send encoded states around for the verification of inputs and T and CNOT gates. After parallelisation the total cost is $\mathcal{O}(nd)$ quantum rounds. [Alo+20] aims to remove the circuit dependency in the number of rounds, obtaining $\mathcal{O}(n^4)$ quantum

rounds in the worst case in the case where the protocol is parallelised.[7]

Qubit Count and Memory Requirements. [LRW20] seeks to optimise the quantum memory requirement of players and therefore their communication is done sequentially, yielding $\mathcal{O}(\eta^2(n+t))$ quantum rounds. Parallelisation lowers it to 3 (or 2 for classical outputs), at a higher quantum memory cost for all parties. Our protocol is optimal as there is only a single quantum round (in the parallel case): sending to the Server all states required for the collaborative state preparation phase.

Finally, the number of qubits required by [Dul+20] during the computation phase is $\mathcal{O}(\eta(n+t))$ for each participant (they encode each of their input qubits, ancillae and magic states using $\mathcal{O}(\eta)$ qubits). However they use $\mathcal{O}(\eta^2(n+t))$ additional qubits in the offline phase to prepare the ancillary qubits (if the quantum communications are performed in parallel). On the other hand, [LRW20] reduces the number of qubits for each participant to $\mathcal{O}(n^2)$ for sequential quantum communication, but this blows up to $\mathcal{O}(\eta^2 n(n+t))$ if parallelised. The construction from [Alo+20] uses a compiler that adds automatically a cost of $\mathcal{O}(n^2)$ for each base qubit. The costly double encryptions and multiple layers of traps, in particular for the magic state distillation procedure, yields a total quantum memory cost per participant of at least $\mathcal{O}(tn^9\eta^2)$ (this is a weak lower bound). In our proposal the Server needs $\mathcal{O}(nd)$ qubits to perform each blind computation or test. Each qubit in these graphs is generated using n qubits via the Collaborative RSP Protocol and the computations and tests are repeated $\mathcal{O}(\eta)$ times in total, resulting in a total qubit cost of $\mathcal{O}(\eta n^2 d)$ for parallelised quantum communication but only $\mathcal{O}(nd)$ if the rounds are performed sequentially. However, the Clients can prepare these states on the fly and the Clients do not need quantum memory.

6.5.2 Impossibility of Single-Qubit Privacy Amplification on the Whole Bloch Sphere

The construction and security of Protocol 7 relies on the composition of a collaborative encryption gadget with the regular robust VBQC protocol driven by the Orchestrator.

The crucial features of the collaborative encryption are that (i) a single honest Client providing a random state from the allowed input set is enough to randomize the output

[7] They send states along a path of size n^2 in the communication graph of the parties, and remove a party if it doesn't deliver a packet before resending the states along a different path of the same size. In the worst case where there are $n-1$ malicious players which do not want to get caught cheating, they can drop $(n-1)(n-2)/2$ packets without being disconnected from the communication graph.

of the gadget, and (ii) no information about the state provided by an honest Client leaks to the Server. These are the two properties that were shown to hold in § 6.3.

It can also be seen that those do not hold whenever the set of input states for the clients not only comprise the 8 $|+_\theta\rangle$ states in the $X-Y$ plane but also the computational basis states $|0\rangle, |1\rangle$. The reason is that in such case, if the central qubit is set to a computational basis state, it cannot be randomized by the states provided by other clients.

While this specific failure is contingent to the chosen transformation implemented by our gadget, we will show here that it is indeed a more generic problem that gadgets fulfilling (i-ii) have in common, thereby restricting these "gadget-assisted" approaches to verification of classical input classical output computations.

First, we give a mathematical definition of (i):

Definition 6.5.1 (Randomizing gadget)**.** *Let P be a protocol with two Clients and one Server such that it takes a quantum state at each Client's input interface, and produces a quantum state at the Server's output interface together with a common classical bit string at each of the parties output interface.*

We say that this gadget is randomizing *whenever conditionned on the value of the common bit string, the linear maps implemented by the protocol when one of the two input states is fixed is invertible for pure input states.*

The motivation for this definition is simple: whenever one of the input is fixed, then the other one is enough to randomize the output at the Server's side. The role of the common bit string shared by all the parties at the end of the protocol is to allow the possibility of having a linear map that depends on this bit string as it is the case in our construction. As a consequence, the output state at the Server's interface might not be normalized in order to encapsulate the probability of a specific common bit string to be produced by the protocol.

The following Lemma 6.5.3 shows that for a fixed common string, there will always exist a specific state for one of the two inputs such that the map will not be invertible. This can result in one of the two following cases. Either the output is a fixed non-zero quantum state or it produces the null vector. In the former, this implies that the gadget is not able to correctly produce random states required by the VBQC protocol to be secure. In the latter, observing a specific common bit string excludes some input state for the honest Client, thereby also violating the assumptions required to obtain the security of the whole protocol.

Lemma 6.5.2. *Every two-dimensional linear subspace $V \subset \mathbb{C}^{2\times 2}$ contains at least one nonzero, singular matrix.*

Proof. Let $A, B \in V$ form a basis of V. If A or B is singular, the claim is trivial. Assume henceforth that both A and B are invertible.

For $\alpha \in \mathbb{C}$, let $C_\alpha = A + \alpha B$. Clearly, $C_\alpha \in \mathrm{span}(\{A, B\})$. Since A and B are linearly independent, $C_\alpha \neq 0$. It further holds that

$$\begin{aligned}\det(C_\alpha) &= \det \begin{bmatrix} a_{11} + \alpha b_{11} & a_{12} + \alpha b_{12} \\ a_{21} + \alpha b_{21} & a_{22} + \alpha b_{22} \end{bmatrix} \\ &= (a_{11} + \alpha b_{11})(a_{22} + \alpha b_{22}) - (a_{12} + \alpha b_{12})(a_{21} + \alpha b_{21}) \\ &= \alpha^2(b_{11}b_{22} - b_{12}b_{21}) + \alpha(a_{11}b_{22} + b_{11}a_{22} - a_{12}b_{21} - b_{12}a_{21}) + a_{11}a_{22} - a_{12}a_{21} \\ &= \alpha^2 \det(B) + \alpha(a_{11}b_{22} + b_{11}a_{22} - a_{12}b_{21} - b_{12}a_{21}) + \det(A). \end{aligned} \tag{6.35}$$

As $\det(B) \neq 0$, this is a polynomial of degree 2 in the variable α. By the fundamental theorem of algebra, this polynomial admits at least one complex root. □

Lemma 6.5.3. *There exists no linear map $\Xi : \mathbb{C}^{2\times 2} \to \mathbb{C}^2$ such that for all nonzero $v \in \mathbb{C}^2$ both $\Xi(\cdot \otimes v)$ and $\Xi(v \otimes \cdot)$ are invertible.*

Proof. Assume the existence of such a map Ξ. By the rank-nullity theorem, it holds then that

$$\dim(\mathrm{Ker}(\Xi)) = \dim(\mathbb{C}^{2\times 2}) - \dim(\mathrm{Im}(\Xi)) \geq 2. \tag{6.36}$$

By Lemma 6.5.2, there exists a rank-one matrix $C \in \mathrm{Ker}(\Xi)$. We can rewrite $C = vw^T = v \otimes w$ with nonzero vectors $v, w \in \mathbb{C}^2$. It follows that $\Xi(v \otimes w) = 0$ which contradicts the invertibility of $\Xi(\cdot \otimes w)$ and $\Xi(v \otimes \cdot)$. □

This leads us to conclude that such gadget assisted approaches will inherently be limited to classical I/O computations.

6.5.3 Open Questions

This work closes a gap between the circuit and MBQC models regarding secure multi-party computations. It shows that both are able to perform the required lift from classical to quantum in a statistically secure way, in spite of the more stringent requirements the

delegation imposes on what clients can do. Yet, this is only partially satisfactory as we do not consider the quantum input/output case. This specific question was considered by some of the authors. This lead to designing a protocol that was similar in spirit to the one presented here, but where the Collaborative Remote State Preparation would not only be able to prepare states in the $X - Y$ plane, but also dummy qubits. An attack on this protocol is analysed in § 6.6. It's discovery initiated the current work using dummyless verification as a way to avoid it. Yet, we also show in § 6.5.2 that such approach based on Collaborative Remote State Preparation outside a single plane is not likely to succeed, thereby leaving open the question of how to perform Delegated QSMPC with quantum I/O.

Other open questions regarding SMPC in the MBQC model include the verification of sampling with possibly better than polynomial security bounds. The question of the delegation of fault-tolerant computation in the MBQC model is also a long standing open question that we believe can benefit from the theoretical tools developed in [Kap+22] and from an approach similar to the one exemplified in this work.

Finally, [Ma+22] showed how to blindly delegate quantum computations with trusted rotations, even if both state preparations and measurements are untrusted, but left open the question whether verification is possible in this setting. The difficulty of verification seems to stem from the fact that (i) their analysis concerns states in the $X - Y$ plane, but not dummy states, and (ii) the remotely prepared states are blind, but not necessarily verifiable. While this work does not overcome the second obstacle, it shows that verification is indeed possible without the remote preparation of dummy states, and therefore constitutes a step towards the solution of this open problem.

6.6 Appendix: Post-Mortem of Previous Protocol

An earlier proposal for QSMPC [Kap+21] claimed similar results, but suffers from serious security shortcomings. We show here the limits of the design and discuss possible paths towards fixing it.

State-Selective Flipping Attack. The principle of the previous protocol was to separate the computation in two parts. The first section, which is blind only and not verifiable, is responsible for preparing the verifiable graph state from [KW17b], i.e. a single graph state which includes traps. This requires to prepare both rotated qubits and dummies. The Collaborative RSP prepares only rotated states which must then

either be transformed into dummies $\{|0\rangle, |1\rangle\}$ or left undisturbed (for computation and trap qubits). This is done with a blind computation on all these qubits, the additional qubits required for this computation being also generated using the Collaborative RSP. This is essentially a way to extend the Collaborative RSP to a bigger set of states.

The blindness of this gadget is proven in the Abstract Cryptography framework, so it would seem that it can be composed with the single-Client SDQC protocol to yield QSMPC in the same way as in the current work. However, this is not the case since we do not verify that the Server acts honestly so the final state is correct up to a global deviation. In general this deviation depends on the state that is being prepared, in particular the deviation can depend on whether the final state is a dummy or computation/trap qubit. Conscious of this, [Kap+21] exhibit a number of sufficient conditions on the computation so that this global deviation is independent of the secret state the the Clients prepare collaboratively.

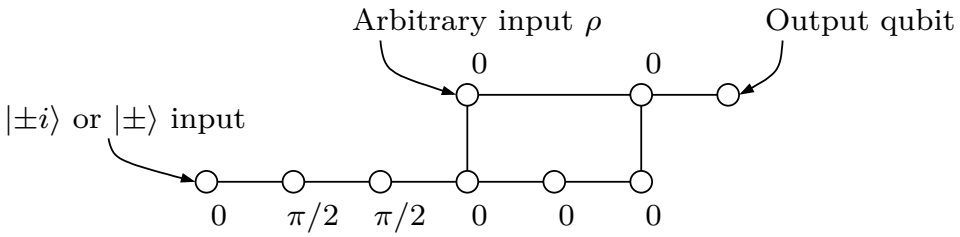
These conditions are as follows:

1. The inputs in the graph of the Clients' desired computation have degree 1.

2. All measurement angles are from the set $\{k\pi/2\}_{0\leq k\leq 3}$, i.e. the computation is Clifford.

3. The graph, flow of this computation and the angles of all vertices beyond the first layer of the gadget are independent of the final desired state of the qubits.

This final condition restrains a lot the possible types of computations that can be performed in this step but [Kap+21] proposes a scheme which seems to satisfy them. We recall it here for completeness. The MBQC pattern is given by the following graph and angles.



The qubit which must be transformed is denoted ρ (upper left qubit) and the lower left qubit's state is chosen depending on whether the upper qubit should be turned into a dummy or not. We refer to [Kap+21] for details. In order to be correct, it requires an

additional correction step after this computation. The correction depends on the state of the second input qubit and the measurement outcome of the last qubit in the lower line.

2nd qubit input	**Outcome**	**Correction**	**Effect**
$\|+_i\rangle$	0	Y	I
$\|+_i\rangle$	1	Y	I
$\|-_i\rangle$	0	I	I
$\|-_i\rangle$	1	I	I
$\|+\rangle$	0	X	H
$\|+\rangle$	1	Z	H
$\|-\rangle$	0	Z	H
$\|-\rangle$	1	X	H

Unfortunately, this correction depends on the final state. More precisely, by flipping the value reported as measurement outcome, the Server can apply an Y operation on dummy qubits and leave the rest unaffected. This flips selectively the state of dummies only, even if the server does not know that the qubit being prepared is in fact a dummy. We show in the next section how this breaks the verifiability of the protocol. The main take-away is that any correction applied after the computation does not also depend on the final state.

A potential patch was constructed using a more compact setup which seemingly satisfies all conditions. The graph that is used consists of a three vertex line for each final qubit. The first qubit in the line is measured either with an angle $\pi/2$ for dummy vertices or 0 for other positions. The second vertex will always be measured with an angle of $\pi/2$. This is presented below in Figure 6.3.

In the first case, the operation that is applied is $\mathsf{HZ}(\pi/2)\mathsf{HZ}(\pi/2) = \mathsf{X}(\pi/2)\mathsf{Z}(\pi/2)$, which has the effect of transforming the state $|+\rangle$ first into $\left|+_{\pi/2}\right\rangle$ with the Z-rotation and then into $|0\rangle$ via the X-rotation. Note that this does not correspond to a Hadamard gate since it transforms $|-\rangle$ into $-i\,|1\rangle$, but it is sufficient for our purposes. In the second case, the operation correspond to $\mathsf{X}(\pi/2)$, which has no effect when applied to a $|+\rangle$ state up to a global phase. Since all qubits are rotated $|+\rangle$ states, the rotation of the last qubit in the three vertex line graph re-encodes the state if the final state is not a dummy. This yields the full set of states required by the SDQC protocol.

(a) Measurement pattern for rotated qubits.

(b) Measurement pattern for dummy qubits.

Figure 6.3: DBQC measurement pattern applied to each qubit in the verifiable graph. The vertices surrounded with squares are inputs, round vertices are measured, diamond vertices are outputs. Blue vertices correspond to a measurement angle of $\pi/2$ while white vertices are measured with angle 0.

Note that once again, the conditions appear to be satisfied. Also, there are no post-processing steps beyond the standard MBQC flow corrections. However, here the input states do not span the full 8 rotated states, but are always considered as $|+\rangle$ states and they are re-encrypted via the rotation of the last qubit. By applying Z on the input before the computation and the output after the computation, it is also possible to selectively flip dummies only: the two Zs will cancel out for rotated qubits, but the second Z will have no effect on dummies while the first Z will flip the dummy.

Attack from Selective Dummy Flipping. We describe here an attack on the VBQC scheme of [KW17b], assuming that the Adversary can flip the value of dummy qubits (without affecting the computation and traps). We assume for this section knowledge about the Dotted-Triple Graph construction of [KW17b]. Consider a line graph of two qubits and its transformation in a Dotted-Triple Graph. This graph contains two primary locations with three qubits and one added location with nine qubits.

Through the application of CZ gates to construct the graph, flipping the value of a dummy is equivalent to applying Z on all adjacent qubits. For a given qubit in the graph, the global effect is $\mathbb{I}$ if an even number of adjacent dummies are flipped, and Z if an odd number of adjacent dummies are flipped. As we wish to disrupt the computation but not affect traps, the key to our attack is to use the difference in the number of dummies in the neighbourhood of traps and computation qubits. Traps are only linked to dummies while a computation qubit will always have at least one other computation qubit among its neighbours. As shown in Figure 6.4 below, we selectively flip added vertices so that each primary vertex is linked to exactly two attacked added vertices.

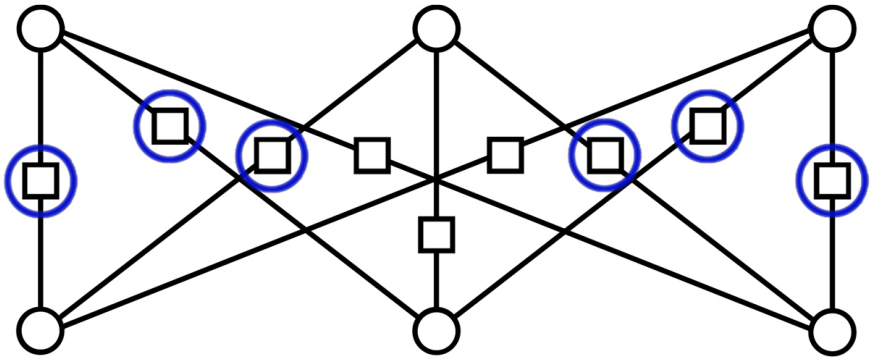
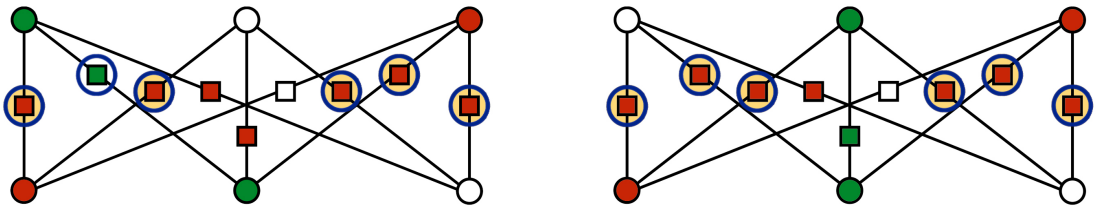


Figure 6.4: Example of attack layout where each top and bottom primary qubit is attached to exactly two attacked added qubits. Qubits that have been chosen for the attack are circled in blue.

In that case, since the primary trap qubits are only linked to dummies, the attack does not trigger either trap (if one of the middle qubits that is attacked is a trap, the effect of the attack on this trap is $\mathbb{I}$ as explained above). However, the attack may either affect two dummies linked to the primary computation qubits, in which case there is no attack since the effects cancel out, or one added dummy and the added computation qubit. Then, the effect on the added computation qubit is $\mathbb{I}$ but the attacked dummies will apply a Z operation on primary computation qubits on both sides of the link. If we assume fixed (but unknown) attack positions, whether this attack succeeds in modifying the computation depends only on the colouring that is used, while never triggering any trap. The probability of success is equal to $2/3$: the attack succeeds if the computational added qubit is chosen for the attack, there are 6 possible choices of attack configuration and each added qubit is left untouched by 2 out of the 6 attack configurations. We give in Figure 6.5 two possible colourings, ones in which the attack has no effect one the computation while the other corrupts it.

Extension and Take-away. Essentially, allowing an attack to depend on the nature of the qubits introduces new attacks compared to those that are possible in the plain VBQC Protocol. We have shown above that even a simple attack of this type is sufficient to break verifiability.

This attack is not specific to the construction of [KW17b] and can also be applied to the robust SDQC Protocol of [Lei+21]. There, a trap is also always linked to dummies



(a) Z attack on both primary computational qubits due to odd number of attacked added dummies.

(b) No attack on either primary computational qubit due to even number of attacked added dummies.

Figure 6.5: Two colourings of the previous graph (computational qubits are green, traps are white and dummies are red) for the same attacked qubits but a different effect on primary computational qubits. Attacked qubits are circled in blue, which translates to an X effect on dummies (yellow-filled circle) and no effect on added computational qubits (empty circle). The primary trap qubits are never affected by the attack since they are always attached to an even number of attacked added dummies.

but computation qubits are never linked to dummies (the test and computation graphs are separated as in the current work). Applying the selective flip and apply a Z on all neighbours will corrupt the computation but leave traps unaffected.

There are two main ingredients to these attacks: (i) the possibility for the server to selectively attack qubits depending on whether they are dummies or rotated qubits, and (ii) the difference in the neighbourhoods of computation and trap qubits in terms of dummies. Regarding the first point, the sufficient conditions above are very restrictive as to what types of computations can be performed to generate a wider range of states. Together with the result from Section 6.5.2, this is a strong indication that starting from a Collaborative RSP for a restricted set of states and expanding it is hard to construct securely. As for the second point, if it is possible to construct an SDQC protocol in which the effect of flipping any number of dummies is the same on the tests and computations, then this attack would have no effect beyond what the SDQC protocol already protects. This proposal provides a solution to this problem by removing dummies altogether in the case of classical inputs and outputs. Other directions can be explored as well in order to construct a protocol which resits these attacks and handles quantum inputs and outputs.

We note that at no point do we break the theorem from [Kap+21] proving the sufficient conditions for constructing an MBQC gadget for blindly generating an SDQC resource state up to state-independent deviations. However, it shows that the following

conditions were implicitly assumed in the proof: (i) the starting states should span the full range of rotated $|+_\theta\rangle$ states, and (ii) any post-processing should be independent of the final state.

Chapter 7

Verifiable quantum computing with trapped ions

We present the first hybrid matter-photon implementation of verifiable blind quantum computing. We use a trapped-ion quantum server and a client-side photonic detection system connected by a fibre-optic quantum network link. The availability of memory qubits and deterministic quantum logic enables interactive protocols without post-selection – a requirement for any scalable blind quantum cloud server which previous realisations could not provide. Our apparatus supports guaranteed privacy with < 0.001 *leaked bits per qubit and shows a clear path to fully verified quantum computing in the cloud.*

This chapter is based on the paper "Verifiable blind quantum computing with trapped ions and single photons" [Drm+23b] which is joint work with Peter Drmota, David Nadlinger, Dougal Main, Bethan Nichol, Ellis Ainley, Atul Mantri, Elham Kashefi, Raghavendra Srinivas, Gabriel Araneda, Christopher Ballance, and David Lucas, and which was accepted for publication in Physical Review Letters (PRL).

7.1 Introduction

Quantum computers are poised to outperform the world's most powerful supercomputers, with applications ranging from drug discovery to cyber security. These computers harness quantum phenomena such as entanglement and superposition to perform calculations that are believed to be intractable with classical computers. As quantum processors control delicate quantum states, they are necessarily complex and physical access to

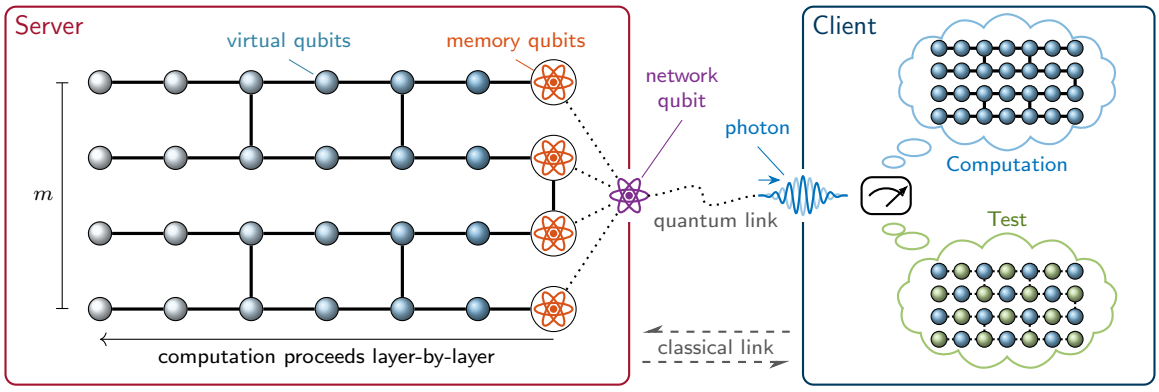


Figure 7.1: Verifiable blind quantum computing in the measurement-based model. The computation is expressed as a sequence of measurements on a brickwork state (two-dimensional graph with vertices representing virtual qubits, and edges indicating CZ gates). The server (left) holds m physical memory qubits (orange atoms) and one physical network qubit (violet atom). The server can entangle these qubits deterministically with each other. The network qubit can also be entangled with a photon; by measuring this photon, the client (right) can steer the network qubit in the server remotely without the server learning about its state. This allows the client to hide the computation (inputs, outputs, and circuit) from the server. Moreover, the client can verify that the computation has not been tampered with by interleaving test rounds, which produce classically simulatable outcomes and cannot be distinguished from the actual computation by the server.

high-performance systems is limited. Cloud-based approaches, where users can remotely access quantum servers, are likely to be the working model in the near term and beyond; many users already perform computations on commercially available devices for state-of-the-art research [Sar+19; ALP20; Pro+22; Ama+22; Kir+22].

However, delegating quantum computations to a server carries the same privacy and security concerns that bedevil classical cloud computing. Users are currently unable to hide their work from the server or to independently verify their results in the regime where classical simulations become intractable. Remarkably, the same phenomena that enable quantum computing can leave the server "blind" in a way that conceals the client's input, output, and algorithm [BFK09; FK17; GKK19]; because quantum information cannot be copied and measurements irreversibly change the quantum state, information stored in these systems can be protected with information-theoretic security, and incorrect operation of the server or attempted attacks can be detected – a surprising possibility which has no equivalent in digital computing. Blind quantum computing (BQC) requires not only a universal quantum computer as the server, but also a quantum link connecting it to the client [Bad+20; Coj+21]. Photons are a natural choice to

provide that link, and indeed the first demonstrations of (BQC) were performed in purely photonic systems [Bar+12; Bar+13; Fis+14; Gre+16]. However unavoidable photon loss, either due to limited photon detection efficiencies or absorption in the link, results in potential security risks [Bar+12; Fis+14] and places hard limits on the scalability of this approach due to the resource overhead incurred by post-selection [Li+15]. Ideally, quantum information at the server should be stored in a stable quantum memory that can be manipulated with high fidelity, yet interfaced readily to a photonic link. The ability to retain quantum information on the server then facilitates the client to perform adaptive mid-circuit adjustments in order to execute the target computation deterministically and securely. Combining two completely different platforms at the single-quantum level is technically challenging [Pfa+14; Huc+15]; so far, quantum network nodes with integrated memory qubits have been realised with solid state systems [Kal+17; Sta+22] and trapped ions [Drm+23a].

Here, we demonstrate (BQC) using a trapped-ion quantum processor (server) that integrates a robust memory qubit encoded in $^{43}Ca^{+}$ with a single-photon interface based on $^{88}Sr^{+}$ to establish a quantum link to the client (photon detection system). We implement an interactive protocol, where the client can remotely prepare single-qubit states on the server adaptively from shot to shot using real-time classical feedforward control. The complexity needed for universal quantum computation is contained entirely within the server, while the client is a simple photon polarisation measurement device that is independent of the size and complexity of the algorithm and supports near-perfect blindness by construction. The client and the server are controlled by independent hardware and connected only by a classical signalling bus and an optical fibre. The combined system of server and client achieves noise levels below a certain threshold for which arbitrary improvements to the protocol security and success rate (robustness) are theoretically possible [Lei+21].

7.2 Protocol

Quantum algorithms can be described in the measurement-based quantum computing model, which prescribes a sequence of measurements on a highly entangled resource state [RB01; Nie06]. Information-theoretic blindness can be achieved, even against maliciously operating servers, if either the state preparation or the measurements are performed by the client [CLN05; BFK09; MF13a; Fit17]. To this end, the client must

ensure that the quantum information stored by the server appears maximally mixed to any adversary [Fis+14]. Additionally, the client encrypts all classical messages using a private secret key.

In the presence of noise, even a faithfully operating server produces erroneous results that are indistinguishable from nefarious deviations from the honest protocol [Aha+17; FK17; Bro18; GKK19]. Blindness allows the client to perform tests on the quantum resources provided by the server without leaking information, and subsequently verify the outcomes to establish confidence in the quantum operations performed by the server. The protocol implemented here achieves this by interleaving "computation" and "test" rounds; the latter use the same quantum resources as the former and are therefore indistinguishable from them. A statistical argument provides bounds for the security and robustness of this protocol for the important class of bounded-error quantum polynomial time (BQP) decision problems [Lei+21]. Accepting incorrect results would be considered a security issue, whilst a protocol that rejects all results cannot be considered robust against noise. The client accepts the result if the observed fraction of failed test rounds, p_{fail}, is below a chosen threshold, ω, which must be below the theoretical maximum, ω_{max}. If this condition is met, the overhead due to repetition is low: the probability of accepting an incorrect result decreases exponentially with the number of rounds. The minimum value for ω depends on the amount of noise in the devices. The client assumes a maximum expected test round failure rate, p_{max}, and chooses $\omega > p_{\text{max}}$ such that the probability of rejecting any result also decreases exponentially with the number of rounds, making the protocol robust to a limited amount of noise.

For universal quantum computation, particular graph states and a discrete set of single-qubit measurements, $\{\hat{B}_\varphi\}$, are sufficient [MDF17]. Without loss of generality, $\hat{B}_\alpha = \cos(\alpha)\mathsf{X} + \sin(\alpha)\mathsf{Y}$, where $\alpha \in \Theta = \{0, \pi/4, \ldots, 7\pi/4\}$, and X and Y are Pauli operators. Graph states are specific multi-qubit states in which vertices represent qubits and edges represent entanglement created by two-qubit CZ gates [Fig. 7.1], where $\mathsf{CZ} = |0\rangle\langle 0| \otimes \mathsf{1} + |1\rangle\langle 1| \otimes \mathsf{Z}$ is the controlled-Z gate, which applies the Pauli operator $\mathsf{Z} = |0\rangle\langle 0| - |1\rangle\langle 1|$ to the target qubit conditioned on the state of a control qubit. The qubits are measured in a fixed order, using the basis $\hat{B}_{\alpha_\ell}$ at node ℓ, where α_ℓ depends on the algorithm and on previous measurement outcomes. In test rounds, the graph state is broken down into isolated "trap" qubits by introducing Z eigenstates at adjacent vertices, so-called "dummies", which commute with the CZ gate.

Here we consider prepare-and-send protocols to implement blind quantum computations on a linear cluster state [Fig. 7.2]. Two physical qubits are sufficient to implement

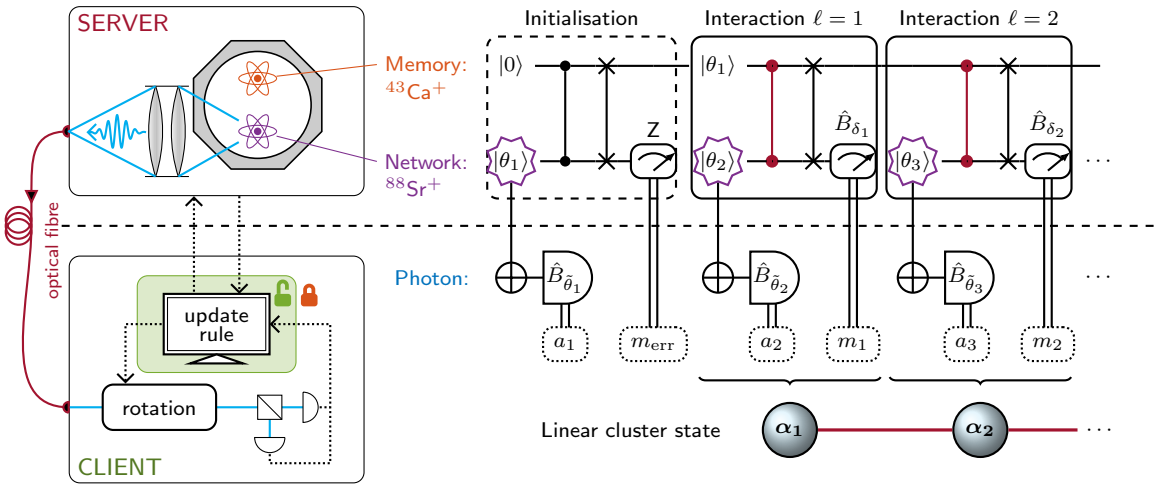


Figure 7.2: Protocol used to generate a linear cluster state using a trapped-ion quantum server and a photonic client. As the network qubit (violet, $^{88}Sr^+$ ion) is entangled with the emitted photon, the client can steer the state of the network qubit by measuring the polarisation of the photon. The outcome $a_\ell \in \{0, 1\}$ of a polarisation measurement in the basis $\hat{B}_{\tilde{\theta}_\ell}$ steers this qubit into the state $|\theta_1\rangle = \left|\tilde{\theta}_\ell + a_\ell \pi\right\rangle$. In the initialisation step, the server transfers this state onto a memory qubit (orange, $^{43}Ca^+$ ion) such that the network qubit can be steered again [Drm+23a]. Every subsequent interaction step extends the size of the cluster state; the client steers the network qubit remotely into $|\theta_{\ell+1}\rangle$, the server entangles it (red CZ gates), and performs a measurement in the basis $\hat{B}_{\delta_\ell}$, where δ_ℓ is provided by the client. See text for details.

linear clusters of arbitrary length, as qubits can be reinitialised after every mid-circuit measurement. The first qubit – the network qubit – can be steered into an arbitrary state by the client using remote state preparation (RSP) [Ben+01], while the second qubit – the memory qubit – carries the information encoded in the leading node of the expanding linear cluster state. To blindly run the measurement-based protocol outlined above with measurement angles α_ℓ, the client performs RSP into superposition states, $|\theta_\ell\rangle = (|0\rangle + \exp(\mathrm{i}\theta_\ell)\,|1\rangle)/\sqrt{2}$, with secret random phase reference, $\theta_\ell \in \Theta$, for every qubit $\ell = 1, 2, \ldots, q$ in the cluster computation. In the server's global phase frame, the measurement angles are then given by $(\alpha_\ell + \theta_\ell)$, where θ_ℓ acts as the classical encryption key for α_ℓ. To ensure that the corresponding measurement outcomes, $m_\ell \in \{0, 1\}$, are maximally mixed, the client hides bit flips in half of the measurement angles that are indicated by secret key bits, $r_\ell \in \{0, 1\}$ [Eq. (7.1)]. The client can recover the unencrypted measurement outcomes as $m_\ell \oplus r_\ell$. We break the cluster state into discrete interaction steps between the server and the client, starting after the initialisation step [Fig. 7.2], which prepares the memory qubit in $|\theta_1\rangle$. At each interaction of a computation

round, indexed by ℓ, the client performs RSP to steer the network qubit into $|\theta_{\ell+1}\rangle$ and communicates

$$\delta_\ell = (-1)^{R_{\ell-1}}\alpha_\ell + \theta_\ell + \pi r_\ell \tag{7.1}$$

to the server, where $R_\ell = \bigoplus_{1\leq j<\ell/2}(m_{\ell-2j} \oplus r_{\ell-2j})$ is the adaptive feedforward correction from decrypted previous measurements. After applying the CZ gate and a SWAP gate, the server measures the network qubit in the $\hat{B}_{\delta_\ell}$ basis and returns the result, m_ℓ, to the client [interaction blocks in Fig. 7.2]. This process leaves the state of the leading cluster state node on the memory qubit, encrypted by R_q, while the network qubit is available for further RSP. To complete the cluster computation, the client needs to specify the angle δ_{q+1} for a final measurement of the memory qubit. The client randomly assigns each round a secret label identifying them as a computation or a test, where the optimal proportion of rounds which are tests depends on the protocol parameters [Lei+21]. In test rounds, trap qubits are prepared in $|\theta_\ell\rangle$ and adjacent dummy qubits in the Z basis eigenstate $|r_\ell\rangle$. This step disentangles adjacent trap qubits, enabling the client to predict their outcomes, $m_\ell \stackrel{!}{=} r_\ell$, if they are measured with $\delta_\ell = \theta_\ell + \pi r_\ell$.

Server. The server controls an ion trap quantum processor containing one $^{88}Sr^+$ and one $^{43}Ca^+$ ion. Ion-photon entanglement needed for RSP is generated by fast excitation and spontaneous decay [Bli+04] on the 422 nm transition of $^{88}Sr^+$. The joint state of the photon polarisation and the spin state of the ion can be described by $|\Psi\rangle \propto |H\rangle|0\rangle + |V\rangle|1\rangle$, where $|H\rangle$ and $|V\rangle$ are orthogonal polarisation states. The single photons are collected by a high-numerical aperture lens and coupled into a single-mode optical fibre [Ste+20], which forms the quantum link with the client. The memory qubit [Drm+23a] is encoded in $^{43}Ca^+$, which provides a long coherence time and is unaffected by concurrent manipulation of $^{88}Sr^+$. Thus, $^{88}Sr^+$ can be used for mid-circuit measurements and sympathetic cooling between interaction steps. Errors during the initialisation step are detected in real time [$m_{\text{err}} = 1$ in Fig. 7.2] in which case the current round is aborted. The CZ gate required to build the cluster state is combined with the SWAP gate into an iSWAP gate. This enables reuse of $^{88}Sr^+$ for RSP whilst the current state of the computation is retained on the memory qubit until the client initiates further interactions, or is measured when the end of a round is reached. The coherence time of the memory qubit is $\sim$100 ms and can be extended to $\sim$10 s using dynamical decoupling [Drm+23a]. The ion trap server automatically performs

calibration routines to maintain the stability of the photonic interface, the micromotion environment [Nad+21], the magnetic field [Mai20], and laser detunings.

Client. The client receives single photons from the server through an optical fibre. The average time taken to obtain a single-photon herald is $\approx 100\,\mu s$; hence the probability for no herald to occur within the timeout period of 1 ms is $< 10^{-4}$. The quantum capability of the client is reduced to projective polarisation measurements of these photons in a basis that can be dynamically changed using a fast-switching polarisation analyser. The polarisation analyser [Fig. 7.3] can be configured to perform an arbitrary polarisation rotation by changing the voltages on two electro-optic modulators (EOMs). Following this rotation, a polarising beamsplitter and two avalanche photodiodes implement the polarisation measurement. This measurement remotely steers the network qubit into a state that depends only on the polarisation measurement basis and the measurement outcome obtained, information known exclusively to the client [$\tilde{\theta}_\ell$ and a_ℓ in Fig. 7.2]. Birefringence in the optical fibre transforms the photonic state before reaching the client by an unknown unitary operation, which drifts on a timescale of $\sim 10\,\text{min}$ due to thermal effects. To compensate for this drift, the client periodically recalibrates the EOM voltages [Fig. 7.3(c)].

Blindness. We now consider information that could leak to an adversarial server, concerning the client's choice of photon measurement basis. We distinguish between information leaked via the network qubit, which is controlled by the server, and leakage through classical signals, which are controlled by the client [Table 7.1]. In our demonstration, mismatched electronic delays between heralds corresponding to different polarisation measurement outcomes are the dominant cause for information leakage in both these cases. However, as the client is in full control of the relevant classical signals, these issues could be eliminated and information leakage to the server could be reduced to ~ 0.001 bits per interaction step. The remaining leakage would be dominated by imperfections in the polarisation optics used by the client to perform the photon polarisation measurement.

7.3 Results

We realise different quantum computations with one and two interaction steps, see Figs. 7.4(a) and 7.4(b), respectively. We could use the output qubit in further interaction

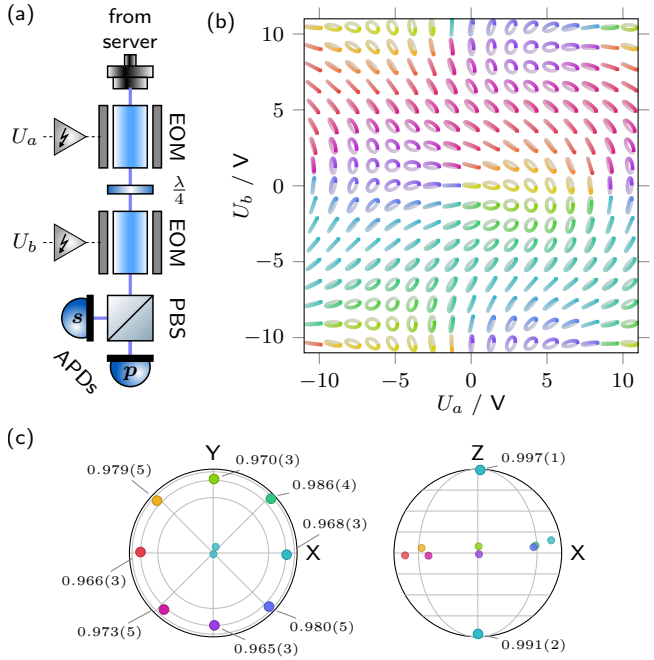


Figure 7.3: The client uses a fast-switching polarisation analyser to perform remote state preparation (RSP). (a) The control voltages of two EOMs separated by a $\lambda/4$ waveplate enable the client to arbitrarily rotate the polarisation measurement basis given by the PBS. (b) Precalibration measurements with laser light are used to reconstruct the measurement basis implemented by the device as a function of the control voltages U_a and U_b. Polarisation ellipses are shown for the basis states heralded by detector p, where the colour represents the phase of these states. (c) To find the control voltage settings U_a and U_b which maximise the fidelity to each of the 10 target states needed during the protocol, we perform tomography on the network qubit after RSP over a range of control voltages. The averaged results from 36 calibrations performed over 2 weeks of operation are shown in the Bloch sphere representation of the network qubit. Values indicate the fidelity of the states to the pure target state, with standard deviations obtained from bootstrapping.

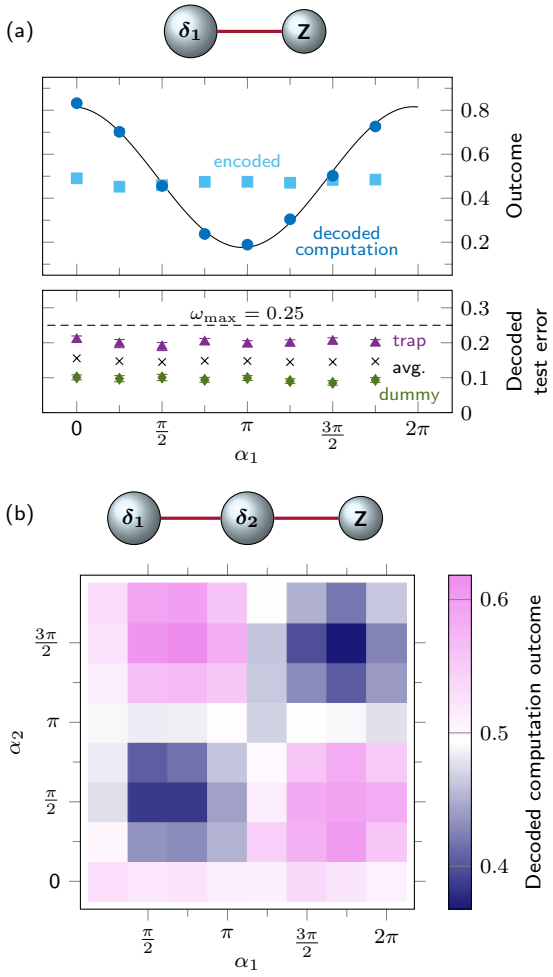


Figure 7.4: Experimental results on an expanding linear cluster state, where the leading qubit is measured in the Z basis after (a) one interaction step, and (b) two interaction steps between the client and the server. (a) While the server observes mixed outcomes (squares), the client can decode the results using the secret keys. A fit (solid curve) to the decoded computation outcomes (circles) is shown to guide the eye. The expected average test round error for the two-node cluster state (crosses) is below the threshold for verification (dashed line). Error bars indicate binomial standard errors. (b) The decoded computation outcome is shown as a function of blind measurement angles α_1 and α_2.

Channel	Source	Method	Leakage / bits	
			Observed	Optimised
classical	measurement angles	enforced	0	0
	heralding efficiency	inferred	0.00006	0.00006
	heralding delay	inferred	0.35	0.00007
quantum	measurement basis	inferred	0.035	0.0007
	imbalanced outcomes	inferred	0.00029	0.00026
	—	measured	0.031(4)	—

Table 7.1: Sources of information leakage. The optimised values assume matching heralding delay (from excitation of $^{88}Sr^{+}$ until electronic detection) and balanced polarisation measurement outcomes, which could be achieved with minimal changes to the existing client apparatus. We use quantum state tomography to quantify the information that the server could gain from measuring the network qubit and find good agreement with independent estimates inferred from known imperfections in the photon polarisation measurement basis and imbalanced polarisation measurement outcomes. The values are to be compared with the number of bits of information (3 bits) that specify the client's photon measurement basis, θ_ℓ.

steps, or make a final measurement in the basis $\hat{B}_{\delta_{q+1}}$ to complete the $(q+1)$-node cluster computation. In this demonstration, however, the output qubit is always measured in the Z basis. Since this measurement commutes with the CZ gate preceding it, the computation result is equivalent to the result for a cluster state with one fewer node. The actions of the one- and two-step interactions are therefore given by the computations $\mathsf{H}Z(\alpha_1)\,|+\rangle$ and $X(\alpha_2)Z(\alpha_1)\,|+\rangle$, respectively, where H is the Hadamard gate, $X(\alpha) = \exp\left(-\mathrm{i}\frac{\alpha}{2}\mathsf{X}\right)$ and $Z(\alpha) = \exp\left(-\mathrm{i}\frac{\alpha}{2}\mathsf{Z}\right)$ are single-qubit rotations, and α_1 and α_2 are encrypted using Eq. (7.1) during the protocol. From the server's perspective, the outcomes appear maximally mixed [squares in Fig. 7.4(a)] as a result of the bit-flip encryption, $\delta_\ell \propto r_\ell \pi$, which is applied by the client in both the computation and test rounds. The client on the other hand can use the round type (computation or test) and encryption key (r_ℓ) to decode the outcomes. The decoded computation outcomes indicated by the circles in Fig. 7.4(a) and the colourmap in Fig. 7.4(b) match the expected fringe pattern as a function of the blind measurement angles α_1 and α_2. Experimental imperfections lead to a reduction in contrast and to phase shifts. The fraction of failed tests [bottom panel in Fig. 7.4(a)] is nevertheless low enough to perform a fully verified two-node cluster computation: we observe an error rate of $p_{\text{fail}}^{\text{trap}} = 0.201(3)$ on the trap qubit. Due to the Z basis measurement performed on the final qubit, the correctness of the dummy qubit can be verified as well: we obtain $p_{\text{fail}}^{\text{dummy}} = 0.095(2)$. The expected

average test round failure probability for the two-node cluster state is therefore ~ 0.15, which is significantly below $\omega_{\max} = 0.25$ required for secure and robust verification of the linear cluster state with two nodes. We could change the final measurement basis from Z to $\hat{B}_{\delta_{q+1}}$ with only one additional $\pi/2$ pulse, which would have no significant impact on the error budget. For the three-node cluster computation, the observed failure rates indicate that verification is not possible in this case, largely due to technical limitations on the ≈ 0.91 fidelity of the iSWAP gate [Drm+23a].

7.4 Discussion

We have implemented a protocol for blindly delegating quantum computations to a trapped-ion quantum processor, using a client apparatus that requires only single-photon polarisation measurements and classical communication. We have established bounds on the information that could be leaked to the server through both classical and quantum channels that are present in our implementation. We have shown that the size of the cluster state can be increased without increasing the number of physical qubits in the server and without modifications to the client hardware. If more memory qubits were added to the server [Wri+19], the computational space could be extended to higher-dimensional cluster states. We have taken steps to include verification into the protocol, and the measured test round error indicates that the amount of noise in the system is low enough to perform a fully verified computation on the two-node cluster state robustly and reliably. For a threshold $\omega = 0.18$ and $n = 20\,000$ repetitions (of which 9800 are test rounds), we predict that the probability of accepting an incorrect result of a BQP decision problem with small inherent algorithmic error would be 3×10^{-9}, and the probability of rejecting any result 2×10^{-9}. This approach is expected to provide both security and robustness for larger cluster states and other algorithms as long as the errors remain below the size-dependent threshold, $\omega_{\max} \approx 1 - (3/4)^{2/q}$, where q is the total number of qubits in the cluster state. As the protocol that we have implemented does not rely on error correction, one cannot expect to achieve scalability without reducing the errors per interaction step. We identify the infidelity of the iSWAP gate as the leading error source in this proof-of-principle demonstration. We note that in other systems, CZ gates between $^{88}Sr^+$ and $^{43}Ca^+$ with fidelity exceeding 0.995 have been demonstrated [Hug+20]. However, the state-of-the-art ion-photon entanglement fidelity of 0.979(1) (this setup) would also need to be improved further in order to meet the

requirements for fault-tolerance. We note that the ion-photon entanglement fidelity is limited primarily by technical imperfections in the optical setup.

In comparison with previous experimental implementations [Bar+12; Bar+13; Fis+14; Gre+16], which were based on purely photonic platforms without quantum memory, this work overcomes several major challenges associated with real-world BQC deployments. As quantum logic operations in the server are deterministic and the interaction with the client is heralded, our implementation eliminates the need for post-selection, avoiding the associated efficiency, scalability, and security issues [Bar+12; Bar+13; Gre+16]. Here, photon losses in particular do not present a security threat, and the use of a memory qubit combined with fast and adaptive hardware facilitates true shot-by-shot randomisation of all protocol parameters in real time.

Future realisations could involve a complex network of servers and clients. Photons could be routed to a number of clients using optical switches, and the distance to the server increased using frequency conversion of the photons to telecommunication wavelengths [Kru+19] or using recent developments in fibre technology [Fok+23]. The photonically interfaced trapped-ion quantum information platform demonstrated here paves the way for secure delegation of confidential quantum computations from a client with minimal quantum resources to a fully capable, but untrusted, quantum server.

7.5 Appendix: Remote state preparation by steering

In this section, we show that steering can be used to securely implement remote state preparation. In particular, only one-way quantum communication from the server to the client is sufficient, and the preparation of entangled pairs by the server does not need to be trusted.

Lemma 7.5.1. *Protocol 8 implements Resource 7 with perfect correctness.*

Proof. After the client's measurement, the remaining single-qubit state in the server's quantum register can be described by $U_1^\dagger \mathrm{X}^m |0\rangle$. The server's output therefore becomes $U_2 U_1^\dagger \mathrm{X}^m |0\rangle = U|0\rangle$. □

Lemma 7.5.2. *Protocol 8 implements Resource 7 with perfect security against a malicious server.*

Resource 7 Remote State Preparation

Inputs:

- Client: the classical description of a single-qubit unitary U.
- Server: no input.

Outputs:

- Client: no output.
- Server: the single-qubit state $U|0\rangle$.

Simulator 3

Instructions:

1. The simulator expects a single-qubit quantum state $|\phi_1\rangle$ as an input from the ideal functionality on its left interface.
2. It expects a single-qubit quantum state $|\phi_2\rangle$ as an input from the distinguisher on its right interface.
3. It samples a single-qubit unitary U_1 randomly from the Haar measure.
4. It applies the two-qubit unitary $U_1 \otimes \mathrm{I}$ to the state $|\phi_1\rangle|\phi_2\rangle$ and performs a Bell measurement on it, obtaining measurement outcomes m_1 and m_2.
5. It then sets $U_2 = U_1^\dagger \mathrm{Z}^{m_1}\mathrm{X}^{m_2}$ and outputs the classical description of U_2 on its right interface to the distinguisher.

Proof. As part of the proof, let σ be defined as in Simulator 3.

It remains to be shown that the composition of Resource 7 with σ (the ideal world) generates the same distribution on its interfaces as the client's instructions of Protocol 8 (the real world).

Let $|\psi\rangle$ be the purification of the distinguisher's quantum register just before sending the first qubit of its register to the client. Then, in the real world, after the client's measurement, the state of the server's quantum register can be described (up to a global phase) by

$$\langle 0|(\mathrm{X}^m U_1 \otimes \mathrm{I})|\psi\rangle, \tag{7.2}$$

and the classical message from the client contains the description of the unitary $UX^m U_1$,

Protocol 8 RSP by steering

Inputs:

- Client: the classical description of a single-qubit unitary U.
- Server: no input.

Required resources:

- Secure one-way quantum channel from server to client.
- Secure one-way classical channel from client to server.

Instructions:

1. The server prepares a two-qubit Bell state $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, and sends one of the qubits to the client.
2. The client samples a single-qubit unitary U_1 randomly from the Haar measure. It then applies U_1 to the state received by the server and performs a measurement on it in the computational basis, obtaining measurement outcome m.
3. The client sends the classical description of the single-qubit unitary $U_2 = U\mathrm{X}^m U_1$ to the server.
4. The server applies U_2 to the remaining single-qubit state, and sets it as its output.

where U_1 is chosen according to the Haar measure. Substituting U_1 for $\mathrm{X}^m U_1$ yields

$$\langle 0|(U_1 \otimes \mathrm{I})|\psi\rangle, \tag{7.3}$$

and the classical description of UU_1 without changing the distribution of U_1.

In the ideal world, after the simulator's measurement, the state of the server's quantum register can be described (up to a global phase) by

$$\langle \Phi^+|(\mathrm{Z}^{m_1}\mathrm{X}^{m_2}U_1 \otimes \mathrm{I})(U|0\rangle \otimes |\psi\rangle), \tag{7.4}$$

and the classical message from the client contains the description of the unitary $U_1^\dagger \mathrm{Z}^{m_1}\mathrm{X}^{m_2}$, where U_1 is chosen according to the Haar measure. This can be rewritten equivalently as

$$\langle 0|(U^\dagger U_1^\dagger \mathrm{Z}^{m_1}\mathrm{X}^{m_2} \otimes \mathrm{I})|\psi\rangle. \tag{7.5}$$

A change of variables from U_1 to $\mathrm{Z}^{m_1}\mathrm{X}^{m_2}U_1^\dagger U^\dagger$, without changing the distribution of U_1,

yields

$$\langle 0|(U_1 \otimes \mathrm{I})|\psi\rangle, \tag{7.6}$$

and the classical description of $(\mathrm{Z}^{m_1}\mathrm{X}^{m_2}U_1^\dagger U^\dagger)^\dagger \mathrm{Z}^{m_1}\mathrm{X}^{m_2} = UU_1$, which concludes the proof. □

Chapter 8

Multi-client blind quantum computing on the Qline

Universal blind quantum computing allows users with minimal quantum resources to delegate a quantum computation to a remote quantum server, while keeping intrinsically hidden input, algorithm, and outcome. State-of-art experimental demonstrations of such a protocol have only involved one client. However, an increasing number of multi-party algorithms, e.g. federated machine learning, require the collaboration of multiple clients to carry out a given joint computation. In this work, we propose and experimentally demonstrate a lightweight multi-client blind quantum computation protocol based on a novel linear quantum network configuration (Qline). Our protocol originality resides in three main strengths: scalability, since we eliminate the need for each client to have its own trusted source or measurement device, low-loss, by optimizing the orchestration of classical communication between each client and server through fast classical electronic control, and compatibility with distributed architectures while remaining intact even against correlated attacks of server nodes and malicious clients.

This chapter is based on the paper "Multi-client distributed blind quantum computation with the Qline architecture" [Pol+23] which is joint work with Beatrice Polacchi, Leonardo Limongi, Gonzalo Carvacho, Giorgio Milani, Nicolò Spagnolo, Marc Kaplan, Fabio Sciarrino, and Elham Kashefi, and which has been published in Nature Communications.

8.1 Introduction

Despite the increasing technological progress in the manipulation of many-qubit systems [Bao+23; CDG21; Aru+19], providing quantum computing as a service for end-users poses several challenges, including scalability, privacy, and integrity. Indeed, in order to achieve true quantum advantage from emerging devices, they must scale up beyond the current monolithically designed noisy intermediate-scale quantum regime. As of today, the only viable solution being pursued by all qubit platforms is modularity and interconnected architecture, where photonic links are considered the best option. Moreover, it is also clear that quantum machines need to be integrated into cloud services or data centers, allowing multiple clients to connect locally or globally to access these devices. In such a context, the issue of keeping the computation and data protected from malicious parties will be a key challenge for such large-scale adaptation. Notably, photonic links to quantum servers enable the capability of achieving informational security for delegated computing, known as blind quantum computing (BQC), which is not achievable using only classical communication between client and server [Aar+19]. Such a protocol builds on the measurement-based model for quantum computation [RB01; Rau09] that exploits mid-circuit measurements for teleportation-based quantum computing on encrypted quantum states sent to a remote server via a quantum link [BFK09; Lei+21]. Over the last decade, many BQC protocols have been proposed [Liu+23; Kap+22; Ma+22; GWK17; Aha+17; GKW15; PF15; HPF15; HM15; Mor14; MF13a; SKM13; MPF13; RUV12; MF12; DKL12], together with proof-of-concept experimental demonstrations in different settings [Drm+23b; SZ18; Hua+17; Gre+16; Tak+16; Mar+16; Fis+14; Bar+13; Bar+12]. However, the challenge of multi-client settings has been explored only theoretically due to the high resource requirements of the proposed protocols [Kap+23; SCY21; QW21; Cia+20; KP17].

Yet, a growing number of classical delegated computing tasks require that multiple clients collaborate to carry out a joint function, e.g. federated machine learning tasks [Kon+16; Yan+19]. Notably, quantum counterparts of such algorithms have been proposed as well [CY21], including a federated quantum machine learning protocol based on BQC [LLD21].

In this chapter, we propose a modular lightweight distributed architecture for multi-client BQC based on the recently proposed Qline quantum network link configuration [Doo+23], that enables scalable client insertion. With such an architecture in mind, we present a tailor-made multi-party blind quantum computing protocol such that the

clients in the Qline only own trusted single qubit rotation devices, while the overall protocol still provides privacy for the joint computation performed by several users on the cloud. In the Qline network architecture, the quantum resource is first generated by a potentially untrusted server, then distributed to the clients, such that each client can apply arbitrary single-qubit operations on the incoming qubits and, at the end of the line, measured by a second again potentially untrusted server. An analogous architecture was first introduced in [HBB99] for quantum-assisted secret sharing, and later used for various tasks such as quantum key distribution or secure computation [Cle+17; Gri+15; Sch+05]. The main advantages of such a structure reside in the possibility to integrate it easily into larger-scale networks, its compatibility with key establishment protocols [Doo+23], and its low hardware complexity. In order to simplify the resource requirements of the multi-client BQC protocol in [KP17; Kap+23], we show that, within such an architecture, it is enough that each client in the Qline adds a new layer of encryption to the flying qubits that will be used as the common key for their later private joint computation on the remote server. Such collaboration may be typical, for instance, of privacy-preserving machine learning algorithms where each client's input data and parameters related to the algorithm should remain private to all parties. To implement it, we employ a fibered photonic platform equipped with genuine measurement adaptivity to enable deterministic computation [RB01]. Within this setup, we are able to show the blindness and the correctness of the protocol, in both cases where the function to be computed has a classical or a quantum output. Our experimental proof-of-concept demonstrates a two-client scenario that can be easily extended to larger and more complex quantum networks featuring any number of clients at arbitrary distances.

8.2 A multi-client BQC protocol

In this section, we describe in detail the protocol proposed and successfully implemented in this work. It is built on the theoretical premises in [Kap+23; KP17], and tailored to a Qline architecture [Doo+23]. Differently from the single client original protocol of [BFK09] depicted in Fig. 8.1, the results of [Kap+23; KP17] enable multi-client BQC by exploiting secure multi-party computation (SMPC), whose aim is to allow several users to collaboratively compute a joint function on their private data. The classical SMPC functionality enables coordination of the parties in a delegated quantum

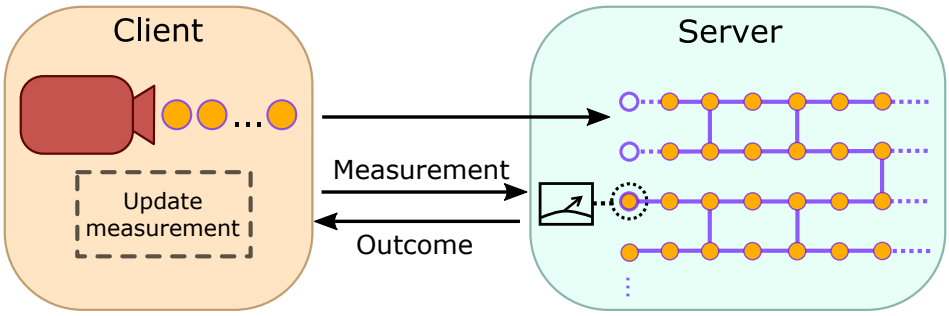


Figure 8.1: Conceptual scheme of BQC. In the BQC protocol, a client randomly prepares n qubits and sends them to a quantum server. The server uses the qubits to form a resource state and, at each of the $\mathcal{O}(n)$ rounds of the computation, it measures one qubit in the measurement basis given by the client. Then, it gives back the measurement outcome to the client, who decides accordingly on the next measurement basis. Figure inspired by [Fit17].

computing task, such that the full computation details are blind not only to the server, but also to potentially dishonest clients that collude with it. However, implementing such functionality would need additional rounds of classical communication among the clients and server, during which it would be unfeasible to coherently store the quantum state. Therefore, to optimize the storage time, in our implementation, we substitute classical SMPC with a trusted third party (TTP) that mediates the communication between the clients and the server reducing the number of rounds, while the blindness of the protocol is still proven against any strict subset of colluding malicious adversaries. In this way, the quantum state needs to be stored for significantly shorter times than if using full classical SMPC, thus enabling our first proof-of-concept experimental demonstration of a two-client BQC. To motivate our experimental design, this section is divided into two parts: the first one is devoted to the description of a two-client example, which we implemented experimentally demonstrating the key building blocks that are required for a fully scalable solution. In the second part, we present the extension to the n clients case and universal quantum computing resources.

8.2.1 The two-client protocol

Consider Alice and Bob wish to run a joint computation on a remote server. Alice has private classical data x_1 and x_2 while Bob has private gate parameters ϕ_1 and ϕ_2, and the target joint circuit is

$$\left(M^X \otimes M^X\right)\left(R_z(\phi_1) \otimes R_z(\phi_2)\right)\ \mathsf{CZ}_{12}\ \left(Z^{x_1} \otimes Z^{x_2}\right)\left(|+\rangle \otimes |+\rangle\right) \tag{8.1}$$

as shown in Fig. 8.2a. This is a typical building block of any large-scale privacy-preserving QML, such as the one proposed in [CY21]. In what follows we demonstrate the steps to make the above joint computation both distributed and secure as shown in Fig. 8.2b.

State preparation. A source of maximally entangled bipartite states, S_1, distributes two-qubit states along two quantum channels, of the form:

$$|\psi\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle - |11\rangle\right) \tag{8.2}$$

Alice receives the two qubits, and applies single-qubit z-rotations of angles θ_i^A to them, randomly chosen in the set $\mathcal{A} = \{0,\ \pi/4,\ 2\pi/4, ...\ 7\pi/4\}$. This will hide (via quantum one-time padding) her classical input data which would be encoded on these qubits via Z^{x_i} operations. Moreover, she chooses two random bits $r_1^A,\ r_2^A$ that will later hide the outcome of the computation. She communicates her secret parameters to the TTP. She then sends her two encrypted qubits to the second client, Bob, who applies further random $\theta_1^B,\ \theta_2^B$ z-rotations, again chosen from the set $\mathcal{A}$ to one-time pad his private algorithm parameter $\phi_1,\ \phi_2$. Bob also chooses two random bits $r_1^B,\ r_2^B$ for the encryption of the output as well. He then communicates his secret parameters to the TTP. From now on, we will use the following definitions: $\theta_i = \theta_i^A + \theta_i^B$ and $r_i = r_i^A \oplus r_i^B$. The resulting quantum state at this stage is the following:

$$|\psi\rangle = \frac{1}{2}\left(|00\rangle + e^{i\theta_2}|01\rangle + e^{i\theta_1}|10\rangle - e^{i(\theta_1+\theta_2)}|11\rangle\right) \tag{8.3}$$

This state is then sent to server S_2.

Interaction and measurement stage. From now on, the clients and S_2 only communicate classically, through the TTP. The protocol requires two rounds, one for each qubit to be measured. The blind measurement angle δ_i at the $i-$th round, for $i = 1,\ 2$, is computed by the TTP according to the formula:

$$\delta_i = \theta_i + x_i\pi + (-1)^{m_{(i-1)}^{true}}\phi_i + r_i\pi \tag{8.4}$$

where $m_0^{true} = 0$ and $m_1^{true} = m_1 \oplus r_1$. Analogously to the first measurement outcome, the outcome of the second measurement is decrypted according to the formula: $m_2^{true} = m_2 \oplus r_2$ before giving it back to the clients. A slight change in the protocol is needed

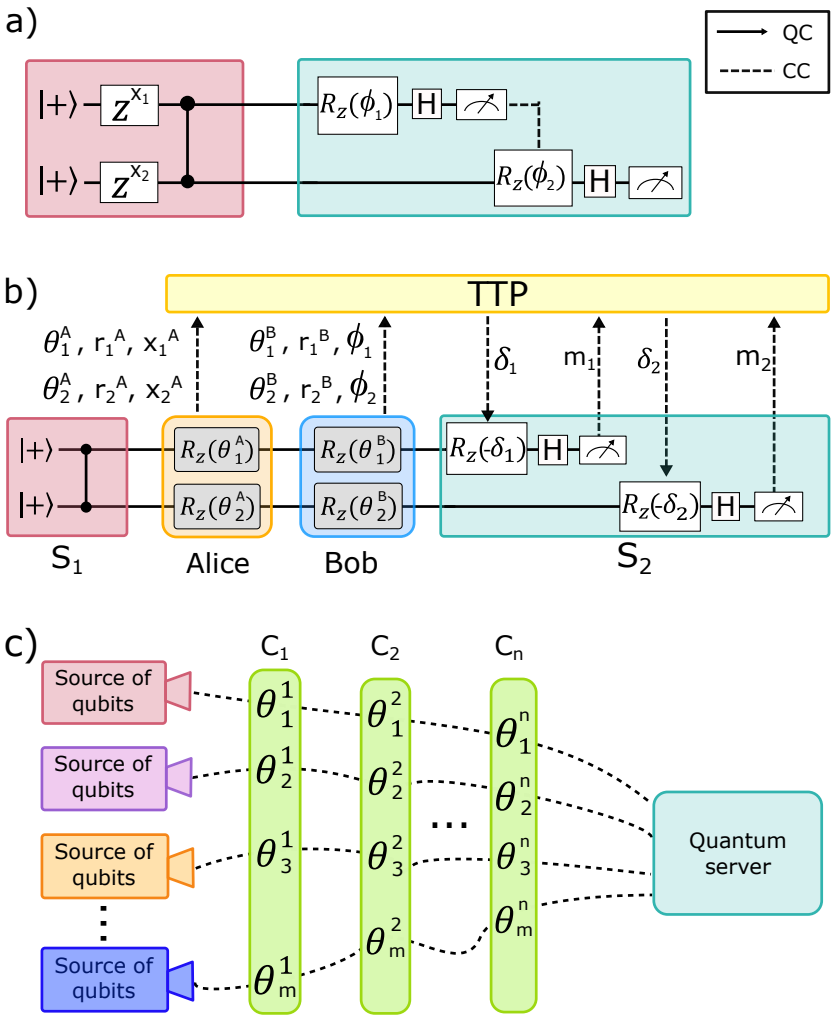


Figure 8.2: Conceptual scheme of the two-client BQC and distributed quantum computing over a Qline architecture. a) The desired joint quantum circuit computation where x_1 and x_2 are Alice's private input data and ϕ_1 and ϕ_2 the private angles of Bob's algorithm. b) The same computation of the circuit presented in a) is encrypted to preserve the privacy of each party's information. In our two-client BQC protocol, a quantum channel connects a source of bipartite quantum states to two clients disposed along the Qline. Each client chooses their secret parameters θ_i^j, r_i^j, x_i, ϕ_i, for $i = 1, 2$ and $j = A, B$, and applies z-rotations to both qubits. All secret parameters and measurement outcomes pass through a TTP to compute the transformed measurement bases δ_i and the corrected outcomes. At the end of the line, the quantum state is sent to the server S_2 to carry out the desired computation, which is carried out through two rounds of classical communication between clients, TTP, and the server S_2. c) Generalization of our architecture to m Qlines with n clients distributed along them. In a Qline, a quantum state source distributes single qubits and each client applies random rotations to all qubits. At the end of the line, a powerful quantum server employs the received qubits to generate the resource state for the computation and calculate a joint function.

in the case where a quantum function is computed, i.e. the output qubit must be prepared by the clients in the state $|+\rangle$ [BFK09]. The correctness of the protocol is straightforward and can be directly obtained from [BFK09; KP17; Kap+22]. However, the security proof is more subtle compared to previous works that were based on trusted state preparation for each client. We first present the generalization of our protocol and then provide the full proof of security that is applicable to this special case as well.

8.2.2 Generalization to the multi-client scenario

In this section, we generalize our protocol to a scenario where n clients want to perform a joint computation on a possibly larger resource state. Blindness should be guaranteed for any single honest client. We consider the target computation to be defined as a measurement pattern [RB01] by the measurement angles $(\phi_v)_{v\in V}$, where $v \in V$ ranges over all qubits in the resource graph state. These angles can be fixed and publicly known, or jointly input by any subset of clients. In the latter case, blindness holds for the measurement angles as well. The input qubits $I \subset V$ are partitioned into sets $(I_j)_{j\in\{1,\dots,n\}}$, where I_j belongs to the j-th client who has the bit string $x_j \in \{0,1\}^{|I_j|}$ as input. To keep each client's input blind, it is required that each qubit in the resource state travels once along the Qline and accumulates random rotations by all clients, as depicted in Fig. 8.2c. In this way, a resource state on $m = |V|$ qubits would require m Qlines. However, this process may be (partially) parallelized, in the sense that multiple qubits can be sent along the Qline at once – as long as the clients can perform the necessary rotations in parallel as well. After the server received the qubits that have passed through the Qline, it follows a standard execution of the BQC protocol (Protocol 2). As all communication from this point forward is entirely classical, the TTP governs the instructions to the server for the remainder of the protocol. A detailed description of the full multi-client protocol on scalable resource states is given in Protocol 9.

Note that the orchestrator from Protocol 9 must be trusted, but is an entirely classical party. To remove all trust assumptions on this party, it can be replaced with any composably secure classical SMPC protocol which is performed by the clients and the server. Moreover, much of the calculations that the orchestrator needs to perform, including the sampling of random coins and the evaluation of the formulae for the corrected measurement angles, can be done in a classical pre-processing step. The computation during the quantum phase then boils down to the choice of one of two possible measurement angles based on the previously reported measurement outcomes.

Protocol 9 Multi-Client Blind Quantum Computation

Public Information:

- A graph $G = (V, E, I, O)$ with input and output vertices I and O, respectively.
- A partition $(I_j)_{j\in\{1,\dots,n\}}$ of the input vertices, where I_j belongs to client j.
- A partial order $\preceq$ on the set V of vertices.

Inputs:

- Client j has a classical bit string $x_j \in \{0,1\}^{|I_j|}$.
- The n clients collaboratively input a set of angles $(\phi_v)_{v\in V}$, and a flow f on G compatible with $\preceq$.
- The orchestrator and the server have no inputs.

Protocol:

1. The n clients send all of their inputs to the orchestrator.
2. The orchestrator and the server perform the BQC Protocol S4. For every $|+_\theta\rangle$-state that the orchestrator would need to send to the server, they instead perform the following:
   (a) The server prepares a $|+\rangle$-state.
   (b) All parties perform one execution of the Collaborative Remote State Rotation Protocol S5, where the server uses the $|+\rangle$-state as an input, and the orchestrator inputs the angle θ.
   (c) The server uses the output state in the BQC Protocol S4.
3. The orchestrator distributes the classical output among the clients. The server sends the output qubits in O to the designated clients, with the orchestrator providing the decryption keys.

Finally, it is worth mentioning that the requirement that every client has access to each Qline can be weakened when accepting stronger trust assumptions. Blindness holds as long as there exists at least one honest client along each Qline. Therefore, even if not every client participates at each Qline, blindness is still guaranteed if we restrict the adversarial patterns to not corrupt all clients along any Qline at once.

In the concrete case of our experiment, the graph G is a two-qubit cluster state, that is, $V = \{1, 2\}$. Both qubits are Alice's input qubits, so $I_1 = I = V$, while Bob has no input qubits ($I_2 = \emptyset$) but chooses the measurement angles (ϕ_1, ϕ_2). We consider two different computations: in the first case, we interpret both measurement outcomes as the classical output of the computation, while in the second case, we consider the second qubit to be the quantum output of the computation, hence $O = \{2\}$.

8.3 Security

In our protocol, the clients' quantum abilities are restricted to receiving single-qubit states, applying a random z-rotation to it, and forwarding it to the next client or server. From the perspective of an honest client, this behavior is exactly captured by the *Remote State Rotation* (RSR) functionality introduced in [Ma+22]. The latter showed that RSR can indeed be used in the context of the BQC protocol to delegate a universal quantum computation with perfect blindness. This immediately implies the security of the proposed protocol with a single honest client and an untrusted server, as the instructions of the protocols exactly coincide. By a similar argument, security can be shown for the two-client and multi-client generalizations. In the spirit of the security proof in [Kap+23], the sequential rotations of a single qubit by all clients along the Qline can be seen as a collaborative version of RSR where the role of the RSR-client is now taken by an entirely classical trusted virtual party, the *orchestrator*. The entire execution of the protocol then becomes one run of the BQC protocol between the orchestrator and the server. Finally, in real-world implementations, the orchestrator can be replaced by a classical SMPC protocol. Security follows by composition of the above-mentioned building blocks which have all been proven to be composably secure in the Abstract Cryptography framework [MR11].

In the following, we provide the details of the security proof for the full multi-client blind quantum computation protocol. The security analysis uses techniques and results from previous work on the Qline architecture [Doo+23], on quantum secure multi-party computing (QSMPC) [Kap+23], and on blind delegated computing using trusted Remote State Rotation (RSR) [Ma+22].

8.3.1 Ideal functionalities

In the AC framework, the desired, perfect behavior of protocols is captured by resources that we call ideal functionalities. We present the ideal functionalities of the resources relevant to the presented security proof in the following.

Resource 8 describes the *Remote State Rotation* functionality, as defined in [Ma+22, Definition 7]. Its purpose is to capture the quantum abilities of the clients which are limited to performing single-qubit rotations around the z-axis.

Resource 1 captures *Blind Delegated Quantum Computing* without verifiability [Dun+14]. In our version of the ideal functionality, we assume the delegated computation to be

Resource 8 Remote State Rotation

Inputs:

- The client sends an angle $\theta \in \mathcal{A} = \{\frac{j\pi}{4} | j = 0, \ldots, 7\}$.
- The server sends a single-qubit quantum state ρ.

Computation by the resource:

1. Set $\rho' = R_z(\theta)\rho(R_z(\theta))^\dagger$ and return ρ' to the server.

restricted to classical inputs. A malicious server (setting $c = 1$) has the option to corrupt the state that the client receives at the end of the protocol.

Note that universal computation is possible with Resource 1 if $\mathcal{U}$ is a universal quantum map, and the description of the target quantum computation is encoded in the client's input x. For honest servers, the filter $\perp_{c=0}$ sets $c = 0$ and blocks access to the other interface on the server's side.

Finally, we need to describe the ideal functionality of our target Resource 9, *Multi-Client Blind Quantum Computation*. As before, the jointly evaluated computations are restricted to classical inputs.

Resource 9 Multi-Client Blind Quantum Computation

Inputs:

- For $j = 1, \ldots, n$, client j sends a classical bit string x_j. It also inputs $c_j \in \{0, 1\}$ as a filtered interface.
- The server inputs a flag $c \in \{0, 1\}$.
- All malicious parties, that is all clients with $c_j = 1$, and the server if $c = 1$, jointly send a quantum state ψ and the description of a map $\mathcal{E}$.

Computation by the resource:

1. If $c_j = c = 0$ for all $j = 1, \ldots, n$, the resource computes the correct output $y = \mathcal{U}\left(\bigotimes_{j=1}^n |x_j\rangle\langle x_j|\right)$ and sends it to the clients.
2. Otherwise, the resource computes the corrupted output $y = \mathcal{E}\left(\bigotimes_{j=1}^n |x_j\rangle\langle x_j| \otimes \psi\right)$ and sends it to the clients.

8.3.2 Security of the full protocol

Universal Blind Quantum Computing [BFK09] is a protocol that allows a client to delegate a computation to a server with the guarantee that the server cannot learn

anything about the computation, its inputs, or its outputs, except for the amount of resources that were used. In other words, only the computation graph and the order of qubits are leaked to the server. Protocol 10 is a version of the UBQC Protocol 2 that does not require universal resource states, but rather works directly on the graph that is used to implement the target computation.

Protocol 10 Blind Quantum Computation

Public Information:

- A graph $G = (V, E, I, O)$ with input and output vertices I and O, respectively.
- A partial order $\preceq$ on the set V of vertices.

Inputs:

- The client has as input a classical bit string $x \in \{0,1\}^{|I|}$. It further inputs a set of angles $(\phi_v)_{v \in V}$, and a flow f on G compatible with $\preceq$.
- The server has no inputs.

Protocol:

1. For all $v \in V$, the client samples an angle $\theta(v) \leftarrow_R \mathcal{A}$ uniformly at random, prepares the single-qubit state $\left|+_{\theta(v)}\right\rangle$, and sends it to the server.
2. The server applies the entangling operation according to the graph G, i.e., for every edge $\{v, w\} \in E$, the server applies the two-qubit gate CZ_{vw}.
3. For all $v \in V \setminus O$, such that the partial order $\preceq$ is respected, the client and the server perform the following interactive steps:
   (a) The client uses the previous (corrected) measurement outcomes to compute the corrected angle $\phi'(v)$ from $\phi(v)$ according to the flow f.
   (b) The clients samples a bit $r(v) \leftarrow_R \{0,1\}$ uniformly at random, calculates $\delta(v) = \phi'(v) + \theta(v) + (r(v) + x(v))\pi$, and sends $\delta(v)$ to the server.
   (c) The server measures qubit v in the $\left|\pm_{\delta(v)}\right\rangle$-basis and returns the measurement outcome $m(v)$ to the client.
   (d) The client calculates the corrected measurement outcome as $m'(v) = m(v) \oplus r(v)$.
4. The client outputs the bit string $(m'(v))_{v \in V \setminus O}$. The server further sends the qubits in O to the client, who decrypts them (using $\theta(v)$, $r(v)$ and $m'(v)$ as keys) and keeps them as additional output.

Previous work showed that Resource 8 can be used in combination with Protocol 10 to securely construct Resource 1. Since we use this result as part of our security proof, we reiterate it here for convenience [Ma+22, Theorem 4].

Theorem 8.3.1. *The BQC Protocol 10 where the client uses Resource 8 to remotely prepare the required single-qubit states perfectly constructs Resource 1 against a malicious server.*

The security analysis of the multi-client protocol proposed in this work follows a modular paradigm. In this spirit, we first analyze the security of Protocol 11, the subprotocol which consists of the communication of a single qubit along the Qline. Since both the photon source and the server are untrusted and potentially colluding in the protocol, they are treated as a single untrusted party in the following.

Protocol 11 Collaborative Remote State Rotation

Inputs:

- The n clients have no input.
- The orchestrator has as input an angle $\theta \in \mathcal{A}$.
- The server receives as input a single-qubit quantum state ρ.

Protocol:

1. For $j = 1, \ldots, n$, client j samples uniformly at random $\theta_j \leftarrow_R \mathcal{A}$, and sends θ_j to the orchestrator.
2. The server sends ρ to client 1.
3. For $j = 1, \ldots, n$, client j applies the operation $R_z(\theta_j)$ to the received quantum state and forwards it to the next client. After applying its own rotation, client n forwards the final state to the server.
4. The orchestrator computes $\theta' = \theta - \sum_{j=1}^{n} \theta_j \pmod{2\pi}$ and sends θ' to the server.
5. The server applies the operation $R_z(\theta')$ to the single-qubit state that it received from client n, and keeps the resulting state as its output.

We continue to prove the security of Protocol 11.

Theorem 8.3.2. *Protocol 11 perfectly constructs Resource 8 between the orchestrator and the server from secure classical and quantum channels against malicious coalitions of at most the server and $n-1$ clients.*

Proof of correctness. If all participating parties are acting honestly, the single-qubit quantum state sent from client n to the server takes the following form:

$$R_z(\bar{\theta})\rho\left(R_z(\bar{\theta})\right)^{\dagger}, \tag{8.5}$$

where $\bar{\theta} = \sum_{j=1}^{n} \theta_j$. After the final correction which is applied to this state by the server, the output becomes

$$R_z(\theta')R_z(\bar{\theta})\rho\left(R_z(\bar{\theta})\right)^{\dagger}\left(R_z(\theta')\right)^{\dagger} = R_z(\theta)\rho\left(R_z(\theta)\right)^{\dagger}, \tag{8.6}$$

which shows that the protocol is correct. □

Proof of soundness. As security in the AC framework is simulation-based, we need to provide the construction of a simulator fit to translate real-world to ideal-world attacks. In the following, we assume the worst case of a colluding malicious coalition of the server and $n-1$ clients. Because the protocol and the ideal resource are sufficiently symmetric in the enumeration of the clients, we can assume without loss of generality that the first client behaves honestly. The construction of the simulator for this scenario is given as Simulator 4.

Simulator 4 Malicious server and clients $2, \ldots, n$

Behavior of the simulator:

1. The simulator expects angles $\theta_j \in \mathcal{A}$ for $j = 2, \ldots, n$ from the malicious clients, and a single-qubit quantum state ρ from the malicious server.
2. The simulator forwards ρ to the ideal functionality described by Resource 8, and receives the state ρ' from it.
3. It samples uniformly at random the angle $\theta_1 \leftarrow_R \mathcal{A}$.
4. It applies the operation $R_z(\theta_1)$ to ρ' and returns the resulting quantum state to the malicious server.
5. Finally, the simulator computes $\theta' = -\sum_{j=1}^{n} \theta_j$ and sends θ' to the malicious server.

It remains to be shown that the views of the distinguisher in the real world where it has access to the inputs θ, ρ and to the views of all malicious parties, and in the ideal world where it has access to the inputs θ, ρ and to the interfaces to the simulator are perfectly equal. These two views can be summarized as follows:

	Real world	**Ideal world**
Input angle	θ	θ
Input state	ρ	ρ
Client 1 output	$R_z(\theta_1)\rho(R_z(\theta_1))^{\dagger}$	$R_z(\theta+\theta_1)\rho(R_z(\theta+\theta_1))^{\dagger}$
Correction	$\theta - \sum_{j=1}^{n} \theta_j$	$-\sum_{j=1}^{n} \theta_j$

Since θ_1 is chosen uniformly at random by the simulator in the ideal world, we can substitute it by $\theta_1 - \theta$ without changing the view of the distinguisher. This yields:

Real world	**Ideal world**
θ	θ
ρ	ρ
$R_z(\theta_1)\rho(R_z(\theta_1))^\dagger$	$R_z(\theta_1)\rho(R_z(\theta_1))^\dagger$
$\theta - \sum_{j=1}^{n} \theta_j$	$\theta - \sum_{j=1}^{n} \theta_j$

Clearly, these two distributions are identical, which proves that the views of the distinguisher in the two worlds are perfectly indistinguishable. □

It now remains to piece together all building blocks to obtain the security of the full protocol.

Theorem 8.3.3. *The Multi-Client Blind Quantum Computation Protocol 1 of the main text, where the orchestrator is replaced by a classical SMPC resource between all other parties, perfectly constructs Resource 9 against malicious coalitions of at most the server and $n-1$ clients.*

Proof. After having established the security of all building blocks, this proof is straightforward and works by repeated application of the general composition principle in Theorem 2.1.4, analogously to the proof of [Kap+23, Theorem 5].

Retracing the steps of the construction of the protocol in question, we begin by replacing the classical SMPC resource by a trusted classical party, the orchestrator. This step does not incur any security loss.

Next, Theorem 8.3.2 allows us to replace every execution of Protocol 11 by one call to Resource 8, with zero security loss.

Finally, Theorem 8.3.1 establishes that the remaining protocol between orchestrator and server is indeed a secure construction of the Blind Delegated Quantum Computing Resource 1. Keeping in mind that the orchestrator additionally is in charge of collecting the clients' inputs and distributing the outcome of the computation, this protocol is indeed a perfect realization of Resource 9. □

Remark 8.3.4 (Removing redundant correction steps)**.** *When considering the realization of a resource by itself, the final correction step in Protocol 11 is indeed necessary, as this is the only way for the simulator to transmit the correct quantum state to the server. However, when using the protocol in the context of UBQC, this correction step can be*

combined with the corrections that are anyway present in the BQC protocol, and do hence not need to be performed as a separate round of communication. This observation is similar to the one made by [Kap+23] in the context of Collaborative RSP and QSMPC.

Remark 8.3.5 (Varying the location of the entangling step)**.** *The security of the protocol is independent of whether the qubits are entangled before or after their communication along the Qline. Therefore, both setups in which the photon source creates cluster states and the server performs only adaptive measurements, as well as setups in which the photon source emits unentangled qubits, and entanglement and measurements are performed by the server are possible. Even combinations of both, in which part of the entanglement is created before, and part after the Qline, are conceivable.*

8.4 Experimental apparatus

In Fig. 8.3 we describe the experimental apparatus we employ to implement the two-client protocol.

Photon source. A Sagnac-based source of polarization-entangled photons, i.e. server S_1, generates pairs of photons in the state defined in Eq. (8.2), where we encode the computational basis vector $|0\rangle$ in the photons' horizontal polarization ($|H\rangle$), and $|1\rangle$ in the vertical one ($|V\rangle$). The photon pairs are sent to the clients who apply their random rotations. The resulting state after these transformations is defined in Eq. (8.3).

Clients' preparation. At each run of the protocol, we set all random parameters through an ID Quantique quantum random number generator (QRNG). Both clients use liquid crystals (LCs) to apply their rotations, which are set in this preparation stage. The TTP is made up of a computer linked to a fast electronic circuit and stores all clients' parameters. With such information, the TTP pre-computes the measurement angle δ_1, and the first measurement station is set accordingly. Moreover, the TTP also pre-computes the two possible values for δ_2, considering that the first outcome is still unknown, namely $\delta_2^{\pm} = \theta_2 + x_2\pi + r_2\pi \pm \phi_2$, to speed up the measurement step of the protocol.

Measurement. The two photons are sent to server S_2 of Fig. 8.3 where measurements of the form $M(\delta) = \cos(\delta)\sigma_x + \sin(\delta)\sigma_y$ are performed. The first one is made up of a quarter-wave plate (QWP), a half-wave plate (HWP), and a polarizing beam splitter (PBS). The two single-photon avalanche photodiodes (APD) of the first measurement

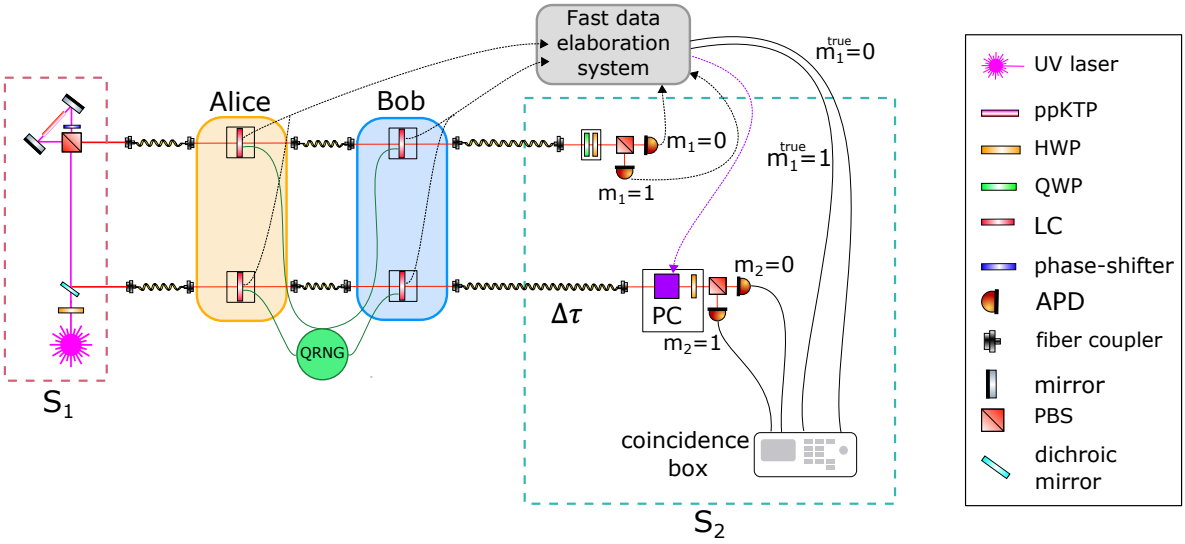


Figure 8.3: Experimental apparatus. The state defined in Eq. (8.2) is generated through a Sagnac-based source of entangled photons, where horizontal polarization ($|H\rangle$) encodes the state $|0\rangle$ while vertical polarization ($|V\rangle$) encodes the state $|1\rangle$. The photons are first sent to Alice and Bob who performs single-qubit rotations by means of liquid crystals (LC). To make the clients' choices random, the clients' secret parameters are chosen by means of a quantum random number generator (QRNG). Then, the two photons are sent to Server 2 (S2). While the second photon is delayed through a ≈ 65 m-long single-mode fiber, the first photon is measured, by using a sequence of a quarter-wave plate (QWP), a half-wave plate (HWP), and a polarizing beam splitter (PBS). The second measurement station, instead, is composed of a Pockels cell (PC), a HWP, and a PBS. A fast data elaboration system, constituted by a computer and a fast electronic circuit, embodies the TTP. The two detectors of the first measurement station are linked to such a system, that directly activates the PC with a suitable high-voltage (HV) pulse according to the desired second measurement basis. All outcomes are collected through a coincidence box that records as two-fold coincidences all events occurring in a given time window.

station are connected to a fast electronic circuit that selects δ_2^+ or δ_2^-, according to the corrected outcome of the first measurement, $m_1^{true} = m_1 \oplus r_1$. In the second measurement station, we substitute the QWP with a Pockels cell (PC), i.e. a fast electro-optical modulator that performs the identity when no voltage is applied, while applying a phase shift between orthogonal polarizations when a voltage is applied. The second photon is delayed with respect to the first one by using a ≈ 65 m single-mode fiber to enable feed-forward in the second measurement station. Finally, the second outcome is corrected according to $m_2^{true} = m_2 \oplus r_2$. All events are collected through a coincidence box that records as two-fold coincidences all detector clicks occurring in a given time window.

8.5 Results

Blindness of the protocol. To show that the server cannot gain any information about the outcome of the computation, we suppose that the clients want to compute a given quantum function whose outcome is represented by the second qubit. We repeated the experiment for both qubits of the cluster state, but we show in the main text only the resulting density matrix for the second qubit. We demonstrate blindness of the second qubit by keeping the measurement angle $\delta_1 = \pi$ fixed and by averaging over all density matrices resulting in the output qubit for different initial rotation angles θ_1^j, where $j = A, B$, namely 64 combinations. The density matrix in Fig. 8.4a shows the resulting quantum state, which has a fidelity with a single-qubit completely mixed state amounting to $F_2 = 0.99870 \pm 0.00003$, while the measured Von Neumann entropy is $S_2 = 0.9963 \pm 0.0001$, to be compared with the expected value of 1 for a completely mixed single-qubit state. Furthermore, we demonstrate the blindness of the whole initial two-qubit cluster state, by averaging over all density matrices corresponding to 64 combinations of the initial z-rotations, for parameters θ_1^A and θ_2^B, while keeping θ_1^B, $\theta_2^A = 0$. We stress that these combinations are enough to demonstrate blindness, as the random rotations on each qubit still take all possible values in the set $\mathcal{A}$. For the two-qubit state, whose density matrix is shown in Fig. 8.4b, we estimated a fidelity $F = 0.99433 \pm 0.00003$ with the completely mixed state and with a Von Neumann entropy of $S = 1.9836 \pm 0.0001$, to be compared with the expected value of 2 for a completely mixed two-qubit state. All density matrices are retrieved from raw experimental data through quantum state tomography [NC00].

Correctness of the protocol. Let us now consider the scenario where the clients want to compute a quantum function. In this case, we take the second qubit as the outcome of the computation, by preparing it in the state $|+\rangle$. We perform the computation $\phi_1 = \pi/4$, with input bits $x_1,\ x_2$ set to 0. We set the clients' secret parameters as $\theta_1^A = \pi/2$, $\theta_1^B = \pi/4$ and $r_1^A = r_1^B = 0$. We show our results in Fig. 8.5. The estimated fidelity with the ideal state amounts to $F_{\pi/4} = 0.972 \pm 0.003$. The algorithm performed over input data $x_1,\ x_2$ is characterized by the two true measurement angles $\phi_1,\ \phi_2$. Choosing x_1 or x_2 equal to 1 has simply the effect of inverting the minima and the maxima of the distributions. In Fig. 8.6, we show ten different probability distributions obtained by trying ten different combinations of the algorithm parameters $(\phi_1,\ \phi_2,\ x_1,\ x_2)$, and the comparison with the ideal and the noisy model case. All the obtained results are in good agreement with our noisy model and follow qualitatively the ideal expectations. Small deviations from the expected values are mainly due to the visibility of the quantum state at the end of the Qline and to imperfections coming from the non-ideal electro-optical modulators employed, i.e. the LCs and the PC.

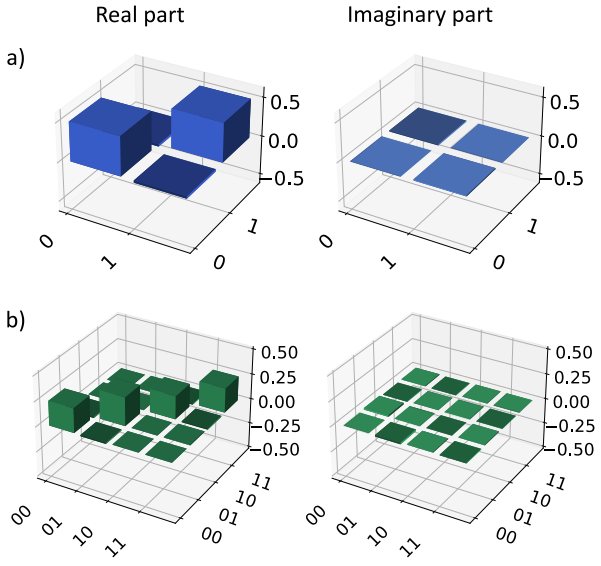


Figure 8.4: Demonstration of blindness. a) Density matrix of the second qubit averaged over all possible θ_1^A and θ_1^B configurations. b) Density matrix of the two-qubit initial state, averaged over all possible values of θ_1^A and θ_2^B.

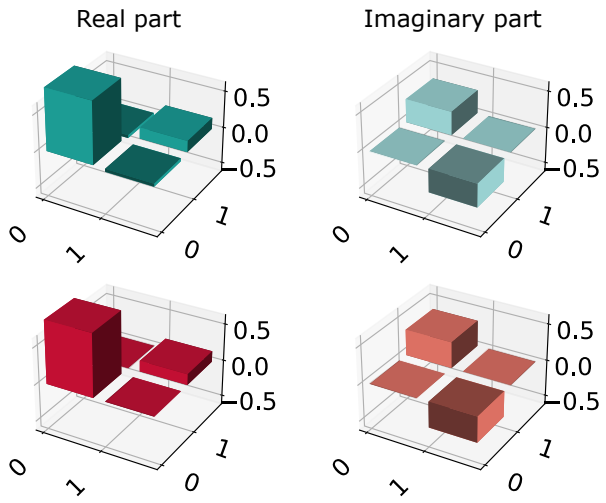


Figure 8.5: Computation of a quantum function. In this figure, we show the results of the computation of a quantum function. Bob chose $\phi_1 = \pi/4$, while the clients' random parameters are $\theta_1^A = \pi/2$ and $\theta_1^B = \pi/4$. Both input bits $x_1,\ x_2$ are set to 0. The first qubit is thus measured in the basis $\delta_1 = \pi$. The experimental density matrix is shown in light blue, while the theoretical one is shown in red.

8.6 Discussion

In this work, we proposed a multi-client version of the BQC protocol [BFK09] and experimentally demonstrated it in a two-client setting. We first simplified the protocol described in [Kap+23; KP17] to tailor it to the photonic Qline network introduced in [Doo+23]. To this end, we studied a photonic platform equipped with a source of polarization-entangled photon pairs, an active feed-forward system, and a fiber-based structure to connect the involved parties. In our scheme, the clients only need to apply single-qubit rotations. Within this setup, we computed the outcomes of ten different classical functions, by changing the input data and the algorithm, and compared the results with a noisy model compatible with our experimental conditions. Also, we demonstrated the correctness of the protocol when the function to be computed has a quantum output. Finally, we showed that the server cannot gain any information about the inputs of the clients or the outcome of the computation.

Scalability. Our proof-of-concept demonstration can represent a step forward toward the realization of a scalable and secure quantum cloud access infrastructure with multiple clients. Indeed, in a real-world protocol, the necessary classical communication as part of the SMPC in between the measurements would considerably increase the time latency, in particular, if run over a slow network, such as the internet. Therefore, to make the

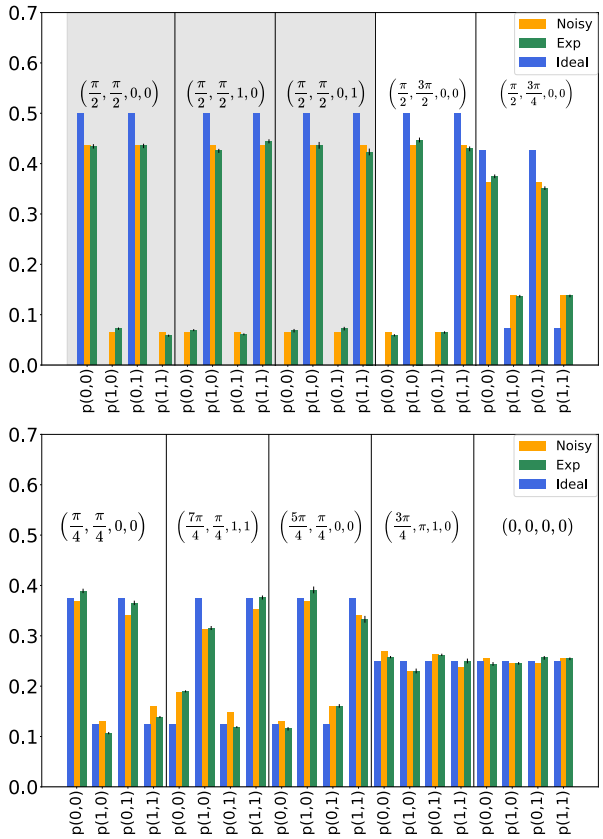


Figure 8.6: Computation of a classical function. In this bar plot, we show ten different measurement angles and Alice's input combinations ($\phi_1,\ \phi_2,\ x_1,\ x_2$). In the grey region, we kept the algorithm fixed while changing the input data, to show the changes in both the expected and experimental distributions. In the white region, instead, we changed both algorithms and input data. The uncertainties on the experimental frequencies were obtained assuming Poissonian statistics, and the black bars correspond to one standard deviation. The eventual absence of black bars means that the uncertainty was too small to be visible in this plot.

experimental realization of the proposed protocol feasible, we replaced the SMPC with a fast electronic data elaboration circuit to both reduce the communication rounds and compensate for the absence of a quantum memory. This allowed us to reduce the delay between measurements to a manageable level. Scaling up the size of the computation would require more qubits and their communication from the photon source past all clients to the server. In principle, this could be realized in two ways that can eventually be combined. First, the qubits could be sent all at once, which would require the clients to be able to apply rotations to multiple states at once. Alternatively, the qubits could be sent sequentially, one at a time. However, this second option would require the ability to store or delay them until the clients were able to adjust their rotation gates since every qubit is rotated by a different angle. In our demonstration, we opted for the first option as it represented the optimal way to minimize time latency and, consequently, photon loss. Moreover, the choice of adopting an optimized feed-forward system for measurement adaptivity in our setup is not only crucial to ensure blind and deterministic computation, but also to scale up the protocol. Indeed, post-selection schemes would require an exponentially growing number of measurements depending on the dimension of the quantum system, which would affect significantly the possibility of applications to larger quantum systems.

Future work. While our implementation guarantees the privacy of the inputs provided by the clients, the outcome of the computation is not verified. We leave the addition of verification to the proposed protocol as future work. One possible path towards verification with Qline architecture might be the employment of the novel *dummyless* testing technique from [Kap+23]. However, as of now, the question whether states prepared by rotation-only clients are sufficient for verification remains open, as [Ma+22] only showed that this kind of states are enough to achieve blindness.

We believe that this work has insightful implications both from a theoretical and an experimental point of view. From a theoretical perspective, it constitutes a strong encouragement toward the development of collaborative computational algorithms over distributed quantum networks, as well as investigations about their verification. From an experimental point of view, instead, it represents a step forward toward the applications of photonic linear quantum networks as building blocks for more complex networks, toward the realization of a large and densely connected quantum cloud.

8.7 Methods

Photon pairs are generated in a parametric down-conversion source, composed of a 25-mm-thick periodically-poled Potassium Titanyl Phosphate (ppKTP) crystal inside a Sagnac interferometer. The source uses a Toptica continuous-wave diode laser with a wavelength equal to 405 nm. Both photons are generated at a wavelength equal to 810 nm. To test the quality of the bipartite state generated by S_1, we perform a CHSH Bell test [Cla+69] and obtain a Bell parameter equal to 2.752 ± 0.006. The generated photons are filtered in wavelength and spatial mode by using, respectively, narrow-band filters and single-mode fibers. The PC is a $LiNbO_3$ crystal made by the Shangai Institute of Ceramics having a rise-time equal to 90 ns. A fast electronic circuit transforms signals coming from the detectors of the first measurement station into high-voltage calibrated pulses, needed to activate the PC. The amount of delay on the second photon was evaluated considering the response time of the detectors, the speed of the signal transmission through a single-mode fiber, whose refraction index is ≈ 1.45, and the activation time of the PC. Therefore, we used a ≈ 65 m long single-mode fiber that allows a delay of ≈ 320 ns of the second photon with respect to the first. The voltages applied to the PC to insert a phase shift equal to $\pi/4,\ \pi/2,\ 3\pi/4$ were, respectively, $V_{\pi/4} = 650$ V, $V_{\pi/2} = 850$ V, $V_{3\pi/4} = 1100$ V. Our experiment is performed shot-by-shot, namely, each event of our data takings is characterized by a different randomly chosen set of initial parameters θ_i^j, r_i^j, for $i = 1,\ 2$ and $j = A,\ B$, while the algorithm ($\phi_i,\ x_i$, for $i = 1,\ 2$) is kept fixed for each data taking.

Chapter 9

Conclusions

We conclude this thesis with a summary of its contributions, and a selection of open questions that are naturally arising from the results presented previously.

9.1 Summary

Our playground for this thesis was the realm of secure delegated quantum computing, a functionality that allows computationally weaker clients to delegate a quantum computation to a more powerful quantum server, all while observing security guarantees such as the blindness of data and algorithm, and the verifiability of the outcomes. During the course of this work, our focus was on designing new protocols with i) improved efficiency, ii) reduced resource and hardware requirements, iii) generalized applicability, and iv) strong security. At the same time, we researched the theoretical foundations of quantum verification, and links to the fields of post-quantum cryptography, error detection and correction, and error mitigation.

Our journey started, in Chapter 3, with an investigation into the role of quantum communication in delegation protocols. While it would be desirable to get rid of quantum communication altogether, we showed that this is not possible in the most general and context-insensitive way while preserving security. However, it turned out that there are concrete cryptographic constructions that can fill in for a quantum channel in the restricted context of blind quantum computing if we are satisfied with the lower level of game-based security only.

Accepting the necessity of single-qubit communication for now, we then shifted our focus to the server and asked, in Chapter 4, whether server-side hardware overheads

were really unavoidable, since those were the main reason that existing blind verification schemes were unfeasible in practice. Fortunately for the security of quantum computations, we found that this was actually not the case and designed a novel protocol for blind BQP verification that managed to entirely avoid any server-side quantum hardware overheads. Additionally, the new protocol required only an efficient number of repetitions to achieve exponential statistical security, and is tolerant to globally bounded noise. This makes it the first practically feasible protocol for full quantum verification.

After finding one specific construction that showed that verification without hardware overhead is possible, we set out to answer the bigger question, in Chapter 5, of how to optimize verification schemes systematically in a more general way. The result was a framework that allowed for the optimization and customization of blind verification protocols, and that could directly be applied to find even more efficient schemes than previously known.

This framework also immediately proved useful in the following Chapter 6 where the need for a verification technique with very particular properties naturally arose. Using the tools from Chapter 5, we were able to construct a dummyless verification scheme, that exactly matched our requirements. As a consequence, we were able to transfer the improvements for delegation protocols previously obtained in the two-party case to the multi-party setting, and give the first quantum secure multi-party computing (QSMPC) scheme feasible already in the near term. This new QSMPC scheme avoids both the superposition evaluation of post-quantum primitives and any kind of fault-tolerant encoding, introduces (almost) no hardware overheads, and even achieves a basic sense of noise robustness.

Finally, we demonstrated in two different experimental scenarios, in Chapters 7 and 8, that our work and optimizations moved secure delegated quantum computing to a regime of current practical feasibility. Both implemented protocols, an implementation of fully verified blind quantum computing with a single client, and blind quantum computing with more than one client, achieve a high level of composable security.

To summarize, the work in this thesis started off in a situation where none of the proposed schemes for secure delegated quantum computing were remotely implementable and managed to move us to the comfortable position of having access to currently realizable protocols, demonstrated by proof-of-concept experimental implementations. During the course of these discoveries, we developed a theory to better understand necessary and unnecessary overheads and assumptions for security, and possible optimizations both in the near-term and in the scalable regimes.

9.2 Future work

Despite the better understanding that we obtained for the security of delegated quantum computing, many bigger and smaller open questions remain. In the following, we give a selection of important and more fundamental problems that remain unsolved, but for which theory and tools from this thesis might prove useful.

Quantum verification without trusted preparations or measurements. The experiment [Pol+23] described in Chapter 8 already realizes multi-client blind quantum computing without trusted preparations or measurements, extending previous work [Ma+22] that achieved the same security guarantees for a single client. This leaves open the natural question of whether it is possible to also achieve verification in this restricted setting. More formally:

> *Is statistical quantum verification possible in a setting where the verifier is restricted to single-qubit operations, and does not have trusted access to state preparations, measurements, or quantum memory?*

Securing quantum computations with noisy verifiers. So far, protocols for the (statistically) secure delegation of quantum computations, including the ones presented in this thesis, needed to assume that the quantum operations performed by the verifier are implemented perfectly. This assumption is often implicit in the cryptographic setting, in which the verifier is honest. However, it does clearly not reflect the reality in which the analogue nature of quantum computing prevents the construction of noiseless devices. Moreover, as the verifier's quantum operations depend on their private coins, we need to realistically expect the noise to be secret-dependent as well.

Hence, one could argue that the problem of quantum verification *under realistic assumptions* still remains untreated. In fact, as the secret-dependency of the noise could lead to potential leakage of information about the verifier's private coins, this question is highly non-trivial. This is reflected in the fact that all known proof techniques and tools to obtain verification break down in the presence of realistic noise. In particular, and to distinguish this open problem from self-testing scenarios, we are interested in the situation where the verifier has access to a small, noisy, but faithfully operated quantum device. Translating this to more formal assumptions, the verifier's quantum operations might suffer from secret-dependent imperfections of arbitrary structure, but of bounded magnitude. We explicitly also wish to capture the inherent uncertainty that the verifier

might have about the structure of the noise on their own device, which is making active mitigation strategies much more intricate. This leads us to the following fundamental question, essential to realistic quantum verification:

> *Is statistical quantum verification possible in a setting in which the verifier has access only to imperfect quantum operations whose noise can be secret-dependent and of arbitrary structure, but only of bounded magnitude?*

Fault-tolerant quantum verification. It has been argued that full fault-tolerance is necessary for the true scalability of quantum computations, since otherwise present noise would cause entropy to accumulate in the outcome, rendering it useless. While multiple blind verification schemes in the prepare-and-send model have been proposed, none of them achieves full quantum fault-tolerance, a prerequisite for *useful* scalable verification. Previous work [Aha+17] discussed the problems with trying to construct a secure and fault-tolerant verification protocol, and has raised the question of whether such a construction is possible at all.

> *Is fully fault-tolerant statistically secure blind verified quantum computing possible?*

The first of the two main obstacles in the design of such a protocol is the verifier-instructed logical encoding of states that must remain hidden from the server, in a scenario in which the verifier is restricted to single-qubit operations, or more generally constant-size quantum operations, on the physical level. Secondly, even when allowing the verifier to operate on a logarithmic number of physical qubits which would allow them to perform the logical encoding locally, the security problems caused by noise in the verifier's setup remain. This intimately links the problem of fault-tolerant verification to the previous open question involving noisy verifiers.

There is hope that this problem could be solved, at least in the special case of stochastic verifier-side noise, by employing techniques similar to the Collaborative Remote State Preparation protocol presented in Chapter 6. Instead of leakage caused by some of the clients involved in the protocol maliciously colluding with the server, in the noisy case leakage results probabilistically from imperfections of the verifier's device. There is hope a similar mitigation technique could be applied here as well, yielding a distillation procedure consuming multiple copies of blind, but possibly leaky states and returning a blind state with the leakage removed. This would also immediately make the

dummyless verification technique [Kap+23], presented in Chapter 6, a natural candidate for the core verification mechanism of a possibly fully fault-tolerant realization.

Verification-inspired benchmarking. Benchmarking protocols aim at capturing the quality and power of quantum devices. Given the emerging availability of multiple commercial, competing quantum computers, the question of quantum benchmarking experiences growing interest. Most proposals focus either on specific algorithms believed to be representative of larger classes of quantum algorithms of interest, or rely entirely on heuristic metrics that are attempting to capture the quality of a device. Unfortunately, these kinds of benchmarking protocols do not allow making more formal and provable statements about the investigated device's properties. We, therefore, ask:

> *How to design a quantum benchmarking protocol that gives provable guarantees about a tested device's computational power?*

As observed in previous work [Dun+14] and in work presented in this thesis [Kap+22] (see Chapter 5) composably secure verification schemes admit a property called *independent verifiability* which guarantees the independence of the verification mechanisms from the target computation. Protocols observing this level of security, therefore, accept or reject entirely independently of the target computation that was delegated. Hence, the successful execution of such a protocol yields more information than the ability of the used quantum device to perform one specific computation. It lets us conclude that the device would have been able to perform equally well for any computation (from some class) that could have been swapped for it. Given the efficiency of state-of-the-art composable verification schemes, it would be extremely interesting to further explore this approach for the construction of rigorous benchmarking techniques.

Publications

[Bad+20] C. Badertscher, A. Cojocaru, L. Colisson, E. Kashefi, D. Leichtle, A. Mantri, and P. Wallden. "Security limitations of classical-client delegated quantum computing". In: *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2020)*. Springer. 2020, pp. 667–696 (cit. on pp. 4, 6, 22, 214).

[Drm+23b] P. Drmota, D. Nadlinger, D. Main, B. Nichol, E. Ainley, D. Leichtle, A. Mantri, E. Kashefi, R. Srinivas, G. Araneda, C. Ballance, and D. Lucas. *Verifiable blind quantum computing with trapped ions and single photons*. Accepted for publication in Physical Review Letters (PRL). 2023. arXiv: 2305.02936 [quant-ph] (cit. on pp. 6, 7, 213, 230).

[Kap+22] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, and H. Ollivier. *Unifying Quantum Verification and Error-Detection: Theory and Tools for Optimisations*. 2022. arXiv: 2206.00631 [quant-ph] (cit. on pp. 5, 6, 103, 166, 168, 169, 171, 172, 175, 176, 183, 185, 205, 230, 235, 255).

[Kap+23] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, and H. Ollivier. *Asymmetric Quantum Secure Multi-Party Computation With Weak Clients Against Dishonest Majority*. 2023. arXiv: 2303.08865 [quant-ph] (cit. on pp. 5, 7, 165, 230, 231, 237, 242, 243, 247, 249, 255).

[Kas+21] E. Kashefi, D. Leichtle, L. Music, and H. Ollivier. *Securing Quantum Computations in the NISQ Era*. 2021. arXiv: 2011.10005 [quant-ph] (cit. on pp. 5, 6, 71).

[Lei+21] D. Leichtle, L. Music, E. Kashefi, and H. Ollivier. "Verifying BQP Computations on Noisy Devices with Minimal Overhead". In: *PRX Quantum* 2 (4 Oct. 2021), p. 040302. DOI: 10.1103/PRXQuantum.2.040302 (cit. on

pp. 5, 6, 71, 106, 109, 118, 143, 150, 156, 157, 159, 209, 215, 216, 218, 230).

[Pol+23] B. Polacchi, D. Leichtle, L. Limongi, G. Carvacho, G. Milani, N. Spagnolo, M. Kaplan, F. Sciarrino, and E. Kashefi. "Multi-client distributed blind quantum computation with the Qline architecture". In: *Nature Communications* 14 (Nov. 2023). DOI: 10.1038/s41467-023-43617-0 (cit. on pp. 6, 7, 229, 253).

Bibliography

[Aar07] S. Aaronson. *The Scott Aaronson 25.00$ Prize*. http://www.scottaaronson.com/blog/?p=284. Accessed: Jan. 30 2015. Oct. 2007 (cit. on p. 104).

[Aar+19] S. Aaronson, A. Cojocaru, A. Gheorghiu, and E. Kashefi. "Complexity-Theoretic Limitations on Blind Delegated Quantum Computation". In: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. 2019 (cit. on pp. 28, 230).

[AFK87] M. Abadi, J. Feigenbaum, and J. Kilian. "On hiding information from an oracle". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM. 1987, pp. 195–203 (cit. on p. 28).

[ABE10] D. Aharonov, M. Ben-Or, and E. Eban. "Interactive Proofs for Quantum Computations". In: *Proceedings of Innovations in Computer Science 2010*. ICS2010. 2010, pp. 453– (cit. on pp. 22, 105).

[Aha+17] D. Aharonov, M. Ben-Or, E. Eban, and U. Mahadev. *Interactive Proofs for Quantum Computations*. arXiv:1704.04487, updated and corrected version of arxiv:0810.5375. 2017. URL: https://arxiv.org/abs/1704.04487 (cit. on pp. 3, 105, 216, 230, 254).

[ALP20] J. Alcazar, V. Leyton-Ortega, and A. Perdomo-Ortiz. "Classical versus quantum models in machine learning: insights from a finance application". In: *Mach. Learn.: Sci. Technol.* 1.3 (2020), p. 035003 (cit. on p. 214).

[Alo+20] B. Alon, H. Chung, K.-M. Chung, M.-Y. Huang, Y. Lee, and Y.-C. Shen. *Round Efficient Secure Multiparty Quantum Computation with Identifiable Abort*. Nov. 2020. URL: https://eprint.iacr.org/2020.1464 (cit. on pp. 199–202).

[Ama+22] D. Amaro, C. Modica, M. Rosenkranz, M. Fiorentini, M. Benedetti, and M. Lubasch. "Filtering variational quantum algorithms for combinatorial optimization". In: *Quantum Sci. Technol.* 7.1 (Feb. 2022), p. 015021. DOI: 10.1088/2058-9565/ac3e54 (cit. on p. 214).

[Aru+19] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510 (cit. on p. 230).

[BPW03] M. Backes, B. Pfitzmann, and M. Waidner. "A composable cryptographic library with nested operations". In: *Proceedings of the 10th ACM conference on Computer and communications security*. ACM. 2003, pp. 220–230 (cit. on p. 10).

[Bao+23] J. Bao, Z. Fu, T. Pramanik, J. Mao, Y. Chi, Y. Cao, C. Zhai, Y. Mao, T. Dai, X. Chen, et al. "Very-large-scale integrated quantum graph photonics". In: *Nature Photonics* (2023), pp. 1–9 (cit. on p. 230).

[Bar21] J. Bartusek. "Secure Quantum Computation with Classical Communication". In: *Theory of Cryptography*. Ed. by K. Nissim and B. Waters. Cham: Springer International Publishing, 2021, pp. 1–30. ISBN: 978-3-030-90459-3 (cit. on p. 168).

[Bar+21] J. Bartusek, A. Coladangelo, D. Khurana, and F. Ma. "On the Round Complexity of Secure Quantum Computation". In: *Advances in Cryptology – CRYPTO 2021*. Ed. by T. Malkin and C. Peikert. Cham: Springer International Publishing, 2021, pp. 406–435. ISBN: 978-3-030-84242-0 (cit. on p. 167).

[Bar+13] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther. "Experimental verification of quantum computation". In: *Nature Physics* 9.11 (Nov. 2013), pp. 727–731. ISSN: 1745-2481. DOI: 10.1038/nphys2763 (cit. on pp. 72, 215, 224, 230).

[Bar+12] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther. "Demonstration of Blind Quantum Computing". In: *Science* 335.6066 (2012), pp. 303–308. ISSN: 0036-8075. eprint: https://science.sciencemag.org/content/335/6066/303.full.pdf. DOI: 10.1126/science.1214707 (cit. on pp. 72, 215, 224, 230).

[Ben+06] M. Ben-Or, C. Crepeau, D. Gottesman, A. Hassidim, and A. Smith. "Secure Multiparty Quantum Computation with (Only) a Strict Honest Majority". In: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 249–260. ISBN: 0-7695-2720-5. DOI: 10.1109/FOCS.2006.68 (cit. on p. 167).

[Ben+01] C. H. Bennett, D. P. DiVincenzo, P. W. Shor, J. A. Smolin, B. M. Terhal, and W. K. Wootters. "Remote State Preparation". In: *Phys. Rev. Lett.* 87 (7 July 2001), p. 077902. DOI: 10.1103/PhysRevLett.87.077902 (cit. on p. 217).

[Bli+04] B. B. Blinov, D. L. Moehring, L.-M. Duan, and C. Monroe. "Observation of entanglement between a single trapped atom and a single photon". In: *Nature* 428.6979 (2004), pp. 153–157. URL: https://www.nature.com/articles/nature02377 (cit. on p. 218).

[Bra18] Z. Brakerski. "Quantum FHE (Almost) As Secure As Classical". In: *Annual International Cryptology Conference*. Springer. 2018, pp. 67–95 (cit. on p. 28).

[Bro15] A. Broadbent. "Delegating private quantum computations". In: *Canadian Journal of Physics* 93.9 (2015), pp. 941–946 (cit. on p. 22).

[Bro18] A. Broadbent. "How to Verify a Quantum Computation". In: *Theory Comput.* 14.11 (2018), pp. 1–37. DOI: 10.4086/toc.2018.v014a011 (cit. on pp. 3, 22, 105, 216).

[BFK09] A. Broadbent, J. Fitzsimons, and E. Kashefi. "Universal blind quantum computation". In: *FOCS'09. 50th Annual IEEE Symposium on Foundations of Computer Science, 2009*. IEEE. 2009, pp. 517–526. DOI: 10.1109/FOCS.2009.36 (cit. on pp. 3, 14–16, 22, 23, 27, 44, 45, 54, 59, 214, 215, 230, 231, 235, 238, 247).

[BFK10] A. Broadbent, J. Fitzsimons, and E. Kashefi. "Measurement-Based and Universal Blind Quantum Computation". In: *Formal Methods for Quantitative Aspects of Programming Languages: 10th International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2010, Bertinoro, Italy, June 21-26, 2010, Advanced Lectures*. Ed. by A. Aldini, M. Bernardo, A. Di Pierro, and H. Wiklicky.

Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 43–86. ISBN: 978-3-642-13678-8. DOI: 10.1007/978-3-642-13678-8_2 (cit. on pp. 71, 82).

[BJ15] A. Broadbent and S. Jeffery. "Quantum homomorphic encryption for circuits of low T-gate complexity". In: *Annual Cryptology Conference*. Springer. 2015, pp. 609–629 (cit. on p. 22).

[Bru+14] N. Brunner, D. Cavalcanti, S. Pironio, V. Scarani, and S. Wehner. "Bell nonlocality". In: *Reviews of Modern Physics* 86.2 (2014), p. 419 (cit. on p. 53).

[Can01] R. Canetti. "Universally composable security: A new paradigm for cryptographic protocols". In: *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*. IEEE. 2001, pp. 136–145 (cit. on p. 10).

[CY21] S. Y.-C. Chen and S. Yoo. "Federated quantum machine learning". In: *Entropy* 23.4 (2021), p. 460 (cit. on pp. 230, 233).

[Chi05] A. M. Childs. "Secure assisted quantum computation". In: *Quantum Information & Computation* 5.6 (2005), pp. 456–466 (cit. on p. 22).

[CLN05] A. M. Childs, D. W. Leung, and M. A. Nielsen. "Unified derivations of measurement-based schemes for quantum computation". In: *Phys. Rev. A* 71 (3 Mar. 2005), p. 032318. DOI: 10.1103/PhysRevA.71.032318 (cit. on p. 215).

[CDG21] J. Chow, O. Dial, and J. Gambetta. "IBM Quantum breaks the 100-qubit processor barrier". In: *IBM Research Blog* (2021) (cit. on p. 230).

[Cia+20] M. Ciampi, A. Cojocaru, E. Kashefi, and A. Mantri. "Secure two-party quantum computation over classical channels". In: *arXiv preprint arXiv:2010.07925* (2020) (cit. on p. 230).

[Cla+69] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. "Proposed experiment to test local hidden-variable theories". In: *Physical review letters* 23.15 (1969), p. 880 (cit. on p. 250).

[Cle+17] M. Clementi, A. Pappa, A. Eckstein, I. A. Walmsley, E. Kashefi, and S. Barz. "Classical multiparty computation using quantum resources". In: *Physical Review A* 96.6 (2017), p. 062317 (cit. on p. 231).

[Coj+19] A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden. "QFactory: Classically-Instructed Remote Secret Qubits Preparation". In: *Advances in Cryptology – ASIACRYPT 2019*. Ed. by S. D. Galbraith and S. Moriai. Springer International Publishing, 2019, pp. 615–645 (cit. on pp. 22, 27, 28, 42, 54–59).

[Coj+21] A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden. "On the Possibility of Classical Client Blind Quantum Computing". In: *Cryptography* 5.1 (2021). ISSN: 2410-387X. DOI: 10.3390/cryptography5010003 (cit. on pp. 23, 28, 42, 214).

[CDN15] R. Cramer, I. B. Damgrd, and J. B. Nielsen. *Secure Multiparty Computation and Secret Sharing*. 1st. USA: Cambridge University Press, 2015. ISBN: 1107043050. URL: https://dl.acm.org/doi/book/10.5555/2846411 (cit. on p. 166).

[CGS02] C. Crépeau, D. Gottesman, and A. Smith. "Secure Multi-party Quantum Computation". In: *Proceedings of the Thiry-fourth Annual ACM Symp. on Theory of Computing*. STOC '02. Montreal, Quebec, Canada: ACM, 2002, p. 643. ISBN: 1-58113-495-9. DOI: 10.1145/509907.510000 (cit. on p. 167).

[DK06] V. Danos and E. Kashefi. "Determinism in the one-way model". In: *Phys. Rev. A* 74 (5 Nov. 2006), p. 052310. DOI: 10.1103/PhysRevA.74.052310 (cit. on pp. 14, 44).

[DL70] E. B. Davies and J. T. Lewis. "An operational approach to quantum probability". In: *Communications in Mathematical Physics* 17.3 (Sept. 1970), pp. 239–260. DOI: 10.1007/bf01647093 (cit. on p. 12).

[Doo+23] M. Doosti, L. Hanouz, A. Marin, E. Kashefi, and M. Kaplan. "Establishing shared secret keys on quantum line networks: protocol and security". In: *arXiv preprint arXiv:2304.01881* (2023) (cit. on pp. 230, 231, 237, 247).

[Drm+23a] P. Drmota, D. Main, D. P. Nadlinger, B. C. Nichol, M. A. Weber, E. M. Ainley, A. Agrawal, R. Srinivas, G. Araneda, C. J. Ballance, and D. M. Lucas. "Robust Quantum Memory in a Trapped-Ion Quantum Network Node". In: *Phys. Rev. Lett.* 130 (9 Mar. 2023), p. 090803. DOI: 10.1103/PhysRevLett.130.090803 (cit. on pp. 215, 217, 218, 223).

[Dul+20] Y. Dulek, A. B. Grilo, S. Jeffery, C. Majenz, and C. Schaffner. "Secure Multi-party Quantum Computation with a Dishonest Majority". In: *Advances in Cryptology – EUROCRYPT 2020*. Ed. by A. Canteaut and Y. Ishai. Cham: Springer International Publishing, 2020, pp. 729–758. ISBN: 978-3-030-45727-3. DOI: 10.1007/978-3-030-45727-3_25 (cit. on pp. 166, 167, 199–202).

[DSS16] Y. Dulek, C. Schaffner, and F. Speelman. "Quantum homomorphic encryption for polynomial-sized circuits". In: *Annual Cryptology Conference*. Springer. 2016, pp. 3–32 (cit. on p. 22).

[Dun+14] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner. "Composable Security of Delegated Quantum Computation". In: *Advances in Cryptology – ASIACRYPT 2014*. Ed. by P. Sarkar and T. Iwata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 406–425. ISBN: 978-3-662-45608-8 (cit. on pp. 10, 17, 18, 22, 23, 26, 45, 46, 77, 80, 82, 108, 172, 237, 255).

[DK16] V. Dunjko and E. Kashefi. "Blind quantum computing with two almost identical states". In: *arXiv preprint arXiv:1604.01586* (2016) (cit. on pp. 10, 23, 28).

[DKL12] V. Dunjko, E. Kashefi, and A. Leverrier. "Blind quantum computing with weak coherent pulses". In: *Physical Review Letters* 108.20 (2012), p. 200502 (cit. on pp. 22, 23, 230).

[Dup+16] F. Dupuis, S. Fehr, P. Lamontagne, and L. Salvail. "Adaptive Versus Non-Adaptive Strategies in the Quantum Setting with Applications". In: *Advances in Cryptology – CRYPTO 2016*. Ed. by M. Robshaw and J. Katz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 33–59. ISBN: 978-3-662-53015-3 (cit. on p. 167).

[DNS10] F. Dupuis, J. B. Nielsen, and L. Salvail. *Secure Two-Party Quantum Evaluation of Unitaries against Specious Adversaries*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 685–706. ISBN: 978-3-642-14623-7. DOI: 10.1007/978-3-642-14623-7_37 (cit. on p. 167).

[DNS12] F. Dupuis, J. B. Nielsen, and L. Salvail. "Actively secure two-party evaluation of any quantum operation". In: *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 794–811 (cit. on p. 167).

[Feh+13] S. Fehr, J. Katz, F. Song, H.-S. Zhou, and V. Zikas. "Feasibility and Completeness of Cryptographic Tasks in the Quantum World". In: *Theory of Cryptography*. Ed. by A. Sahai. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 281–296. ISBN: 978-3-642-36594-2 (cit. on p. 167).

[Fel91] W. Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, 1991. ISBN: 978-0-471-25709-7. URL: https://www.wiley.com/en-us/An+Introduction+to+Probability+Theory+and+Its+Applications%2C+Volume+2%2C+2nd+Edition-p-9780471257097 (cit. on p. 99).

[FKD18] S. Ferracin, T. Kapourniotis, and A. Datta. "Reducing resources for verification of quantum computations". In: *Physical Review A* 98.2 (2018), p. 022323 (cit. on pp. 106, 129, 169).

[Fis+14] K. A. Fisher, A. Broadbent, L. Shalm, Z. Yan, J. Lavoie, R. Prevedel, T. Jennewein, and K. J. Resch. "Quantum computing on encrypted data". In: *Nature communications* 5.1 (2014), pp. 1–7 (cit. on pp. 215, 216, 224, 230).

[FHM18] J. F. Fitzsimons, M. Hajdušek, and T. Morimae. "Post hoc verification of quantum computation". In: *Physical Review Letters* 120.4 (2018), p. 040501. DOI: 10.1103/PhysRevLett.120.040501 (cit. on pp. 22, 105).

[FK17] J. F. Fitzsimons and E. Kashefi. "Unconditionally verifiable blind quantum computation". In: *Physical Review A* 96.1 (2017), p. 012303. DOI: 10.1103/PhysRevA.96.012303 (cit. on pp. 3, 18, 22, 72, 76, 85, 99, 105, 108, 150, 154, 159, 167, 214, 216).

[Fit17] J. F. Fitzsimons. "Private quantum computation: an introduction to blind quantum computing and related protocols". In: *npj Quantum Information* 3.1 (June 2017), p. 23. DOI: 10.1038/s41534-017-0025-3 (cit. on pp. 22, 215, 232).

[Fok+23] E. N. Fokoua, S. A. Mousavi, G. T. Jasion, D. J. Richardson, and F. Poletti. "Loss in hollow-core optical fibers: mechanisms, scaling rules, and limits". In: *Adv. Opt. Photon.* 15.1 (Mar. 2023), pp. 1–85. DOI: 10.1364/AOP.470592 (cit. on p. 224).

[Gen17] R. Gennaro. "Verifiable outsourced computation: A survey". In: *Proceedings of the ACM Symposium on Principles of Distributed Computing*. 2017, pp. 313–313 (cit. on p. 104).

[Gen09] C. Gentry. "Fully homomorphic encryption using ideal lattices". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 2009, pp. 169–178 (cit. on p. 104).

[GHK18] A. Gheorghiu, M. J. Hoban, and E. Kashefi. "A simple protocol for fault tolerant verification of quantum computation". In: *Quantum Science and Technology* 4.1 (Nov. 2018), p. 015009. DOI: 10.1088/2058-9565/aaeeb3 (cit. on p. 72).

[GKK19] A. Gheorghiu, T. Kapourniotis, and E. Kashefi. "Verification of Quantum Computation: An Overview of Existing Approaches". In: *Theory of Computing Systems* 63.4 (May 2019), pp. 715–808. ISSN: 1433-0490. DOI: 10.1007/s00224-018-9872-3 (cit. on pp. 3, 18, 22, 72, 172, 214, 216).

[GKW15] A. Gheorghiu, E. Kashefi, and P. Wallden. "Robustness and device independence of verifiable blind quantum computing". In: *New Journal of Physics* 17.8 (2015), p. 083040 (cit. on p. 230).

[GV19] A. Gheorghiu and T. Vidick. "Computationally-Secure and Composable Remote State Preparation". In: *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)* (2019), pp. 1024–1033 (cit. on pp. 23, 28, 39, 40).

[GWK17] A. Gheorghiu, P. Wallden, and E. Kashefi. "Rigidity of quantum steering and one-sided device-independent verifiable quantum computation". In: *New Journal of Physics* 19.2 (2017), p. 023043 (cit. on p. 230).

[GRW80] G. C. Ghirardi, A. Rimini, and T. Weber. "A general argument against superluminal transmission through the quantum mechanical measurement process". In: *Lettere al Nuovo Cimento (1971-1985)* 27 (1980), pp. 293–298 (cit. on p. 52).

[Gol01] O. Goldreich. *Foundations of Cryptography*. Cambridge University Press, Aug. 2001. ISBN: 9780511546891. DOI: 10.1017/cbo9780511546891 (cit. on p. 12).

[GW17] E. Greene and J. A. Wellner. "Exponential bounds for the hypergeometric distribution". In: *Bernoulli* 23.3 (Aug. 2017), pp. 1911–1950. ISSN: 1350-7265. DOI: 10.3150/15-bej800 (cit. on p. 100).

[Gre+16] C. Greganti, M.-C. Roehsner, S. Barz, T. Morimae, and P. Walther. "Demonstration of measurement-only blind quantum computing". In: *New Journal of Physics* 18.1 (2016), p. 250. DOI: 10.1088/1367-2630/18/1/013020 (cit. on pp. 72, 215, 224, 230).

[Gri+15] W. P. Grice, P. G. Evans, B. Lawrie, M. Legré, P. Lougovski, W. Ray, B. P. Williams, B. Qi, and A. Smith. "Two-party secret key distribution via a modified quantum secret sharing protocol". In: *Optics Express* 23.6 (2015), pp. 7300–7311 (cit. on p. 231).

[HPF15] M. Hajdušek, C. A. Pérez-Delgado, and J. F. Fitzsimons. "Device-independent verifiable blind quantum computation". In: *arXiv preprint arXiv:1502.02563* (2015) (cit. on p. 230).

[Han+17] D. Hangleiter, M. Kliesch, M. Schwarz, and J. Eisert. "Direct certification of a class of quantum simulations". In: *Quantum Science and Technology* 2.1 (Feb. 2017), p. 015004. DOI: 10.1088/2058-9565/2/1/015004 (cit. on p. 105).

[HM15] M. Hayashi and T. Morimae. "Verifiable measurement-only blind quantum computing with stabilizer testing". In: *Physical Review Letters* 115.22 (2015), p. 220502. DOI: 10.1103/PhysRevLett.115.220502 (cit. on pp. 22, 105, 230).

[HEB03] M. Hein, J. Eisert, and H. J. Briegel. *Multi-party entanglement in graph states*. 2003 (cit. on p. 14).

[HBB99] M. Hillery, V. Bužek, and A. Berthiaume. "Quantum secret sharing". In: *Physical Review A* 59.3 (1999), p. 1829 (cit. on p. 231).

[Hou+18] M. Houshmand, M. Houshmand, S.-H. Tan, and J. Fitzsimons. "Composable secure multi-client delegated quantum computation". In: *arXiv preprint arXiv:1811.11929* (2018) (cit. on p. 168).

[Hua+17] H.-L. Huang, Q. Zhao, X. Ma, C. Liu, Z.-E. Su, X.-L. Wang, L. Li, N.-L. Liu, B. C. Sanders, C.-Y. Lu, et al. "Experimental blind quantum computing for a classical client". In: *Physical review letters* 119.5 (2017), p. 050503 (cit. on p. 230).

[Huc+15] D. Hucul, I. V. Inlek, G. Vittorini, C. Crocker, S. Debnath, S. M. Clark, and C. Monroe. "Modular entanglement of atomic qubits using photons and phonons". In: *Nat. Phys.* 11.1 (2015), pp. 37–42. DOI: 10.1038/nphys3150 (cit. on p. 215).

[Hug+20] A. C. Hughes, V. M. Schäfer, K. Thirumalai, D. P. Nadlinger, S. R. Woodrow, D. M. Lucas, and C. J. Ballance. "Benchmarking a High-Fidelity Mixed-Species Entangling Gate". In: *Phys. Rev. Lett.* 125 (8 Aug. 2020), p. 080504. DOI: 10.1103/PhysRevLett.125.080504 (cit. on p. 223).

[JM17] D. Jost and U. Maurer. "Context-Restricted Indifferentiability: Generalizing UCE and Implications on the Soundness of Hash-Function Constructions". In: *IACR Cryptol. ePrint Arch.* 2017 (2017), p. 461. URL: http://eprint.iacr.org/2017/461 (cit. on pp. 41, 43).

[Kal+17] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson. "Entanglement distillation between solid-state quantum network nodes". In: *Science* 356.6341 (2017), pp. 928–932. DOI: 10.1126/science.aan0070 (cit. on p. 215).

[Kap16] T. Kapourniotis. "Efficient verification of universal and intermediate quantum computing". PhD thesis. University of Edinburgh: School of Informatics, 2016 (cit. on p. 121).

[KD19] T. Kapourniotis and A. Datta. "Nonadaptive fault-tolerant verification of quantum supremacy with noise". In: *Quantum* 3 (July 2019), p. 164. ISSN: 2521-327X. DOI: 10.22331/q-2019-07-12-164 (cit. on pp. 72, 76).

[KDK15] T. Kapourniotis, V. Dunjko, and E. Kashefi. *On optimising quantum communication in verifiable quantum computing*. Presented at AQIS'15 conference. 2015 (cit. on pp. 106, 167).

[Kap+21] T. Kapourniotis, E. Kashefi, L. Music, and H. Ollivier. *Delegating Multi-Party Quantum Computations vs. Dishonest Majority in Two Quantum Rounds*. 2021. arXiv: 2102.12949 [quant-ph] (cit. on pp. 77, 95, 168, 205, 206, 210).

[KMW17] E. Kashefi, L. Music, and P. Wallden. *The Quantum Cut-and-Choose Technique and Quantum Two-Party Computation*. 2017. arXiv: 1703.03754 [quant-ph] (cit. on pp. 22, 167).

[KP17] E. Kashefi and A. Pappa. "Multiparty Delegated Quantum Computing". In: *Cryptography* 1.2 (July 2017), pp. 1–20. ISSN: 2410-387X. DOI: 10.3390/cryptography1020012 (cit. on pp. 22, 166, 168, 230, 231, 235, 247).

[KW17a] E. Kashefi and P. Wallden. "Garbled quantum computation". In: *Cryptography* 1.1 (2017), p. 6 (cit. on pp. 22, 167).

[KW17b] E. Kashefi and P. Wallden. "Optimised resource construction for verifiable quantum computation". In: *Journal of Physics A: Mathematical and Theoretical; preprint arXiv:1510.07408* (2017). URL: http://iopscience.iop.org/10.1088/1751-8121/aa5dac (cit. on pp. 3, 18, 76, 99, 105, 167, 205, 208, 209).

[KRK12] A. Kay, R. Ramanathan, and D. Kaszlikowski. "Optimal Asymmetric Quantum Cloning". In: *arXiv e-prints*, arXiv:1208.5574 (Aug. 2012), arXiv:1208.5574. arXiv: 1208.5574 [quant-ph] (cit. on p. 39).

[Kir+22] J. J. M. Kirsopp, C. Di Paola, D. Z. Manrique, M. Krompiec, G. Greene-Diniz, W. Guba, A. Meyder, D. Wolf, M. Strahm, and D. Muñoz Ramo. "Quantum computational quantification of protein–ligand interactions". In: *Int. J. Quantum Chem.* 122.22 (2022), e26975. DOI: https://doi.org/10.1002/qua.26975 (cit. on p. 214).

[Kon+16] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. "Federated learning: Strategies for improving communication efficiency". In: *arXiv preprint arXiv:1610.05492* (2016) (cit. on p. 230).

[Kön+07] R. König, R. Renner, A. Bariska, and U. Maurer. "Small Accessible Quantum Information Does Not Imply Security". In: *Phys. Rev. Lett.* 98 (14 Apr. 2007), p. 140502. DOI: 10.1103/PhysRevLett.98.140502 (cit. on p. 77).

[Kru+19] V. Krutyanskiy, M. Meraner, J. Schupp, V. Krcmarsky, H. Hainzer, and B. P. Lanyon. "Light-matter entanglement over 50 km of optical fibre". In: *npj Quantum Inf.* 5 (1 2019), p. 72. DOI: 10.1038/s41534-019-0186-3 (cit. on p. 224).

[LLD21] W. Li, S. Lu, and D.-L. Deng. "Quantum federated learning through blind quantum computing". In: *Science China Physics, Mechanics & Astronomy* 64.10 (2021), pp. 1–8 (cit. on p. 230).

[Li+15] Y. Li, P. C. Humphreys, G. J. Mendoza, and S. C. Benjamin. "Resource Costs for Fault-Tolerant Linear Optical Quantum Computing". In: *Phys. Rev. X* 5.4 (2015), p. 041007. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.5.041007 (cit. on p. 215).

[LRW20] V. Lipinska, J. Ribeiro, and S. Wehner. "Secure multi-party quantum computation with few qubits". In: *arXiv e-prints*, arXiv:2004.10486 (Apr. 2020), arXiv:2004.10486. arXiv: 2004.10486 [quant-ph] (cit. on pp. 168, 199–202).

[Liu+23] W.-J. Liu, Z.-X. Li, W.-B. Li, and Q. Yang. "Public verifiable measurement-only blind quantum computation based on entanglement witnesses". In: *Quantum Information Processing* 22.3 (2023), p. 137 (cit. on p. 230).

[Ma+22] Y. Ma, E. Kashefi, M. Arapinis, K. Chakraborty, and M. Kaplan. "QEnclave – A practical solution for secure quantum cloud computing". In: *npj Quantum Information* 8.1 (2022), p. 128 (cit. on pp. 205, 230, 237, 239, 249, 253).

[Mah18a] U. Mahadev. "Classical Homomorphic Encryption for Quantum Circuits". In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. Ed. by M. Thorup. IEEE Computer Society, 2018, pp. 332–338 (cit. on pp. 22, 28).

[Mah18b] U. Mahadev. "Classical Verification of Quantum Computations". In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. Ed. by M. Thorup. IEEE Computer Society, 2018, pp. 259–267. DOI: 10.1109/FOCS.2018.00033 (cit. on pp. 3, 22, 72, 105, 138).

[Mai20] D. Main. "Magnetic Field Stabilisation in Ion Traps". MA thesis. University of Oxford, 2020 (cit. on p. 219).

[Man19] A. Mantri. *Secure delegated quantum computing, PhD thesis*. 2019 (cit. on p. 28).

[Man+17] A. Mantri, T. F. Demarie, N. C. Menicucci, and J. F. Fitzsimons. "Flow ambiguity: A path towards classically driven blind quantum computation". In: *Physical Review X* 7.3 (2017), p. 031004 (cit. on p. 28).

[MDF17] A. Mantri, T. F. Demarie, and J. F. Fitzsimons. "Universality of quantum computation with cluster states and (X, Y)-plane measurements". In: *Sci. Rep.* 7.1 (Feb. 2017), p. 42861. ISSN: 2045-2322. DOI: 10.1038/srep42861 (cit. on pp. 44, 216).

[MPF13] A. Mantri, C. A. Pérez-Delgado, and J. F. Fitzsimons. "Optimal blind quantum computation". In: *Physical review letters* 111.23 (2013), p. 230502 (cit. on p. 230).

[Mar+16] K. Marshall, C. S. Jacobsen, C. Schäfermeier, T. Gehring, C. Weedbrook, and U. L. Andersen. "Continuous-variable quantum computing on encrypted data". In: *Nature communications* 7.1 (2016), p. 13795 (cit. on p. 230).

[Mau11] U. Maurer. "Constructive cryptography–a new paradigm for security definitions and proofs". In: *Theory of Security and Applications*. Springer, 2011, pp. 33–56 (cit. on pp. 10, 23).

[MR11] U. Maurer and R. Renner. "Abstract cryptography". In: *Innovations in Computer Science*. Tsinghua University Press. Jan. 2011, pp. 1–21. ISBN: 978-7-302-24517-9. URL: https://conference.iiis.tsinghua.edu.cn/ICS2011/content/papers/14.html (cit. on pp. 10, 12, 21, 23, 37, 72, 76, 108, 237).

[McC+16] W. McCutcheon, A. Pappa, B. A. Bell, A. McMillan, A. Chailloux, T. Lawson, M. Mafu, D. Markham, E. Diamanti, I. Kerenidis, J. G. Rarity, and M. S. Tame. "Experimental verification of multipartite entanglement in quantum networks". In: *Nature Communications* 7.1 (Nov. 2016), p. 13251. ISSN: 2041-1723. DOI: 10.1038/ncomms13251 (cit. on p. 72).

[McK16] M. McKague. "Interactive Proofs for BQP via Self-Tested Graph States". In: *Theory of Computing* 12.3 (2016), pp. 1–42. DOI: 10.4086/toc.2016.v012a003 (cit. on p. 150).

[MP12] D. Micciancio and C. Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *Lecture Notes in Computer Science* (2012), pp. 700–718. ISSN: 1611-3349. DOI: 10.1007/978-3-642-29011-4_41 (cit. on p. 42).

[Mor14] T. Morimae. "Verification for measurement-only blind quantum computing". In: *Physical Review A* 89.6 (2014), p. 060302 (cit. on p. 230).

[MF12] T. Morimae and K. Fujii. "Blind topological measurement-based quantum computation". In: *Nature communications* 3.1 (2012), p. 1036 (cit. on pp. 22, 230).

[MF13a] T. Morimae and K. Fujii. "Blind quantum computation protocol in which Alice only makes measurements". In: *Phys. Rev. A* 87 (5 May 2013), 050301(R). DOI: 10.1103/PhysRevA.87.050301 (cit. on pp. 215, 230).

[MF13b] T. Morimae and K. Fujii. "Secure Entanglement Distillation for Double-Server Blind Quantum Computation". In: *Phys. Rev. Lett.* 111 (2 July 2013), p. 020502. DOI: 10.1103/PhysRevLett.111.020502 (cit. on p. 72).

[MK13] T. Morimae and T. Koshiba. "Composable security of measuring-Alice blind quantum computation". In: *arXiv preprint arXiv:1306.2113* (2013) (cit. on p. 23).

[MK14] T. Morimae and T. Koshiba. "Impossibility of perfectly-secure delegated quantum computing for classical client". In: *arXiv preprint arXiv:1407.1636* (2014) (cit. on p. 28).

[Nad+21] D. P. Nadlinger, P. Drmota, D. Main, B. C. Nichol, G. Araneda, R. Srinivas, L. J. Stephenson, C. J. Ballance, and D. M. Lucas. *Micromotion minimisation by synchronous detection of parametrically excited motion*. arXiv:2107.00056. 2021. URL: https://arxiv.org/abs/2107.00056 (cit. on p. 219).

[Nie06] M. A. Nielsen. "Cluster-state quantum computation". In: *Reports on Mathematical Physics* 57.1 (2006), pp. 147–161. ISSN: 0034-4877. DOI: https://doi.org/10.1016/S0034-4877(06)80014-5 (cit. on pp. 45, 215).

[NC00] M. A. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000 (cit. on pp. 9, 245).

[Paw+09] M. Pawłowski, T. Paterek, D. Kaszlikowski, V. Scarani, A. Winter, and M. Żukowski. "Information causality as a physical principle". In: *Nature* 461.7267 (2009), pp. 1101–1104 (cit. on p. 53).

[PF15] C. A. Pérez-Delgado and J. F. Fitzsimons. "Iterated gate teleportation and blind quantum computation". In: *Physical review letters* 114.22 (2015), p. 220502 (cit. on p. 230).

[Pfa+14] W. Pfaff, B. J. Hensen, H. Bernien, S. B. van Dam, M. S. Blok, T. H. Taminiau, M. J. Tiggelman, R. N. Schouten, M. Markham, D. J. Twitchen, and R. Hanson. "Unconditional quantum teleportation between distant solid-state quantum bits". In: *Science* 345.6196 (2014), pp. 532–535. DOI: 10.1126/science.1253512 (cit. on p. 215).

[PR14] C. Portmann and R. Renner. "Cryptographic security of quantum key distribution". In: *arXiv e-prints*, arXiv:1409.3525 (Sept. 2014), arXiv:1409.3525. arXiv: 1409.3525 [quant-ph] (cit. on p. 77).

[Pro+22] T. Proctor, K. Rudinger, K. Young, E. Nielsen, and R. Blume-Kohout. "Measuring the capabilities of quantum computers". In: *Nat. Phys.* 18.1 (2022), pp. 75–79. ISSN: 1745-2481. DOI: 10.1038/s41567-021-01409-7 (cit. on p. 214).

[QW21] G.-J. Qu and M.-M. Wang. "Secure Multi-Party Quantum Computation Based on Blind Quantum Computation". In: *International Journal of Theoretical Physics* 60 (2021), pp. 3003–3012 (cit. on p. 230).

[RJK07] R. Raussendorf, J. Harrington, and K. Goyal. "Topological fault-tolerance in cluster state quantum computation". In: *New Journal of Physics* (2007) (cit. on p. 118).

[Rai99] E. M. Rains. "Nonbinary quantum codes". In: *IEEE Transactions on Information Theory* 45.6 (1999), pp. 1827–1832. DOI: 10.1109/18.782103 (cit. on p. 199).

[Rau09] R. Raussendorf. "Measurement-based quantum computation with cluster states". In: *International Journal of Quantum Information* 7.06 (2009), pp. 1053–1203 (cit. on p. 230).

[RB01] R. Raussendorf and H. J. Briegel. "A One-Way Quantum Computer". In: *Phys. Rev. Lett.* 86 (22 May 2001), pp. 5188–5191. DOI: 10.1103/PhysRevLett.86.5188 (cit. on pp. 13, 45, 105, 112, 215, 230, 231, 235).

[Reg09] O. Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), p. 34 (cit. on p. 61).

[Řeh+07] J. Řeháček, Z. Hradil, E. Knill, and A. I. Lvovsky. "Diluted maximum-likelihood algorithm for quantum tomography". In: *Phys. Rev. A* 75 (4 Apr. 2007), p. 042108. DOI: 10.1103/PhysRevA.75.042108.

[RUV12] B. W. Reichardt, F. Unger, and U. Vazirani. "A classical leash for a quantum system: Command of quantum systems via rigidity of CHSH games". In: *arXiv preprint arXiv:1209.0448* (2012) (cit. on pp. 22, 230).

[Sar+19] A. Sarma, R. Chatterjee, K. Gili, and T. Yu. "Quantum unsupervised and supervised learning on superconducting processors". In: *Quantum Inf. Comput.* 20 (2019), pp. 541–552 (cit. on p. 214).

[Sch+05] C. Schmid, P. Trojek, M. Bourennane, C. Kurtsiefer, M. Żukowski, and H. Weinfurter. "Experimental single qubit quantum secret sharing". In: *Physical review letters* 95.23 (2005), p. 230505 (cit. on p. 231).

[Ser74] R. J. Serfling. "Probability Inequalities for the Sum in Sampling without Replacement". In: *Ann. Statist.* 2.1 (Jan. 1974), pp. 39–48. DOI: 10.1214/aos/1176342611 (cit. on p. 100).

[SCY21] R.-T. Shan, X. Chen, and K.-G. Yuan. "Multi-party blind quantum computation protocol with mutual authentication in network". In: *Science China Information Sciences* 64 (2021), pp. 1–14 (cit. on p. 230).

[SZ18] Y.-B. Sheng and L. Zhou. "Blind quantum computation with a noise channel". In: *Physical Review A* 98.5 (2018), p. 052343 (cit. on p. 230).

[Sho99] P. W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332 (cit. on p. 2).

[SCS12] B. N. Simon, C. M. Chandrashekar, and S. Simon. "Hamilton's turns as a visual tool kit for designing single-qubit unitary gates". In: *Phys. Rev. A* 85.2 (Feb. 2012), p. 022323. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.85.022323.

[Sta+22] P.-J. Stas, Y. Q. Huan, B. Machielse, E. N. Knall, A. Suleymanzade, B. Pingault, M. Sutula, S. W. Ding, C. M. Knaut, D. R. Assumpcao, Y.-C. Wei, M. K. Bhaskar, R. Riedinger, D. D. Sukachev, H. Park, M. Lončar, D. S. Levonian, and M. D. Lukin. "Robust multi-qubit quantum network node with integrated error detection". In: *Science* 378.6619 (2022), pp. 557–560. DOI: 10.1126/science.add9771 (cit. on p. 215).

[Ste+20] L. J. Stephenson, D. P. Nadlinger, B. C. Nichol, S. An, P. Drmota, T. G. Ballance, K. Thirumalai, J. F. Goodwin, D. M. Lucas, and C. J. Ballance. "High-Rate, High-Fidelity Entanglement of Qubits Across an Elementary Quantum Network". In: *Phys. Rev. Lett.* 124 (11 Mar. 2020), p. 110501. DOI: 10.1103/PhysRevLett.124.110501 (cit. on p. 218).

[SKM13] T. Sueki, T. Koshiba, and T. Morimae. "Ancilla-driven universal blind quantum computation". In: *Physical Review A* 87.6 (2013), p. 060301 (cit. on p. 230).

[Tak+16] Y. Takeuchi, K. Fujii, R. Ikuta, T. Yamamoto, and N. Imoto. "Blind quantum computation over a collective-noise channel". In: *Physical Review A* 93.5 (2016), p. 052307 (cit. on p. 230).

[Tak+18] Y. Takeuchi, A. Mantri, T. Morimae, A. Mizutani, and J. F. Fitzsimons. "Resource-efficient verification of quantum computing using Serfling's bound". In: *arXiv preprint arXiv:1806.09138* (2018) (cit. on p. 22).

[Tom19] T.-M. Tomescu. "Qubit Encryption by Rotation of Polarization States". MA thesis. University of Oxford, 2019.

[Unr10] D. Unruh. "Universally Composable Quantum Multi-party Computation". In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by H. Gilbert. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 486–505. ISBN: 978-3-642-13190-5 (cit. on pp. 10, 13, 167, 170).

[Vaz07] U. Vazirani. *Workshop on the computational worldview and the sciences*. 2007. URL: http://users.cms.caltech.edu/~schulman/Workshops/CS-Lens-2/report-comp-worldview.pdf (cit. on p. 104).

[Vid20] T. Vidick. "Verifying quantum computations at scale: A cryptographic leash on quantum devices". In: *Bulletin of the American Mathematical Society* 57.1 (2020), pp. 39–76 (cit. on p. 22).

[WEP22] F. Wiesner, J. Eisert, and A. Pappa. *Equivalence in delegated quantum computing*. 2022. DOI: 10.48550/ARXIV.2206.07469 (cit. on pp. 106, 138, 159).

[Wri+19] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. "Benchmarking an 11-qubit quantum computer". In: *Nat. Commun.* 10.1 (Nov. 2019), p. 5464. DOI: 10.1038/s41467-019-13534-2 (cit. on p. 223).

[XTH20] Q. Xu, X. Tan, and R. Huang. "Improved Resource State for Verifiable Blind Quantum Computation". In: *Entropy* 22.9 (2020). ISSN: 1099-4300. DOI: 10.3390/e22090996 (cit. on pp. 3, 18, 105).

[Yan+19] Q. Yang, Y. Liu, T. Chen, and Y. Tong. "Federated machine learning: Concept and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19 (cit. on p. 230).

[Yao86] A. Yao. "How to generate and exchange secrets". In: *Foundations of Computer Science, 1986., 27th Annual Symposium on*. IEEE. 1986, pp. 162–167 (cit. on p. 166).

[ZN42] J. G. Ziegler and N. B. Nichols. "Optimum Settings for Automatic Controllers". In: *Trans. Am. Soc. Mech. Eng.* 64 (8 1942), pp. 759–765.