THE UNIVERSITY
of EDINBURGH

Efficient Verification of Universal and Intermediate Quantum Computing

Theodoros Kapourniotis

Doctor of Philosophy

Laboratory for Foundations of Computer Science

School of Informatics

University of Edinburgh

2016

Abstract

The promise of scalable quantum technology appears more realistic, after recent advances in both theory and experiment. Assuming a quantum computer is developed, the task of verifying the correctness of its outcome becomes crucial. Unfortunately, for a system that involves many particles, predicting its evolution via classical simulation becomes intractable. Moreover, verification of the outcome by computational methods, i.e. involving a classical witness, is believed inefficient for the hardest problems solvable by a quantum computer. A feasible alternative to verify quantum computation is via cryptographic methods, where an untrusted prover has to convince a weak verifier for the correctness of his outcome. This is the approach we take in this thesis.

In the most standard configuration the prover is capable of computing all polynomial-time quantum circuits and the verifier is restricted to classical with very modest quantum power. The goal of existing verification protocols is to reduce the quantum requirements for the verifier - ideally making it purely classical - and reduce the communication complexity. In Part II we propose a composition of two existing verification protocols [Fitzsimons and Kashefi, 2012], [Aharonov et al., 2010] that achieves quadratic improvement in communication complexity, while keeping the quantum requirements for the verifier modest. Along this result, several new techniques are proposed, including the generalization of [Fitzsimons and Kashefi, 2012] to prime dimensions.

In Part III we discuss the idea of model-specific quantum verification, where the prover is restricted to intermediate quantum power, i.e. between full-fledged quantum and purely classical, thus more feasible experimentally. As a proof of principle we propose a verification protocol for the One-Pure-Qubit computer [Knill and Laflamme, 1998], which tolerates noise and is capable of computing hard problems such as large matrix trace estimation. The verification protocol is an adaptation of [Fitzsimons and Kashefi, 2012] running on Measurement-Based Quantum Computing with newly proved properties of the underlying resources.

Connections of quantum verification to other security primitives are considered in Part IV. Authenticated quantum communication has been already proved to relate to quantum verification. We expand this by proposing a quantum authentication protocol derived from [Fitzsimons and Kashefi, 2012] and discuss implications to verification with purely classical verifier.

Connections between quantum security primitives, namely blindness - prover does not learn the computation -, and classical security are considered in Part V. We introduce a protocol where a client with restricted classical resources computes blindly a

universal classical gate with the help of an untrusted server, by adding modest quantum capabilities to both client and server. This example of quantum-enhanced classical security we prove to be a task classically impossible.

Lay Summary

'Quantum computers' are an emerging technology aiming to extend the capabilities of the current 'classical' computers. Instead of 'bits', which can be in the classical state 0 or 1, quantum computers manipulate 'qubits', quantum states that can be in 0 and 1 at the same time (called a 'superposition' of the two states). Only when we measure the output of a quantum computer the state takes a definite classical value 0 or 1. In general, a quantum computer consisting of N qubits will be internally in a superposition of all the possible combinations of values for the N bits: there is a vast number of such configurations, growing exponentially with N. This explains why it is difficult to mimic (or 'simulate') the behaviour of a quantum computer with a classical computer.

We are concerned with the question that follows: If one cannot use a classical computer or a small quantum computer to predict the behaviour of a large quantum computer, how he will ever be sure that the large quantum computer is producing the correct answers. For some problems, verifying the correctness of the outcome is easy. For example, a task that a quantum computer can solve, as opposed to a classical computer, is factoring big numbers. When we get the factors of a number as the output of a quantum computer, we can check their correctness by simply multiplying them together. Unfortunately, it is not always easy to verify the result of a complicated quantum computation. A known technique to build a 'verification protocol', where a device called the 'verifier' can verify the correct operation of a device called the 'prover', is by secretly interspersing small questions in the normal computation we send to the quantum computer. These questions produce an easy to predict result and therefore are used as traps to check the honesty of the quantum computer.

Our first contribution is to reduce the requirements for the verifier (the amount of communication between her and the prover and the size of her quantum device), by combining two existing protocols in a constructive composition that improves over both and produces a series of side results. The second contribution is to propose a verification protocol for a version of the quantum computer, called the one-pure-qubit quantum computer, believed to be easier to build because it tolerates a significant amount of noise. Inspired by the verification protocol, we propose a protocol for the 'authentication' of a quantum state when transmitted over an untrusted channel. Finally, inspired by the techniques of hiding information inside a computation, we propose a protocol to securely delegate a classical computation in a scenario that we prove would be impossible classically, but becomes possible using a small quantum enhancement.

Acknowledgements

This work has been accomplished under the guidance and the continuous support of my main supervisor Elham Kashefi. I am deeply thankful to her for giving me the opportunities, devoting her time to share her wisdom and making me feel part of a wider community. Most of all, she has been an invaluable mentor throughout all the good and the hard times of these years.

I would like to thank the School of Informatics for providing the means and its continuous support in order to produce my work. This research was supported by the Mary and Armeane Choksi Postgraduate Scholarship, EPSRC and the School of Informatics Graduate School. Summer school and conference support was provided by SICSA and the LFCS Student Travel Fund.

My deepest gratitude goes to my co-supervisors, Vedran Dunjko who had the patience to advise and correct my mistakes from the earliest stages and to Petros Wallden for his guidance and crucial suggestions. To all my supervisors I owe the appreciation to the scientific way of thinking and communicating. A big thank you to my closest collaborators, Animesh Datta, with whom we had many exciting research meetings and Einar Pius, who provided his precious help. Thank you to Joe Fitzsimons for the very helpful discussions and for hosting me in the Centre for Quantum Technologies in the National University of Singapore. I would like also to thank all my teachers in the Informatics and especially Rahul Santhanam and Kousha Etessami for their suggestions and supervision.

I am grateful to Damian Marham and other members of the quantum group in Telecom ParisTech, Eleni Diamanti, Romain Allaume and researchers Anna Pappa and Leonardo Disilvestro. Thank you for hosting me multiples times (when an important part of the thesis was written) and giving me the opportunity to present and discuss my work, which gave me confidence in my first steps. I would like to thank the Simons Institute for the Theory of Computing for inviting me to their Quantum Games workshop, where precious experience was attained.

I would like to extend my gratitude to my Phd- and office-mates for their support and friendship over these years. Thank you to Andru Gheorghiu and Alistair Stewart, the discussions with whom were both enjoyable and provided significant feedback. For the same reasons, thank you to Michael Basios, Vladimir Nikishkin, Danel Ahman, Weily Fu and Kristjan Liiva.

Last but not least I would like to thank my parents for their continuous support.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Theodoros Kapourniotis*)

Table of Contents

I Introduction **1**

1 Overview **3**

1.1 General Preliminaries 8

1.1.1 Quantum Computing and Density Operators 8

1.1.2 Circuit Model 9

1.1.3 Measurement Based Quantum Computing 11

1.1.4 Computational Complexity Classes 13

1.2 Security Definitions 14

1.2.1 Interactive Proofs 16

1.3 Overview of Existing Techniques 18

1.3.1 Three Roads to Verification 19

1.3.2 Other Protocols 24

II Efficient Universal Quantum Verification **27**

2 Overview **29**

2.1 Main Results 31

2.2 Preliminaries 35

2.2.1 d-level Quantum Operations 35

2.2.2 Polynomial Quantum Error Correcting Code 36

2.2.3 Polynomial Verified Quantum Computing 38

3 Revisiting Verifiable Blind Quantum Computing **41**

3.1 Verifiable Blind Quantum Computing with Quantum Output 42

3.1.1 The Role of Fault Tolerance 46

3.1.2 Dotted-Complete Graph and Trap Independence 48

3.2 A Refined Proof of Verifiability 51

3.2.1 Expanding the Prover's Operation . . . 53
3.2.2 Decomposing the Attack . . . 56
3.2.3 Reducing the Attacks to Pauli . . . 58
3.2.4 Detection of the Pauli Attacks . . . 64

4 Verifiable Blind Quantum Computing with Localised Output 69
4.1 Localisation Gadget and Protocol . . . 69
4.2 Verifiability of the Localisation Protocol . . . 73

5 *d*-level Security 83
5.1 *d*-level Measurement-Based Quantum Computing . . . 83
5.1.1 *d*-level Universal Graph States . . . 85
5.2 *d*-level Blind Protocol . . . 91
5.3 *d*-level Verification Protocol . . . 97
5.3.1 Verifiability Proof in *d*-level . . . 97

6 An Efficient Verification Protocol 103
6.1 Impossibility of Qubit to Qudit Translation . . . 104
6.2 Composite Protocol . . . 106
6.2.1 Verifiability of the Composite Protocol . . . 107
6.2.2 Alternative Composition with Toffoli Inputs . . . 116
6.3 Noise and Abstract Security . . . 120

III Quantum-Intermediate Verification 123

7 Overview 125
7.1 Preliminaries . . . 127
7.2 Main Results . . . 128

8 One-Pure-Qubit Model Verification 135
8.1 Secure Computation with Restricted Purity . . . 135
8.1.1 Blind One-Pure-Qubit Computation . . . 138
8.1.2 Blindness Proof . . . 141
8.2 Verification of One-Pure-Qubit . . . 144
8.3 Verifiability Proof . . . 152

IV Verification and Quantum Security **163**

9 Overview **165**

9.1 Preliminaries and Related Work 166

10 From Quantum Encryption to Verification **171**

10.1 An Authentication Protocol 171

10.2 A Recipe for Quantum Authentication 176

10.3 Authentication to Verification and Classical Impossibility 177

V Blindness and Classical Security **181**

11 Overview **183**

11.1 Main Results 184

12 Secure-NAND **187**

12.1 Secure NAND Protocols 187

12.1.1 Preparing Client 189

12.1.2 Measuring Client 193

12.1.3 Bounce Protocol 193

12.1.4 Single Qubit Protocols 197

12.2 No-go Result 198

12.2.1 Generalisation: QO2 202

12.2.2 Multi Rounds 212

VI Conclusion **215**

Bibliography **223**

List of Figures

1.1 Relations between quantum and classical complexity classes 4
1.2 Verifier - Prover model of delegated quantum computation 5
1.3 Qubit Brickwork state 13

2.1 Verifiable Universal Blind Quantum Computing with quantum output 32
2.2 Generalized Toffoli gate decomposition 37

3.1 Raussendorf-Harrington-Goyal prime and dual lattice 47
3.2 Non-Clifford gate implementation in Raussendorf-Harrington-Goyal . 48
3.3 Dotted-complete graph 49
3.4 Attack analysis in VUBQC 55

4.1 VUBQC localising output gadget 70

5.1 d-level J gate implementation 86
5.2 d-level brickwork state 87
5.3 Qudit gates in MBQC 90
5.4 Qutrit gates in MBQC 91
5.5 Qutrit brickwork state 91

8.1 MBQC flow example 136
8.2 One-pure-qubit verification technique 146

Part I

Introduction

Chapter 1

Overview

Quantum technology has recently seen a surge in development, both theoretically and experimentally, and the promise of scalable quantum devices appears more realistic than it was thought before. To provide a small sample, important developments in the theoretical field include the introduction of efficient quantum surface codes [Fowler et al., 2012], the demonstration of supremacy over classical of simple optical quantum simulators [Aaronson and Arkhipov, 2011] and the advent of quantum machine learning [Lloyd et al., 2014]. In the experimental field, among many recent exciting results there is a multiple qubit superconducting computer [Kelly et al., 2015], a commercial quantum processor which demonstrates quantum entanglement [Lanting et al., 2014] and integrated photonic simulators [Spring et al., 2013].

An important challenge that arises when we increase the size of a quantum computer is that it becomes hard to predict its correct outcome. Indeed, classical simulation of an arbitrary quantum computer that runs in polynomial time requires classical exponential time. If one cannot directly predict the outcome of a quantum computation, alternative methods need to be considered. In classical complexity theory there exists an important class of problems, Non-deterministic Polynomial time or NP, containing all the problems that can be verified using a polynomial size classical string, usually called the witness, by a classical computer running in polynomial time. Factoring is a problem that belongs in this class, the witness being the factors themselves, and therefore one can verify classically the correctness of a quantum computer running Shor's algorithm [Shor, 1997]. However, the most common belief is that the class of problems, named Bounded-error Quantum Polynomial time or BQP, that contains all problems decidable with bounded error by a polynomial time quantum computer is not contained in NP (Figure 1.1). In other words, there exist problems in BQP, including the hardest problems of

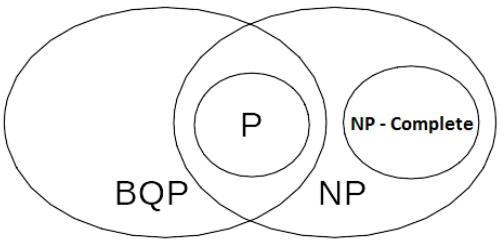

Figure 1.1: Suspected relations between some well-known classical and quantum complexity classes. It is believed that problems solvable by a quantum computer (class BQP) is not contained in the class NP of classically efficiently verifiable problems and also NP is not believed to be contained in BQP.

the class, that are believed not to be classically efficiently verifiable using a classical witness. An example of such a problem is approximating a knot invariant called the Jones polynomial, which has many applications, from DNA folding to Topological Quantum Field Theory (TQFT). Other problems that are candidates to demonstrate quantum supremacy by a medium-size quantum computer and do not have known generic verification techniques are quantum sampling problems, such as the scattering of identical bosons through a linear optical network [Aaronson and Arkhipov, 2011].

In order to verify problems outside NP one can give up on determinism and apply techniques of cryptography to get *cryptographic verifiability*. The basic setting in this approach is that of a *trusted* and weak computationally verifier delegating a quantum computation to an *untrusted* but powerful prover. The verifier is using a classical secret key to encrypt her interaction with the prover and should be able to decide whether or not the prover is returning the correct outcomes. More specifically, as depicted in Figure 1.2, the verifier (Alice) sends messages through a quantum and a classical channel to the prover (Bob). These messages may contain an encryption of the input state and an encryption of the description of the unitary to be applied to this input. Bob sends messages to Alice which contain the outcomes of the computation, which Alice can decrypt using her private key (or an update of her original private key). Alice accepts or rejects the outcome depending on the outcomes of her tests. Assuming that Bob's deviation is within the laws of quantum mechanics, the verification protocol ensures with high confidence that Alice will not accept an incorrect outcome. This problem is also studied within the context of interactive proof systems, where one ideally would like to have a purely classical verifier to verify a fully quantum prover. In **Part I** an

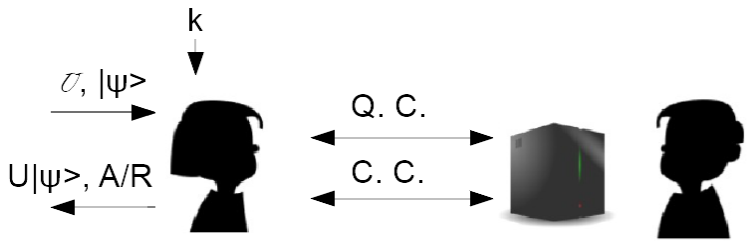

Figure 1.2: In delegated computation Alice (the verifier) has the task to compute the output of unitary U applied on input $|\psi\rangle$ but not enough power to run the computation herself, so she uses the services of untrusted but powerful Bob (the prover). Alice uses a secret key k to encode the input $|\psi\rangle$ and the description of the unitary $\mathcal{U}$ so that she is able to run secret tests on the prover and eventually decide on whether she accepts (A) or rejects (R) his returned outcome.

overview of the definitions and the existing architectures used in verification will be given.

The usual setting for quantum verification in the literature is with a prover capable of computing any polynomial size quantum circuit (computational complexity class BQP) and a verifier being universal classical but quantum restricted. Several verification protocols have been proposed, with different requirements for the verifier - all aiming to minimize the quantum preparation or quantum measurement requirements of the verifier and the communication rounds between the prover and verifier. In this setting, it is still open whether or not there exists a quantum verification protocol where the verifier is purely classical. Two prominent and efficient protocols are the protocol proposed by Fitzsimons and Kashefi (will be referred to as the FK protocol) [Fitzsimons and Kashefi, 2012], which benefits from the requirement of single state preparation for the verifier, and the protocol proposed by Aharonov, Ben-Or and Eban (will be referred to as the ABE protocol) [Aharonov et al., 2010], which benefits from linear round complexity. In this thesis, we introduce a composite protocol, which includes elements of both the FK and ABE protocols, that benefits from the single state preparation of the FK protocol and the linear round complexity of the ABE protocol (as opposed to quadratic of the FK). Therefore, this protocol is more efficient than each of its components when taken as standalone protocols. This result is accompanied by a few side-results: techniques that relate to the implementability and the composability of the original protocols. One contribution is a new proof of verifiability of the FK protocol which demonstrates the function of the protocol from a different viewpoint and is helpful when we consider both

serial and parallel compositions of the protocol with itself or other protocols. Another technique is the introduction of a gadget for the FK protocol that localizes the final output of the computation to a fixed position at the prover's side and thus is useful again in the context of composition with a different (or the same) protocol. A final contribution is the generalization of the FK protocol from 2^n-dimensional systems to d^n-dimensional systems where d is an odd prime, which might be useful for future implementations of the FK protocol, especially when novel techniques of d-level fault tolerance are employed. **Part II** of the thesis contains these results that all relate to a single universal quantum prover. These results have been published, as an extended abstract and oral presentation, in *15th Asian Quantum Information Science Conference, Seoul, 2015* [Kapourniotis et al., 2015].

Given the hardness of the engineering problem of building a fully-fledged quantum computer, some intermediate steps are worth exploring. Quantum intermediate models have been proposed, which drop some of the requirements of universal quantum computation, such as having pure quantum input, and are still interesting from a computational perspective, being able to compute answers to problems considered hard classically. In this thesis, we examine the applicability of the universal quantum verification techniques in those models. The prover will be restricted to quantum intermediate power, i.e. having characteristics that classify him between fully quantum and purely classical and the verifier wants to delegate a classically hard problem which is not able to solve by herself. A specific example of intermediate quantum power is the One-Pure-Qubit (OPQ) model that solves the problem of estimating the trace of a large matrix when given in a suitable polynomial representation [Knill and Laflamme, 1998] [Shepherd, 2006]. This model assumes that only one qubit of the input is prepared in a pure state and the rest is maximally mixed. The OPQ computer will then be able to apply coherently a polynomial-time quantum circuit. We propose a verification protocol for a OPQ prover, where a restricted verifier, being able to prepare single qubits one by one and send them to the prover, can verify the correctness of all outputs of OPQ computations. Since all the known verification techniques are based on Measurement-Based models of Quantum Computing (MBQC) and these models require, apart from the input, the supply of auxiliary pure states to implement the gates, we slightly modify the definition of OPQ to admit auxiliary states but keep the principle of limited purity over time. A constructive proof is provided for the availability of resource states for MBQC in this model of limited purity, which allows for the direct adaptation of the existing techniques used in verification, which in this case are implemented in a serial fashion.

Part III of the thesis contains these results on intermediate quantum verification which have been presented orally in *Theory of Quantum Computation, Communication and Cryptography, Singapore 2014* and published in its proceedings [Kapourniotis et al., 2014].

Quantum verification appears to have some deep connections with other security primitives, such as quantum blindness - the property of the prover not learning the input and description of the computation [Broadbent et al., 2009]. Moreover, the property of authenticated quantum communication, has already appeared in literature to be connected to verification [Aharonov et al., 2010]. Authenticated quantum communication has been proven to require quantum encryption - the property of the eavesdropper in a quantum channel not learning the transmitted quantum message [Barnum et al., 2002]. We reinforce these connections by proposing a quantum authentication protocol that is based on techniques used in quantum verification of [Fitzsimons and Kashefi, 2012] with the viewpoint that these connections might provide an indication on how to resolve of the (im)possibility problem for classical verifier quantum verification. **Part IV** is dedicated to these results.

On the other hand, a connection is established between quantum blindness and classical security, in the case a classically restricted client wants to delegate a universal gate to a classically unrestricted server and the server must not learn the input. We prove this task to be classically unobtainable and propose a protocol that achieves it by having on both client and sever a very modest quantum capability - preparation or measurement of single qubit states. This demonstrates, in the context of quantum-enhanced classical computation, that quantum techniques might be exploitable for the sake of classical security. **Part V** is dedicated to these results which have been published in the Journal of *Quantum Information and Computation* [Dunjko et al., 2016].

The rest of Part I is an introduction to the basic elements needed for understanding the rest of the thesis. In Sections 1.1.1 to 1.1.3 we establish the basic notation for quantum computation in both the circuit and the measurement-based model. In Section 1.1.4 a minimal necessary exposition to computational complexity theory is given. We proceed by introducing formally, in Section 1.2, quantum delegated computation and verification. Section 1.3 contains an overview of the existing verification techniques and protocols in literature, that will serve as a reference point to our techniques.

1.1 General Preliminaries

Here are presented briefly some basic elements of quantum computation and quantum information that will be useful for the rest of the thesis.

1.1.1 Quantum Computing and Density Operators

The state of a quantum system is represented by $|\psi\rangle \in \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space and $|||\psi\rangle|| = 1$. In this thesis we consider only systems of finite dimension N and, using the standard convention, we fix the computational basis to be $\{|j\rangle\}_{j=0}^{N-1}$. Also, when N is a power of 2 and j is in the binary representation, we speak about *qubit* systems.

Density operators is an alternative formalism to represent quantum systems, especially those whose state is not completely known. Suppose that a system is in one of the states $|\psi_i\rangle \in \mathcal{H}$, with index i, with probability p_i. The corresponding density operator or density matrix $\rho \in \mathcal{L}(\mathcal{H})$, where $\mathcal{L}(\mathcal{H})$ is the set of bounded operators in $\mathcal{H}$, is:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \tag{1.1}$$

There exists a simple test in order to find if an operator is a density operator [Nielsen and Chuang, 2010]:

Lemma 1. *An operator* ρ *in Hilbert space* $\mathcal{H}$ *is a density operator iff:*

1. ρ *has trace equal to one*
2. ρ *is a positive operator*

A two dimensional density operator can be expressed in the standard basis of identity I and Pauli matrices $\{X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\}$:

$$\rho = \frac{1}{2}(I + r_1 X + r_2 Y + r_3 Z) \tag{1.2}$$

where $r \in \mathbb{R}^3$. From positivity of ρ, r forms a unit sphere, called the Bloch sphere, where all the pure states reside in the boundary, where $||r|| = 1$.

The basic principles of quantum mechanics can be expressed in the formalism of density operators. If a system, originally in state described by density operator ρ, evolves according to unitary transformation U, the final state of the system can be written as: $U\rho U^\dagger$. Measurements of quantum systems can be represented in this formalism in the

following way: If we perform a measurement described by measurement operators M_m on state ρ, the probability of getting result m is $tr(M_m^\dagger M_m \rho)$ and the state will 'collapse' to $M_m \rho M_m^\dagger / tr(M_m^\dagger M_m \rho)$.

Another important operator used in describing composite quantum systems is the reduced density operator. First we define the *partial trace* operation: Suppose we have systems A and B. The partial trace over system B, denoted by tr_B is defined by:

$$\mathrm{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = |a_1\rangle\langle a_2|\mathrm{tr}(|b_1\rangle\langle b_2|) \tag{1.3}$$

where $|a_1\rangle$ and $|a_2\rangle$ and $|b_1\rangle$ and $|b_2\rangle$ are any vectors in space of system A and B respectively.

Then, the reduced density operator for system A is:

$$\rho^A = \mathrm{tr}_B(\rho^{AB}) \tag{1.4}$$

The reasoning behind using the reduced density operator for system A is that it provides the correct measurement statistics for the measurements made on system A.

In the formalism of density matrices the most general quantum channel can be described as a map $\mathcal{E} \in \mathcal{L}(\mathcal{H}) \rightarrow \mathcal{L}(\mathcal{H})$ that is:

- linear
- complete positive, which means that it is positive and also all extensions $\mathcal{E} \otimes I$, where I is the identity map of arbitrary dimension, are positive
- trace preserving

Thus, a quantum channel is often described as a completely-positive trace-preserving (CPTP) map. More information on quantum computing can be found in [Nielsen and Chuang, 2010] and [Heinosaari and Ziman, 2008].

1.1.2 Circuit Model

In the circuit model the computation is described as a sequence of unitary transformations, which are the quantum analogues of classical logic gates. Measurements are only taken at the end of the computation. Here, we represent unitary transformations as matrices with respect to the computational basis $\{|0\rangle, |1\rangle\}$.

An important set of unitary transformations are the Pauli matrices. The set of Pauli matrices together with multiplicative factors ± 1, $\pm i$, that is: $\{\pm I, \pm iI, \pm X, \pm iX, \pm Y,$

$\pm iY, \pm Z, \pm iZ\}$, forms a group under matrix multiplication. The general Pauli group P_n is defined to be the set of all the n-fold tensor products of the Pauli matrices together with multiplicative factors ± 1, $\pm i$ under the operation of matrix multiplication.

Another important group of unitary operators is the, so called, Clifford group [*]. The general Clifford group C_n is defined to be the group that maps the general Pauli group P_n to itself under algebraic conjugation: $C_n \equiv \{U | U^\dagger P_n U \subseteq P_n\}$. Some important members of the Clifford group are: the Hadamard gate H, the controlled-NOT or cX and the phase gate S.

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, cX \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, S \equiv \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{1.5}$$

Families of quantum gates that can be used to construct circuits that implement any arbitrary unitary operator of any size *exactly* are said to be universal for quantum computation. For example, the set that contains the cX gate and all single qubit gates is universal for quantum computation. For practical reasons, however, we need to consider discrete sets of gates. Therefore, we restrict our interest to families of gates that can be used to construct circuits that approximate any arbitrary unitary operator. Those families of quantum gates are said to be approximately universal for quantum computation.

Unless otherwise stated, we assume a quantum circuit to apply on a tensor product of computational $|0\rangle$ states, also called the *blank* states. In the case of blank state input, a circuit that consists of elements of the Pauli group and the the Clifford group followed by measurements on the computational basis is not capable of performing universal quantum computation. In fact, this circuit can be efficiently simulated by a classical computer (Gottesman-Knill theorem, see [Nielsen and Chuang, 2010]). We introduce two more gates, the *phase* $\pi/8$ gate T and the Toffoli gate, each of them being able to supplement the Clifford group into constructing a universal set of gates:

[*] defined in [Gottesman, 1998] and not related to the theory of Clifford algebras

$$T \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \text{Toffoli} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.6}$$

We note that the gate T is a special instance of the general θ-rotation around axis Z, represented as $Z(\theta) \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$. Another useful gate is the $SWAP \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

A universal set is composed by the gates: $\{H, T, cX, S\}$. Note that phase gate S can be constructed by $\pi/8$ gates, although it is usually contained in the set for convenience. This is usually referred as the standard set of universal gates. An alternative is the set composed of the gates: $\{H, \text{Toffoli}, cX, S\}$. Note that cNOT gate can be constructed by Toffoli gates, although it is usually contained in the set for convenience.

1.1.3 Measurement Based Quantum Computing

Another popular model for describing quantum computing is the Measurement-based Quantum Computing (MBQC) or one-way quantum computer [R.Raussendorf and H.J.Briegel, 2001]. This model is based on applying a sequence of measurement operators on a sufficiently large quantum resource (which can be represented by a graph, with qubits residing on the vertices), where each measurement can be adapted depending on results of previous measurements. There are different (equivalent) formalisms to describe MBQC operations. Here, we adapt the notation from [Danos et al., 2007].

A generic computation consists of a sequence of commands acting on qubits:

- $N_i(|q\rangle)$: Prepare the single auxiliary qubit i in the state $|q\rangle$;
- $E_{i,j}$: Apply entangling operator cZ to qubits i and j;
- M_i^{α}: Measure qubit i in the basis $\{\frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle)\}$ followed by trace out the measured qubit. The result of measurement of qubit i is called

result and is denoted by s_i;

- $X_i^{s_j}, Z_i^{s_j}$: Apply a Pauli X or Z correction on qubit i depending on the result s_j of the measurement on the j-th qubit.

The corrections could be combined with measurements to perform 'adaptive measurements' denoted as ${}^{s_z}[M_i^{\alpha}]^{s_x} = M_i^{(-1)^{s_x}\alpha+s_z\pi}$. A computation is formally defined by the choice of a finite set V of qubits, two not necessarily disjoint sets the input and the output, $I \subset V$ and $O \subset V$ determining the pattern inputs and outputs, and a finite sequence of commands acting on V.

Definition 1. *[Danos and Kashefi, 2006] A pattern is said to be runnable if*

(R0) no command depends on an outcome not yet measured;

(R1) no command (except the preparation) acts on a measured or not yet prepared qubit;

(R2) a qubit is measured (prepared) if and only if it is not an output (input).

The entangling commands $E_{i,j}$ define an undirected graph over V referred to as (G,I,O). Along with the pattern we define a partial order of measurements and a dependency function D which is a partial function from O^C to $\mathcal{P}^{I^C}$, where $\mathcal{P}$ denotes the power set. Then, $j \in D_i^x$ if j gets a Pauli X correction depending on the measurement outcome of i and $j \in D_i^z$ if j gets a Pauli Z correction depending on the measurement outcome of i. In what follows, we will focus on patterns that realise (strongly) deterministic computation, which means that the pattern implements a unitary on the input up to a global phase. A sufficient condition on the geometry of the graph state to allow unitary computation is given in [Danos and Kashefi, 2006],[Browne et al., 2007] and will be used later in this thesis. In what follows, $x \sim y$ denotes that x is adjacent to y in G.

Definition 2. *[Danos and Kashefi, 2006] A* flow $(f, \preceq)$ *for a geometry* (G,I,O) *consists of a map* $f : O^c \mapsto I^c$ *and a partial order* $\preceq$ *over* V *such that for all* $x \in O^c$

(F0) $x \sim f(x)$;

(F1) $x \preceq f(x)$;

(F2) *for all* $x,y : y \neq x, y \sim f(x)$ *we have* $x \preceq y$.

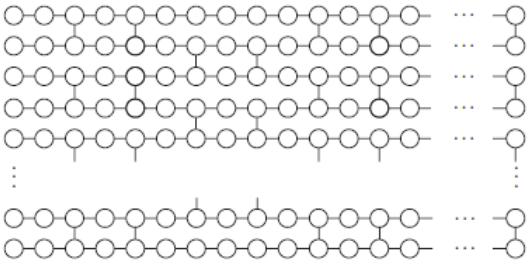

Figure 1.3: Brickwork state: A universal resource state for Measurement Based Quantum Computing with xy-plane measurements

An example of an open graph with a flow is given in Figure 1.3. In this graph, the subset of vertices of the first column correspond to the input qubits I and the subset of vertices of the final column correspond to the output qubits O. This graph state has flow function $f((i,j)) = (i, j+1)$ and the following partial order for measuring the qubits: $\{(1,1),(2,1),\ldots,(w,1)\} \prec \{(1,2),(2,2),\ldots,(w,2)\} \prec \ldots \prec \{(1,d-1),(2,d-1),\ldots,(w,d-1)\}$, where w is the width and d is the depth of the graph. The dependency functions for the corrections are: $D^x_{(i,j)} = (i, j-1)$ for $j > 1$, else $D^x_{(i,1)} = \emptyset$ and $D^z_{(i,j)} = \{(k,l-1) : (k,l) \sim (i,j), l \leq j\}$ for $j > 2$, else $D^z_{(i,j)} = \emptyset$. This graph state, named the brickwork state, since it is composed of repetitions of the same 'brick' element, has some very useful properties, such as universality for MBQC computation with only xy-plane measurements in the Bloch sphere, that will be presented later in this thesis.

1.1.4 Computational Complexity Classes

We also need to introduce some definitions from both classical and quantum complexity theory. First, we define the classical complexity class BPP (Bounded-error Probabilistic Polynomial time):

Definition 3. *A language L is in BPP if and only if there exists a probabilistic Turing machine M, such that*

1. *M runs for polynomial time on all inputs.*
2. *For all $x \in L$, M accepts with probability $\geq 2/3$.*
3. *For all $x \notin L$, M accepts with probability $\leq 1/3$.*

Next, we define the quantum complexity class BQP (Bounded-error Quantum Polynomial time):

Definition 4. *A language L is in BQP if and only if there exists a polynomial-time uniform family of quantum circuits* $\{Q_n | n \in \mathbb{N}\}$*, such that*

1. *For all* $n \in \mathbb{N}$*,* Q_n *takes n qubits as input and outputs 1 bit*
2. *For all* $x \in L$*,* $\Pr(Q_{|x|}(x) = 1) \geq 2/3$
3. *For all* $x \notin L$*,* $\Pr(Q_{|x|}(x) = 0) \geq 2/3$

Finally, the classical complexity class NP (Non-deterministic Polynomial time) is the set of decision problems solvable by a non-deterministic Turing machine that runs in polynomial time, which means that there is an accepting computation path if a word is in the language. Equivalently, it can be defined in the verifier model:

Definition 5. *A language L is in NP if and only if there exist polynomials p and q, and a deterministic Turing machine M (verifier), such that*

1. *For all x and y, the machine M runs in time* $p(|x|)$ *on input* (x, y)*.*
2. *For all x in L, there exists a string y (witness) of length* $q(|x|)$ *such that* $M(x, y) = 1$*.*
3. *For all x not in L and all strings y of length* $q(|x|)$*,* $M(x, y) = 0$*.*

1.2 Security Definitions

In this thesis we consider verification in the context of *delegated quantum computation*. There are three components in the model of delegated quantum computation: a computationally weak (compared to BQP) trusted verifier, a computationally powerful (but quantum realistic, i.e. no more powerful than BQP) and untrusted quantum prover and a secure channel, capable of establishing two-way quantum and classical communication between the verifier and the prover. When an alternative model is considered (e.g. multiple provers, see Section 1.3) this will be made explicit. The verifier is assigned the task to run a quantum computation that is incapable to run on her own due to the computational restrictions and therefore has to collaborate with the prover to get the final output, which can be in general a quantum state (classical output can be seen as a sub-case). Moreover, the verifier has access to a source of perfectly random binary strings. The strategy for the verifier is to send to the prover an encrypted version of the input and description of the computation, on which the prover will apply its operations

and produce an encrypted version of the outcome. The verifier must be able to decide, through a series of tests on the returned states, if she accepts or rejects these outcomes.

A security property that is defined in the delegated computing scenario and used as a stepping stone in the construction of some verification protocols (e.g. [Fitzsimons and Kashefi, 2012]) is blindness [Broadbent et al., 2009]. The verifier (Alice) wants to delegate a computation to the prover (Bob) while hiding both the input and the computation. Bob's possible deviation is not constrained in any way.

Definition 6 (Perfect Blindness)**.** *Let P be a protocol for delegated computation: Alice's input is a description of a computation on a quantum input, which she needs to perform with the aid of Bob and return the correct quantum output. Let* ρ_{AB} *denote the joint initial state of Alice and Bob and* σ_{AB} *their joint state after the execution of the protocol, when Bob is allowed to do any deviation from the correct operation during the execution of P, averaged over all possible choices of random parameters by Alice. The protocol P is perfectly blind if*

$$\forall \rho_{AB} \in \mathcal{L}(\mathcal{H}_{AB}), \exists \mathcal{E} : \mathcal{L}(\mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_B), \text{ s.t.} \quad Tr_A(\sigma_{AB}) = \mathcal{E}(Tr_A(\rho_{AB})) \tag{1.7}$$

Intuitively this means that at the end Bob will get a system that depends only on his private system and the choice of a deviation map that is independent of the input to the protocol. We can also extend this to allow some information to leak (e.g. the size of the computation) by making Bob's deviation explicitly dependent on this information.

In the verification cryptographic setting Alice wants to delegate a quantum computation to Bob and accept or reject the result depending on whether she thinks the returned outcome is correct (in the quantum output case, it has high fidelity to the correct output), or Bob has deviated [†]. This definition only concerns pure state inputs and unitary computations. Also, Bob's deviation is constrained only by the framework of quantum mechanics. For a delegated computation protocol to be ε-secure, it has to be correct and ε-verifiable. Correct means that when the deviation of Bob is the identity operator, the protocol should produce the correct output and accept with probability 1. ε-verifiable means that the probability of Alice accepting and the result being incorrect is bounded by a small ε, more formally:

[†]In this work we will assume that the apparatuses of Alice and Bob are perfect. Without this assumption on Bob, the protocol will also detect any errors which may stem from Bob's faulty devices.

Definition 7 (Verifiability)**.** *A protocol for delegated computation, which admits as input $|\psi\rangle$ from a set of allowed quantum states and a polynomial description of a unitary U from a set of allowed computations, is ε-verifiable ($0 \leq \varepsilon < 1$) if for any choice of Bob's strategy j, it holds that for any allowed input:*

$$Tr(\sum_{\nu} p(\nu) P^{\nu}_{incorrect} B_j(\nu)) \leq \varepsilon \tag{1.8}$$

where $B_j(\nu)$ is the state of Alice's system A at the end of the run of the protocol, for choice of Alice's random parameters ν and Bob's strategy j. If Bob is honest we denote this state by $B_0(\nu)$. Let $P_\perp$ be the projection onto the orthogonal complement of the the correct quantum output $U|\psi\rangle$. Then,

$$P^{\nu}_{incorrect} = P_\perp \otimes |ACC\rangle\langle ACC| \tag{1.9}$$

where $|ACC\rangle$ is the accept state for the indicator that Alice sets at the end of the protocol.

In the case that U are selected from all possible computations in BQP, we call this property *universal* verifiability. Typically, we can set the input state to the blank state $|0\rangle^{\otimes n}$ as we can use the first part of the computation to prepare the desired input. Generalizations for mixed input exist, but one needs to be careful who is in possession of the purification of the state. We will come back to this point in Part III.

1.2.1 Interactive Proofs

The concept of efficient verifiability can also be viewed from a complexity theoretical perspective though the formalism of Interactive Proof (IP) systems. These are systems consisting of a trusted verifier and an untrusted prover, which are allowed to interact, and were first defined in the context of the computational complexity class IP ([Goldwasser et al., 1985], [Babai, 1985]). We briefly make the connection to this formalism in this section, but we keep the rest of the thesis within the context of computer security in order to keep it as general as possible, e.g. to be able to discuss about scenarios, such as the quantum output case, which are not directly translatable as classical complexity classes. We begin we the definition of class IP, which was introduced as an extension of the complexity class NP.

Definition 8. *A language L is in IP if there is an Interactive Proof system, consisting of a polynomial probabilistic verifier A and an unbounded prover B which allowed to interact, such that*

1. *For all $x \in L$ given as input to (A,B), and for a certain (honest) B, A halts and accepts with probability at least $c = 2/3$. (completeness)*

2. *For all x not in L given as input to (A,B), and any B, A accepts with probability at most $s = 1/3$. (soundness)*

By standard classical amplification techniques, the class will not be changed if we replace the requirement by $c - s \geq \frac{1}{n^k}$ for some k and sufficiently large n.

Since it is believed that there are problems in *BQP* that are not in *NP* it makes sense to define an interactive proof with classical interaction between the verifier and the prover for proving *BQP* problems. From the known result PSPACE=IP [Shamir, 1992] and since BQP $\subseteq$ PSPACE it follows that, for an unbounded prover, BQP has an IP. However, we concentrate on the more realistic case of a bounded prover, in particular a prover that is a quantum computer (i.e. can solve BQP problems).

First, the class QPIP is defined ([Aharonov et al., 2010], [Aharonov and Vazirani, 2012]) as:

Definition 9. *A language L is in QPIP if there is an Interactive Proof system, consisting of a polynomial probabilistic (BPP) verifier A and an BQP prover B which allowed to interact, such that the same conditions of completeness and soundness as on class IP hold.*

Unfortunately, there is no protocol to demonstrate that BQP is in QPIP (the opposite is trivial). QPIP* ([Aharonov et al., 2010], [Aharonov and Vazirani, 2012] - star notation differs in the different papers) is another class that comes from adding to the verifier the ability to prepare quantum states and send them through a quantum channel. Providing this quantum enhancement to the verifier appears to be crucial for the existence of verification protocols as we will observe shortly. Then, using these protocols one can prove the weaker result, that BQP=QPIP*.

To sum up, by working with a QPIP*, we have a mostly classical verifier (BPP+the ability to prepare quantum states) and a full quantum prover and we want to verify all polynomial size quantum computations. In order to prove that there is a QPIP* for BQP we consider classical input classical output verification protocols and prove that the gap between completeness (probability of accepting a YES instance with honest prover) and soundness (accepting a NO instance with any prover) is $\geq \frac{1}{\text{poly}(n)}$ where n is the size of the problem (the description of the computation in this case). Standard poly-time classical amplification techniques involving repetition can be applied in the case Bob's

output is classical (Bob measures everything before sending the result to Alice so there is no entanglement between the received systems of different rounds). In the case Bob's output is quantum, *serial* repetition is possible, because Alice can measure her system before running the next round, but *parallel repetition* is not necessarily possible (in some protocols, such as [Fitzsimons and Kashefi, 2012], parallel repetition is possible since the parallel runs will be proven later to be separable systems for any possible deviation of Bob if we average over random parameters). In the case Alice's output is quantum, classical majority voting is not an option but one can apply alternative techniques involving quantum encoding to amplify the error probability, as we will explain later.

1.3 Overview of Existing Techniques

Having established the notion of quantum delegated computation and the property of verifiability, we present a general overview of the existing approaches. We include the protocols that are the most efficient and most representative of their own categories, and have come to our attention until the moment of writing this thesis.

An important categorizing factor between the protocols is the basic configuration of the system. Most of the protocols are based in the single-verifier/single-prover/quantum-channel configuration outlined in the previous sections. The verifier is a classical computer with a simple constant size device that is able to perform some elementary quantum operations (preparation or single gates or measurement) and the prover is usually universal for quantum computation. Ideally one wants to have a purely classical verifier, without any extra capabilities, to verify a quantum computer but whether or not this is possible without extra assumptions is a long-standing open question [Aaronson, 2007]. Its importance relates to the ability of using the scientific method of laying out an experiment and classically predicting and verifying its outcome in the limit of the high complexity of quantum mechanics, as discussed in [Aharonov and Vazirani, 2012]. There is a second configuration considered in literature, where a single verifier has to verify multiple provers that share scalable initial entanglement. There are protocols that achieve this goal using techniques for testing quantum correlations, such as CHSH games or self-testing, and have the property that the verifier is purely classical. However, these protocols impose a strong assumption, that of non-communication between the different provers during the execution of the protocol. Also, the communication complexity of these protocols is much higher, compared to the single verifier-single

prover protocols.

In the protocols of the first category (single-verifier/single-prover) there are important differences in their approach to verification. We can separate to three broad subcategories: the protocols based on some type of trapification of the delegated system (e.g. [Fitzsimons and Kashefi, 2012]), the protocols that are based on some type of quantum authentication scheme (e.g. [Aharonov et al., 2010]) and the protocols where the verifier only tests quantum correlations on the states prepared by the prover (e.g. [Hayashi and Morimae, 2015]). There also a difference in the use of hiding (encryption) in each of these schemes. In Section 1.3.1, we present the basic approaches for the single-verifier/single-prover category by outlining some representative protocols. These protocols will be used as components to our optimized verification in Part II and will also be relevant to the rest of the thesis. Later, in Section 1.3.2, we will give an overview of the rest of the protocols, including the single-verifier/multiple-prover ones.

1.3.1 Three Roads to Verification

The first approach to verification is based on injecting and testing a subsystem of traps at a random position among the qubits of the normal computation. The computation needs to be hidden from the prover so that he cannot discover the position of the traps. The second scheme is based on encoding the quantum input by a secretly randomized family of error correcting codes with some covering properties (authentication schemes), and checking if the system remains in the correct subspace after prover's operation. The hiding is on the input states so that the prover cannot retrieve the secret key used in encoding. In this thesis we will focus more on these two approaches which, given the differences in their structure, can be referred to as *subsystem* and *subspace* verification correspondingly. The third scheme is based on the prover preparing of a large entangled resource state and the verifier performing measurements on it to run the computation and at the same time verify the resource by testing stabilizer correlations.

The three schemes have different assumptions of trust and different requirements for the verifier and therefore are useful in different scenarios. In brief, the trap-based verification and authentication-based verification are appropriate when we trust the preparation devices, while the resource-testing is appropriate when we trust the measurements. Between the first two, the existing trap-based protocols have the benefit of having to prepare only single qubit states, while the existing authentication-based protocols have the benefit of fewer rounds of communication between the verifier and

the prover (number of rounds is equal to the Toffoli depth). In the next paragraphs we overview a protocol from each scheme and give more details on the assumptions and requirements.

1.3.1.1 Trap-based verification

Verifiable Universal Blind Quantum Computing (VUBQC) [Fitzsimons and Kashefi, 2012], or *FK protocol* from the names of the authors, is a trap-based protocol and will be relevant to the optimizations we attempt in the next two parts of this thesis. It is based on the pre-existing Universal Blind Quantum Computation protocol [Broadbent et al., 2009]. The latter is a protocol for secure delegated quantum computation where the server does not learn anything about the client's input and computation (thus blind, as defined earlier). The underlying model for both UBQC and VUBQC is a delegated version of the Measurement-based quantum computing, where the verifier prepares the single qubits composing the resource state and the measurement angles which define the basis to measure, while the prover entangles and performs the measurements. For blindness, verifier has to encrypt the quantum and classical states she sends to the prover, with whom she has to interact during the execution of the computation to be able to implement the adaptive measurements. For verification the verifier has to add a trapification subsystem at a random position among the sent states, which gives deterministic results for the measurements of the prover, thus can be checked to see if the prover is honest. Since prover does not know when he measures the computation and when he measures the trap he cannot cheat without getting caught with some probability. A not very detailed description of the protocol is given here, while a formal protocol with more explanation is given in Section 3.1, so that it can be used in our optimizations.

FK protocol (sketch) [‡]

1. Alice has a description of a computation in the measurement-based quantum computing (MBQC) model.

2. Alice embeds this MBQC computation in a different graph that has the following property: A subsystem of isolated qubits (named traps) can be positioned uniformly at random between the normal computational qubits without corrupting the flow. In order to isolate the traps, qubits in state $|0\rangle$ (named dummies) are

[‡]Protocol 8 in [Fitzsimons and Kashefi, 2012]).

placed as their neighbours and measured in the XY plane of the Bloch sphere so that the effect of the entangling operations of Bob is cancelled.

3. Alice prepares all single qubits needed for the construction of the resource state by Bob: Instead of preparing them as $|+\rangle_i$ qubits, she performs a random rotation by angle $\theta_i \in A \equiv \{0, \pi/4, 2\pi/4, \ldots, 7\pi/4\}$ around Z axis. For the dummies Alice instead applies a random X Pauli operation and a pre-rotation by Z on all their neighbours to cancel this effect.

4. Bob receives the qubits one by one and entangles them (cZ) to construct the entangled state, the structure of which is public.

5. Alice sends to Bob at each step i the measurement angle δ_i that encrypts the computational angle ϕ_i (updated to ϕ'_i based on the previous measurement results): $\delta_i = \phi'_i + \theta_i + r_i\pi$. This effectively cancels the pre-rotation by θ_i (since it commutes with cZ) and adds a bit flip on the outcome to hide the measurement result.

6. Alice receives the measurement result and decrypts it by XOR-ing with r_i.

7. For the isolated traps $\phi_i = 0$ so that the measurement outcome depends only on r_i and is deterministic.

8. At the end Alice accepts the outcome returned by Bob if all traps give the correct outcome.

9. To achieve exponentially small ε verifiability, Alice encodes the computation in a fault tolerant MBQC pattern, using a QECC that detects $\log 1/\varepsilon$ errors. This procedure is explained in Section 3.1.1. The underlying idea is that, in order to corrupt the computation, Bob has to corrupt at least $\log 1/\varepsilon$ qubits, which increases his probability of hitting a trap and get caught.

In order to pick a graph that has the desired property of hiding the trap without interrupting the flow, the degree of each vertex becomes linear on the graph size. The underlying graph state, named the dotted-complete graph and presented later, is of quadratic size. Thus, the quantum requirement of Alice is to prepare and send $\widetilde{O}(n^2) \times O(log(1/\varepsilon))$ [§] *single qubit states* (the ε factor comes from the FT encoding)

[§]Notation $\widetilde{O}$ is a variant of O that ignores logarithmic factors, i.e. $f(n) \in \widetilde{O}(g(n))$ means that $\exists k : f(n) \in O(g(n)\log^k g(n))$.

and on-line classical communication of $\widetilde{O}(n^2) \times O(log(1/\varepsilon))$, where n is the size of the computation. The goal in Part II will be to reduce these requirements to linear on n while keeping the modest requirements for the verifier preparation device.

1.3.1.2 Authentication-based verification

The second verification protocol, that will be used in our optimization in Part II, is the Polynomial QAS-based verification [Aharonov et al., 2010] or *ABE protocol* from the names of the authors. This protocol is based on the Quantum Authentication Schemes (QAS) [Barnum et al., 2002] which are used to transmit quantum states through an untrusted quantum channel and check the integrity of the received state (more on authentication schemes in Part IV). The transmitter and receiver share a secret classical key which is used for secret (encrypted) encoding and decoding of the quantum state. The main idea of Polynomial QAS-based verification is that the quantum states sent from the verifier to the prover in the delegated computation scenario are secretly encoded by a polynomial QAS. A QAS uses a family of quantum codes, parametrised by a secret key, that have the special covering property that any Pauli error on the state is detected by all but an exponentially small number of members of the family. Then the prover must apply the computation on the QAS-encoded state in a way that the output remains in the valid subspace (up to an update on the secret key on the verifier side). The prover cannot deviate and keep the state in the valid subspace if he does not know the secret key of the QAS. An extra Pauli key is used to reduce all general attacks to Pauli operators. Here, we give a short description of the protocol, while a more technical one is provided in Section 2.2.3. It's worthwhile to observe that this protocol, as it was the case in the FK protocol, is defined in a model of computation that is based on measurement.

ABE protocol (sketch) [¶]

1. Alice has a description of a computation in the Gate Teleportation model where the non-Clifford operations, Toffoli gates in this case, are performed using auxiliary states, $Toffoli|++0\rangle$ in this case (Toffoli states), computational basis measurements and Clifford corrections depending on the measurement outcomes (the principle is similar to MBQC, more details in Section 2.2.1). For the protocol to work the computation must apply on *qudits*, i.e. d-level systems, where d is an odd prime.

[¶] Protocol 4.2 in [Aharonov et al., 2010]

2. She encodes the input (typically the blank state plus the magic states) using a polynomial-QECC code where the encoded states are superpositions of all polynomials (in point-value representation) of degree $\leq \mathbb{p}$. She also applies a random sign on each polynomial value, and keeps this key secret from Bob.

3. She applies an independent random Pauli operator on each of the physical states of the QAS state and sends it to Bob.

4. Bob performs the computation while classically communicating with Alice to implement the corrections needed after each measurement step. Alice updates her secret keys accordingly.

5. Alice receives the final output, undoes the Pauli rotations and the sign key and checks if the resulting state is encoded by a polynomial code of low degree. If not, she rejects otherwise she accepts. This gives an ε-verifiable protocol with ε exponentially small on $\mathbb{p}$.

The quantum requirement for the verifier, which comes from the need to prepare secretly encoded states, is to be able (by means of a fixed size universal quantum computer) to prepare and send $O(n)$ states of $O(1/log(\varepsilon))$ entangled qudits of $O(1/log(\varepsilon))$-level, where n is the size of the computation. The classical communication requirement is $O(n) \times O(log(1/\varepsilon))$ rounds during the execution of the computation. The goal in Part II is to use the techniques of this protocol as part of a composite construction to take advantage of the linear (on the input) communication complexity.

1.3.1.3 Resource testing verification

In [Hayashi and Morimae, 2015] an alternative approach is taken to verification, where the verifier has a single state measuring device instead of a preparing device and uses it to apply the measurements needed to implement an MBQC computation. On the other hand, the prover is the one who prepares, entangles and stores the quantum states. Alice verifies that Bob keeps the correct state by checking the correlations of stabilizer measurements. It can be seen therefore as a protocol for state certification for large graph states. Also, the protocol provides hiding of the computation, since the Bob never learns the measurement angles by the non-signalling principle. In terms of resources Bob has to prepare $O(n)$ states and send them to Alice one by one. Alice should be able to apply a universal set of single qubit measurements. Quantum communication

is on-line. Therefore, it requires preserving long distance entanglement between the verifier and the prover during the computation.

1.3.2 Other Protocols

In [Aharonov et al., 2010], another protocol was proposed based on the principle of QAS-based verification, this time with a verifier able to apply one-qubit and two-qubit gates. This protocol is based on a random Clifford applied on the input before sending it to the prover. The prover is used as an untrusted quantum storage device between the operations applied by the verifier. Every time she wants to apply a quantum gate, she asks for the corresponding encoded qubits (constant size), decodes them by applying the Hermitian conjugate of the secret Clifford and measures each of the appended qubits in the computational basis and aborts if she gets a 1. After, she applies the gate and encodes the output qubits again with new secret Clifford operators. This process is repeated until all gates have been applied. In terms of resources, the verifier needs to be constant size ($O(log\frac{1}{\varepsilon})$) universal quantum computer and the prover a polynomial size quantum memory. The dimension of the single quantum systems is not dependent on ε, as it was in the case for the ABE protocol. Also, the verifier needs to exchange with the prover during the execution of the protocol $O(n) \times O(\log \frac{1}{\varepsilon})$ quantum states. From the description of the protocol, also one understands that it requires preserving long distance entanglement between the verifier and the prover during the computation.

In the recent work of [Kashefi and Wallden, 2015], a modification of the trap-based verification of the FK protocol is presented. In particular, the underlying graph, used as the resource for both the computation and the traps, is simplified over the original FK protocol. The graph used, named the triple dotted graph, since it is made of components that are bipartitions of three qubits with a vertex (dot) injected in each edge, has the desired property of hiding the trap without interrupting the flow. But, this graph has also linear size on the computation since the degree of each vertex is constant which is an improvement of the FK protocol. The requirement for the verifier is to prepare single qubit states as in the original protocol. This result is comparable to our work, especially with the composite protocol presented in Part II, in terms of complexity but has some differences in both the construction (type of the graphs) and the depth complexity that will be progressively discussed through the thesis.

Another protocol with a verifier that prepares auxiliary states, appeared recently in [Broadbent, 2015] and is also comparable in efficiency to our work in Part II. This

protocol and the rest that follow in this section do not give an option to verify the quantum output with exponentially low ε, as it was the case with the trap-based and authentication-based protocols. The scope of this protocol and of the ones that follow is to prove the existence of an interactive proof, therefore they are interested only in the classical output. The main technique, here, is similar to trapification: in each run the verifier chooses at random between running the actual computation or running a test to detect the prover's deviation. The protocol is such that the prover cannot distinguish between the different cases and therefore cannot cheat without the risk of getting caught. Alice has to prepare random state from the set A, same to the FK protocol, however she needs a linear number of them. There is a constant size classical interaction for each T or H gate therefore the overall classical communication is linear on n, where n is the size of the computation. The prover should be able to perform universal Clifford computation and Pauli measurements.

A different approach on verification is taken in [Reichardt et al., 2013], where a purely classical verifier is able to verify two quantum provers which share entanglement but are not allowed to communicate to each other. In this case the verifier, by being able to interact only classically with the two computers, is able to verify whether or not they share the correct initial state and they perform correctly the instructed operations so that they jointly produce the correct outcome of the computation. Verifiability comes from the fact that the verifier can perform CHSH games on the prover's devices and that the prover cannot have the incorrect state or perform the incorrect measurements and win many games. The protocol is not practical because of the unrealistic number of resources and rounds of communication needed (the number of states and rounds of communication scale with $O(n^c)$, for some appropriate $c \geq 8192$), however since they are still polynomial it demonstrates the theoretical possibility of such a protocol to exist. Also this result is a way to achieve device independence in quantum cryptography (since the verifier is purely classical). Device independence is the concept that can be attached to quantum security if the protocol remains secure even when the quantum devices used are faulty. In the case of full device independence, the devices are totally untrusted and can behave in any possible way. Moreover, the delegated computation remains blind from the prover.

The protocol in [McKague, 2013] applies on a more distributed setting, where a classical verifier delegates her computation to a polynomial number of untrusted quantum devices. The assumptions are again that these quantum devices share entanglement but are not allowed to communicate between them. During the protocol each quantum

device is instructed to perform only one measurement and the correctness of the induced states is checked by the verifier by a graph state self-test protocol. The authors prove a tighter, compared to the existing, bound for the self-testing of graph states, where the error scales polynomially on the number of test repetitions and therefore can be used for a theoretically efficient protocol - the complexity is improved also compared to the two prover protocol of [Reichardt et al., 2012], however it remains high (quantum states and communication required is of $O(n^{22})$).

In [Gheorghiu et al., 2015] by combining techniques of the double-server verification of [Reichardt et al., 2012] and the single-server trap-based verification of [Fitzsimons and Kashefi, 2012] a composite protocol is presented which achieves device independent verification with fewer resources. In particular, the verifier instructs an untrusted device that is not allowed to communicate with the prover to prepare on the prover's side the quantum states needed for the execution of the single-server verification protocol. The verifier is able to verify the final outcome of the two phases, without having to trust his preparatory quantum device, thus achieves device independent verification. However, since the full mechanism of the double-server verification protocol is not needed the resources reduce significantly, yet still being high enough for a realistic implementation.

In [Hajdusek et al., 2015], simultaneously with [Gheorghiu et al., 2015], a similar composition of the two mechanisms, of verified state preparation and trap-based verified computation, is considered. Here the verified state preparation is achieved through self-testing instead of state tomography. This achieves a device-independent protocol, where the verifier has in his hands an untrusted quantum device that is teleporting states to the prover but is not able to communicate with the prover directly so that it can be tested by the self-testing procedure. The prover is able to run measurement-based quantum computing in order to perform the trap-based protocol. This protocol gives verifiability with resources reduced to $O(n^4)$.

A number of protocols exist that verify a state shared by many parties. E.g. in [Pappa et al., 2012] a resource verification protocol is presented where GHZ states are distributed to many parties by an untrusted device and the parties can verify the existence of the GHZ state by trusting only their measurements.

Focusing on the blindness property only, several recent papers [Giovannetti et al., 2013, Mantri et al., 2013, Perez-Delgado and Fitzsimons, 2014] have achieved better than linear bound while compromising either unconditional security or the simplicity of the verifier's operation. We believe that it may be worthwhile to explore these new blind techniques with the goal of designing more efficient verification protocols.

Part II

Efficient Universal Quantum Verification

Chapter 2

Overview

The key element for constructing a practical verification protocol is to have a minimal verifier, that will be easy to build and control. A practical verifier should have only very limited quantum capabilities, in terms of the preparation, measurement or evolution operations that she can apply, thus being far from a universal quantum computer. On the other hand, such a protocol should be limited on the amount of classical and quantum communication between verifier and prover, especially during runtime. These requirements are important, not only from a theoretical viewpoint, as we will be approaching the regime of a purely classical, non-interactive verifier, but also from a practical viewpoint, since verifying the emerging quantum devices will be crucial for the proof of their existence and potential. One can envision that, in the future, full scale quantum computers will be available only in research institutes and thus the idea of delegated quantum computing, or quantum cloud will become relevant.

There are two verification protocols, namely Verifiable Universal Blind Quantum Computing (VUBQC) [Fitzsimons and Kashefi, 2012] and Polynomial QAS-based verification [Aharonov et al., 2010], already outlined in the previous part, that have the most modest, among the known protocols, requirements for the verifier. The first has very modest preparation requirements, while the second has a lower communication cost. In particular, as already stated in the previous part, in the FK protocol the verifier is required to have the minimal quantum capacity of preparing single random qubits, while due to the complexity of the underlying resource state necessary for the server's universal quantum computation, the communication overhead is quadratic in the input size. In contrast, the ABE protocol enjoys only a linear overhead in the input size, however, the cost is the verifier's requirement of preparing highly entangled secretly encoded states which scale with the security parameter. The main contribution of this

part of the thesis is to show the best properties of the two approaches may be achieved, by combining aspects of the two protocols. In doing so, we also prove several new properties of these two protocols that could become useful for other purposes as well, such as protocol implementability and composability.

The main idea is to use the verifiable computation of the FK protocol to prepare, on the prover's side, the inputs encoded by the randomized polynomial quantum error correcting code (QECC) used in the quantum authentication scheme (QAS) of the ABE protocol. In other words, the states the verifier would have to prepare and send in the first step of the original ABE protocol. On these states the public logical circuit of the ABE protocol is applied, and the output is tested to verify that it is in the correct encoded subspace. If either the trapification (verification process of the FK protocol) or decoding procedures (verification aspects of the ABE protocol) fails, the verifier rejects. The partitioning of the underlying resource state used in the FK protocol into smaller sub-states for each separable logical qudit leads to an improved communication complexity of the composite protocol compared to the original FK protocol, while maintaining the FK protocol's preparation simplicity.

The main contributions of this part are:

- A refined modular proof of verification for the FK protocol, in which any deviation of the prover can be written as a combination of the correct operation - that disentangles the trap from the computational system - and a local attack on each of the qubits of the held system, thus the system remains always disentangled from the private system of the prover. This allows for both parallel and serial composition of the FK protocol. Also, if we reduce our verification criterion to have the correct output up to a CPTP map that is independent of the computation - a scenario that arises in composite protocols such as ours - a modified version of the FK protocol that uses an extra gadget graph, is proven suitable.

- An odd prime dimension adaptation of the FK protocol is presented, after building odd prime dimension MBQC and MBQC blind protocol. A new bound on verifiability is proven, which is better than the bound of 2-level systems.

- A composite FK-ABE verification protocol is given that reduces quantum and classical communication complexity of FK from quadratic to linear, while keeping modest quantum preparation requirement for the verifier. Also, an alternative version of this protocol is proposed, when Toffoli states are available to the verifier,

where the the delegated preparation of the encoded QAS states is a Clifford computation and therefore can be done with one round of communication.

2.1 Main Results

A few results are proven prior to the construction of the composite protocol, which might be useful in other contexts. We outline these results here, while a more detailed presentation is given in the following chapters of this part of the thesis.

Since we use the FK protocol for the preparation of the QAS verifiable states, which are quantum entangled states, the version of the FK protocol that is appropriate is the one that admits *quantum output*. Moreover, we require an exponentially low probability ε of failing to prepare a correct QAS verifiable state so that the composite protocol itself has exponentially low failure probability. In the case of quantum output (as opposed to classical output where on can apply classical amplification techniques, such as majority vote) to amplify the probability we encode the computation using a quantum Fault Tolerant (FT) scheme that forces the attacker to have a bigger footprint on his attack and therefore increases his chances of getting caught by the trap (a technique described in the original FK, see also Figure 2.1 and Section 3.1.1 for more details). In Chapter 3 we give an explicit description of an amplified FK protocol with quantum output, not given in the original paper. Special care has been taken to provide a protocol where Alice has the extra quantum requirement of applying Clifford gates for decoding the final quantum output she receives from Bob. Crucially, in the composite protocol later, this Clifford circuit can be delegated to Bob and therefore the requirement for Alice will remain the single qubit preparation.

The verifiability of the amplified FK protocol with quantum output is also proven in Chapter 3, formally stated as Theorem 1, the essence of which is given here:

Theorem 1. *(sketch) FK protocol can be modified to admit quantum output and is perfectly blind and ε-verifiable, where $\varepsilon = \frac{1}{c^{\mathbb{p}_1}}$, where $\mathbb{p}_1$ is a parameter of the FT procedure and c a constant that depends on the graph used. The extra requirement compared to the original FK protocol is the ability of Alice to apply a decoding Clifford circuit and Pauli measurements.*

To prove this theorem an alternative approach is taken, compared to the proof of verifiability in the original paper. The new technique reduces first any attack from Bob to a convex combination of Pauli operators, using some of the randomness of the

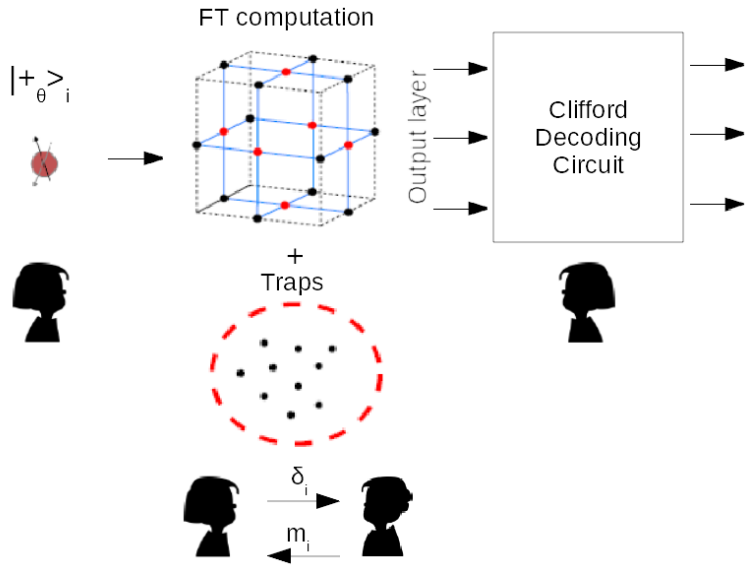

Figure 2.1: FK protocol with quantum output and verifiability amplification to exponentially small ε. Alice delegates to Bob the blind execution of an MBQC pattern that contains both a fault tolerant implementation of the computation using a QECC that detects $\log 1/\varepsilon$ errors and a set of traps mixed within. In order to corrupt the computation, Bob has to corrupt at least $\log 1/\varepsilon$ qubits which amplifies his probability of hitting a trap. Alice has to decode the MBQC output layer herself, otherwise Bob would increase his chances of cheating by attacking the decoded output.

protocol, so that the analysis for the traps and the FT procedure applies only for Pauli attacks. This technique is useful when we need to write the state of the system at any point during the operation of the protocol, not only at the end, as it is the case with the composite protocol where the output is not returned to Alice. Also, it makes easier to prove the following corollary:

Corollary 1. *For a parallel composition of FK protocols with quantum output, where each protocol uses an independent set of random variables having* ε*-verifiability individually, the final aggregate quantum output of all protocols will also be* ε*-verifiable when averaged over all parameters.*

Towards building a full composite protocol, a number of intermediary protocols, or variations of the FK protocol, are presented in Chapters 4 and 5. As mentioned before, in the composite protocol the first phase is to prepare remotely on Bob's side the state needed for the execution of the ABE protocol. Bob then has to run a public logical circuit on this state to perform the actual computation. Therefore, the output of the first phase must be in a fixed position, known to Bob, and cannot contain any trap (in fact it

could contain a trap, but revealing the position of it will make it useless for verification purposes). In general, having the output at a fixed position will break the verifiability property of the FK protocol, since it will reveal some information about the positions of the traps in the previous layers. In Chapter 4, we give a version of the FK protocol that uses a special output gadget to fix the output position without breaking verifiability, up to a CPTP map on the output (since Bob can always replace the output with his private system in this case). This property is stated in Theorem 2 in Chapter 4, a sketch of which is given here:

Theorem 2. *(sketch) The output of the FK protocol with the localising gadget after decoding of the FT procedure and decrypting of the quantum one-time-pad and averaged over all random parameters of Alice, is* ε*-verifiable equal to the correct output of the computation* $|\psi_c\rangle$ *up to a CPTP-map where all Kraus operators are Pauli operators and is independent of* $|\psi_c\rangle$*. Moreover, the output is at a fixed position on the graph, independent of the random parameters.*

If $|\psi_c\rangle$ is a QAS encoded state and is prepared by the localisation protocol as above and then fed to the ABE protocol, the requirement that the deviation is independent of the state (and therefore the secret parameters of the QAS) is enough to prove the verifiability of the final output of the ABE protocol as we will see shortly.

Another variation of the FK protocol, presented in Chapter 5, is the d-level version of the protocol, where d is an arbitrary odd prime dimension. This would be proven useful for the composition of the FK protocol with d-level protocols, such as the ABE protocol, but also can be useful in general for the implementability of the FK protocol, especially with the appearance of more efficient d-level FT procedures [Watson et al., 2015]. In Section 5.1 we present MBQC for d-level systems, using the same notation of the FK protocol, blind MBQC for d-level systems and eventually a d-level of the FK protocol, where Alice has to prepare single qudits and send them to Bob. A new bound, that depends on d, is proven as Theorem 3 in Chapter 5 for the case of d-level systems. It follows a sketch of this theorem.

Theorem 3. *(sketch) There exists a* d*-level version of the FK protocol, up to the existence of* d*-level topological codes, that is* ε *verifiable with* $\varepsilon = (1 - \frac{(d-1)c'}{d})^{\mathbb{p}}$*, for graph parameter* $c' < 1$*, FT parameter* $\mathbb{p}$ *and* d *odd prime.*

For the d-level FK protocol with quantum output to exist, we conjecture the existence of the qudit version of the fault tolerant QECC used in FK [Raussendorf et al., 2007] or equivalent.

We examined two different ways to build the composite protocol and found that only the second is possible. Our first approach was to use 2-level (normal) FK protocol to simulate the preparation of the d-level QAS verifiable state and translate the quantum one-time-pad that comes naturally at the end of this phase (and we cannot reveal it without breaking the FK protocol) to a d-level one-time-pad that will be used in the ABE protocol phase. This translation we proved in Chapter 6 to be impossible deterministically for d being an odd prime. Notice that the dimension d being a prime is essential of the ABE protocol to get the field structure for the polynomials. The second approach, which we followed, is to use the d-level FK protocol for the preparation of the QAS verifiable state and directly use the output d-level one-time-pad as the Pauli key of the ABE protocol.

The composite protocol is presented in Section 6.2, an outline of it given here:

FK-ABE composite protocol (sketch)

1. Alice and Bob execute a specified number of instances of the qudit analogue of the FK protocol (that includes the localising gadget) to prepare in Bob's system each of the desired polynomial-QECC encoded state of the ABE protocol.

2. Bob applies the publicly known Clifford decoding circuit for the FT used in FK protocol for probability amplification and Alice updates her one-time-pad keys.

3. Alice and Bob apply the polynomial-QECC logical circuit on the encoded states following the steps of the ABE protocol. Alice updates her secret keys.

4. Bob sends the final measurement outcomes to Alice and Alice employs the detection and decoding procedure of the randomised polynomial-QECC.

The composite protocol is proven to have the required properties of correctness and verifiability, as described in Theorem 4 in Chapter 6 and sketched here.

Theorem 4. *(sketch) The composite protocol is correct and* ε*-verifiable, with* $\varepsilon \leq \varepsilon_1 + \varepsilon_2$ *where* $\varepsilon_1 = \frac{1}{c^{\mathbb{p}_1}}$, $\varepsilon_2 = \frac{1}{2^{\mathbb{p}_2}}$ *where* $\mathbb{p}_1$ *and* $\mathbb{p}_2$ *are the QECC distance parameters of the FK and ABE phase and* $c > 1$ *is a parameter of the FK phase which depends on the structure of the graph used.*

In terms of preparation requirements, in the composite protocol Alice needs to prepare random single $O(1/log(\varepsilon))$-dimensional qudit states. To put this into context, we remind that in the FK protocol the quantum requirement of Alice is to prepare single

qubit states and in ABE protocol the quantum requirement of Alice is to prepare states of $O(1/log(\varepsilon))$ entangled qudits of $O(1/log(\varepsilon))$-level.

In terms of communication complexity, the composite protocol requires classical and quantum ($O(1/log(\varepsilon))$-level) communication with number of rounds that scales in $O(n\text{Polylog}(\frac{1}{\varepsilon}))$, where n is the size of the computation. In the FK protocol the classical and quantum (qubit) communication was $\widetilde{O}(n^2) \times O(log(1/\varepsilon))$, therefore the composite protocol gives a quadratic improvement over FK in its scaling with n. In the ABE protocol the communication cost is in $O(nlog(1/\varepsilon))$ of classical and quantum states.

An alternative version of the composite protocol is also given, that reduces the round complexity, but assumes the availability of (unencoded) Toffoli states. This version is described in Section 6.2.2 and its verifiability is proven as Theorem 5 and outlined here:

Theorem 5. *(sketch) There exists an alternative composite protocol, with extra Toffoli inputs (or the ability to prepare them) for Alice, and gives the same verifiability with the first composite protocol and at the same time has round complexity equal to the Toffoli-depth of the delegated computation, where Toffoli-depth is the depth of the computation if we consider only the subset of Toffoli gates composing it.*

2.2 Preliminaries

In Section 2.2.1 follows a short exposition to d-level quantum computation that is relevant only to this part of the thesis. In Section 2.2.2 the quantum error correcting codes used in the QAS of the ABE protocol are described. In Section 2.2.3 the ABE protocol is presented more formally than previously, so that it can be used as a component to the composite protocol.

2.2.1 d-level Quantum Operations

First we describe the d-level generalized gates that will be useful in the following constructions. All additions and multiplications in these definitions are performed modulo d, unless otherwise stated. The generalized Pauli operators over $\mathbf{F}_d$ are defined as: $X|a\rangle \equiv |a+1\rangle$, $Z|a\rangle \equiv \omega_d^a|a\rangle$, where $\omega_d = e^{2\pi i/d}$ is the primitive dth-root of unity. The generalized Hadamard (or Fourier) gate F is defined as:

$$|a\rangle \mapsto \frac{1}{\sqrt{d}} \sum_{b\in\mathbf{F}_d} \omega_d^{ab}|b\rangle.$$

The cX is defined as: $|a,b\rangle \mapsto |a,a+b\rangle$ and cZ as: $|a,b\rangle \mapsto \omega_d^{ab}|a,b\rangle$. The generalized phase gate S is defined as:

$$|a\rangle \mapsto \omega_d^{\frac{(a+1)a}{2}}|a\rangle$$

Another important family of Clifford gates are the permutations S_c, $c \in \mathbf{F}_d$, defined as:

$$|a\rangle \mapsto |ca\rangle$$

The generalized Toffoli gate which complements the set of Clifford gates to provide universality ([Howard and Vala, 2012], [Anwar, 2014]) is defined as: $|a,b,c\rangle \mapsto |a,b,c+ab\rangle$. An alternative to the Toffoli gate to complement Clifford gates for universality is the family of 'so-called' generalized '$\pi/8$' gates [Howard and Vala, 2012] one of which, used in this paper, is defined as: $T|a\rangle \equiv \omega_d^{a^3}|a\rangle$ with the exception of qutrits where $T_3|a\rangle \equiv \omega_9^{(a^3 \bmod 9)}|a\rangle$.

An explicit construction for a universal set of gates for arbitrary number of qudits of odd prime dimension that we will use in this part is the following: $\{Z,F,S,T,cX\}$. Gate Z is not necessary to complete the set, since we can generate it from S and F for odd prime dimensions, but we include it for convenience. The fact that this set is universal comes directly from the following: From [Clark, 2006] (Theorem 2.3) we have that the Clifford group can be generated by $\{F,S,cX\}$. Finally, by [Howard and Vala, 2012] we have to add the T gate to the Clifford computations to get universality.

2.2.1.1 d-level Gate teleportation

The model of computation used in the ABE protocol ([Aharonov et al., 2010]), sometimes called the gate teleportation model, all Clifford gates are performed by applying unitary operators, while the non-Clifford gates are performed by entangling with magic non-Clifford states (Toffoli states in this occasion), measuring in the computational basis and performing Clifford correction operators on the remaining states. See Figure 2.2 for the circuit that implements the generalized Toffoli.

2.2.2 Polynomial Quantum Error Correcting Code

The class of quantum error correcting codes used in the ABE protocol, is based on polynomial error-correcting codes [Aharonov and Ben-Or, 2008]. By randomizing

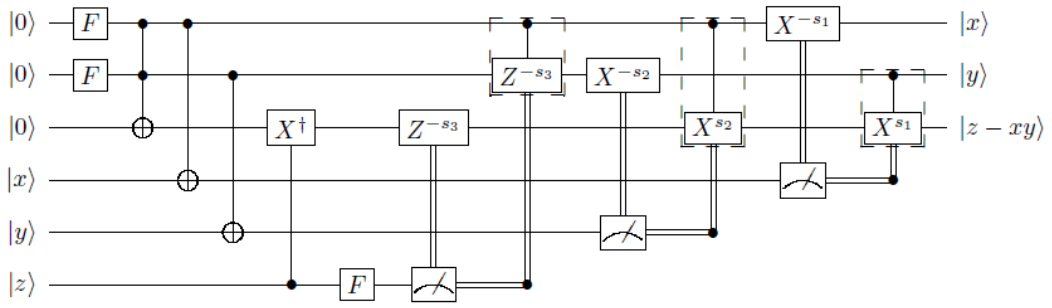

Figure 2.2: Implementation of generalized Toffoli (adapting from the qubit Toffoli in [Nielsen and Chuang, 2010]).

these codes using a secret sign key one can acquire a Quantum Authentication Scheme (QAS) [Barnum et al., 2002], usable by two parties who want to authenticate a quantum state exchanged through a malicious quantum channel. A QAS scheme can be used as a component for a secure quantum multiparty computation [Ben-Or et al., 2006], where the property of self-duality of the the *signed* polynomial error-correcting codes makes the protocol simpler. The *signed* polynomial error-correcting codes are suitable in the context of verification of a single quantum prover as demonstrated in the ABE protocol, due to some useful properties discussed later.

Specifically, the code operates over the field $\mathbf{F}_d$, where d is prime and represents the dimension of the quantum system. The code uses polynomials $f(\cdot)$ with coefficients from $\mathbf{F}_d$ and with a maximum degree of $\mathbb{p}$, where $\mathbb{p}$ is a parameter of the code. These polynomials are evaluated at *m distinct non-zero points* from $\mathbf{F}_d$: $\{a_1 \dots a_m\}$. To be able to choose m distinct non-zero points from $\mathbf{F}_d$ we impose the restriction: $m \leq d$. Moreover, we set $m = 2\mathbb{p}+1$. Finally, the code uses a secret sign key that is a uniformly randomly chosen string $\mathbf{k} \leftarrow \{\pm 1\}^m$.

Given a quantum state $|a\rangle$, where $a \in \mathbf{F}_d$, we can encode it using the signed polynomial error-correcting code:

$$|a\rangle \rightarrow \frac{1}{\sqrt{q^{\mathbb{p}}}} \sum_{\{f|deg(f)\leq \mathbb{p}, f(0)=a\}} |(k_1 f(a_1), \dots, k_m f(a_m)) \mod d\rangle$$

where the mod d is taken element-wise in the label of the right-hand side ket.

To detect if there was any Pauli error in the transmitted codeword, the receiver undoes the sign key (by applying a Clifford circuit), applies the syndrome measurements and checks if the resulting string is the representation, at m distinct points, of a polynomial of maximum degree $\mathbb{p}$. If not, it aborts and declares the state invalid.

The probability of accepting an incorrect state (when the channel applies an arbitrary Pauli attack) is exponentially small on $\mathbb{p}$. In order to acquire the same property for any type of malicious attack, an extra quantum one-time-pad is added on top of the existing randomization, as it is the case in the ABE protocol.

2.2.3 Polynomial Verified Quantum Computing

In the ABE protocol the input is encoded by Alice using the secretly randomised QAS presented in Section 2.2.2 which has algebraic properties that allow for the secret updating of the random key throughout the execution of the computation. Specifically, all Clifford gates applied are transversal or semi-transversal which allows the sign key on each qubit to remain the same. The non-Clifford gates are applied by state injection and teleportation using Clifford gates and Pauli measurements. Since all gates are Clifford the Pauli key commutes and is updated by the verifier who knows which gates are applied. The QAS has also the covering properties that allow Alice to detect Bob with high probability after decoding the final state. The computation performed on the encoded input can be publicly announced without compromising the verifiability of the protocol, thus the protocol does not provide blindness. A detailed description of the protocol is given in Protocol 1.

Theorem 6. *The ABE protocol is* ε*-verifiable where* $\varepsilon = \frac{1}{2^{\mathbb{p}}}$ *(based on Theorem* 4.1 *in [Aharonov et al., 2010] and ignoring the circuit error probability) where* $\mathbb{p}$ *is the maximum degree of polynomials in the polynomial-QECC. The quantum requirement of Alice is to prepare* $O(n)$ *states of* $O(1/log(\varepsilon))$ *entangled qudits of* $O(1/log(\varepsilon))$*-level and the communication requirement is* $O(n)$ *states of* $O(log(1/\varepsilon))$ *entangled qudits sent from Alice to Bob off-line and* $O(n) \times O(log(1/\varepsilon))$ *bits of on-line classical communication between Alice to Bob, where n is the size of the computation.*

The degree $\mathbb{p}$ of polynomials used in the polynomial-QECC will be referred to as the security parameter of the ABE protocol in the rest of this paper.

Protocol 1 ABE protocol (Protocol 4.2 in [Aharonov et al., 2010])

Alice's input. Description of a computation in the Gate Teleportation model based on generalized Toffoli states of dimension d, an odd prime $\geq 2\mathbb{p}+1$, where $\mathbb{p}$ is a parameter of the protocol that gives $\varepsilon = \frac{1}{2^{\mathbb{p}}}$. The input is set to be the Fourier basis state of n qudits: $|+_0\rangle^{\otimes n}$, where $|+_0\rangle = \frac{1}{\sqrt{d}}\sum_a |a\rangle$ but can also be an arbitrary quantum input. The total number of gates is denoted by N and let t be the number of Toffoli gates of the circuit.

Alice's output. The result of measurement of the quantum output of the circuit and a bit indicating if the result is accepted or not.

The protocol

1. Alice has to prepare a number of generalized Toffoli states, equal to the number of Toffoli gates of the circuit.

2. Alice chooses a single random sign key $\mathbf{k} \leftarrow \{\pm 1\}^m$ (where m is the size of a codeword) that will be applied on all encoded inputs (including Toffoli states).

3. Alice chooses a random Pauli key $\mathbf{r} \leftarrow \{0,\ldots,d-1\}^{2(n+3t)m}$ that will be applied on the physical qudits of the encoded input.

4. Alice has to encode both the inputs and the Toffoli states according to the polynomial QECC described in Section 2.2.2, with polynomials of maximum degree $\mathbb{p}$ and parametrized by $\mathbf{k}$. Alice also applies a random generalized Pauli rotation on all $(n+3t)m$ physical qudits parametrized by $\mathbf{r}$.

5. Alice sends all the quantum states to Bob though the quantum channel.

6. Alice and Bob perform the logical operators that correspond to the desired computation on the encoded state. For each application of logical Toffoli gate, Bob sends measurement results to Alice and Alice calculates the actual correction to be performed. She then sends this classical information to Bob who performs the corresponding correction. For this we need a classical on-line channel.

7. Bob measures the output qudits in the computational basis and returns the measurement results to Alice.

8. Alice undoes the Pauli rotation and applies the detection and decoding procedure of the signed polynomial QECC, which checks if the measurement results correspond to $\leq \mathbb{p}$ degree polynomial, and sets an indicator bit accordingly, which shows if she accepts or rejects the result.

Chapter 3

Revisiting Verifiable Blind Quantum Computing

Verifiable Universal Blind Quantum Computing protocol or *FK protocol* is given in Section 3.1 in full detail so that it can be used later in the context of the composite protocol. Moreover, this presentation of the protocol intends to clarify some aspects of the original protocol. In the original paper the protocol is presented (Protocol 8, p.35 [Fitzsimons and Kashefi, 2012] - also outlined in Section 1.3.1 of this thesis) with Bob at the end of the protocol measuring the final output of the computation and returning the result to Alice. In our presentation we assume the more interesting case of Bob returning to Alice the quantum output of the computation, which Alice has to decrypt and decode applying a quantum operation. Therefore, we explicitly describe the encryption and encoding that the final returned state has and what are the requirements for Alice to decrypt and decode it. Our intention while constructing this slight variation of the protocol was to minimize these requirements for Alice. This will be crucial for the composite protocol since the state that ABE protocol requires is a quantum state.

A second aspect to be clarified is the underlying fault tolerant procedure that is used in FK protocol to amplify the verification probability. We provide a description of the protocol which abstracts the fault Tolerant procedure from the rest of the description and is generic enough to admit any FT procedure that has some specific properties that we define. This is similar to what is described in Theorem 7.1 of the original paper, while giving explicitly the description of the protocol involved. A similar approach we take to abstract from the particular graph used to hide the traps.

In Section 3.2 a proof of verifiability of our version of the FK protocol is given, which, as discussed before, takes a different approach at some of the steps of the original

proof and is used in the context of the composition of protocols later.

3.1 Verifiable Blind Quantum Computing with Quantum Output

FK protocol with quantum output is given as Protocol 2. The difference from the original (Protocol 8, p.35 [Fitzsimons and Kashefi, 2012]) is that we consider explicitly the case of quantum output. The qubits that correspond to the output layer of the MBQC pattern are denoted by the set O' (prime here is used to differentiate between the encoded output that Alice receives and the decoded output O that she produces). These qubits are not measured by Bob; thus Alice never sends a measurement angle to Bob that corresponds to these qubits. This makes the need for a pre-rotation in the XY plane obsolete, at least for the needs of blindness. Let us consider each case separately. If the output qubit is a dummy qubit then Alice prepares it in a random computational state (and corrects its neighbours appropriately), as it was the case of the original protocol. If the output qubit is a computational qubit (i.e. non-trap, non-dummy qubit), then it is prepared in state $|+_\theta\rangle$ with $\theta = r_i\pi$ for $r_i \leftarrow_R \{0,1\}$. This is the minimal requirement to maximally mix the state sent to Bob and therefore not reveal any information about the quantum state sent. If the output qubit is a trap then Alice chooses $\theta = \beta_i\frac{\pi}{2} + r_i\pi$ for $r_i, \beta_i \leftarrow_R \{0,1\}$. Effectively this is a $|+\rangle$ or a $|+_i\rangle$ trap state, with extra Z Pauli that maximally mixes it, as in the previous case. The need for the two different types of traps will become evident in the course of the proof of verification, where we need to be able to detect all Pauli attacks on the output, and therefore have a minimal selection of state that will be affected by the attack and therefore detected. Note that we could choose the output computational qubits and the traps to be of the set of 8 angles as it is the case for the non-output qubits and still have the same properties (since our set of angles is a sub-set of the 8 angles). But in this case, Alice would need to undo these pre-rotations and to achieve this she would need operators that are outside of the Clifford group and therefore considered harder to implement.

While each step of the Protocol 2 is expanded to all the technical detail, at the same time we intentionally abstract from the specific FT procedure used for the amplification of ε and from the specific graph used to get the uniform placement of the traps inside the computation without interrupting the flow. In particular, for the FT procedure we ask to be able to detect a fixed number of errors at any stage of the computation (including

encoding, logical computation) and also to be compatible with the operations supported by blindness (e.g. adaptive measurements only on the XY plane). For the graph we require to have the property that we can divide its vertices to sets where each vertex can be with equal probability a computational qubit or a trap. This implies that the distribution of the traps is independent between the different sets. In fact, we can only give graphs that approximate this behaviour, in the sense that we have this behaviour only when we consider sub-graphs that are sufficient nevertheless for out attack analysis. This complication together with the specific function of the dummies is discussed in more detail in the Section 3.1.2 and in the proof of verifiability that follows in Section 3.2.

Protocol 2 Qubit Verifiable Universal Blind Quantum Computation (VUBQC) Protocol with quantum output (FK protocol)

Alice's input:

- Description of a unitary computation U in the MBQC model (or equivalent) using a convenient underlying open graph state (G,I,O). The computation is represented, for any vertex $i \in G \setminus O$, as a measurement angle φ_i (together with the set of X-dependences D_i^X and Z-dependences D_i^Z and a fixed partial order of measuring depending on the graph structure). The input is set to the state of n qubits: $|+\rangle^{\otimes n}$. Protocol can be extended to admit quantum input by applying techniques described in the FK protocol.

Alice's output:

- A system that contains the quantum output of the computation ($U|+\rangle^{\otimes n}$) and a bit to indicate of Alice has accepted the output of the computation.

The protocol

1. Preprocessing 1: Alice translates the computation to a Fault Tolerant (FT) MBQC pattern that can correct Pauli errors on $\mathbb{p}$ qubits. Let the updated open graph be (G',I',O'), where $|G| = m'$ and $|I'| = |O'| = n'$. An example is the RHG topological code of size $m' = O(m\log^3 m) \times O(\log 1/\varepsilon)$ where ε is the verification error.

Protocol 2 Cont'd

2. Preprocessing 2: Alice embeds the encoded computation pattern into a suitable graph which has the following property: There exist ordered X-Y plane measurements that implement the encoded computation and each computational qubit (including bridges) belongs to a constant size subset of qubits S_i in which a trap can be selected uniformly at any position. Break operations are implemented by including computational basis qubits (dummies). The total number of qubits of the final graph G'' is N, with $|I''| = |O''| = n''$ consisting of all S_i that contain qubits of the O' system and the necessary traps/dummies. An example (up to the independent errors assumption) is the dotted complete graph of size $N = O(m'^2)$.

3. Alice prepares the rotated qubits. For $i = 1$ to N:

   (a) If qubit is a dummy: prepares $|d_i\rangle$, $d_i \leftarrow_R \{0,1\}$, otherwise:

   (b) If qubit is not in O'': prepares $\prod_{j \in N_G(i) \cap D} Z^{d_j} |+_{\theta_i}\rangle$, where $|+_{\theta_i}\rangle \equiv R_Z(\theta_i)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $\theta_i \leftarrow_R \{0, \frac{\pi}{4}, \frac{\pi}{2}, \ldots, \frac{7\pi}{4}\}$.

   (c) If qubit is in O'' and is not a trap: Same as previous step but with $\theta_i = r_i\pi$ for $r_i \leftarrow_R \{0,1\}$.

   (d) If qubit is in O'' and is a trap: Same as previous step but with $\theta = \beta_i\frac{\pi}{2} + r_i\pi$ for $r_i, \beta_i \leftarrow_R \{0,1\}$.

   (e) She sends the qubit to Bob

4. Bob entangles the states according to the graph state by applying cZ gates.

Protocol 2 Cont'd

5. Bob performs the rest of the computation using classical help from Alice. For i which ranges over all qubits (respecting the order given by the flow), except the qubits belonging to O'':

   (a) Alice computes the actual measurement angle ϕ'_i using the dependences and the previous measurement results ($\phi'_i = 0$ for dummy qudits).

   (b) Alice chooses $r_i \leftarrow_R \{0,1\}$ and computes $\delta_i = \phi'_i + \theta_i + r_i\pi$ (for the case of dummies a random θ_i is chosen).

   (c) Alice transmits δ_i to Bob.

   (d) Bob performs measurement $M_i^{\delta_i}$ on qubit i. Measurements on measurement angle δ_i correspond to projective measurements on basis $\{Z^j R_Z(\delta_i)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\}_{j=0}^{1}$.

   (e) Bob transmits the result to Alice.

   (f) Alice corrects the result by adding $r_i \mod 2$.

6. Bob returns the qubits of O'' to Alice.

7. Alice measures traps and applies the final Pauli corrections on qubits of O'.

8. Alice applies the decoding procedure of the FT encoding to produce actual output O.

9. Alice sets her indicator bit to accept if all trap tests where positive.

In the following Subsections 3.1.1 and 3.1.2 we will give a specific example of a FT procedure and graph to be used, which are the same used in the original FK: Raussendorf-Harrington-Goyal (RHG) topological FT [Raussendorf et al., 2007] and dotted-complete graph introduced for the FK protocol. Specifically, without going into the proof which is given in full in the original paper, we explain how the RHG code has the required by the protocol properties. Also, we explain the independence property of the trap placement in the case of the dotted complete and how this behaviour is only achieved if the idea of independently detectable errors is introduced. In this way, we aim to clarify the role of the specific parts of the original paper on VUBQC so that they can be used without problem in the rest of this thesis.

3.1.1 The Role of Fault Tolerance

The condition, stated in the protocol, that is required from the fault tolerant procedure for the probability amplification is that, if there is a Pauli operator attack on $\leq \mathbb{p}$ qubits, an error is detected during the detection/decoding procedure of Alice. For Bob to succeed in his cheating, he has to attack on more than $\mathbb{p}$ positions in the graph, which means that he is more likely to hit a trap, therefore the probability of corrupting the computation and not getting caught becomes smaller (will be proven exponentially small on $\mathbb{p}$).

In the FK Protocol the FT procedure used is the topological encoding of Raussendorf-Harrington-Goyal (RHG) proposed in [Raussendorf et al., 2007] (also explained in more detail in [Fowler and Goyal, 2008]). The computation runs through a series of topologically protected cX, preparations and measurements in the X and Z eigenbasis together with state distillation of XY-plane states $|Y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ and $|A\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{\pi}{4}i}|1\rangle)$ to complete a universal set of gates. The resource state is the 3-dimensional lattice state, usually called RHG or Raussendorf lattice, depicted in figure 3.1.

In the original paper of the FK protocol, in Lemma 5, a particular selection of parameters is given for the RHG FT procedure that the condition of the protocol regarding FT is satisfied. Special care had to be taken that an error, in both the topologically protected gates and during the distillation procedure of $|Y\rangle$ and $|A\rangle$ states, that is under the required threshold will be detected.

When writing the FT procedure as a MBQC pattern, special care has to be taken so that we apply only operations allowed by MBQC. States $|Y\rangle$ and $|A\rangle$ are handled as quantum inputs (therefore a random rotation around the XY plane by θ is added for

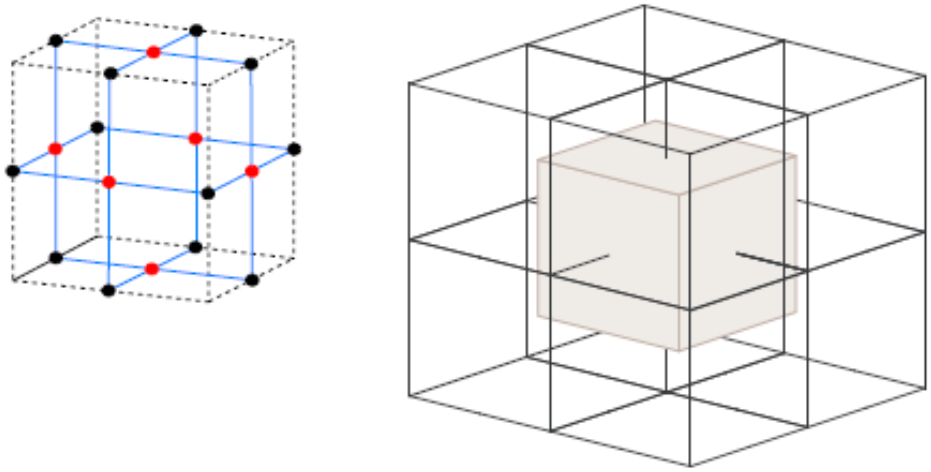

Figure 3.1: RHG prime and dual lattice (taken from [Fowler and Goyal, 2008]). The second picture shows how the dual lattice is placed between 4 prime lattice blocks. Logical qubits are represented by primal or dual defects (qubits measured in the Z basis). Single qubit logical operations are chains connecting or rings encircling defects, while cX can be produced by braiding primal and dual defects.

blindness) and their distillation circuit is part of the blind pattern that Bob executes. Since all the qubits are sent from Alice to Bob at the beginning of the protocol the size of the computation is fixed from the start. Distillation is a stochastic procedure and therefore fixing the maximum size of operations allowed will have implications. In the ideal case, when Alice prepares perfect states and Bob applies perfect operations the correctness of the protocol will not be affected. In the ideal case again, soundness will not be affected also: Since the only part of the distillation procedure used in this case is the error detection procedure, when Bob corrupts the states the protocol will abort immediately (except for the case Bob's footprint is large enough which means that the traps will be triggered with high probability). However, in a realistic noisy scenario the correctness of the protocol will be affected, since there is a probability that after the maximum allowed distillation rounds the state is still corrupted by noise. Soundness will not be affected for the same reason as in the ideal case.

The RHG fault tolerant procedure requires Z measurements that are not allowed in blind MBQC, but can be simulated by dummy qubits as explained in Section 1.3.1 and in more detail later in 3.1.2. The problem is that there are some Z measured positions that are not known from the beginning of the protocol (when Alice creates the pattern and sends the dummy qubits to Bob) but depend on measurements during Bob's

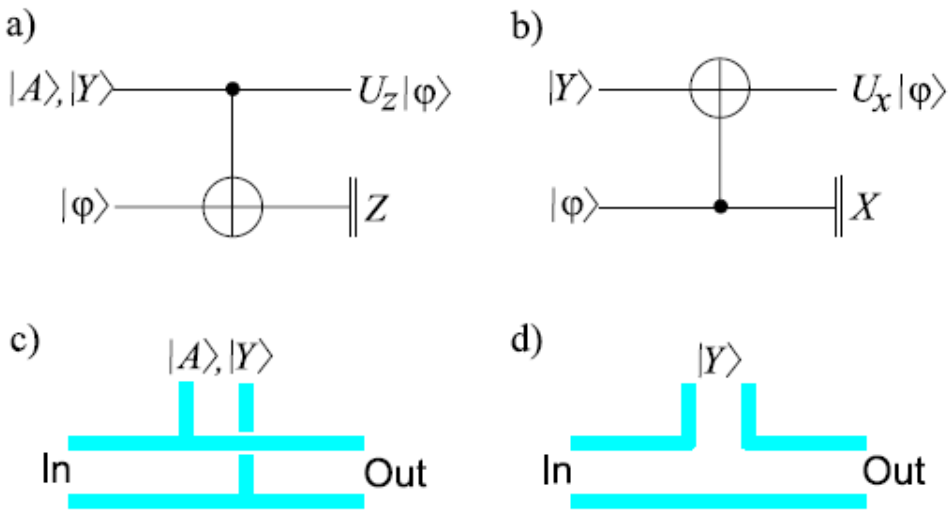

Figure 3.2: Implementation of U_z and U_x gates in the RHG scheme (taken from [Raussendorf et al., 2007]). $U_z(\pi/8)$, which is T gate in our context, consumes a $|A\rangle$ state and $U_z(\pi/4)$, which is S gate in our context, consumes a $|Y\rangle$ state (a). Accordingly $U_x(\pi/4)$ is a rotation around the X axis (Clifford gate) which consumes a $|Y\rangle$ state (b). Then the circuits are translated to defect chains in the RHG lattice (c) and (d), which are implemented blindly in FK. All operations depicted are up to Pauli corrections.

runtime. These adaptive Z measurements are needed, in particular, for the T gate $(\pi/8)$ (Section 3 in [Raussendorf et al., 2007], also Figure 3.2 here). T gate is implemented by teleporting state $|A\rangle$ and, depending on the measurement outcome, might require an extra S gate correction which, in turn, is implemented by measuring a $|Y\rangle$ state in the Z basis (and some Pauli corrections). To overcome this obstacle, the authors of the original FK protocol propose a fixed correction step after each T gate. This is the circuit of Figure 3.2 (a), which can selectively implement an S gate or an identity depending on whether we inject state $|Y\rangle$ or $|+\rangle$. The selection of the state can be seen as an update on the θ angle that encrypts the corresponding MBQC input state, and can be done by classical on-line computation on Alice's side. An alternative to this approach, to tackle the problem of adaptive Z measurements, is to use the decorated lattice proposed in [Morimae and Fujii, 2012].

3.1.2 Dotted-Complete Graph and Trap Independence

The second requirement has to do with the selection of the graph state on which we embed the computation. This graph needs to be generic so that it hides the underlying computation (only leaking an upper bound on the size) and convenient for placing the

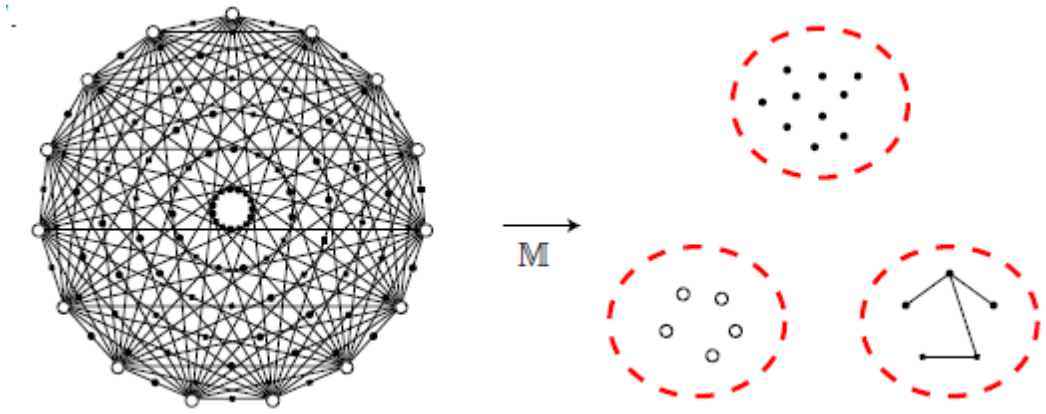

Figure 3.3: Dotted-complete graph as an example of resource state (taken from [Fitzsimons and Kashefi, 2012]). There are two types of vertices, the ones coming from a complete graph, $P(G)$, and the ones that are injected on the edges, $A(G)$. By appropriate measurements and dummy qubit injections we can carve a computational graph, a $P(G)$-vertice graph and a $A(G)$-vertice graph.

traps at random positions without interrupting the flow. More specifically we ask that each qubit of the computation belongs to a constant size subset of qubits S_i (which we can reveal to Bob without breaking the proof of verification) in which a trap can be at any position with uniform probability. In the original paper the dotted-complete graph is given to satisfy this condition (up to a subtlety that we will examine right after).

The dotted-complete graph, depicted in Figure 3.3, has two types of vertices: $A(G)$, with degree 2, and $P(G)$, with degree $n-1$, where n is the size of the graph. This graph has the proven (in [Fitzsimons and Kashefi, 2012]) property that, if of large enough dimension, by measuring the $A(G)$ vertices we can produce a graph suitable for doing universal measurement based quantum computation. At the same time we can get a sub-graph of isolated traps on the $P(G)$ vertices (called *white traps*). By measuring qubits on $P(G)$ vertices we can get isolated traps on the $A(G)$ vertices (called *black traps*).

To understand the procedure of carving out a sub-graph by XY plane measurements we need to explain two types of operations. The first type is bridge operators: Y basis measurements followed by a Pauli correction on neighbours depending on the measurement result. In particular, in every vertex i of degree 2 that we want to remove and join its neighbours by an new edge, we inject state $|+_{\theta_i}\rangle$, as it was the case for the normal qubits, and measure them in the $\{|+_{\pi/2+\theta_i+r_i\pi}\rangle, |-_{\pi/2+\theta_i+r_i\pi}\rangle\}$ basis (i.e. $\phi = \pi/2$ in this case and there are no corrections). Let s_i being the corrected outcome

of this measurement. Depending on s_i, an extra S (s_i=0) or $S^\dagger$ (s_i=1) correction has to be applied in calculating the measurement angle of each of the neighbours of i. Note that a Pauli attack on a bridging qubit will give a wrong correction to its neighbours and this, being equivalent to an attack on the computational graph, will be detected by the FT procedure, if the threshold of the total number of errors is low enough.

The second type is break operators (which have already been outlined in previous sections): In every vertex i we want to remove (together with the entangling operation with its neighbours), instead of injecting a $|+_{\theta_i}\rangle$ state in the graph at position i, as it was the case for normal vertices, we inject *dummy* state $|d_i\rangle$ for d_i chosen independently and uniformly at random from $\{0,1\}$ and measure them in the $\{|+_{\theta_i+r_i\pi}\rangle, |-_{\theta_i+r_i\pi}\rangle\}$ basis (i.e. $\phi = 0$ in this case and there are no corrections). Note that θ_i can be an arbitrary angle since the measurement does not have any effect on the rest of the system and the measurement outcome is discarded, but we choose θ_i to be uniform from the set of eight angles so that we randomize δ_i (and therefore the position of the dummies). Also, before sending each state to the prover, we apply on each of its neighbours Pauli operation Z^{d_i} so that it will cancel the effect that the prover's entangling operation will have on that neighbour.

3.1.2.1 Independently detectable errors

The subtle point, when examining the possibilities for dividing all qubits in sets S_i and positioning a trap among the qubits that belong to each set, is that there will be some dependence in the position of the trap between some particular sets. In particular, selecting a $P(G)$ qubit to be white trap, means that all its $A(G)$ neighbours are excluded from being black traps (they have to be dummies to isolate the white traps). Also, after one fixes the positions of the computational qubits on the $P(G)$ vertices, this fixes the positions of the bridge and break qubits on the $A(G)$ vertices that create the required computational sub-graph, which means that these qubits cannot be black traps. A new concept, the idea of *independently detectable errors* is introduced in the original FK paper, to deal with this problem. In particular, one needs to consider not the whole graph in the trap analysis, but only a subset of vertices that correspond to a set of errors that are independently detected by the FT procedure. The independently detectable errors are the Pauli errors that can corrupt the computation and each one has a distinct effect on the syndrome measurements, as opposed to errors that are detected simultaneously by the same error vector. A formal definition of independently detectable errors in given in [Fitzsimons and Kashefi, 2012], Definition 7.5. In short these are the tensor

products of Pauli operators that: (1) flip measurement results on the measured qubits of corrupt output qubits and (2) for an attack on qubit $i \in A(G)$ we map this attack as two attacks on each of its $P(G)$ neighbours $j,k \in N_G(i)$ and consider a subset of tensor attack elements that (after doing this mapping) contains each $P(G)$ vertex only once. For the FT procedure used there is a bound on the number of independently errors that can be perfectly corrected by the code, which is proven (in [Fitzsimons and Kashefi, 2012], Corollary 7.6) for the RHG encoding to be $\mathbb{p}' = 2\mathbb{p}/5$ (the number is smaller than $\mathbb{p}$ because some single qubit attacks effectively correspond to two qubit attacks as we saw).

For example, when there is an error on qubit $i \in A(G)$ and, at the same time, on its two $P(G)$ neighbours $j,k \in N_G(i)$, we will consider in our independently detectable attacks qubit i or qubits j,k but not both. The interesting property of tensor products of attacks that cause independently detectable errors is that the probability of coinciding with a trap is independent in each position (in our example we avoid considering the probabilities of having traps in position i, j, k which will be dependent on each other). Note that this analysis for the attacks is done for the whole graph G and before we partition it into the computational, dummy and trap sub-graphs. If we can detect a certain number of independently detectable attacks in total, we can certainly detect the smaller number of attacks that occur on the computational sub-graph. We should keep this intuition of the corresponding part of the verification proof.

3.2 A Refined Proof of Verifiability

A proof of verification is given here, that differs from the original proof in certain aspects, the importance of which will be progressively understood in the sections to follow. The main extra property of this proof is that we are able to write the state even if we interrupt the protocol at some specific intermediate stages and ask Bob to return the output of the computation, excluding traps and dummies, and have Alice decrypt and decode this output. In the context of the composite protocol this property is crucial since the circuit of ABE protocol has to apply on the computational qubits, while Alice has no access to any output traps. Our proof is also more modular than the original one. Also from our proof we get that the parallel composition of FK protocols is possible as a corollary. Finally, since we show that the trap qubit remains disentangled for any possible deviation of Bob, we provide intuition on the concept of independent security [Dunjko et al., 2014], where the detection procedure (that depends only on the trap

system) is independent on the particular computation that is delegated.

Let us restate the main theorem, which we will prove in this section:

Theorem 1. *Protocol 2 is perfectly blind and* ε*-verifiable where* $\varepsilon = \frac{1}{c^{\mathbb{p}_1}}$ *where* $\mathbb{p}_1$ *is the number of errors detectable by the QECC used for amplification and some constant* $c > 1$*. The quantum requirement of Alice is to prepare and send single qubit states and apply the Clifford decoding circuit for the QECC used, together with Pauli measurements. The communication requirement is* $\widetilde{O}(n^2) \times O(log(1/\varepsilon))$ *separable single qubit states sent from Alice to Bob off-line and* $\widetilde{O}(n^2) \times O(log(1/\varepsilon))$ *bits of on-line classical communication between Alice to Bob, where n is the size of the computation.*

Proof of blindness is the same as the proof of Theorem 12 in [Fitzsimons and Kashefi, 2012]. The proof of verifiability (originally the proof of Theorem 13 in [Fitzsimons and Kashefi, 2012], but re-proven here using the rewritten version of the protocol and the new proof technique) will roughly follow the following steps:

- Express the state at the end of the protocol as a function of the correct operation and the most generic attack.
- Decompose the attack to the Pauli basis.
- Average over the random parameters of the Pauli encoding to reduce the attack to a convex combination of Pauli attacks.
- Break the sum over the Pauli attacks to those who are detected by the FT procedure and those who are not detected and for the latter calculate the probability of not hitting a trap, averaging over all trap positions.

The third step is the main difference from the original proof and is the one that allows us to study the attack on the computational output separately from the attack on the output traps. The technique we use to reduce the attack to a convex combination of Pauli operators by averaging over the quantum one-time-pad is often called the *Pauli twirl*. It is useful in the context of unitary t-designs [Dankert et al., 2009] and in other verification proofs [Aharonov et al., 2010]. In our case we average over the Pauli Z rotation that comes from the random r parameters of the protocol and commutes with the honest computation, to twirl the attack operator (a random Pauli X rotation is not needed in the protocol because of the special structure of the MBQC graph state, as we see in the proof).

Also, in the fourth step we clarify the role of Alice's decoding of the returned state at the end of the protocol.

This proof has also the structure that helps proving Corollary 1 that states that the parallel composition of ε-verifiable FK protocols with quantum output is also an ε-verifiable protocol. Proof intuition can be gained from observing Equation 3.20 later in this proof and the fact that we can write the attack as a conjugation by a tensor product of local operators (Pauli+identity). The full proof is part of the proof of the composite protocol, where many runs of the FK protocol are run in parallel to verifiably prepare the encoded states needed for the ABE verification.

3.2.1 Expanding the Prover's Operation

Single index notation is followed to enumerate the qubits participating in the graph where N is the total number of qubits. Also, remember that n'' is the number of qubits that are returned from Bob to Alice at his last step in the protocol.

To represent the state of the protocol it is convenient to introduce first some extra notation. Vector ν is used to represent all random secret parameters chosen by Alice throughout the execution of protocol, including $\beta = \{\beta_i\}$, $r = \{r_i\}$, $\theta = \{\theta_i\}$, $d = \{d_i\}$ and positions of traps $t = \{t_i\}$. Parameter $p(\nu)$ gives the probability of a particular choice of random secret parameters. Summing over a vector (e.g. $\sum_r$) means that we sum over all possible choices for the elements of that vector (e.g. all possible bit-strings of size N for r).

For convenience we name the subsystems of the joint Alice-Bob system:

- System $\mathcal{M}$ is the union of qubits $G' \setminus O'$ and the necessary bridge qubits which are introduced when embedding G' in G'' (and are also measured by Bob). This system will be called the system of the measured computational qubits.
- System D contains all dummy qubits.
- System T contains all trap qubits.
- System O'' is the quantum system returned by Bob to Alice.
- System Δ is the system of the measurement angles send by Alice to Bob, where each angle is represented by three qubits in the computational basis.
- System B is Bob's private system, assumed to be initially in the blank state.

Each measurement performed at Step 5d of the protocol is mathematically decomposed into a unitary part and a Pauli Z projective measurement. Without loss of generality we can represent any dishonest behaviour of Bob at any step as applying the correct unitary operator and then an arbitrary unitary *attack* operator that applies on all the states received up to this step and his own private subsystem.

The output of Alice after all steps of the protocol are applied, averaged over all random parameters ν and for all possible measurement outcomes $b = \{b_i\}$ with the corresponding probability, is:

$$\begin{aligned}\rho_{out} = \\ Tr_{B,\mathcal{M},\Delta,D}(\sum_b \sum_\nu p(\nu)\mathcal{D}(C|b_{N-n''}\rangle\langle b_{N-n''}|U_{N-n''}H_{N-n''}R_{N-n''}(|\delta_{N-n''}\rangle\langle\delta_{N-n''}| \\ \otimes\ldots\otimes|b_1\rangle\langle b_1|U_1H_1R_1(|\delta_1\rangle\langle\delta_1|\otimes E_G|M\rangle\langle M|\otimes|0\rangle\langle 0|^{\otimes B}E_G)R_1^\dagger H_1U_1^\dagger|b_1\rangle\langle b_1| \\ \ldots)R_{N-n''}^\dagger H_{N-n''}U_{N-n''}^\dagger|b_{N-n''}\rangle\langle b_{N-n''}|C^\dagger)) \quad (3.1)\end{aligned}$$

where:

- $|M(\theta,d,t)\rangle$ is the state that Bob receives after Step 3 of the protocol (input, auxiliary qubits, dummy qubits).
- E_G is the global entangling operator that Bob applies at Step 4 of the protocol.
- $|\delta_i(\theta,r,b)\rangle\langle\delta_i(\theta,r,b)|$ is the density matrix of the measurement angle that Alice sends to Bob at iteration i of Step 5 of the protocol, which is composed of three computational basis states ($\delta_i(\theta_i,r,b)$ can take 8 possible values). These states will be used by Bob to control his rotations on the graph state. In the calculation of δ's by Alice there is an implicit error correction procedure that comes from the FT encoding of the computation (this will be used as an extra property at a later stage of the proof).
- $R_i(\delta_i)$ is a controlled unitary rotation of qubit i around z-axis by angle δ_i.
- H_i is a Hadamard gate applied on qubit i.
- U_i is the unitary attack of Bob with index i and applies on qubits $\geq i$, private Bob's system and all measurement angles.
- $C(r,b)$ includes the Pauli correction $C_{O'}$ that Alice performs to system O' according to the flow dependences and the post-processing on the trap system:

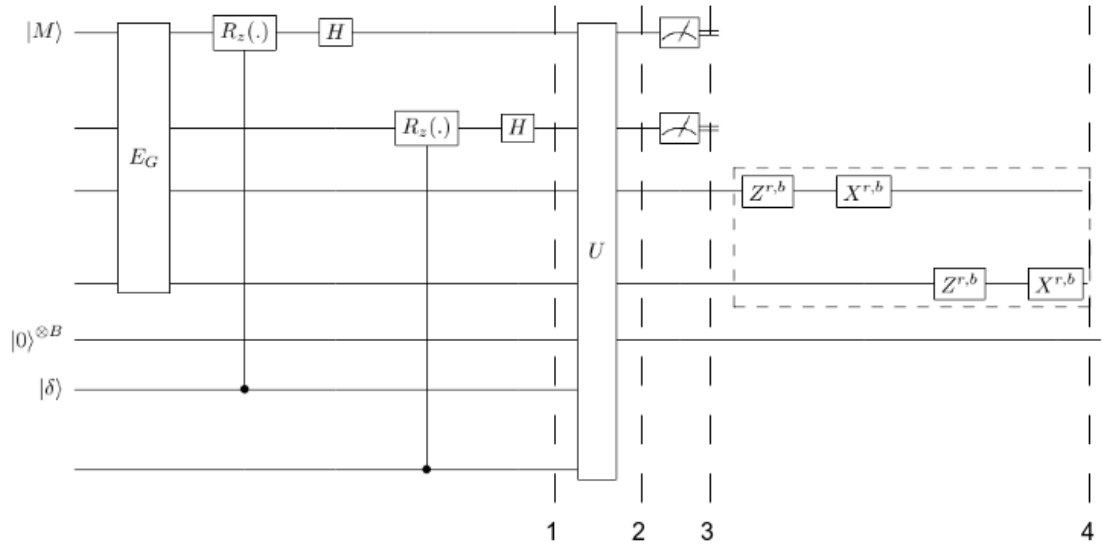

Figure 3.4: Analysis of attacks in the qubit VUBQC with quantum output protocol. Part inside dashed box is applied by Alice. Dummy qubits and trap qubits on returned system are discarded and measured respectively by Alice.

corrections $H_{t_i} S_{t_i}^{\dagger \beta_{t_i}} Z_{t_i}^{r_{t_i}}$ and a Pauli Z measurement for output traps t_i and corrections $Z_{t_i}^{r_{t_i}}$ for measurement results sent from Bob for non-output traps t_i.

- $\mathcal{D}$ is a CPTP-map that represents the decoding procedure of Alice for the quantum output system O'.

We commute to the left (right) all projectors $|b_i\rangle\langle b_i|$ of the left (right) hand side of $|M\rangle\langle M|$ by observing that the operators they commute with do not apply on the same qubits. We commute to the left (right), and just before the $|b_i\rangle\langle b_i|$'s, all attack unitary operators U_i of the left (right) hand side of $|M\rangle\langle M|$ by observing that the measurement angle density matrices $|\delta_j\rangle\langle\delta_j|$ that they commute with have indices $j \geq i$ and therefore the attack does not apply on them. The commutation of the attack operators with the rest of the operators will change the attack operators to different ones but these crucially do not depend on any of the protocol parameters. We merge all commuted and updated attack operators into a new unitary U that applies on the whole of the system just before the measurements. We commute to the right (left) all measurement angle density matrices $|\delta_i\rangle\langle\delta_i|$ of the left (right) hand side of $|M\rangle\langle M|$ by observing that the only operators that apply on the measurement angles are the controlled rotation operators R_j but always have index $j \leq i$. We get (see also Figure 3.4):

$$\rho_{out} = Tr_{B,\mathcal{M},\Delta,D}(\sum_b \sum_\nu p(\nu)\mathcal{D}(C|b\rangle\langle b|U(H_{N-n''}R_{N-n''}(.)\dots H_1R_1(.)E_G(|M\rangle\langle M|\otimes|\delta\rangle\langle\delta|)E_G R_1(.)^\dagger H_1\dots R_{N-n''}(.)^\dagger H_{N-n''}\otimes|0\rangle\langle 0|^{\otimes B})U^\dagger|b\rangle\langle b|C^\dagger)) \quad (3.2)$$

where U is a unitary attack operator that is chosen by Bob and applies on all graph qubits, private Bob's system and all measurement angles.

The trap system is not entangled with the rest when the honest computation is applied. We can separate the terms that apply on the trap system. Applying the entangling operators between the traps and their neighbours, which are always dummy qubits does not have any effect on the trap other than undoing the pre-rotation of the trap (in the case that the dummy is selected to be $|1\rangle$). Moreover, applying $H_{t_i}R_{t_i}(\delta_{t_i})$ on each of the traps at positions t_i that belong to the measured system, where $\delta_{t_i} = \theta_{t_i} + r_{t_i}\pi$, results in getting states $|r_{t_i}\rangle$.

Therefore, it holds:

$$\mathcal{P}(|M\rangle\langle M|\otimes|\delta\rangle\langle\delta|)\mathcal{P}^\dagger = \mathcal{P}'(|M'\rangle\langle M'|\otimes|\delta\rangle\langle\delta|)\mathcal{P}'^\dagger\bigotimes_i|\eta_{t_i}\rangle\langle\eta_{t_i}| \quad (3.3)$$

where

- $|\eta_{t_i}\rangle \equiv |r_{t_i}\rangle$ if t_i points to a trap measured by Bob and is $Z^{r_i}S^{\beta_i}|+_{t_i}\rangle$ if t_i points to a trap measured by Alice.
- $\mathcal{P} = H_{N-n'}R_{N-n'}(.)\dots H_{t_i}R_{t_i}(.)\dots H_1R_1(.)E_G$ is the correct unitary operation in Figure 3.4.
- $\mathcal{P}' = H_{N-n'}R_{N-n'}(.)\dots H_1R_1(.)E'_G$, which is produced from $\mathcal{P}$ by omitting all operations applied on traps (E'_G is produced from E_G by omitting all entangling operators between traps and their neighbours)
- $|M'\rangle$ is produced from $|M\rangle$ by omitting the trap qubits.

3.2.2 Decomposing the Attack

In the next step we express the output state with Bob's private system traced out and his attack decomposed to the Pauli basis. We use the following general property:

For any unitary U and n-qubit state ρ, there holds:

$$Tr_B(U(\rho \otimes |0\rangle\langle 0|^B)U^\dagger) = \sum_{k,u,u'} a_{k,u} a^*_{k,u'} P_u \rho P_{u'} \tag{3.4}$$

where

- $a_{k,u}$ are complex numbers, with $\sum_{k,u} |a_{k,u}|^2 = 1$.
- P_u (and $P_{u'}$) ranges over all possible tensor products of Pauli+Identity operators (Pauli basis) on n qubits.

One way to prove the above expression is the following: by tracing out B, the effect of unitary U is a CPTP-map on the remaining system. This CPTP-map is decomposed into a sum of conjugations by (Kraus) operators E_k. Each operator E_k and its Hermitian conjugate $E_k^\dagger$ are decomposed into the Pauli basis. The equation $\sum_{k,u} |a_{k,u}|^2 = 1$ derives from the completeness relation between the operators: $\sum_k E_k^\dagger E_k = I$.

In the following we apply Equation 3.4 in order to trace out B from the state in Equation 3.2 and at the same time apply Equation 3.3 to separate the traps from the rest of the system. We use notation $P_{u|i}$ to point to the i-th tensor element (single qubit Pauli operator or identity) of the operator P_u (tensor product of Pauli or identity operators).

$$\begin{aligned}\rho_{out} = Tr_{\mathcal{M},\Delta,D}(\sum_b \sum_\nu p(\nu) \sum_{k,u,u'} a_{k,u} a^*_{k,u'} \mathcal{D}(C_{O'}|b'\rangle\langle b'|P_{u|i:\forall j, i\neq t_j} \\ (\mathcal{P}'(|M'\rangle\langle M'| \otimes |\delta\rangle\langle\delta|)\mathcal{P}'^\dagger) P_{u'|i:\forall j, i\neq t_j} |b'\rangle\langle b'|C^\dagger_{O'}) \\ \bigotimes_i Q_{t_i} P_{u|t_i} |\eta_{t_i}\rangle\langle\eta_{t_i}| P_{u'|t_i} Q^\dagger_{t_i})\end{aligned} \tag{3.5}$$

where

- $a_{k,u}$ are complex numbers, with $\sum_{k,u} |a_{k,u}|^2 = 1$.
- P_u (and $P_{u'}$) ranges over all possible tensor products of Pauli+Identity operators (Pauli basis) applying on the resource state G'' and measurement angle system Δ.
- b' is the vector that is generated from b by removing elements $\{b_{t_i}\}$.
- $Q_{t_i} \equiv X_{t_i}^{r_{t_i}} |b_{t_i}\rangle\langle b_{t_i}|$ if $t_i \notin O''$ and $Q_{t_i} \equiv |b_{t_i}\rangle\langle b_{t_i}| H_{t_i} S_{t_i}^{\dagger \beta_{t_i}} Z_{t_i}^{r_{t_i}}$ if $t_i \in O''$, represents the measurements and corrections on the trap.

Trivially we can trace over terms $|b'\rangle\langle b'|$:

$$\rho_{out} = Tr_{\Delta,D}(\sum_{b}\sum_{\nu} p(\nu) \sum_{k,u,u'} a_{k,u}a^{*}_{k,u'}\mathcal{D}(C_{O'}\langle b'|P_{u|i:\forall j,i\neq t_j} (\mathcal{P}'(|M'\rangle\langle M'|\otimes|\delta\rangle\langle\delta|)\mathcal{P}'^{\dagger})P_{u'|i:\forall j,i\neq t_j}|b'\rangle C^{\dagger}_{O'}) \bigotimes_{i} Q_{t_i}P_{u|t_i}|\eta_{t_i}\rangle\langle\eta_{t_i}|P_{u'|t_i}Q^{\dagger}_{t_i}) \tag{3.6}$$

3.2.3 Reducing the Attacks to Pauli

In this part of the proof the attack of Bob will be reduced to a convex combination of Pauli+identity operators, by summing over all random parameters except the random trap positions. This means cancelling the cross terms in Equation 3.6: each term of the sum containing P_u and $P_{u'}$ where $u \neq u'$ has to be eliminated so that only terms with $P_u = P_{u'} = P_u^{\dagger}$ remain - a mixture of conjugations by Pauli+identity operators, each with real coefficient $a_{k,u}a^{*}_{k,u} = |a_{k,u}|^2$, where $\sum_{k,u}|a_{k,u}|^2 = 1$. Note that in the original proof of the FK protocol they also eliminate the cross attack terms at some stage, but to achieve this they reduce the state of Alice (except the trap) to identity, which is undesired here - the reasons will be made clear later.

The process of eliminating the cross terms will take a few steps in which the following lemma, proven in [Dankert et al., 2009], will be useful:

Lemma 2. *(Operator Pauli Twirling)*

$$\sum_{i} P_iQP_i\rho P_iQ'P_i = 0, \text{ if } Q \neq Q' \tag{3.7}$$

where ρ *is a matrix, Q, Q′ are two arbitrary tensor products of Pauli+identity operators of the same dimension, and* $\{P_i\}$ *is the set of all tensor products of Pauli+identity operators of dimension same as Q and Q′.*

Eliminating cross-terms on system Δ*:* By observing that $R_i(.)(|\delta_i\rangle\otimes|\psi\rangle_i)$, where R_i is controlled by the three qubit state $|\delta_i\rangle$ and applies on state $|\psi\rangle_i$, is equivalent mathematically to $|\delta_i\rangle\otimes R_i(\delta_i)|\psi\rangle$, where $R_i(\delta_i)$ is a single qubit gate (since $|\delta_i\rangle$ is by definition always a classical state), we can rewrite operator $\mathcal{P}'$ as:

$$\mathcal{P}' = H_{N-n''}R_{N-n''}(\delta_{N-n''})\ldots H_1R_1(\delta_1)E'_G$$

Or,

$$
\begin{aligned}
&= H_{N-n''}R_{N-n''}(\phi'_{N-n''} + \theta_{N-n''} + r_{N-n''}\pi)\dots H_1R_1(\phi'_1 + \theta_1 + r_1\pi)E'_G \\
&= H_{N-n'}R_{N-n''}(\phi'_{N-n''} + r_{N-n''}\pi)\dots H_1R_1(\phi'_1 + r_1\pi)E'_G \\
&R_{N-n''}(\theta_{N-n''})\dots R_{N-n''}(\theta_i) \\
&\equiv \mathcal{P}''R_{N-n''}(\theta_{N-n''})\dots R_{N-n''}(\theta_i) \qquad (3.8)
\end{aligned}
$$

Operations $R_{N-n''}(\theta_{N-n''})\dots R_{N-n''}(\theta_i)$ applied to state $|M'\rangle$ cancel the pre-rotations by the θ's (or have no effect in the case of dummies) and the state is updated to a state which does not depend on θ, which we will denote by $|M''\rangle$.

Now that we have eliminated all dependences on θ except from system Δ (states $|\delta\rangle$), we can sum over θ to get the maximally mixed state for system Δ and in Equation 3.6 the terms $P_{u|\Delta}\frac{1}{2^{|\Delta|}}IP_{u'|\Delta}$. Remember that the Δ system is traced out. Terms where $P_{u|\Delta} = P_{u'|\Delta}$ have trace 1 and where $P_{u|\Delta} \neq P_{u'|\Delta}$ have trace 0, since all Pauli operators except identity are traceless. Therefore, we can ignore the attack cross terms applied on the system which holds the measurement angles.

Eliminating cross-terms on dummy system D: Part of the entangling E'_G is applied between the dummy qubits and their computational neighbours with the effect of cancelling the pre-rotation of the latter when $d_i = 1$ (the dummy is in state $|1\rangle$). Let us denote E''_G the remaining entangling operators. The remaining unitary part of the protocol applied on the dummy system is single qubit unitary operators applied on each dummy $i \in D$:

$$
\sum_{d_i,r_i} H_iR_i(r_i\pi)|d_i\rangle\langle d_i|R_i(r_i\pi)H_i) = \frac{1}{2}I \qquad (3.9)
$$

Again, $Tr(P_{u|D}\frac{1}{2^{|D|}}IP_{u'|D})$ is 1 for $P_{u|D} = P_{u'|D}$ and 0 for $P_{u|D} \neq P_{u'|D}$ so we can ignore the cross terms of the attack applying on the dummy qubits. Let $|M^{(3)}\rangle$ denote state $|M''\rangle$ after the entangling between dummies and their computational neighbours is applied and omitting the dummy qubits.

Eliminating the cross-terms on computational system $\mathcal{M} \times O'$: The next step is an iteration over all qubits i of the computational graph (G'+bridges) and using summation over each r_i to twirl the attack operator on qubit $f(i)$ respectively, where f is the flow function. We begin from qubits of system O' and follow the reverse measuring order. For the qubits of G' we distinguish between three types of qubits depending on their position and we apply a different technique for each one:

- Qubit $i \in O'$: We assume that E''_G does contain entangling operators among the output qubits in our computational graph (e.g. take G' to be the brickwork graph). We extract stabilizer $Z_{f^{-1}(i)}^{r_{f^{-1}(i)}} X_i^{r_{f^{-1}(i)}}$ from our graph $E''_G|M^{(3)}\rangle$. This cancels dependence on $r_{f^{-1}(i)}$ on the protocol unitary operation applied on qubit $f^{-1}(i)$ (changing it from $H_{f^{-1}(i)}R_{f^{-1}(i)}(\phi'_{f^{-1}(i)} + r_{f^{-1}(i)}\pi)$ to $H_{f^{-1}(i)}R_{f^{-1}(i)}(\phi'_{f^{-1}(i)})$) and allows as to sum over $r_{f^{-1}(i)}$ and r_i to get property for qubit i (using Lemma 2):

$$\sum_{r_{f^{-1}(i)}, r_i} Z_i^{b_{f^{-1}(j\sim i)} + r_{f^{-1}(j\sim i)}} X_i^{b_{f^{-1}(i)} + r_{f^{-1}(i)}} Z_i^{r_i} P_{u|i} X_i^{r_{f^{-1}(i)}} Z_i^{r_i} E''_G(|M^{(4)}\rangle\langle M^{(4)}|)$$
$$E''_G Z_i^{r_i} X_i^{r_{f^{-1}(i)}} P_{u'|i} Z_i^{r_i} X_i^{b_{f^{-1}(i)} + r_{f^{-1}(i)}} Z_i^{b_{f^{-1}(j\sim i)} + r_{f^{-1}(j\sim i)}} = 0 \text{ if } P_{u|i} \neq P_{u'|i} \quad (3.10)$$

where $|M^{(4)}\rangle$ comes from $|M^{(3)}\rangle$ by extracting $Z_i^{r_i}$ from the output states.

Note that for the terms that remain ($P_{u|i} = P_{u'|i}$) operations that depend on $r_{f^{-1}(i)}, r_i$ can be eliminated by looking at their commutation properties with $P_{u|i}$ thus eliminating all dependences from $r_{f^{-1}(i)}, r_i$ in the formula. However, operations $Z_i^{r_{f^{-1}(j\sim i)}}$ and all the b corrections remain.

- Qubit $i \notin (O' \cup I)$: We extract stabilizer $Z_{f^{-1}(i)}^{r_{f^{-1}(i)}} X_i^{r_{f^{-1}(i)}} Z_{j\sim i, j\neq f^{-1}(i)}^{r_{f^{-1}(i)}}$ from graph state $E''_G|M^{(4)}\rangle$. This cancels dependence on $r_{f^{-1}(i)}$ on the unitary operation applied on qubit $f^{-1}(i)$ and on the unitary operation applied on qubits $\{j \sim i, j \neq f^{-1}(i)\}$ (in case $\{j \sim i, j \neq f^{-1}(i)\}$ are output qubits their attack cross terms have already been cancelled and therefore terms $Z_{j\sim i, j\neq f^{-1}(i)}^{r_{f^{-1}(i)}}$ commute with the attack and cancel with the final corrections - this is why following reverse measuring order is crucial). By summing the elements that still depend on $r_{f^{-1}(i)}$:

$$\sum_{r_{f^{-1}(i)}} \langle b_i|P_{u|i}H_iR_i(\phi'_i)$$
$$X_i^{r_{f^{-1}(i)}} E''_G(|M^{(4)}\rangle\langle M^{(4)}|)E''_G X_i^{r_{f^{-1}(i)}} R_i(\phi'_i)^\dagger H_i P_{u'|i}|b_i\rangle \quad (3.11)$$

where $\phi'_i = (-1)^{b_{f^{-1}(i)} + r_{f^{-1}(i)}}\phi_i + (b_{f^{-1}(j\sim i, j\neq f(i))} + r_{f^{-1}(j\sim i, j\neq f(i))})\pi$.

We commute $X_i^{r_{f^{-1}(i)}}$ with $R_i(\phi'_i + r_i\pi)$ and H_i and also extract a $Z_i^{r_{f^{-1}(i)}}$ from both $\langle b_i|$ and $|b_i\rangle$ without affecting the state:

$$\sum_{r_{f^{-1}(i)}} \langle b_i | Z_i^{r_{f^{-1}(i)}} P_{u|i} Z_i^{r_{f^{-1}(i)}} H_i R_i(\phi_i'')$$
$$E_G''(|M^{(4)}\rangle\langle M^{(4)}|) E_G'' R_i(\phi_i'')^\dagger H_i Z_i^{r_{f^{-1}(i)}} P_{u'|i} Z_i^{r_{f^{-1}(i)}} |b_i\rangle \quad (3.12)$$

where $\phi_i'' = (-1)^{b_{f^{-1}(i)}}\phi_i + (b_{f^{-1}(j\sim i, j\neq f(i))} + r_{f^{-1}(j\sim i, j\neq f(i))})\pi$.

To apply Lemma 2 we also need the random X elements. These are acquired by the following mathematical trick: We extract stabilizer $Z_i^{r_i'} X_{f(i)}^{r_i'} Z_{j\sim f(i), j\neq i}^{r_i'}$ for $r_i' \leftarrow_R \{0,1\}$ from graph state $E_G''|M^{(4)}\rangle$ and at the same time changing variable $\hat{b}_i \leftarrow b_i + r_i'$ everywhere. The new terms cancel everywhere (again following reverse measurement order is crucial) except at qubit i so that we get:

$$\sum_{r_{f^{-1}(i)}, r_i'} \langle \hat{b}_i | X^{r_i'} Z_i^{r_{f^{-1}(i)}} P_{u|i} Z_i^{r_{f^{-1}(i)}} X^{r_i'} H_i R_i(\phi_i'') E_G''(|M^{(4)}\rangle\langle M^{(4)}|) E_G''$$
$$R_i(\phi_i'')^\dagger H_i X^{r_i'} Z_i^{r_{f^{-1}(i)}} P_{u'|i} Z_i^{r_{f^{-1}(i)}} X^{r_i'} |\hat{b}_i\rangle = 0 \text{ if } P_{u|i} \neq P_{u'|i} \quad (3.13)$$

- Qubit $i \in I$: We assume that E_G'' does not contain entangling operators between the input qubits (e.g. brickwork graph). Let us employ the following mathematical trick to generate the random Z elements: Extract stabilizer $X_i^{z_i} Z_{f(i)}^{z_i}$ from graph $E_G''|M^{(4)}\rangle$, where $z_i \leftarrow_R \{0,1\}$. Commuting with the rotation on system i: $R_i(\phi_i) X_i^{z_i} = X_i^{z_i} R_i((-1)^{z_i}\phi_i)$, up to a phase which gets cancelled when you commute on both sides. On system $f(i)$: $R_{f(i)}((-1)^{b_i}\phi_{f(i)}) Z_{f(i)}^{z_i} = R_{f(i)}((-1)^{b_i}(\phi_{f(i)} + z_i\pi))$. For any choice of attack $P_{u|i}$ ($P_{u'|i}$) from Bob, it implements operation $J(\phi_{f(i)} + z_i\pi) J((-1)^{z_i}\phi_i)|+\rangle_i$ ($\langle +|_i J((-1)^{z_i}\phi_i)^\dagger J(\phi_{f(i)} + z_i\pi)^\dagger$), up to a sign flip of $\phi_{f(i)}$ depending on the attack, which is equivalent up to a global phase to operation $J(\phi_{f(i)}) J(\phi_i)|+\rangle_i (\langle +|_i J(\phi_i)^\dagger J(\phi_{f(i)})^\dagger)$, up to the same sign flip on $\phi_{f(i)}$, so that we can replace the angles with the latter. Also, we can extract Z^{z_i} from $\langle b_i|$ and from $|b_i\rangle$.

  Let us also employ the same trick for generating the random X elements as we did in the previous step, using parameters $x_i \leftarrow_R \{0,1\}$ and then apply Lemma 2 to get:

$$\sum_{x_i, z_i} \langle b_i | X^{x_i} Z^{z_i} P_{u|i} Z^{z_i} X^{x_i} H_i R_i(\phi_i)$$
$$E_G''(|M^{(4)}\rangle\langle M^{(4)}|) E_G'' R_i(\phi_i)^\dagger H_i X^{x_i} Z^{z_i} P_{u'|i} Z^{z_i} X^{x_i} |b_i\rangle = 0 \text{ if } P_{u|i} \neq P_{u'|i} \quad (3.14)$$

There are also the qubits of the computational system that are bridges:

- Qubit i is a bridge: E''_G entangles i with two qubits from G', $j : (j,i) \in G''$. To create the random Z elements we pick $r'_i \leftarrow_R \{0,1\}$ and extract stabilizer $X_i^{r'_i} \prod_j Z_j^{r'_i}$ from graph state $E''_G|M^{(4)}\rangle$. Commuting with the rotations: For system i, where measurement angle ϕ is always fixed to $\pi/2$, $R_i(\pi/2+\pi r_i)X_i^{r'_i} = X_i^{r'_i} R_i((-1)^{r'_i}\pi/2+\pi r_i) = X_i^{r'_i} R_i(\pi/2+\pi r'_i+\pi r_i)$, up to a phase which gets cancelled when you commute on both sides. For each neighbour j: $R_j(\phi_j^{(3)}+\pi/2+\pi(b_i+r_i))Z_j^{r'_i} = R_j(\phi_j^{(3)}+\pi/2+\pi(b_i+r_i)+\pi r'_i)$, where $\phi_j^{(3)} = (-1)^{b_{f^{-1}(j)}}\phi_j + b_{f^{-1}(k\sim j,k\neq f(j))}\pi$. For any choice of attack $P_{u|i}$ ($P_{u'|i}$) from Bob, the operation applied is equivalent to replacing the rotation of i with $R_i(\pi/2+\pi r_i)$ and of neighbours j with $R_j(\phi_j^{(3)}+\pi/2+\pi(b_i+r_i))$, thus eliminating all other dependences on r'_i except on the generated Pauli element. Also, we can extract $Z^{r'_i}$ from $\langle b_i|$ and from $|b_i\rangle$.

  To generate the random X elements we extract a Z^{r_i} from $R_i(\pi/2+\pi r_i)$ and also change variable $\hat{b} \leftarrow b_i + r_i$ to cancel any dependence on r_i on neighbours j, thus being able to apply Lemma 2 on i:

$$\sum_{r_i,r'_i} \langle \hat{b}_i|X^{r_i}Z^{r'_i}P_{u|i}Z^{r'_i}X^{r_i}H_iR_i(\pi/2)$$
$$E''_G(|M^{(4)}\rangle\langle M^{(4)}|)E''_G R_i(\pi/2)^\dagger H_i X^{r_i}Z^{r'_i}P_{u'|i}Z^{r'_i}X^{r_i}|\hat{b}_i\rangle = 0 \text{ if } P_{u|i} \neq P_{u'|i} \quad (3.15)$$

Eliminating the cross-terms on system T: For the trap qubits we have two cases depending on their position:

- For positions t_i in the measured system we have states:

$$\sum_{r_{t_i},b_{t_i}} |b_{t_i}+r_{t_i}\rangle\langle b_{t_i}|P_{u|t_i}|r_{t_i}\rangle\langle r_{t_i}|P_{u'|t_i}|b_{t_i}\rangle\langle b_{t_i}+r_{t_i}| \quad (3.16)$$

  By changing variable $\hat{b}_{t_i} \leftarrow b_{t_i} + r_{t_i}$

$$\sum_{r_{t_i},\hat{b}_{t_i}} |\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|X^{r_{t_i}}P_{u|t_i}X^{r_{t_i}}|0\rangle\langle 0|X^{r_{t_i}}P_{u'|t_i}X^{r_{t_i}}|\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|$$
$$= \sum_{r_{t_i},r'_{t_1},\hat{b}_{t_i}} |\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|Z^{r'_{t_i}}X^{r_{t_i}}P_{u|t_i}X^{r_{t_i}}Z^{r'_{t_i}}|0\rangle\langle 0|Z^{r'_{t_i}}X^{r_{t_i}}P_{u'|t_i}X^{r_{t_i}}Z^{r'_{t_i}}|\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|$$
$$= 0 \text{ if } P_{u|t_i} \neq P_{u'|t_i} \quad (3.17)$$

- For positions t_i in the returned system:

$$\sum_{r_{t_i},b_{t_i}} |b_{t_i}\rangle\langle b_{t_i}|HS^{\dagger\beta_{t_i}}Z^{r_{t_i}}P_{u|t_i}Z^{r_{t_i}}S^{\beta_{t_i}}|+_{t_i}\rangle\langle+_{t_i}|S^{\dagger\beta_{t_i}}Z^{r_{t_i}}P_{u'|t_i}Z^{r_{t_i}}S^{\beta_{t_i}}H|b_{t_i}\rangle\langle b_{t_i}| \quad (3.18)$$

Let $P_{u|t_i} = X^{u_x}Z^{u_z}$ and $P_{u'|t_i} = X^{u'_x}Z^{u'_z}$. A simplified version of Lemma 2 is:

$$\sum_i Z^i X^q Z^i \rho Z^i X^{q'} Z^i = 0, \text{ if } q \neq q' \quad (3.19)$$

where ρ is a 2×2 matrix, q, q', i are bits. Therefore, applying Equation 3.19 on state in 3.18 it becomes:

$$= \sum_{b_{t_i}} |b_{t_i}\rangle\langle b_{t_i}|HS^{\dagger\beta_{t_i}}X^{u_x}Z^{u_z}S^{\beta_{t_i}}|+_{t_i}\rangle\langle+_{t_i}|S^{\dagger\beta_{t_i}}Z^{u'_z}X^{u_x}S^{\beta_{t_i}}H|b_{t_i}\rangle\langle b_{t_i}|$$

Notice that index u_x on the attack is the same on both sides.

Similarly to the previous cases, we add an extra random operation which does not affect the state and depends on $r'_{t_i} \leftarrow_R \{0,1\}$:

$$= \sum_{r'_{t_i},b_{t_i}} |b_{t_i}\rangle\langle b_{t_i}|HX^{r'_{t_i}}S^{\dagger\beta_{t_i}}X^{u_x}Z^{u_z}S^{\beta_{t_i}}X^{r'_{t_i}}|+\rangle\langle+|X^{r'_{t_i}}S^{\dagger\beta_{t_i}}Z^{u'_z}X^{u_x}S^{\beta_{t_i}}X^{r'_{t_i}}H|b_{t_i}\rangle\langle b_{t_i}|$$

$$= (-1)^{\beta_{t_i}u_x} \sum_{r'_{t_i},b_{t_i}} |b_{t_i}\rangle\langle b_{t_i}|HX^{r'_{t_i}}Z^{\beta_{t_i}u_x}X^{u_x}Z^{u_z}X^{r'_{t_i}}|+\rangle\langle+|X^{r'_{t_i}}Z^{u'_z}X^{u_x}Z^{\beta_{t_i}u_x}X^{r'_{t_i}}H|b_{t_i}\rangle\langle b_{t_i}|$$

$$= (-1)^{\beta_{t_i}u_x} \sum_{r'_{t_i},b_{t_i}} |b_{t_i}\rangle\langle b_{t_i}|HZ^{\beta_{t_i}u_x}X^{u_x}X^{r'_{t_i}}Z^{u_z}X^{r'_{t_i}}|+\rangle\langle+|X^{r'_{t_i}}Z^{u'_z}X^{r'_{t_i}}X^{u_x}Z^{\beta_{t_i}u_x}H|b_{t_i}\rangle\langle b_{t_i}|$$

Using a similarly simplified version of Lemma 2 as in Equation 3.19 but for the case of Pauli X twirl:

$$= (-1)^{\beta_{t_i}u_x} \sum_{b_{t_i}} |b_{t_i}\rangle\langle b_{t_i}|HZ^{\beta_{t_i}u_x}X^{u_x} \sum_{r'_{t_i}} (X^{r'_{t_i}}Z^{u_z}X^{r'_{t_i}}|+_{t_i}\rangle\langle+_{t_i}|X^{r'_{t_i}}Z^{u'_z}X^{r'_{t_i}})$$
$$X^{u_x}Z^{\beta_{t_i}u_x}H|b_{t_i}\rangle\langle b_{t_i}| = 0 \text{ if } u_z \neq u'_z$$

Eventually, tackling every case separately, we have managed to eliminate all terms where $u \neq u'$, which gives:

$$\rho_{out} = \sum_{b} \sum_{t,\{\beta_{t_i}:t_i>N-n''\}} p(t,\{\beta_{t_i}:t_i>N-n''\}) \sum_{k,u} |a_{k,u}|^2 \mathcal{D}(C'_{O'}\langle b'|P_{u|\{i:i\notin T,\Delta,D\}} (\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger})P_{u|\{i:i\notin T,\Delta,D\}}|b'\rangle C'^{\dagger}_{O'}) \bigotimes_i Q'_{t_i} P_{u|t_i}|\eta'_{t_i}\rangle\langle\eta'_{t_i}|P_{u|t_i}Q'^{\dagger}_{t_i} \quad (3.20)$$

where

- $|M^{(4)}\rangle$ is the tensor product of $|+\rangle$ for all qubits of system $\mathcal{M}\times O'$.
- $\mathcal{P}^{(3)}$ is the unitary that contains E''_G for $\mathcal{M}\times O'$ and $H_iR_i((-1)^{b_{f^{-1}(i)}}\phi_i + b_{f^{-1}(j\sim i, j\neq f(i))}\pi)$ operators for all qubits of system $\mathcal{M}$.
- C' are the remaining Pauli corrections on the output system O' (after eliminating the r's) and depend only on b.
- $|\eta'_{t_i}\rangle$ is $|0_{t_i}\rangle$ if t_i points to a trap measured by Bob ($t_i \leq N-n''$) and is $S^{\beta_i}|+_{t_i}\rangle$ if t_i points to a trap returned and measured by Alice ($t_i > N-n''$).
- Q'_{t_i} is $|b_{t_i}\rangle\langle b_{t_i}|$ if t_i points to a trap measured by Bob, and is $|b_{t_i}\rangle\langle b_{t_i}|H_{t_i}S^{\dagger\beta_{t_i}}_{t_i}$ if t_i is part of the returned system.

3.2.4 Detection of the Pauli Attacks

The strategy will be to split the sum over the attack operators P_u in Equation 3.20 into the operators which are perfectly correctable during the FT procedure or do not have any effect on the output and the rest, the latter having a significant footprint to be caught by the trapification procedure with high probability.

Let us split ρ_{out} in Equation 3.20 into a sum of two terms: σ_1 and σ_2. The first term, σ_1 contains all terms u which correspond to attacks correctable or having no effect at all (Pauli Z attacks on Pauli Z measured qubits). Specifically, $u \in \mathcal{S}_1$ is the set of indices which correspond to attacks P_u with the following property:

We map the overall attack operator P_u to an operator that contains the maximum number of independently detectable errors (see Section 3.1.2). Then, the latter attack operator must have altogether $\leq \mathbb{p}'$ Pauli tensor elements $P_{u|i}$ that corrupt the computation: for $i \leq n''$ Pauli X or Y and for $i > n''$ non-identity (therefore Pauli X or Y or Z). Then, we have term:

$$\sigma_1 =$$

$$\sum_b \sum_{t,\{\beta_{t_i}:t_i>N-n''\}} p(t,\{\beta_{t_i}:t_i>N-n''\}) \sum_{k,u\in\mathcal{S}_1} |a_{k,u}|^2 \mathcal{D}(C'_{O'}\langle b'|P_{u|\{i:i\notin T,\Delta,D\}}$$
$$(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger})P_{u|\{i:i\notin T,\Delta,D\}}|b'\rangle C'^{\dagger}_{O'}) \bigotimes_i Q'_{t_i}P_{u|t_i}|\eta'_{t_i}\rangle\langle\eta'_{t_i}|P_{u|t_i}Q'^{\dagger}_{t_i}$$
$$= p'|\psi_c\rangle\langle\psi_c| \otimes |ACC\rangle\langle ACC| + p''|\psi_c\rangle\langle\psi_c| \otimes |REJ\rangle\langle REJ| \quad (3.21)$$

where

- p', p'' are probabilities with $p' + p'' = \sum_{k,u\in\mathcal{S}_1} |a_{k,u}|^2$ and they come from separating the attacks that leave all traps untouched and the rest.
- $|ACC\rangle\langle ACC|$ is the state where all trap qubits are $|0\rangle\langle 0|$ and $|REJ\rangle\langle REJ|$ denotes any state that at least one trap is $|1\rangle\langle 1|$.
- $|\psi_c\rangle\langle\psi_c|$ is the correct state for the output system O.

The second term of the sum, σ_2, contains the remaining indices u, say $u \in \mathcal{S}_2$, which correspond to attacks that can corrupt the computation even after the error correction. These are attacks P_u of the following kind: when considering only the independently detectable errors there are $> \mathbb{p}'$ total elements in the tensor product which are Pauli X or Y on measured qubits and Pauli X or Y or Z on returned qubits.

Let $P_\perp$ be the projection to the orthogonal space to the correct output state:

$$P_\perp = I - |\psi_c\rangle\langle\psi_c| \quad (3.22)$$

where $|\psi_c\rangle = U|+\rangle^{\otimes n}$, U being the target unitary computation (the description of which is given to Alice at the beginning of the protocol).

We calculate the 'bad' probability p_{bad} (sub-normalized) that the state σ_2 collapses to the 'incorrect' subspace and no trap is activated, therefore Alice accepting.

$$p_{\text{bad}} = Tr(\sum_b \sum_{t,\{\beta_{t_i}:t_i>N-n''\}} p(t,\{\beta_{t_i}:t_i>N-n''\}) P_\perp \bigotimes_i |0\rangle_{t_i}\langle 0|_{t_i} (\sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2$$
$$\mathcal{D}(C'_{O'}\langle b'|P_{u|\{i:i\notin T,\Delta,D\}}(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger})P_{u|\{i:i\notin T,\Delta,D\}}|b'\rangle C'^{\dagger}_{O'})$$
$$\bigotimes_i Q'_{t_i}P_{u|t_i}|\eta'_{t_i}\rangle\langle\eta'_{t_i}|P_{u|t_i}Q'^{\dagger}_{t_i})) \quad (3.23)$$

Tracing out everything except the trap terms:

$$\leq Tr(\sum_{b} \sum_{t,\{\beta_{t_i}:t_i>N-n''\}} p(t,\{\beta_{t_i}:t_i>N-n''\}) \sum_{k,u\in S_2} |a_{k,u}|^2 \bigotimes_i |0\rangle_{t_i}\langle 0|_{t_i} Q'_{t_i} P_{u|t_i} |\eta'_{t_i}\rangle\langle\eta'_{t_i}| P_{u|t_i} Q'^{\dagger}_{t_i}) \quad (3.24)$$

Let us clarify a subtle point. Term $\sum_t$ means $\sum_{t_1}\dots\sum_{t_{|t|}}$, where $\sum_{t_1}$ is the summation of each trap position t_i over the set of possible positions which we have denoted by S_i. For the dotted-complete graph, it is not possible to get an independent uniform distribution for t_i's simultaneously for all i (if we select one qubit to be a trap this will affect the selection of the next trap). What we actually do is to consider only a subset of vertices of the total graph that correspond to independently detectable errors in P_u (therefore the subset we consider every time depends on u). For the rest of the vertices we can set the attack operators to the identity without decreasing the total trace of the expression above. For each qubit j where the independently detectable errors apply, belonging to set S_i, we have two desirable properties: (1) attack $P_{u|j}$ will affect the trap if it coincides with it and therefore reduce the trace of the above expression (will give trace 0 when it coincides with a non-output trap and trace $\frac{1}{2}$ averaged over β when it coincides with an output trap) and (2) for all possible selection of trap positions $\{t_l : \forall l \neq i\}$ that (for the S_l' that there are errors) coincide with independently detectable errors of u we have $p(t_i = j|\{t_l\}) \geq c'$, where c' is independent of the particular selection and is equal to $\frac{1}{\max_i(|S_i|)}$ (in the case of dotted complete graph $c' = \frac{1}{9}$). This comes as a consequence of Definition 7.5 in [Fitzsimons and Kashefi, 2012]. The second property means, in other words, that the terms $\bigotimes_i |0\rangle_{t_i}\langle 0|_{t_i} Q'_{t_i} P_{u|t_i} |\eta'_{t_i}\rangle\langle\eta'_{t_i}| P_{u|t_i} Q'^{\dagger}_{t_i}$ that have trace < 1 in the above summation have a lower bound in their probabilities (taking the marginal over the trap positions in S_l's that there are no independently detectable errors) and therefore set an upper bound in the overall expression.

Separating S_i's that contain qubits measured by Bob and those that contain qubits returned to Alice:

$$= \sum_{k,u\in S_2} |a_{k,u}|^2 \sum_{\{t_i:t_i\leq N-n''\}} p(\{t_i : t_i \leq N-n''\}) |\langle 0|_{t_i} P_{u|t_i} |0\rangle_{t_i}|^2$$
$$\sum_{\{t_i,\beta_{t_i}:t_i>N-n''\}} p(\{t_i,\beta_{t_i} : t_i > N-n''\}|\{t_i : t_i \leq N-n''\}) |\langle +|_{t_i} S^{\dagger\beta_{t_i}} P_{u|t_i} S^{\beta_{t_i}} |+\rangle_{t_i}|^2 \quad (3.25)$$

For each i where S_i contains qubits of the measured by Bob system and each attack with index u we denote by $w_{u,i}$ the number of positions in S_i on which acts

an independently detectable error element (Pauli X or Y) of P_u. For each i where S_i contains returned qubits and each attack with index u we denote by $w_{u,i}$ the number of positions in S_i on which acts an independently detectable error element (Pauli X or Y or Z) of P_u. By the above assumption for the lower bound in the probabilities of having traps at the positions of the independent detectable errors:

$$\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 \prod_{i:\forall j\in S_i, j\leq N-n''} (1-c'w_{u,i})) \prod_{i:\forall j\in S_i, j>N-n''} (1-\frac{1}{2}c'w_{u,i}) \tag{3.26}$$

Or,

$$\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 \prod_i (1-\frac{c'}{2}(w_{u,i})) \tag{3.27}$$

Or, from the fact that $w_{u,i}$ is non-negative integer:

$$\begin{aligned} &\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 \prod_i (1-\frac{c'}{2})^{w_{u,i}}) \\ &= \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 (1-\frac{c'}{2})^{\sum_i (w_{u,i})} \end{aligned} \tag{3.28}$$

From the fact that for every $u \in \mathcal{S}_2$ the total footprint $\sum_i (w_{u,i})$ of independently detectable errors in P_u is $> \mathbb{p}'$:

$$\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 (1-\frac{c'}{2})^{\mathbb{p}'} \tag{3.29}$$

Therefore (sub-normalized) state σ_2 can be written as:

$$\begin{aligned} \sigma_2 \approx_{\sqrt{(1-\frac{c'}{2})^{\mathbb{p}'}}} & p(\sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2) |\psi_c\rangle\langle\psi_c| \otimes |ACC\rangle\langle ACC| \\ & + (1-p)(\sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2) \rho \otimes |REJ\rangle\langle REJ| \end{aligned} \tag{3.30}$$

for some probability p and density matrix ρ. Or,

$$\sigma_2 \approx_{(1-\frac{c'}{2})^{\mathbb{p}'}} p_1 |\psi_c\rangle\langle\psi_c| \otimes |ACC\rangle\langle ACC| + p_2 \rho \otimes |REJ\rangle\langle REJ| \tag{3.31}$$

where

- $p_1 + p_2 = \sum_{k,u \in \mathcal{S}_2} |a_{k,u}|^2$
- ρ is a density matrix.

Summing the terms σ_1 and σ_2, Theorem 1 is satisfied with $\varepsilon = (1 - \frac{c'}{2})^{\mathbb{p}'}$.

Chapter 4

Verifiable Blind Quantum Computing with Localised Output

4.1 Localisation Gadget and Protocol

In this section we give an explicit description of the version of the FK protocol which localises the position of the output on Bob's side, called the *localising verification protocol*. For any choice of random variables, including the positioning of the traps on the graph that describes the resource state, the position of the qubits that contain the output of the delegated computation becomes fixed and known to both Alice and Bob - this is achieved by running blindly an extra output gadget graph. Blindness of the FK protocol is retained, while for verifiability, since there are no traps on the output to be tested, we have to relax the requirement for the output state to be ε close to the correct output up to an arbitrary CPTP map. Crucially, for our composition in the composite protocol, this CPTP map should be fixed for a choice of Bob's deviation and should not depend on the particular computation which is delegated.

In more technical terms, the main difference from the original FK protocol is the modification of the graph state by appending on the output a certain number of gadget states, one of which is depicted in Figure 4.1. Specifically, one gadget is attached on each subset S_γ of the output of the FK protocol graph (system to be normally returned to Alice). In the dotted-complete graph construction of the FK protocol, all output qubits belong to the set of $P(G)$ qubits and by construction they contain an equal number of trap qubits, dummy qubits and qubits participating in the actual computation and are in fact that outputs of this computation. Following the same construction of the FK

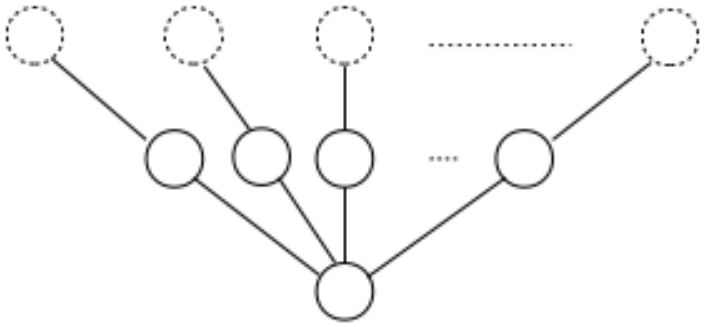

Figure 4.1: Single gadget state attached to a subset S_γ (represented by dashed circles) of the output of the normal graph used in the FK protocol. The number of branches depends on $|S_\gamma|$. Only one of the qubits of the first row of the gadget is a non-dummy state allowing to teleport blindly the actual output to the bottom qubit.

protocol we choose the sets S_γ such that each set contains one trap qubit, one dummy qubit and one computational qubit (therefore the size of all output sets S_γ is 3). The main idea is that if Bob learns the sets S_γ, which he does in our case because we place a gadget on top of each S_γ, there is always a trap that can take any position in the set S_γ. The scheme can generalize to other constructions with different sizes of S_γ by increasing the size of the gadget.

The goal of the gadget system is to teleport only the computational qubit to the fixed vertex at the bottom of each gadget. Therefore the first layer of the gadget qubits (in Figure 4.1 the first row of solid circles, that are connected directly to the existing graph of the FK protocol) contains $|S_\gamma| - 1$ dummy qubits which isolate the trap and the dummies of the output S_γ from the bottom qubit. The computational qubit and the gadget qubit that is connected to it are both measured with $\phi = 0$ so that the effective computation is identity and the qubit is teleported to the bottom of the gadget. The gadget structure is such (one gadget per S_γ and symmetric) that they can be added without breaking the uniformity of the positioning of the traps in the FK protocol graph. The steps of the localising protocol are given in Protocol 3.

Protocol 3 Localising Verification Protocol (based on the FK protocol)

Alice's input. Description of a computation in the MBQC model (or equivalent) using a convenient underlying open graph state (G,I,O). The computation is represented, for any qubit $i \in G \setminus O$, as a measurement angle φ_i (together with the set of X-dependences D_i^X and Z-dependences D_i^Z and a fixed partial order of measuring depending on the graph structure). The input is set to be the Hadamard basis state of n qubits: $|+\rangle^{\otimes n}$. Protocol can be extended to admit external quantum input by applying the techniques described in the FK protocol.

Bob's output. A quantum state that contains the encoded and one time padded quantum output of the computation.

The protocol

1. Preprocessing 1. Alice translates the computation to a Fault Tolerant (FT) MBQC pattern that can detect $\mathbb{p}$ errors. Let the updated open graph be (G',I',O'), where $|G| = m'$ and $|I'| = |O'| = n'$.

2. Preprocessing 2. Alice embeds the encoded computation pattern into a suitable graph G'' which has the following property: There exists a fixed order of measurement which respects the computational flow and each computational qubit belongs to a constant size subset of qubits S_γ in which a trap can be at any position with uniformly random probability. An example (up to the assumption that we consider only independently detectable errors) is the dotted complete graph of size $O(m'^2)$.

3. Preprocessing 3. Alice attaches a gadget graph (Figure 4.1) on each subset S_γ of qubits of the output system of the previous step's graph (G''). Let the new graph, that contains all gadgets, be denoted by G'''. The effect of these extra graphs is that they fix the position of system O'. Since we attach a constant number of qubits for every qubit of the output system, the total number of qubits of G''' is $N = O(m'^2)$.

Protocol 3 Cont'd

4. Alice prepares the rotated qubits. For $i = 1$ to N:

   (a) If qubit is a dummy: prepares $|d_i\rangle$, $d_i \leftarrow_R \{0,1\}$ (where $\leftarrow_R$ means *chosen uniformly at random from*).

   (b) If qubit is not dummy and not in O': prepares $\prod_{j \in N_G(i) \cap D} Z^{d_j} |+_{\theta_i}\rangle$, where $|+_{\theta_i}\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta_i}|1\rangle)$, $\theta_i \leftarrow_R \{0, \pi/4, 2\pi/4, \ldots, 7\pi/4\}$.

   (c) If qubit is not a dummy and is in O': Same as previous step but θ is fixed to 0 and an extra Z^{r_i} rotation is applied where r_i is a random bit.

   (d) She sends the qubit to Bob.

5. Bob entangles the states according to the graph state by applying cZ gates.

6. Bob performs the rest of the computation using classical help from Alice. For i which ranges over all qubits (respecting the flow) except system O':

   (a) Alice computes the actual measurement angle ϕ'_i using the dependences and the previous measurement results ($\phi_i = 0$ for dummies and traps).

   (b) Alice chooses a random bit r_i and computes $\delta_i = \phi'_i + \theta_i + \pi r_i$.

   (c) Alice transmits δ_i to Bob.

   (d) Bob performs measurement $M_i^{\delta_i}$ on qubit i.

   (e) Bob transmits the result to Alice.

   (f) Alice flips the result if $r_i = 1$, otherwise does nothing.

7. Alice sets her indicator bit to accept if all trap tests where successful.

8. Bob's system will contain the output qubits O' placed at a fixed position on the graph.

4.2 Verifiability of the Localisation Protocol

The following theorem is an alternative way to express verifiability and is indeed equivalent to the standard definition using the projectors [Dunjko et al., 2014], up to the final attack on the state (which is crucially independent on the computation). We prefer this notation here as we are interested in the form of the state that will be later used in the composite protocol.

Theorem 2. *The output of Protocol 3 when decoded for the FT procedure and decrypted for the quantum one-time-pad (assuming that these are done honestly by Bob or are done by Alice), averaged over all random parameters of Alice, denoted as* ν*, is of the following form:*

$$\sum_{\nu} \rho_{out}^{\nu} \approx_{\varepsilon} p_{acc} \mathcal{E}(|\psi_c\rangle\langle\psi_c|) \otimes |ACC\rangle\langle ACC| + (1 - p_{acc}) \rho' \otimes |REJ\rangle\langle REJ|$$

where $\varepsilon = \frac{1}{c^{\mathbb{p}}}$ *for some* $c > 1$ *and* $\mathbb{p}$ *parameter of FT,* p_{acc} *is a probability,* $|\psi_c\rangle$ *is the correct output of the computation,* $\mathcal{E}$ *is a CPTP-map where all Kraus operators are Pauli operators and is independent of* $|\psi_c\rangle$ *and* ρ' *is a normalized state.*

The proof of Theorem 2, is given below. The proof is similar to the proof in Section 3.2 with some notable differences:

- As before, we decompose the attack operators into the Pauli basis, which are tensor products of Pauli+identity operators on the overall state of Bob (tracing out his private subsystem). We can split each tensor product into two smaller tensor products, one that applies on the trapified system, which is the normal graph G''', and one that applies on the non-trapified system, that is the gadgets and measurement angles. Due to blindness on both G''' and the gadget we can reduce both attacks to Pauli attacks, say P_u and P_v respectively.

- For P_u we do the standard analysis: errors with weight less than a certain bound can be corrected and for the rest there is an exponentially high probability of hitting at least one trap.

- For P_v we rewrite these attacks as attacks on the output qubits by commuting with the gadget computation (which crucially does not depend on the actual

computation). Moreover, since the gadget computation is Clifford the final attack is still a Pauli attack. Therefore, the only attacks that remain on the output after we do the trap analysis are these attacks that do not depend on the computation, as we wanted.

Proof. Single index notation is followed to enumerate the qubits participating in the graph where N is the total number of qubits and the last n' qubits are the encoded output qubits (system O').

Extra notation has to be introduced first. Vector ν is used to represent all random secret parameters chosen by Alice throughout the execution of protocol, including $r = \{r_i\}$, $\theta = \{\theta_i\}$, $d = \{d_i\}$ and positions of traps $t = \{t_i\}$. Parameter $p(\nu)$ gives the probability of a particular choice of random secret parameters. Summing over a vector (e.g. $\sum_r$) means that we sum over all possible choices for the elements of that vector (e.g. all possible bit-strings of size N for r).

For convenience we name the subsystems of the joint Alice-Bob system:

- System $\mathcal{M}$ is the union of all the non-dummy, non-trap qubits of G'' (therefore includes all the necessary bridge qubits introduced when embedding G' in G'') and the non-dummy qubits of the first layer of each gadget (the ones that are measured by Bob) introduced when embedding G'' into G'''. Therefore $\mathcal{M}$ contains all the qubits of the graph state that are measured and participate in the computation. We call them measured computational qubits.

- System D contains all dummy qubits.

- System T contains all trap qubits (here all of them are measured by Bob).

- System O' is the encoded output system at fixed position. These are all the qubits of the second layer of each gadget (the ones that are not measured by Bob).

- System Δ is the system of the measurement angles send by Alice to Bob, where each angle is represented by three qubits in the computational basis.

- System B is Bob's private system, assumed to be initially in the blank state.

Each measurement performed at Step 6d of the protocol is analysed into a unitary part and a Pauli Z measurement. Without loss of generality we can represent any dishonest behaviour of Bob at any step as applying the correct unitary operators and then an arbitrary unitary attack operator.

In order to prove the theorem argument we need to consider the state of the protocol after the extra steps of Alice receiving system O' and applying the output Pauli corrections $C_{O'}$ and decoding map $\mathcal{D}$ for the FT QECC. The output that Alice holds after these extra steps, averaged over random parameters and for all measurement outcomes with the corresponding probabilities, is:

$$\rho_{out} = Tr_{B,\mathcal{M},\Delta,D}(\sum_b \sum_\nu p(\nu)\mathcal{D}(C|b_{N-n'}\rangle\langle b_{N-n'}|U_{N-n'}H_{N-n'}R_{N-n'}(|\delta_{N-n'}\rangle\langle\delta_{N-n'}|\otimes\ldots \ldots|b_1\rangle\langle b_1|U_1H_1R_1(|\delta_1\rangle\langle\delta_1|\otimes E_G|M\rangle\langle M|\otimes|0\rangle\langle 0|^{\otimes B}E_G)R_1^\dagger H_1U_1^\dagger|b_1\rangle\langle b_1|\ldots)R_{N-n'}^\dagger H_{N-n'}U_{N-n'}^\dagger|b_{N-n'}\rangle\langle b_{N-n'}|C^\dagger)) \quad (4.1)$$

where:

- $|M(\theta,d,t)\rangle$ is the state that Bob receives after Step 4 of the protocol (input, auxiliary qubits, dummy qubits).
- E_G is the global entangling operator that Bob applies at step 5 of the protocol.
- $\delta(\theta,r,b)$ is the vector of $\delta_i(\theta,r,b)$ which are measurement angles that Alice sends to Bob at iteration i of step 6 of the protocol.
- $R_i(\delta_i)$ is a controlled unitary rotation of qubit i around z-axis by angle δ_i.
- H_i is a Hadamard gate applied on qubit i.
- U_i is the unitary attack of Bob with index i and applies on qubits $\geq i$, private Bob's system and all measurement angles.
- $C(r,b)$ includes the Pauli correction $C_{O'}$ that Alice performs to the output system O' and the $Z^{r_{t_i}}$ corrections on the trap measurement results.
- $\mathcal{D}$ is the map that Alice applies to decoding O' for the QECC used in the fault tolerant procedure.

By commuting all measurement angles (trivially) and all attack unitary operators (by observing that they trivially commute with all measurements $\leq i$) and merge them into a new unitary U that applies on the whole of the system we get (see also Figure 3.4):

$$\rho_{out} = Tr_{B,\mathcal{M},\Delta,D}(\sum_{b}\sum_{\nu} p(\nu)\mathcal{D}(C|b\rangle\langle b|U(H_{N-n'}R_{N-n'}(.)\ldots H_1R_1(.)E_G(|M\rangle\langle M|\otimes|\delta\rangle\langle\delta|)E_GR_1(.)^{\dagger}H_1\ldots R_{N-n'}(.)^{\dagger}H_{N-n'}\otimes|0\rangle\langle 0|^{\otimes B})U^{\dagger}|b\rangle\langle b|C^{\dagger})) \quad (4.2)$$

where U is a unitary attack operator that is chosen by Bob and applies on all graph G''' qubits, private Bob's system and all measurement angles.

The trap system is not entangled with the rest when the honest computation is applied. We can separate the terms that apply on the trap (remember that the trap is placed among the measured qubits). Applying the entangling operators between the traps and their neighbours, which are always dummy qubits does not have any effect other than undoing the pre-rotation of the trap (for the case the dummy is selected to be $|1\rangle$). Applying $H_tR_t(\delta_t)$ on each trap, where $\delta_t = \theta_t + r_t\pi$, results in getting state $|r_t\rangle$.

Therefore, it holds that:

$$\mathcal{P}(|M\rangle\langle M|\otimes|\delta\rangle\langle\delta|)\mathcal{P}^{\dagger} = \mathcal{P}'(|M'\rangle\langle M'|\otimes|\delta\rangle\langle\delta|)\mathcal{P}'^{\dagger}\bigotimes_i |r_{t_i}\rangle\langle r_{t_i}| \quad (4.3)$$

where

- $\mathcal{P} = H_{N-n'}R_{N-n'}(.)\ldots H_{t_i}R_{t_i}(.)\ldots H_1R_1(.)E_G$ is the correct unitary operation in Figure 3.4.
- $\mathcal{P}' = H_{N-n'}R_{N-n'}(.)\ldots H_1R_1(.)E'_G$, which is produced from $\mathcal{P}$ by omitting all operations applied on traps (E'_G is produced from E_G by omitting all entangling operators between traps and their neighbours)
- $|M'\rangle$ is produced from $|M\rangle$ by omitting the trap qubits.

In the next step we express the output state with Bob's private system traced out and his attack decomposed in the Pauli basis. We will use again the general property for tracing out Bob's private system and decomposing to the Pauli basis (all tensor products of Pauli+identity operators) given in Equation 3.4.

Applying this property to the state in Equation 4.2, by tracing out system B, and at the same time applying Equation 4.3 to separate the traps we get the following form. Again, $P_{u|i}$ is the i-th Pauli or identity element of P_u (tensor product of Pauli+identities with index u).

$$\rho_{out} = Tr_{\mathcal{M},\Delta,D}(\sum_b \sum_\nu p(\nu) \sum_{k,u,v,u',v'} a_{k,u,v} a^*_{k,u',v'} \mathcal{D}(C_{O'}|b'\rangle\langle b'|P_{u|i:\forall j,i\neq t_j} \otimes P_v \\ (\mathcal{P}'(|M'\rangle\langle M'| \otimes |\delta\rangle\langle\delta|)\mathcal{P}'^\dagger) P_{u'|i:\forall j,i\neq t_j} \otimes P_{v'}|b'\rangle\langle b'|C^\dagger_{O'}) \\ \bigotimes_i |b_{t_i} + r_{t_i}\rangle\langle b_{t_i}|P_{u|t_i}|r_{t_i}\rangle\langle r_{t_i}|P_{u'|t_i}|b_{t_i}\rangle\langle b_{t_i} + r_{t_i}|) \quad (4.4)$$

where

- $a_{k,u,v}$ are complex numbers, with $\sum_{k,u,v} |a_{k,u,v}|^2 = 1$.
- P_u (and $P_{u'}$) ranges over all tensor products of Pauli+identity operators and *applies on the system that is trapified*: all qubits of graph G''. Note that all these systems are measured by Bob.
- P_v (and $P_{v'}$) ranges over all tensor products of Pauli+identity operators and *applies on the system that is not trapified*: all qubits of the gadget systems, i.e. all systems added when embedding G'' to G''', and the measurement angle system Δ.
- b' is the vector that is generated from b by removing elements $\{b_{t_i}\}$.

Trivially we can trace over terms $|b'\rangle\langle b'|$:

$$\rho_{out} = Tr_{\Delta,D}(\sum_b \sum_\nu p(\nu) \sum_{k,u,v,u',v'} a_{k,u,v} a^*_{k,u',v'} \mathcal{D}(C_{O'}\langle b'|P_{u|i:\forall j,i\neq t_j} \otimes P_v \\ (\mathcal{P}'(|M'\rangle\langle M'| \otimes |\delta\rangle\langle\delta|)\mathcal{P}'^\dagger) P_{u'|i:\forall j,i\neq t_j} \otimes P_{v'}|b'\rangle C^\dagger_{O'}) \\ \bigotimes_i |b_{t_i} + r_{t_i}\rangle\langle b_{t_i}|P_{u|t_i}|r_{t_i}\rangle\langle r_{t_i}|P_{u'|t_i}|b_{t_i}\rangle\langle b_{t_i} + r_{t_i}|) \quad (4.5)$$

In the next part of the proof the attack of Bob will be reduced to a convex combination of Pauli+identity operators, consuming all random parameters except the random position of the traps to cancel the cross Pauli+identity terms that are not the same (i.e. where $u \neq u'$ and where $v \neq v'$). This follows exactly the same steps as in the proof of normal VUBQC with quantum output except the following step:

Eliminating cross-terms on the trap system T: Since all trap qubits t_i are measured and in graph G'' we have only one case:

$$\sum_{r_{t_i},b_{t_i}} |b_{t_i} + r_{t_i}\rangle\langle b_{t_i}|P_{u|t_i}|r_{t_i}\rangle\langle r_{t_i}|P_{u'|t_i}|b_{t_i}\rangle\langle b_{t_i} + r_{t_i}| \quad (4.6)$$

By changing variable $\hat{b}_{t_i} \leftarrow b_{t_i} + r_{t_i}$

$$\sum_{r_{t_i},\hat{b}_{t_i}} |\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|X^{r_{t_i}}P_{u|t_i}X^{r_{t_i}}|0\rangle\langle 0|X^{r_{t_i}}P_{u'|t_i}X^{r_{t_i}}|\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|$$
$$= \sum_{r_{t_i},r'_{t_1},\hat{b}_{t_i}} |\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|Z^{r'_{t_i}}X^{r_{t_i}}P_{u|t_i}X^{r_{t_i}}Z^{r'_{t_i}}|0\rangle\langle 0|Z^{r'_{t_i}}X^{r_{t_i}}P_{u'|t_i}X^{r_{t_i}}Z^{r'_{t_i}}|\hat{b}_{t_i}\rangle\langle\hat{b}_{t_i}|$$
$$= 0 \text{ if } P_{u|t_i} \neq P_{u'|t_i} \tag{4.7}$$

Thus, tackling each case separately, we have zeroed all terms where $u \neq u'$ and $v \neq v'$ which gives:

$$\rho_{out} = \sum_b \sum_t p(t) \sum_{k,u,v} |a_{k,u,v}|^2 \mathcal{D}(C'_{O'}\langle b'|P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}}$$
$$(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger})P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}}|b'\rangle C'^{\dagger}_{O'}) \bigotimes_i P_{u|t_i}|0\rangle\langle 0|P_{u|t_i} \tag{4.8}$$

where

- $|M^{(4)}\rangle$ is the tensor product of $|+\rangle$ for all qubits of system $\mathcal{M} \times O'$.
- $\mathcal{P}^{(3)}$ is the unitary that contains E''_G for $\mathcal{M} \times O'$ and $H_i R_i((-1)^{b_{f^{-1}(i)}}\phi_i + b_{f^{-1}(j\sim i, j\neq f(i))}\pi)$ operators for all qubits of system $\mathcal{M}$.
- C' are the remaining Pauli corrections on the output system O' (after eliminating the r's) and depend only on b.

We split the sum over the attack operators P_u (those applying on the trapified system) in Equation 4.8 into two sums: a sum over the operators P_u which are perfectly corrected by the FT QECC and the sum of the rest of the operators P_u (which are not corrected but have a significant footprint to be caught by the trapification procedure with high probability).

More formally, restricting the summation in Equation 4.8 to $u \in \mathcal{S}_1$ which have the following property: if we map each attack P_u to the corresponding attack that contains the maximum number of independently detectable errors, the latter will contain $\leq \mathbb{p}'$ tensor elements which are Pauli X or Y, we have state:

$$\sigma_1 = \sum_b \sum_t p(t) \sum_{k,u\in\mathcal{S}_1,v} |a_{k,u,v}|^2 \mathcal{D}(C'_{O'}\langle b'|P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}}$$
$$(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger})P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}}|b'\rangle C'^{\dagger}_{O'}) \bigotimes_i P_{u|t_i}|0\rangle\langle 0|P_{u|t_i} \tag{4.9}$$

Let the tensor elements of each attack $P_{v|\{i:i\notin\Delta,D\}}$ on the gadget system be separated to those applying on set of measured gadget qubits $G'''\setminus(O'\cup G'')$ (first layer of each gadget), say P_{v_1}, and those who apply on the output system O' (second layer of each gadget), say P_{v_2}.

Any single qubit Pauli operator can be analysed into a X and a Z component. For each qubit i of the first layer of each gadget, the X-component of attack $P_{v_1|i}$, denoted by $P'_{v_1|i}$, can be replaced by a $P'_{v_1|i}$ attack on output system O' (since its only effect is to flip the Pauli X corrections of the output) while the Z-component of attack $P_{v_1|i}$ can be ignored since it does not have any effect on the state. This way, every Pauli attack $P_v = P_{v_1}\otimes P_{v_2}$ on the gadgets can be reduced to a Pauli attack $P'_{v_1}P_{v_2}$ applying only on the output system O', without catching any dependence on the actual computation (independence of attack from the computation is a property of the theorem we want to prove and will be useful in the composite construction later). This analysis was easy in our case, where we have only one layer of measurements in the gadget, however one might wonder what happens if there are more untrapified measurement layers (thus P_v can potentially flip the outcomes of many layers of measurement). If these layers of measurement correspond to Clifford computation (therefore angles from the restricted set of $\{0,\pi/2,\pi,3\pi/2\}$), then the attack can indeed be rewritten as a Pauli operator on the output system O' but the new operator will potentially have a dependence on the computation performed (i.e. the computational measurement angles). If the operation performed by these layer, however, does not contain any secret, we can tolerate this dependence and continue with the subsequent analysis (this property will be better understood later in the analysis of the composite protocol and it variations).

Commuting the output Pauli attacks $P'_{v_1}P_{v_2}$ with the decoding operator $\mathcal{D}$ (fixed Clifford computation and Pauli measurements) updates them to a different Pauli operator $P''_{v_1}P'_{v_2}$:

$$
\begin{aligned}
&= \sum_b \sum_t p(t) \sum_{k,u\in\mathcal{S}_1,v} |a_{k,u,v}|^2 P''_{v_1}P'_{v_2}\mathcal{D}(C'_{O'}\langle b'|P_{u|\{i:i\notin T,D\}} \\
&\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger}P_{u|\{i:i\notin T,D\}}|b'\rangle C'^{\dagger}_{O'})P'_{v_2}P''_{v_1}\bigotimes_i P_{u|t_i}|0\rangle\langle 0|P_{u|t_i}
\end{aligned}
\tag{4.10}
$$

Since $P_u\in\mathcal{S}_1$, attack $P_{u|\{i:i\notin T,D\}}$ will be perfectly corrected by the FT QECC (which means that calculations of b's will be correct at each layer and the final decoding procedure $\mathcal{D}$ will produce the correct output).

By separating the terms in which the attack leaves all traps untouched and the rest

and tracing out $O' \setminus O$ we get:

$$\sigma_1 = \sum_i p'_i P_i |\psi_c\rangle\langle\psi_c| P_i \otimes |ACC\rangle\langle ACC| + \sum_j p''_j P_j |\psi_c\rangle\langle\psi_c| P_j \otimes |REJ\rangle\langle REJ| \quad (4.11)$$

where

- P_i (P_j) range over all general Pauli operators on the output system O.
- p'_i, p''_i are probabilities with $\sum_i (p'_i + p''_i) = \sum_{k,u\in\mathcal{S}_1,v} |a_{k,u,v}|^2$.
- $|ACC\rangle\langle ACC|$ is the state where all trap qubits are $|0\rangle\langle 0|$ and $|REJ\rangle\langle REJ|$ denotes any state that at least one trap is $|1\rangle\langle 1|$.
- $|\psi_c\rangle\langle\psi_c|$ is the correct state for the output system O, which crucially is placed on a fixed position which does not depend on the selection of the trap positions (t) (since the position of O' was also fixed due to the structure of the gadget and the decoding circuit is fixed).

We examine the rest of the u terms of the summation in Equation 4.8: for these terms, denoted by $u \in \mathcal{S}_2$, it holds that the independently detectable errors of P_u are $> \mathbb{p}'$, where the corresponding tensor elements are Pauli X or Pauli Y.

Let $P_\perp$ be the projection to the orthogonal space to the correct output state:

$$P_\perp = I - |\psi_c\rangle\langle\psi_c| \quad (4.12)$$

We calculate the 'bad' (sub-normalized) probability p_{bad} the state collapses to the 'incorrect' subspace and no trap is activated.

$$p_{\text{bad}} =$$
$$Tr(\sum_t p(t) P_\perp \bigotimes_i |0\rangle_{t_i}\langle 0|_{t_i} (\sum_b \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 \mathcal{D}(C'_{O'} \langle b'| P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}}$$
$$(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger}) P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}} |b'\rangle C'^{\dagger}_{O'}) \bigotimes_i P_{u|t_i} |0\rangle_{t_i}\langle 0|_{t_i} P_{u|t_i}) \quad (4.13)$$

Tracing out $P_\perp(\sum_b \mathcal{D}(C'_{O'}\langle b'| P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}}(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger})$ $P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D\}} |b'\rangle C'^{\dagger}_{O'})$, we have:

$$\leq Tr(\sum_t p(t) |0\rangle_{t_i}\langle 0|_{t_i} \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 \bigotimes_i P_{u|t_i} |0\rangle_{t_i}\langle 0|_{t_i} P_{u|t_i} \quad (4.14)$$

Or:

$$= \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 \sum_{t_i} p(t_i) |\langle 0|_{t_i} P_{u|t_i} |0\rangle_{t_i}|^2 \ldots \sum_{t_1} p(t_1|t_{k:k\neq 1}) |\langle 0|_{t_1} P_{u|t_1} |0\rangle_{t_1}|^2 \quad (4.15)$$

Each t_i takes values from set S_i and let its maximum size ($\forall i$) be $\frac{1}{c'}$. For each attack indexed by u, let us denote by $w_{u,i}$ the number of positions j in S_i on which acts an independently detectable element of P_u. Also, we have the fact that all positions in G'' can be trapified and the fact that for each $j \in S_i$ with an independently detectable error and for all possible selections of trap positions $\{t_l : \forall l \neq i\}$ that coincide with independently detectable errors of u we have $p(t_i = j|\{t_l\}) \geq c'$. Doing the same analysis as in the proof of Section 3.2 we have:

$$\leq \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 \prod_{i=1}^{|t|} (1 - c' w_{u,i}) \quad (4.16)$$

Or, from the fact that $w_{u,i}$ is non-negative integer:

$$\leq \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 \prod_{i=1}^{|t|} (1 - c')^{w_{u,i}}$$
$$= \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 (1 - c')^{\sum_{i=1}^{|t|} w_{u,i}} \quad (4.17)$$

Since $u \in \mathcal{S}_2$, it follows that the total footprint $\sum_{i=1}^{|t|} w_{u,i}$ of attack P_u is $> \mathbb{p}'$.

$$\leq \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 (1 - c')^{\mathbb{p}'} \quad (4.18)$$

Therefore (sub-normalized) state can be written as:

$$\sigma_2 \approx_\varepsilon p_1 |\psi_c\rangle\langle\psi_c| \otimes |ACC\rangle\langle ACC| + p_2 \rho \otimes |REJ\rangle\langle REJ| \quad (4.19)$$

where

- $\varepsilon = \sqrt{(1 - c')^{\mathbb{p}'}}$
- $p_1 + p_2 = \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2$
- ρ is a density matrix.

Summing the terms σ_1 and σ_2, Theorem 2 is satisfied with $\varepsilon = \sqrt{(1 - c')^{\mathbb{p}'}}$.

□

Chapter 5

d-level Security

Here we adapt Verifiable Universal Blind Quantum Computing (FK) protocol, which is a protocol for qubit systems, to odd prime d-level systems. The benefits are twofold. First, this construction can be easily composed with the ABE Protocol, which requires systems of dimension $d \geq 3$ to be secure, and as we prove later, this is the only (deterministic) way to compose the two protocols if the system is not returned to the verifier. Second, the existence of efficient fault tolerant schemes in the d-level systems [Watson et al., 2015] is a reason to consider d-level systems for computation and therefore verification. In the following Sections we present the d-level version of the FK protocol, based on d-level Blind QC, both using a d-level formalism of MBQC analogous to the qubit MBQC. The main difference from 2-level verification is the different bound we acquire for verifiability: depending on the levels d of the system and for protocol parameters $\mathbb{p}'$ and c', verifiability for quantum output becomes $\varepsilon = (1 - \frac{(d-1)c'}{d})^{\mathbb{p}'}$, as opposed to $\varepsilon = (1 - \frac{c'}{2})^{\mathbb{p}'}$ in the 2-level. This small improvement is due to the availability of more stabilizer states to choose for the type of the output trap, so when averaging over the types of the trap any Pauli attack is more likely to anti-commute with the trap stabilizer and get caught.

5.1 d-level Measurement-Based Quantum Computing

Measurement-based Quantum Computing (MBQC) on d-level systems has been studied in [Zhou et al., 2003] and [Hall, 2005]. Here we present a formalism for d-level MBQC that is the d-level analogue of the formalism presented in Section 1.1.3 and used in blind protocols such as in [Broadbent et al., 2009]. Based on this formalism, in Section 5.1.1 we provide the d-level analogue of the brickwork states (universal MBQC using

only XY-plane measurements), while in Section 5.3 we give the methods to 'carve' any graph out of a generic one by using the d-level analogues of the bridge and break operators.

The resource state can again be represented by an undirected graph (G,I,O), with qudits placed on the vertices and entangling operators (generalized cZ) applied whenever the two vertices are connected by an edge. Not that in the d-level case there is not only one but a family of entangling operators $cZ^k, k \in \{1,\dots d-1\}$, where k defines what we call here the *level of entanglement*. All qudits $\notin I$ are prepared in the $|+_0\rangle \equiv \frac{1}{\sqrt{d}}\sum_j^{d-1}|j\rangle$ state. All qudits $\notin O$ are measured in basis $\{R_z(\mathbf{a})F^\dagger|k\rangle\}_k$ ($\{|k\rangle\}_k$ represents the standard computational basis, where $k \in \{0,\dots,q-1\}$), which we will denote by $M^{\mathbf{a}}$. To implement a unitary on an arbitrary input, placed on I, space-time dependent generalized measurements and Pauli corrections are applied on qudits according to a given measurement pattern. One can specify the dependent generalized measurements and corrections in many different ways (e.g. [Zhou et al., 2003] and [Hall, 2005]). Here, we use the idea of the flow (see Section 1.1.3 for 2-level case).

In particular, together with graph (G,I,O) and a specific set of measurements $\{a_i\}$, we define a partial order and dependency functions D^x which is a partial function from O^C to I^C and D^z which is a partial function from O^C to $\mathcal{P}^{I^C}$, where $\mathcal{P}$ denotes the power set. Then, $j \in D_i^x$ means that j gets a Pauli X correction raised to the power of the measurement outcome of i. The choice of D_i^x is restricted such that the level of the entanglement operator between i and D_i^x is 1. Also, $j \in D_i^z$ means that j gets a Pauli Z correction raised to the power of the measurement outcome of i times the level of the entanglement operator that has been applied between j and D_i^x in the construction of the graph. The definition of the flow, which is sufficient condition for a graph to be used for unitary computation (up to global phase) and helps to define D^x and D^z, is the same as in the 2-level case (see Definition 2) with the extra property that i and $f(i)$ have level of entanglement 1. One can easily generalize by allowing for X corrections that depend on the level between i and $f(i)$ and thus make the flow depend only on the connectivity of the graph, but this is beyond the scope of our construction. The d-level version of a basic theorem (Theorem 1 in [Danos and Kashefi, 2006]) will be:

Theorem 7. *Let us have a measurement pattern on an open graph state (G,I,O) with flow $(f,\preceq)$ and measurement angles a rewritten as:*

$$P_a = \prod_{i\in O^c}^{\preceq}\left(X_{f(i)}^{s_i}\prod_{\{k:k\sim f(i),k\neq i\}}Z_k^{s_i e(k,f(i))}M_i^{a_i}\right)E_G N_{I^c}$$

where $e(k, f(i))$ is the level of the entanglement operator applied between k and $f(i)$ in E_G. The above pattern is runnable and implements the following unitary

$$U_{G,I,O,a} = 2^{|O^c|/2} \left(\prod_{i \in O^c} \langle +_{a_i}|_i \right) E_G N_{I^c} \tag{5.1}$$

where E_G and N_{I^c} represent the global entangling operator and global preparation respectively.

Proof. Deriving the first equation from the second and using the idea of anachronical patterns, i.e. that mathematically we can write each projector as a measurement and a correction before the measurement:

$$\langle +_{a_i}|_i = M_i^{a_i} Z_i^{s_i} \tag{5.2}$$

we have:

$$\prod_{i \in O^c} M_i^{a_i} Z_i^{s_i} E_G N_{I^c} \tag{5.3}$$

Extracting d-level stabilizers from E_G:

$$\begin{gathered} \prod_{i \in O^c} M_i^{a_i} Z_i^{s_i} (X_{f(i)}^{s_i} \prod_{\{k : k \sim f(i)\}} Z_k^{s_i e(k,f(i))}) E_G N_{I^c} \\ \prod_{i \in O^c} \left(X_{f(i)}^{s_i} \prod_{\{k : k \sim f(i), k \neq i\}} Z_k^{s_i e(k,f(i))} M_i^{a_i} \right) E_G N_{I^c} \end{gathered} \tag{5.4}$$

which is easy to check that is runnable as it was in the 2-level case.

□

5.1.1 d-level Universal Graph States

First we show that a patterns of the form given in Theorem 7 can perform universal unitary computation. Let us define generalized gate J as:

$$J(\mathbf{a}) = FR_z(\mathbf{a})$$
$$\text{where } \mathbf{a} = (a_0, \ldots, a_{q-1}) \text{ and } R_z(\mathbf{a}) = \sum_{i=0}^{q-1} e^{ia_i} |i\rangle\langle i|$$

In Figure 5.1 it is shown how we can implement a J gate using two qudits. The following property was used:

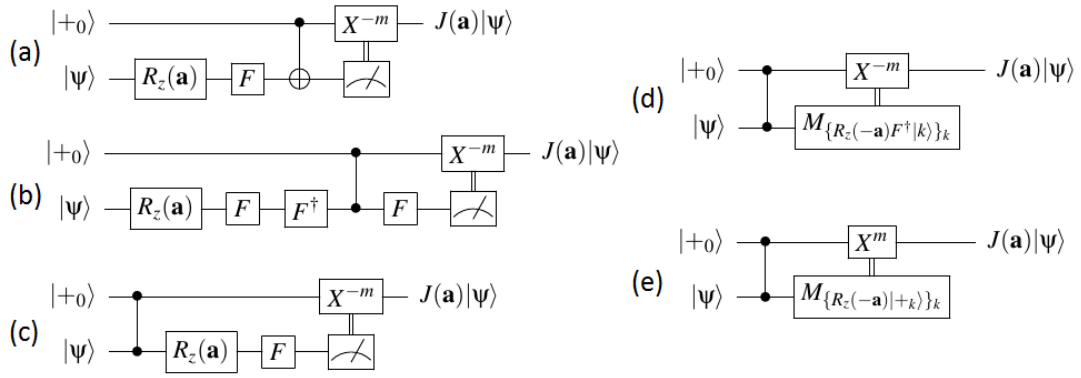

Figure 5.1: Starting from the dit teleportation circuit in (a) we show how to derive the measurement pattern for the J gate.

Lemma 3. *It holds that:*

$$(I\otimes F)cZ(I\otimes F^\dagger) = cX \tag{5.5}$$

Proof.

$$\begin{aligned}
&(I\otimes F)\mathrm{cZ}(I\otimes F^\dagger)\\
&= \sum_{a,b,c}\omega^{bc}|a,c\rangle\langle a,b|\sum_{a,b}\omega^{ab}|a,b\rangle\langle a,b|\sum_{a,b,c}\omega^{-bc}|a,c\rangle\langle a,b|\\
&= \sum_{a,b,c}\omega^{bc}|a,c\rangle\langle a,b|\sum_{a,b,c}\omega^{ac-bc}|a,c\rangle\langle a,b|\\
&= \sum_{a,b,c,d}\omega^{ac-bc+cd}|a,d\rangle\langle a,b|\\
&= \sum_{a,b,d}\left(\sum_{c}\omega^{c(a-b+d)}\right)|a,d\rangle\langle a,b|
\end{aligned}$$

The term inside parentheses is equal to 1 when $a-b+d=0\Leftrightarrow d=b-a$ and equal to 0 in all other cases. Thus we can rewrite the formula:

$$= \sum_{a,b}|a,b-a\rangle\langle a,b| = \mathrm{cX} \tag{5.6}$$

□

Gate J together with cZ operators we can implement gates of a universal set as it was the case in the 2-level MBQC.

In the original UBQC protocol all computations are performed on the brickwork state. This state has the property that we can implement any universal computation

by performing only XY-plane measurements. We give the steps to construct the graph that corresponds to one possible generalization of the brickwork state, also depicted in Figure 5.2:

Definition 10 (d-level Brickwork State Graph)**.** *The d-level Brickwork State Graph is a special graph that is composed of two different sets of edges: the grey edges and the black edges. These edges will represent different entanglement operators during the construction of the d-level brickwork state. In particular grey edges represent entanglement operator cZ, while black edges represent entanglement operator cZ^{d-2}. To construct the graph:*

1. *Assign to each vertex an index (i, j), where $1 \leq i \leq n$ is the row and $1 \leq j \leq m$ is the column.*

2. *For each row i and for all $1 \leq j \leq m-1$ connect vertices (i, j) and $(i, j+1)$ with an edge.*

3. *For each column $j \equiv (5 \bmod 25)$ and each odd row i connect vertices (i, j) and $(i+1, j)$ with a grey edge, vertices $(i, j+4)$ and $(i+1, j+4)$ with a gray edge and also vertices $(i, j+8)$ and $(i+1, j+8)$ with a black edge.*

4. *For each column $j \equiv (17 \bmod 25)$ and each even row i connect vertices (i, j) and $(i+1, j)$ with a grey edge, vertices $(i, j+4)$ and $(i+1, j+4)$ with a grey edge and also vertices $(i, j+8)$ and $(i+1, j+8)$ with a black edge.*

An interesting property of d-level gates, that, as we will see later, is the basic reason that the 'brick' component in the d-level brickwork state does not scale with the dimension d is the following:

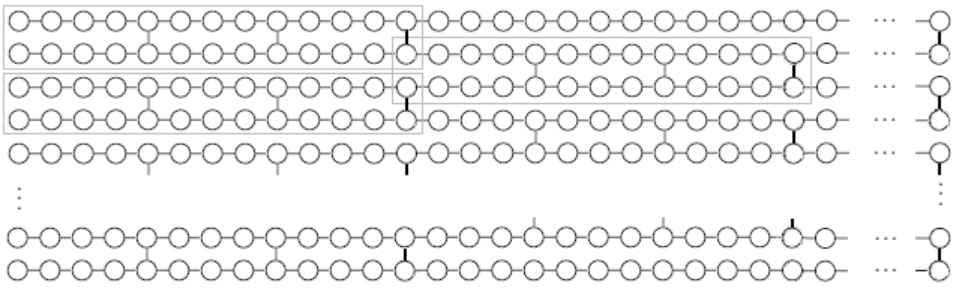

Figure 5.2: Qudit brickwork state where the 'brick' component is identified by a grey rectangle. Grey lines represent entanglement operator cZ, while black lines represent entanglement operator cZ^{d-2}, where d is the level of the system

Lemma 4. *It holds that:*

$$FFF = F^\dagger \tag{5.7}$$

Proof.

$$\begin{aligned}
&\sum_{a,b=0}^{q-1} \omega^{ba}|b\rangle\langle a| \sum_{a,b=0}^{q-1} \omega^{ba}|b\rangle\langle a| \sum_{a,b=0}^{q-1} \omega^{ba}|b\rangle\langle a| \\
&= \sum_{a,b=0}^{q-1} \omega^{ba}|b\rangle\langle a| \sum_{a,b,c=0}^{q-1} \omega^{cb+ba}|c\rangle\langle a| \\
&= \sum_{a,b,c,d=0}^{q-1} \omega^{dc+cb+ba}|d\rangle\langle a| \\
&= \sum_{a,d=0}^{q-1} \omega^{-da} \left(\sum_{b,c=0}^{q-1} \omega^{dc+cb+ba+da} \right) |d\rangle\langle a| \\
&= \sum_{a,d=0}^{q-1} \omega^{-da} \left(\sum_{b,c=0}^{q-1} \omega^{b(c+a)+dc+da} \right) |d\rangle\langle a| \\
&= \sum_{a,d=0}^{q-1} \omega^{-da} \left(1 + \sum_{b,c\neq -a} \omega^{b(c+a)+dc+da} \right) |d\rangle\langle a| \\
&= \sum_{a,d=0}^{q-1} \omega^{-da}|d\rangle\langle a| = F^\dagger
\end{aligned} \tag{5.8}$$

□

We show the universality of the d-level brickwork graph by constructing the measurement patterns of a universal set of gates, $\{Z, F, S, T, cX\}$, (see Section 2.2.1 for the explanation of why they are universal). Another gate, which we denote by D, is used to facilitate our construction and is defined as:

$$D = \sum_{j=0}^{q-1} \omega^{\frac{j^2}{2}}|j\rangle\langle j|, d > 2 \tag{5.9}$$

A series of lemmas are needed to be able to arrive to our universal pattern construction.

Lemma 5. *The following holds:*

$$\exists k : \omega^k DFDF^\dagger D = F \tag{5.10}$$

Proof.

$$\sum_{j=0}^{q-1}\omega^{\frac{j^2}{2}}|j\rangle\langle j|\sum_{j,k=0}^{q-1}\omega^{jk}|k\rangle\langle j|\sum_{j=0}^{q-1}\omega^{\frac{j^2}{2}}|j\rangle\langle j|\sum_{j,k=0}^{q-1}\omega^{-jk}|k\rangle\langle j|\sum_{j=0}^{q-1}\omega^{\frac{j^2}{2}}|j\rangle\langle j|$$
$$=\sum_{j=0}^{q-1}\omega^{\frac{j^2}{2}}|j\rangle\langle j|\sum_{j,k=0}^{q-1}\omega^{jk}|k\rangle\langle j|\sum_{j=0}^{q-1}\omega^{\frac{j^2}{2}}|j\rangle\langle j|\sum_{j,k=0}^{q-1}\omega^{\frac{j^2}{2}-jk}|k\rangle\langle j|$$

$$=\sum_{j=0}^{q-1}\omega^{\frac{j^2}{2}}|j\rangle\langle j|\sum_{j,k=0}^{q-1}\omega^{jk}|k\rangle\langle j|\sum_{j,k=0}^{q-1}\omega^{\frac{k^2}{2}+\frac{j^2}{2}-jk}|k\rangle\langle j|$$
$$=\sum_{j=0}^{q-1}\omega^{\frac{j^2}{2}}|j\rangle\langle j|\sum_{j,k,l=0}^{q-1}\omega^{kl+\frac{k^2}{2}+\frac{j^2}{2}-jk}|l\rangle\langle j|$$
$$=\sum_{j,k,l=0}^{q-1}\omega^{\frac{l^2}{2}+kl+\frac{k^2}{2}+\frac{j^2}{2}-jk}|l\rangle\langle j|$$
$$=\sum_{j,k,l=0}^{q-1}\omega^{\frac{k^2}{2}+k(l-j)+\frac{(l-j)^2}{2}+lj}|l\rangle\langle j|$$
$$=\sum_{j,k,l=0}^{q-1}\omega^{\frac{1}{2}(k+l-j)^2+lj}|l\rangle\langle j|$$
$$=\sum_{j,l=0}^{q-1}\omega^{lj}\left(\sum_{k=0}^{q-1}\omega^{\frac{1}{2}(k+l-j)^2}\right)|l\rangle\langle j|$$
$$=\sum_{j,l=0}^{q-1}\omega^{lj}\left(\sum_{k=0}^{q-1}\omega^{\frac{1}{2}k^2}\right)|l\rangle\langle j|$$
$$=\sum_{j,l=0}^{q-1}\omega^{lj}|l\rangle\langle j|=F \tag{5.11}$$

□

Lemma 6. *It also holds that:*

$$\exists k:\omega^k D^\dagger F D^\dagger F^\dagger D^\dagger = F^\dagger \tag{5.12}$$

Lemma 7. *Also:*

$$cZ^\dagger(I\otimes S^\dagger)cX(S^\dagger\otimes S)=cX \tag{5.13}$$

Proof.

$$\sum_{a,b}\omega^{-ab}|a,b\rangle\langle a,b|\sum_{a,b}\omega^{-\frac{b(b+1)}{2}}|a,b\rangle\langle a,b|\sum_{a,b}|a,b-a\rangle\langle a,b|$$
$$\sum_{a,b}\omega^{\frac{-a(a+1)+b(b+1)}{2}}|a,b\rangle\langle a,b|$$
$$=\sum_{a,b}\omega^{-a(b-a)-\frac{(b-a)(b-a+1)}{2}+\frac{-a(a+1)+b(b+1)}{2}}|a,b-a\rangle\langle a,b|$$
$$=\sum_{a,b}|a,b-a\rangle\langle a,b| \tag{5.14}$$

□

In order to show how to implement all gates of this set we first represent the 'brick' element in the circuit model notation and then we find the proper J gates for each case. The circuits used to construct each gate are given in Figure 5.3. All gates are in the form of a diagonal followed by a Fourier gate thus can directly be implemented by J gates with the appropriate parameters. An example for the parameters to use for a universal set of qutrit gates is given in 5.4 where qutrit brickwork state is used, depicted in Figure 5.5.

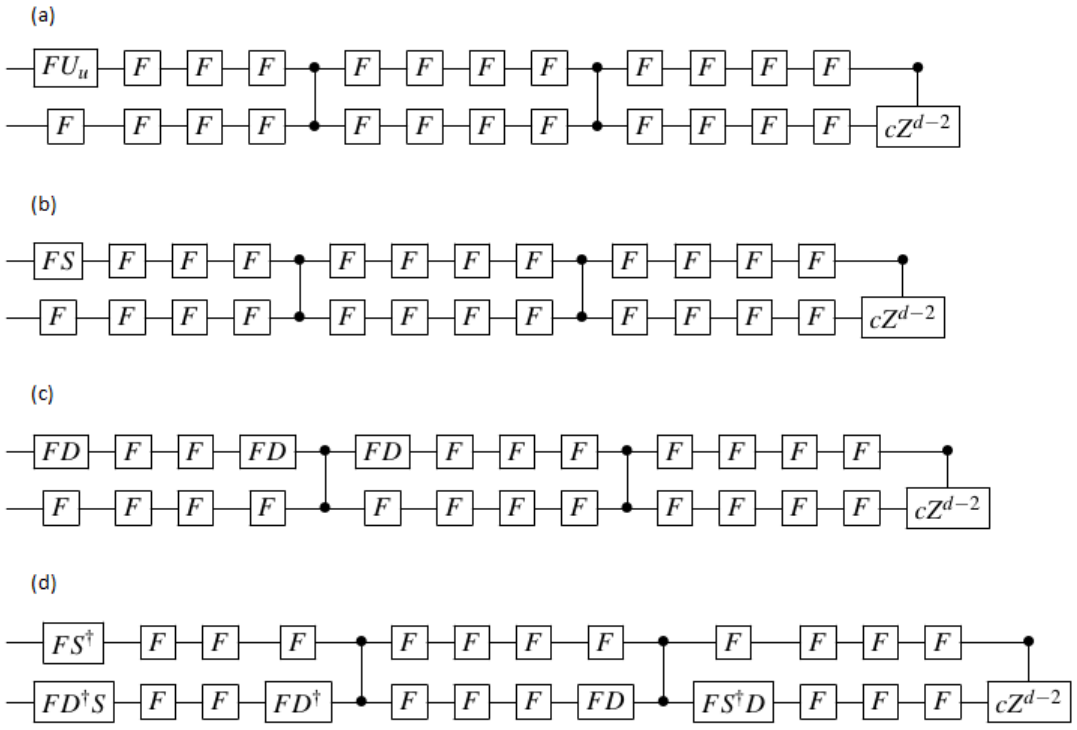

Figure 5.3: Decomposition for the quantum gates (a) T (b) S (c) F (d) cX

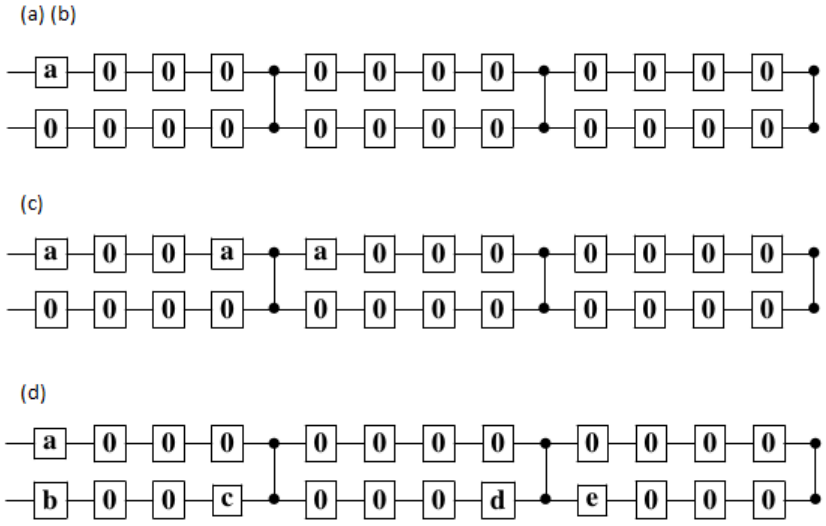

Figure 5.4: Decomposition in $J(a)$ gates for qutrit gates: (a) T: $a = (0, 2\pi/9, 16\pi/9)$ (b) S: $a = (0, 6\pi/9, 0)$ (c) F: $a = (0, 12\pi/9, 12\pi/9)$ (d) cX: $a = (0, 12\pi/9, 0), b = (0, 12\pi/9, 6\pi/9), c = (0, 6\pi/9, 6\pi/9), d = (0, 12\pi/9, 12\pi/9), e = (0, 6\pi/9, 12\pi/9)$

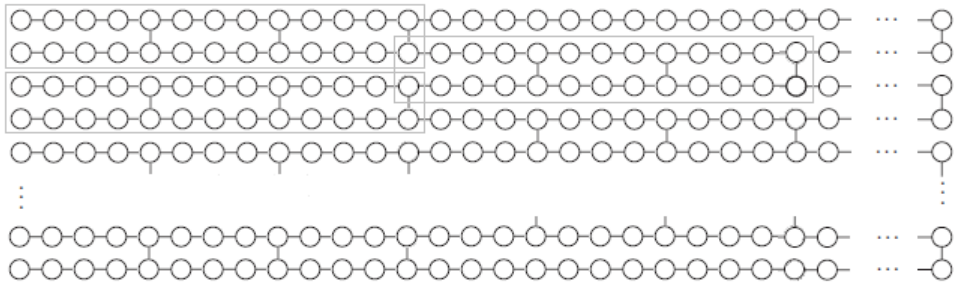

Figure 5.5: Example: brickwork state for d=3 (qutrit). The single 'brick' elements are highlighted.

5.2 d-level Blind Protocol

The idea for constructing a blind protocol in d-level system delegated computation is similar to 2-level systems. Again verifier has to provide isolated single system states that have a secret quantum encryption, in this case states $U_E|+_0\rangle$, where $|+_0\rangle \equiv \frac{1}{\sqrt{d}}\sum_j^{d-1}|j\rangle$ and $U_E \equiv T^{\gamma_i}S^{\beta_i}Z^{\alpha_i}$ is a unitary that depends on secret parameters $\alpha, \beta, \gamma \in \mathbf{F}_d$ (exception T gate for qutrits). This encryption does not correspond to all possible combinations of rotations of the relative phases by multiples of a constant angle, as it was in the case in 2-level blindness - this would have been unnecessary and too much to ask from the verifier because we only need to implement (and thus hide) the particular set of

universal gates (which are gates that belong to the 3-rd level of the Clifford hierarchy as introduced in [Gottesman and Chuang, 1999]).

Prover entangles the received states by applying generalized cZ gates on the qudits corresponding to the connected vertices in the graph representing the resource. This entangled resource state must be universal so that we do not reveal anything about the computation, except an upper bound on the size. One can use a cluster state and simulate the Z measurement using dummy qudits, which are qudits prepared in the computational basis, or use a resource that does not require Z measurements, such as the brickwork state in the 2-level system. In Figure 5.5 we present the analogue of the brickwork state for the qutrit (3-level) system.

Then, computation proceeds in rounds, where the verifier sends *measurement vectors* to the prover who uses them to determine the measurement to apply on the qudits and returns the result to the verifier to be used for the corrections of the next round of measurements. This is analogous to the 2-level case: the measurement vectors play the role of the measurement angles. Measurement vectors φ determine measurements in the $\{U_E|+_i\rangle\}_i$ basis, which gives universality for odd prime dimensions (see previous Section for more details):

A measurement in $\{0,0,0\}$ implements an F gate. Diagonal gates $Z^\alpha, S^\beta, T^\gamma$ can be implemented by the corresponding measurement vector $\{\alpha,\beta,\gamma\}$, followed by a F gate. cX can be implemented by using cZ gates of the graph together with F gates.

The order of measurements and the corrections are dictated by the flow which is exactly the same as in 2-level case. Corrections are implemented by updating the measurement vectors: correction by Z^z is $\{\alpha,\beta,\gamma\} \leftarrow \{\alpha+z,\beta,\gamma\}$ and correction by X^x is $\{\alpha,\beta,\gamma\} \leftarrow \{\alpha+x\beta+3\gamma x(x-1),\beta+6x\gamma,\gamma\}$. All operations are modulo d with the exception being the qutrit as explained earlier. These corrections can be easily verified by using the following commutation equations.

$$S^\beta X^x = X^x Z^{x\beta} S^\beta \qquad T^\gamma X^x = X^x Z^{3\gamma x(x-1)} S^{6\gamma x} T^\gamma \tag{5.15}$$

Thus correctness of the protocol comes from the fact that gates U_E commute with generalized cZ and Z rotations on measurements are corrected by addition operation on the result.

Theorem 8. *Protocol 4 is perfectly blind, leaking only an upper bound on the size of the computation.*

Protocol 4 Qudit Universal Blind Quantum Computing (UBQC) Protocol (based on qubit UBQC)

Alice's input:

- Description of a computation in the MBQC model (or equivalent) using a convenient universal underlying open graph state (G,I,O). The computation is represented, for any vertex $i \in G \setminus O$, as a measurement vector φ_i (together with the set of X-dependences D_i^X and Z-dependences D_i^Z and a fixed partial order of measuring depending on the graph structure). The input is set to the state of n qudits: $|+_0\rangle^{\otimes n}$. Protocol can be extended to admit quantum input by applying techniques described in the UBQC protocol.

Alice's output:

- A quantum state that contains the quantum output of the computation.

The protocol

1. Alice prepares the rotated qudits. For $i = 1$ to N:

   (a) She prepares $|+_{\theta_i}\rangle \equiv T^{\gamma_i} S^{\beta_i} Z^{\alpha_i} \frac{1}{\sqrt{d}}(|0\rangle + \ldots + |d-1\rangle)$, where $\theta_i = (\alpha_i, \beta_i, \gamma_i)^T, \alpha_i, \beta_i, \gamma_i \leftarrow_R \mathbf{F}_d$.

   (b) If qubit is in O: Same as in previous step if we want to have a composable with itself protocol (but will require Alice to undo the pre-rotation on the output qubits by applying non-Clifford gates), or Alice prepares state $|+_0\rangle$.

   (c) She sends the qudit to Bob

Protocol 4 Cont'd

2. Bob entangles the states according to the graph state by applying generalized cZ gates.

3. Bob performs the rest of the computation using classical help from Alice. For i which ranges over all qudits (respecting the order given by the flow), except the qudits of the O system:

   (a) Alice computes the actual measurement vector $\phi'_i = (\alpha'_i, \beta'_i, \gamma'_i)^T$ using the dependences and the previous measurement results.

   (b) Alice chooses $r_i \leftarrow_R \mathbf{F}_d$ and computes $\delta_i = \phi'_i + \theta_i + (r_i, 0, 0)^T$.

   (c) Alice transmits δ_i to Bob.

   (d) Bob performs measurement $M_i^{\delta_i}$ on qubit i. Measurements on measurement vector $\delta_i = (\alpha''_i, \beta''_i, \gamma''_i)^T \in \mathbf{F}_d^3$ correspond to generalized measurements on basis $\{Z^j T^{\gamma''_i} S^{\beta''_i} Z^{\alpha''_i} \frac{1}{\sqrt{d}}(|0\rangle + \ldots + |d-1\rangle)\}_{j=0}^{d-1}$.

   (e) Bob transmits the result to Alice.

   (f) Alice corrects the result by adding $r_i \mod d$.

4. Bob returns the output system O to Bob.

5. Alice applies the final Pauli corrections (and possibly undoes pre-rotations by $\theta_i, i \in O$) and outputs O.

Proof. We write the state of Bob system at every stage of the computation:

Step 1: Alice sends to Bob the auxiliary qubits and the first measurement vector and Bob's state becomes (ignoring his private system), averaged over random secret parameters:

$$\sum_{\theta_1\dots\theta_N,r_1} p(\theta_1\dots\theta_N,r_1)|\phi_1+\theta_1+(r_1,0,0)^T\rangle\langle\phi_1+\theta_1+(r_1,0,0)^T| \otimes_{i=1}^N |+_{0,\theta_i}\rangle\langle+_{0,\theta_i}|$$
$$= \sum_{\theta_1,r_1} p(\theta_1,r_1)|\phi_1+\theta_1+(r_1,0,0)^T\rangle\langle\phi_1+\theta_1+(r_1,0,0)^T| \otimes |+_{0,\theta_1}\rangle\langle+_{0,\theta_1}| \otimes_{i=1}^N \sum_{\theta_i} p(\theta_i)|+_{0,\theta_i}\rangle\langle+_{0,\theta_i}|$$

Defining $\theta_1' = \theta_1 + (r_1,0,0)^T$ (and noticing that it takes the same values as θ_1) :

$$= \sum_{\theta_1',r_1} p(\theta_1',r_1)|\phi_1+\theta_1'\rangle\langle\phi_1+\theta_1'| \otimes Z^{r_1}|+_{0,\theta_1'}\rangle\langle+_{0,\theta_1'}|Z^{r_1} \otimes_{i=1}^N \sum_{\theta_i} p(\theta_i)|+_{0,\theta_i}\rangle\langle+_{0,\theta_i}|$$

Summing first over r_1 and then over θ_1' we get the maximally mixed state for qudit 1 and its corresponding measurement vector.

Step N: Alice has also sent all measurement vectors for qudits up to i and Bob has applied an arbitrary CPTP-map attack and returned measurement result b_i after each step $1 \le i \le N$. For any fixed choice of b_i's:

$$\sum_{\substack{\theta_1,\dots,\theta_N,\\ r_1,\dots,r_N}} p(\theta_1,\dots,\theta_N,r_1,\dots,r_N)\mathcal{E}_N|\phi_N'+\theta_N+(r_N,0,0)^T\rangle\langle\phi_N'+\theta_N+(r_N,0,0)^T| \otimes \dots \otimes \mathcal{E}_1(|\phi_1+\theta_1+(r_1,0,0)^T\rangle\langle\phi_1+\theta_1+(r_1,0,0)^T| \otimes_{i=1}^N |+_{0,\theta_i}\rangle\langle+_{0,\theta_i}|)\dots)$$

And commuting all $\mathcal{E}_i$'s trivially and merging them to a global $\mathcal{E}$:

$$\sum_{\substack{\theta_1,\dots,\theta_N,\\ r_1,\dots,r_N}} p(\theta_1,\dots,\theta_N,r_1,\dots,r_N)\mathcal{E}(|\phi_N'+\theta_N+(r_N,0,0)^T\rangle\langle\phi_N'+\theta_N+(r_N,0,0)^T| \otimes \dots \otimes |\phi_1+\theta_1+(r_1,0,0)^T\rangle\langle\phi_1+\theta_1+(r_1,0,0)^T| \otimes_{i=1}^N |+_{0,\theta_i}\rangle\langle+_{0,\theta_i}|)$$

Changing variables: $\theta_i' = \theta_i + (r_i,0,0)^T$ for all i and rearranging the terms:

$$\sum_{\substack{\theta'_1,\ldots,\theta'_N,\\ r_1,\ldots,r_N}} p(\theta'_1,\ldots,\theta'_N,r_1,\ldots,r_N)\mathcal{E}(|\phi'_N+\theta'_N\rangle\langle\phi'_N+\theta'_N|\otimes Z^{r_N}|+_{0,\theta'_N}\rangle\langle+_{0,\theta'_N}|Z^{r_N}\otimes\ldots$$
$$\otimes|\phi_1+\theta'_1\rangle\langle\phi_1+\theta'_1|\otimes Z^{r_1}|+_{0,\theta'_1}\rangle\langle+_{0,\theta'_1}|Z^{r_1})$$

Notice that each ϕ'_i depends on some of $\{r_j : j<i\}$. Thus, we can start by summing over r_N which appears only in the terms Z^{r_N}, thus taking the maximally mixed state for qudit N. Then we can sum over θ'_N to get the maximally mixed state for the measurement vector of qudit N. Iteratively, following the inverse arithmetic order we sum over all random parameters taking the maximally mixed state as output.

We can add Bob's prior knowledge to the above proof as a probability distribution over all secret measurement vectors $\phi_i,\forall i$ and notice that there is the same distribution at the end of the protocol (method described in [Dunjko, 2012].

□

Lemma 8. *Protocol 4 where all measurement vectors have only Clifford elements ($\forall i, c_i = 0$) requires no classical communication and the measurements can be performed in one step.*

Proof. It suffices to prove the following statement which means that all Pauli X corrections can be written as Pauli Z corrections, which can all be performed on the final output returned to Alice.

$$\forall C : \text{diagonal Clifford operator}, \forall j \in F_q, \forall m_1 \in F_q, \exists m_2 \in F_q :$$
$$X^{m_1}C|+_j\rangle = Z^{m_2}C|+_j\rangle$$

This is true because:

$$C^\dagger X^{m_1}C|+_j\rangle = C^\dagger Z^{m_2}C|+_j\rangle \Leftrightarrow$$
$$Z^{m'_2}X^{m'_1}|+_j\rangle = C^\dagger Z^{m_2}C|+_j\rangle \Leftrightarrow$$
$$Z^{m'_2}\left(\sum_{i=0}^{q-1}\omega^{(i-m'_1)j}|i\rangle\right) = C^\dagger Z^{m_2}C|+_j\rangle \Leftrightarrow$$
$$Z^{m'_2}\left(\omega^{-m'_1 j}\sum_{i=0}^{q-1}\omega^{ij}|i\rangle\right) = C^\dagger Z^{m_2}C|+_j\rangle \Leftrightarrow$$

$$\omega^{-m_1' j} Z^{m_2'} |+_j\rangle = C^\dagger Z^{m_2} C |+_j\rangle$$

This is true up to a global phase for some m_2' since C is a diagonal Clifford, thus $C^\dagger Z^{m_2} C$ maps always to a diagonal Pauli.

□

5.3 d-level Verification Protocol

The main idea of the d-level VUBQC is the same as in 2-level system VUBQC. We hide the position of the traps by implementing a blind version of the computation and randomize the dummy qudits (which in this case are states of the generalized computational basis). Also, bridge qudits work in a similar way to bridge qubits. We list the d-level VUBQC in Protocol 5 without mentioning the specific graph or FT encoding used. Also we consider the version where we amplify the failure probability, to cover the case of quantum output, as usual.

Correctness coming from correctness of d-level UBQC and the pre-corrections for the neighbours of the d-level dummies.

The resources needed for Alice are to prepare and send single qudit states and apply the d-level Clifford decoding circuit for the QECC used, together with Pauli measurements. The communication requirement is $\widetilde{O}(n^2) \times O(log(1/\varepsilon))$ separable single qudit states sent from Alice to Bob off-line and $\widetilde{O}(n^2) \times O(log(1/\varepsilon))$ *dits* (d-level classical systems) of on-line classical communication between Alice to Bob, where n is the size of the computation. The comparison with 2-level verification is that while the number of rounds of communication remains the same, each round carries $log_2 d$ bits of information about the computation instead of one.

5.3.1 Verifiability Proof in d-level

Theorem 3. *Protocol 5 is* ε *verifiable with* $\varepsilon = (1 - \frac{(d-1)c'}{d})^{\mathbb{p}}$*, for some* $c' < 1$ *that depends on the graph,* $\mathbb{p}$ *is a parameter of FT and d the levels of the system with d an odd prime. Protocol 5 assumes the existence of the qudit version of the fault tolerant QECC used in FK [Raussendorf et al., 2007] or equivalent.*

Protocol 5 Qudit Verifiable Universal Blind Quantum Computation (VUBQC) Protocol with quantum output (based on FK)

Alice's input:

- Description of a computation in the MBQC model (or equivalent) using a convenient underlying open graph state (G,I,O). The computation is represented, for any vertex $i \in G \setminus O$, as a measurement vector φ_i (together with the set of X-dependences D_i^X and Z-dependences D_i^Z and a fixed partial order of measuring depending on the graph structure). The input is set to the state of n qudits: $|+_0\rangle^{\otimes n}$. Protocol can be extended to admit quantum input by applying techniques described in the FK protocol.

Alice's output:

- A system that contains the quantum output of the computation and a bit to indicate of Alice has accepted the output of the computation.

The protocol

1. Preprocessing 1: Alice translates the computation to a Fault Tolerant (FT) MBQC pattern that can correct errors on $\mathbb{p}$ qudits. Let the updated open graph be (G',I',O'), where $|G| = m'$ and $|I| = |O| = n'$.

2. Preprocessing 2: Alice embeds the encoded computation pattern into a suitable graph which has the following property: There exists a fixed order of measurement which respects the computational flow and each computational qudit belongs to a constant size subset of qudits S_γ in which a trap can be at any position with uniform random probability. The total number of qudits of the final graph is N. An example is the dotted complete graph of size $N = O(m'^2)$.

3. Alice prepares the rotated qudits. For $i = 1$ to N:

   (a) If qudit is a dummy: prepares $|d_i\rangle$, $d_i \leftarrow_R \mathbf{F}_d$.

   (b) If qudit is not dummy and not in O': prepares $\prod_{j \in N_G(i) \cap D} Z^{d_j} |+_{\theta_i}\rangle$, where $|+_{\theta_i}\rangle \equiv T^{\gamma_i} S^{\beta_i} Z^{\alpha_i} \frac{1}{\sqrt{2}}(|0\rangle + \ldots + |d-1\rangle)$, $\theta_i = (\alpha_i, \beta_i, \gamma_i)^T, \alpha_i, \beta_i, \gamma_i \leftarrow_R \mathbf{F}_d$.

   (c) If qudit is not a dummy and is in O' and is not a trap: Same as previous step but with $\theta = (r_i, 0, 0)^T, r_i \leftarrow_R \mathbf{F}_d$.

   (d) If qudit is not a dummy and is in O' and is a trap: Same as previous step but with $\theta = (r_i, \beta_i, 0)^T, r_i, \beta_i \leftarrow_R \mathbf{F}_d$.

   (e) She sends the qudit to Bob

Protocol 5 Cont'd

4. Bob entangles the states according to the graph state by applying generalized cZ gates.

5. Bob performs the rest of the computation using classical help from Alice. For i which ranges over all qudits (respecting the order given by the flow), except the qudits belonging to sets S_γ which contain qudits of the O' system:

   (a) Alice computes the actual measurement vector $\phi'_i = (\alpha'_i, \beta'_i, \gamma'_i)^T$ using the dependences and the previous measurement results ($\phi'_i = (0,0,0)^T$ for dummy qudits).

   (b) Alice chooses $r_i \leftarrow_R \mathbf{F}_d$ and computes $\delta_i = \phi'_i + \theta_i + (r_i, 0, 0)^T$.

   (c) Alice transmits δ_i to Bob.

   (d) Bob performs measurement $M_i^{\delta_i}$ on qudit i. Measurements on measurement vector $\delta_i = (\alpha''_i, \beta''_i, \gamma''_i)^T \in \mathbf{F}_d^3$ correspond to generalized measurements on basis $\{Z^j T^{\gamma''_i} S^{\beta''_i} Z^{\alpha''_i} \frac{1}{\sqrt{d}}(|0\rangle + \ldots + |d-1\rangle)\}_{j=0}^{d-1}$.

   (e) Bob transmits the result to Alice.

   (f) Alice corrects the result by adding $r_i \mod d$.

6. Bob returns the qudits of all the sets S_γ that contain the output system O' to Bob.

7. Alice applies the final Pauli corrections on qudits of O'.

8. Alice applies the decoding procedure of the FT encoding to produce actual output O.

9. Alice sets her indicator bit to accept if all trap tests where positive including the test on the traps of the returned system.

Proof. Everything is the same to the proof of Theorem 1 up to the point that we write the state as:

$$= \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 \sum_{\{t_i:t_i\leq N-n''\}} p(\{t_i : t_i \leq N-n''\})|\langle 0|_{t_i} P_{u|t_i}|0\rangle_{t_i}|^2) \sum_{\{t_i,\beta_{t_i}:t_i>N-n''\}} p(\{t_i,\beta_{t_i} : t_i > N-n''\}|\{t_i : t_i \leq N-n''\}) |\langle +_0|_{t_i} S^{\dagger\beta_{t_i}} P_{u|t_i} S^{\beta_{t_i}}|+_0\rangle_{t_i}|^2) \quad (5.16)$$

where β_{t_i} take uniform random values from $\mathbf{F}_d$ and P_u are tensor products of generalized Pauli+identity operators (generalized Pauli basis). Let us remind the reader that indices $u \in \mathcal{S}_2$ denote attacks that cannot be corrected by the FT QECC procedure (but will have an effect on the traps).

For attack with index u and each i where S_i contains qudits of the measured by Bob system, we denote by $w_{u,i}$ the number of positions in S_i that the independently detectable reduction of the attack $P_{u|t_i}$ has a non-zero generalized Pauli X component when written in the X^iZ^j decomposition (i.e. i is non-zero in this decomposition). For attack u and i where S_i contains returned qudits we denote by $w_{u,i}$ the number of positions in S_i that the independently detectable reduction of the attack $P_{u|t_i}$ is non-identity or equivalently has in the X^iZ^j decomposition either i or j (or both) are non-zero. For attacks with indices $u \in \mathcal{S}_2$ by definition: $\sum_i(w_{u,i}) > \mathbb{p}'$ where $\mathbb{p}'$ is the number of independently detectable errors of the qudit FT QECC. Also, for each $j \in S_i$ with an independently detectable error and for all possible selections of trap positions $\{t_l : \forall l \neq i\}$ that (for the S_l' that there are errors) coincide with independently detectable errors of u we have $p(t_i = j|\{t_l\}) \geq c'$.

Also we use the following property, which is easy to verify:

$$\forall t_i, P_{u|t_i} \neq I : \sum_{\beta_{t_i}} |\langle +_0|_{t_i} S^{\dagger\beta_{t_i}} P_{u|t_i} S^{\beta_{t_i}}|+_0\rangle_{t_i}|^2 \leq 1 \quad (5.17)$$

Using all the above:

$$\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 \prod_{\{i:\forall j\in S_i, j\leq n''\}} (c'(\frac{1}{c'} - w_{u,i})) \prod_{\{i:\forall j\in S_i, j>n''\}} (\frac{c'}{d}|a_{k,u}|^2(d\frac{1}{c'} - (d-1)w_{u,i}))$$

Or,

$$\begin{aligned}
&\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 \prod_i (1-\frac{(d-1)c'}{d} w_{u,i}) \\
&\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 \prod_i (1-\frac{(d-1)c'}{d})^{w_{u,i}} \\
&= \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 (1-\frac{(d-1)c'}{d})^{\sum_i (w_{u,i})} \\
&\leq \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2 (1-\frac{(d-1)c'}{d})^{\mathbb{p}'} \qquad (5.18)
\end{aligned}$$

Therefore state can be written as:

$$\sigma_2 \approx_\varepsilon p_1 |\psi_c\rangle\langle\psi_c| \otimes |ACC\rangle\langle ACC| + p_2 \rho \otimes |REJ\rangle\langle REJ| \qquad (5.19)$$

where

- $\varepsilon = (1-\frac{(d-1)c'}{d})^{\mathbb{p}'}$
- $p_1 + p_2 = \sum_{k,u\in\mathcal{S}_2} |a_{k,u}|^2$
- ρ is a density matrix.

Summing the terms σ_1 and σ_2, Theorem 3 is satisfied with $\varepsilon = (1-\frac{(d-1)c'}{d})^{\mathbb{p}'}$.

□

Chapter 6

An Efficient Verification Protocol

As mentioned before, the main idea for the composite protocol is that Bob uses the FK protocol to secretly prepare the polynomial-QAS encoded state used in the ABE protocol. Bob uses the localizing version of the FK protocol so he does not have to return the state to Alice, but rather he applies the polynomial-code logical circuit to implement the actual computation on the encoded state. In this composition of the localizing protocol with the ABE protocol we had to resolve a few issues which prevent a straightforward composition of the protocols.

Firstly, the state that Bob holds is encoded by the QECC used for the amplification of the detection probability in the FK protocol. In order to apply the logical circuit of the ABE protocol on this state, Alice needs to ask Bob to decode the first QECC, which from a verifiability perspective may seem problematic. Fortunately, the QECC of the FK protocol is based on a stabiliser code with a Clifford decoding circuit [Raussendorf et al., 2007]. The decoding circuit is publicly known and there is no need for any communication between Alice and Bob. An honest Bob performs the operators and Alice simply updates her quantum one-time-padding secret keys accordingly. Hence, no information is leaked on the secret keys, and also uniformity of the keys is preserved.

Secondly, the ABE protocol requires quantum states of prime dimension d where $d > 2$, whereas the original FK protocol uses qubits. To resolve this, we use the analogue of the localizing FK protocol which can use systems of any dimension while preserving all the security properties. The resources requirement for Alice is the ability of preparing random single qudit states and the overall communication complexity remains the same as the original FK protocol. We also examine the possibility of using the qubit version of the localizing FK protocol for Bob to prepare a state which can be translated to d-levels by a direct mapping of the basis. At the same time, we need to

investigate if the qubit one-time-pad that the output state always has on Bob's side, with a key hidden from Bob, can be translated to a qudit one-time-pad by local operations on Alice's side, without leaking any information to Bob. The no-go result, presented in 5.1 prevents any attempt to deterministically implement this translation of the one-time-pad. Therefore, we revert to our original approach to use the d-level version of the localizing FK protocol.

Thirdly, since the computation we use from the ABE protocol is based on gate teleportation, there is a constant number of rounds of classical communication between Alice and Bob per non-Clifford gate teleportation. We prove, however, that this communication does not leak any of the secret parameters of the trapification phase, crucial for the security of the composite protocol.

Finally, we prove that, since the detection and decoding procedure that Alice applies to the returned state of the ABE protocol is a CPTP map, the resulting state of the overall protocol is ε-close to the final state of the ABE protocol. Thus this state is also verifiable.

The above comprise tools sufficient to build a composite ε-verifiable protocol, however, this protocol would not yet give any improvement in terms of complexity over the existing protocols. To achieve this, we partition the underlying entangled state necessary for the FK protocol phase of the composite protocol into smaller separable sub-states, each used for the preparing of a polynomial-QECC encoded input state for the ABE protocol. Since the quadratic round complexity of the FK protocol comes from the complex structure of the overall required underlying resource state, simplifying the state for the purposes of this composition (observing that the polynomial-QECC encoded logical inputs are separable) succeeds in reducing the communication complexity. This modification does not violate the verification properties of the FK protocol since the fact that the states are separate is public knowledge and the detection and encoding procedures for each of the separate states can be done locally.

The steps which are essential for the construction of this protocol can potentially be used for constructing different composite schemes.

6.1 Impossibility of Qubit to Qudit Translation

Theorem 9. *It is impossible to have a deterministic translation (isometry) from a qubit system to a qudit system that spans all states of the qudit system and moreover translates a known full qubit one-time-pad to a full qudit one-time-pad, when the dimension of*

the qudit system is an odd prime. (Impossibility of composing qubit and qudit security protocols without returning the state to Alice to update the one-time-pad)

Proof. We require a basis translation of a qubit system of dimension 2^n to a system of dimension d, where $2^n \geq d$. This basis translation is represented by function $g : \{0,\ldots,2^n-1\} \to \{0,\ldots,d-1\}$, which needs to be surjective to be able to prepare any arbitrary state in the qudit space.

Moreover, the state in qubit space has a full quantum one-time-pad, which also needs to be translated from the qubit description to the qudit description. For the rest we consider only the Pauli X pad of the system, and we prove that it is impossible to achieve this translation and still have a full Pauli X qudit one-time-pad, even before considering the Pauli Z pad.

A Pauli X quantum one-time-pad on the qubit system is general Pauli X^{x_1}, where $x_1 \in \{0,\ldots,2^n-1\}$, applied on the state. By definition of general qubit Pauli X:

For any basis state $|b_1\rangle$, $b_1 \in \{0,\ldots,2^n-1\}$: $X^{x_1}|b_1\rangle = |\sum_{i=0}^{n-1} 2^i(x_1^i \oplus b_1^i)\rangle$.

where x_1^i, b_1^i are the i-th digits in the binary representation of x_1, b_1, and the addition is modulo two. Let us define: $h(x,b) \equiv \sum_{i=0}^{n-1} 2^i(x_1^i \oplus b_1^i)$. For a fixed b_1 it defines a bijection $h(\cdot, b_1)$ (in other words, for a fixed b_1 there is always a unique x_1 that gives any value $h(x,b)$). It is trivial to see that the following property holds:

$$h(x_1, h(x_1, b_1)) = b_1 \tag{6.1}$$

In the qudit system a Pauli X one-time-pad is a generalized Pauli X^{x_2} on the state, where $x_2 \in \{0,\ldots,d-1\}$. Generalized Pauli X is defined as:

$\forall |b_2\rangle$ where $b_2 \in \{0,\ldots,d-1\}$: $X^{x_2}|b_2\rangle = |b_2 + x_2 \bmod d\rangle$.

Let $f : \{0,\ldots,2^n-1\} \to \{0,\ldots,d-1\}$ represent the function that translates the X one-time-pad key from the qubit to the qudit representation.

To have a consistent one-time pad, for all basis b_1 and all keys x_1:

$$g(h(b_1, x_1)) = g(b_1) + f(x_1) \bmod d$$

This can be rewritten as:

$$f(x_1) = g(h(b_1, x_1)) - g(b_1) \bmod d \tag{6.2}$$

This equation should hold for any b_1 (the translation of the one-time-pad f should be independent of b_1) and thus it should hold also for $b_2 = h(x, b_1)$, i.e.

$$\begin{aligned} f(x) &= [g(h(x,h(x,b_1))) - g(h(x,b_1))] \mod d \\ &= [g(h(x,b_2)) - g(b_2)] \mod d \\ f(x) &= [g(b_1) - g(h(x,b_1))] \mod d \end{aligned} \tag{6.3}$$

From Equations (6.2) and (6.3) we see that

$$2[g(h(x,b_1)) - g(b_1)] \mod d = 0 \tag{6.4}$$

Given that d and 2 are coprimes by assumption $(d,2) = 1$ it follows that

$$\begin{aligned} g(b_1) &= g(h(x,b_1)) \\ f(x) &= 0 \end{aligned} \tag{6.5}$$

Recall that g has range within d. Moreover, since for every b_1, b_2 there exists x such that $b_2 = h(x,b_1)$ it follows

$$\begin{aligned} g(b_2) - g(b_1) &= 0 \text{ for every } b_1, b_2 \\ g(x) &= \text{ constant function} \end{aligned} \tag{6.6}$$

This means that neither $f(x)$ nor $g(x)$ are surjective, which was the requirement to be able to prepare any arbitrary input.

□

6.2 Composite Protocol

The proposed FK-ABE composite protocol combines all the techniques discussed so far, including the partitioning of the high complexity resource state of the FK protocol to sub-graphs, each used for the preparation of a separate encoded state for the ABE protocol, thus reducing the quantum and classical communication complexity.

The protocol, described here as Protocol 6, is based on a d-level version of Protocol 3, for d odd prime, which corresponds to the following changes (see also Chapter 5 for more detail):

- The state is entangled by applying generalized cZ gates.

- Alice's states $|+_{\theta_i}\rangle$ are replaced by states $T^{c_i}S^{b_i}Z^{a_i}\frac{1}{\sqrt{d}}(|0\rangle+\ldots+|d-1\rangle)$, where T are generalized '$\pi/8$' gates, S generalized phase gates, Z generalized Pauli Z gates and $a_i, b_i, c_i \leftarrow_R \mathbf{F}_d$ (except qutrits, where T gate is replaced by T_3 gate and $c_i \leftarrow_R \mathbf{F}_9$).

- Dummy states are generalized computational basis states $|d_i\rangle$, $d_i \leftarrow_R \mathbf{F}_d$

- Measurements on vector $(a'_i, b'_i, c'_i) \in \mathbf{F}_d^3$ correspond to generalized measurements on basis $\{Z^j T^{c'_i} S^{b'_i} Z^{a'_i}\frac{1}{\sqrt{d}}(|0\rangle+\ldots+|d-1\rangle)\}_{j=0}^{d-1}$. Measurements vectors are corrected to incorporate generalized Pauli X and Z corrections according to flow dependencies.

- Alice cancels the pre-rotation by adding (a_i, b_i, c_i) on her (corrected) measurement vector. In the measurement vector she sends to Bob, extra term $(r_i, 0, 0)^T$, $r_i \leftarrow_R \mathbf{F}_d$ is added to randomize the output of Bob's measurement. She corrects the measurement result she receives from Bob by adding r_i.

- The output gadget has the necessary number of layers to teleport the encoded output at a fixed position. Since the computation in the gadget is Clifford and it does not contain any part of the secret, the proof technique we have developed goes through.

Correctness of the d-level version of the FK protocol follows the same argument of the correctness of qubit case, applying d-level MBQC and noticing that the d-level dummy qudits have the same effect as in the qubit case. Universality comes from the fact that we are able to perform a universal set of gates for d-level computation: $\{cX, F, S, T, Z\}$, with the special case of qutrits where T becomes T_3.

6.2.1 Verifiability of the Composite Protocol

Protocol 6 is ε-verifiable with the proof given later in this section.

Theorem 4. *The composite protocol is correct and ε-verifiable, with $\varepsilon \leq \varepsilon_1 + \varepsilon_2$ where $\varepsilon_1 = \frac{1}{c^{\mathbb{p}_1}}$, $\varepsilon_2 = \frac{1}{2^{\mathbb{p}_2}}$ where $\mathbb{p}_1$ and $\mathbb{p}_2$ are the QECC distance parameters of the FK and ABE phase and $c > 1$ is a parameter of the FK phase which depends on the structure of the graph used.*

Protocol 6 Composite Verifiable Quantum Computation

Alice's input. Description of a computation in the Gate Teleportation model based on generalized Toffoli states. The input is set to be the Fourier basis state of n qudits: $|+_0\rangle^{\otimes n}$, where $|+_0\rangle = \frac{1}{\sqrt{d}}\sum_a |a\rangle$. The total number of gates is denoted by t. The total number of Toffoli gates is denoted by t'. Protocol can be extended to admit arbitrary quantum input using the same methods as in the FK protocol.

Alice's output. The result of measurement of the quantum output of the circuit and a bit indicating if the result is accepted or not.

The protocol

1. Alice chooses a single random sign key (of size equal to the size of a codeword) that will be used to encode all inputs (including Toffoli states) according to the signed-polynomial QECC.

2. A d-level fault tolerant QECC is selected for amplification of the detection probability of the FK protocol.

3. Alice prepares for the delegation of the preparation of the necessary Toffoli states and the encoding of both the inputs and the Toffoli states according to the ABE protocol QECC, randomized by the selected sign key. For each encoded state she selects a separate graph state that is the d-level version of the graph G''' (trapified graph+output gadget) described in Protocol 3. Therefore, the size of each sub-graph depends only on the security parameters of the protocol and not on the size of the computation t. Let the union of all these sub-graphs be denoted by $G^{(4)}$ and its size $N = O(n) \times O(t')$.

4. Bob (interacting with Alice) encodes all inputs and Toffoli states by executing Protocol 3 adapted for d-level systems. The output of Bob is encoded by the QECC used for amplification of the FK protocol and the final quantum one time pad, which depends on the secret parameters of the FK protocol and will referred to as the Pauli key. Alice holds an indicator bit set on accept or reject depending on the outcome of the traps.

Protocol 6 Cont'd

5. Bob applies the Clifford circuit for decoding the QECC used for amplification of the FK protocol and Alice updates her Pauli keys accordingly. This requires some extra randomized single qudits sent from Alice to Bob so that the output is encrypted with a uniform key.

6. Alice and Bob perform the logical operators that correspond to the desired computation on the encoded by polynomial QECC state as described in the ABE protocol. For each application of logical Toffoli gate, Bob sends measurement results to Alice and Alice calculates the actual correction to be performed on the state and sends it to Bob who performs it (see Figure 2.2). No new randomness is introduced at this stage.

7. Bob measures the output qudits in the computational basis and returns the measurement results to Alice.

8. Alice applies the detection and decoding procedure of the signed polynomial QECC and sets a second indicator bit accordingly.

9. Alice accepts if both indicator bits of the FK and ABE protocol phase of the protocol are set to accept, otherwise she rejects.

Finally, we consider the resources. In the FK-ABE composite protocol the quantum requirement for Alice is to prepare single qudit states of the form $T^{c_i}S^{b_i}Z^{a_i}|+_0\rangle$, for $a_i, b_i, c_i \leftarrow_R \mathbf{F}_d$, where d is an odd prime. The dimension d of each system is $O(1/log(\varepsilon))$. Alice has to send to Bob, at the beginning of the protocol before getting the computation description, $O(n\text{Polylog}(\frac{1}{\varepsilon}))$ separable single qudit states, where n is the size of the computation. After, Alice and Bob have to exchange only classical information in $O(n\text{Polylog}(\frac{1}{\varepsilon}))$ rounds, where each round includes a constant size message sent from Bob to Alice and a constant size message from Alice to Bob.

The proof is similar to the proof of the localising protocol, modified for d-level systems, except some notable differences:

- We analyse all the separate runs of FK (for the encoding of each ABE state) as one run of FK on a graph that is partitioned into identical sub-graphs and we prove that even if this structure is known to Bob, we get the same bound for verifiability, which depends on the FT encoding parameters and graph properties of a sub-graph.

- Since Bob has to perform an extra MBQC pattern to decode of the FT code used in the FK phase, Alice will provide him an extra system of $|+_0\rangle$'s rotated by a random generalized Pauli Z, a rotation which is used in the proof to reduce the attack on this extra system to a Pauli attack. Intuitively, these extra rotations are effectively one-time-padding the state even after the decoding operations with an independent pad on each qudit.

- In the ABE phase Alice has to provide to Bob a system of corrections for the implementation of the Toffoli gates. We reduce this system to the maximally mixed state, thus effectively providing no extra information of Bob (and again reducing the attack on this system to a Pauli attack).

- Since the detection procedure of the ABE phase is contractive (it is a CPTP-map) it cannot increase the failure probability of the first phase. Moreover, Alice has a second indicator bit to detect any Pauli attack that comes from the first or the second phase of the protocol applying on the correct encoded state by the randomized polynomial code.

Proof. Single index notation is followed to enumerate the qubits participating in the graph $G^{(4)}$ where N is the total number of qudits and the last n' qudits are the encoded

output qudits (system O' consisting of the qudits of the output layer of all the gadgets of all the sub-graphs).

Extra notation has to be introduced first. Vector ν is used to represent all random secret parameters, except the random sign key, chosen by Alice throughout the execution of protocol, including $r = \{r_i\}$ (there are also $O(n')$ extra r's used to encrypt the qudits sent for the decoding of the FK QECC), $\theta = \{\theta_i\}$, $d = \{d_i\}$ and positions of traps $t = \{t_i\}$. Parameter $p(\nu)$ gives the probability of a particular choice of random secret parameters. Summing over a vector (e.g. $\sum_r$) means that we sum over all possible choices for the elements of that vector (e.g. all possible dit-strings of size $N+O(n')$ for r).

For convenience we denote the subsystems of the joint Alice-Bob system:

- System $\mathcal{M}$ is the union of all the non-dummy, non-trap qudits of G'''s (therefore includes all the necessary bridge qudits introduced when embedding G' in G'') and the non-dummy qudits of the first layers of each gadget (the ones that are used to teleport the state to the fixed position in the final layer of the gadget) introduced when embedding G'' into G'''. Therefore $\mathcal{M}$ contains all the qudits of the graph state $G^{(4)}$ that are measured and participate in the computation. We call them measured computational qudits.

- System D contains all dummy qudits of $G^{(4)}$.

- System T contains all trap qudits of $G^{(4)}$ (here all of them are measured by Bob).

- System O' is the encoded output system at fixed position in $G^{(4)}$. These are all the qudits of the last layer of each gadget (the ones that are not measured by Bob).

- System Δ is the system of the measurement angles send by Alice to Bob, where each angle is represented by three qudits in the computational basis.

- System B is Bob's private system, assumed to be initially in the blank state.

- System $\mathcal{M}_A$ of auxiliary decoding qudits.

- System $\mathcal{L}$ contains all the corrections send from Alice to Bob during the ABE phase.

Each measurement performed at Step 6d of the FK protocol component is analysed into a unitary part and a generalized Pauli Z measurement. Without loss of generality

we can represent any dishonest behaviour of Bob at any step as applying the correct unitary operators and then an arbitrary unitary attack operator.

For any run of Protocol 6, for any choice of Alice's input computation (therefore for any choice of sign keys used in the ABE protocol phase), the output that Alice has at the end of the protocol before applying the ABE detection procedure, averaged over all random parameters except the sign key and for all measurement outcomes (b in graph $G^{(4)}$+decoding and $\widetilde{b}$ in the ABE phase) with the corresponding probabilities, is:

$$\begin{aligned}\rho_{out} = Tr_{B,\mathcal{M},\Delta,D,\mathcal{L}}(\sum_{b,\widetilde{b}}\sum_{\nu} p(\nu)C''(|b\rangle\langle b|\otimes|\widetilde{b}\rangle\langle\widetilde{b}|UC(\mathcal{D}(F_{N-n'}R_{N-n'}(.)\dots \\ F_1R_1(.)E_G(|M\rangle\langle M|\otimes|\delta\rangle\langle\delta|)E_GR_1(.)^\dagger F_1^\dagger\dots R_{N-n'}(.)^\dagger F_{N-n'}^\dagger\otimes|0\rangle\langle 0|^{\otimes B} \\ \otimes|M_A\rangle\langle M_A|)\mathcal{D}^\dagger\otimes|\widetilde{r}\rangle\langle\widetilde{r}|)C^\dagger U^\dagger|b\rangle\langle b|\otimes|\widetilde{b}\rangle\langle\widetilde{b}|)C'') \qquad (6.7)\end{aligned}$$

where:

- $R_i(\delta_i)$ corresponds the diagonal generalized Clifford operation $T^{c_i}S^{b_i}Z^{a_i}$ controlled by the state of vector δ_i.
- F_i is the Fourier gate applied on qubit i.
- $|M_A\rangle = \bigotimes_{i>N} Z^{r_i}|+\rangle$ is the auxiliary system sent from Alice to Bob for the implementation of the decoding/detection procedure by means of a unitary $\mathcal{D}$. These are qubits chosen are random from the qudit Hadamard basis (mixed state from Bob's perspective) and Alice needs to update the Pauli key of the final decoded output accordingly by using the commutation relations of Pauli operators with the Clifford circuit $\mathcal{D}$.
- $\mathcal{D}$ is the Clifford unitary used to decode the output O' of the FK protocol phase which is encoded by the QECC used by the FK protocol. The decoding is applied by Bob (as opposed to Alice in the case of normal FK) and let the decoded state be contained in system O which is a subsystem of systems O' and the system of qudits $|M_A\rangle$.
- $|\widetilde{b}\rangle\langle\widetilde{b}|$ implement the measurements for the Toffoli gates at the ABE protocol phase and apply on some of the qubits of system O (see Figure 2.2), which depicts the implementation of Toffoli gate with Toffoli state, Clifford operators and Pauli Z measurements and the corrections, which have to be performed by

Bob depending on his communication with Alice, are the Clifford operators in the dashed boxes).

- $\widetilde{r}(r,\widetilde{b})$ is the classical bit string (system $\mathcal{L}$) that Alice sends to Bob for the corrections of the Toffoli part of the ABE protocol.

- $\mathcal{C}(\widetilde{r})$ is the Clifford part of the ABE protocol phase that implements the public polynomial-QECC logical circuit on system O.

- $C''(r,b)$ contains the Pauli correction $C''_{O',\mathcal{M}_A}$ that Alice performs to the final returned system, that takes into account the updates of the keys due to the application of circuits $\mathcal{D}$ and $\mathcal{C}$. C'' also contains the corrections to trap measurement results.

- U represents global Bob's attack that in this case applies also to system of qudits $|M_A\rangle$ and system $\mathcal{L}$.

Alice's output can be rewritten, by decomposing the attack to the Pauli basis, tracing out system B and applying the honest computation on system T to separate it, as:

$$\begin{aligned}\rho_{out} = Tr_{\mathcal{M},\Delta,D,\mathcal{L}}\Big(\sum_{b,\widetilde{b}}\sum_{\nu} p(\nu) \sum_{k,u,v,u',v'} a_{k,u,v}a^*_{k,u',v'} C''_{O',\mathcal{M}_A}\big(|b'\rangle\langle b'|\otimes|\widetilde{b}\rangle\langle\widetilde{b}|P_{u|i:\forall j,i\neq t_j}\\ \otimes P_v \mathcal{C}\big(\mathcal{D}\big(\mathcal{P}'\big(|M'\rangle\langle M'|\otimes|\delta\rangle\langle\delta|\big)\mathcal{P}'^{\dagger}\otimes|M_A\rangle\langle M_A|\big)\mathcal{D}^{\dagger}\otimes|\widetilde{r}\rangle\langle\widetilde{r}|\big)\mathcal{C}^{\dagger}P_{u'|i:\forall j,i\neq t_j}\\ \otimes P_{v'}|b'\rangle\langle b'|\otimes|\widetilde{b}\rangle\langle\widetilde{b}|\big)C''^{\dagger}_{O',\mathcal{M}_A}\bigotimes_i |b_{t_i}+r_{t_i}\rangle\langle b_{t_i}|P_{u|t_i}|r_{t_i}\rangle\langle r_{t_i}|P_{u'|t_i}|b_{t_i}\rangle\langle b_{t_i}+r_{t_i}|\Big)\Big)\end{aligned} \quad (6.8)$$

where

- $a_{k,u,v}$ are complex numbers, with $\sum_{k,u,v}|a_{k,u,v}|^2 = 1$.

- P_u (and $P_{u'}$) ranges over all tensor products of d-level Pauli+identity operators and applies on the system that is trapified (all G'''s).

- P_v (and $P_{v'}$) ranges over all tensor products of d-level Pauli+identity operators and applies on the system that is not trapified (gadget systems $\times\mathcal{M}_A\times\Delta\times\mathcal{L}$).

- $P_{u|i}$ is the i-th generalized tensor element of P_u (similarly for P_v).

- b' is the vector that is generated from b by removing elements $\{b_{t_i}\}$.

The next step will be to eliminate all terms where $u \neq u'$ and $v \neq v'$. We notice that we can extract a random logical Pauli $Z^{r'_1} \otimes Z^{r'_2} \otimes Z^{r'_3}$ from state $|000\rangle$ that is used to prepare the Toffoli gate (Figure 2.2) and notice that all dependences on (r'_1, r'_2, r'_3) cancel in all systems except system $\mathcal{L}$. Then, summing over (r'_1, r'_2, r'_3) gives the maximally mixed state for system $\mathcal{L}$ and thus taking the trace of it cancels all cross terms of the corresponding attack. However, the dependence on $\widetilde{r}$ remains on $\mathcal{C}$.

For the system of auxiliary decoding qudits $|M_A\rangle$, random rotations Z^{r_i} can be used to twirl the attacks on this system.

The rest of the arguments for the elimination of the cross attack terms follow the corresponding part of the proof of Theorem 2. We only need to notice that when the r's in the measurement angles commute with $\mathcal{D}$ and $\mathcal{C}$ they cancel the dependence of $\mathcal{C}$ on r's and they can be used to twirl the attack on the output.

Thus, state ρ_{out} can be rewritten to eliminate all attack cross terms that are different.

$$\rho_{out} = \sum_{b,\widetilde{b}} \sum_{t} p(t) \sum_{k,u,v} |a_{k,u,v}|^2$$
$$C^{(3)}_{O'}(\langle b'| \otimes |\widetilde{b}\rangle\langle\widetilde{b}| P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin \Delta,D,\mathcal{L}\}} (\mathcal{C}'\mathcal{D}(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger} \otimes$$
$$|M'_A\rangle\langle M'_A|)\mathcal{D}^\dagger \mathcal{C}'^\dagger) P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin \Delta,D,\mathcal{L}\}} |b'\rangle \otimes |\widetilde{b}\rangle\langle\widetilde{b}|) C^{(3)\dagger}_{O'} \bigotimes_i P_{u|t_i}|0\rangle\langle 0|P_{u|t_i} \quad (6.9)$$

where

- $|M^{(4)}\rangle$ is the tensor product of $|+_0\rangle$ for all qudits of system $\mathcal{M} \times O'$.
- $\mathcal{P}^{(3)}$ is the unitary that contains E'_G for $\mathcal{M} \times O'$ and $F_i R_i(\phi'_i(b_{f^{-1}(i)}, b_{f^{-1}}(j \sim i, j \neq f(i)))$ operators for all qudits of system $\mathcal{M}$.
- $|M'_A\rangle$ is the tensor product of $|+_0\rangle$ for all qudits of system of auxiliary qudits used in the decoding procedure.
- $\mathcal{C}'(\widetilde{b})$ is the Clifford part of the ABE protocol phase that implements the public polynomial-QECC logical circuit on system O and depends only on the measurement outcomes $\widetilde{b}$ of the ABE phase.
- $C^{(3)}_{O'}$ are the corrections on the output of the ABE protocol phase (after eliminating the r's) and depend on b' and $\widetilde{b}$.

Restricting the summation to $u \in \mathcal{S}_1$ where $\mathcal{S}_1$ is the set of tensor products of d-level Pauli+identity operators that when we map to the operator with maximum independently detectable errors for the whole graph $G^{(4)}$ contain $\leq \mathbb{p}_1$ tensor elements which have a generalized Pauli X component when written in the $X^i Z^j$ decomposition (where $\mathbb{p}_1$ is the number of independent detectable errors of the code used in the FK phase) and applying the same steps as in the corresponding part of the proof of Theorem 2 (observing that the gadget again is a Clifford MBQC and thus we can rewrite the attack on the gadget as an attack on the output), the state becomes:

$$\sigma_1 = \sum_b \sum_t p(t) \sum_{k,u\in\mathcal{S}_1,v} |a_{k,u,v}|^2 C^{(3)}_{O'} P''_{v_1} P'_{v_2} (C^{(3)}_{O'} |\psi'_c\rangle\langle\psi'_c| C^{(3)}_{O'}) P'_{v_2} P''_{v_1} C^{(3)}_{O'}) \bigotimes_i P_{u|t_i}|0\rangle\langle 0|P_{u|t_i} \quad (6.10)$$

The fact that the graph is partitioned into sub-graphs, to encode the different logical states to be used in the ABE protocol phase does not change the above statement because a global attack with footprint $\leq \mathbb{p}_1$ will necessarily have footprint $\leq \mathbb{p}_1$ in each sub-graph. We can also eliminate $C^{(3)}_{O'}$ by commuting with the Pauli attack operators.

By separating the terms that leave all traps untouched and the rest we get:

$$\sigma_1 = \sum_i p'_i P_i |\psi'_c\rangle\langle\psi'_c| P_i \otimes |ACC_1\rangle\langle ACC_1| + \sum_j p''_j P_j |\psi'_c\rangle\langle\psi'_c| P_j \otimes |REJ_1\rangle\langle REJ_1| \quad (6.11)$$

where

- P_i (P_j) range over all tensor products of d-level Pauli+identity operators on the final returned system.
- p'_i , p''_i are probabilities with $\sum_i (p'_i + p''_i) = \sum_{k,u\in\mathcal{S}_1,v} |a_{k,u,v}|^2$.
- $|ACC_1\rangle\langle ACC_1|$ is the state where all trap qubits are $|0\rangle\langle 0|$ and $|REJ_1\rangle\langle REJ_1|$ denotes any state that at least one trap is $|1\rangle\langle 1|$.
- $|\psi'_c\rangle\langle\psi'_c|$ is the state, encoded by the signed polynomial QECC, after the correct computation of the ABE protocol phase is performed and undoing the Pauli key.

The rest of the terms of the summation in the state ρ_{out} ($u \in \mathcal{S}_2$) are considered. Let $P_\perp$ be the projection to the orthogonal space to the correct output state:

$$P_\perp = I - |\psi'_c\rangle\langle\psi'_c| \quad (6.12)$$

We calculate the 'bad' probability p_{bad} the state collapses to the 'incorrect' subspace and no trap is activated.

$$p_{\text{bad}} = Tr(\sum_{b,\widetilde{b}} \sum_{t} p(t)$$
$$P_{\perp} \bigotimes_{i} |0\rangle_{t_i}\langle 0|_{t_i} \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 C^{(3)}_{O'}(\langle b'| \otimes |\widetilde{b}\rangle\langle\widetilde{b}| P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D,\mathcal{L}\}}$$
$$(\mathcal{C}'\mathcal{D}(\mathcal{P}^{(3)}(|M^{(4)}\rangle\langle M^{(4)}|)\mathcal{P}^{(3)\dagger} \otimes |M'_A\rangle\langle M'_A|)\mathcal{D}^{\dagger}\mathcal{C}'^{\dagger}) P_{u|\{i:i\notin T,D\}} \otimes P_{v|\{i:i\notin\Delta,D,\mathcal{L}\}}$$
$$|b'\rangle \otimes |\widetilde{b}\rangle\langle\widetilde{b}|) C^{(3)\dagger}_{O'} \bigotimes_{i} P_{u|t_i}|0\rangle\langle 0|P_{u|t_i}) \quad (6.13)$$

Tracing out everything except the trap system:

$$\leq Tr(\sum_{t} p(t)|0\rangle_{t_i}\langle 0|_{t_i} \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2 \bigotimes_{i} P_{u|t_i}|0\rangle_{t_i}\langle 0|_{t_i} P_{u|t_i}) \quad (6.14)$$

The rest is the same to the corresponding part in the proof of Theorem 2 (having a d-level system does not affect the result in the case of no output traps, as it is the case here) giving the final state:

$$\sigma_2 \approx_{\sqrt{(1-c')^{\mathbb{p}_1}}} p_1|\psi'_c\rangle\langle\psi'_c| \otimes |ACC_1\rangle\langle ACC_1| + p_2\rho \otimes |REJ_1\rangle\langle REJ_1| \quad (6.15)$$

where $p_1 + p_2 = \sum_{k,u\in\mathcal{S}_2,v} |a_{k,u,v}|^2$ and ρ is a density matrix.

By summing the states σ_1 and σ_2 and by contractivity of the ABE protocol detection procedure that Alice applies to check if the state is in the valid space of the signed polynomial QECC, Theorem 4 is satisfied with $\varepsilon_1 = \sqrt{(1-c')^{\mathbb{p}_1}}$ and $\varepsilon_2 = (1/2)^{\mathbb{p}_2}$.

□

6.2.2 Alternative Composition with Toffoli Inputs

An alternative version of the composite protocol could be considered if we modify the restrictions imposed on the verifier. In particular the verifier must be a preparation-only device (as before) that can generate single qudit states $U_E|+_0\rangle$ and encrypted generalized Toffoli states of the form $U_{E,1} \otimes U_{E,2} \otimes U_{E,3}|\text{Toffoli}\rangle$, where $U_E, U_{E,i}$ are single qudit gates classically controlled by the random keys and will be given explicitly later. This verifier is not yet capable of universal computation because she does not have memory and quantum measurement capabilities. The important difference in using this

verifier comes from the fact that the encoding phase, which prepares the verifiable states on which the ABE Protocol applies on and has to be delegated to the prover, is now a Clifford computation. The question is, do we still need the verifiable blind protocol or the blind protocol in the preparation of the verifiable encoded states is enough to get global verifiability. In the latter case we could simplify the graph, using a planar graph used in the blind protocol such as the brickwork state. The intuitive reason that this simplification might be possible is that in the case of the Clifford circuit, any Pauli attack at any time step of the computation can be written as a Pauli attack on the output of the computation, which is one the requirements for the protocol to work.

Unfortunately, there is a reason why this simplification does not work. When we rewrite the Pauli attacks before any measurement as Pauli attacks on the quantum output, the correction operators will depend in general on the underlying computation. In the case of the composite protocol this computation is different for any choice of the random sign key, therefore the attack at the output will, in general, depend on the secret and verification of it is not possible.

We give the description of the protocol in Protocol 7, and a proof that this protocol is correct and verifiable. The benefit of this protocol is that the MBQC part of the delegated computation is Clifford and therefore can be run in one round. The Toffoli-depth of the overall delegated computation gives us the total communication round complexity. Toffoli-depth represents the depth of our computation in the circuit model if we count only the Toffoli gates. Similarly, the more usual T-depth represents the depth of the T or $\pi/8$ gates in the circuit representation.

Other directions to simplify the original composite protocol might take into consideration the fact that the preparation of the verifiable states is a known computation (varies only for the choices of the random key), therefore there might be a more efficient way of placing the traps, simplifying the underlying graph used.

Theorem 5. *Protocol 7 is ε-verifiable. The round complexity of Protocol 7 is the Toffoli-depth n' of the delegated computation which corresponds to cn' T-gate depth, for some small constant c.*

Protocol 7 Composite Verifiable Quantum Computation with Toffoli inputs

Alice's input. Description of a computation in the Gate Teleportation model based on generalized Toffoli states. The input is set to be the Fourier basis state of n qudits: $|+_0\rangle^{\otimes n}$, where $|+_0\rangle = \frac{1}{\sqrt{d}}\sum_a |a\rangle$. The total number of gates is denoted by t and the number of Toffoli gates within them is t'. Protocol can be extended to admit arbitrary quantum input using the same methods as in the FK protocol.

Alice's output. The result of measurement of the quantum output of the circuit and a bit indicating if the result is accepted or not.

The protocol

1. Alice chooses a single random sign key (of size equal to the size of a codeword) that will be used to encode all inputs (including Toffoli states) according to the signed-polynomial QECC.

2. Alice has to prepare a number of generalized Toffoli states, equal to the number of Toffoli gates of the circuit and apply the following operation on them:

$$X^{x_i}Z(\theta_i)\otimes X^{x_{i+1}}Z(\theta_{i+1})\otimes X^{x_{i+2}}Z(\theta_{i+2})|\text{Toffoli}\rangle \tag{6.16}$$

   where $x_i, x_{i+1}, x_{i+2} \leftarrow_R \{0,1\}^3$ and $i = [1, 3t']$. Also Alice adapts the computation so that she corrects for the X pad, by adjusting for each i the measurement angle of the first layer qudit to $(-1)^{x_i}\phi$ and of the second layer qudit to $\phi + x_i\pi$.

3. Alice prepares for the delegation of the encoding of both the inputs and the Toffoli states. The encoding will be according to the ABE protocol QECC randomized by the selected sign key. For each encoded state she selects a separate graph state that is a dotted-complete state.

4. Bob (interacting with Alice) encodes all inputs (including Toffoli states) by executing d-level VUBQC with localizing gadget. The output of Bob is encoded only by the final quantum one time pad, which depends on the secret parameters of the FK protocol and will referred to as the Pauli key.

5. Bob applies the Clifford circuit for decoding the QECC used for amplification of the FK protocol and Alice updates her Pauli keys accordingly.

Protocol 7 Cont'd

6. Alice and Bob perform the logical operators that correspond to the desired computation on the encoded by polynomial QECC state as described in the ABE protocol. For each application of logical Toffoli gate, Bob sends measurement results to Alice and Alice calculates the actual correction to be performed on the state and sends it to Bob who performs it.

7. Bob measures the output qudits in the computational basis and returns the measurement results to Alice.

8. Alice applies the detection and decoding procedure of the signed polynomial QECC and sets a second indicator bit accordingly.

9. Alice accepts if both indicator bits of the FK and ABE protocol phase of the protocol are set to accept, otherwise she rejects.

Proof. The proof of Theorem 5 follows the same steps of the proof of Theorem 4 with the only difference being that some of the inputs for the FK phase sent from Alice to Bob are encrypted generalized Toffoli states (as opposed to single qudit rotated states).

In particular the set of random parameters ν now includes extra random parameters $\{x_i\}$ for $i = [1, 3t']$ used for the Pauli X random rotation of the Toffoli inputs. The total quantum state $|M\rangle\langle M|$ sent by Alice to Bob will include the system of t' encrypted Toffoli states. Finally, the measurement angles of system Δ need to be modified so that the first two layers of computation cancel the Pauli X rotation, and therefore Δ depends also on $\{x_i\}$.

All the steps of the proof will be the same except the twirling of the input Toffoli qudits which consumes the parameters $\{x_i\}$. Therefore, when we want to cancel the cross attack terms ($u \neq u'$) in the computational graph and in particular:

Qubit $i \in I$ and part of a Toffoli state: We assume that E''_G does not contain entangling operators between the input qubits (e.g. brickwork graph). We can extract X^{x_i} from $|M^{(4)}\rangle$ ($\langle M^{(4)}|$), having the remaining state represented by $|M^{(5)}\rangle$ ($\langle M^{(5)}|$). Commuting, on both sides, X^{x_i} with E''_G will give a Z^{x_i} correction on qudit $f(i)$ which will undo the pre-existing $x_i\pi$ correction on measurement angle $\phi_{f(i)}$ of the $f(i)$ system. On system i again, commuting X^{x_i} with $R_i((-1)^{x_i}\phi_i)$ (due to the pre-existing correction) will undo the $(-1)^{x_i}$ correction and then commuting with H_i will change it to Z^{x_i}. Now the only

dependence on x_i is on the random Z element we have commuted to the point before the attack.

Let us also employ the following trick for generating the random X elements: We extract stabilizer $Z_i^{r'_i} X_{f(i)}^{r'_i} Z_{j\sim f(i), j\neq i}^{r'_i}$ for $r'_i \leftarrow_R \{0,1\}$ from graph state $E''_G|M^{(5)}\rangle$ and at the same time changing variable $\hat{b}_i \leftarrow b_i + r'_i$ everywhere. The new terms cancel everywhere except at qubit i so that we get:

$$\sum_{r'_i, x_i} \langle b_i | X^{r'_i} Z^{x_i} P_{u|i} Z^{x_i} X^{r'_i} H_i R_i(\phi_i)$$

$$E''_G(|M^{(5)}\rangle\langle M^{(5)}|) E''_G R_i(\phi_i)^\dagger H_i X^{r'_i} Z^{x_i} P_{u'|i} Z^{x_i} X^{r'_i} |b_i\rangle = 0 \text{ if } P_{u|i} \neq P_{u'|i} \qquad (6.17)$$

The rest is exactly the same (except for replacing all $|M^{(4)}\rangle$ by $|M^{(5)}\rangle$) as in the proof of Theorem 4.

□

6.3 Noise and Abstract Security

Real implementations of any verification protocol should take into account the effect of noise. We emphasise, once more, that the fault tolerance discussed so far in all the trap-based protocols is not used to tackle the noise but to amplify the verification probability, by forcing the adversary to attack more physical qub(d)its. But the noise will be present in any implementation, in the form of a noisy quantum preparation device of the verifier, a noisy quantum channel between the verifier and the prover and the errors that are introduced by the prover's devices at each step of the computation (entangling, measurement). This will have an effect in both the correctness and the verifiability probability of the protocol.

The existence of noise necessitates the use of an extra layer of fault tolerance in the execution of the protocol. This can be achieved in the same way fault tolerance was implemented for probability amplification and described in Section 3.1.1; a topological code and an extra correction step after each T gate to deal with the adaptive measurements. An alternative method to perform blindly the adaptive Z measurements can be found in [Morimae and Fujii, 2012] where the authors use a modified - *decorated* by extra vertices - RHG lattice on prover's side. Note that in all cases the verifier is the party who does the classical processing for the error correction, by using the prover's measurement results. In [Chien et al., 2013] the authors consider the capability of being

able to correct the errors on qubits during transmission from verifier to prover (e.g. useful when many repeaters mediate the transmission). This is achieved by having the verifier use the Steane code to prepare the logical $|+_\theta\rangle_L$ states and the prover to apply the corresponding logical circuit [*]. Note that these fault tolerant implementations are provided for the blind-only protocols, but can be readily adapted for verification protocols, up to the following implication discussed in [Gheorghiu et al., 2015].

Errors will inevitably affect the traps which are single qubits; this can be counteracted by two techniques, one can allow some threshold of acceptance of erroneous trap measurements and therefore boost correctness of the protocol (but deteriorate verifiability). A second technique will be to encode both the computation, the dummy qubits and the traps (therefore the whole of the dotted-complete state) using the same topological code and consequently boost both correctness and verifiability. These techniques can be used in our composite protocol, in particular for the FK part of it. For the ABE part we can use the fault tolerant implementation proposed for it in the original paper: the prover adds his own encoding on the received states together with purification techniques that assume extra classical communication complexity.

A crucial property of any security protocol is to be able to use it in a larger context (e.g. in conjunction with other security protocols) and still trust its security properties will remain intact. In this thesis we provided one such composition, in the form of the Composite Verification protocol, and shown how the composite protocols retain their properties. A more systematic way of dealing with the issue of composability is studied in the context of abstract cryptography (AC) [Dunjko et al., 2014]. To demonstrate the need of such a construction, imagine that at the end of the protocol prover learns whether verifier has accepted or rejected the final output. Then, if the verification procedure depends on the input, learning the indicator bit might reveal something about the input, thus break the blindness property of the protocol. In [Dunjko et al., 2014] the notion of independent verifiability is defined, which requires the ability to find a CPTP map for the prover that can generate a state where, if we trace out verifier's private system, this state is ε-close to a state where the prover has learned the indicator bit. Therefore, if prover can produce the indicator bit himself, clearly learning it does not add anything new to his knowledge. Some protocols, such as FK, have been demonstrated to satisfy this stronger notion of verifiability. Also, in [Dunjko et al., 2014] it is proven that if a protocol is blind and independent verifiable, then it is (composable) blind-verifiable.

[*] An alternative version assigns the task of preparing the logical qubits to prover who sends 8 of them to verifier who has to chose one at random and send it back to prover, thus having a simpler verifier

An alternative, but equivalent in power, composability framework is Universal Composability (UC), introduced by [Unruh, 2010]. The QAS-based protocols are proven to be composable [Broadbent et al., 2012] according to the UC framework. Since all the protocols we use in our construction are proven to be composable in one framework or the other, we believe that the same should hold for the composite protocol, but, at the moment, this is left as a topic for future work.

Part III

Quantum-Intermediate Verification

Chapter 7

Overview

The physical realisation of quantum information processing requires the fulfilment of the five criteria collated by DiVincenzo [DiVincenzo, 2000]. While enormous progress had been made in realising them since, we are still some way from constructing a universal quantum computer. This raises the question whether quantum advantages in computation are possible without fulfilling one or more of DiVincenzo's criteria. From a more foundational perspective, the computational power of the intermediate models of quantum computation are of great value and interest in understanding the computational complexity of physical systems. Several such models are known, including fermionic quantum computation [Bravyi and Kitaev, 2002], instantaneous quantum computation [Bremner et al., 2010], permutational quantum computation [Jordan, 2010], and boson sampling [Aaronson and Arkhipov, 2011].

These models of quantum intermediate computation are considered easier to implement, yet their verification problem has not attracted much attention. In approaches such as [Aaronson and Arkhipov, 2014] or [Shepherd and Bremner, 2009], the issue of distinguishing between the output of a boson sampling device or an instantaneous quantum computer and a classical device has been tackled. In this thesis, however, we are interested in verifying that the output produced by the device is the correct output. As it was the case with the universal quantum computer, there exist some problems solvable by quantum intermediate computers that are believed to be outside NP, and therefore not efficiently verifiable using a witness. On the other hand, the correctness of the output of Shor's factoring algorithm [Shor, 1997] can be checked efficiently on a classical machine. In this part of the thesis we consider a verifier close to classical and a prover able to solve all problems in a quantum intermediate model A. We will therefore employ general techniques to verify the result of any problem belonging in this class A.

One of the earliest restricted models of quantum computation was proposed by Knill and Laflamme, named 'Deterministic Quantum Computation with One quantum bit (DQC1)', also referred to as the one pure qubit model [Knill and Laflamme, 1998]. It addresses the challenge of DiVincenzo's first criterion, that of preparing a pure quantum input state, usually the state of n separate qubits in the computational basis state zero. Instead, in the DQC1 model, only one qubit is prepared in a pure state (computational basis zero state) and the rest of the input qubits exist in the maximally mixed state. This model corresponds to a noisier, more feasible experimental setting and was initially motivated by liquid-state NMR proposals for quantum computing. The DQC1 model was shown to be capable of estimating the coefficients of the Pauli operator expansion efficiently. Following this, Shepherd defined the complexity class 'Bounded-error Quantum 1-pure-qubit Polynomial-time (BQ1P)', to capture the power of the DQC1 model [Shepherd, 2006], and proved that a special case of Pauli operator expansion, the problem of estimating the normalised trace of a unitary matrix to be complete for this class. This problem, and others that can be reduced to it, such as the estimation of the value of the Jones polynomial, is interesting from a complexity theoretical point of view since it has no known efficient classical algorithm. Moreover, they are not known to belong to the class NP, therefore the problem of verifying the correctness of the result is non-trivial. The estimation of fidelity decay in chaos, the estimation of the value of the Jones and other knot-invariant polynomial at the fifth root of unity for the trace closure of knots, partition functions of spin models, enumeration of quadratically signed weight enumerators and other interesting problems can be reduced to the estimation of the normalised trace of a unitary matrix. For more on such connections, see Ref. [Datta and Shaji, 2011].

The approach of the Verifiable Universal Blind Quantum Computing [Fitzsimons and Kashefi, 2012], already presented in this thesis, is based on a blind protocol where the verifier is able to prepare single qubits. In this part, we take the same approach towards verification by first adapting this existing protocol for blind computing to the DQC1 model. Thus, the first goal is to define what it means to have a DQC1 computation in the MBQC setting. Fixing the input state to almost maximally mixed as it is done in the circuit picture of the DQC1 model does not suffice since the required auxiliary qubits for MBQC could potentially increase the number of pure qubits in the system by more than a logarithmic amount *. This adaptation is necessary as currently

*Increasing the number of pure qubits in the input to the order of logarithmic in the size of the computation is shown not to add extra power to the one pure qubit complexity class [Shepherd, 2006].

all the optimal schemes [Aharonov et al., 2010, Broadbent et al., 2009, Dunjko et al., 2012, Morimae et al., 2015, Barz et al., 2012, Morimae and Fujii, 2012, Morimae, 2012, Morimae and Fujii, 2013, Sueki et al., 2013, Mantri et al., 2013, Giovannetti et al., 2013] for the blind computation exploit the possibility of adaptive computation based on the measurement, a freedom not allowed in the original DQC1 model. The main results presented in this Part are the following:

- We introduce a new definition of DQC1 computation within the MBQC framework, called the DQC1-MBQC model [†], which captures the essential property of its original definition in the circuit model. Moreover, we show that the original definition of complexity class BQ1P is contained in DQC1-MBQC, where the latter is able to capture the process where new qubits are introduced or traced out during the execution of the computation.

- We provide a sufficient condition for a graph state (underlying resource for an MBQC computation [Hein et al., 2004]) to be usable within DQC1-MBQC. A direct consequence of this is that the universal blind protocol, which satisfies this condition, can be directly adapted to the setting where the server is a DQC1-MBQC machine and the client is able to send one single qubits at a time.

- Building on the blind protocol and adapting the methods presented in [Fitzsimons and Kashefi, 2012] (FK protocol), a verification protocol for the class DQC1-MBQC is given, where the probability of the client being forced to accept an incorrect result can be adjusted by setting the security parameter of the model. Since the FK protocol does not satisfy the sufficient condition and hence not runnable in the DQC1-MBQC, an alternative method is presented which also leads to different complexity results.

7.1 Preliminaries

We define the class BQ1P formally as introduced by Shepherd [Shepherd, 2006], so that we can later recast it into the MBQC framework.

Definition 11 (Bounded-error Quantum 1-pure-qubit Polynomial-time complexity class). *[Shepherd, 2006] BQ1P is defined using a bounded-error uniform family of quantum*

[†] We use a different acronym than DQC1 to emphasis the structural distinction with the standard DQC1 model.

circuits – DQC1. A DQC1 circuit takes as input a classical string x, of size n, which encodes a fixed choice of unitary operators applied on a standard input state $|0\rangle\langle 0| \otimes I_{w-1}/2^{w-1}$. *The width of the circuit w is polynomially bounded in n. Let* $Q_n(x)$ *be the result of measuring the first qubit of the final state of a DQC1 circuit. A language in BQ1P is defined by the following rule:*

$$\forall a \in L : Pr(Q_n(a) = 1) \geq \frac{1}{2} + \frac{1}{2q(n)} \tag{7.1}$$

$$\forall a \notin L : Pr(Q_n(a) = 1) \leq \frac{1}{2} - \frac{1}{2q(n)} \tag{7.2}$$

for some polynomially bounded q(n).

In the same paper a problem named *Trace Estimation* is given and proven to be complete for this class (under classical logspace reductions). Trace Estimation it is a decision problem which takes as input a unitary in a polynomial-size description in the form of a circuit. Given the promise that the real part of the trace of a unitary is polynomially bounded away from zero we need to decide whether the sign of the real part of the trace of the matrix is positive. There is yet no known efficient classical algorithm to solve this problem, which makes the one-pure-qubit computer a candidate to demonstrate quantum supremacy.

7.2 Main Results

An essential physical property of DQC1 that we mean to preserve in DQC1-MBQC is its limited purity. To capture this we introduce the *purity parameter*:

$$\pi(\rho) = \log_2 (\mathrm{Tr}(\rho^2)) + d, \tag{7.3}$$

where d is the logarithm of the dimension of the state ρ. For a DQC1 circuit with k pure qubits, at each state of the computation the value of purity parameter π for that state remains constant equal to k. In fact, Shepherd showed that the class BQ1P is not extended by increasing the number of pure input qubits logarithmically. Thus, a purity that does not scale too rapidly with the problem size still remains in the same complexity class.

A characterisation of MBQC patterns compatible with the idea of the DQC1 model as introduced above is presented next. Any MBQC pattern is called DQC1-MBQC when there exists a runnable rewriting of this pattern such that after every elementary operation (for any possible branching of the pattern) the purity parameter π does not

increase over a fixed constant. We assume that the system at the beginning has only the input state and at the end has only the output state.

We define a new complexity class that captures the idea of one pure qubit computation in the MBQC model. This complexity class, that we name DQC1-MBQC, can be based on any universal DQC1-MBQC resource pattern, which is defined analogously to the DQC1 circuits [Shepherd, 2006] as a pattern that can be adapted to execute any DQC1-MBQC pattern of polynomial size. A particular example of such a resource, as we will present later, can be built using the brickwork state of [Broadbent et al., 2009] designed for the purpose of universal blind quantum computing. The input to a universal pattern is the description of a computation as a measurement angle vector and is used to classically control the measurements of the MBQC pattern. The quantum input of the open graph is always fixed to a state that has a constant number of pure qubits and the rest of the state is the maximally mixed state, in correspondence to the DQC1 model.

Definition 12. *A language in DQC1-MBQC complexity class is defined based on a universal DQC1-MBQC resource pattern P_α that takes as input an angle vector α of size n and is applied on the quantum state $|+\rangle\langle+| \otimes I_{w-1}/2^{w-1}$, $w \in O(n)$. A word α belongs to the language depending of the probabilities of the measurement outcome ($R_n(\alpha)$) of the first output qubit of pattern P_α which are defined identically to Definition 11:*

$$\forall a \in L : Pr(R_n(\alpha) = 1) \geq \frac{1}{2} + \frac{1}{2r(n)} \tag{7.4}$$

$$\forall a \notin L : Pr(R_n(\alpha) = 1) \leq \frac{1}{2} - \frac{1}{2r(n)} \tag{7.5}$$

for some polynomially bounded r(n).

Corollary 2. *BQ1P $\subseteq$ DQC1-MQBC.*

Proof. Any circuit description using a fixed set of gates can be efficiently translated into a measurement pattern applicable on the brickwork state. A specific example of translating each gate from the universal set {Hadamard, $\pi/8$, c-NOT} to a 'brick' element of the brickwork state is given in [Broadbent et al., 2009]. The quantum input state in the resulting measurement pattern is in the almost-maximally-mixed state, therefore the pattern is a valid DQC1-MBQC pattern. □

Definition 13. *An MBQC pattern is a DQC1-MBQC pattern if there is a runnable sequence of commands where for every elementary command and measurement outcome,*

there exists a fixed constant value c such that the overall quantum state of the system (ρ_i with dimension d_i) after the i^{th} operation satisfies the following relation

$$\pi(\rho_i) < \pi(\rho_{in}) + c, \tag{7.6}$$

where ρ_{in} is the quantum input of the pattern with dimension d_{in}, which is fixed to be the product of c_{in} (constant) pure qubits in state $|+\rangle$ and a maximally mixed state of $d_{in} - c_{in}$ qubits.

The above definition captures the essence of DQC1 in that it maintains a low purity, high entropy state in MBQC, in contrast to DiVincenzo's first criterion. Regarding implementation, the difference from the original DQC1 is that in DQC1-MBQC you need to be able to inject or trace out qubits during the execution of the computation. We derive a sufficient condition (that is also constructive) for the open graph state leading to DQC1-MBQC, capturing the universal blind quantum computing protocol as a special case. However, a general characterisation and further structural link with determinism in MBQC [Danos and Kashefi, 2006, Browne et al., 2007, Mhalla et al., 2014] is left as an open question for future work.

Theorem 10. *Any measurement pattern on an open graph state (G,I,O) with flow $(f,\preceq)$ (as defined in Definition 2) and measurement angles α where either $|I| = |O|$ or the flow function is surjective and all auxiliary preparations are on the $(X-Y)$ plane represents a DQC1-MBQC pattern.*

The full details and the proof of this theorem is provided in Section 8.1.

A direct consequence of Theorem 10 is that the Universal Blind Computing Protocol (UBQC) introduced in [Broadbent et al., 2009] can be easily adapted to fit within the DQC1-MBQC class, since it is based on an MBQC pattern on a graph state with surjective flow.

In the blind cryptographic setting a client (Alice) wants to delegate the execution of an MBQC pattern to a more powerful server (Bob) and hide the information at the same time. The UBQC protocol is based on the separation of the classical and quantum operations when running an MBQC pattern. The client prepares some randomly rotated quantum states and sends them to the server and from this point on the server executes the quantum operations on them (entangling according to the graph and measuring) and the client calculates the measurement angles for the server and corrects the measurement outcomes she receives (to undo the randomness and get the correct result).

To define blindness formally (see Definition 6) we allow Bob to deviate from the normal execution in any possible way, and this is captured by modelling his behaviour during the protocol by an arbitrary CPTP map. The main requirement for blindness is that for any input and averaged over all possible choices of parameters by Alice, Bob's final state can always be written as a fixed CPTP map applied on his initial state, thus not offering any new knowledge to him.

To adapt the original UBQC protocol into the DQC1-MBQC setting we change the order of the operations so that the client does not send all the qubits to the server at the beginning, but during the execution of the pattern, following a rewriting of the pattern that is consistent with the purity requirement. The details are described in Section 8.1.1. Alice is considered weak because, even though she can prepare a polynomial number of pure qubits overall, she cannot store more than one pure qubit at any time, while Bob is more powerful because he has also the capacity to perform unitary computation and measurements.

Theorem 11. *There exists a blind protocol for any DQC1-MBQC computation where the client is restricted to BPP and the ability to prepare single qubits and the server is within DQC1-MBQC.*

In the verification cryptographic setting a client (Alice) wants to delegate a quantum computation to a more powerful server (Bob) and accept if the result is correct or reject if the result is incorrect (server is behaving dishonestly). The main idea of the original protocol of [Fitzsimons and Kashefi, 2012] is to test Bob's honesty by hiding a trap qubit among the others in the resource state sent to him by Alice. Blindness means that Bob cannot learn the position of the trap, nor its state. During the execution of the pattern Bob is asked to measure this trap qubit and report the result to Alice. If Bob is honest this measurement gives a deterministic result, which can be verified by Alice. Bob being dishonest means that Alice will receive the wrong result with no-zero probability. Depending on that result, Alice accepts or rejects the final output received by Bob.

To define verifiability formally in the case of OPQ (building on Definition 7) we need to establish an important difference with the original protocol [Fitzsimons and Kashefi, 2012]: In a DQC1-MBQC pattern the quantum input is in a mixed state as opposed to a pure state. Reverting to the original definition that was presented in Section 1.2 we need to add an extra reference system R, that is used to purify the mixed input that exists in Alice's private system A. The assumption is that Bob does not learn

anything about the reference system (ex. Alice is provided with the quantum input from a third trusted party which also holds the purification). Bob is allowed to choose any possible cheating strategy and our goal is to minimise the probability of Alice accepting the incorrect output of the computation at the end of the protocol. Similarly to the original definition:

Definition 14. *A protocol for delegated computation is* ε*-verifiable (*$0 \leq \varepsilon < 1$*) if for any choice of Bob's strategy* j*, it holds that for any input of Alice:*

$$Tr(\sum_{\nu} p(\nu) P^{\nu}_{incorrect} B_j(\nu)) \leq \varepsilon \tag{7.7}$$

where $B_j(\nu)$ *is the state of Alice's system A together with the purification system R at the end of the run of the protocol, for choice of Alice's random parameters* ν *and Bob's strategy* j*. If Bob is honest we denote this state by* $B_0(\nu)$*. Let* $P_\perp$ *be the projection onto the orthogonal complement of the the correct purified quantum output. Then,*

$$P^{\nu}_{incorrect} = P_\perp \otimes |\eta_t^{\nu_c}\rangle\langle\eta_t^{\nu_c}| \tag{7.8}$$

where $|\eta_t^{\nu_c}\rangle$ *is a state that indicates if Alice accept or reject the result.*

A verification protocol should also be correct, which means that in case Bob is honest Alice's state at the end of the run of the protocol is the correct output of the computation and an extra qubit set in the accept state (this property is also referred to as completeness).

In the FK protocol, in order to adjust the parameter ε to any arbitrary value between 0 and 1 (a technique called probability amplification), one needs to add polynomially many trap qubits within the MBQC pattern. Specifically, adding polynomially many traps and incorporating the pattern into a fault tolerance scheme that corrects d errors, gives parameter ε exponentially small on d. Adding polynomially many traps, following the same scheme as the FK protocol, creates a pattern that is not a DQC1-MBQC pattern. This is because the trap positions need to be random and there is always a chance that all the traps gather in the same layer of the MBQC computation (traps do not participate in teleportation steps therefore do not preserve purity as we will see later). Therefore to achieve an amplification of the error probability we need to develop a modified trapping scheme.

In Section 8.2 we give a verification protocol for DQC1-MBQC problems where, instead of running the pattern once, s computations of the same size are run in series, one being the actual computation and the others being trap computations. A similar

approach is also considered for the restricted setting of the photonic implementation of the FK protocol [Barz et al., 2013] and a verification protocol of the entanglement states [Pappa et al., 2012]. In our setting each trap computation contains an isolated trap injected in a random position between the qubits of the pattern. We prove that in this verification protocol the server is within DQC1-MBQC complexity class, while the client is within BPP together with single qubit preparations (as in the original FK protocol). Moreover in this verification protocol we achieve the goal of probability amplification by choosing the appropriate value for parameter s.

Theorem 12. *There exists a correct* ε*-verifiable protocol where the client is restricted to BPP and the ability to prepare single qubits and the server is within DQC1-MBQC. Using* $O(sm)$ *qubits and* $O(sm)$ *time steps, where m is the size of the input computation, we have:*

$$\varepsilon = \frac{2m}{s} \tag{7.9}$$

Therefore, in order to have constant failure probability ε one needs to run $s = O(m)$ trap+actual computations, meaning that the overall communication complexity will be quadratic on the input size. This is worse than our linear complexity protocol of Chapter 6, but it comes as a result of not being able to keep a polynomial number of pure qubits at the same time in DQC1-MBQC.

Chapter 8

One-Pure-Qubit Model Verification

8.1 Secure Computation with Restricted Purity

In this section we give a constructive proof of our main theorem for DQC1-MBQC and show how to construct a blind protocol as a consequence. The first step for proving Theorem 10 is the following rewriting scheme for patterns with flow.

Lemma 9. *Any measurement pattern on an open graph state (G,I,O) with flow $(f,\preceq)$ (as defined in Definition 2) and measurement angles a where either $|I| = |O|$ or the flow function is surjective can be rewritten as*

$$P_a = \prod_{i\in O} X_i^{S_i^x} Z_i^{S_i^z} \overrightarrow{\prod_{i\in O^c}} \left({}^{S_i^z}[M_i^{a_i}]^{S_i^x} \left(\prod_{\{k:k\sim i, k\succeq i\}} E_{i,k} \right) N_{f(i)}(|+\rangle) \right) \tag{8.1}$$

where $S_i^x = s_{f^{-1}(i)}$ for $i \in I^c$, else $S_i^x = 0$ and $S_i^z = \sum_{\{k:k\in I^c, k\sim i, i\neq f^{-1}(k)\}} s_{f^{-1}(k)} \mod 2$. The above pattern is runnable and implements the following unitary

$$U_{G,I,O,a} = 2^{|O^c|/2} \left(\prod_{i\in O^c} \langle +_{a_i}|_i \right) E_G N_{I^c} \tag{8.2}$$

where E_G and N_{I^c} represent the global entangling operator and global preparation respectively.

Proof. First we need to prove that P_a is runnable (cf. Definition 1). For condition (R0) we make the following observations: At step i, for $i \in I^c$, we need the result $s_{f^{-1}(i)}$ which is generated at step $f^{-1}(i)$, where $f^{-1}(i) \prec i$ from flow condition (F1). We also need the results $s_{f^{-1}(k)}$, for $\{k : k \in I^c, k \sim i, i \neq f^{-1}(k)\}$, which are generated at step $f^{-1}(k)$, where $f^{-1}(k) \prec i$ from flow condition (F2). Thus, condition (R0) is satisfied (see Figure 8.1 for a particular example). For condition (R1) we make the following

observations: At step i, for $i \in O^c$, the entangling operator and measurement operator act on qubit i which either belongs in the set of inputs I or is created at step $f^{-1}(i)$, where $f^{-1}(i) \prec i$ from flow condition (F1). Entangling operator acts also on qubits $\{k : k \sim i, k \succeq i\}$. If $k = f(i)$ then qubit k is created at the same step (i) by operator $N_{f(i)}$. If $k \neq f(i)$ then qubit k is either an input or it is created at step $f^{-1}(k)$, and we have by flow condition (F2): i is a neighbour of k and $i \neq f^{-1}(k)$, thus $f^{-1}(k) \prec i$ (Figure 8.1). Final correction operators act on qubits that belong to the set of outputs O, which either belong also to the set of inputs I or are created at steps $\{f^{-1}(i) : i \in O\}$, where $\forall i \in O \setminus I, f^{-1}(i) \prec i$ from flow condition (F1). In addition, they have not yet been measured since $i \notin O^C$. Thus, condition (R1) is satisfied. It is easy to see that condition (R2) is satisfied.

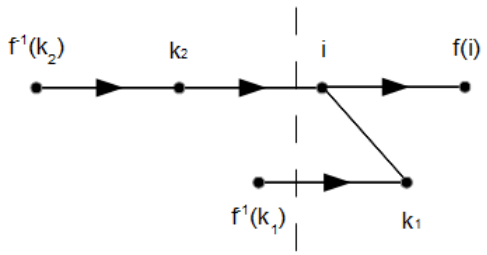

Figure 8.1: Qubit i gets an X correction from k_2 and Z corrections from $f^{-1}(k_2)$ and $f^{-1}(k_1)$. Qubits on the left of the dashed line are in the past of i. Qubit k_1 is created at timestep $f^{-1}(k_1)$ which is before timestep i from flow condition (F2).

Next we prove that the pattern of Equation 8.1 is implementing the unitary operation of Equation 8.2 when applied on an open graph with the properties described above. Since condition (R1) is satisfied, all preparation operators trivially commute with all previous operators

$$P_a = \prod_{i \in O} X_i^{S_i^x} Z_i^{S_i^z} \overrightarrow{\prod_{i \in O^c}} \left({}^{S_i^z}[M_i^{a_i}]^{S_i^x} \left(\prod_{\{k : k \sim i, k \succeq i\}} E_{i,k} \right) \right) N_{I^c}$$

Each entangling operator commutes with all previous measurements since it is applied on qubits with indices $\succeq i$.

$$P_a = \prod_{i \in O} X_i^{S_i^x} Z_i^{S_i^z} \overrightarrow{\prod_{i \in O^c}} \left({}^{S_i^z}[M_i^{a_i}]^{S_i^x} \right) E_G N_{I^c}$$

We can decompose the conditional measurements into simple measurements and corrections

$$P_a = \prod_{i \in O} X_i^{S_i^x} Z_i^{S_i^z} \overrightarrow{\prod_{i \in O^c}} \left(M_i^{a_i} X_i^{S_i^x} Z_i^{S_i^z} \right) E_G N_{I^c}$$

By rearranging the order of correction operators we take

$$P_a = \prod_{i\in O^c}^{\preceq} \left(X_{f(i)}^{s_i} \prod_{\{k:k\sim f(i),k\neq i\}} Z_k^{s_i} M_i^{a_i} \right) E_G N_{I^c}$$

The above equation implements the unitary operation presented in the lemma (Equation 8.2) as proved in [Danos and Kashefi, 2006]. □

Next, we notice that there exist many universal families of open graph states satisfying the requirements of the above lemma. One such example is the brickwork graph state originally defined in [Broadbent et al., 2009] and already presented here in Section 1.1.3. We remind the reader of its properties: In this graph state (Figure 1.3), the subset of vertices of the first column correspond to the input qubits I and the subset of vertices of the final column correspond to the output qubits O. This graph state has flow function $f((i,j)) = (i,j+1)$ and the following partial order for measuring the qubits: $\{(1,1),(2,1),\ldots,(w,1)\} \prec \{(1,2),(2,2),\ldots,(w,2)\} \prec \ldots \prec \{(1,d-1),(2,d-1),\ldots,(w,d-1)\}$, where w is the width and d is the depth of the graph and hence from Lemma 9 we obtain the following corollary.

Corollary 3. *Any computation over the brickwork open graph state G with qubit index* $(i \leq w, j \leq d)$ *can be rewritten as follows.*

$$P_a = \prod_{i=1}^{w} X_{(i,d)}^{S^x_{(i,d)}} Z_{(i,d)}^{S^z_{(i,d)}} \prod_{j=1}^{d-1} \prod_{i=1}^{w} S^z_{(i,j)} \left[M_{(i,j)}^{a_{(i,j)}} \right]^{S^x_{(i,j)}} \left(\prod_{\substack{\{k,l:(k,l)\sim(i,j),\\ k\geq i, l\geq j\}}} E_{(i,j),(k,l)} \right) N_{(i,j+1)} \quad (8.3)$$

where $S^x_{(i,j)} = s_{(i,j-1)}$ *for* $j > 1$, *else* $S^x_{(i,1)} = 0$
and $S^z_{(i,j)} = \sum_{\{k,l:(k,l)\sim(i,j),l\leq j\}} s_{(k,l-1)} \mod 2$ *for* $j > 2$, *else* $S^z_{(i,j)} = 0$.

We show that patterns defined in Lemma 9 are within the framework of Definition 13 hence obtaining a sufficient condition for DQC1-MBQC.

Lemma 10. *Any measurement pattern that can be rewritten in the form of Equation 8.1 represents a DQC1-MBQC pattern.*

Proof. A first general observation about the purity parameter π is that adding a new pure qubit σ to state ρ means that π increases by unity

$$\pi_{\rho\otimes\sigma} = \log_2 \mathrm{Tr}((\rho\otimes\sigma)^2) + d + 1 = \log_2 \mathrm{Tr}(\rho^2)\mathrm{Tr}(\sigma^2) + d + 1 = \pi_\rho + 1.$$

Additionally, applying any unitary U does not change the purity parameter π of the system since $\mathrm{Tr}((U\rho U^\dagger)^2) = \mathrm{Tr}(\rho^2)$ and dimension remains the same.

Returning to Equation 8.1, we notice that for every step $i \in O^c$ of the product the total computation performed corresponds mathematically to the following: On the qubit tagged with position i, a $J(a'_i)$ unitary gate is applied (where a'_i is an angle that depends on a_i and previous measurement results) up to a specific Pauli correction (depending on the known measurement result) and some specific Pauli corrections on its entangled neighbours (again depending on the measurement result). At the end the qubit is tagged with position $f(i)$ (where f is the flow function). Since this mathematically equivalent computation is a unitary and the dimension of the system remains the same (there is only a change of position tags) we conclude that each step $i \in O^c$ does not increase the purity parameter of the system. To finish the proof, we need to ensure that the individual operations within each step $i \in O^c$ and for $i \in O$ do not increase the purity parameter by more than a constant (and since there is only a constant number of operations within each step this does not increase the purity at any point more than constant). This is true since all these operations apply on (or add or trace over) a constant number of qubits.* □

From the above Theorem 10 follows directly.

8.1.1 Blind One-Pure-Qubit Computation

Building on this result, we can translate the UBQC protocol of [Broadbent et al., 2009] (and in fact many other existing protocols) to allow the blind execution of any DQC1-MBQC computation, where the server is restricted to DQC1-MBQC complexity class. The UBQC protocol is based on the brickwork graph state described above. Alice prepares all the qubits of the graph state, adding a random rotation around the (X,Y) plane to each one of them: $|+_{\theta_i}\rangle$, where θ_i is chosen at random from the set $A = \{0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/5, 3\pi/2, 7\pi/4\}$ and sends them to Bob, who entangles them according to the graph. The protocol then follows the partial order given by the flow: Alice calculates the corrected measurement angle α'_i for each qubit using previous measurement results according to the flow dependences. She sends to Bob measurement angle $\delta_i = \alpha'_i + \theta_i + r_i\pi$, using an extra random bit r_i. Bob measures according to δ_i, reports the result back to Alice who corrects it by XOR-ing with r_i. In the case of

*Assuming that the graph input has purity one, the purity parameter of the state of the system at any point reaches a maximum of two, so DQC2-MBQC would have been a more accurate name for our class. In terms of complexity this is not relevant since by [Shepherd, 2006] increasing the pure qubits to logarithmic does not give more computational power. Two qubits seem necessary because MBQC is based on teleportation using an auxiliary state.

quantum output, the final layer is sent to Alice and is also corrected according to the flow dependences by applying the corresponding Pauli operators.

Since the brickwork graph state satisfies the requirements of Theorem 10 we can adapt the Universal Blind Quantum Computing protocol by making Alice and Bob follow the order of Equation 8.3 and operate on input $|+\rangle\langle+|\otimes I_{w-1}/2^{w-1}$. A detailed description is given in Protocol 8.

Theorem 13. *Protocol 8 is correct.*

Proof. Correctness comes from the fact that what Alice and Bob jointly compute is mathematically equivalent to performing the pattern of Equation 8.3 on input $|+\rangle\langle+|\otimes I_{w-1}/2^{w-1}$. The argument is the same as in the original universal blind quantum computing protocol [Broadbent et al., 2009] repeated here for completeness. Firstly, since entangling operators commute with R_z operators, preparing the pure qubits in a rotated state does not change the underlying graph state; only the phase of each qubit is locally changed, and it is as if Bob had performed the R_z rotation after the entanglement. Secondly, since a measurement in the $|+_a\rangle, |-_a\rangle$ basis on a state $|\phi\rangle$ is the same as a measurement in the $|+_{a+\theta}\rangle, |-_{a+\theta}\rangle$ basis on $R_z(\theta)|\phi\rangle$, and since $\delta = a' + \theta + \pi r$, if $r = 0$, Bob's measurement has the same effect as Alice's target measurement; if $r = 1$, all Alice needs to do is flip the outcome.

□

Note that Protocol 8 can be trivially simplified by omitting all the measurements that are applied on maximally mixed states (i.e. all measurements applied on qubits in rows 2 to w from the beginning of the computation until each one is entangled with a non-maximally mixed qubit). However, this does not give any substantial improvement in the complexity of the protocol.

Theorem 14. *Protocol 8 is blind.*

(Proof Sketch). A detailed proof is provided in the next section. Intuitively, rotation by angle $\theta_{i,j}$ serves the purpose of hiding the actual measurement angle, while rotation by $r_{i,j}\pi$ hides the result of measuring the quantum state. This proof is consistent with definition of blindness based on the relation of Bob's system to Alice's system which takes into account prior knowledge of the secret and is a good indicator that blindness can be composable [Dunjko et al., 2014].

□

Protocol 8 Blind BQ1P protocol

Alice's input:

- A vector of angles $a = (a_{1,1}, \ldots, a_{w,d})$, where $a_{i,j}$ comes from the set $A = \{0, \pi/4, 2\pi/4, \ldots, 7\pi/4\}$, that when plugged in the measurement pattern P_a of Equation 8.3 applied on the brickwork state, implements the desired computation. This computation is applied on a fixed input state $|+\rangle\langle+| \otimes I_{w-1}/2^{w-1}$.

Alice's output:

- The top output qubit (qubit in position $(1,d)$).

The protocol

1. Alice picks a random angle $\theta_{1,1} \in A$, prepares one pure qubit in state $R_z(\theta_{1,1})|+\rangle$ and sends it to Bob who tags it as qubit $(1,1)$.

2. Bob prepares the rest of input state (qubits $(2,1), \ldots, (w,1)$) in the maximally mixed state $I_{w-1}/2^{w-1}$.

3. Alice and Bob execute the rest of the computation in rounds. For $j = 1$ to $d-1$ and for $i = 1$ to w

   (a) **Alice's preparation**

      i. Alice picks a random angle $\theta_{i,j+1} \in A$.

      ii. Alice prepares one pure qubit in state $R_z(\theta_{i,j+1})|+\rangle$.

      iii. Alice sends it to Bob. Bob tags it as qubit $(i, j+1)$.

   (b) **Entanglement and measurement**

      i. Bob performs the entangling operator(s):

$$\prod_{\{k,l:(k,l)\sim(i,j),k\geq i,l\geq j\}} E_{(i,j),(k,l)}$$

      ii. Bob performs the rest of the computation using classical help from Alice:

         A. Alice computes the corrected measurement angle $a'_{i,j} = (-1)^{S^x_{i,j}} a_{i,j} + S^z_{i,j}\pi$.

         B. Alice chooses a random bit $r_{i,j}$ and computes $\delta_{i,j} = a'_{i,j} + \theta_{i,j} + r_{i,j}\pi$.

         C. Alice transmits $\delta_{i,j}$ to Bob.

Protocol 8 Cont'd

3. (b) ii. D. Bob performs operation $M_{i,j}^{\delta_{i,j}}$ which measures and traces over the qubit (i,j) and retrieves result $b_{i,j}$.

   E. Bob transmits $b_{i,j}$ to Alice.

   F. Alice updates the result to $s_{i,j} = b_{i,j} + r_{i,j} \mod 2$.

4. Bob sends to Alice the final layer of qubits, Alice performs the required corrections and outputs the result.

Regarding the complexity of the protocol, Alice needs to pick a polynomially large number of random bits and perform polynomially large number of modulo additions that is to say Alice classical computation is restricted to the class *BPP*. Alice's quantum requirement is only to prepare single qubits, she has access to no quantum memory or quantum operation. Therefore assuming $BQ1P \not\subset BPP$ suggests Alice's quantum power is more restricted than $BQ1P$ and hence DQC1-MBQC. On the other hand, Bob performs a pattern of the form given in Equation 8.3, with the difference that instead of preparing the pure qubits himself, he receives the pure qubits through the quantum channel that connects him with Alice. Also, the qubits are not prepared in state $|+\rangle$, but in some state on the (X,Y) plane, but this doesn't alter the reasoning in the complexity proofs. Thus, Bob has computational power that is within the DQC1-MBQC complexity class according to the Corollary 3 and Theorem 10.

8.1.2 Blindness Proof

In this proof of blindness for Protocol 8 we use techniques developed in [Dunjko, 2012]. The basic difference from the proof of [Dunjko, 2012] arises from the different order in which Bob receives the states from Alice. Nevertheless, after commuting all CPTP maps into a single operator at the end, the methodology for proving blindness is the same as in the original proof. We give the full proof here for the sake of clarity.

To prove the property of blindness, we do not separate Alice's system into a classical and a quantum part but we consider the whole of Alice's system as quantum. This is a reasonable assumption since a classical system can be viewed as a special case of a quantum system. Therefore, by proving blindness for the more general case we also prove blindness for the special case.

For the sake of clarity, we use single indexing for all the qubits of the resource state. The total number of qubits is denoted by m and the number of qubits in a single column of the brickwork state is denoted by n.

Our goal will be to explicitly write the state $\sigma_B = \mathrm{Tr}_A(\sigma_{AB})$ that Bob holds at the end of the execution of the protocol. To achieve this we express Bob's behaviour at each step i of the protocol as a collection of completely-positive trace-preserving (CPTP) maps $\mathcal{E}_i^{b_i}$, each for every possible classical response b_i from Bob to Alice.

At step 1 of the main loop of the protocol Bob has already been given the top input qubit at position 1 (position $(1,1)$ in the protocol notation) and the qubit at position $f(1) = 1+n$ (position $(1,2)$ in the protocol notation) together with the angle for measuring qubit 1 (angle can be represented as a quantum state composed of 3 qubits). State $\mathrm{Tr}_A(\rho_{AB})$ represents Bob's state before the protocol begins and can, in general, be dependent on Alice's secret measurement angles. The state of Bob averaged over all possible choices of Alice and possible classical responses from Bob, after step 1 is:

$$\sum_{b_1,r_1,\theta_1,\theta_{1+n}} \mathcal{E}_1^{b_1}\left(|\delta_1^{\theta_1,r_1}\rangle\langle\delta_1^{\theta_1,r_1}| \otimes |+_{\theta_{1+n}}\rangle\langle+_{\theta_{1+n}}| \otimes |+_{\theta_1}\rangle\langle+_{\theta_1}| \otimes \mathrm{Tr}_A(\rho_{AB})\right)$$

Note the all binary parameters in sums range over 0 and 1, ex. $\sum_{b_1}$ stands for $\sum_{b_1=0}^{1}$ and all angles range over the 8 possible values in A.

We can write the state of Bob after step 2 of the main iteration as:

$$\sum_{b_2,b_1,r_2,r_1,\theta_{2+n},\theta_{1+n},\theta_2,\theta_1} \mathcal{E}_2^{b_2}\Big(|\delta_2^{\theta_2,r_2}\rangle\langle\delta_2^{\theta_2,r_2}| \otimes |+_{\theta_{2+n}}\rangle\langle+_{\theta_{2+n}}| \otimes \mathcal{E}_1^{b_1}\left(|\delta_1^{\theta_1,r_1}\rangle\langle\delta_1^{\theta_1,r_1}| \otimes |+_{\theta_{1+n}}\rangle\langle+_{\theta_{1+n}}| \otimes |+_{\theta_1}\rangle\langle+_{\theta_1}| \otimes \mathrm{Tr}_A(\rho_{AB})\right)\Big)$$

Following this analysis, after the last step of the iteration Bob's state will be:

$$\sigma_B = \sum_{\substack{b_{\le m-n},\\ r_{\le m-n},\theta_{\le m}}} \mathcal{E}_{m-n}^{b_{m-n}}\Big(|\delta_{m-n}^{b_{<m-n},r_{\le m-n},\theta_{m-n}}\rangle\langle\delta_{m-n}^{b_{<m-n},r_{\le m-n},\theta_{m-n}}| \otimes |+_{\theta_m}\rangle\langle+_{\theta_m}| \otimes \ldots \otimes \mathcal{E}_2^{b_2}\Big(|\delta_2^{\theta_2,r_2}\rangle\langle\delta_2^{\theta_2,r_2}| \otimes |+_{\theta_{2+n}}\rangle\langle+_{\theta_{2+n}}| \otimes \mathcal{E}_1^{b_1}\left(|\delta_1^{\theta_1,r_1}\rangle\langle\delta_1^{\theta_1,r_1}| \otimes |+_{\theta_{1+n}}\rangle\langle+_{\theta_{1+n}}| \otimes |+_{\theta_1}\rangle\langle+_{\theta_1}| \otimes \mathrm{Tr}_A(\rho_{AB})\right)\Big)\ldots\Big)$$

Notation $b_{<m-n}$ stands for all the elements of b with index less than $m-n$.

Collecting all CPTP maps by commuting them with systems which they do not apply on into a single operator $\mathcal{E}$ and rearranging the terms of the tensor product inside gives:

$$= \sum_{\substack{b_{\leq m-n},\\ r_{\leq m-n},\theta_{\leq m}}} \mathcal{E}^{b_{\leq m-n}}\Big(\bigotimes_{i=m-n}^{m} |+_{\theta_i}\rangle\langle+_{\theta_i}| \bigotimes_{i=n+1}^{m-n-1} (|\delta_i^{b_{<i},r_{\leq i},\theta_i}\rangle\langle\delta_i^{b_{<i},r_{\leq i},\theta_i}| \otimes |+_{\theta_i}\rangle\langle+_{\theta_i}|) \bigotimes_{i=2}^{n} (|\delta_i^{\theta_i,r_i}\rangle\langle\delta_i^{\theta_i,r_i}|) \otimes |\delta_1^{\theta_1,r_1}\rangle\langle\delta_1^{\theta_1,r_1}| \otimes |+_{\theta_1}\rangle\langle+_{\theta_1}| \otimes \mathrm{Tr}_A(\rho_{AB})\Big)$$

We introduce the controlled unitary:

$$U = \prod_{n+1\leq i\leq m-n-1, i=1} Z_i(-\delta_i)$$

and rewrite the state as:

$$\sum_{\substack{b_{\leq m-n},\\ r_{\leq m-n},\theta_{\leq m}}} \mathcal{E}^{b_{\leq m-n}}\Big(U^\dagger U \bigotimes_{i=m-n}^{m} |+_{\theta_i}\rangle\langle+_{\theta_i}| \bigotimes_{i=n+1}^{m-n-1} (|\delta_i^{b_{<i},r_{\leq i},\theta_i}\rangle\langle\delta_i^{b_{<i},r_{\leq i},\theta_i}| \otimes |+_{\theta_i}\rangle\langle+_{\theta_i}|) \bigotimes_{i=2}^{n} (|\delta_i^{\theta_i,r_i}\rangle\langle\delta_i^{\theta_i,r_i}|) \otimes |\delta_1^{\theta_1,r_1}\rangle\langle\delta_1^{\theta_1,r_1}| \otimes |+_{\theta_1}\rangle\langle+_{\theta_1}| U^\dagger U \otimes \mathrm{Tr}_A(\rho_{AB})\Big)$$

After applying the innermost unitary and absorbing the outermost into the CPTP-map we have:

$$\sum_{\substack{b_{\leq m-n},\\ r_{\leq m-n},\theta_{\leq m}}} \mathcal{E}'^{b_{\leq m-n}}\Big(\bigotimes_{i=m-n}^{m} |+_{\theta_i}\rangle\langle+_{\theta_i}| \bigotimes_{i=n+1}^{m-n-1} \Big(|\delta_i^{b_{<i},r_{\leq i},\theta_i}\rangle\langle\delta_i^{b_{<i},r_{\leq i},\theta_i}| \otimes |+_{-a_i'^{\,b_{<i},r_{<i}}-r_i\pi}\rangle\langle+_{-a_i'^{\,b_{<i},r_{<i}}-r_i\pi}| \Big) \bigotimes_{i=2}^{n} \Big(|\delta_i^{\theta_i,r_i}\rangle\langle\delta_i^{\theta_i,r_i}| \Big) \otimes |\delta_1^{\theta_1,r_1}\rangle\langle\delta_1^{\theta_1,r_1}| \otimes |+_{-a_1'-r_1\pi}\rangle\langle+_{-a_1'-r_1\pi}| \otimes \mathrm{Tr}_A(\rho_{AB})\Big)$$

It is essential for the proof that each term with index i in the tensor products depends only on parameters with index $\leq i$. This allows to break the summations over $r_{\leq m-n}$ and $\theta_{\leq m}$ and calculate them iteratively from left to right, given the following:

$$\sum_{\theta_i} |+_{\theta_i}\rangle\langle+_{\theta_i}| = \frac{I_1}{2}$$

where $I_n = \bigotimes_n I$. Also,

$$\sum_{r_i,\theta_i} |\delta_i^{r_{\leq i},\theta_i}\rangle\langle\delta_i^{r_{\leq i},\theta_i}| \otimes |+_{-a_i'^{r_{<i}}-r_i\pi}\rangle\langle+_{-a_i'^{r_{<i}}-r_i\pi}|$$
$$= \sum_{r_i}\Big(\sum_{\theta_i}\Big(|a_i'^{r_{<i}}+\theta_i+r_i\pi\rangle\langle a_i'^{r_{<i}}+\theta_i+r_i\pi|\Big) \otimes |+_{-a_i'^{r_{<i}}-r_i\pi}\rangle\langle+_{-a_i'^{r_{<i}}-r_i\pi}|\Big)$$
$$= \sum_{r_i}\frac{I_3}{2^3} \otimes |+_{-a_i'^{r_{<i}}-r_i\pi}\rangle\langle+_{-a_i'^{r_{<i}}-r_i\pi}|$$
$$= \frac{I_4}{2^4}$$

and

$$\sum_{r_i,\theta_i} |\delta_i^{\theta_i,r_i}\rangle\langle\delta_i^{\theta_i,r_i}| = \frac{I_3}{2^3}$$

This procedure will produce the state:

$$\sigma_B = \mathcal{E}'\left(\frac{I_{4m-4n+1}}{2^{4m-4n+1}} \otimes \mathrm{Tr}_A(\rho_{AB})\right) = \mathcal{E}''(\mathrm{Tr}_A(\rho_{AB}))$$

where $\mathcal{E}''$ is some CPTP map. Therefore Definition 6 is satisfied.

8.2 Verification of One-Pure-Qubit

We remind the basic properties of the FK protocol. The FK protocol is based on the ability to hide a trap qubit inside the graph state while not affecting the correct execution of the pattern. Both the trap qubit and the qubits which participate in the actual computation are prepared in the (X,Y) plane of the Bloch sphere. To keep them disentangled, some qubits (called dummy) prepared in the computational basis $\{|0\rangle,|1\rangle\}$, are injected between them. Being able to choose between the two states is essential for blindness (Theorem 4 in [Fitzsimons and Kashefi, 2012]). In particular, if a dummy qubit is in state $|0\rangle$, applying the entangling operator cZ between this qubit and a qubit prepared on the (X,Y) plane has no effect. If a dummy qubit is in state $|1\rangle$ then applying cZ will introduce a Pauli Z rotation on the qubit prepared on the (X,Y) plane. This effect can be cancelled by Alice in advance, by introducing a Pauli Z rotation on all the neighbours of $|1\rangle$'s when preparing the initial state.

In the simplest version of the FK protocol, a single trap, prepared in state $|+_{\theta_t}\rangle$, where θ is chosen at random from the angles set A (defined above) and placed at

position t, chosen at random between all the vertices of the open graph state (G,I,O). During the execution of the pattern, if $t \notin O$, Bob is asked to measure qubit t with angle $\theta_t + r\pi$ and return the classical result b_t to Alice. If $b_t = r_t$ Alice sets an indicator bit to state *acc* (which means that this computation is accepted), otherwise she sets it to *rej* (computation is rejected). If $t \in O$, Alice herself measures the trap qubit and sets the indicator qubit accordingly. This version of the protocol is proven to be correct and ε-verifiable, with $\varepsilon = (m-1)/m$, where m is the size of the computation.

A generalisation of this technique which allows for arbitrary selection of parameter ε is also presented in [Fitzsimons and Kashefi, 2012]. By allowing for a polynomial number of traps to be injected in the graph state and adapting the computation inside a fault tolerant scheme with parameter d one can have ε inversely exponential to d. The question is whether this amplification method can also be used to design a verification protocol for DQC1-MBQC with arbitrary small ε. Unfortunately the underlying graph state used by this protocol does not have flow and not all qubits are prepared in the (X,Y) plane, so that one cannot apply Theorem 10 to get a compatible rewriting of the pattern. Moreover, having the requirement that we should be able to place every trap qubit (which is a pure qubit) at any position in the graph, means that there exist patterns that will never be possible to be rewritten to satisfy the purity requirement. This leads us to seek a different approach for probability amplification for verification in the DQC1-MBQC model.

Instead of placing a polynomial number of isolated traps within the same graph, which is also used to perform the actual computation, we utilise s isolated brickwork subgraphs, one used for the computation and the rest being trap subgraphs (see Figure 8.2). Thus, at the beginning of the protocol, Alice chooses random parameter t_g, which denotes which graph will be the computational subgraph, and for each of the remaining trap subgraphs i, she chooses a random position t_i to hide one isolated trap. The rest of each trap subgraph will be a trivial computation (all measurement angles set to 0) on a totally mixed state, and a selected set of dummy qubits are placed to isolate this computation from the trap. Computation subgraph and trap subgraphs are of the same size, and by taking advantage of the blindness of the protocol, Bob cannot distinguish between them. Therefore, to be able to cheat, he needs to deviate from the correct operation only during the execution on the computational subgraph and never deviate while operating on any of the traps. This gives the desirable ε parameter that will be proved later. The full description of protocol is given in Protocol 9. Each isolated pattern k is executed separately and according to the DCQ1-MBQC rewriting on the brickwork

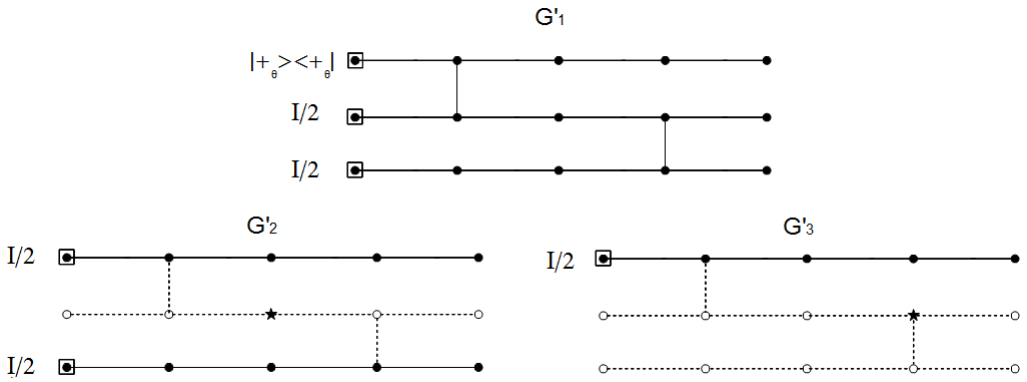

Figure 8.2: Let G' be the graph which consists of s isolated brickwork graphs (each denoted as G'_i), each of the same dimensions required for the desired computation. An example construction with $s = 3$ and one trap per graph together with a small brickwork state for computation is given above. Black vertices correspond to auxiliary qubits prepared on the $(X-Y)$ plane or mixed state when they are inputs (inside square), star vertices correspond to trap qubits and white vertices to auxiliary qubits prepared in the computational basis. Edges represent entangling operators, dashed where entangling has no effect (except of local rotations).

state given in Equation 8.3 in the blind setting. Pre-rotations on the neighbours of dummy qubits guarantee that the computation is not affected by the choice of dummies as described before.

Protocol 9 Verifiable DQC1-MBQC protocol with $s-1$ trap computations

---

Alice's input:

- An angle vector $a = (a_{1,1}, \ldots, a_{w,d-1})$, where $a_{i,j}$ comes from the set $A = \{0, \pi/4, 2\pi/4, \ldots, 7\pi/4\}$, that, when plugged in the measurement pattern P_a of Equation 8.3 on the brickwork open graph state G of dimension (w,d) and flow $(f, \preceq)$, it implements the desired computation on fixed input $|+\rangle\langle+| \otimes I_{w-1}/2^{w-1}$.

Alice's output:

- The top output qubit of G (qubit in position $(1,d)$ in G) together with a 1-bit, named *acc*, that indicates if the result is accepted or not.

Protocol 9 (cont'd)

The protocol

- **Preparation steps.** Alice picks t_g at random from $\{1,\ldots,s\}$. Let G' be the graph which consists of s isolated brickwork graphs, each of the dimension the same as G. Then the t_g-th isolated graph (named G'_{t_g}) will be the computational subgraph for this run of the protocol.

- Alice maps the measurement angles of the computational subgraph G'_{t_g} to angles of graph G: $a'_{G_{t_g}\setminus O_{t_g}} = a$ and appropriately set the dependency sets S^x and S^z for all the vertices of G'_{t_g} (according to the standard flow), while for the rest of the vertices (graph $G' \setminus G'_{t_g}$) the sets S^x and S^z are empty.

- For $k = 1$ to s except t_g:

  1. Alice chooses one random vertex $t_k = (t_x, t_y)_k$ among all vertices of G'_k for placing the trap.
  2. By G'_k's geometry, vertex (t_x, t_y) may be connected by a vertical edge to vertex (t'_x, t_y), where t'_x represents either $t_x + 1$ or $t_x - 1$. We add in D (set of dummies) all vertices of rows t_x, t'_x (if it exists) of G'_k, except the trap itself.
  3. All elements of a'_{G_k} are mapped to 0.

- Alice chooses random variables $\theta_{G'\setminus D}$, each uniformly at random from A.

- Alice chooses random variables $r_{G'}$ and d_D, each $\in_R \{0,1\}$.

- For $k = 1$ to s:

  1. **Initial step.** If $k = t_g$ then: Let $(1,1)_k$ be the position of the top input qubit in G'_k. Alice prepares the following states and sends them to Bob:

$$\begin{array}{rl} \{(1,1)_k\} & |+_{\theta_{(1,1)_k}}\rangle \\ \forall (i,1)_k \notin \{(1,1)_k\} & I/2 \end{array}$$

  Otherwise: Alice prepares the following states and sends them to Bob:

$$\begin{array}{rl} \forall (i,1)_k \in D & |d_{(i,1)_k}\rangle \\ (i,1)_k = t_k & \prod_{\{m,l:(m,l)_k \sim (i,1)_k, (m,l)_k \in D\}} Z^{d_{(m,l)_k}} |+_{\theta_{(i,1)_k}}\rangle \\ \forall (i,1)_k \notin \{D, t_k\} & I/2 \end{array}$$

Protocol 9 (cont'd)

2. **Main Iteration.** For $j = 1$ to $d-1$, for $i = 1$ to w:

   (a) **Alice's preparation**

      i. Alice prepares one pure qubit in one of the following states, depending on $(i, j+1)_k$:

$$\begin{array}{ll} (i,j+1)_k \in D & |d_{(i,j+1)_k}\rangle \\ (i,j+1)_k \notin D & \prod_{\{m,l:(m,l)_k \sim (i,j+1)_k, (m,l)_k \in D\}} Z^{d_{(m,l)_k}} |+_{\theta_{(i,j+1)_k}}\rangle \end{array}$$

      ii. Alice sends it to Bob. Bob labels it as qubit $(i, j+1)_k$.

   (b) **Entanglement and measurement**

      i. Bob performs the entangling operator(s):

$$\prod_{\{m,l:(m,l)_k \sim (i,j)_k, m \geq i, l \geq j\}} E_{(i,j)_k,(m,l)_k}$$

      ii. Bob performs the rest of the computation using classical help from Alice:

         A. Alice computes the corrected measurement angle $a''_{(i,j)_k} = (-1)^{S^x_{(i,j)_k}} a'_{(i,j)_k} + S^z_{(i,j)_k}\pi$.
         B. Alice computes actual measurement angle $\delta_{(i,j)_k} = a''_{(i,j)_k} + \theta_{(i,j)_k} + r_{(i,j)_k}\pi$.
         C. Alice transmits $\delta_{(i,j)_k}$ to Bob.
         D. Bob performs operation $M^{\delta_{(i,j)_k}}_{(i,j)_k}$ which measures and traces over the qubit $(i, j)_k$ and retrieves result $b_{(i,j)_k}$.
         E. Bob transmits $b_{(i,j)_k}$ to Alice.
         F. Alice updates the result to $s_{(i,j)_k} = b_{(i,j)_k} + r_{(i,j)_k} \mod 2$.

3. Bob sends the final layer to Alice and Alice applies the final corrections if needed (only in round t_g).

4. If the trap qubit is within the qubits received, Alice measures it with angle $\delta_{t_k} = \theta_{t_k} + r_{t_k}\pi$ to obtain b_{t_k}. Also, Alice discards all qubits received by Bob in this round except qubit $(1,d)_{t_g}$.

- Alice outputs qubit in position $(1,d)_{t_g}$ and sets bit *acc* to 1 if $b_{t_k} = r_{t_k}$ for all k.

To prove the complexity of the protocol we need to notice that although the graph used satisfies the conditions of Theorem 10, the existence of the dummy qubits prepared in the computational basis creates the need of a new proof.

Theorem 15. *The computational power of Bob in Protocol 9 is within DQC1-MBQC.*

Proof. Note that the s patterns are executed in series and Bob does not keep any qubits between executions. The inputs to these patterns are almost maximally mixed, in accordance with the purity requirement and this 'mixedness' propagates through both computational and trap subgraphs. For the computational subgraph (which is not entangled with the rest) the reasoning of the proof of Theorem 10 applies, since this subgraph satisfies the sufficient conditions and no dummy qubits are used. In the case of a trap subgraph k consider first those operations that apply on the isolated trap and dummy subgraph only. Then for each step $(i, j)_k$ of the main iteration of the protocol (where $(i, j)_k$ is a trap or a dummy) a new pure qubit is sent to Bob, which increases the purity parameter by 1. Entangling will not have any effect on the purity parameter. While the measurement does not increase the purity of the qubit since it was already pure (dummy or trap remain always pure through the computation), and tracing out the resulting qubit will decrease the purity by 1. Thus, the whole step will not change the purity. On the other hand, for the remaining operations the reasoning of the proof of Theorem 10 goes through, since this subgraph satisfies the sufficient conditions. Also operations that apply on both subgraphs are all unitary therefore they do not affect purity. □

Using the definition of verifiability given in Definition 14 we prove the main theorem for the existence of a correct and verifiable DQC1-MBQC protocol (Theorem 12). The full proof is given in the next section, while here we describe the main steps.

Proof of Theorem 12 (Sketch). Correctness of Protocol 9 comes from the fact that the computational subgraph is disentangled from the rest of the computation and if Bob performs the predefined operations, from the correctness of the blind protocol Alice will receive the correct output. Also, in this case, (and since the traps are corrected to cancel the effect of their entanglement with their neighbouring dummies) the measurement of the traps will give the expected result and Alice will accept the computation.

The proof of verifiability follows the same general methodology of the proof of the original FK protocol [Fitzsimons and Kashefi, 2012], except the last part which contains the counting arguments. For the rest we use single indexing for the qubits,

where subgraph G'_i consists of m qubits indexed $(i-1)+1$ to im. Consequently, the total number of qubits in the protocol is sm. Parameter n represents the size of the input of each subgraph (parameter w in the protocol).

Based on Definition 14 we need to bound the probability of the (purified) output collapsing onto the wrong subspace and accepting that result. To explicitly write the final state $B_j(\nu)$ we need to define the following notations. Alice's chosen random parameters are denoted collectively by ν, a subset of those are related to the traps: ν_T including t_g, t_k's and θ_{t_k}'s for $k \in \{1,\ldots,s\} \setminus t_g$. Also $\nu_C = \{\nu \setminus \nu_T\}$. The projection onto the correct state for each trap t_k is denoted by $|\eta_{t_k}^{\nu_T}\rangle$, where $|\eta_{t_k}^{\nu_T}\rangle = |+_{\theta_{t_k}}\rangle$ when $t_k \in O_k$ and $|\eta_{t_k}^{\nu_T}\rangle = |r_{t_k}\rangle$ otherwise (since the trap has been already measured). C_r denotes the Pauli operators that map the output state of the computational subgraph to the correct one. c_r is used to compactly deal with the fact that in the protocol each measured qubit i is decrypted by XOR-ing them with r_i, except for the trap qubits which remain uncorrected: $\forall k : (c_r)_{t_k} = 0$. $\rho_{M_k^\nu}$ denotes the density matrix representing the total quantum state received by Bob from Alice for each round k of the protocol. A special case is the t_kth round where $\rho_{M_k^\nu}$ represents the total state received by Bob together with its purification (not known to Bob). The classical information received by Bob at each elementary step i (measurement angles) are represented by $|\delta_i\rangle$'s.

We allow Bob to have an arbitrary deviation strategy j, at each elementary step i which is represented as CPTP map $\mathcal{E}_i^j$, followed by a Pauli Z measurement of qubit i (since Bob has to produce a classical bit at each step and return it to Alice), which is represented by taking the sum over projectors on the computational basis $|b_i\rangle$, for $b_i \in \{0,1\}$. All measurement operators can be commuted to the end of the computation and all CPTP maps can be gathered to a single map $\mathcal{E}^j$ after Bob has received everything from Alice, so that the failure probability can be written as:

$$p_{incorrect} = \sum_{b',\nu} p(\nu)\mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} |\eta_{t_k}^{\nu_T}\rangle\langle\eta_{t_k}^{\nu_T}|$$
$$C^{b',\nu_C}|b'+c^r\rangle\langle b'|\mathcal{E}^j\left(\bigotimes_{k=1}^{s}\bigotimes_{i=1}^{m-n}|\delta_{(k-1)m+i}^{b',\nu}\rangle\langle\delta_{(k-1)m+i}^{b',\nu}| \otimes \rho_{M_k^\nu}\right)|b'\rangle\langle b'+c^r|C^{b',\nu_C\dagger})$$

Our strategy will be to rewrite this probability by introducing the correct execution of the protocol before the attack, on each subgraph k: $\mathcal{P}_k = \bigotimes_{i=1}^{m-n}(H_{(k-1)m+i}Z_{(k-1)m+i}(\delta_{(k-1)m+i}))E_{G'_k}$ and at the same time decomposing the attack to the Pauli basis, using general Paulis

$\sigma_{i,k}$ applying on qubits $(k-1)m+1 \leq \gamma \leq km$ for each k.

$$p_{incorrect} = \sum_{b',\nu,\nu,i,j} \alpha_{\nu i}\alpha^*_{\nu j} p(\nu) \mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} |\eta^{\nu_T}_{t_k}\rangle\langle\eta^{\nu_T}_{t_k}| C^{b',\nu_C}|b'+c^r\rangle\langle b'|$$
$$\bigotimes_{k=1}^{s}(\sigma_{i,k}\left(\mathcal{P}_k \bigotimes_{i=1}^{m-n} |\delta^{b',\nu}_{(k-1)m+i}\rangle\langle\delta^{b',\nu}_{(k-1)m+i}| \otimes \rho_{M^\nu_k}\mathcal{P}^\dagger_k\right)\sigma_{j,k})|b'\rangle\langle b'+c^r|C^{b',\nu_C\dagger}$$

This way we can characterise which Pauli attacks give non-zero failure probability when the final state is projected on the correct one. For convenience we introduce the following sets for an arbitrary Pauli $\sigma_{i,k}$:

$$A_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = I \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$
$$B_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = X \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$
$$C_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Y \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$
$$D_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Z \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$

We use the superscript O to denote subsets subject to the constraint $km \geq \gamma \geq km-n+1$. For an arbitrary t_g, the only attacks that give the corresponding term of the sum not equal to zero: are those that (i) produce an incorrect measurement result for qubits $(t_g-1)m+1 \leq \gamma \leq t_g m-n$ or (ii) operate non-trivially on qubits $t_g m-n < \gamma \leq t_g m$. We denote this condition by $i \in E_{i,t_g}$ and $j \in E_{j,t_g}$: $|B_{i,t_g}|+|C_{i,t_g}|+|D^O_{i,t_g}| \geq 1$ and $|B_{j,t_g}|+|C_{j,t_g}|+|D^O_{j,t_g}| \geq 1$.

The next step will be to characterise which attacks of these subsets remain undetected by the trap mechanism and try to find an upper bound on their contribution to the failure probability. By applying blindness and observing that only the terms where $\sigma_{i,k} = \sigma_{j,k}$ contribute we obtain the following upper bound (details in the next section):

$$p_{incorrect} \leq \sum_{t_g} \sum_{\nu,i\in E_{i,t_g}} |\alpha_{\nu i}|^2 p(t_g) \prod_{k=\{1,\ldots,s\}\setminus t_g} \Big(\sum_{\substack{km-n<t_k\leq km,\\ \theta_{t_k}}} p(t_k,\theta_{t_k})(\langle +_{\theta_{t_k}}|\sigma_{i|t_k}|+_{\theta_{t_k}}\rangle)^2$$
$$+ \sum_{\substack{(k-1)m<t_k\leq km-n,\\ r_{t_k}}} p(t_k,r_{t_k})(\langle r_{t_k}|\sigma_{i|t_k}|r_{t_k}\rangle)^2\Big)$$

The rest is based on a counting argument using $\forall k$, $|A_{i,k}|+|B_{i,k}|+|C_{i,k}|+|D_{i,k}| = m$.

$$p_{incorrect} \leq \sum_{t_g} \sum_{v,i\in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m}(2|A_{i,k}| + |B^O_{i,k}| + |C^O_{i,k}| + 2|D_{i,k} \setminus D^O_{i,k}|)$$
$$\leq \sum_{t_g} \sum_{v,i\in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

We denote the product term $\prod_{k=\{1,2,3,\ldots,s\}\setminus z} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$ as $P_{i,z}$. We also denote each set $\{E^*_{i,1} \cap E^*_{i,2} \cap \ldots \cap E^*_{i,s}\}$, where each term $E^*_{i,w}$ is either $E_{i,w}$ or its complement, $E^C_{i,w}$, depending on whether the w-th value of a binary vector y (size s) is 1 or 0 respectively, as $W_{i,y}$. Let the function $\#y$ give the number of positions i such that y_i=1.

$$= \frac{1}{s}(\sum_{k=1}^{s} \sum_{\{y:\#y=k\}} \sum_{i\in W_{i,y},v} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} P_{i,z}))$$

The condition $i \in W_{i,y}$ means that the following conditions hold together: $\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| \geq 1 : y_w = 1\}$,$\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| = 0 : y_w = 0\}$.

$$\leq \frac{1}{s}(\sum_{k=1}^{s} \sum_{\{y:\#y=k\}} \sum_{i\in W_{i,y},v} |\alpha_{vi}|^2 k \left(\frac{2m-1}{2m}\right)^{k-1}) = \frac{1}{s}(\sum_{k=1}^{s} c_k k \left(\frac{2m-1}{2m}\right)^{k-1})$$

where $c_k = \sum_{\{y:\#y=k\}} \sum_{i\in W_{i,y},v} |\alpha_{vi}|^2$.

An upper bound on the above expression is:

$$p_{incorrect} < \frac{2m}{s} \tag{8.4}$$

□

8.3 Verifiability Proof

In this section we give the full proof of Theorem 12, the sketch of which proof was given in the previous section. The same notation is followed.

The first step is to write the state of Alice's system at the end of the execution of the protocol for fixed Bob's behaviour j and choices of Alice ν. We have utilised the fact that all measurements can be moved to the end. Also, we have commuted all Bob's operations to the end (before the measurements) merging them to a single CPTP map. The state of Alice is:

$$B_j(\nu) = \sum_b \otimes_{i=k}^{s} |+_{\theta_{t_k}+b_{t_k}\pi}\rangle\langle+_{\theta_{t_k}+b_{t_k}\pi}| C^{b,\nu_C} |b + c^r\rangle\langle b|$$

$$\mathcal{E}^j \left(\bigotimes_{k=1}^{s} \bigotimes_{i=1}^{m-n} |\delta^{b,\nu}_{(k-1)m+i}\rangle\langle\delta^{b,\nu}_{(k-1)m+i}| \otimes \rho_{M_k^\nu} \right) |b\rangle\langle b + c^r| C^{b,\nu_C\dagger} \otimes_{i=k}^{s} |+_{\theta_{t_k}+b_{t_k}\pi}\rangle\langle+_{\theta_{t_k}+b_{t_k}\pi}|$$

where $|+_{\theta_{t_k}+b_{t_k}\pi}\rangle\langle+_{\theta_{t_k}+b_{t_k}\pi}|$ are used to define Alice's measurement of the traps which are part of the output state of each round k (if they exist).

To bound the failure probability, observe that projectors orthogonal to $|\eta^{\nu_T}_{t_k}\rangle$'s vanish, thus we have (where $b' = \{b_i\}_{i \neq t_1 \ldots t_s}$):

$$p_{incorrect} = \sum_{b',\nu} p(\nu) \mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} |\eta^{\nu_T}_{t_k}\rangle\langle\eta^{\nu_T}_{t_k}|$$
$$C^{b',\nu_C} |b' + c^r\rangle\langle b'| \mathcal{E}^j \left(\bigotimes_{k=1}^{s} \bigotimes_{i=1}^{m-n} |\delta^{b',\nu}_{(k-1)m+i}\rangle\langle\delta^{b',\nu}_{(k-1)m+i}| \otimes \rho_{M_k^\nu} \right) |b'\rangle\langle b' + c^r| C^{b',\nu_C\dagger})$$

We introduce the following unitary, which characterises the correct operation on each subgraph k: $\mathcal{P}_k = \bigotimes_{i=1}^{m-n} (H_{(k-1)m+i} Z_{(k-1)m+i}(\delta_{(k-1)m+i})) E_{G'_k}$.

We can rewrite the failure probability, introducing $\mathcal{P}_k^\dagger \mathcal{P}_k$'s on both sides of the quantum state of the system before the attack, and absorbing the outermost unitaries into the updated CPTP map $\mathcal{E}'^j$:

$$p_{incorrect} = \sum_{b',\nu} p(\nu) \mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} |\eta^{\nu_T}_{t_k}\rangle\langle\eta^{\nu_T}_{t_k}| C^{b',\nu_C}$$
$$|b' + c^r\rangle\langle b'| \mathcal{E}'^j \left(\bigotimes_{k=1}^{s} (\mathcal{P}_k \bigotimes_{i=1}^{m-n} |\delta^{b',\nu}_{(k-1)m+i}\rangle\langle\delta^{b',\nu}_{(k-1)m+i}| \otimes \rho_{M_k^\nu} \mathcal{P}_k^\dagger) \right) |b'\rangle\langle b' + c^r| C^{b',\nu_C\dagger})$$

We decompose $\mathcal{E}'^j$ using the following facts: There exist some matrices $\{\chi_v\}$ of dimension $s(4m-3n) \times s(4m-3n)$, with $\sum_v \chi_v \chi_v^\dagger = I$ such that for every density operator ρ: $\mathcal{E}'^j(\rho) = \sum_v \chi_v \rho \chi_v^\dagger$. Also, each χ_v can be decomposed to the Pauli basis: $\chi_v = \sum_i \alpha_{vi} \sigma_i$, with $\sum_{v,i} \alpha_{vi} \alpha^*_{vi} = 1$. Setting $\sigma_{i,k}$ to be the part of σ_i that applies on the qubits $(k-1)m + 1 \leq \gamma \leq km$.

$$p_{incorrect} = \sum_{b',\nu,v,i,j} \alpha_{vi} \alpha^*_{vj} p(\nu) \mathrm{Tr}(P_\perp \bigotimes_{k=1}^{s} |\eta^{\nu_T}_{t_k}\rangle\langle\eta^{\nu_T}_{t_k}| C^{b',\nu_C}$$
$$|b' + c^r\rangle\langle b'| \bigotimes_{k=1}^{s} (\sigma_{i,k} \left(\mathcal{P}_k \bigotimes_{i=1}^{m-n} |\delta^{b',\nu}_{(k-1)m+i}\rangle\langle\delta^{b',\nu}_{(k-1)m+i}| \otimes \rho_{M_k^\nu} \mathcal{P}_k^\dagger \right) \sigma_{j,k}) |b'\rangle\langle b' + c^r| C^{b',\nu_C\dagger}$$

Without loss of generality we can assume that σ_i, σ_j do not change the δ's.

For an arbitrary t_g, the only attacks that give the corresponding term of the sum not equal to zero:

$$P_\perp(C^{b',\nu_C}|b'\rangle\langle b'+c^r|\sigma_{i,t_g} \\ (\mathcal{P}_{t_g}\bigotimes_{i=1}^{m-n}|\delta^{b',\nu}_{(t_g-1)m+i}\rangle\langle\delta^{b',\nu}_{(t_g-1)m+i}|\otimes\rho_{M^\nu_{t_g}}\mathcal{P}^\dagger_{t_g})\sigma_{j,t_g}|b'\rangle\langle b'+c^r|C^{b',\nu_C\dagger})\neq 0$$

are those that (i) produce an incorrect measurement result for qubits $(t_g-1)m+1\leq\gamma\leq t_g m-n$ or (ii) operate non-trivially on qubits $t_g m-n<\gamma\leq t_g m$. We denote this condition by $i\in E_{i,t_g}$ and $j\in E_{j,t_g}$.

We can rewrite the probability by eliminating $P_\perp$ (observing that it applies to a positive operator) and C^{b',ν_C} (by the cyclical property of the trace):

$$p_{incorrect}\leq\sum_{\nu,v,i\in E_{i,t_g},j\in E_{j,t_g}}\alpha_{vi}\alpha^*_{vj}p(\nu)\prod_{k=1}^{s}\mathrm{Tr}(|\eta^{\nu_T}_{t_k}\rangle\langle\eta^{\nu_T}_{t_k}| \\ |b'\rangle\langle b'+c^r|\sigma_{i,k}\left(\mathcal{P}_k\bigotimes_{i=1}^{m-n}|\delta^{b',\nu}_{(k-1)m+i}\rangle\langle\delta^{b',\nu}_{(k-1)m+i}|\otimes\rho_{M^\nu_k}\mathcal{P}^\dagger_k\right)\sigma_{j,k})$$

We extract a trace over R from $\rho_{M^\nu_{t_g}}$. And extract the sums over $\nu_{C,k}$'s from the general sum, where $\nu_{C,k}$ is the subset of random parameters ν_C that are used for the computation of round r:

$$=\sum_{\nu_T,v,i\in E_{i,t_g},j\in E_{j,t_g}}\alpha_{vi}\alpha^*_{vj}p(\nu_T)\prod_{k=1}^{s}\mathrm{Tr}(|\eta^{\nu_T}_{t_k}\rangle\langle\eta^{\nu_T}_{t_k}| \\ |b'\rangle\langle b'+c^r|\sigma_{i,k}\left(\mathcal{P}_k\sum_{\nu_{C,k}}(p(\nu_{C,k})\bigotimes_{i=1}^{m-n}|\delta^{b',\nu}_{(k-1)m+i}\rangle\langle\delta^{b',\nu}_{(k-1)m+i}|\otimes\mathrm{Tr}_R(\rho_{M^\nu_k}))\mathcal{P}^\dagger_k\right)\sigma_{j,k})$$

To take advantage of the blindness property we use the following lemma where the proof is given later.

Lemma 11 (Blindness (excluding the traps))**.**

$$\forall k,\sum_{\nu_{C,k}}p(\nu_{C,k})\bigotimes_{i=1}^{m-n}|\delta^{b',\nu}_{(k-1)m+i}\rangle\langle\delta^{b',\nu}_{(k-1)m+i}|\otimes Tr_R(\rho_{M^\nu_k}) \\ =\frac{I^{t_k}_k}{Tr(I^{t_k}_k)}\otimes|\delta^{\theta_{t_k},r_{t_k}}_{t_k}\rangle\langle\delta^{\theta_{t_k},r_{t_k}}_{t_k}|\otimes|+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}|$$

If $k \neq t_g$, $I_k^{t_k} = \bigotimes_{4m-3n-1} I$ *when* $km-n < t_k \leq km$ *and* $I_k^{t_k} = \bigotimes_{4m-3n-4} I$ *when* $(k-1)m < t_k \leq km-n$ *. And if* $k = t_g$, $I_k^{t_k} = \bigotimes_{4m-3n} I$.

Lemma 11 allows us to simplify the big sum above based on the position of the traps. We also sum over b' since there are no longer any dependencies on it in the sum, obtaining:

$$= \sum_{t_g, v, i \in E_{i,t_g}, j \in E_{j,t_g}} \alpha_{vi} \alpha_{vj}^* p(t_g) \prod_{k=1}^{s} \mathrm{Tr}(\sum_{\substack{km-n<t_k \leq km, \\ \theta_{t_k}}} p(t_k, \theta_{t_k}) |+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}| \sigma_{i,k} (\frac{I}{\mathrm{Tr}(I)} \otimes |+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}|) \sigma_{j,k} + \sum_{\substack{(k-1)m<t_k \leq km-n, \\ r_{t_k}}} p(t_k, r_{t_k}) |r_{t_k}\rangle\langle r_{t_k}| \sigma_{i,k} (\frac{I}{\mathrm{Tr}(I)} \otimes |r_{t_k}\rangle\langle r_{t_k}|) \sigma_{j,k})$$

where $I = \bigotimes_{4m-3n-1} I$ when $k \neq t_g$. And $I = \bigotimes_{4m-3n} I$ when $k = t_g$.

Note that $\sum_{\theta_{t_k}} \mathrm{Tr}(|+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}| \sigma_{i,k} (\frac{I}{\mathrm{Tr}(I)} \otimes |+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}|) \sigma_{j,k})$ is zero if $\sigma_{i,k} \neq \sigma_{j,k}$. The same is true for $\sum_{r_{t_k}} \mathrm{Tr}(|r_{t_k}\rangle\langle r_{t_k}| \sigma_{i,k} (\frac{I}{\mathrm{Tr}(I)} \otimes |r_{t_k}\rangle\langle r_{t_k}|) \sigma_{j,k})$. Therefore we can only keep those terms where $\sigma_{i,k} = \sigma_{j,k}$ and the failure probability becomes:

$$= \sum_{t_g} \sum_{v, i \in E_{i,t_g}} |\alpha_{vi}|^2 p(t_g) \prod_{k=\{1,\ldots,s\} \setminus t_g} (\sum_{\substack{km-n<t_k \leq km, \\ \theta_{t_k}}} p(t_k, \theta_{t_k}) (\langle+_{\theta_{t_k}}| \sigma_{i|t_k} |+_{\theta_{t_k}}\rangle)^2 + \sum_{\substack{(k-1)m<t_k \leq km-n, \\ r_{t_k}}} p(t_k, r_{t_k}) (\langle r_{t_k}| \sigma_{i|t_k} |r_{t_k}\rangle)^2)$$

The rest of the proof is based on a counting argument. For convenience we introduce the following sets for an arbitrary Pauli $\sigma_{i,k}$:

$$A_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = I \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$
$$B_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = X \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$
$$C_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Y \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$
$$D_{i,k} = \{\gamma \text{ s.t. } \sigma_{i|\gamma} = Z \text{ and } (k-1)m+1 \leq \gamma \leq km\}$$

and use the superscript O to denote subsets subject to the constraint $km \geq \gamma \geq km-n+1$.

The failure probability is then:

$$= \sum_{t_g} \sum_{v, i \in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} ((\frac{1}{8m}(8|A^O_{i,k}| + 4|B^O_{i,k}| + 4|C^O_{i,k}|) + \frac{1}{2m}(2|A_{i,k} \setminus A^O_{i,k}| + 2|D_{i,k} \setminus D^O_{i,k}|))$$

Merging the terms:

$$= \sum_{t_g} \sum_{v, i \in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m}(2|A_{i,k}| + |B^O_{i,k}| + |C^O_{i,k}| + 2|D_{i,k} \setminus D^O_{i,k}|)$$

Using the fact that for every k, $|A_{i,k}| + |B_{i,k}| + |C_{i,k}| + |D_{i,k}| = m$:

$$\leq \sum_{t_g} \sum_{v, i \in E_{i,t_g}} |\alpha_{vi}|^2 \frac{1}{s} \prod_{k=\{1,\ldots,s\}\setminus t_g} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$$

The conditions $i \in E_{i,t_g}$ that we obtained at the first part of the proof are translated to $|B_{i,t_g}| + |C_{i,t_g}| + |D^O_{i,t_g}| \geq 1$. In order to be able to use these conditions we need to rewrite the formula. First we expand it:

$$= \frac{1}{s}(\sum_{v, i \in E_{i,1}} |\alpha_{vi}|^2 \prod_{k=\{2,3,\ldots,s\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|) + \sum_{v, i \in E_{i,2}} |\alpha_{vi}|^2 \prod_{k=\{1,3,4,\ldots,s\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|) \ldots + \sum_{v, i \in E_{i,d}} |\alpha_{vi}|^2 \prod_{k=\{1,2,\ldots,s-1\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|))$$

We denote the product term $\prod_{k=\{1,2,3,\ldots,s\}\setminus z} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)$ as $P_{i,z}$. We also denote each set $\{E^*_{i,1} \cap E^*_{i,2} \cap \ldots \cap E^*_{i,s}\}$, where each term $E^*_{i,w}$ is either $E_{i,w}$ or its complement, $E^C_{i,w}$, depending on whether the w-th value of a binary vector y (size s) is 1 or 0 respectively, as $W_{i,y}$. Then we have:

$$= \frac{1}{s}(\sum_{y \setminus (0\ldots0)} \sum_{i \in W_{i,y}, v} (|\alpha_{vi}|^2 \sum_{\{z : y_z = 1\}} P_{i,z}))$$

Let the function $\#y$ give the number of positions i such that y_i=1.

$$= \frac{1}{s}(\sum_{k=1}^{s} \sum_{\{y : \#y = k\}} \sum_{i \in W_{i,y}, v} (|\alpha_{vi}|^2 \sum_{\{z : y_z = 1\}} P_{i,z}))$$

We separately consider the following term for any arbitrary y with $\#y = r$.

$$\sum_{i \in W_{i,y}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} P_{i,z})$$

The condition $i \in W_{i,y}$ means that the following conditions hold together: $\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| \geq 1 : y_w = 1\}$,$\{|B_{i,w}| + |C_{i,w}| + |D^O_{i,w}| = 0 : y_w = 0\}$. We expand:

$$\begin{aligned}
&= \sum_{i \in W_{i,y}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \prod_{k=\{1,2,3,\ldots,s\}\setminus z} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|) \\
&= \sum_{i \in W_{i,y}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \prod_{\{k:y_k=1, k\neq z\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|) \\
&\qquad \prod_{\{k:y_k=0\}} \frac{1}{2m}(2m - |B_{i,k}| - |C_{i,k}| - |D^O_{i,k}|)
\end{aligned}$$

And by using the above conditions:

$$\begin{aligned}
&\leq \sum_{i \in W_{i,y}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \prod_{\{k:y_k=1, k\neq z\}} \frac{1}{2m}(2m-1) \prod_{\{k:y_k=0\}} \frac{1}{2m}(2m) \\
&= \sum_{i \in W_{i,y}} (|\alpha_{vi}|^2 \sum_{\{z:y_z=1\}} \left(\frac{2m-1}{2m}\right)^{r-1} \\
&= \sum_{i \in W_{i,y}} |\alpha_{vi}|^2 r \left(\frac{2m-1}{2m}\right)^{r-1}
\end{aligned}$$

Thus, the bound of our failure probability will be:

$$\begin{aligned}
p_{\text{incorrect}} &\leq \frac{1}{s}(\sum_{k=1}^{s} \sum_{\{y:\#y=k\}} \sum_{i \in W_{i,y},v} |\alpha_{vi}|^2 k \left(\frac{2m-1}{2m}\right)^{k-1}) \\
&= \frac{1}{s}(\sum_{k=1}^{s} k \left(\frac{2m-1}{2m}\right)^{k-1} \sum_{\{y:\#y=k\}} \sum_{i \in W_{i,y},v} |\alpha_{vi}|^2) \\
&= \frac{1}{s}(\sum_{k=1}^{s} c_k k \left(\frac{2m-1}{2m}\right)^{k-1})
\end{aligned}$$

where $c_k = \sum_{\{y:\#y=k\}} \sum_{i \in W_{i,y},v} |\alpha_{vi}|^2$
subject to conditions:

$$\sum_{k=1}^{s} c_k \leq 1 \tag{8.5}$$

and

$$\forall k : c_k \geq 0 \tag{8.6}$$

Proof of Lemma 11. First we define state $|q_i\rangle$ as:

$$\begin{aligned} i \in D \qquad & |q_i\rangle \equiv |d_i\rangle \\ i \notin D \qquad & |q_i\rangle \equiv (\prod_{\{j:j\sim i, j\in D\}} Z^{d_j})|+_{\theta_i}\rangle \end{aligned}$$

By substituting $\rho_{M_k^\nu}$'s and taking the trace over R:

If $k \neq t_g$ the state becomes:

$$\sum_{\nu_{C,k}} p(\nu_{C,k})(\bigotimes_{i=km-n+1}^{km} |q_i\rangle\langle q_i| \bigotimes_{i=(k-1)m+n+1}^{km-n} \left(|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}| \otimes |q_i^\nu\rangle\langle q_i^\nu|\right) \bigotimes_{i=1}^{2} \left(|\delta_{p_{i,k}}^{b',\nu}\rangle\langle\delta_{p_{i,k}}^{b',\nu}| \otimes |q_{p_{i,k}}^\nu\rangle\langle q_{p_{i,k}}^\nu|\right) \otimes I_{4(n-2)}/2^{4(n-2)})$$

where $|q_{p_{i,k}}^\nu\rangle$ denote the first layer pure qubits (a maximum of two) of the k-th graph state, used as padding (dummies) or trap and their positions are defined as: $1+(k-1)m \leq \{p_{1,k}, p_{2,k}\} \leq n+(k-1)m$.

Otherwise, if $k = t_g$ the state becomes:

$$\sum_{\nu_{C,k}} p(\nu_{C,k})(\bigotimes_{i=t_g m-n+1}^{t_g m} |q_i\rangle\langle q_i| \bigotimes_{i=(t_g-1)m+n+1}^{t_g m-n} \left(|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}| \otimes |q_i^\nu\rangle\langle q_i^\nu|\right) \otimes |\delta_u^{\theta_u,r_u}\rangle\langle\delta_u^{\theta_u,r_u}| \otimes |q_u^{\theta_u}\rangle\langle q_u^{\theta_u}| \otimes I_{4(w-1)}/2^{4(w-1)})$$

where $u = (t_g-1)m+1$ is the position of the single pure qubit of the input to the DQC1-MBQC computation.

An implicit assumption was that all δ's that are used to implement the measurements of maximally mixed inputs are maximally mixed states themselves, without any loss of generality.

We define a new controlled unitary:

$$\mathcal{P}'_k = \left(\prod_{\{i:i\notin D,(k-1)m+1\leq i\leq km-n\}} Z_i(-\delta_i)\right) \prod_{\{i:i\notin D_k\}} \prod_{\{j:j\sim i, j\in D_k\}} Z_i(d_j) \tag{8.7}$$

where D_k denotes the set of dummies of subgraph G'_k.

Using this unitary we rewrite the state. If $k \neq t_g$ it becomes:

$$\sum_{\nu_{C_k}} p(\nu_{C,k}) \mathcal{P}'^{\dagger} \mathcal{P}' (\bigotimes_{i=km-n+1}^{km} |q_i\rangle\langle q_i| \bigotimes_{i=(k-1)m+n+1}^{km-n} \left(|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}| \otimes |q_i^{\nu}\rangle\langle q_i^{\nu}| \right) \bigotimes_{i=1}^{2} \left(|\delta_{p_{i,k}}^{b',\nu}\rangle\langle\delta_{p_{i,k}}^{b',\nu}| \otimes |q_{p_{i,k}}^{\nu}\rangle\langle q_{p_{i,k}}^{\nu}| \right) \otimes I_{4(n-2)}/2^{4(n-2)}) \mathcal{P}'^{\dagger} \mathcal{P}'$$

Otherwise:

$$\sum_{\nu_{C,k}} p(\nu_{C,k}) \mathcal{P}'^{\dagger} \mathcal{P}' \bigotimes_{i=t_g m-n+1}^{t_g m} |q_i\rangle\langle q_i| \bigotimes_{i=(t_g-1)m+n+1}^{t_g m-n} \left(|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}| \otimes |q_i^{\nu}\rangle\langle q_i^{\nu}| \right) \otimes |\delta_u^{\theta_u,r_u}\rangle\langle\delta_u^{\theta_u,r_u}| \otimes |q_u^{\theta_u}\rangle\langle q_u^{\theta_u}| \otimes I_{4(w-1)}/2^{4(w-1)}) \mathcal{P}'^{\dagger} \mathcal{P}'$$

After applying the innermost unitary, if $k \neq t_g$:

$$\sum_{\nu_{C,k}} p(\nu_{C,k}) \mathcal{P}'^{\dagger} (\bigotimes_{i=km-n+1}^{km} |q'_i\rangle\langle q'_i| \bigotimes_{i=(k-1)m+n+1}^{km-n} \left(|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}| \otimes |q_i'^{\nu}\rangle\langle q_i'^{\nu}| \right) \bigotimes_{i=1}^{2} \left(|\delta_{p_{i,k}}^{b',\nu}\rangle\langle\delta_{p_{i,k}}^{b',\nu}| \otimes |q_{p_{i,k}}'^{\nu}\rangle\langle q_{p_{i,k}}'^{\nu}| \right) \otimes I_{4(n-2)}/2^{4(n-2)}) \mathcal{P}'$$

where state $|q'_i\rangle$ is defined as:

$$\begin{aligned} i \in D & \qquad |q'_i\rangle \equiv |d_i\rangle \\ i \notin D, \forall k : km \geq i \geq km-n+1 & \qquad |q'_i\rangle \equiv |+_{\theta_i}\rangle \\ i \notin D, \forall k : km-n \geq i \geq (k-1)m+1 & \qquad |q'_i\rangle \equiv |+^{\prime\prime\, b',r<i}_{-a_i} -r_i\pi\rangle \end{aligned}$$

Otherwise, if $k = t_g$:

$$\sum_{\nu_{C,k}} p(\nu_{C,k}) \mathcal{P}'^{\dagger} (\bigotimes_{i=t_g m-n+1}^{t_g m} |q'_i\rangle\langle q'_i| \bigotimes_{i=(t_g-1)m+n+1}^{t_g m-n} \left(|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}| \otimes |q_i'^{\nu}\rangle\langle q_i'^{\nu}| \right) \otimes |\delta_u^{\theta_u,r_u}\rangle\langle\delta_u^{\theta_u,r_u}| \otimes |q_u'^{\theta_u}\rangle\langle q_u'^{\theta_u}| \otimes I_{4(w-1)}/2^{4(w-1)}) \mathcal{P}'$$

It is essential for the proof that each term with index i in the tensor product depends only on parameters with index $\leq i$ and the term with index $(t_g-1)m+1$ (input qubit) and the trap qubit and its measurement angle (if it is not an output) depend only on their own parameters. This allows to break the summations and calculate them iteratively from left to right, given the following:

$$\sum_{d_i} p(d_i)|d_i\rangle\langle d_i| = \frac{I}{2}$$

$$\sum_{\theta_i} p(\theta_i)|+_{\theta_i}\rangle\langle+_{\theta_i}| = \frac{I}{2}$$

$$\sum_{\theta_i,r_i,d_i} p(\theta_i,r_i,d_i)|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}|\otimes|d_i\rangle\langle d_i| = \frac{I_4}{2^4}$$

$$\begin{aligned}
&\sum_{\theta_i,r_i} p(\theta_i,r_i)|\delta_i^{b',\nu}\rangle\langle\delta_i^{b',\nu}|\otimes|+_{-a_i^{''b',r_{<i}}-r_i\pi}\rangle\langle+_{-a_i^{''b',r_{<i}}-r_i\pi}| \\
&= \sum_{r_i} p(r_i)\left(\sum_{\theta_i} p(\theta_i)|a_i^{''b',r_{<i}}+\theta_i+r_i\pi\rangle\langle a_i^{''b',r_{<i}}+\theta_i+r_i\pi|\right) \\
&\qquad\otimes|+_{-a_i^{''b',r_{<i}}-r_i\pi}\rangle\langle+_{-a_i^{''b',r_{<i}}-r_i\pi}| \\
&= \sum_{r_i} p(r_i)\frac{I_3}{2^3}\otimes|+_{-a_i^{''b',r_{<i}}-r_i\pi}\rangle\langle+_{-a_i^{''b',r_{<i}}-r_i\pi}| \\
&= \frac{I_4}{2^4}
\end{aligned}$$

where $I_n = \bigotimes_n I$. The last step was possible because each corrected computation angle a_i'' depends only on past r's.

And finally (for $u=(t_g-1)m+1$),

$$\begin{aligned}
&\sum_{\theta_u,r_u} p(\theta_u,r_u)|\delta_u^{\theta_u,r_u}\rangle\langle\delta_u^{\theta_u,r_u}|\otimes|+_{-a_u'-r_u\pi}\rangle\langle+_{-a_u'-r_u\pi}| \\
&= \sum_{r_u} p(r_u)\left(\sum_{\theta_u} p(\theta_u)|a_u'+\theta_u+r_u\pi\rangle\langle a_i'+\theta_u+r_u\pi|\right) \\
&\qquad\otimes|+_{-a_u'-r_u\pi}\rangle\langle+_{-a_u'-r_u\pi}| \\
&= \frac{I_4}{2^4}
\end{aligned}$$

For $k\neq t_g$, if $km\geq t_k\geq km-n+1$ the above procedure will eventually give:

$$\mathcal{P}'^{\dagger}(\frac{I_{4m-3n-1}}{2^{4m-3n-1}} \otimes |+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}|)\mathcal{P}' = \frac{I_{4m-3n-1}}{2^{4m-3n-1}} \otimes |+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}|$$

If $km-n \geq t_k \geq (k-1)m+1$ the above procedure will eventually give:

$$\mathcal{P}'^{\dagger}(\frac{I_{4m-3n-4}}{2^{4m-3n-4}} \otimes |\delta_{t_k}^{\nu_T}\rangle\langle\delta_{t_k}^{\nu_T}| \otimes |+_{r_{t_k}\pi}\rangle\langle+_{r_{t_k}\pi}|)\mathcal{P}' = \frac{I_{4m-3n-4}}{2^{4m-3n-4}} \otimes |\delta_{t_k}^{\nu_T}\rangle\langle\delta_{t_k}^{\nu_T}| \otimes |+_{\theta_{t_k}}\rangle\langle+_{\theta_{t_k}}|$$

And for $k = t_g$ the result will be: $\bigotimes_{4m-3n} I$, which concludes the proof.

□

Part IV

Verification and Quantum Security

Chapter 9

Overview

Looking at the bigger picture, quantum verification can be seen as a member in the family of quantum security primitives, i.e. primitives that intend to secure the communication and manipulation of *quantum* information (as opposed to quantum protocols used to promote classical security). Starting with an insecure quantum channel, one wants to *encrypt* a state before transmitting it so that no eavesdropper gains any new information about the state. The next step, is to take measures against malleability, i.e. the possibility of the adversary who controls the channel to change the encrypted state so that it encodes a different message. This is achieved by quantum *authentication*, where some extra elements added on top of encryption allow the receiver to decide if it accepts or rejects the state it receives. Verification can be seen as the authentication of a state that has gone through a transformation and the receiver has to decide if the final output is the correct output of the computation. An extra property in the delegated computation scenario, which does not necessarily assume verification, is the property of blindness, where the prover does not learn anything about the transmitted state and the computation to be performed - thus state and computation should be encrypted.

The goal of this part is to study the relations of these primitives, and use them to derive new results or point to possibly helpful directions. In particular, taking the same approach as in [Barnum et al., 2002], which studies the relation between encryption and authentication, and in [Aharonov et al., 2010], which provides a verification protocol based on authentication, we build the verification protocol of [Fitzsimons and Kashefi, 2012] out of the primitives of encryption (and blindness) and authentication. We pose the question of what are the necessary ingredients to build the simplest verification protocol and finally we try to make connections to the impossibility of classical verifier question, which is still unanswered and central to the field.

The main contributions of this part are:

- Providing a protocol that enables the authentication of a quantum state that is transmitted through an untrusted quantum channel, given that the two end parties share a secret classical key. This protocol is based on the idea of trapification, as in [Fitzsimons and Kashefi, 2012], and quantum encoding. Note that these results where proven independently in [Broadbent et al., 2012] but with a different motivation.

- Drawing the roadmap on the sufficient and the necessary steps for building a verification protocol and relate to the classical verifier (im)possibility question which is our motivation.

9.1 Preliminaries and Related Work

Before studying their relations, we need to provide the formal definitions of the security primitives we consider in this part.

Quantum encryption and quantum blindness are two security primitives that relate to the secrecy or confidentiality of information in the presence of an adversary. A quantum encryption scheme is defined in the context of insecure quantum channels, where one party (Alice) wants to communicate a quantum state to another party (Bob) and needs to keep the transmitted state hidden from any possible eavesdropper (Eve) [Barnum et al., 2002].

Definition 15. *An encryption scheme with error* ε *for quantum states hides information so that if* ρ_0 *and* ρ_1 *are any two distinct encrypted states, then the trace distance* $D(\rho_0 - \rho_1) = \frac{1}{2}Tr|\rho_0 - \rho_1| \leq \varepsilon$

Notice that this definition considers approximate encryption, with perfect secrecy (perfect encryption) being the case when $\varepsilon = 0$.

Quantum blindness is defined in the context of insecure delegated quantum computing. We already gave the definition of perfect blindness in Section 1.2, and discussed it extensively in the previous parts. If we want to compare it with quantum encryption, we can imagine that Alice sends to a channel an encrypted quantum state and an encrypted description of a transformation. The untrusted channel (now Bob) is able to apply the transformation to the state without learning anything about the original (decrypted) state and the original (decrypted) transformation applied on it. Notice that such a protocol

might require many rounds of interaction between Alice and Bob and the requirement is that no round leaks any of the confidential information. Approximate blindness can be defined similarly to the definition of perfect blindness:

Definition 16 (Approximate Blindness)**.** *Let P be a protocol for delegated computation: Alice's input is a description of a computation on a quantum input, which she needs to perform with the aid of Bob and return the correct quantum output. Let ρ_{AB} denote the joint initial state of Alice and Bob and σ_{AB} their joint state after the execution of the protocol, when Bob is allowed to do any deviation from the correct operation during the execution of P, averaged over all possible choices of random parameters by Alice. The protocol P is ε-blind if*

$$\forall \rho_{AB} \in \mathcal{L}(\mathcal{H}_{AB}), \exists \mathcal{E} : \mathcal{L}(\mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_B), \text{ s.t. } \quad Tr_A(\sigma_{AB}) \approx_\varepsilon \mathcal{E}(Tr_A(\rho_{AB})) \tag{9.1}$$

Quantum Authentication is defined using the following scheme [Barnum et al., 2002]:

Definition 17. *A quantum authentication scheme (QAS) is a pair of polynomial time quantum algorithms A and B together with a set of classical keys $\mathcal{K}$ such that:*

- *A takes as input an m-qubit message system M and a classical key $k \in \mathcal{K}$ and outputs a transmitted system T of $m+t$ qubits.*
- *B takes as input the (possibly altered) transmitted system T and the classical key k and outputs two systems: an m-qubit message state M', and a single qubit V which indicates acceptance or rejection. The classical basis states of V are called $|ACC\rangle, |REJ\rangle$ by convention.*

For any fixed key k, we denote the corresponding super-operators by A_k and B_k.

A QAS should have the following property [Barnum et al., 2002]:

Definition 18. *A QAS is secure with error ε for a state $|\psi\rangle$ if it satisfies:*
Completeness: $\forall k \in \mathcal{K} : B_k(A_k(|\psi\rangle\langle\psi|)) = |\psi\rangle\langle\psi| \otimes |ACC\rangle\langle ACC|$
Soundness: For all CPTP-maps O which represent adversary's intervention:

$$Tr\left(P_{good} \sum_k p(k) B_k(O(A_k(|\psi\rangle\langle\psi|)))\right) \geq 1 - \varepsilon$$

where P_{good} is the projection into the desired ('good') subspace :

$$P_{good} = |\psi\rangle\langle\psi| \otimes I_V + (I_M - |\psi\rangle\langle\psi|) \otimes |REJ\rangle\langle REJ|$$

A QAS is secure with error ε *if it is secure with error* ε *for all states* $|\psi\rangle$.

Equivalently we could define soundness using the 'bad' projector ($P_{\text{bad}} = I - P_{\text{good}} = (I_M - |\psi\rangle\langle\psi|) \otimes |\text{ACC}\rangle\langle\text{ACC}|$) and demand it is small.

Also, it is important to clarify that if we ask the condition to be 'the probability of the state collapsing into the incorrect subspace *when* the state is accepted, is low' (therefore making the probability conditional) this would make such a protocol much harder to construct. The security of the existing protocols will always be broken by an attacker who always replaces the state by a random state, i.e the state seen by Alice is the maximally mixed state. Then - for the existing protocols that we study - there will always be some probability that the state is accepted and, when it is accepted, the state will collapse with a fixed (not bounded by ε) probability into the incorrect subspace. *

Verification, already defined in Section 1.2, can be seen as an extension of QAS in the delegated computing scenario, were the untrusted 'channel' is also able to perform a computation on the state transmitted, and the receiver (which could be the same as the sender in this case, so that there is no need for classical secret key distribution) is able to tell if the state received is the correct transformed one. It is important here to emphasise the different 'flavours' of verification definitions, depending on whether the input is quantum or classical and whether the output is quantum or classical. For example, the question of a classical verifier protocol has no clear meaning if we consider a protocol that admits an arbitrary quantum input, which is unknown to the verifier. In that case the verifier will have to measure the state and therefore lose some of the information before transmitting it to the prover. In the case of a classical input verification protocol, without loss of generality we can assume that the unencrypted input to the computation is the blank state $|0\rangle^{\otimes m}$. It is important to stress that when we speak about classical or quantum input we speak only about the input to the computation that is to be delegated and not about any auxiliary states send by Alice to Bob to accommodate this computation (e.g. in MBQC the $|+\rangle$ states of the subgraph $G \setminus I$).

A known result, proved in [Barnum et al., 2002], is the relation between quantum authentication and encryption. Quoting from this paper:

*For example in the trap-based authentication scheme, when the attacker replaces the state by the maximally mixed state and we condition on the traps being accepted the rest of the state is still in the maximally mixed state therefore the conditional probability of failure cannot be bounded by an arbitrary ε

For classical information, authentication and encryption can be considered completely separately, but in this section we will show that quantum information is different. While quantum states can be encrypted without any form of authentication, the converse is not true: any scheme which guarantees authenticity must also encrypt the quantum state almost perfectly.

More formally [Barnum et al., 2002]:

Theorem 16. *A Quantum Authentication scheme with error* ε *is an encryption scheme with error at most* $4\varepsilon^{1/6}$.

Proof intuition for the above theorem comes from considering the following extreme case. Let $\rho_{|0\rangle}$ and $\rho_{|1\rangle}$ be the density operators of the encodings of the computational basis states of an input qubit according to the authentication scheme. Let $\rho_{|0\rangle}$ and $\rho_{|1\rangle}$ be perfectly distinguishable quantum states which is the opposite end to perfect encryption. Since, when proving soundness of a QAS, the prover is able to apply any possible attack, he might apply the following attack unitary: the basis elements of the support of $\rho_{|1\rangle}$ get a -1 phase, while the orthogonal elements (which include the support of $\rho_{|0\rangle}$) remain unchanged. Due to linearity, this attack will transform the encoding of the state $|0\rangle + |1\rangle$ to the encoding of the state $|0\rangle - |1\rangle$, thus the detection will find no error and the decoding will give an incorrect state. The analogous attack is not possible in classical state authentication. After a series of lemmas, they prove that there are similar attacks on states that are not perfectly distinguishable, but not totally indistinguishable, and therefore prove the theorem.

Another interesting result that relates quantum blindness with a classical impossibility statement comes from [Morimae and Koshiba, 2014]:

Theorem 17. *If a classical input delegated quantum computing protocol, which uses affine encryption, can give perfect blindness then BQP* $\subseteq$ *BPP.*

Notice that for a classical input blind protocol, since we assume the input to be the blank state, we need to hide only the description of the transformation, which is a classical state. The assumption is that this description is encrypted by a classical affine encryption, e.g. a classical one-time-pad.

Finally, it is important to mention that all definitions of security seen in this chapter (and in fact in the whole of this thesis) fall to the category of information theoretical security. One can consider also computational theoretical security, where the security depends on the computational restrictions imposed on the adversary, and derive different results.

Chapter 10

From Quantum Encryption to Verification

We start this chapter by building an authentication protocol that is inspired by the FK protocol. We use the same trapification technique, i.e. placing a number of traps between the normal qubits that contain the information to be transmitted. First we present a protocol with a single trap and later we place more traps and employ a QECC encoding for probability amplification.

10.1 An Authentication Protocol

In Protocol 10 the description of a QAS protocol based on traps is given. It applies to the context of secure communication: Alice sends a state to Bob and if state was unaltered Bob receives the correct state and if the state was altered (in any possible way) Bob declares correct an incorrect state with low probability.

Protocol 10 Trap-based QAS

Alice's input. A quantum state $|\psi\rangle$ of size $n = m - 1$ qubits.
Bob's output. A quantum state (which should be $|\psi\rangle$ for an honest protocol) and an indicator bit for the acceptance or rejection of the output.

The protocol

1. Alice selects a random quantum one-time padding key $(\mathbf{x}, \mathbf{z}) \in_R \{0,1\}^{2m}$.

Protocol 10 Cont'd

2. Alice selects a random position $1 \leq t \leq m$ for placing the trap and one random bit to select the stabilizer of the trap between X or Y (or any pair of non-commuting stabilizers ex. X and Z)
3. Alice produces the 'logical' state by adding the trap qubit $|\eta\rangle\langle\eta|$ and applying the Pauli padding $P^{x,z}$ to the whole state of size m.
4. Alice sends state to Bob
5. We assume that Bob learns from Alice via a secure classical channel the following parameters: quantum one-time padding key, random position and type of the trap.
6. Bob undoes the Pauli padding using the same key.
7. Bob does a stabilizer measurement on the trap (depending on the choice of parameters) and if the result is 1 (or equivalently the error vector is zero) he accepts otherwise he rejects.

Theorem 18. *Protocol 10 is a QAS with* $\varepsilon = \frac{2m-1}{2m}$

Proof. Since the declaration of the correctness of the state depends only on the trap measurement we can write the failure probability as the probability of the trap measurement giving 1 (state declared correct) and the state being corrupted (collapsing to the orthogonal subspace after measurement), averaged over all keys:

$$p_{\text{fail}} = \sum_{x,z,t,\eta} p(x,z,t,\eta) \operatorname{Tr}\left(|\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)(P^{x,z}\mathcal{E}(P^{x,z}(|\eta\rangle\langle\eta|_t \otimes |\psi\rangle\langle\psi|)P^{x,z})P^{x,z})\right) \tag{10.1}$$

where $\mathcal{E}$ is an arbitrary CPTP-map modelling the channel. $p(x,z,t,\eta)$ is the probability of choosing a particular set of parameters (all are drawn from uniform random distribution).

Writing the channel operator in the Pauli basis $\mathbb{P}_m$:

$$= \sum_{x,z,t,\eta} p(x,z,t,\eta)$$

$$\mathrm{Tr}\left(|\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)(\sum_{i,j} \alpha_{i,j} P^{x,z} P^i P^{x,z} (|\eta\rangle\langle\eta|_t \otimes |\psi\rangle\langle\psi|) P^{x,z} P^j P^{x,z}) \right) \quad (10.2)$$

where P^i, P^j take all possible values from $\mathbb{P}_m$ and each term has complex coefficient $\alpha_{i,j}$.

We can use the Pauli twirling lemma given as Lemma 2, which states that a random Pauli rotation converts the channel attack to a mixture of Pauli attacks (not a uniform random mixture, but *some* mixture of Pauli operators). After the lemma is applied we have:

$$= \sum_{t,\eta} p(t,\eta) \mathrm{Tr}\left(|\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)(\sum_i \alpha_i P^i (|\eta\rangle\langle\eta|_t \otimes |\psi\rangle\langle\psi|) P^i) \right) \quad (10.3)$$

where P^i takes all possible values from $\mathbb{P}_m$. Elements $\{\alpha_i\}$ are real numbers which depend on $\mathcal{E}$ and sum up to 1.

We can eliminate the all identity 'attacks' (say with index $i = 0$) from the sum because in this case the state is always (for any t) projected to its orthogonal subspace giving probability 0:

$$= \sum_{t,\eta} p(t,\eta) \mathrm{Tr}\left(|\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)(\sum_{i\setminus 0} \alpha_i P^i (|\eta\rangle\langle\eta|_t \otimes |\psi\rangle\langle\psi|) P^i) \right)$$

$$= \sum_{i\setminus 0} \alpha_i \left(\sum_{t,\eta} p(t,\eta) \mathrm{Tr}\left(|\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)(P^i (|\eta\rangle\langle\eta|_t \otimes |\psi\rangle\langle\psi|) P^i) \right) \right) \quad (10.4)$$

Since any (non-identity) Pauli attack is possible we need to examine each possible attack separately and find a common maximum. This will determine the upper bound since the formula contains a convex combination of attack terms.

We observe that every (non-identity) Pauli attack will be caught by at least one stabilizer measurement in the sum over t, η: A (non-identity) Pauli attack will have at least one X, Y or Z at one position, say t'. This will be 'caught' respectively by Y, X or Y (and X)-stabilized traps when $t = t'$ (ex. $\langle +|Z|+\rangle^2 = 0$). Given that we have $2m$ equiprobable elements in the sum, a maximum for any attack is always $\frac{2m-1}{2m}$, which gives the global upper bound for the failure probability.

□

Therefore the single trap protocol is a valid QAS with completeness 1 and soundness $\frac{2m-1}{2m}$. Of course, a practically useful QAS should have a better soundness bound.

From the above one can easily see that the random keys related to the position and the choice of stabilizer for the trap are *necessary* for the protocol to work, since we need to detect (at least) *any* Pauli attack at *any* position for the detection procedure to work for any type of channel (thus giving a QAS). The use of Pauli key is justified by the need to eliminate complex coefficients $\alpha_{i,j}$ and the cross terms P_i, P_j so that we can study the attack as a convex combination of unitary terms. By Theorem 16 it follows that some level of encryption is indeed a necessary element of any QAS protocol.

It is worthwhile mentioning that the same bound is true even when the adversary has some prior knowledge of the state, because the proof works for any possible attack by the adversary that is independent of the secret keys.

Another version of the trap-based QAS is presented in Protocol 11, where the error probability is amplified to exponential on a security parameter d.

Protocol 11 Amplified Trap-based QAS

Alice's input. A quantum state $|\psi\rangle$ of size n qubits.

Bob's output. A quantum state (which should be $|\psi\rangle$ for an honest protocol) and an indicator bit for the acceptance or rejection of the output.

The protocol

1. The protocol is parametrized by a QECC that detects d errors.
2. Alice selects a random quantum one-time padding key $(\mathbf{x},\mathbf{z}) \in_R \{0,1\}^{2m}$.
3. Alice selects a set T of random positions $1 \leq t \leq m$ for placing the N traps and one random bit for each trap to select the stabilizer the trap between Pauli X and Y (or any pair of non-commuting stabilizers ex. X and Z)
4. Alice encodes the quantum input of size n using this particular QECC into a codeword of size m'.
5. Alice produces the final state by adding the N trap qubits $|\eta\rangle\langle\eta|_t$ at random positions and applying the Pauli padding $P^{x,z}$ to whole state of size $m = m' + N$.

Protocol 11 Cont'd

6. Alice sends state to Bob
7. Bob undoes the Pauli padding using the same key (we assume that Bob knows all keys from before).
8. Bob applies a syndrome measurement on the m' qubits and aborts if an error is detected. (or corrects the error)
9. Bob does stabilizer measurement on the traps (depending on the choice of parameters) and if the result is 1 (or equivalently the error vector is zero) he accepts otherwise he rejects.

Theorem 19. *Protocol 11 is a QAS with* $\varepsilon = (1 - \frac{N}{2m})^d$

Proof. Since the declaration of the correctness of the state depends only on the trap measurement we can write the error probability as the probability of the trap measurement giving 1 and the state being corrupted, as before, averaged over all keys:

$$p_{\text{fail}} = \sum_{x,z,T,\eta} p(x,z,T,\eta)$$
$$\text{Tr}\left(\otimes_{t\in T}|\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)\mathcal{D}(P^{x,z}\mathcal{E}(P^{x,z}(\otimes_{t\in T}|\eta\rangle\langle\eta|_t \otimes |\Psi\rangle\langle\Psi|)P^{x,z})P^{x,z})\right) \quad (10.5)$$

where $\mathcal{E}$ is an arbitrary CPTP-map modelling the channel. Term $p(x,z,t,\eta)$ is the probability of choosing a particular set of parameters (all are drawn from a uniform random distribution). State $|\Psi\rangle\langle\Psi|$ is the encoding of the state $|\psi\rangle\langle\psi|$ transmitted, according to the specific QECC. Map $\mathcal{D}$ is a CPTP-map that represents the decoding procedure for the QECC.

We can use the Pauli twirling lemma again to get:

$$= \sum_k \sum_{T,\eta} p(T,\eta)$$
$$\text{Tr}\left(\otimes_{t\in T}|\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)\mathcal{D}(\sum_i \alpha_{i,k}P^i(\otimes_{t\in T}|\eta\rangle\langle\eta|_t \otimes |\Psi\rangle\langle\Psi|)P^i)\right) \quad (10.6)$$

where P^i takes all possible values from $\mathbb{P}_m$.

Analysis of attacks: Assuming that the decoding procedure $\mathcal{D}$ will reconstruct the correct state $|\psi\rangle\langle\psi|$ unless the footprint of the Pauli error is bigger that d, we can

eliminate all P^i's that contain more or equal than $m-d$ identities. We denote this set of Pauli operators as E.

$$= \sum_k \sum_{i \in E} \alpha_{i,k}$$

$$\left(\sum_{T,\eta} p(T,\eta) \mathrm{Tr} \left(\otimes_{t \in T} |\eta\rangle\langle\eta|_t \otimes (I - |\psi\rangle\langle\psi|)(P^i(\otimes_{t \in T} |\eta\rangle\langle\eta|_t \otimes |\Psi\rangle\langle\Psi|)P^i) \right) \right) \quad (10.7)$$

The above can be simplified to:

$$= \sum_k \sum_{i \in E} \alpha_{i,k} \left(\sum_{T,\eta} p(T,\eta) \mathrm{Tr} \left(\otimes_{t \in T} (\langle\eta|_t P^{i|t} |\eta\rangle_t)^2 \right) \right) \quad (10.8)$$

We continue by breaking the whole state (of size m) into N partitions of the same size (m/N), each one containing a single trap.

$$= \sum_k \sum_{i \in E} \alpha_{i,k} \left(\mathrm{Tr} \left(\otimes_{\gamma=1}^{N} (\sum_{t_\gamma, \eta} p(t_\gamma, \eta) \langle\eta|_{t_\gamma} P^{i|t_\gamma} |\eta\rangle_{t_\gamma})^2 \right) \right) \quad (10.9)$$

A common upper bound for the elements of the inner summation is $\frac{2m/N - w_{i,\gamma}}{2m/N} = 1 - \frac{N w_{i,\gamma}}{2m}$, where $w_{i,\gamma}$ is the number of non-identity elements of $P^{i|t_\gamma}$:

$$\begin{aligned}
&= \sum_k \sum_{i \in E} \alpha_{i,k} \prod_{\gamma=1}^{N} (1 - \frac{N w_{i,\gamma}}{2m}) \\
&\leq \sum_k \sum_{i \in E} \alpha_{i,k} \prod_{\gamma=1}^{N} (1 - \frac{N}{2m})^{w_{i,\gamma}} \\
&= \sum_k \sum_{i \in E} \alpha_{i,k} (1 - \frac{N}{2m})^{\sum_{\gamma=1}^{N} w_{i,\gamma}} \\
&= \sum_k \sum_{i \in E} \alpha_{i,k} (1 - \frac{N}{2m})^{w_i} \\
&\leq \sum_k \sum_{i \in E} \alpha_{i,k} (1 - \frac{N}{2m})^{d} \\
&\leq (1 - \frac{N}{2m})^{d}
\end{aligned}$$

□

10.2 A Recipe for Quantum Authentication

By looking in the structure of the trap-based QAS and the polynomial QAS on which the ABE verification protocol is based we trace some common properties. These might

imply the existence of a more general methodology for constructing a QAS such as:

1. Encode the quantum message to enable a random channel error detection procedure. In the trap-based QAS protocol we insert an extra trap subsystem which is measured to detect an error. This is indeed an encoding of the state, since the transmitted system contains the original system and some overhead (the trap) used for error detection - e.g. a random phase flip channel will always be detected with some fixed probability by a $|+\rangle$ trap qubit padded at the end of the message. In polynomial QAS protocol we encode the message using the quantum polynomial code.

2. Randomise the encoding to enable detection of arbitrary (adversarial) Pauli errors. In the trap-based QAS protocol this corresponds to the random position and type of the trap. In the polynomial QAS protocol it is the random sign key for the values of the polynomial. One other example of such randomized families of codes are the family of *purity testing codes* in [Barnum et al., 2002] which are used to purify shared EPR pairs.

3. Encrypt the encoded state by a quantum one-time-padding to enable detection of arbitrary (adversarial) errors. This is performed in both trap-based QAS and polynomial QAS protocols. It emerges naturally in the QAS in [Barnum et al., 2002] from the Pauli corrections that follow the teleportation of the state from Alice to Bob using the EPR pairs.

It is worthwhile mentioning that in the field of quantum error correction it suffices to be able to correct Pauli errors to correct any type of error (based on a linearity argument). This is not the case here, since we always need the extra encryption to detect adversarial non-Pauli errors by reducing them to Pauli errors by the encryption.

10.3 Authentication to Verification and Classical Impossibility

We argue how we can go from the QAS to the verification protocol in the case of both FK and ABE protocols.

In the FK protocol we add the extra element of blindness. This happens for the following reason: Alice asks Bob to measure the qubits following the MBQC pattern,

and since some of them are trap measurements, all the measurement angles need to be hidden so that Bob cannot distinguish between the case of measuring a trap or a normal qubit and cheat selectively (and also should not know what was the type of the trap). Blindness of the measurement angles is achieved by rotating them by a random angle, thus effectively applying a classical full one-time-pad on the bits that describe the angles. Crucially, to cancel this rotation we need to pre-rotate all the quantum states (input and auxiliary qubits) that Alice sends to Bob, thus the necessity of a quantum verifier. Also, the quantum state that Bob holds at any step of the execution of the protocol is encrypted with a quantum one-time-pad. This comes from the following facts: firstly, in the case of quantum input, the input qubits are quantumly one-time-padded when sent to Bob. Secondly, a quantum one-time-pad is kept at any step of the protocol by the random Z rotations on the measurement angles (parameter r_i) effectively creating a random Pauli X rotation after the teleportation on the target qubit. Finally, in the case of quantum output, the auxiliary $|+\rangle$ qubits that are sent to Bob are randomly Z rotated which adds to the one-time pad of the final state. Alice needs always to keep track of the current Pauli corrections that undo the quantum one-time-pad.

In the ABE protocol blindness of computation is not necessary, since the measurement angles of Bob are always fixed. Crucially, the states that Bob receives, should be already secretly encoded by the QAS so Alice has to have a constant size quantum computer. The state remains quantumly one-time-padded at any step of the computation since all the input and the auxiliary states used are quantumly one-time padded with independent keys. However, the keys change after applying the logical operations, and Alice needs to keep track of these changes, as it is also the case in the FK protocol.

In both FK and ABE protocols, the randomization of the encoding, type and position of the trap in the FK protocol or sign key in the ABE protocol, are not affected by the computation. Thus the state at any stage of the protocols can be seen as a QAS state and Alice (being both the sender and the receiver of the state) can check if the state is encoded in the correct code space.

Having seen a way of constructing a verification protocol, starting from the principles of encryption and, in the case of the FK protocol, blindness we turn our attention in the backwards direction, i.e. what are the necessary elements in a verification protocol.

We start by the observation that a verification protocol with quantum input and quantum output must necessarily 'contain' a QAS scheme and consequently a quantum encryption scheme.

Lemma 12. *Any quantum-input / quantum-output* ε*-verifiable protocol is also a Quan-*

tum Authentication scheme with error ε *and a quantum encryption with error at most* $4\varepsilon^{1/6}$.

Proof. The simulation is straightforward. Alice encodes the quantum input according to the verification protocol and sends it through an untrusted quantum channel to another trusted party say, Bob, with whom Alice has shared the secret key of the verification (verification assumes the existence of secure classical channels). Bob applies the decoding procedure for the verification protocol with the correct key and accepts or rejects the output. The untrusted channel can apply any CPTP-map and by the properties of any verification protocol where the delegated computation is the identity and the attack of the untrusted prover is the attack of the channel, the probability of the state sent from Alice to Bob to be corrupted and Bob accepting is bounded by ε. The second implication follows from Theorem 16. □

Is it true that any verification protocol contains a blind protocol? If it was the case then by Theorem 17 we would make a big step towards the much sought result of impossibility of classical verifier. However, this is not the case for some verification protocols - we already have an example on a non-blind verification protocol, the ABE protocol.

A short discussion follows for the possible next steps towards the classical verifier impossibility result. A common element of all existing verification protocols with a preparing verifier and BQP prover is that encryption is always a random Pauli rotation (quantum one-time-pad) and that the computation is always based on a measurement-based model (Clifford unitary operators and single qubit measurements, which allows the Pauli encryption to 'propagate' until the end of the computation).

Let us assume that there is a verification protocol with a classical verifier and with only classical interaction with the single prover. Also this verification protocol is universal, i.e. can verify any problem in BQP with failure probability bounded by ε. The computation of the prover can be written, as usual, as the honest computation, which must be a poly-time quantum circuit or equivalent model of quantum computation since BQP is believed not to be classically simulable, and any possible quantum deviation at any computational step. Then, if the state is distinguishable at any step, Bob will be able to apply an (-1) phase attack of the kind mentioned in the proof sketch of Theorem 16 and produce a different encoded state that can potentially lead to an incorrect, but undetectable by the verification procedure, output. The question is whether classical encryption, introduced by the restricted computationally devices of the verifier, is

enough to make the internal state of Bob indistinguishable at any step, for any universal quantum computation, and thus avert this kind of attack. We think that the study of such encryption techniques, moving further from the standard Pauli rotations used in all single prover protocols so far, can provide an insight in the existence or not of classical verifier verification schemes. This, however, cannot preclude the existence of verification techniques that do not use an error detection procedure as a final step. Also, one needs to consider verification schemes with a weaker verifier, e.g. with the ability to prepare a logarithmic number of qubits. We think that these questions may be relevant to further exploration of this topic.

Part V

Blindness and Classical Security

Chapter 11

Overview

In this part we turn our attention from solving a quantum security task to solving a classical security task by using quantum means. The principle of secure delegated computation, where a remote device computes on encrypted data without decrypting them first [Rivest et al., 1978], has been applied succesfully in the classical setting (e.g. in secure multi party computation [Yao, 1982, Goldwasser, 1997] or fully homomorphic encryption [Gentry, 2009]). However, one can envision that adapting the techniques of quantum blind computation, by introducing some modest quantum enhancement on the client-server system, may lead to improvement in the classical setting. The idea of utilizing quantum means to achieve classical security has already proved successful [Bennett, 1984]. The goal here is to provide a protocol, that demonstrates a novel usage of quantum, and uses quantum devices already available in scientific or commercial labs. Blindness can also be seen as a stepping stone for verifiability, and while the latter will not be considered in this part of the thesis, this extension is left as an open question.

By restricting the task to classical computation only, we derive a family of protocols for unconditionally secure delegation of any classical computation to a remote server that has access to basic quantum devices. Concretely, we present how a client with access to a non-universal classical gate (e.g. parity gate) could securely delegate the computation of a universal gate (e.g. NAND gate) to an untrusted server with capability of manipulating a single qubit. We note that, in this part, the security of the protocol pertains to the privacy, or confidentiality, or blindness of the protocol - we require that no information about the clients input leaks to the server.

The main contributions of this part are:

- A family of protocols (called Secure-NAND) to blindly calculate a NAND gate, with the client classically restricted to XOR operations and random bit generation

and quantumly restricted to manipulation of either a 3-qubit state (GHZ) or a single qubit state. Also, a proof is given for the impossibility of achieving the same task when the devices are purely classical.

- A no-go result for a quantum off-line version of the Secure-NAND. In fact, in all Secure-NAND protocols the state of the qubits that the client sends to the server depends on the inputs of the client, i.e. the classical bit that are the input to NAND. This is in contrast to protocols for secure delegated quantum computation, presented in the previous parts, where the quantum state is independent of the input and description of the computation.

This part of the thesis was published in [Dunjko et al., 2016] as joint work with Vedran Dunjko and Elham Kashefi and is contained here with the permission of the main author of the paper Vedran Dunjko. All authors contributed equally in the construction of the protocols and the classical impossibility result (Theorem 20), all presented in Section 12.1 of this thesis. The optimality results for our protocols presented in Section 12.2 are predominantly work of Vedran Dunjko and the author of this thesis contributed in providing feedback in the form of comments.

11.1 Main Results

The general idea behind our protocol for the secure computation of the universal NAND gate is based on the following simple fact presented for the first time in [Anders and Browne, 2009] which was further utilised for a multi-party cryptographic setting in [Loukopoulos and Browne, 2010], the role of contextuality in computational speedup in [Raussendorf, 2013b] and the relation of entangled quantum states and multi-party computational games in [Hoban et al., 2011]. Let M^0 to denote a Pauli-X measurement and M^1 a Pauli-Y, then the three qubit measurement $M^a \otimes M^b \otimes M^{a\oplus b}$ of the GHZ state (denoted here as $|\Psi\rangle = 1/\sqrt{2}(|001\rangle - |110\rangle)$) computes NAND$(a,b)$ (see more details in Section 11.1). We then show how instead of switching the measurements (based on the input a and b) one can simply apply the pre-rotation operation based on a, b and $a \oplus b$ to the GHZ state and then the Pauli-X measurements of all three qubits achieve the same task. This will allow us to achieve a client-server scenario where the client effectively chooses the measurement basis by this pre-rotation while hiding his secret input bits (a and b). The next step for obtaining the full security property is the application of an additional random Z gates to hide the outcome computed by the server.

These hiding steps leads to the necessity of quantum communication (as we prove next) and as a result we can replace the requirement of GHZ state with sequential rotation and one final measurement on a single qubit state as well. In other words, if we denote the $\pi/2$ rotation along the Z axis by S then we prove that the local operators of the form

$$S^{\dagger^{a\oplus b}} S^b S^a |+\rangle \tag{11.1}$$

encode the input of the client in the resource state, $|+\rangle$, while permitting the server to perform the other operations required to compute the NAND gate. Here, for any unitary U and bit x, with U^x we denote the identity $\mathbb{1}$ (for $x = 0$) and U (for $x = 1$). Since all the information of the client is encoded in the phase of the states, additional randomly chosen Z gates achieve a full one-time pad of the client's information, which can easily be decoded by the client by a bit-flip (for details see Section 12.1.4). Here we present specific protocols based on various manipulations of the single qubit $|+\rangle$ and three qubits entangled GHZ state, however one could easily adapt these protocols to cover various encodings necessary for the specific noise model or available resources within a particular implementation platform as we have recently done for an experimental demonstration of our protocol in an optical setting [Barz et al., 2015].

The actual setting of our protocols (a restricted client with XOR gate only), on its own, is not a realistic set up. The potential advantage of the employment of such quantum scheme in a classical secure multi-party protocol for reducing the overall overhead should be explored elsewhere. However, one could think of our protocols as a game scenario that exhibits the power of quantum communication. It is straightforward to prove that purely classical players (i.e. a classical protocol with classical client and server) could not win deterministically the game of computing securely the NAND of encoded input bits (Theorem 20). The proof is based on a reduction to the impossibility of computing non-linear function (e.g NAND) with linear function (e.g Parity) given a generic advice string*. On the other hand to prove having pre-shared quantum correlation or equivalently using offline quantum communication (quantum states independent of classical inputs) will also not lead to a winning strategy, a completely different proof technique had to to be developed (Theorems 21 and 22). Through a series of lemmas we show any quantum offline protocol for secure delegated NAND computing could be reduced to a simple protocol with one round only where the classical communication will be necessarily dependent on the client's secret inputs. Then the correctness criteria for a deterministic secure NAND computation is proven

*By a generic advice string we mean any string which does not depend on the input bits.

to be equivalent to perfect discrimination of the classical encoded messages and hence the leakage of client's secret. To the best of our knowledge this is the first time that a security game manifests the structural difference between static pre-shared correlation versus a dynamic quantum communication.[†]

[†]While obviously quantum communication could be achieved using shared entanglement and classical communication, however the restricted client in our set up could not employ teleportation. This is in fact chosen so that to highlight the differences.

Chapter 12

Secure-NAND

12.1 Secure NAND Protocols

There are three types of protocols that we introduce here, to address various implementation scenarios. These families achieve the same goal and differ only in the required quantum gadgets of the client. Following the construction steps explained in the introduction, in the first family of the protocols, it is assumed that client can create or have secure access to some simple (few qubits) entangled states. On the other hand, in the second family it is assumed that the client is able to measure the flying qubit that it receives through an untrusted channel to perform its desired universal computation. In the third setting, the client needs only to have the capacity to perform simple single qubit rotations. Importantly, in all three scenarios the classical computation of the client is restricted to XOR operations. For all of our SecureNAND protocols will always assume that the client is honest, and behaves as specified by the protocols. In our setting, the server is the untrusted party. However, we will require and that the protocols are correct - yield correct outputs in the case of an honest server. This property corresponds to completeness in the terminology of interactive proof systems, whereas we do not require soundness (verifiability, in the context of delegated computing).

Aside from correctness, regarding the guarantees for the honest client, we require the protocols are secure, specifically that they reveal no information about the client's input (aside from the input's size) to the server. This property, also known as blindness, is defined formally as follows.

Definition 19. *We will say any SecureNAND protocol is secure (also referred as* blind*) if the quantum states sent during the execution of the protocol from the client to the server, once averaged over the client's internal secret parameters, do not depend on the*

secret input bits to the client i.e. the inputs to the NAND gate which is to be computed in a secure delegated fashion. In other words, the averaged states sent by the client are the same for any choice of the inputs.

In other words, the system the server receives from the client could have been generated by the server without receiving any information from the client.

We first prove that it is impossible to achieve the similar task of secure delegated computing of our protocols by removing the quantum requirement.

Theorem 20. *No classical protocol, in which the client is restricted to XOR computations can delegate deterministically computation of NAND to a server while keeping the blindness.*

Proof. We prove this result first for the case of two rounds of communication, and no initial shared randomness. Any such protocol will have the following three stages: client's encoding, server's computation, and client's decoding.

client's encoding. In this stage, the only thing the client can do is to compute $C_1(a,b,\overrightarrow{x})$, where a,b are the input bits, $\overrightarrow{x}$ is a random bit string (of any length) and C_1 is a computation which can be implemented using only XOR gates. However, the state $C_1(a,b,\overrightarrow{x})$ must be independent from a and b to maintain blindness when averaged over all $\overrightarrow{x}$.

server's computation. The only thing the server can do is to apply some computable function S on $C_1(a,b,\overrightarrow{x})$, thus returning $S(C_1(a,b,\overrightarrow{x}))$.

client's decoding. The only thing the client can do is to run some function C_2, on all the data he has, which is implementable using XOR gates only:

$$C_2(a,b,\overrightarrow{x},S(C_1(a,b,\overrightarrow{x}))) = NAND(a,b) \text{ (correctness)}$$

and the output must (deterministically) be the NAND of the inputs.

Let $c = C_1(a',b',\overrightarrow{x'})$ be some constant the client may send to the server. Then, because of blindness it must hold that for all a,b there must exist $\overrightarrow{x}(a,b)$, which depends on a,b such that

$$C_1(a,b,\overrightarrow{x}(a,b)) = c.$$

To see this, note that if the client could send c, but not for some inputs a'' and b'', then upon receiving c the server learns something about the input, namely that it is not a'',b'', which violates blindness. Note also that since all the computations the client

can perform use only XOR gates (and without the loss of generality, reversible), the client *can* compute $\overrightarrow{x}(a,b)$ given a,b using only XOR operations. But then, by the correctness of the protocol we have that

$$C_2(a,b,\overrightarrow{x}(a,b),S(c)) = NAND(a,b) \text{ (correctness)}.$$

But $S(c)$ is constant as well. This implies that given a fixed string $S(c)$ the client can compute the NAND of any input using just XOR gates, which is not possible.

This argument can be further generalized to a setting with shared randomness and many rounds of communication. It is easy to see that the randomness cannot help as the protocol must be deterministic (hence work for any sampling of the joint random variable), whereas using multiple rounds (all of which must be independent of the input, from the viewpoint of the server) just yields a longer constant string (analogous to $S(c)$) using which the client can compute the *NAND* on her own, which is again impossible. □

We proceed now with constructing the family of quantum protocols for the same task. While the simplest protocol is demonstrated last, we present the sequence of adapting the GHZ game as shown in [Anders and Browne, 2009] to a security scenario to present a simple security and correctness proof. Then with each new family we reduce the client's requirement while maintaining the same properties.

12.1.1 Preparing Client

In Protocol 12 the client generates a GHZ state of 3 qubits which are rotated depending on the values of the inputs a,b, $a \oplus b$ and a random bit r. Qubits are sent through an untrusted quantum channel from client to server who applies a Pauli-X measurement on the qubits and sends the classical result to the client via an untrusted classical channel. The client produces the final output by applying classical XOR gates between the received classical bits and the random bit. In what follows we denote a random selection of an element of a set by $\in_R$.

We will say any SecureNAND protocol is *correct* if for every run of the protocol where both players are honest (adhere to the protocol) and for all inputs a,b we have

$$out = \neg(a \wedge b) = 1 \oplus (ab)$$

This definition will be used for all the presented protocols in this chapter. Throughout this chapter we will be using the notation for the logical *and* between two bits a,b as $a \wedge b$ and ab interchangeably.

Protocol 12 Entangled-based Preparing client SecureNAND

- Input (to client): two bits a, b
- Output (from client): $\neg(a \wedge b)$
- The Protocol:
  - client's round
    1. $r \in_{\mathsf{R}} \{0,1\}$
    2. client generates

$$|\Psi'\rangle = Z_1^r \left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a \oplus b} |\Psi\rangle$$

       and sends it to the server.
  - server's round
    1. server measures the qubits 1,2 and 3, with respect to the observables X_1, X_2, and X_3, obtaining outcomes b_1, b_2 and b_3, respectively.
    2. server sends b_1, b_2, b_3 to client
  - client's round
    1. client computes

$$out = b_1 \oplus b_2 \oplus b_3 \oplus r \tag{12.1}$$

    2. client outputs out.

Lemma 13. *Protocol 12 is correct.*

Proof. First note that the protocol is correct if the following eigenstate equality is true for all binary variables a, b, r

$$X_1 X_2 X_3 |\Psi'\rangle = (-1)^{1 \oplus ab \oplus r} |\Psi'\rangle \tag{12.2}$$

as this equality guarantees that the parity of the outcomes of stabiliser measurements of server (in basis $X_1X_2X_3$) equals $1 \oplus ab \oplus r$ which implies client will decode the correct outcome in Equation 12.1 of Protocol 12. For simplicity define the following notion for Pauli observable

$$P^i = \begin{cases} X & \text{if } i{=}0 \\ Y & \text{if } i{=}1 \end{cases}$$

we then have the following commutation relations $\forall\, b, r \in \{0,1\}$

$$\begin{aligned} P^b Z^r &= (-1)^r Z^r P^b \\ P^b S^r &= (-1)^{(b\oplus 1)r} S^r P^{b\oplus r} \\ P^b \left(S^\dagger\right)^r &= (-1)^{br} \left(S^\dagger\right)^r P^{b\oplus r} \end{aligned}$$

and in particular

$$X \left(S^\dagger\right)^r = \left(S^\dagger\right)^r P^r$$

and hence as stated in [Anders and Browne, 2009] we obtain that $\forall\, a, b \in \{0,1\}$

$$P_1^a P_2^b P_3^{a\oplus b} |\Psi\rangle = (-1)^{(1\oplus ab)} |\Psi\rangle$$

We proceed to show that Equation (12.2) holds:

$$\begin{aligned} X_1X_2X_3|\Psi'\rangle &= X_1X_2X_3 Z_1^r \left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b} |\Psi\rangle \\ &= \left[X_1 Z_1^r \left(S_1^\dagger\right)^a\right]_1 \left[X_2 \left(S_2^\dagger\right)^b\right]_2 \left[X_3 \left(S_3^\dagger\right)^{a\oplus b}\right]_3 |\Psi\rangle \\ &= (-1)^r \left[Z_1^r \left(S_1^\dagger\right)^a P_1^a\right]_1 \left[\left(S_2^\dagger\right)^b P_2^b\right]_2 \left[\left(S_3^\dagger\right)^{a\oplus b} P_3^{a\oplus b}\right]_3 |\Psi\rangle \\ &= (-1)^r Z_1^r \left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b} P_1^a P_2^b P_3^{a\oplus b} |\Psi\rangle \\ &= (-1)^{1\oplus ab\oplus r} Z_1^r \left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b} |\Psi\rangle = (-1)^{1\oplus ab\oplus r} |\Psi'\rangle \end{aligned}$$

In the derivation above we have simply used the trivial commutativity of operators acting on disjoint subsystems. So the Lemma holds. □

Next, we prove the blindness of this protocol

In the remainder of this chapter we will use the following short-hand notation:

$$\boxed{X} := |X\rangle\langle X|$$

for all labels X. This is a non-standard notation used here for the sake of brevity.

Lemma 14. *Protocol 12 is blind.*

Proof. For fixed input bits a and b the state the server receives from the client can be written as

$$\rho_S = \sum_r \frac{1}{2} Z_1^r \eta Z_1^r \tag{12.3}$$

where

$$\eta = \left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b} |\Psi\rangle\langle\Psi| (S_1)^a (S_2)^b (S_3)^{a\oplus b}$$

Note that η can be written as $\mathbf{S}|\Psi\rangle\langle\Psi|\mathbf{S}^\dagger$, where

$$\mathbf{S} = \left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b}$$

The operator $\mathbf{S}$ does not depend on the r_i variables, and is diagonal in the computational basis so it commutes with Pauli Z operators and hence we have

$$\rho_S = \mathbf{S}\left(\sum_r \frac{1}{2} Z_1^r |\Psi\rangle\langle\Psi| Z_1^r\right)\mathbf{S}^\dagger$$

The operator $\sum_r \frac{1}{2} Z_1^r |\Psi\rangle\langle\Psi| Z_1^r$ can explicitly be written as

$$\frac{1}{2}\left(\frac{1}{2}\left(|001\rangle\langle 001| + |110\rangle\langle 110| - |001\rangle\langle 110| - |110\rangle\langle 001|\right) + \frac{1}{2}\left(|001\rangle\langle 001| + |110\rangle\langle 110| + |001\rangle\langle 110| + |110\rangle\langle 001|\right)\right)$$

which in our notation is equal to $\frac{1}{2}\left(\boxed{001} + \boxed{110}\right)$. Hence the operator is diagonal in the computational basis, and again commutes with $\mathbf{S}$ so we get:

$$\rho_s = \left(\sum_r \frac{1}{2} Z_1^r |\Psi\rangle\langle\Psi| Z_1^r\right)\mathbf{S}\mathbf{S}^\dagger = \frac{1}{2}\left(\boxed{001} + \boxed{110}\right)$$

This state is independent from a and b and the lemma is proved. □

Protocol 13 Entangled-based Measuring client SecureNAND

- Input (to client): two bits a, b
- Output (from client): $\neg(a \wedge b)$
- The Protocol:
  - server's round
    1. The server prepares the state $|\Psi\rangle$ and sends it to the client
  - client's round
    1. The client computes $c = a \oplus b$, measures the qubits 1,2 and 3, with respect to the observables P^a, P^b, and P^c, obtaining outcomes b_1, b_2 and b_3, respectively.
    2. client computes

$$out = b_1 \oplus b_2 \oplus b_3 \tag{12.4}$$

    3. client outputs *out*.

12.1.2 Measuring Client

In Protocol 13 server generates a GHZ state of 3 qubits. The qubits are sent through an untrusted quantum channel from the server to the client. The client applies a Pauli-X or Pauli-Y measurement on the qubits depending on the classical inputs a and b and their classical XOR and produces the final output by applying classical XOR gates between the measurement outputs.

Lemma 15. *Protocol 13 is blind and correct.*

Proof. The correctness of this protocol follows directly from the result in [Anders and Browne, 2009]. The blindness of the protocol trivially follows from the fact that no information is sent from the client to the server, thus the protocol is blind in all no-signalling theories (including standard Quantum Mechanics). □

12.1.3 Bounce Protocol

In Protocol 14 we reduce the requirements on the client side, which no longer has to measure or prepare states, but rather only modify locally the GHZ state of 3 qubits

prepared by the server. The client applies single-qubit quantum operators depending on the values of the inputs a, b, $a \oplus b$ and 3 classical random bits. The client sends the rotated qubits to the server via an untrusted quantum channel. The server applies a Pauli-X measurement on the qubits and sends the classical result to the client via an untrusted classical channel. The client produces the final output by applying classical XOR gates between the received classical bits and the random bits.

Protocol 14 Entangled-based Bounce SecureNAND

- Input (to client): two bits a, b
- Output (from client): $\neg(a \wedge b)$
- The Protocol:
  - server's round
    1. The server prepares the state $|\Psi\rangle$ and sends it to the client
  - client's round
    1. client receives the state $|\Psi\rangle$ from the server.
    2. client generates $r_1, r_2, r_3 \in_{\mathsf{R}} \{0,1\}$
    3. client modifies the state $|\Psi\rangle$ to $|\Psi'\rangle$ as follows

$$|\Psi'\rangle = Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b} |\Psi\rangle$$

       and sends it to the server.
  - server's round
    1. server measures the qubits 1,2 and 3, with respect to the observables X_1, X_2, and X_3, obtaining outcomes b_1, b_2 and b_3, respectively.
    2. server sends b_1, b_2, b_3 to client
  - client's round
    1. client computes

$$out = b_1 \oplus b_2 \oplus b_3 \oplus r_1 \oplus r_2 \oplus r_3. \tag{12.5}$$

    2. client outputs out.

Lemma 16. *Protocol 14 is correct.*

Proof. The correctness is directly obtained from Lemma 13 on the correctness of Protocol 12. To see this note that the states the server performs the measurements on are identical in the two protocols, up to the existence of possible $Z_2^{r_2}$ and $Z_3^{r_3}$ rotations on the second and third qubit. Since we have

$$XZ^r = (-1)^r Z^r X, \text{and}$$
$$YZ^r = (-1)^r Z^r Y,$$

these rotations cause an additional (multiplicative) phase of $(-1)^{r_2 \oplus r_3}$. But this is compensated for in the modified decoding of the client (see Equation 12.5 in Protocol 14) so the output is correct in this protocol as well. □

Lemma 17. *Protocol 14 is blind.*

Proof. For fixed input a and b the final state server obtains in the protocol can be written as

$$\rho_S^{fin} = \sum_{r_1,r_2,r_3} \frac{1}{8} \left(Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \eta \left(Z_1^{r_1} Z_3^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right)$$

with

$$\eta = \left(\left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b} \otimes \mathbb{1}_S\right) \rho_S^{init} \left((S_1)^a (S_2)^b (S_3)^{a\oplus b} \otimes \mathbb{1}_S\right)$$

where ρ_S^{init} is any state the malevolent server could have initially prepared. In the expression above, we have made no assumptions on the dimensionality of the initial state the server may have prepared, and we only assume that the local operations of the client are correct, single qubit operations, acting on three distinct qubits.

Note further that the actions of the client are only on a subsystem of the whole system in the state ρ_S^{init}, signifying that the server might have prepared an entangled state, and sent only a subsystem to the client to be modified, while keeping the remainder of the system. To simplify the state we could commute the Z operators with the phase $S^\dagger$ operators since the parameters of the phase operators do not depend on r_i values. Introducing the shorthand

$$\mathbf{S} = \left(\left(S_1^\dagger\right)^a \left(S_2^\dagger\right)^b \left(S_3^\dagger\right)^{a\oplus b} \otimes \mathbb{1}_S\right)$$

we can rewrite the state of the server's system as

$$\rho_S^{fin} = (\mathbf{S} \otimes \mathbb{1}_S) \sum_{r_1,r_2,r_3} \frac{1}{8} \left(Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \rho_S^{init} \left(Z_1^{r_1} Z_3^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \left(\mathbf{S}^\dagger \otimes \mathbb{1}_S\right)$$

The state ρ_S^{init} has two partitions - the partition corresponding to the subsystem the server sends to the client, and the subsystem he keeps. Thus ρ_S^{init} can be written (in the Pauli operator basis) as

$$\sum_{i,j} \alpha_{i,j} \underbrace{\sigma_i}_{C} \otimes \underbrace{\sigma_j}_{S'}$$

where C denotes the subsystem sent to the client, and S' the subsystem kept by the server, and σ_i and σ_j denote general Pauli operators acting on the two respective subsystems. Next, we have the following derivation:

$$\begin{aligned}
& \sum_{r_1,r_2,r_3} \frac{1}{8} \left(Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \rho_S^{init} \left(Z_1^{r_1} Z_3^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \\
= & \sum_{r_1,r_2,r_3} \frac{1}{8} \left(Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \sum_{i,j} \alpha_{i,j} \underbrace{\sigma_i}_{C} \otimes \underbrace{\sigma_j}_{S'} \left(Z_1^{r_1} Z_3^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \\
= & \frac{1}{8} \sum_{i,j} \alpha_{i,j} \left(\sum_{r_1,r_2,r_3} Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \sigma_i Z_1^{r_1} Z_3^{r_2} Z_3^{r_3}\right) \otimes \sigma_j
\end{aligned}$$

Note that since both X and Y anticommute with Z, the expression

$$\sum_{r_1,r_2,r_3} Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \sigma_i Z_1^{r_1} Z_3^{r_2} Z_3^{r_3}$$

is non-zero only if all the single qubit operators making up σ_i are either Z or identity, and in both cases diagonal in the computational basis. Thus, we can write the final expression of the derivation above as

$$\sum_{i,j} \alpha'_{i,j} \sigma'_i \otimes \sigma_j$$

where σ'_i is diagonal in the computational basis. So, overall, for the state of the server's system we have

$$\begin{aligned}
& (\mathbf{S} \otimes \mathbb{1}_S) \sum_{r_1,r_2,r_3} \frac{1}{8} \left(Z_1^{r_1} Z_2^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \rho_S^{init} \left(Z_1^{r_1} Z_3^{r_2} Z_3^{r_3} \otimes \mathbb{1}_S\right) \left(\mathbf{S}^\dagger \otimes \mathbb{1}_S\right) \\
= & (\mathbf{S} \otimes \mathbb{1}_S) \sum_{i,j} \alpha'_{i,j} \sigma'_i \otimes \sigma_j \left(\mathbf{S}^\dagger \otimes \mathbb{1}_S\right) \\
= & \sum_{i,j} \alpha'_{i,j} (\mathbf{S} \otimes \mathbb{1}_S) \sigma'_i \otimes \sigma_j \left(\mathbf{S}^\dagger \otimes \mathbb{1}_S\right)
\end{aligned}$$

and since σ'_i commute with $\mathbf{S}$ we get:

$$\sum_{i,j} \alpha'_{i,j} \sigma'_i \mathbf{S}\mathbf{S}^\dagger \otimes \sigma_j = \sum_{i,j} \alpha'_{i,j} \sigma'_i \otimes \sigma_j$$

Since $\alpha'_{i,j}$ is independent from a and b, this state is independent from a and b and the lemma is proved. □

Protocol 15 Single Qubit Bounce SecureNAND

- Input (to client): two bits a, b
- Output (from client): $\neg(a \wedge b)$
- The Protocol:
  - server's round
    1. The server prepares the state $|+\rangle$ and sends it to the client
  - client's round
    1. client receives the state $|+\rangle$ from the server.
    2. client generates $r \in_{\mathsf{R}} \{0,1\}$
    3. client modifies the state $|+\rangle$ to $|\Psi\rangle$ as follows

$$|\Psi\rangle = Z^r S^a S^b \left(S^\dagger\right)^{a \oplus b} |+\rangle$$

       and sends it to the server.
  - server's round
    1. The server measures the qubit with respect to the X basis, obtaining the outcome s
    2. server sends s to client
  - client's round
    1. client computes

$$out = s \oplus r \oplus 1 \tag{12.6}$$

    2. client outputs out.

12.1.4 Single Qubit Protocols

Here, we give variants of a new class of secure NAND protocols which only require single qubit manipulations. Similarly to the variants we have given for the GHZ-based protocols, the single qubit protocol can also be modified to a client preparation or a measuring client protocol. In the former, it is the client which would prepare the initial $|+\rangle$ state, whereas in the measuring client protocol, the client would perform the final

measurements. Similar to the entangled-based scenario, all variations of protocols are blind and correct as a simple consequence of the Single Qubit Bounce SecureNAND protocol (that we describe next).

In Protocol 15, the server generates a single qubit state ($|+\rangle$) and sends it via an untrusted quantum channel to the client who applies a series of single qubit rotation operator depending on the values of the inputs a, b, $a \oplus b$, and a classical random bit. The client sends the rotated qubit to the server via untrusted quantum channel. Sever applies a Pauli-X measurement on the qubit and sends the classical result to the client via an untrusted classical channel. The client produces the final output by applying classical XOR gates between the received classical bit and the random bit and constant bit 1. The correctness and blindness are directly obtained from the proof for the entangled-based protocols. To see the correctness note that if the server was honest, it is a straightforward calculation to see the state of the qubit the server receives is

$$Z^r Z^{a \wedge b} |+\rangle$$

Then the result of the measurement performed by the server is $s = r \oplus a \wedge b$, and the decoding produces $out = 1 \oplus a \wedge b$ as required. To see the security, note that the most general strategy of the server is to prepare a bipartite state $\pi_{1,2}$ and send the first subsystem to the client. Then the state of the server system (up to a normalisation factor), once the client performed his round is

$$\sum_r \left(Z^r Z^{a \wedge b} \otimes \mathbb{1}_2\right) \pi_{1,2} \left(Z^r Z^{a \wedge b} \otimes \mathbb{1}_2\right) = \sum_{r'} \left(Z^{r'} \otimes \mathbb{1}_2\right) \pi_{1,2} \left(Z^{r'} \otimes \mathbb{1}_2\right)$$

where $r' = r \oplus a \wedge b$. Since r is distributed uniformly at random, so is r' so the state above does not depend on a or b.

12.2 No-go Result

The main contribution of this chapter is to prove the optimality of the quantum protocols of the last section. We prove that it is impossible to achieve the similar task of the secure delegated NAND computing if one attempts to remove any quantum communication. Next we show that the communicated quantum states must also depend on the classical input of the client as it is done in our protocols. More precisely, we will show that any quantum protocol where a XOR-restricted client computes $NAND(a,b)$, by initially sending a quantum state ρ to the server, followed by classical communication only, can be perfectly blind and perfectly correct only if the state ρ depends on the input bits

a, b of the client. The protocols without this dependence, so where all the quantum communication can be done independently from the input of the client (hence can be done before the client decides on her input bits), we call quantum off-line protocols. Thus, we show that a blind quantum-offline protocol with a XOR-restricted client is not possible *. We begin by addressing protocols with two rounds of communication between the client and the server. By round we refer to an instance of either the client sending a message to the server, or the server sending a message to the client. Since the last message, for it to have any meaning, must come from the server, the order of the two rounds is client $\rightarrow$ server, followed by server $\rightarrow$ client. The generic description of a potential secure NAND quantum offline protocol with two rounds is given later in Protocol 7. In order to prove the impossibility of obtaining such a protocol we prove several lemmas proving first the impossibility of a particular class of somehow 'minimal' NAND quantum offline protocols (see Protocol 16 and 17 below). Following this, we present the reduction between these protocols i.e. if a generic protocol of type Protocol 18 is possible then so is the minimal protocol, hence proving the impossibility of obtaining any offline quantum protocol. Finally, we extend our argument to the multi-rounds scenario.

These types of protocols are intimately linked to the composability of secure NAND computations in a larger computation †. Note that since, for the second layer of any computation, the client does not know the inputs in advance (since she cannot compute them herself) but knows the encryption of the outputs in advance, thus, quantum offline protocols are necessary and probably sufficient for the composition of NANDs in a larger computation, without requiring additional run-time the multi-round scenarios communication. The case where run-time communication is allowed will be studied presently. Note also that it does not matter what function, which in tandem with XOR and NOT gates forms a universal set, we use. For simplicity, here we focus on AND. To shorten our expressions, in this section we will be predominantly use ab to denote $a \wedge b$ the logical AND operator of two bits a and b.

The simple quantum offline secure AND computation with two rounds of communication (Simple AND QO2, Protocol 16) is the most natural first attempt, which is inspired by information-theoretic considerations - since the client's input is two bits a

*Note, in contrast, that the blind quantum computing protocol in [Broadbent et al., 2009] is quantum-offline, as the initial qubits the client sends are chosen uniformly at random. However, in this protocol, the computing power of the client is beyond just XOR gates.

† We do not explicitly address the security issues of composability of our protocols. However, note that our obtained lower bounds on what is possible implies also that the impossibility results will also hold true in any composable security setting.

and b, hence the quantum state encodes two bits of x and y. Therefore, to hide the two bits in the quantum state, additional randomness of two bits r_1 and r_2 is needed.

Protocol 16 Simple SecureAND QO2

The functionality of the Small AND protocol:

- Input (to the client): two bits a, b
- Output (from the client): $(a \wedge b)$
- The Protocol:
  - client's round
    1. client generates a quantum state $\rho^{x,y}_{r_1,r_2}$, characterized by random bits x, y, r_1, r_2 and sends it to the server.
    2. client receives her input bits a, b.
    3. client computes $m_c = (x \oplus a, y \oplus b)$ and sends it to the server.
  - Cerver's round
    1. server performs a (generalized) measurement of $\rho^{x,y}_{r_1,r_2}$, parametrized by m_c. He obtains the outcome m_s and sends it to the client.
  - client's round
    1. client computes $out = m_s \oplus r_1 \oplus r_2$.
    2. client outputs out.

Recall that the correctness of these protocols are defined by requesting $out = ab$, and blindness is defined by the equation

$$\Sigma_{\mathbf{x}} \boxed{m(a,b)} \otimes \rho^{\mathbf{x}} = \eta \qquad \forall a, b,$$

where a, b are the input bits, $m(a,b)$ the classical message which may depend on the input, ρ^x a quantum state which depends on some random parameters x (but may also depend on a, b), and η is a positive-semidefinite operator, independent from a, b. For simplicity, we are omitting any normalisation factors, so η may be of non-unit trace.

Lemma 18. *No Simple SecureAND QO2 can be correct and blind.*

Proof. As in any Simple SecureAND QO2 protocol the client sends two classical bits of information to the server (here denoted a', b'), without the loss of generality, we may

assume that the message the server returns to the client is a single bit measurement outcome of one of four (generalised) measurements (one for each message (a',b')) which we denote $M^{a',b'}(\rho^{x,y}_{r_1,r_2})$. The correctness of the protocol entails that

$$M^{a',b'}(\rho^{x,y}_{r_1,r_2}) = (a' \oplus x)(b' \oplus y) \oplus r_1 \oplus r_2$$

For clarity we briefly comment on the equation above. Since, for message (a',b') the server performs a generalised two-outcome measurement, this measurement can be represented by the POVM elements $\Pi_0^{a',b'}, \Pi_1^{a',b'}$ (which are positive operators summing to the identity), corresponding to outcomes 0 and 1, respectively. Then the equation above means that

$$Tr(\Pi^{a',b'}_{(a'\oplus x)(b'\oplus y)\oplus r_1\oplus r_2}\rho^{x,y}_{r_1,r_2}) = 1$$

Then, by taking $r = r_1 \oplus r_2$ and defining $\rho^{x,y}_r = 1/2(\rho^{x,y}_{0,r} + \rho^{x,y}_{1,1\oplus r})$ we get, by linearity, that

$$M^{a',b'}(\rho^{x,y}_r) = (a' \oplus x)(b' \oplus y) \oplus r,$$

or equivalently,

$$Tr(\Pi^{a',b'}_{(a'\oplus x)(b'\oplus y)\oplus r}\rho^{x,y}_r) = 1$$

and also that

$$Tr(\Pi^{a',b'}_{(a'\oplus x)(b'\oplus y)\oplus r}\rho^{x,y}_{r\oplus 1}) = 0$$

The two equations above immediately entail that $\rho^{x,y}_r$ and $\rho^{x,y}_{r\oplus 1}$ must be (mixtures of mutually) orthogonal states, which we denote as

$$\rho^{x,y}_r \perp \rho^{x,y}_{r\oplus 1}$$

But, more generally, the equations above imply that two states $\rho^{x,y}_r$ and $\rho^{x',y'}_{r'}$ must be in orthogonal subspaces, whenever any of the sub/superscripts differ. To see this, we will consider the remaining cases separately. First, assume that $r = r'$, but $x \neq x'$ and/or $y \neq y'$. Then if we set $a' = x \oplus 1$ and $b' = y \oplus 1$ we see that

$$M^{a',b'}(\rho^{x,y}_r) = (a' \oplus x)(b' \oplus y) \oplus r = 1 \oplus r$$

but

$$M^{a',b'}(\rho^{x',y'}_r) = (a' \oplus x')(b' \oplus y') \oplus r = r$$

so the outcomes *deterministically* differ, meaning that the two states must be in orthogonal subspaces. We have already seen that the same conclusion follows if $r \neq r'$, and

$x = x'$ and $y = y'$. The next case is when $r \neq r'$, and either $x \neq x'$ or $y \neq y'$ (but one is an equality). Assume that $x = x'$, $y \neq y'$ and $r = 0$. Then if we set $a' = x = x'$ we see that

$$M^{x,b'}(\rho_0^{x,y}) = (x \oplus x)(b' \oplus y) = 0$$

and

$$M^{x,b'}(\rho_1^{x,y'}) = (x \oplus x')(b' \oplus y') \oplus 1 = 1$$

Similarly, if $r = 1$ we get opposite results, and if $x \neq x'$ and $y = y'$ we get the same by setting $b' = y = y'$. Finally, we must consider the case when all the parameters differ. First, assume $r = 0$, then by setting $a' = x$ and $b' = 1 \oplus y$ we get

$$M^{a',b'}(\rho_0^{x,y}) = (x \oplus x)(b' \oplus y) = 0$$
$$M^{a',b'}(\rho_1^{x',y'}) = (x \oplus x')(1 \oplus y \oplus y') \oplus 1 = 1$$

since $y \neq y'$, if $r = 1$ then the first equation above would yield 1, and the last would yield 0, since $1 \oplus y \oplus y' = 0$. Thus we can conclude that the states $\{\rho_r^{x,y}\}_{x,y,r}$ are all in orthogonal subspaces. But this means, in particular, that the states $1/4(\sum_{r_1,r_2} \rho_{r_1,r_2}^{x,y})$ are in orthogonal subspaces for all x, y which implies that there exists a measurement which perfectly reveals x and y given any $\rho_{r_1,r_2}^{x,y}$. Thus, the server can perfectly learn x and y and, given the classical message of the client, the inputs of the client, and the protocol is not blind. □

In the above proof we have quickly concluded that the two bits r_1, r_2 are superfluous and one will suffice (which is intuitive as only one random bit is needed to one-time pad the one bit outcome). This gives us the definition of the next general family of protocols (Small AND QO2, Protocol 17) as we describe below and will refer to later.

Lemma 19. *No small SecureAND QO2 can be correct and blind.*

Proof. Obvious from the proof of impossibility of simple AND QO2, where we have actually reduced simple to small protocols. □

12.2.1 Generalisation: QO2

In order to prove a reduction between the general case of Protocol 18 and the simple scenario of Protocol 17 we start with a supposedly given blind and correct QO2 protocol and iteratively construct a blind correct small QO2, using a sequence of claims which define increasingly simpler protocols.

Protocol 17 Small SecureAND QO2

The functionality of the Small AND protocol:

- Input (to the client): two bits a, b
- Output (from the client): $(a \wedge b)$
- The Protocol:
  - client's round
    1. client generates a quantum state $\rho_r^{x,y}$, characterized by random bits x, y, r and sends it to the server.
    2. client receives her input bits a, b.
    3. client computes $m_c = (x \oplus a, y \oplus b)$ and sends it to the server.
  - server's round
    1. server performs a (generalized) measurement of $\rho_r^{x,y}$, parametrized by m_c. He obtains the outcome m_s and sends it to the client
  - client's round
    1. client computes $out = m_s \oplus r$.
    2. client outputs out.

Theorem 21. *If there exists a blind, correct SecureAND QO2 then there exists a blind correct* Small *SecureAND QO2.*

The objects which appear in the protocol (which differ from the objects in the small QO2) are as follows:

$\rho^{\mathbf{x}}$, with $\mathbf{x} = (x_1, \ldots, x_n)$ – the quantum state parametrized by n bits
$m_c = \mathrm{XOR}_E(a, b, \mathbf{x})$ – the m bit message from the client
m_s, the k bit message from the server
$ab = out = \mathrm{XOR}_D(a, b, \mathbf{x}, m_s)$ – the calculation of the output

Protocol 18 SecureAND QO2

The functionality of the AND protocol:

- Input (to the client): two bits a, b
- Output (from the client): $(a \wedge b)$
- The Protocol:
  - client's round
    1. client generates a quantum state $\rho^{\mathbf{x}}$, characterised by a sequence of random parameters $\mathbf{x} = (x_1, \ldots, x_n)$, and sends it to the server.
    2. client receives her input bits a, b (the client could have had her bits all along. It is however the defining property of quantum-offline protocols that the parameters $\mathbf{x}$ are independent from a, b).
    3. client computes an XOR-computable function

       $$m_c = \mathrm{XOR}_E(a, b, \mathbf{x})$$

       (E for encryption) of the input and the random parameters. Note that it would be superfluous for the client to generate additional random values at this stage - they could be part of $\mathbf{x}$, without influencing the state the client generates.
    4. client sends m_c to the server.
  - server's round
    1. server performs a (generalized) measurement of $\rho^{\mathbf{x}}$, parametrized by m_c. He obtains the outcome m_s and sends it to the client.
  - client's round
    1. client computes an XOR-computable function

       $$out = \mathrm{XOR}_D(a, b, \mathbf{x}, m_s)$$

       (D for decryption).
    2. client outputs out.

Lemma 20. *Nothing is gained from using multi-bit m_s.*

Proof. Note that since the client is restricted to computing XOR operations, we can dissect

$$\mathrm{XOR}_D(a,b,\mathbf{x},m_s)$$

and see that it must be of the form

$$\mathrm{XOR}_D(a,b,\mathbf{x},m_s) = \mathrm{XOR}'_D(a,b,\mathbf{x}) \oplus \bigoplus_{j\in I\subseteq[k]} [m_s]_j,$$

where $[m_s]_j$ is the j^{th} bit of the k-bit message m_s. That is, it is a mod 2 addition of something which does not depend on the server's message, and the mod 2 addition of some of the bits of the message responded by the server. Since the form of the message (*i.e* the explicitly description of the function XOR_D) is public, being in the protocol description, the protocol remains secure and correct if the server himself computes the bit $\bigoplus_{j\in I\subseteq[k]} [m_s]_j$, and returns this to the client. Thus, for every correct, blind QO2 there exists a correct blind $\mathrm{QO2}_1$ where the server's message comprises only one bit. The remainder of the claims assumes we are dealing with a $\mathrm{QO2}_1$ protocol. □

Lemma 21. *No random parameters which do not appear in the encryption or decryption are needed.*

Proof. Let $S \subset [n]$ be a subset of indices of the random parameters which appear in either encryption (as variables of XOR_E) or decryption (XOR_D), and let $S' = [n] \setminus S$ be the subset which does not appear. Then, by exchanging the state $\rho^{\mathbf{x}}$ with the state

$$(\rho')^{\mathbf{x}'} = \sum_{x_j | j\in S'} \frac{1}{2^{|S'|}} \rho^{\mathbf{x}}$$

in a $\mathrm{QO2}_1$ protocol it is easy to see we again obtain a protocol (which we refer to as $\mathrm{QO2}_2$) which is correct and blind. In $\mathrm{QO2}_2$ protocols, all the random parameters appear either in the decryption or encryption. The remainder of the claims assumes we are dealing with a $\mathrm{QO2}_2$ protocol. □

Lemma 22. *No more than one random parameter which appears only in the decryption is needed.*

Proof. Let $S_{D\setminus E} \subset [n]$ be the set of indices of random parameters which appear only in the decryption, that is, as a variable of the function XOR_D. Without the loss of generality, we will assume that the last k indices are such. Then $\mathrm{XOR}_D(a,b,\mathbf{x},m_s)$ (due to the restrictions on the client) can be written as:

$$\mathrm{XOR}_D(a,b,\mathbf{x},m_s) = \mathrm{XOR}'_D(a,b,m_s,x_1\ldots,x_{n-k}) \oplus x_{n-k+1} \oplus \cdots \oplus x_n,$$

Then, by exchanging the state $\rho^{\mathbf{x}}$ with the state

$$(\rho')^{x_1,\ldots,x_{N-k},x} = \sum_{\substack{x_j | j \in S_{D\setminus E} \\ s.t. \\ \oplus_j x_j = x}} \frac{1}{2^{|S_{D\setminus E}|-1}} \rho^{\mathbf{x}}$$

in a $\mathrm{QO2}_2$ protocol we again obtain a protocol (which we refer to as $\mathrm{QO2}_3$) which is correct and blind. Blindness is trivial, as the sum over all the random parameters for the state $\rho^{\mathbf{x}}$ yields the same density operator as the sum over all random parameters for the state $(\rho')^{x_1,\ldots,x_{n-k},x}$ (and no message correlated to the summed up random parameters is sent from the client to the server). Correctness holds as the correctness of the (original) $\mathrm{QO2}_2$ protocol only depended on the parity of the k random parameters, and the construction above preserves this. □

In $\mathrm{QO2}_3$ protocols, at most one random parameter appears in the decryption only. The remainder of the claims assumes we are dealing with a $\mathrm{QO2}_3$ protocol.

Lemma 23. *The client's input bits a and b do not need to appear in the decryption function.*

Proof. In general the decryption function of the client (for $\mathrm{QO2}_3$) protocols attains the form

$$\mathrm{XOR}_D(a,b,\mathbf{x},m_s) = \mathrm{XOR}'_D(a,b,m_s) \oplus \bigoplus_{j\in S_{E\cap D}} x_j \oplus x_n \text{ or}$$
$$\mathrm{XOR}_D(a,b,\mathbf{x},m_s) = \mathrm{XOR}'_D(a,b,m_s) \oplus \bigoplus_{j\in S_{E\cap D}} x_j$$

where $S_{E\cap D}$ is the set of indices of random parameters which appear in both the decryption and encryption function, and x_n may appear only in the decryption function. Here, we have assumed without the loss of generality that it is the last random parameter that (possibly) appears only in the decryption function. First, we show that at least one random parameter must appear in the decryption, meaning that either x_n must appear or $S_{E\cap D}$ is non-empty (or both). Assume this is not the case. Then we have

$$\mathrm{XOR}_D(a,b,\mathbf{x},m_s) = \mathrm{XOR}'_D(a,b,m_s)$$

and this must be equal to ab by the correctness of the protocol. But, due to the restrictions of the client we have

$$\mathrm{XOR}'_D(a,b,m_s) = \mathrm{XOR}''_D(a,b) \oplus m_s = ab \text{ or}$$
$$\mathrm{XOR}'_D(a,b,m_s) = \mathrm{XOR}''_D(a,b) = ab$$

The latter is not possible as no function computable using only XOR can yield the output ab, so

$$\begin{aligned}&\mathrm{XOR}'_D(a,b,m_s) = \mathrm{XOR}''_D(a,b) \oplus m_s = ab \Leftrightarrow \\ &m_s = ab \oplus \mathrm{XOR}''_D(a,b).\end{aligned}$$

The function $\mathrm{XOR}''_D(a,b)$ can only be one of six functions, which are such that either a or b appear in the decryption:

$$\begin{aligned}&\mathrm{XOR}''_D(a,b) = a;\ \mathrm{XOR}''_D(a,b) = 1 \oplus a \\ &\mathrm{XOR}''_D(a,b) = b;\ \mathrm{XOR}''_D(a,b) = 1 \oplus b; \\ &\mathrm{XOR}''_D(a,b) = a \oplus b;\ \mathrm{XOR}''_D(a,b) = 1 \oplus a \oplus b.\end{aligned}$$

But, for all of these functions we have that $ab \oplus \mathrm{XOR}''_D(a,b)$ is correlated to a,b, hence not blind. For example $a \oplus b \oplus ab = a \vee b$, so if the server obtains $m_s = 0$ this means $a = b = 0$. Thus, for the protocol to be blind, at least one random parameter must appear in the decryption.

Let j be the index of this random parameter. Then x_j either appears or does not appear in the encryption. First assume x_j appears in the encryption, and let $\mathrm{XOR}''_D(a,b) = a$. Then by modifying XOR_D in such a way that it no longer depends on a (by substituting $\mathrm{XOR}''_D(a,b)$ with 0 in the definition of XOR_D) and by modifying the encryption function in such a way that all instances of x_j are substituted with $x_j \oplus \mathrm{XOR}''_D(a,b)$, we obtain a new protocol, in which the inputs a,b no longer appear in the decryption function. This protocol is correct, as the initial protocol was correct for all possible inputs and random variables, and all we have done is a substitution of variables. Since, from the perspective of the server, both $x_j \oplus \mathrm{XOR}''_D(a,b)$ and x_j are equally distributed (uniformly at random), the protocol is blind as well.

Consider now the case where x_j does not appear in the encryption (thus no random parameters appearing in the encryption appear in the decryption), and let $\mathrm{XOR}''_D(a,b)$ be the function which appears in the evaluation of the decryption, and is not constant. Then, we need to modify the messages the client sends, and the measurement the server does. Let m_c be the message the client sends in the original protocol. Then, in the modified protocol, the client will send the message $(m_c, \mathrm{XOR}''_D(a,b) \oplus y)$, where y is a new random bit. The server will perform the same measurement as in the original protocol, as defined by m_c but will output $m_s^{new} = m_s^{original} \oplus \mathrm{XOR}''_D(a,b) \oplus y$. Note that this process can be viewed as a redefinition of the measurement the server does. the client decrypts almost the same as in the original protocol, altered by substituting

$\mathrm{XOR}''_D(a,b)$ with 0, and by XORing with y . So we have:

The original decryption in original protocol :
$out = \mathrm{XOR}''_D(a,b) \oplus m_s^{original} \oplus x_j$
The new decryption in new protocol :
$0 \oplus m_s^{new} \oplus x_j \oplus y = m_s^{original} \oplus \mathrm{XOR}''_D(a,b) \oplus y \oplus x_j \oplus y = out.$

Thus, the new protocol is also correct. To see that it is blind, note that the only piece of additional information given to the server, relative to the original protocol is the bit $\mathrm{XOR}''_D(a,b) \oplus y$. However, since y is chosen uniformly at random, this reveals no extra information so the protocol is blind as well.

Thus for every $\mathrm{QO2}_3$ blind correct protocol, there exists a blind correct $\mathrm{QO2}_4$ protocol where the inputs of the client do not appear in the decryption function. □

To summarise, to this point we have shown that we only need to consider protocols in which the server's output is a single bit, at most one random parameter which appears in the decryption (but not in encryption) is used, and the decryption function does not take the inputs of the client as parameters. Additionally, we have shown that we only need the random parameters which appear either in encryption or decryption. Next, we deal with the size of the client's messages, and the number of required random parameters appearing in the encryption.

Consider the encryption, and the generated quantum state in the protocol:

$m_c = \mathrm{XOR}_E(a,b,\mathbf{x})$ − the m bit message from the client
$\rho^{\mathbf{x}},\ for\ \mathbf{x} = (x_1,\ldots,x_n)$ − the quantum state parametrized by n bits.

and let $(m_c)_j$ denote the j^{th} bit of the m bit message m_c.

Lemma 24. *No single isolated random variables are needed.*

Proof. Assume that, for some j and k we have, $(m_c)_j = x_k$. Then, the protocol reveals x_k. But this means that if we fix $x_k = 0$ (that is, by dropping that random parameter from the protocol) we yield again a blind correct protocol (with one less random parameter). We get the same if the negation of x_k appears. By repeating this, we obtain a protocol for which no part of the message is equal to a single random parameter, or its negation. □

Lemma 25. *No arbitrary XOR functions of random variables are needed.*

Proof. Next, assume that for some j and k,l we have, $(m_c)_j = x_k \oplus x_l$. Then, we can introduce the variable $x_{k,l} = x_k \oplus x_l$, and substitute all instances of x_l in the protocol with $x_{k,l} \oplus x_l$. This again yields a correct blind protocol, with the same number of

random parameters as the original protocol. However, the modified protocol has the new variable $x_{k,l}$ appearing in $(m_c)_j$ isolated, so it (by the argument in the last paragraph) be dropped from the protocol.

We can perform analogous substitutions whenever arbitrary XOR functions of random parameters appear in isolation: for a function $b \oplus x_{k_1} \oplus \cdots \oplus x_{k_p}$ we can define the substituting variable $x^b_{k_1,\ldots,k_p} = b \oplus x_{k_1} \oplus \cdots \oplus x_{k_p}$, and substitute all instances of x_{k_1} with $x^b_{k_1,\ldots,k_p} \oplus b \oplus x_{k_2} \oplus \cdots x_{k_p}$. Thus we retain exactly the same number of random parameters, but $x^b_{k_1,\ldots,k_p}$ now appears in isolation. So, this variable can be dropped.

Thus, for any $\mathrm{QO2}_4$ protocol, there exists a protocol (blind and correct) where no functions of random parameters appear in isolation in m_c.

Thus, each entry of m_c is of the form $\mathrm{XOR}(a,b,x_1,\ldots x_n)$, where this function is not constant in a or b (or both). However, it is clear that this function cannot be constant in all the random parameters $\mathbf{x}$ as otherwise the protocol would not be blind. □

We can now complete the main proof of the impossibility of quantum offline protocol by showing how the redundancies could be removed.

Proof of Theorem 21. Define

$$(m_c)_j = \mathrm{XOR}(a,b) \oplus \bigoplus_{k \in S_j \subseteq [N]} x_k$$
$$(m_c)_{k \neq j} = \mathrm{XOR}(a,b) \oplus \bigoplus_{k \in S_k \subseteq [N]} x_k$$

Then, the XOR of those two entries reveals the XOR of the random parameters with indices in the intersection $S_j \cap S_k$. Let

$$\tilde{x} = \bigoplus_{k \in S_j \subseteq [N]} x_k \oplus \bigoplus_{k \in S_k \subseteq [N]} x_k = \bigoplus_{k \in S_k \cap S_j \subseteq [N]} x_l$$

Then the original protocol is equally blind as the protocol (we will call it MOD1 for modification 1) in which the message element $(m_c)_k$ is substituted with $\tilde{x}$ and the server, upon the receipt of the message redefines $(m_c)_k := (m_c)_j \oplus x$.

For simplicity, assume that $S_k \cap S_j = \{1,2,\ldots l\}$. If we further modify MOD1 to MOD2 by substituting all instances of x_1 in this protocol with $\tilde{x} \oplus x_2 \ldots x_l$ we obtain a protocol in which $\tilde{x}$ is a randomly chosen variable, and note that it appears isolated in message element $(m_c)_k$. Thus, it can by the arguments we presented earlier, be dropped from the protocol, by setting it to zero. Note that analogous transformations of the protocol can be done if the XOR functions on two positions differ by a bit flip.

Hence, we only need to consider protocols where each function of a,b in the message of the client appears only once, where functions which differ by a bit flip can

be considered duplicates as well. There are only three XOR computable non-constant functions of two binary parameters, up to a bit flip:

$$\mathrm{XOR}(a,b) = a,\ \mathrm{XOR}(a,b) = b,\ \mathrm{XOR}(a,b) = a \oplus b$$

Thus, the message the client sends to the server, without the loss of generality, is of the form:

$$m_c = (a \oplus \bigoplus_{k \in S_1 \subseteq [n]} x_k, b \oplus \bigoplus_{k \in S_2 \subseteq [n]} x_k, a \oplus b \oplus \bigoplus_{k \in S_3 \subseteq [n]} x_k)$$

Now, we can eliminate any single one of the three, and for our purposes of reduction to the small QO2 protocol, we will eliminate the last one. Note that

$$(m_c)_3 = (m_c)_1 \oplus (m_c)_2 \oplus \bigoplus_{k \in S_1 \subseteq [n]} x_k \oplus \bigoplus_{k \in S_2 \subseteq [N]} x_k \oplus \bigoplus_{k \in S_3 \subseteq [N]} x_k,$$

and that the server can obtain

$$\tilde{x} = \bigoplus_{k \in S_1 \subseteq [n]} x_k \oplus \bigoplus_{k \in S_2 \subseteq [n]} x_k \oplus \bigoplus_{k \in S_3 \subseteq [n]} x_k$$

by XORing the three bits of the client's message. Thus, similarly to the approach we used earlier, the protocol can be further modified in such a way that $\tilde{x}$ is given as the third bit of the message. Furthermore, by substitution, the third bit can be eliminated as well. Thus we obtain the third modification of the protocol, in which the client's message is of the form

$$m_c = (a \oplus \bigoplus_{k \in S_1 \subseteq [n]} x_k, b \oplus \bigoplus_{k \in S_2 \subseteq [n]} x_k)$$

with $S_1 \cup S_2 = [n]$. Note $S_1 \neq S_2$ as otherwise the protocol would not be blind. Let S_{DE} be the subset of indices of the random parameters which appear in the decryption and encryption. Then all the random parameters in $S_1 \setminus (S_2 \cup S_{DE})$ can be substituted by only one random parameter $\tilde{x}_1$ which is the mod 2 sum of random parameters indexed in $S_1 \setminus (S_2 \cup S_{DE})$. Additionally, the quantum state the client sends to the server needs to be averaged over all states where the mod 2 sum of random parameters indexed in $S_1 \setminus (S_2 \cup S_{DE})$ is zero (for $\tilde{x}_1 = 0$) and for the case it is one (for $\tilde{x}_1 = 1$). The same can be done for all the random parameters in $S_2 \setminus (S_1 \cup S_D E)$, generating the single random parameter $\tilde{y}_1$ appearing only in $(m_c)_2$. The indices in S_{DE} must appear either in S_1 or in S_2. Let $p_1 \ldots p_q$ be the set which appears in both. Then we can substitute these random parameters with one $\tilde{p} = p_1 \oplus \cdots \oplus p_q$ by again modifying the state the client sends to the server, by averaging over those states for which $p = 0$ or $p = 1$. Similarly can be

done for those indices in S_{DE} which appear only in $(m_c)_1$ (same for $(m_c)_2$) resulting in one random parameter $\tilde{x}_2$ ($\tilde{y}_2$). Thus we obtain the protocol in which the client sends

$$m_c = (a \oplus \tilde{x}_1 \oplus \tilde{x}_2 \oplus p, b \oplus \tilde{y}_1 \oplus \tilde{y}_2 \oplus p)$$

and the decryption is given with:

$$out = m_s \oplus \tilde{x}_2 \oplus \tilde{y}_2 \oplus p \oplus r$$

where r was the random parameter not appearing in the encryption, and the quantum state is parametrized with:

$$\rho^{\tilde{x}_1,\tilde{x}_2,\tilde{y}_1,\tilde{y}_2,p,r}$$

We will refer to such protocols as $QO2_5$ protocols. Note that

$$M^{\alpha,\beta}(\rho^{\tilde{x}_1,\tilde{x}_2,\tilde{y}_1,\tilde{y}_2,p,r}) = (\alpha \oplus \tilde{x}_1 \oplus \tilde{x}_2 \oplus p)(\beta \oplus \tilde{y}_1 \oplus \tilde{y}_2 \oplus p) \oplus \tilde{x}_2 \oplus \tilde{y}_2 \oplus p \oplus r$$

and equivalently that

$$M^{\alpha,\beta}(\rho^{\tilde{x}'_1,\tilde{x}'_2,\tilde{y}'_1,\tilde{y}_2,p',r'}) = (\alpha \oplus \tilde{x}'_1 \oplus \tilde{x}'_2 \oplus p')(\beta \oplus \tilde{y}'_1 \oplus \tilde{y}'_2 \oplus p') \oplus \tilde{x}'_2 \oplus \tilde{y}'_2 \oplus p' \oplus r'.$$

Therefore we obtain the following relation:

$$\begin{aligned}
&M^{\alpha,\beta}(\rho^{\tilde{x}_1,\tilde{x}_2,\tilde{y}_1,\tilde{y}_2,p,r}) = M^{\alpha,\beta}(\rho^{\tilde{x}'_1,\tilde{x}'_2,\tilde{y}'_1,\tilde{y}'_2,p',r'}) \; if \\
&\tilde{x}_1 \oplus \tilde{x}_2 \oplus p = \tilde{x}'_1 \oplus \tilde{x}'_2 \oplus p', \; and \\
&\tilde{y}_1 \oplus \tilde{y}_2 \oplus p = \tilde{y}'_1 \oplus \tilde{y}'_2 \oplus p' \; and \\
&\tilde{x}_2 \oplus \tilde{y}_2 \oplus p \oplus r = \tilde{x}'_2 \oplus \tilde{y}'_2 \oplus p' \oplus r'.
\end{aligned}$$

Since the state ρ is parametrized by 6 independent parameters and we have three independent equations, this implies that there are 8 differing equivalency classes (as defined by the three equalities) over the set of all possible random parameters. The equivalency classes can be represented by three bits c_1, c_2, c_3 as follows:

$$\begin{aligned}
(c_1, c_2, c_3) \equiv \{&(\tilde{x}_1, \tilde{x}_2, \tilde{y}_1, \tilde{y}_2, p, r) | \tilde{x}_1 \oplus \tilde{x}_2 \oplus p = c_1 \\
&\tilde{y}_1 \oplus \tilde{y}_2 \oplus p = c_2, \tilde{x}_2 \oplus \tilde{y}_2 \oplus p \oplus r = c_3\}
\end{aligned}$$

We can then define the states ρ, averaged per equivalency class:

$$\rho^{c_1,c_2,c_3} = 1/8 \sum_{(\tilde{x}_1,\tilde{x}_2,\tilde{y}_1,\tilde{y}_2,p,r) \in (c_1,c_2,c_3)} \rho^{\tilde{x}_1,\tilde{x}_2,\tilde{y}_1,\tilde{y}_2,p,r}$$

Note that the first bit of the message the client sends to the server in $QO2_5$ is given with $(a \oplus x_1 \oplus x_2 \oplus p)$ which is equal to c_1. Similarly, the second bit $(b \oplus y_1 \oplus y_2 \oplus p)$

is equal to c_2. The decryption is given with $out = m_s \oplus x_2 \oplus y_2 \oplus p \oplus r$ which is equal to $m_s \oplus c_3$. This gives us a protocol in which the client sends

$$m_c = (a \oplus c_1, b \oplus c_2)$$

and the decryption is given with:

$$out = m_s \oplus c_3$$

where c_3 was the random parameter not appearing in the encryption, and the quantum state is parametrized with:

$$\rho^{c_1,c_2,c_3}$$

This protocol is correct by construction, and it is also blind as the classical messages the client sends are the same as in the $QO2_5$ protocol, and the quantum state is averaged over the degrees of freedom which do not appear in the abbreviated protocol - but then the averaging over the remaining free parameters yields the same state on the server's side as in the $QO2_5$ protocol. Thus it is blind as well.

But this is also a small QO2 protocol. Thus, symbolically, we have shown:

$$\exists \mathrm{QO2} \to \exists \mathrm{QO2}_1 \to \exists \mathrm{QO2}_2 \to \exists \mathrm{QO2}_3 \to \exists \mathrm{QO2}_4 \to \exists \mathrm{QO2}_5 \to \exists \text{ small QO2}$$

which implies the proof of the main theorem since we have already proven no small QO2 protocol exists. □

12.2.2 Multi Rounds

In the definition of QO2 protocols, we have explicitly demanded that client and server use only two rounds of classical communication to achieve the desired functionality. That is, after the quantum offline preparation stage, client sends one classical message to server, to which server responds. This offers the possibility that including multiple rounds of classical communication may circumvent the no-go result of Theorem 21. Now we show that this is not the case. To develop the proper intuition, first consider the very first possible extension - that the protocol ends with client sending an extra message to server. This trivially cannot help, as client's output then cannot depend on whatever server does. Next consider the case where client is allowed to send an additional message to server, to which server responds. To clarify this case, we shall use the notation of Protocol SecureAND QO2. Consider the client's round in protocol

SecureAND QO2 in which the client computes the output *out*. In the most general setting of a 4 round, instead of computing an output, client stores server's message m_s, computes some XOR_{E2} function of m_s a,b and perhaps new random parameters, which she then forwards back to server. Denote this message m_c^2. Note that the function XOR_{E2} is specified by the protocol (that is, it is known to server), and that it can only be a combination of XOR functions and negations of its arguments. It can also be a multi-bit XOR function: if $m_c^2 = (y_1, \ldots, y_k)$ is a $k-$ bit message, each bit y_l is of the form $XOR_{E2_l}(a,b,m_s,\mathbf{x},r)$ (where r are random bits), where some of the arguments may appear with a negation, or may not appear at all. Then each bit y_l of m_c^2 can be written as

$$XOR_{E2_l}(a,b,m_s,\mathbf{x},r) = XOR_{E2_l'}(a,b,\mathbf{x},r) \oplus XOR_{E2_l''}(m_s) \tag{12.7}$$

where we have just separated the function into parts which depend on m_s an which do not. We can do this since all operations that client can do, commute. Now, since the server knows all the component functions and m_s, and since the protocol is by assumption blind, the part $XOR_{E2_l'}(a,b,\mathbf{x},r)$ must be independent from a,b, when averaged over $\mathbf{x}$ and r, otherwise it would reveal information to server. We emphasize two properties of $XOR_{E2_l'}(a,b,\mathbf{x},r)$. First it does not depend on m_s, and second it does not reveal anything about the input. Hence, client could have sent $XOR_{E2_l'}(a,b,\mathbf{x},r)$ as a part of m_c, and the protocol would be extended by having server compute

$$XOR_{E2_l}(a,b,m_s,\mathbf{x},r) = XOR_{E2_l'}(a,b,\mathbf{x},r) \oplus XOR_{E2_l''}(m_s) \tag{12.8}$$

himself, after his measurement, without jeopardizing blindess or correctness. This shows that any 4 round blind and correct quantum offline protocol can be reduced to a two round protocol. However, this argument trivially generalizes: in an $2n$ round protocol, at client's k^{th} interaction step, any computation client does on server's prior responses, the inputs and random parameters, can be split into parts which depend on server's input and those which do not. Since server knows his responses, and the protocol is blind, the parts which do not depend on server's input cannot reveal any information about client's inputs, and also (by definition) do not depend on server's responses. Hence, client could have sent all of them in the first round of communication, and delegate the computations to server while maintaining security and correctness. This shows that any blind and correct, quantum off-line, $n-$round SecureAND protocol (denoted QOn) implies the existence of $QO2$. Since we have shown that the latter is impossible, so are QOn protocols. This proves our ultimate no-go result.

Theorem 22. *Blind, correct quantum-offline, n−round SecureAND protocols are impossible for every number of rounds n.*

Part VI

Conclusion

Efficient Universal Verification Two verification protocols have been built by composing elements from existing protocols and achieve a reduction in the overall complexity. After our work was published in [Kapourniotis et al., 2015] a few protocols appeared ([Kashefi and Wallden, 2015], [Hayashi and Morimae, 2015], [Broadbent, 2015]) with comparable performance. In Table 12.1 we summarise the resources needed for the protocols presented in this thesis: The first five protocols in the table are existing ones and the last two are the new ones we introduced. Our approach offers classical round complexity which scales linearly with the depth of the Toffoli gates and requires single or three qudit preparation (having a universal or a Clifford prover respectively). Most importantly it demonstrates the potential for the composition of existing protocols into protocols that benefit from more than one approaches to verification. Compared to [Kashefi and Wallden, 2015], which is also linear on the size of the computation, we need qudits instead of qubits, but our scaling of communication rounds is with respect to the Toffoli depth instead of the total depth of the computation. Compared to [Hayashi and Morimae, 2015] we have different trust assumptions (trusted preparation instead of measurement). Compared to [Broadbent, 2015] we have the extra feature of verifying the quantum output instead of the classical output, with exponentially low ε, and also our communication rounds scale with the Toffoli depth.

Further exploration could include the following topics:

- All approaches are based on the principle of adaptive measurement and injection of auxiliary states. This results to protocols that require as many steps of interaction between the verifier and the prover as the rounds of measurement, which in turn depends on the non-Clifford elements, in the best case. Further reduction of the round complexity is a desired property. Also, extension of verification to models without adaptive quantum measurement will be a great advance in our understanding.

- Verifiability has been presented in this thesis as a property unconditional to the computational power of the prover. In fact, all the approaches presented are secure against unbounded provers. Computationally secure soundness might be an interesting property to investigate, especially in combination to post-quantum cryptography, i.e. methods that are secure against computational bounded quantum adversaries. It is possible that such an approach can help in reducing the

[*] $(\log \frac{1}{\varepsilon})$-level systems
[†] not offering verification of quantum output with exponentially small ε

Protocol	**Verifier**	**Prover**	**V $\leftrightarrow$ P**
Poly-QAS [Aharonov et al., 2010]	Prep. $O(\log \frac{1}{\varepsilon})$ *	Clifford *	$O(N')$ *
Trap-based (dot.-compl.) [Fitzsimons and Kashefi, 2012]	Prep. Single	Universal	$O(N^2)$
Trap-based (triple-dot.) [Kashefi and Wallden, 2015]	Prep. Single	Universal	$O(N)$
Auxiliary states [Broadbent, 2015]	Prep. Single	Clifford	$O(N)$ †
Measuring verifier [Hayashi and Morimae, 2015]	Meas. Single	Entangl. + Q. mem.	$O(N)$ †
Composite 1 (Protocol 6)	Prep. Single *	Universal *	$O(N')$ *
Composite 2 (Protocol 7)	$\|$Toffoli$>$ *	Clifford *	$O(N')$ *

Table 12.1: Resource comparison for single verifier / single prover protocols. We list the resources for the verifier (size of systems to be prepared and sent), the resources for the prover (universal computation, Clifford computation or just quantum memory) and the number of classical (or quantum in the case of the measuring verifier) on-line rounds between the verifier and the prover. Notice that the delegated computation is any computation in BQP even in the case of a Clifford prover. The delegated computation has depth N and Toffoli-depth N'

resources for verification.

- Instead of verifying the correctness of a quantum computation, one might want to verify the existence of genuine quantum properties on the prover, such as quantum entanglement. Our definitions of verification do not cover this need, and it is possible that the existing protocols can already provide such properties (e.g. [Aharonov et al., 2010] offers a method to verify if the prover has quantum memory).

- The topic of fault tolerance of the verification protocols was not covered in much detail and is indeed essential when it comes to implementability. Both the FK protocols and the ABE protocols claim to have a fault tolerant implementation, however further study is needed to understand the complexity of these approaches.

- In a more abstract sense, verification of a quantum information processing device

by a nearly classical device is related to the falsifiability of quantum mechanics in general [Aharonov and Vazirani, 2012]. Specifically, we are concerned with the case that we have no information about the misbehaviour of a quantum object other than it follows the laws of quantum mechanics and we want to be able to verify or falsify our predictions. Since simulation of the outcome is not efficient, verifiability in the sense presented here may be a possible way to avoid this bottleneck.

Intermediate model verification We presented a verification scheme for a member of the so-called intermediate quantum models, i.e. models of quantum computing considered to have limited power compared to the full quantum computer but still solve problems not known to have an efficient classical algorithm. In particular, we adapted the approach of universal trap-based verification to a modified version of the one-pure-qubit model, which preserves the same underlying principle of restricted purity. As purity we have defined the number of pure qubits that one needs to use to evolve unitarily the state to the particular one. The need for this modification was due to the dependence of all existing approaches on adaptive measurement and state injection. Such modifications can be applicable to other intermediate models, such as the model of instantaneous quantum computation.

Further study would be beneficial in the following open questions:

1. Other models of intermediate quantum computation have been identified as strong candidates to demonstrate experimentally quantum supremacy over classical computation. An example is the boson sampling model, which is believed to contain problems (up to plausible classical complexity assumptions) that are hard classically [Aaronson and Arkhipov, 2011]. Moreover, these models exhibit the potential to be used as quantum simulators [Huh et al., 2015] and therefore their verification is crucial. The topic of boson sampling verification is considered from the viewpoint of verifying quantum supremacy in [Aaronson and Arkhipov, 2014].

2. Further study in the implementability of different versions of the one-pure-qubit model (e.g. [Liu et al., 2015]) and the application of the current verification methods. These methods have been shown to strongly rely on properties of quantum resource states and further characterization is needed.

Secure-NAND The problem of secure quantumly enhanced delegated classical computation was considered, in the setting of a client with minimal computational capabilities. In particular, the client we consider is, on the classical side, restricted only to XOR operations and random bit generation. This is arguably a minimal setting for the client where security can be obtained - XOR gates and random bits suffice for a one-time pad of the classical information of the client, and, at least for the simpler task of transmitting of confidential information, both are necessary. We contributed towards the construction of a family of such protocols that can enable such a client to compute the NAND gate on two bits (which is necessary for universal classical computation), inspired by the results in [Anders and Browne, 2009]. The simplest protocol only requires the client to prepare a single qubit state - however, the state of the qubit depends on the inputs of the client. That is, the required quantum state cannot be prepared before the input is known to the client, and thus the protocol is not quantum-offline. The task, of computing a classical NAND gate securely on a remote server by using only XOR computations was proven to be impossible for purely classical devices.

Some other directions are of interest:

- Universal delegated computation over encrypted data is considered a difficult problem. It is only recently that computationally secure solutions have been found for this general problem (fully homomorphic encryption) [Gentry, 2009], and thus far, the proposed protocols are still impractical. It is thus conceivable that hybrid approaches to secure delegated computation may be possible, raising the security (or reducing the complexity) of classical schemes, at the price of a small amount of quantum capabilities.

- In our setting, the client takes advantage of the freedom of a single-qubit system to be in a coherent superposition of two states (achieved in the temporal sequence of gates the client applies), to obtain the outcome of the computation. Alternatively, in the GHZ-based settings, it is the entanglement which helps achieve the required input-output correlations obtained by measurements on three distinct qubits. This opens up the question of what fundamental properties or resources of quantum theory allow for the shown enhancement. Indeed, our demonstration of the potential power of quantum communication in the setting with a limited-client and untrusted-server seems to be closely related to the work of [Raussendorf, 2013a] on the power of contextuality. In the latter it was shown that a necessary resource for a multi-party quantum computation of any non-linear function where

each party could only perform classical linear gates (such as XOR) together with local quantum operations is contextuality. Further work in [Hoban et al., 2011] studies the optimality of non-local resources, e.g. generalized GHZ states, in winning multi-party quantum computation games in a related setting.

Bibliography

Scott Aaronson. The scott aaronson 25.00$ prize. http://www.scottaaronson.com/blog/?p=284. accessed: Jan. 30 2015. 2007.

Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2011.

Scott Aaronson and Alex Arkhipov. Bosonsampling is far from uniform. *Quantum Info. Comput.*, 14(15-16):1383–1423, November 2014. ISSN 1533-7146.

Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008.

Dorit Aharonov and Umesh Vazirani. *Is Quantum Mechanics Falsifiable? A computational perspective on the foundations of Quantum Mechanics*. June, 2012.

Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In *Proceedings of Innovations in Computer Science 2010*, ICS2010, pages 453–, 2010.

Janet Anders and Dan E Browne. Computational power of correlations. *Physical Review Letters*, 102(5):050502, 2009.

Hussain Anwar. *Towards Fault-Tolerant Quantum Computation with Higher-Dimensional Systems*. PhD thesis, UCL (University College London), 2014.

László Babai. Trading group theory for randomness. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 421–429. ACM, 1985.

Howard Barnum, Claude Crepeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 449–458. IEEE, 2002.

S. Barz, V. Dunjko, F. Schlederer, M. Moore, E. Kashefi, and I. A. Walmsley. Secure delegated classical computing exploiting coherence. arXiv:1501.06730, 2015.

Stefanie Barz, Elham Kashefi, Anne Broadbent, Joseph F Fitzsimons, Anton Zeilinger, and Philip Walther. Demonstration of blind quantum computing. *Science*, 335(6066): 303–308, 2012.

Stefanie Barz, Joseph F Fitzsimons, Elham Kashefi, and Philip Walther. Experimental verification of quantum computation. *Nature Physics*, 9(11):727–731, 2013.

Michael Ben-Or, Claude Crepeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 249–260. IEEE, 2006.

CH Bennett. Quantum cryptography: Public key distribution and coin tossing. In *International Conference on Computer System and Signal Processing, IEEE, 1984*, pages 175–179, 1984.

Sergey B Bravyi and Alexei Yu Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002.

Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20100301. The Royal Society, 2010.

Anne Broadbent. How to verify a quantum computation. *arXiv preprint arXiv:1509.09180*, 2015.

Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 517–526. IEEE, 2009.

Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. *arXiv preprint arXiv:1211.1080*, 2012.

Daniel E Browne, Elham Kashefi, Mehdi Mhalla, and Simon Perdrix. Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics*, 9(8):250, 2007.

Chia-Hung Chien, Rodney Van Meter, and Sy-Yen Kuo. Fault-tolerant operations for universal blind quantum computation. *arXiv preprint arXiv:1306.3664*, 2013.

Sean Clark. Valence bond solid formalism for d-level one-way quantum computation. *Journal of Physics A: Mathematical and General*, 39(11):2701, 2006.

Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80(1):012304, 2009.

Vincent Danos and Elham Kashefi. Determinism in the one-way model. *Physical Review A*, 74(5):052310, 2006.

Vincent Danos, Elham Kashefi, and Prakash Panangaden. The measurement calculus. *Journal of the ACM (JACM)*, 54(2):8, 2007.

Animesh Datta and Anil Shaji. Quantum discord and quantum computing - an appraisal. *International Journal of Quantum Information*, 9(07n08):1787–1805, 2011.

David P DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9-11):771–783, 2000.

Vedran Dunjko. Ideal quantum protocols in the non-ideal physical world. *PhD Thesis, Heriot-Watt University*, 2012.

Vedran Dunjko, Elham Kashefi, and Anthony Leverrier. Blind quantum computing with weak coherent pulses. *Physical Review Letters*, 108(20), 2012. doi: 10.1103/PhysRevLett.108.200502. 16 pages, 1 figure.

Vedran Dunjko, Joseph F Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *Advances in Cryptology–ASIACRYPT 2014*, pages 406–425. Springer, 2014.

Vedran Dunjko, Theodoros Kapourniotis, and Elham Kashefi. Quantum-enhanced secure delegated classical computing. *Quant. Inf. Comput.*, 16(01, 02), 2016.

Joseph F Fitzsimons and Elham Kashefi. Unconditionally verifiable blind computation. *arXiv preprint arXiv:1203.5217*, 2012.

Austin G Fowler and Kovid Goyal. Topological cluster state quantum computing. *arXiv preprint arXiv:0805.3202*, 2008.

Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86 (3):032324, 2012.

C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, 2009.

Alexandru Gheorghiu, Elham Kashefi, and Petros Wallden. Robustness and device independence of verifiable blind quantum computing. *arXiv preprint arXiv:1502.02571*, 2015.

Vittorio Giovannetti, Lorenzo Maccone, Tomoyuki Morimae, and Terry G Rudolph. Efficient universal blind quantum computation. *Physical review letters*, 111(23): 230501, 2013.

S. Goldwasser. Multi party computations: past and present. In *PODC*, 1997.

Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 291–304. ACM, 1985.

Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.

Daniel Gottesman and Isaac L Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402 (6760):390–393, 1999.

Michal Hajdusek, Carlos A Perez-Delgado, and Joseph F Fitzsimons. Device-independent verifiable blind quantum computation. *arXiv preprint arXiv:1502.02563*, 2015.

William Hall. Cluster state quantum computation for many-level systems. *arXiv preprint quant-ph/0512130*, 2005.

Masahito Hayashi and Tomoyuki Morimae. Verifiable measurement-only blind quantum computing with stabilizer testing. *arXiv preprint arXiv:1505.07535*, 2015.

M. Hein, J. Eisert, and H. J. Briegel. Multi-party entanglement in graph states. *Physical Review A*, 69, 2004. quant-ph/0307130.

Teiko Heinosaari and Mario Ziman. Guide to mathematical concepts of quantum theory. *Acta Physica Slovaca*, 58:487–674, 2008.

M. J. Hoban, E. T. Campbell, K. Loukopoulos, and D. E. Browne. Non-adaptive measurement-based quantum computation and multi-party bell inequalities. *New Journal of Physics*, 13(2):023014, 2011. URL `http://stacks.iop.org/1367-2630/13/i=2/a=023014`.

Mark Howard and Jiri Vala. Qudit versions of the qubit pi/8 gate. *Physical Review A*, 86(2):022316, 2012.

Joonsuk Huh, Gian Giacomo Guerreschi, Borja Peropadre, Jarrod R McClean, and Alán Aspuru-Guzik. Boson sampling for molecular vibronic spectra. *Nature Photonics*, 2015.

Stephen P Jordan. Permutational quantum computing. *Quantum Information and Computation*, 10(5-6):470–497, 2010.

Theodoros Kapourniotis, Elham Kashefi, and Animesh Datta. Blindness and verification of quantum computation with one pure qubit. In *9th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2014), in Leibniz International Proceedings in Informatics*, volume 27, pages 176–204, 2014.

Theodoros Kapourniotis, Vedran Dunjko, and Elham Kashefi. On optimising quantum communication in verifiable quantum computing. *Extended abstract, Proceedings of the 15th Asian Quantum Information Science Conference (AQIS)*, 2015.

Elham Kashefi and Petros Wallden. Optimised resource construction for verifiable quantum computation. *arXiv preprint arXiv:1510.07408*, 2015.

J Kelly, R Barends, AG Fowler, A Megrant, E Jeffrey, TC White, D Sank, JY Mutus, B Campbell, Yu Chen, et al. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, 519(7541):66–69, 2015.

Emanuel Knill and Raymond Laflamme. Power of one bit of quantum information. *Physical Review Letters*, 81(25):5672, 1998.

T Lanting, AJ Przybysz, A Yu Smirnov, FM Spedalieri, MH Amin, AJ Berkley, R Harris, F Altomare, S Boixo, P Bunyk, et al. Entanglement in a quantum annealing processor. *Physical Review X*, 4(2):021041, 2014.

Nana Liu, Jayne Thompson, Christian Weedbrook, Seth Lloyd, Vlatko Vedral, Mile Gu, and Kavan Modi. The power of one qumode. *arXiv preprint arXiv:1510.04758*, 2015.

Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

K. Loukopoulos and D. E. Browne. Secure multiparty computation with a dishonest majority via quantum means. *Phys. Rev. A*, 81, 2010.

Atul Mantri, Carlos A Perez-Delgado, and Joseph F Fitzsimons. Optimal blind quantum computation. *Physical review letters*, 111(23):230502, 2013.

Matthew McKague. Interactive proofs for BQP via self-tested graph states, 2013. arXiv:1309.5675.

Mehdi Mhalla, Mio Murao, Simon Perdrix, Masato Someya, and Peter S Turner. Which graph states are useful for quantum information processing? In *Theory of Quantum Computation, Communication, and Cryptography*, pages 174–187. Springer, 2014.

Tomoyuki Morimae. Continuous-variable blind quantum computation. *Physical review letters*, 109(23):230502, 2012.

Tomoyuki Morimae and Keisuke Fujii. Blind topological measurement-based quantum computation. *Nature communications*, 3:1036, 2012.

Tomoyuki Morimae and Keisuke Fujii. Blind quantum computation protocol in which alice only makes measurements. *Physical Review A*, 87(5):050301, 2013.

Tomoyuki Morimae and Takeshi Koshiba. Impossibility of secure cloud quantum computing for classical client. *arXiv preprint arXiv:1407.1636*, 2014.

Tomoyuki Morimae, Vedran Dunjko, and Elham Kashefi. Ground state blind quantum computation on aklt state. *Quantum information & computation*, 15(3&4):0200–0234, 2015.

Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

Anna Pappa, André Chailloux, Stephanie Wehner, Eleni Diamanti, and Iordanis Kerenidis. Multipartite entanglement verification resistant against dishonest parties. *Physical review letters*, 108(26):260502, 2012.

Carlos A Perez-Delgado and Joseph F Fitzsimons. Overcoming efficiency constraints on blind quantum computation. *arXiv preprint arXiv:1411.4777*, 2014.

R. Raussendorf. Contextuality in measurement-based quantum computation. *Physical Review A*, 88(2), 2013a.

R. Raussendorf. Contextuality in measurement-based quantum computation. *Physical Review A*, 88(2), 2013b.

Robert Raussendorf, Jim Harrington, and Kovid Goyal. Topological fault-tolerance in cluster state quantum computation. *New Journal of Physics*, 9(6):199, 2007.

Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems via rigidity of chsh games. *arXiv preprint arXiv:1209.0449*, 2012.

Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7446):456–460, 2013.

R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. *Found. Secure Computation*, 1978.

R.Raussendorf and H.J.Briegel. A one-way quantum computer. *Physical Review Letters*, 86:5188, 2001.

Adi Shamir. Ip= pspace. *Journal of the ACM (JACM)*, 39(4):869–877, 1992.

Dan Shepherd. Computation with unitaries and one pure qubit. *arXiv preprint quant-ph/0608132*, 2006.

Dan Shepherd and Michael J Bremner. Temporally unstructured quantum computation. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 465, pages 1413–1439. The Royal Society, 2009.

Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997.

Justin B Spring, Benjamin J Metcalf, Peter C Humphreys, W Steven Kolthammer, Xian-Min Jin, Marco Barbieri, Animesh Datta, Nicholas Thomas-Peter, Nathan K Langford, Dmytro Kundys, et al. Boson sampling on a photonic chip. *Science*, 339 (6121):798–801, 2013.

Takahiro Sueki, Takeshi Koshiba, and Tomoyuki Morimae. Ancilla-driven universal blind quantum computation. *Physical Review A*, 87(6):060301, 2013.

Dominique Unruh. Universally composable quantum multi-party computation. In *Advances in Cryptology–EUROCRYPT 2010*, pages 486–505. Springer, 2010.

Fern HE Watson, Earl T Campbell, Hussain Anwar, and Dan E Browne. Qudit color codes and gauge color codes in all spatial dimensions. *Physical Review A*, 92(2): 022312, 2015.

Andrew C. Yao. Protocols for secure computations. In *FOCS*, 1982.

DL Zhou, B Zeng, Z Xu, and CP Sun. Quantum computation based on d-level cluster state. *Physical Review A*, 68(6):062303, 2003.