Compiling Nonlocal Games without Quantum Homomorphic Encryption

by

Kaniuar Bacho

Master's Thesis

for the degree of

Master of Science in IT Security/Networks and Systems

submitted to the Faculty of Computer Science at

Ruhr University Bochum

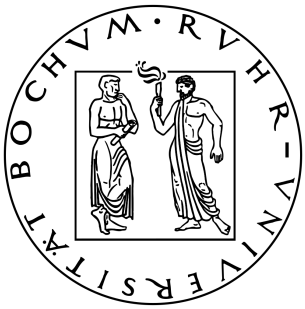
1. Examiner & Advisor: Prof. Dr. Michael Walter[1]

2. Examiner & Advisor: Prof. Dr. Giulio Malavolta[2]

3. Advisor: Dr. Simon Schmidt[1]

[1]Ruhr University Bochum

[2]Bocconi University



Abstract

In this thesis, we present a novel compiler that transforms any nonlocal game into a single-prover protocol, ensuring both quantum completeness and quantum soundness. This compiler is built upon the framework of measurement-based quantum computation and can be instantiated assuming the existence of any plain trapdoor claw-free function. Our construction relies on two cryptographic primitives:

(1) *Blind Remote State Preparation*: We present a new blind remote state preparation protocol, which is constructed from any plain trapdoor claw-free function.

(2) *Half-Blind Quantum Computation*: We present a generalization of the universal blind quantum computation protocol by Broadbent, Fitzsimons, and Kashefi. This generalization enables a client to perform computations on arbitrary quantum states held by a server while preserving the blindness of the computation.

By combining these two primitives, we construct a classical version of the half-blind quantum computation protocol, which serves as the core mechanism of our compiler.

Acknowledgments

I would like to express my gratitude to my advisor Michael Walter for making this thesis possible and for being a phenomenal teacher throughout my journey into the fascinating world of quantum information. The spark of excitement in Michael's eyes when discussing quantum information, along with his joy in teaching, has been a constant source of inspiration during my studies. His talks were not only enlightening but also more exciting than a thriller movie, giving me a reason to wake up early in the morning to ensure I never missed them. Thank you, Michael, for opening the door to this beautiful field and for being such a great professor.

I would like to truly thank my advisor Giulio Malavolta for providing this highly interesting project idea and for generously dedicating his time to supporting me whenever I needed help. This thesis would not have been possible without Giulio's remarkable ingenuity and his careful efforts to ensure everything ran smoothly throughout my work. I feel incredibly fortunate to have had Giulio as one of my advisors, offering invaluable insights into how research and great collaboration work. Thank you, Giulio, for showing me what true dedication and endurance mean.

I also want to thank Simon Schmidt and Alexander Kulpe for providing excellent advice and ensuring that my thesis progressed smoothly. I deeply appreciate their voluntary participation in our weekly meetings and their engagement in helpful discussions.

A special thank you goes to the wonderful friends I had the privilege of meeting during my master's studies. I cannot imagine how different this journey would have been without you. Thank you for the enriching conversations, memorable walks, shared meals, barbecue gatherings, vacations, chess games, and countless other joyful moments. You truly made this time unforgettable, guys!

My gratitude also goes to my parents, my brothers, and my sisters for their unconditional love. Thank you for filling my life with joy and for the precious memories we have created together. I am especially grateful to my beloved mother for always ensuring that I feel good whenever I arrive home.

Finally, I want to express my deepest gratitude to Laura for positively changing my life and reminding me of the beautiful sides of life to enjoy. Thank you for your happiness, your kindness, and your love.

Contents

Chapter 1

Introduction

A nonlocal game is a classical interaction between a referee and two non-communicating players. The game starts with the referee sampling a question pair and sending one question to each of the players. The players then respond with answers. Finally, the referee evaluates whether the answers, in conjunction with the questions, satisfy a predefined condition. If they do, the players win; otherwise, they lose. An important aspect of such games is that the players are not allowed to communicate during the execution of the game, and so they are unaware of the questions or answers of the other players. However, the rules of the game, including the winning conditions, are fixed ahead of time and are known to the players. This allows them to meet and agree beforehand on a strategy to maximize their winning probability. Additionally, they are permitted to share resources prior to the game, such as shared classical randomness or quantum resources like entangled qubits. Depending on their allowed shared resources, one speaks of the *classical value* or *quantum value* of the nonlocal game, representing the maximal winning probability under the given constraint to behave classically or quantumly.

Nonlocal games were introduced by physicist John Stewart Bell in 1964 [Bel64] in response to the Einstein–Podolsky–Rosen (EPR) paradox [EPR35], which presents a thought experiment questioning whether quantum mechanics provides a complete description of physical reality, leading to the conclusion that it should be supplemented by another specific physical theory. Bell's groundbreaking work, however, showed that this specific theory must satisfy a constraint now known as Bell's inequality. Bell further argued that the correlations of quantum mechanics violate these inequalities, demonstrating the incompatibility between those physical theories. Subsequent work by Clauser, Horne, Shimony, and Holt (CHSH) refined Bell's theorem into a more experimentally testable form [CHSH69], which is nowadays celebrated as the CHSH game, the prime example of a nonlocal game. It was shown that in this nonlocal game, the quantum value is strictly greater than the classical value. This enables the CHSH game to experimentally detect a difference between classical and quantum correlations, or in other words, to demonstrate the presence of quantum entanglement. These theoretical insights were validated through experimental tests, commonly referred to as Bell tests. The first such test was conducted by Clauser and Freedman in 1972 [FC72], followed by Alain Aspect et al. in 1982 [AGR82, ADR82] and by Anton Zeilinger et al. in 1998 [WJSWZ98]. Together, Alain Aspect, John F. Clauser, and Anton Zeilinger were awarded the 2022 Nobel Prize in Physics [Out22]. Nonlocal games have also become a fundamental subject of study in other disciplines, such as computer science, providing, for example, a protocol for *Classical Verification of Quantum Computations* [Gri19].

Since the assumption that players do not communicate is difficult to enforce from a practical point of view, recent work by Kalai, Lombardi, Vaikuntanathan, and Yang (KLVY) in [KLVY23] introduced a generic procedure for transforming any k-player nonlocal game into an interactive

protocol involving a single computationally bounded player. One refers to a generic procedure for converting any nonlocal game into a single-player protocol as a *compiler*. To achieve this, the KLVY compiler relies on the existence of a *quantum fully homomorphic encryption* (QFHE) scheme. The basic idea is to ask the single player to simulate the computations of all players in the original nonlocal game and, to ensure that no communication occurs, the question of each player is encrypted under a different key. The player should be able to compute on the encrypted questions, which is why the KLVY compiler relies on the existence of quantum homomorphic encryption schemes such as those presented in [Mah18a, Bra18].

Such a compiler has been shown to be a useful primitive not only for overcoming the spatial separation assumption but also for other applications. In the same paper, the authors of the KLVY compiler presented a new protocol for a *proof of quantumness* by transforming the CHSH game into a single-prover protocol. A protocol for a proof of quantumness is particularly interesting as it allows a classical verifier to confirm, through classical interaction with a device claiming to be quantum, that the device indeed behaves quantumly by observing behavior that cannot be explained classically. This is analogous to the case in nonlocal games, where the winning probability exceeding the classical value implies that the players cannot be classical. Furthermore, subsequent work by Natarajan and Zhang [NZ23] leveraged this compiler to develop an alternative approach for *Classical Verification of Quantum Computations* protocols. More recently, a protocol with a *succinct verifier* was proposed [MNZ24], improving on prior work [BKLMM+22], which relied on stronger cryptographic assumptions. Additional work has been done to understand the KLVY compiler itself by studying the winning probability of the compiled game, trying to bound it in terms of the original nonlocal game it arose from. Bounding the quantum value of the compiled CHSH game was done in [NZ23], followed by bounds on compiled XOR games [CMMNP+24, BVBDM+24, MPW24], a specific class of nonlocal games. Recently, [KMPSW24] provided a bound on the quantum value of all compiled nonlocal games.

As these applications demonstrate, compilers provide a modular framework for constructing quantum cryptographic protocols. Researchers can focus on the information-theoretic multi-player setting, which is typically simpler and well-studied, and then compile these nonlocal games into single-prover protocols. The established theorems about the compiler take care of the rest.

At present, we know of only two such recipes to transform nonlocal games into compiled ones, i.e., the aforementioned KLVY compiler and the one presented in the work of Arora et al. [ABCC24]. The latter proposes a compiler for *contextuality games*, a generalization of nonlocal games. As the work was published during the preparation of this thesis, the author did not explore Arora et al.'s work in full detail and will therefore not discuss it further to avoid spreading incorrect information. The KLVY compiler relies on a rather strong cryptographic primitive, namely the existence of QFHE schemes.

From a cryptographic perspective, QFHE is a rather strong primitive, both in terms of functional guarantees and in terms of the underlying computational assumptions. There is evidence that the functionality offered by QFHE is not necessary for compiling nonlocal games, as some form of *blind computation* suffices. This situation is, from a theoretical point of view, unsatisfactory, as it underscores a lack of understanding of this cryptographic process and places compiled nonlocal games on potentially thin cryptographic foundations.

The goal of our work is to improve our understanding of this cryptographic process and to place compilers for nonlocal games on more solid cryptographic foundations. Motivated by this, we propose and investigate an alternative compilation method based on potentially weaker cryptographic assumptions. Our compiler can be constructed from plain *trapdoor claw-free functions*, which themselves can be built from various computational assumptions, such as *isogeny-based group actions* presented in [AMR22], the *learning with errors* (LWE) problem

in [BCMVV18], or the *Ring-LWE* assumption in [BKVV20], whereas the only known QFHE schemes working for the KLVY compiler rely on the LWE problem.

Organization of the thesis. The remainder of the thesis is organized as follows:

In Chapter 2, we begin by providing the preliminary background and required knowledge for this thesis to make it as self-contained as possible. We define the quantum information-theoretic and cryptographic objects we will be dealing with and introduce new notation and terminology used throughout this work.

In Chapter 3, we present a new protocol for blind remote state preparation for a specific class of quantum states, based on plain trapdoor claw-free functions, which are defined in that chapter. Furthermore, we provide proofs of correctness and blindness for our construction.

In Chapter 4, we recap the measurement-based quantum computation model by gradually developing the necessary theory and reproving its key theorems. The chapter also includes a proof of the universality of the brickwork state.

In Chapter 5, we present a new protocol that generalizes the universal blind quantum computation protocol by Broadbent, Fitzsimons, and Kashefi. We also provide proofs of correctness and information-theoretical blindness for this protocol. Subsequently, we use the blind remote state preparation protocol to make the interaction in the protocol purely classical, and we once again provide proofs of correctness and computational blindness for the protocol.

In Chapter 6, we provide a brief introduction to nonlocal games and recap the KLVY compiler. Next, we introduce our novel compiler and provide proofs for quantum completeness and quantum soundness. Lastly, we present a concrete compilation process by compiling the CHSH game and compare our compiler to the KLVY compiler.

Chapter 2

Preliminaries

In this chapter, we provide the basic definitions of the quantum information-theoretic and cryptographic objects we will be dealing with. Additionally, we introduce new notation and terminology used throughout this work to establish the foundation for our results.

We denote the set of positive integers by $\mathbb{N}$, i.e., $\mathbb{N} := \{1, 2, 3, \ldots\}$. For $n \in \mathbb{N}$, we denote by $\mathbb{Z}_n$ the ring of integers modulo n, with elements in $\{0, \ldots, n-1\}$. The inner product of two bit strings $a, b \in \{0,1\}^n$ of length n is defined as

$$a \cdot b := \bigoplus_{i=1}^{n} a_i \cdot b_i \in \{0,1\},$$

where a_i and b_i refer to the i-th bits of the respective strings. Given two bit strings r_0 and r_1, we denote their concatenation by $r_0 \parallel r_1$. We denote by $\omega_n := e^{2\pi i/n}$ the n-th root of unity. For $a, b \in \mathbb{Z}$, the notation $[\![a, b]\!]$ is used to indicate the set $\{a, a+1, a+2, \ldots, b\}$. Moreover, we define $[n] := [\![1, n]\!] = \{1, \ldots, n\}$ for $n \in \mathbb{N}$. Lastly, we define the set

$$\Theta := \{k \cdot \pi/4 \mid k = 0, \ldots, 7\},$$

since we will be working extensively with it.

2.1 Quantum Information and Computation

In this section, we provide preliminary background and recap the most fundamental concepts of quantum information and quantum computation, without any claim to completeness, as this is not the goal of the thesis. For a more in-depth introduction to quantum information and quantum computation, we refer the reader to [NC10].

Quantum Information. In quantum mechanics, physical systems are often identified with Hilbert spaces. We will briefly recall the definition of a Hilbert space.

Definition 2.1 (Hilbert Space)**.** *A* Hilbert space $\mathcal{H}$ *is a complex vector space equipped with an inner product* $\langle \cdot | \cdot \rangle$ *such that the induced metric space is complete.*

Unless stated otherwise, we will restrict our discussion to finite-dimensional Hilbert spaces, where the completeness of $\mathcal{H}$ is automatically satisfied. When the dimension of the Hilbert space is d, one often identifies $\mathcal{H}$ with $\mathbb{C}^d$ equipped with the standard inner product, as these spaces are mathematically isomorphic. Elements of $\mathcal{H}$ are usually written using *bra-ket notation*; that is, vectors in $\mathcal{H}$ are denoted by $|\psi\rangle$. The Hilbert space of dual vectors is denoted by $\mathcal{H}^*$ and

consists of linear maps $\mathcal{H} \to \mathbb{C}$. Vectors in $\mathcal{H}^*$ are denoted by $\langle\psi|$, corresponding to the linear map $|\phi\rangle \mapsto \langle\psi|\phi\rangle$. We will denote the set of linear operators from $\mathcal{H}$ to $\mathcal{H}$ by $\mathrm{Lin}(\mathcal{H})$, the adjoint of $U \in \mathrm{Lin}(\mathcal{H})$ by $U^\dagger$, and the identity operator by I.

The state of a quantum system with Hilbert space $\mathcal{H}$ is represented by a density operator.

Definition 2.2 (Density Operator)**.** *A* density operator $\rho \in \mathrm{Lin}(\mathcal{H})$ *is a positive semi-definite* (PSD) *operator with trace one. A quantum state is called* pure *if the density operator has rank one; otherwise, it is called* mixed*.*

Pure states can be identified with unit-norm vectors $|\psi\rangle \in \mathcal{H}$ using the formula $\rho = |\psi\rangle\langle\psi|$, which also holds conversely, meaning that any pure state can be expressed in this way. This allows us to represent pure states as unit vectors in $\mathcal{H}$ instead of density operators.

Quantum states can be manipulated by applying unitary transformations to them.

Definition 2.3 (Unitary Operator)**.** *A* unitary operator *is a linear operator* $U \in \mathrm{Lin}(\mathcal{H})$ *that satisfies* $UU^\dagger = U^\dagger U = I$*.*

When a unitary U is applied to a state ρ (resp. $|\psi\rangle$), the resulting state is given by $U\rho U^\dagger$ (resp. $U|\psi\rangle$).

To extract information from a quantum system, measurement devices are required, which provide classical outcomes to work with. The general formalism will now be defined.

Definition 2.4 (Positive Operator-Valued Measure and Projection-Valued Measure)**.** *A* positive operator-valued measure (POVM) *on* $\mathcal{H}$ *with a finite outcome set* $\mathcal{O}$ *is a collection of PSD operators* $\{M_i\}_{i\in\mathcal{O}}$ *acting on* $\mathcal{H}$*, which satisfy the completeness property* $\sum_{i\in\mathcal{O}} M_i = I$*.*

If each M_i *is additionally an orthogonal projection, then the collection is called a* projection-valued measure (PVM)*.*

Naimark's well-known dilation theorem demonstrates how POVMs can be obtained from PVMs acting on a larger Hilbert space. This result is critically important in quantum mechanics, as it provides a way for physically realizing POVM measurements. When performing a measurement on a quantum state ρ using a POVM $\{M_i\}_{i\in\mathcal{O}}$, the outcome $i \in \mathcal{O}$ is observed with probability $p_i := \mathrm{tr}(M_i\rho)$. Furthermore, when a PVM is used for the measurement, the state collapses to $M_i\rho M_i/p_i$. For POVMs, however, the post-measurement state is not determined by the POVM itself but rather by the specific PVM that realizes it (there may be infinitely many possible realizations).

A measurement on a quantum state yields a post-measurement state that is, by definition, normalized—that is, it has trace one (or norm one in the case of pure states). In this work, however, we will also consider states that are not renormalized, as they are mathematically convenient to work with.

Definition 2.5 (Subnormalized State)**.** *A* subnormalized state *is a PSD operator acting on* $\mathcal{H}$ *with a trace less than or equal to 1 (for pure states, this corresponds to a pure state with a norm less than or equal to 1).*

Operationally, this corresponds to post-selecting on a measurement outcome without renormalizing the state.

We will often consider measurement apparatuses with multiple measurement settings, labeled by an index set $\mathcal{I}$, but with the same outcome set $\mathcal{O}$ for each setting. This is denoted by $\{\{M_{xa}\}_{a\in\mathcal{O}} : x \in \mathcal{I}\}$, where $\{M_{xa}\}_{a\in\mathcal{O}}$ is a POVM with outcomes in $\mathcal{O}$ for each $x \in \mathcal{I}$. When clear from context, we abbreviate this as $\{M_{xa}\}_{a\in\mathcal{O},x\in\mathcal{I}}$.

Another formalism for measurements involves the use of so-called observables. This formalism will be convenient for our purposes, as it provides a more concise way to describe specific procedures. It is in fact a special case of PVMs, where the outcomes are real numbers.

Definition 2.6 (Observable)**.** *An* observable $O \in \mathrm{Lin}(\mathcal{H})$ *is a Hermitian operator. If it additionally satisfies* $O^2 = I$*, it is called a* binary observable*.*

By the spectral theorem, observables have a spectral decomposition $O = \sum_i \lambda_i \Pi_i$, with eigenvalues $\lambda_i \in \mathbb{R}$, and Π_i being projections onto the i-th eigenspace of O. These projections form a PVM with an outcome set consisting of the corresponding eigenvalues. We say that we measure an observable O if we use its projections as a PVM, with the outcome set corresponding to the eigenvalues. For binary observables, the eigenvalues are obviously restricted to $\{-1, +1\}$. When measuring a binary observable O, we adopt the convention that the outcome corresponds to $m \in \{0, 1\}$ if the measured eigenvalue was originally $(-1)^m$, which is simply a reinterpretation of the outcome.

Quantum Computation. Alan Turing introduced the concept of an idealized theoretical computer, the Turing machine, even before physical computers existed [Tur36]. A Turing machine serves as a computational model in theoretical computer science, used to analyze classical algorithms and capture the fundamental aspects of algorithms regarding time complexity (the amount of computer time it takes to run the algorithm) and space complexity (the amount of memory space it takes to execute the algorithm), independent of the specific machine on which the algorithm is executed. Again, for a more in-depth introduction, we refer the reader to [NC10]. In practice, however, we typically use the so-called circuit model, which is equivalent in computational power to the Turing machine but more convenient and realistic for many applications. A circuit consists of wires encoding the information 0 or 1 (i.e., each wire contains a bit of information) and gates applied to these wires to manipulate the classical state in a controlled manner. By repeating this process, the circuit can perform computational tasks. In both models, this process is usually referred to as an algorithm, which may use random bits as a resource. We now define what it means to have a probabilistic polynomial-time algorithm.

Definition 2.7 (Probabilistic Polynomial-Time Algorithm)**.** *A* probabilistic polynomial-time (PPT) algorithm *is a probabilistic Turing machine that runs within a polynomial time bound, meaning there exists a polynomial* poly *such that for every input* $x \in \{0, 1\}^*$*, the machine halts after at most* $\mathsf{poly}(|x|)$ *steps.*

In the quantum world, there are also several equivalent computational models, each with distinct features that can be useful for practically building quantum computers. Among the well-known models are the adiabatic quantum computation model, the measurement-based quantum computation model, and the quantum circuit model. In the following, we will briefly describe the quantum circuit model. For this, we need the concept of a universal quantum gate set, which is, loosely speaking, a set of quantum gates S such that any unitary operation can be approximated to arbitrary accuracy by a finite sequence of gates from the set S. The Solovay–Kitaev theorem provides a quantitative statement indicating that the number of gates needed to achieve a desired level of approximation grows only slowly with its required accuracy. One typically considers $S = \{\mathrm{CX}, H, T\}$ as a universal quantum gate set [BMPRV00], where the definitions of these three gates are provided later in this section. We can now formally define what is meant by a quantum circuit.

Definition 2.8 (Quantum Circuit)**.** *A* quantum circuit *is a unitary operator that operates on the Hilbert space* $\mathcal{H} = (\mathbb{C}^2)^{\otimes k}$ *for some number* k *of qubits. It is given by a composition of unitary gates, each of which operates on one or two qubits, chosen from a fixed universal quantum gate set. The* size *of a quantum circuit refers to the number of gates it contains.*

Typically, the qubits are divided into input qubits and auxiliary qubits, which are assumed to be initialized in the $|0\rangle$ state unless stated otherwise. If a classical outcome is desired, a

subset of the qubits is measured after the unitary circuit has been applied. We will now extend the concept of a polynomial-time algorithm to the quantum case.

Definition 2.9 (Quantum Polynomial-Time Algorithm)**.** *A* quantum polynomial-time (QPT) algorithm *consists of a family of quantum circuits* $\{C_n\}_{n\in\mathbb{N}}$ *and a deterministic polynomial-time Turing machine that, on input* 1^n*, outputs a classical description of* C_n*.*

We emphasize that any PPT algorithm can be converted into a QPT algorithm. Specifically, for any n, there exists a quantum circuit C_n with n input qubits such that, given the input $|x\rangle$ and by measuring a suitable number of qubits, it simulates the behavior of the PPT algorithm on any bit string x of length $|x| = n$. As we are concerned with efficient algorithms, we also want to define what it means to efficiently implement a family of POVMs.

Definition 2.10 (QPT-Implementable)**.** *A family of POVMs* $\{\{\Pi_{n,i}\}_{i\in I_n}\}_{n\in\mathbb{N}}$ *is said to be* QPT-implementable *if there exists a QPT algorithm with a family of quantum circuits* $\{C_n\}_{n\in\mathbb{N}}$ *such that* C_n *realizes the POVM* $\{\Pi_{n,i}\}_{i\in I_n}$*. That is, by measuring certain output qubits and performing classical post-processing, one obtains the same probabilities as those given by the POVM.*

Lastly, we will introduce some notation used throughout this work. We begin by denoting the usual *Pauli operators* by

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ and } Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Moreover, we denote the *controlled-X*, the *controlled-Z*, the *Hadamard matrix*, and the T *matrix* by

$$\mathrm{CX}, \mathrm{CZ}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{pmatrix}.$$

Sometimes, we also write $\mathrm{CZ}_{i,j}$ to denote the application of the CZ operator between the qubits at the i-th and j-th positions, with the control on the i-th qubit. Lastly, for any $\theta \in \mathbb{R}$, we define the *rotation operators* as follows:

$$R_x(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix},$$
$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix},$$
$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}.$$

Additionally, the *phase shift* operator is defined as:

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} = e^{i\theta/2} \cdot R_z(\theta),$$

which is equivalent to $R_z(\theta)$ up to a global phase. The rotation operators are commonly known as the *rotation about the x-axis*, *rotation about the y-axis* and the *rotation about the z-axis*, respectively, with respect to the *Bloch sphere*, a visualization of the quantum states of a single qubit. Since a deeper understanding of the Bloch sphere is unnecessary here, we refer the curious reader to [NC10].

As we will be working with the quantum Fourier transform, let us define it here for the sake of completeness.

Definition 2.11 (Quantum Fourier Transform)**.** *The* quantum Fourier transform $\mathrm{QFT}_n : \mathbb{C}^n \to \mathbb{C}^n$ *is defined for any* $n \geq 2$ *as the linear map given by*

$$\mathrm{QFT}_n |x\rangle := \frac{1}{\sqrt{n}} \sum_{z \in \mathbb{Z}_n} \omega_n^{xz} |z\rangle \qquad \forall x \in \mathbb{Z}_n,$$

where we write $|z\rangle$ *with* $z \in \mathbb{Z}_n$ *for the standard basis of* $\mathbb{C}^n$*.*

It is a well-known fact that the quantum Fourier transform is a unitary operator and can be efficiently implemented [Kit95].

2.2 Cryptography

Throughout this work, we denote the security parameter by $\lambda \in \mathbb{N}$. Let 1^λ denote the bit string of length λ consisting only of ones. Now, we come to one of the most central definitions in modern cryptography.

Definition 2.12 (Negligible Function)**.** *A function* $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ *is called* negligible *if for every positive polynomial* poly *there exists an integer* $N > 0$ *such that*

$$|\mathsf{negl}(\lambda)| < \frac{1}{\mathsf{poly}(\lambda)} \qquad \forall \lambda > N.$$

This term often arises in the context of problems or assumptions indicating that specific tasks are computationally infeasible for algorithms with bounded computational resources, thereby providing a security guarantee. This concept will be demystified throughout the thesis, with explicit examples illustrating it in action.

For a finite set S, a probability distribution μ, and a (classical or quantum) algorithm $\mathcal{A}$, we write $s \leftarrow S$ or $s \leftarrow\!\!\$\ S$, $s \leftarrow \mu$, and $s \leftarrow \mathcal{A}$ to denote, respectively, sampling a uniformly random element s from S, sampling s according to the distribution μ, and running the algorithm $\mathcal{A}$ that produces the output s.

Next, we introduce probability ensembles that consider sequences of probability distributions indexed by the security parameter.

Definition 2.13 (Probability Ensemble)**.** *Let* X_λ *be a probability distribution for each* $\lambda \in \mathbb{N}$*. Then, the family of probability distributions* $\mathcal{X} := \{X_\lambda\}_{\lambda \in \mathbb{N}}$*, indexed by the security parameter* λ*, is called a* probability ensemble*.*

We are now ready to define another central concept in modern cryptography.

Definition 2.14 (Computational Indistinguishability)**.** *Let* $\mathcal{X} := \{X_\lambda\}_{\lambda \in \mathbb{N}}$ *and* $\mathcal{Y} := \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ *be two probability ensembles. We say that* $\mathcal{X}$ *and* $\mathcal{Y}$ *are* computationally indistinguishable *if there exists a negligible function* negl *such that for all QPT algorithms* $\mathcal{A}$*, it holds that*

$$\left| \Pr_{x \leftarrow X_\lambda} \left[1 \leftarrow \mathcal{A}(1^\lambda, x) \right] - \Pr_{y \leftarrow Y_\lambda} \left[1 \leftarrow \mathcal{A}(1^\lambda, y) \right] \right| \leq \mathsf{negl}(\lambda).$$

We often abbreviate computational indistinguishability by $\mathcal{X} \approx_c \mathcal{Y}$*.*

A possible interpretation of computational indistinguishability is that QPT algorithms attempting to distinguish between the two ensembles cannot succeed with reasonable probability; any such algorithm performs only negligibly better than random guessing. This definition encapsulates the idea that algorithms with computationally bounded resources gain no significant advantage in attempting to break the security of real-world schemes.

Chapter 3

A New Blind Remote State Preparation Protocol

In this chapter, we present a new protocol for *blind remote state preparation* (blind RSP) for a specific class of quantum states. We begin by formally defining a blind RSP protocol within our setting, then introduce a well-known cryptographic primitive called *trapdoor claw-free functions* (TCFs), which we ultimately use to construct the blind RSP protocol. Lastly, we provide proofs of correctness and blindness to show that our construction satisfies the properties required in the definition.

The results in this chapter enable us to replace the quantum communication in the protocol presented in Section 5.1 with classical communication, forming the central component of our compiler as outlined in Section 5.2. While this chapter offers the theoretical foundation to make this possible, it is not necessary to understand the inner workings of our blind RSP protocol, as we later adopt a modular approach that uses the abstract definition of a blind RSP protocol rather than a specific implementation. This approach allows any protocol that meets our definition to be used in the compiler, potentially enabling constructions based on new computational post-quantum assumptions. Our blind RSP protocol relies only on the existence of plain TCFs, which can be constructed from various computational assumptions. One example is the construction in [AMR22], which uses isogeny-based group actions, indicating that our compiler can ultimately be constructed from isogeny-based cryptography, for example.

3.1 Definition of Blind Remote State Preparation

A *remote state preparation* (RSP) protocol is, loosely speaking, a classical interaction between two parties, commonly referred to as the *verifier* and the *prover*. In this setup, the verifier is considered classical, while the prover is quantum. By the end of the protocol, the prover should hold a quantum state from a specified fixed set, and the verifier holds a classical description of this quantum state.

In practice, additional properties are often desirable for an RSP protocol to satisfy. Two commonly desired properties are *blindness* and *verifiability*. Loosely speaking, blindness ensures that the prover gains no information about the quantum state during the interaction with the verifier, while verifiability guarantees that the verifier can be sure that an arbitrary (computationally bounded) prover successfully interacting with the verifier must have prepared a specific quantum state. In our work, we focus on the blindness property, as it suffices for our purposes.

The general purpose of RSP protocols is to replace quantum communication (i.e., transmitting qubits between two parties) with classical communication between the parties involved. This

makes it possible to leverage the advantages provided by powerful quantum computers even when classical parties are participating. One example application is the *verifiability of delegated quantum computation* [GV19].

We will now provide a formal definition of a blind RSP protocol for remotely preparing states of the form

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

up to a random Pauli Z operator, where $\theta \in \Theta := \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$.

Definition 3.1 (Blind Remote State Preparation)**.** *A* remote state preparation (RSP) *protocol consists of a pair of interactive algorithms (V, P), with the security parameter in unary 1^λ as input: A classical probabilistic polynomial-time algorithm V, called the* verifier*, and a quantum polynomial-time algorithm P, called the* prover*. We require the protocol to satisfy the following properties:*

- (Correctness) *The protocol successfully terminates with a probability of at least $\frac{1}{\mathsf{poly}(\lambda)}$, which is inverse-polynomial in the security parameter. Furthermore, upon successful completion, the honest prover P holds the state*

  $$Z^b\,|+_\theta\rangle$$

  for some bit $b \in \{0,1\}$ and angle $\theta \in \Theta$. On the other hand, the verifier holds the pair (b, θ).

- (Blindness) *Consider the following experiment $\mathrm{Exp}(1^\lambda, V, P^*)$ played between an honest verifier V and a possibly malicious prover P^*.*

  - *The players engage in the interactive RSP protocol. If the protocol does not terminate successfully, the experiment aborts.*
  - *Let (b, θ) be the output of V.*
  - *The verifier flips a coin $c \leftarrow\!\!\$\ \{0,1\}$. If $c = 0$, V sets $\theta' := \theta$, otherwise V samples a uniform $\theta' \leftarrow\!\!\$\ \Theta$.*
  - *V sends θ' to P^*, who returns a bit c'.*
  - *The experiment outputs 1 if $c' = c$ and 0 otherwise.*

We say that an RSP protocol is blind *if for all QPT adversaries P^* there exists a negligible function* negl *such that for all $\lambda \in \mathbb{N}$ it holds that:*

$$\left|\Pr\Big[\mathrm{Exp}(1^\lambda, V, P^*) = 1 \mid \text{no abort}\Big] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda).$$

The intuition behind the definition of blindness is that, upon successful completion, no QPT adversary P^* can distinguish between the distribution of $\theta \in \Theta$ and that of a uniformly random $\theta' \in \Theta$. We expressed the game as a decisional game; however, since Θ has constant size in λ, we could also phrase it as a computational game, where the adversary must guess θ directly. Both variants are equivalent in the sense that if one has a negligible advantage for all QPT adversaries, so does the other (this holds even for polynomial-sized sets), a well-known fact in the cryptography community.

3.2 Trapdoor Claw-Free Functions

The concept of a *trapdoor claw-free function* (TCF) was first introduced in [GMR84]. Nowadays, they form a powerful post-quantum secure cryptographic primitive. Roughly speaking, a TCF

is a family of injective function pairs $(f_0, f_1) : X \to Y$ sharing the same domain and range, along with a trapdoor td. *Claw-freeness* refers to the property that, without the trapdoor, it is infeasible to find a *claw*—two elements $x_0, x_1 \in X$ such that $f_0(x_0) = f_1(x_1)$. However, with access to the trapdoor, it becomes possible to efficiently invert an image $y \in Y$ to obtain a claw (x_0, x_1) such that $f_0(x_0) = f_1(x_1) = y$.

TCFs (and their modified variants) are used in numerous applications, such as *Proof of Quantumness* [BCMVV18], *Classical Verification of Quantum Computations* [Mah18b], and *Remote State Preparation* [GV19, AMMW24], to name a few. We will utilize TCFs to construct our blind RSP protocol.

We will now recap the formal definition of a TCF, primarily following [BGKPV23].

Definition 3.2 (Trapdoor Claw-Free Function)**.** *Let λ be the security parameter. A* trapdoor claw-free function (TCF) *consists of a family of injective function pairs $(f_{0,\lambda}, f_{1,\lambda})$ and finite sets $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ with*

$$\{f_{b,\lambda} : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda\}_{(b,\lambda)\in\{0,1\}\times\mathbb{N}},$$

where we omit the subscript λ when it is clear from the context. Additionally, a TCF pair is augmented with two algorithms.

- $\mathsf{Gen}(1^\lambda)$*: On input the security parameter in unary 1^λ, the polynomial-time generation algorithm outputs a function pair (f_0, f_1) and a trapdoor* td*.*
- $\mathsf{Invert}(\mathsf{td}, y)$*: On input an image $y \in \mathcal{Y}$ and the trapdoor* td*, the polynomial-time deterministic inversion algorithm returns two preimages (x_0, x_1).*

We require a TCF to satisfy the following properties:

- (Correctness) *For all $\lambda \in \mathbb{N}$, all $x \in \mathcal{X}$, and all $b \in \{0,1\}$, it holds that:*

$$f_0(x_0) = f_1(x_1) = y \quad \text{where } (x_0, x_1) \leftarrow \mathsf{Invert}(\mathsf{td}, f_b(x)) \text{ and } ((f_0, f_1), \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda).$$

- (Efficient Superposition) *There exists a QPT algorithm that, on input the description of the functions (f_0, f_1), prepares the state*

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x\in\mathcal{X}} |x\rangle .$$

- (Claw-Freeness) *For all QPT algorithms A^* there exists a negligible function* negl *such that for all $\lambda \in \mathbb{N}$ it holds that:*

$$\Pr[(x_0^*, x_1^*) \leftarrow A^*(f_0, f_1) : f_0(x_0^*) = f_1(x_1^*)] \leq \mathsf{negl}(\lambda).$$

  where $((f_0, f_1), \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$.

Note that, given the trapdoor, we can efficiently check membership in $\mathcal{Y}$ by simply running the inversion algorithm and checking whether it succeeds. Moreover, we assume that there exists an embedding of the set $\mathcal{X}_\lambda$ into the bit strings $\{0,1\}^{p(\lambda)}$ for some fixed polynomial p.

In this thesis, we work with a plain TCF without any modifications, as defined above. Additionally, we adopt a black-box approach without referencing the internal workings of a specific TCF instantiation, making our approach much more modular. This implies that our compiler ultimately relies on the computational assumption used in the chosen TCF implementation, allowing our compiler to accommodate a variety of computational assumptions. To this end, we will briefly discuss different constructions of TCFs based on various computational assumptions.

Examples include the *Learning with Errors* (LWE) problem, which was used to create a variant of a TCF called a noisy TCF in [BCMVV18]. This was later extended to the *Ring-LWE* assumption in [BKVV20]. Finally, [AMR22] explored the use of *general cryptographic group actions*, such as isogenies on elliptic curves, to construct TCFs. Our blind RSP protocol also supports noisy TCFs, as we do not rely on the additional properties. However, for notational convenience, we will stick to the plain TCF definition established earlier.

3.3 A New Protocol

At a technical level, our work is inspired by Gheorghiu and Vidick [GV19] and by Brakerski et al. [BGKPV23]. As mentioned earlier, our protocol relies on the existence of an arbitrary plain TCF $(\mathsf{Gen}, \mathsf{Invert})$, as described in Section 3.2. We emphasize again that there exists an embedding of the set $\mathcal{X}_\lambda$ into the bit strings $\{0,1\}^{p(\lambda)}$ for some fixed polynomial p.

We are now prepared to outline our blind RSP protocol. Before presenting the main protocol, however, we will first describe a subroutine that the verifier V and prover P will execute multiple times within it.

Subroutine. The input and output of the subroutine are:

- (Input) The subroutine is parameterized by the security parameter in unary 1^λ, an integer n, and a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ held by the prover P.
- (Output) At the end of the interaction, the verifier holds a pair $(b,\theta) \in \{0,1\} \times \{0,1,2\}$, and the prover holds the state $\alpha|0\rangle + \beta(-1)^b \omega_n^\theta |1\rangle$.

The interaction between P and V proceeds as follows:

- (Verifier 1st Message) Sample $((f_0, f_1), \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$ and send (f_0, f_1) to P.
- (Prover 1st Message) Prepare the state

$$|\psi\rangle \otimes \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x\in\mathcal{X}} |x\rangle = \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x\in\mathcal{X}} \alpha|0,x\rangle + \beta|1,x\rangle\,.$$

  Then, apply the isometric mapping that evaluates f_b coherently on input the second register, with the function controlled on the first register, to obtain the state

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x\in\mathcal{X}} \alpha|0,x,f_0(x)\rangle + \beta|1,x,f_1(x)\rangle\,.$$

  Measure the last register to obtain some $y \in \mathcal{Y}$, with the residual state being

$$\alpha|0,x_0\rangle + \beta|1,x_1\rangle\,,$$

  where $f_0(x_0) = f_1(x_1) = y$. Send y to V.
- (Verifier 2nd Message) Check if $y \in \mathcal{Y}$ and abort if not. Sample two strings $r_0, r_1 \leftarrow\!\!\$\ \{0,1\}^{p(\lambda)}$ uniformly at random. Send (r_0, r_1) to P.
- (Prover 2nd Message) Consider the isometric mapping

$$\begin{aligned} M : \{0,1\} \times \mathcal{X} &\to \{0,1\} \times \mathcal{X} \times \mathbb{Z}_n \\ (b, x_b) &\mapsto (b, x_b, (-1)^{1-b}(x_b \cdot r_b)), \end{aligned}$$

where the inner product $z_b := x_b \cdot r_b \in \{0,1\}$ is computed over $\mathbb{Z}_2$ and then parsed as an element of $\mathbb{Z}_n$. Apply M to the current state to compute

$$\alpha \left|0, x_0, -(x_0 \cdot r_0)\right\rangle + \beta \left|1, x_1, x_1 \cdot r_1\right\rangle = \alpha \left|0, x_0, -z_0\right\rangle + \beta \left|1, x_1, z_1\right\rangle .$$

Apply QFT_n to the last register to obtain

$$\frac{1}{\sqrt{n}} \sum_{d' \in \mathbb{Z}_n} (\omega_n^{-d' \cdot z_0} \alpha \left|0, x_0\right\rangle + \omega_n^{d' \cdot z_1} \beta \left|1, x_1\right\rangle) \left|d'\right\rangle ,$$

where $-d' \cdot z_0, d' \cdot z_1 \in \mathbb{Z}_n$. Measure the last register in the computational basis and abort if the output $d' \neq 1$. The state becomes

$$\omega_n^{-z_0} \alpha \left|0, x_0\right\rangle + \omega_n^{z_1} \beta \left|1, x_1\right\rangle \equiv \alpha \left|0, x_0\right\rangle + \omega_n^{z_0 + z_1} \beta \left|1, x_1\right\rangle .$$

Conditioning on not aborting, measure the second register in the Hadamard basis to obtain some $d \in \{0,1\}^{p(\lambda)}$, and return the state

$$\alpha \left|0\right\rangle + \beta (-1)^{d \cdot (x_0 \oplus x_1)} \omega_n^{z_0 + z_1} \left|1\right\rangle .$$

Send d to V.

- (Verifier Output) Recompute $(x_0, x_1) \leftarrow \mathsf{Invert}(\mathsf{td}, y)$ and set $b := d \cdot (x_0 \oplus x_1)$ and $\theta := z_0 + z_1 = x_0 \cdot r_0 + x_1 \cdot r_1 \in \{0,1,2\}$, where the sum is computed over $\mathbb{Z}$.

We are now in a position to describe our main protocol.

Blind RSP Protocol. Our main protocol uses the subroutine three times, as follows:

- Run the subroutine with $n = 2$ and set $\left|+\right\rangle$ to be P's input state. Let (b_1, θ_1) be the output of V, and let $\left|\psi_1\right\rangle$ be the output state of P.
- Run the subroutine with $n = 4$ and set $\left|\psi_1\right\rangle$ to be P's input state. Let (b_2, θ_2) be the output of V, and let $\left|\psi_2\right\rangle$ be the output state of P.
- Run the subroutine with $n = 8$ and set $\left|\psi_2\right\rangle$ to be P's input state. Let (b_3, θ_3) be the output of V, and let $\left|\psi_3\right\rangle$ be the output state of P.

The prover P returns the final state $\left|\psi_3\right\rangle$, whereas the verifier V sets

$$b := b_1 \oplus b_2 \oplus b_3 \text{ and } \theta := 4\theta_1 + 2\theta_2 + \theta_3 \mod 8$$

and must multiply θ by $\pi/4$ to obtain the desired angle.

3.3.1 Correctness

Now, we will prove the correctness of our RSP protocol.

Theorem 3.3. *The RSP protocol described in Section 3.3 is correct.*

Proof. The probability that all three subroutines do not abort is $\frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{8} = \frac{1}{64}$, as in all three subroutines we need $d' = 1$. In the first subroutine, d' is uniformly random from $\mathbb{Z}_2$; in the second, it is uniformly random from $\mathbb{Z}_4$; and in the third, it is uniformly random from $\mathbb{Z}_8$. Hence, the success probability of our RSP protocol is also $\frac{1}{64}$.

Now, we will investigate the evolution of the state held by the prover, starting with $|+\rangle$. The first iteration implements the mapping

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1}\omega_2^{\theta_1}|1\rangle),$$

whereas the second implements

$$\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1}\omega_2^{\theta_1}|1\rangle) \mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1\oplus b_2}\omega_2^{\theta_1}\omega_4^{\theta_2}|1\rangle),$$

and finally, from the last iteration, we obtain

$$\begin{aligned}\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1\oplus b_2}\omega_2^{\theta_1}\omega_4^{\theta_2}|1\rangle) &\mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1\oplus b_2\oplus b_3}\omega_2^{\theta_1}\omega_4^{\theta_2}\omega_8^{\theta_3}|1\rangle)\\ &= \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b_1\oplus b_2\oplus b_3}\omega_8^{4\theta_1+2\theta_2+\theta_3}|1\rangle)\\ &= \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b}\omega_8^{\theta}|1\rangle),\end{aligned}$$

as desired. □

3.3.2 Blindness

To show blindness, we need to invoke the well-known *quantum Goldreich-Levin* theorem [AC01], specifically the version with auxiliary input that was proven in [CLLZ21].

Theorem 3.4 (Quantum Goldreich-Levin [CLLZ21])**.** *If there exists a quantum algorithm, that given a random r and an auxiliary quantum input ρ_x, it computes $r \cdot x$ with probability at least $1/2 + \varepsilon$; then there exists a quantum algorithm that takes ρ_x and extracts x with probability $4\varepsilon^2$.*

Before proving blindness, we will first show a simple technical lemma.

Lemma 3.5. *Let $n \geq 2$ be an integer and $m := \lfloor \log_2(n) \rfloor$. Then, for all $0 \leq i \leq m$, there exist functions $g_i : \{0,1\}^n \to \{0,1\}$ such that for all bit strings $(a_1, \ldots, a_n) \in \{0,1\}^n$ of length n, the following holds:*

$$a_1 + \ldots + a_n = 2^m \cdot g_m(a_1, \ldots, a_n) + \ldots + 2 \cdot g_1(a_1, \ldots, a_n) + g_0(a_1, \ldots, a_n). \tag{3.1}$$

Proof. The proof follows directly from considering the integer $a_1 + \ldots + a_n$ in its binary representation. Since this integer is at most n, we need $m + 1$ bits to represent it in the binary system. We now define $g_i(a_1, \ldots, a_n)$ as the bit corresponding to the $(i+1)$-th position (from the right). □

By considering Eq. (3.1) modulo 2, we see that

$$g_0(a_1, \ldots, a_n) = g_0(a_1, \ldots, a_n) \mod 2 = (a_1 + \ldots + a_n) \mod 2 = a_1 \oplus \ldots \oplus a_n.$$

Thus, we get

$$a_1 + \ldots + a_n = 2^m \cdot g_m(a_1, \ldots, a_n) + \ldots + 2 \cdot g_1(a_1, \ldots, a_n) + (a_1 \oplus \ldots \oplus a_n).$$

Finally, we prove that our protocol ensures blindness.

Theorem 3.6. *If* $(\mathsf{Gen}, \mathsf{Invert})$ *is a TCF, then the RSP protocol described in Section 3.3 is blind.*

Proof. We will use Lemma 3.5 for $n = 2$ and $n = 3$, where in both cases $m = 1$. We will denote the function for $n = 2$ by g_1 and the function for $n = 3$ by g'_1 for better distinction (both functions can, in fact, be represented by a mapping that outputs the most significant bit of the sum of the inputs, represented as a two-bit binary number). The following equations are now true over $\mathbb{Z}$:

$$\begin{aligned}4 \cdot \theta_1 + 2 \cdot \theta_2 + \theta_3 &= 4 \cdot (z_{1,0} + z_{1,1}) + 2 \cdot (z_{2,0} + z_{2,1}) + (z_{3,0} + z_{3,1}) \\ &= 4 \cdot (z_{1,0} + z_{1,1}) + 2 \cdot (z_{2,0} + z_{2,1} + \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \\ &= 4 \cdot (z_{1,0} + z_{1,1} + \tilde{z}_2) + 2 \cdot (z_{2,0} \oplus z_{2,1} \oplus \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}),\end{aligned}$$

where

$$\tilde{z}_3 := g_1(z_{3,0}, z_{3,1}) \text{ and } \tilde{z}_2 := g'_1(z_{2,0}, z_{2,1}, \tilde{z}_3).$$

Now, we have

$$\begin{aligned}\theta &= 4 \cdot \theta_1 + 2 \cdot \theta_2 + \theta_3 \mod 8 \\ &= 4 \cdot (z_{1,0} + z_{1,1} + \tilde{z}_2) + 2 \cdot (z_{2,0} \oplus z_{2,1} \oplus \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \mod 8 \\ &= 4 \cdot (z_{1,0} \oplus z_{1,1} \oplus \tilde{z}_2) + 2 \cdot (z_{2,0} \oplus z_{2,1} \oplus \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \mod 8 \\ &= 4 \cdot (z_{1,0} \oplus z_{1,1} \oplus \tilde{z}_2) + 2 \cdot (z_{2,0} \oplus z_{2,1} \oplus \tilde{z}_3) + (z_{3,0} \oplus z_{3,1}) \\ &=: 4 \cdot \theta'_1 + 2 \cdot \theta'_2 + \theta'_3.\end{aligned}$$

We will now gradually change the way we compute θ in the blindness experiments using a hybrid argument. First, we claim that the following distributions are computationally indistinguishable:

$$\theta = 4\theta'_1 + 2\theta'_2 + \theta'_3 \approx_c 4\theta^*_1 + 2\theta'_2 + \theta'_3$$

where $\theta^*_1 \leftarrow\$ \{0, 1\}$. Recall that

$$\theta'_1 = z_{1,0} \oplus z_{1,1} \oplus \tilde{z}_2 = x_{1,0} \cdot r_{1,0} \oplus x_{1,1} \cdot r_{1,1} \oplus \tilde{z}_2.$$

Since $\tilde{z}_2$ is independent from

$$x_{1,0} \cdot r_{1,0} \oplus x_{1,1} \cdot r_{1,1} = (x_{1,0} \parallel x_{1,1}) \cdot (r_{1,0} \parallel r_{1,1}),$$

it suffices to show that the latter is computationally indistinguishable from uniform. This follows by Theorem 3.4 (Quantum Goldreich-Levin), as otherwise there would exist an efficient extractor for $(x_{1,0} \parallel x_{1,1})$, contradicting the claw-freeness of the TCF. Repeating the same argument, we can conclude that

$$\theta \approx_c 4\theta^*_1 + 2\theta'_2 + \theta'_3 \approx_c 4\theta^*_1 + 2\theta^*_2 + \theta'_3 \approx_c 4\theta^*_1 + 2\theta^*_2 + \theta^*_3,$$

where $\theta^*_1, \theta^*_2, \theta^*_3 \leftarrow\$ \{0, 1\}$. This shows that θ is computationally indistinguishable from a uniformly random element in $\mathbb{Z}_8$ and completes our proof. □

Our blind RSP protocol succeeds with a constant probability of $\frac{1}{64}$. As a standard method, one can boost the probability to be exponentially close to 1 by simply repeating the protocol sequentially up to λ times until the first success, boosting the success probability to $1 - \left(\frac{63}{64}\right)^\lambda$. Security is obviously still guaranteed. Therefore, we can assume from now on that our blind RSP protocol succeeds with probability $1 - \mathsf{negl}(\lambda)$.

Remark 3.7. *Note that we can use the same strategy to remotely construct states in $Z^b |+_\theta\rangle$ for $\theta \leftarrow\$ \{k \cdot \pi/2^{m-1} \mid k \in \mathbb{Z}_{2^m}\}$ for any $m \in O(1)$. Simply follow the same steps in the main protocol, beginning with $n = 2$ and ending with $n = 2^m$. The blindness proof remains nearly identical. The procedure requires replacing the sums with their XORs m times, allowing the quantum Goldreich-Levin theorem to be applied again.*

Chapter 4

Recap of Measurement-Based Quantum Computation

In this chapter, we recap the *measurement-based quantum computation* (MBQC) model, originally introduced by Robert Raussendorf and Hans J. Briegel in [RB01]. We start with an introduction to the MBQC model, gradually developing the theory by reproving its key concepts step by step. Additionally, we reprove the universality of the *brickwork state*, implying that arbitrary quantum computations can be performed within the MBQC model using brickwork states, as shown by Broadbent, Fitzsimons, and Kashefi [BFK09].

In the context of this thesis, this chapter presents the theoretical model used to construct our compiler. Similar to how the well-known *universal blind quantum computation* (UBQC) protocol in [BFK09] is described within the MBQC model, our compiler—whose core component is presented in Section 5.2—is also based on the MBQC model. While it may be useful to examine the UBQC protocol more closely, it is not necessary, as we will generalize the UBQC protocol in Section 5.1 before making it the central object of our compiler in Section 5.2.

4.1 Introduction to Measurement-Based Quantum Computation

The *measurement-based quantum computation* (MBQC) model is an equivalent way to implement quantum computations, alongside other approaches like the well-known quantum circuit model or adiabatic quantum computation model. It was originally introduced by Robert Raussendorf and Hans J. Briegel in [RB01], and offers a distinct framework for quantum computing.

In the quantum circuit model, which operates similarly to the classical circuit model used in nearly all commercial laptops, computations usually begin with qubits initialized in the $|0\rangle$ state. A series of unitary transformations is then applied to subsets of these qubits, until the desired quantum state is achieved. Finally, selected qubits are measured to yield classical outcomes, which either represent the result or can be used in further computations.

In MBQC, however, the process is somewhat different. Loosely speaking, one begins by preparing a highly entangled generic *resource state*, which is essentially independent of the specific computation to be performed. Then, individual qubits are measured successively and adaptively in an appropriate basis, leading the remaining qubits into the desired quantum state by the end. The MBQC workflow can thus be divided into the following two main procedures:

1. (State Preparation) In the first step, qubits in the quantum state $|+\rangle$ are entangled in a specific way using the CZ operator to construct the resource state.

2. (Computation) In the second step, one-qubit measurements are performed on almost all qubits in a fixed order and in a specific basis, which depends on the previous measurement

outcomes. These measurements can be viewed as implementing a unitary operation such that the remaining qubits are in the desired quantum state.

Thus, the resource state and one-qubit measurements are all that are needed to perform arbitrary quantum computations.

One attractive property of this model is that, once the resource state is prepared, only single-qubit measurements are needed (i.e., no non-measurement gates, as in the quantum circuit model). Another feature is that it provides a natural formalism for separating a quantum algorithm into 'classical parts' and 'quantum parts'. In contrast, in the quantum circuit model, every computational step is regarded as quantum. This insight also led to the first protocol for *universal blind quantum computation* (UBQC), proposed in the MBQC model, in which the client requires no quantum memory [BFK09].

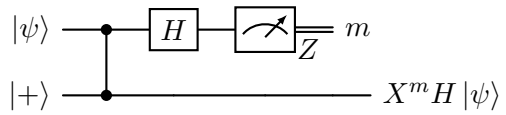
In the following, we will gradually develop the theory by systematically reviewing its key concepts step by step. To do this, we will closely follow the works of [Nie06, Joz05, MDF17]. We strongly recommend that the reader refer to Chapter 2, particularly Section 2.1, for terminology and notational conventions.

We begin with the *one-bit teleportation* scheme to understand the mechanics of the MBQC model.

Lemma 4.1 (One-Bit Teleportation [MDF17, Figure 1])**.** *Let* $|\psi\rangle \in \mathbb{C}^2$ *be an arbitrary one-qubit quantum state, and define*

$$|\Psi\rangle := (H \otimes I)\,\mathrm{CZ}(|\psi\rangle \otimes |+\rangle).$$

If the binary observable Z *is measured on the first qubit of* $|\Psi\rangle$ *with outcome* $m \in \{0,1\}$*, the quantum state of the second qubit collapses to* $X^m H\,|\psi\rangle$*. This can be expressed using the following circuit:*



Proof. The following identity

$$(\langle m|\, H \otimes I)\,\mathrm{CZ}\,(I \otimes |+\rangle) = \frac{1}{\sqrt{2}} X^m H \tag{4.1}$$

directly implies our lemma by applying $|\psi\rangle$ to both sides from the right. Proving this identity is straightforward by evaluating both sides on the basis vector $|b\rangle$ for $b \in \{0,1\}$.
On the right-hand side, we get

$$\frac{1}{\sqrt{2}} X^m H\,|b\rangle = \frac{1}{\sqrt{2}} X^m \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^b\,|1\rangle\right) = \frac{1}{2}\left(|m\rangle + (-1)^b\,|1 \oplus m\rangle\right).$$

On the left-hand side, we get

$$\begin{aligned}
(\langle m|\, H \otimes I)\,\mathrm{CZ}\,(|b\rangle \otimes |+\rangle) &= (\langle m|\, H \otimes I)\left(|b\rangle \otimes Z^b\,|+\rangle\right) \\
&= (\langle m| \otimes I)\left(\frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^b\,|1\rangle\right) \otimes Z^b\,|+\rangle\right) \\
&= (\langle m| \otimes I)\left(\frac{1}{\sqrt{2}}\,|0\rangle \otimes Z^b\,|+\rangle + \frac{(-1)^b}{\sqrt{2}}\,|1\rangle \otimes Z^b\,|+\rangle\right) \\
&= \frac{1}{2}\left(|m\rangle + (-1)^b\,|1 \oplus m\rangle\right),
\end{aligned}$$

where the last equality can be verified by checking the cases $m = 0$ and $m = 1$ separately. □

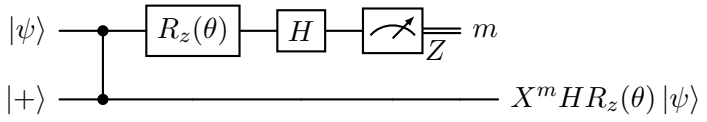
We have expressed the lemma obviously in terms of a quantum circuit. Let us take a closer look at how the lemma can be understood through the lenses of measurement-based quantum computation. Imagine the following scenario: We are given a qubit in the quantum state $|\psi\rangle$, and we want to apply the Hadamard transformation H to it. In the quantum circuit model, we would simply implement the quantum gate H to achieve this. However, in the language of MBQC, we instead create the quantum state $\mathrm{CZ}\,(|\psi\rangle \otimes |+\rangle)$, which serves conceptually as the resource state. For now, consider this state as provided by a third party, so that no quantum gate needs to be applied by us directly. Next, we interpret the Hadamard gate followed by a computational basis measurement as a Hadamard measurement, so that no gate has to be applied. This yields an outcome $m \in \{0, 1\}$. Consequently, our remaining qubit is in the quantum state $X^m H\,|\psi\rangle$, which is nearly the desired quantum state $H\,|\psi\rangle$, differing only by a local Pauli X operator. The presence of the local Pauli X operator, determined by the measurement outcome m, does not cause any problem. In a quantum computation, we would ultimately measure $H\,|\psi\rangle$ in the computational basis, and the presence of the local Pauli X operator only requires a simple reinterpretation of the output result m' as $m' \oplus m$. This reinterpretation yields the same probability distribution as if we had measured $H\,|\psi\rangle$ directly in the computational basis. Thus, we successfully implemented the Hadamard transformation H (up to a local Pauli operator) without using any quantum gates, relying solely on single-qubit measurements, assuming the resource state has been provided to us. We will mention in Section 4.2 how a specific resource state can actually be prepared without using quantum gates. All in all, this way of thinking embodies the spirit of the MBQC framework.

As a direct consequence of Lemma 4.1, we obtain the following result, referred to as the *generalized one-bit teleportation* scheme, which serves as the foundation for all further observations in the MBQC model.

Corollary 4.2 (Generalized One-Bit Teleportation [MDF17, Figure 2])**.** *Let $|\psi\rangle \in \mathbb{C}^2$ be an arbitrary one-qubit quantum state, and define*

$$|\Psi\rangle := (HR_z(\theta) \otimes I)\,\mathrm{CZ}(|\psi\rangle \otimes |+\rangle)$$

for any $\theta \in \mathbb{R}$. If the binary observable Z is measured on the first qubit of $|\Psi\rangle$ with outcome $m \in \{0, 1\}$, the quantum state of the second qubit collapses to $X^m H R_z(\theta)\,|\psi\rangle$. This can be expressed using the following circuit:



(4.2)

Proof. This follows directly from Lemma 4.1 applied to the one-qubit quantum state $R_z(\theta)\,|\psi\rangle$, along with the fact that $R_z(\theta) \otimes I$ commutes with CZ, as both operators are diagonal matrices in the standard basis. □

Note that the measurement outcome is always uniformly random for any state $|\psi\rangle$, as can be directly observed from Eq. (4.1).

Similarly to how we argued in the interpretation of the previous lemma, the z-rotation and Hadamard gate followed by measuring the binary observable Z can be considered as measuring another binary observable, which we will describe now. First, in the quantum circuit in (4.2), we can think of $R_z(\theta)$ as $P(\theta)$, since these operators are the same up to a global phase. Now, we define the following basis for $\mathbb{C}^2$:

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

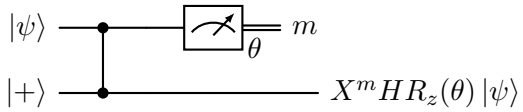
and

$$|-_\theta\rangle := \frac{1}{\sqrt{2}}\Big(|0\rangle - e^{i\theta}\,|1\rangle\Big) = P(\theta)H\,|1\rangle$$

for any $\theta \in \mathbb{R}$, which, as a side note, lie in the (X, Y)-plane of the Bloch sphere. The corresponding binary observable is given by

$$|+_\theta\rangle\langle+_\theta| - |-_\theta\rangle\langle-_\theta| = \begin{pmatrix} 0 & e^{-i\theta} \\ e^{i\theta} & 0 \end{pmatrix} = \cos(\theta)X + \sin(\theta)Y.$$

In general, applying a unitary U followed by a computational basis measurement is equivalent to performing a single-qubit measurement in the basis $\left\{U^\dagger\,|0\rangle\,, U^\dagger\,|1\rangle\right\}$. Using the notation we defined above, we can now say that the unitary $H \cdot R_z(\theta)$ followed by measuring Z gives arise to the same behavior as measuring $\cos(-\theta)X + \sin(-\theta)Y$ (since we discard the measured qubit and do not care about its post-measurement state). We will henceforth say that we are measuring in the θ-basis if we measure the binary observable $\cos(-\theta)X + \sin(-\theta)Y$. The quantum circuit in (4.2) is now rewritten as



(4.3)

where the θ in the lower right corner of the measurement gate indicates that we are measuring the binary observable $\cos(-\theta)X + \sin(-\theta)Y$.

Next, we aim to make the visualization of the quantum circuit in (4.3) even more compact and more 'MBQC-like' with the graph-like picture in Fig. 4.1.

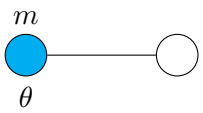
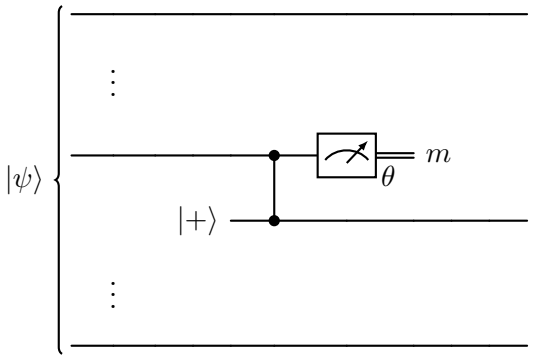


Figure 4.1: The blue circle represents a qubit in an arbitrary state $|\psi\rangle$, while the white circle always represents a qubit in the $|+\rangle$ state. The edge represents the CZ operator applied to both connected qubits (note that the CZ operator is symmetric, meaning it does not matter which qubit is the control and which is the target). Finally, the θ indicates that we are measuring in the θ-basis, and $m \in \{0, 1\}$ represents the measurement outcome.

Equipped with this new visualization, one typically speaks of an information flow from left to right in the MBQC language.

Before moving on, we want to mention that a similar result to Corollary 4.2 holds for an n-qubit quantum state. Assume that we are working with an n-qubit quantum state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$, rather than a one-qubit quantum state. In the quantum circuit (4.2), we use the i-th qubit of $|\psi\rangle$ for entangling and measuring. Finally, after measuring, we replace the i-th qubit of $|\psi\rangle$ with the

unmeasured qubit in the quantum circuit. See the quantum circuit in (4.4) for a visualization.



(4.4)

Then, the quantum state of the n qubits at the end is given by

$$\left(I^{\otimes(i-1)} \otimes X^m H R_z(\theta) \otimes I^{\otimes(n-i)}\right) |\psi\rangle \,.$$

This follows directly from Eq. (4.1) in the proof of Lemma 4.1, as we only have to add identity operators to the left and right of the equation.

To conclude this section, we demonstrate how to unleash the full potential of MBQC by considering a larger number of qubits, which also provides a clearer understanding of the more complex MBQC processes that we will encounter later. Specifically, we will show how to implement an arbitrary one-qubit unitary (up to a global phase). We rely on the well-known fact that any one-qubit unitary U can be expressed in its so-called *Euler representation*, meaning it can be decomposed as

$$U = e^{i\delta} R_x(\gamma) R_z(\beta) R_x(\alpha),$$

where α, β, γ, and δ are real numbers.

To implement this unitary transformation in the MBQC model, we use a sequence of measurements performed from left to right on a linear graph state consisting of five qubits, as illustrated in Fig. 4.2.

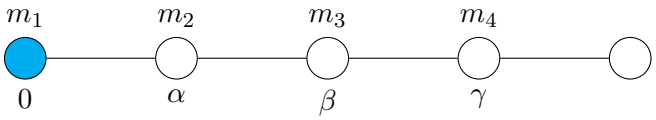


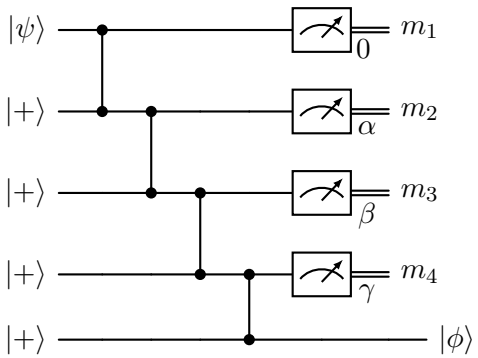
Figure 4.2: The same convention in Fig. 4.1 applies here as well.

Note that the CZ operators commute with each other, as all of these operators are diagonal matrices in the standard basis. Therefore, we do not need to specify the order in which the operators are applied. By applying Corollary 4.2 four times in this scenario—where the output qubit of each step serves as the input qubit for the next—the quantum state $|\phi\rangle$ of the final remaining qubit is

$$|\phi\rangle = X^{m_4} H R_z(\gamma) X^{m_3} H R_z(\beta) X^{m_2} H R_z(\alpha) X^{m_1} H R_z(0) |\psi\rangle \,. \tag{4.5}$$

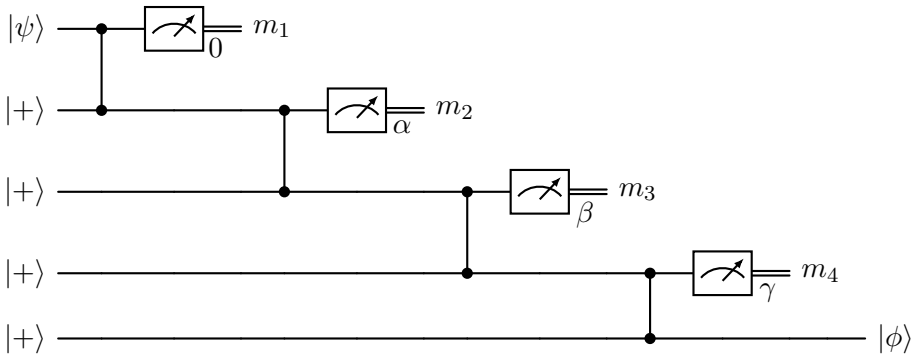
This result follows directly, as we can express the measurement-based quantum computation in

terms of an equivalent quantum circuit shown in (4.6).



(4.6)

This quantum circuit can equivalently be rewritten as (4.7), since the CZ operations can be delayed to occur after the measurement of the preceding qubit.



(4.7)

This observation shows the equality in Eq. (4.5). To simplify that expression, we use the following 'propagation' relations, which are easily verified:

$$\begin{aligned} HX &= ZH \\ HR_x(\theta) &= R_z(\theta)H \\ R_z(\theta)X &= XR_z(-\theta) \\ Z &\equiv R_z(\pi), \end{aligned} \tag{4.8}$$

where the last congruence denotes equality up to a global phase. For quantum states, we will nevertheless use an equality sign, even when the equality holds only up to a global phase (as we already did in the previous lemma and corollary). By repeatedly applying these identities to Eq. (4.5), we finally obtain

$$|\phi\rangle = X^{m_4}Z^{m_3} \cdot R_x((-1)^{m_3}\gamma + m_2\pi) \cdot R_z((-1)^{m_2}\beta + m_1\pi) \cdot R_x((-1)^{m_1}\alpha)\,|\psi\rangle\,. \tag{4.9}$$

If all measurement outcomes were 0, i.e., $m_i = 0$ for $1 \leq i \leq 4$, then this would exactly implement the intended unitary U. However, this is not always the case (the probability of this occurrence is actually $\frac{1}{2^4}$). This presents us a new challenge in handling the local Pauli X operators that arise in the middle of the computation. Since these local Pauli operators are an undesired byproduct of implementing the unitary, they are often called *byproducts* in the MBQC framework. We will now discuss how to handle these byproducts in this specific case, giving a sneak peek of a general procedure for another class of resource states in Section 4.2.2. The solution to this problem is to select subsequent measurement angles based on the outcomes of previous measurements. In

this way, most of the byproducts will disappear. Specifically, one proceeds as follows: After the first measurement, the second measurement angle is chosen as $\alpha' := (-1)^{m_1}\alpha$. After the second measurement, the third angle is set to $\beta' := (-1)^{m_2}\beta + m_1\pi$, and finally, $\gamma' := (-1)^{m_3}\gamma + m_2\pi$ is used for the last angle. The updated angles are illustrated in Fig. 4.3.

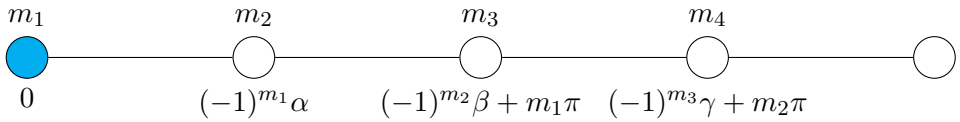


Figure 4.3: Adaptive measurements: taking previous measurement outcomes into account.

Note that the equality in Eq. (4.9) still holds; we only need to replace α, etc., with their adjusted values (i.e., α', etc.). Plugging these into the equation, we find:

$$\begin{aligned} |\phi\rangle &= X^{m_4}Z^{m_3} \cdot R_x((-1)^{m_3}\gamma' + m_2\pi) \cdot R_z((-1)^{m_2}\beta' + m_1\pi) \cdot R_x((-1)^{m_1}\alpha') \, |\psi\rangle \\ &= X^{m_4}Z^{m_3} \cdot R_x(\gamma) \cdot R_z(\beta) \cdot R_x(\alpha) \, |\psi\rangle \\ &= X^{m_4}Z^{m_3} \cdot U \, |\psi\rangle \, . \end{aligned} \tag{4.10}$$

This example shows that measurements must be carried out adaptively to maintain control over the executed computation. Moreover, we have shown that we can successfully implement an arbitrary one-qubit unitary U in the MBQC model, up to local Pauli operators. We have already seen how to handle the Pauli X by reinterpreting the measurement outcome. To handle the Pauli Z, we don't need to do anything, as it does not affect the probability distribution and can therefore be ignored.

4.2 Universality

In this section, we will show that arbitrary quantum computations can be performed within the MBQC model using brickwork states, thereby establishing the universality of brickwork states [BFK09]. Brickwork states are a specific type of resource states and will be defined in Section 4.2.1. The proof in Section 4.2.3 provides a concrete implementation of this MBQC procedure to realize any unitary U to arbitrary precision.

The first resource state proven to be universal appeared in [RB01], referred to as the *2-dimensional cluster state*. Loosely speaking, a 2-dimensional cluster state consists of qubits in the $|+\rangle$ state, arranged in a lattice and entangled using the CZ operator. See Fig. 4.4 for an example.

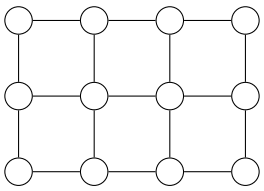


Figure 4.4: The cluster state of dimension 3×4.

Strictly speaking, we should refer to a *family* of 2-dimensional cluster states, as their size can vary based on the number of rows and columns they have. From an implementation perspective, cluster states can be efficiently created in systems with quantum Ising-type interactions (at very low temperatures) between two-state particles arranged in a lattice configuration [RB01].

Brickwork states were introduced because they have the advantage of being universal using only measurements in the (X, Y)-plane of the Bloch sphere, enabling the construction of *blind delegated quantum computing* (blind DQC) protocols [BFK09]. The work in [RB01] established the universality of cluster states for measurements in the (X, Y)-plane combined with Z-measurements. Later, it was shown in [MDF17] that Z-measurements are not necessary for cluster states to achieve universality. This result implicitly showed that any blind DQC protocol based on MBQC can also use cluster states as a resource. Therefore, cluster states could theoretically be used in this work as well to achieve the same results. However, we choose to proceed with the brickwork state introduced in [BFK09], as our work builds upon it.

4.2.1 Brickwork State

We will now define a generic resource state, which we refer to as the brickwork state.

Definition 4.3 (Brickwork State [BFK09, Definition 1])**.** *A* brickwork state $\mathcal{G}_{n\times m}$*, where* $m \equiv 5 \mod 8$*, is an entangled state of* $n \times m$ *qubits constructed as follows:*

1. *Prepare all qubits in state* $|+\rangle$ *and assign to each qubit an index* (i, j)*,* i *being a row* $(i \in [n])$ *and* j *being a column* $(j \in [m])$*.*
2. *For each row, apply the operator* CZ *on qubits* (i, j) *and* $(i, j + 1)$ *where* $j \in [m - 1]$*.*
3. *For each column* $j \equiv 3 \mod 8$ *and each odd row* i*, apply the operator* CZ *on qubits* (i, j) *and* $(i + 1, j)$ *and also on qubits* $(i, j + 2)$ *and* $(i + 1, j + 2)$*.*
4. *For each column* $j \equiv 7 \mod 8$ *and each even row* i*, apply the operator* CZ *on qubits* (i, j) *and* $(i + 1, j)$ *and also on qubits* $(i, j + 2)$ *and* $(i + 1, j + 2)$*.*

We provide an illustration of the brickwork state $\mathcal{G}_{n\times m}$ in Fig. 4.5.

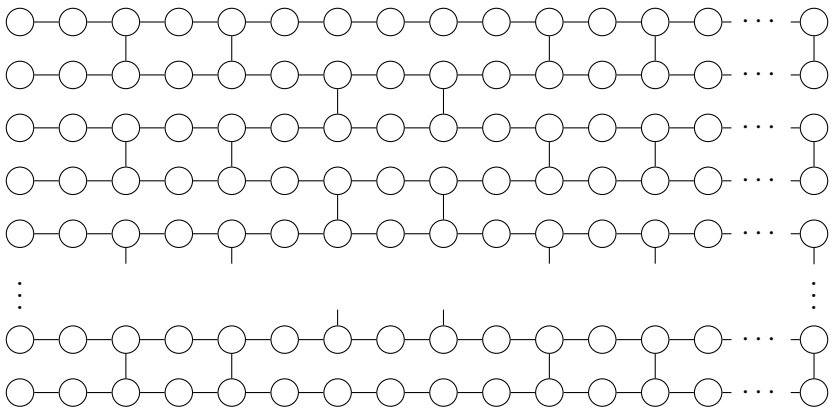


Figure 4.5: The brickwork state $\mathcal{G}_{n\times m}$.

Moreover, we refer to the specific brickwork state $\mathcal{G}_{2\times 5}$ as the *unit cell*, illustrated in Fig. 4.6.

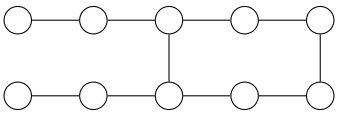


Figure 4.6: The unit cell.

The special role of the unit cell will become evident in the proof of universality in Section 4.2.3. As a teaser, the general brickwork state can (to some extent) be subdivided into unit cells, which explains the restriction $m \equiv 5 \mod 8$.

As an important side note, while the original brickwork state is defined for specific values of m, we can, of course, define it for all m. In fact, we will later drop this restriction to simplify the proof of a specific result by using induction on m. Furthermore, we will see us often in the scenario where we have an n'-qubit state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n'}$ and use $n \leq n'$ of those qubits as the first layer in the brickwork state instead of qubits in the $|+\rangle$ state. In other words, we allow the n qubits in the first layer to be in an arbitrary quantum state, which may even be entangled with the environment (i.e., being part of a larger quantum state).

4.2.2 Measurement Pattern

Now that we have our resource state in hand, we need to describe how to measure the qubits within it. We begin by defining what a measurement pattern is.

Definition 4.4 (Measurement Pattern)**.** *A* measurement pattern *for $\mathcal{G}_{n\times m}$ consists of a series of angles $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$, where $\phi_{x,y} \in \mathbb{R}$, specifying that qubit (x, y) should be measured in the $\phi_{x,y}$-basis. The measurements are performed in a specific order, starting with the leftmost column and proceeding from top to bottom. Specifically, we begin by measuring the qubit at position $(1, 1)$, then $(2, 1)$, and so forth up to $(n, 1)$. After completing the first column, the process continues to the next column, following the same procedure, until all but the final column have been measured. The qubits in the last column are referred to as* output qubits*. We will refer to this overall measurement process as* executing the measurement pattern*.*

Assume we are executing a measurement pattern $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ on $\mathcal{G}_{n\times m}$. Using Corollary 4.2, we can describe the unitary evolution of this computation, as illustrated in the example from Fig. 4.2. Each row of physical qubits in the brickwork state can be viewed as representing one logical qubit, analogous to the quantum circuit model. For now, assume that every measurement outcome is 0 during the execution of the measurement pattern. Under this assumption, no local Pauli operators appear as byproducts during the computation, making the computation deterministic. This means by repeatedly executing the same measurement pattern with all outcomes assumed to be 0, we always implement the same unitary V. Ideally, we would like this scenario to always occur to gain more control over the implemented unitary. However, as we have seen, measurement outcomes are uniformly random, leading to the appearance of local Pauli operators during the computation. These byproducts may result in a quantum computation yielding a unitary completely different from V. This phenomenon was previously observed in Fig. 4.2, where the implemented unitary was explicitly written on the right-hand side of Eq. (4.5). Specifically, let

$$\begin{aligned} V &:= HR_z(\gamma)HR_z(\beta)HR_z(\alpha)H \\ &= R_x(\gamma)R_z(\beta)R_x(\alpha) \end{aligned}$$

represent the unitary that arises when all measurement outcomes are 0. It is obvious that if not all measurement outcomes are 0, the resulting unitary may differ from V. Thus, executing a measurement pattern with the specified angles may result in the implementation of an unintended unitary.

We solved this issue, by adaptively modifying the measurement angles during the computation, as demonstrated in Fig. 4.3, ensuring that the implemented unitary at the end is $X^{m_4}Z^{m_3} \cdot V$ (see Eq. (4.10)), which corresponds to our desired unitary V, up to local Pauli operators. While it is impossible to eliminate local Pauli operators entirely, this approach allows us to achieve a

satisfactory form, where the desired unitary is followed by local Pauli operators. As previously argued, these local Pauli operators do not interfere with the computation when the state is measured in the computational basis. Therefore, the implementation of the desired unitary V followed by local Pauli operators is effectively equivalent to implementing V directly.

Our ultimate goal now is to describe a procedure for updating the angles in the measurement pattern during execution, ensuring that we implement the unitary corresponding to the case where all measurement outcomes are 0, thereby enabling deterministic computation (up to local Pauli operators in the end). We now provide a formula for the updated measurement angle $\phi'_{x,y}$ at position (x, y), which depends on $\phi_{x,y}$ and prior measurement outcomes.

A general theory for updating measurement angles during the execution of a measurement pattern for arbitrary resource states that satisfy specific properties was described by Danos and Kashefi in [DK06]. However, we will not delve deeply into this work. Instead, we use its results, tailored to our specific purposes, which suffice for our analysis and still ensure a clear understanding.

First, define $f : [n] \times [m-1] \to [n] \times [m]$ by $f(x, y) = (x, y+1)$. In the context of the brickwork state, this function maps a qubit to the qubit directly to its right. This visualization is particularly helpful when calculating the updated measurement angle. Next, define the *X-dependencies* of the qubit at position (x, y) by the set

$$D_{x,y} := f^{-1}(x, y) = \begin{cases} \emptyset & \text{if } y = 1 \\ \{(x, y-1)\} & \text{if } y > 1 \end{cases} \qquad \forall x \in [n], y \in [m]$$

and the *Z-dependencies* by

$$D'_{x,y} := \{(a, b) \mid b < y, (x, y) \in N(f(a, b))\} \qquad \forall x \in [n], y \in [m],$$

where $N(x, y)$ denotes the set of neighbors of (x, y) in the brickwork state, i.e., all vertices connected to (x, y). We measure the brickwork state in the order described earlier (from left to right and top to bottom). Denote the measurement outcome at position (x, y) by $s_{x,y}$. We define

$$s^X_{x,y} := \bigoplus_{i \in D_{x,y}} s_i = \begin{cases} 0 & \text{if } y = 1 \\ s_{x,y-1} & \text{if } y > 1 \end{cases} \qquad \forall x \in [n], y \in [m]$$

and

$$s^Z_{x,y} := \bigoplus_{i \in D'_{x,y}} s_i \qquad \forall x \in [n], y \in [m].$$

Finally, we define the modified measurement angles as

$$\phi'_{x,y} := (-1)^{s^X_{x,y}} \cdot \phi_{x,y} + s^Z_{x,y} \cdot \pi. \tag{4.11}$$

Thus, $\phi'_{x,y}$ depends on the outcomes of at most two previous layers.

Lastly, we must prove that these updated measurement angles indeed work as intended. Specifically, by executing the updated measurement pattern $\{\phi'_{x,y}\}_{x \in [n], y \in [m-1]}$, we implement the same unitary operation as if we had executed the original measurement pattern $\{\phi_{x,y}\}_{x \in [n], y \in [m-1]}$, with the measurement outcomes always being 0 (up to local Pauli operators at the end). To prove this, we will consider the brickwork state for arbitrary dimensions m, without restricting it to $m \equiv 5 \mod 8$, as discussed earlier in Section 4.2.1.

Before proceeding with the proof, we require a small technical lemma.

Lemma 4.5. *For all $a, b, c, d \in \{0, 1\}$, the following equality holds, up to a global phase:*

$$\mathrm{CZ} \cdot \left(X^a Z^b \otimes X^c Z^d\right) = \left(X^a Z^{b \oplus c} \otimes X^c Z^{d \oplus a}\right) \cdot \mathrm{CZ}\,.$$

Proof. This result follows directly from the two identities:

$$\mathrm{CZ}\cdot(X\otimes I) = (X\otimes Z)\cdot \mathrm{CZ}$$

and

$$\mathrm{CZ}\cdot(I\otimes X) = (Z\otimes X)\cdot \mathrm{CZ},$$

as well as the fact that CZ commutes with $Z\otimes I$ and $I\otimes Z$. □

We are now prepared to prove the following proposition. It is important to note that throughout this discussion, the equality sign denotes equality up to a global phase, which suffices for our purposes. Furthermore, when we state that two unitaries are the same, we mean that this equality is also up to a global phase.

Proposition 4.6. *Let $\mathcal{G}_{n\times m}$ be a brickwork state, where the n qubits in the first layer are in an arbitrary n-qubit quantum state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$. Additionally, let $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ be a measurement pattern on $\mathcal{G}_{n\times m}$, and let U be the unitary corresponding to the quantum computation when the measurement pattern is executed with all measurement outcomes being 0. Now, let $\{\phi'_{x,y}\}_{x\in[n],y\in[m-1]}$ denote the updated measurement pattern, as described in Eq. (4.11). When the updated measurement pattern is executed, the unitary corresponding to this quantum computation is given by*

$$\left(X^{s^X_{1,m}}Z^{s^Z_{1,m}}\otimes\ldots\otimes X^{s^X_{n,m}}Z^{s^Z_{n,m}}\right)\cdot U.$$

The quantum state at the end of the execution is then given by

$$\left(X^{s^X_{1,m}}Z^{s^Z_{1,m}}\otimes\ldots\otimes X^{s^X_{n,m}}Z^{s^Z_{n,m}}\right)U\,|\psi\rangle\,.$$

Proof. We will prove this by induction on m.

For $m=1$, the statement is trivially true, as we have the empty measurement pattern, which implements the identity, i.e., $U=I$. Moreover, $s^X_{i,m}=s^Z_{i,m}=0$ for all i.

For $m=2$, the statement is also easily seen to be true, as the unitary corresponding to the execution of the updated measurement pattern equals

$$\begin{aligned}
&X^{s_{1,1}}HR_z(\phi'_{1,1})\otimes\ldots\otimes X^{s_{n,1}}HR_z(\phi'_{n,1})\\
&=(X^{s_{1,1}}\otimes\ldots\otimes X^{s_{n,1}})\cdot\left(HR_z(\phi'_{1,1})\otimes\ldots\otimes HR_z(\phi'_{n,1})\right)\\
&=(X^{s_{1,1}}\otimes\ldots\otimes X^{s_{n,1}})\cdot(HR_z(\phi_{1,1})\otimes\ldots\otimes HR_z(\phi_{n,1}))\\
&=(X^{s_{1,1}}\otimes\ldots\otimes X^{s_{n,1}})\cdot U\\
&=\left(X^{s^X_{1,2}}Z^{s^Z_{1,2}}\otimes\ldots\otimes X^{s^X_{n,2}}Z^{s^Z_{n,2}}\right)\cdot U.
\end{aligned}$$

To show that the proposition holds for arbitrary m, we assume that it is true for a specific $m\geq 2$ and show that this implies it is also true for $m+1$. For this, consider a measurement pattern $\{\phi_{x,y}\}_{x\in[n],y\in[m]}$ on $\mathcal{G}_{n\times(m+1)}$ that implements U when all measurement outcomes are 0. Let U' denote the unitary obtained by measuring the 'sliced measurement pattern' $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$, with all measurement outcomes being 0.

We now provide a formula for the unitary implemented when executing $\{\phi'_{x,y}\}_{x\in[n],y\in[m]}$ and justify it afterward:

$$\begin{aligned}
&\left(\mathrm{CZ}^{a_1}_{1,2}\cdot\mathrm{CZ}^{a_2}_{2,3}\cdots\mathrm{CZ}^{a_{n-1}}_{n-1,n}\right)\cdot\left(X^{s_{1,m}}HR_z(\phi'_{1,m})\otimes\ldots\otimes X^{s_{n,m}}HR_z(\phi'_{n,m})\right)\\
&\cdot\left(X^{s^X_{1,m}}Z^{s^Z_{1,m}}\otimes\ldots\otimes X^{s^X_{n,m}}Z^{s^Z_{n,m}}\right)\cdot U',
\end{aligned}\tag{4.12}$$

for specific values of $a_1, \ldots, a_{n-1} \in \{0,1\}$ depending on $m+1$. Note that by executing the updated measurement pattern $\{\phi'_{x,y}\}_{x\in[n],y\in[m]}$, we are simultaneously executing the 'sliced updated measurement pattern' $\{\phi'_{x,y}\}_{x\in[n],y\in[m-1]}$, as the formulas for updating the angles remain the same in both cases. Hence, we can apply the induction hypothesis, justifying the second line in (4.12). However, we are not done yet, as we still need to measure the qubits in the m-th layer, which yields the second product in the first line in (4.12). Depending on the value of $m+1$, we have to consider the vertical CZ operators appearing in the brickwork state in Fig. 4.5, which are part of the quantum computation. There are three possible cases depending on the value of $m+1$:

1. Case $m+1 \equiv 3,5 \mod 8$: Here, $a_i = 1$ if and only if i is odd.
2. Case $m+1 \equiv 7,9 \mod 8$: Here, $a_i = 1$ if and only if i is even.
3. All other cases: Here, $a_i = 0$ for all i.

So far, we have justified that our unitary is indeed given by the formula in (4.12). What remains is to show that this unitary equals

$$\left(X^{s^X_{1,m+1}} Z^{s^Z_{1,m+1}} \otimes \ldots \otimes X^{s^X_{n,m+1}} Z^{s^Z_{n,m+1}}\right) \cdot U.$$

Let us first consider the third case where $a_i = 0$ for all i. We compute:

$$\begin{aligned}
&\left(X^{s_{1,m}} HR_z(\phi'_{1,m}) \otimes \ldots \otimes X^{s_{n,m}} HR_z(\phi'_{n,m})\right) \cdot \left(X^{s^X_{1,m}} Z^{s^Z_{1,m}} \otimes \ldots \otimes X^{s^X_{n,m}} Z^{s^Z_{n,m}}\right) \cdot U' \\
&= \left(X^{s_{1,m}} HR_z(\phi'_{1,m}) Z^{s^Z_{1,m}} X^{s^X_{1,m}} \otimes \ldots \otimes X^{s_{n,m}} HR_z(\phi'_{n,m}) Z^{s^Z_{n,m}} X^{s^X_{n,m}}\right) \cdot U' \\
&= \left(X^{s_{1,m}} HX^{s^X_{1,m}} R_z(\phi_{1,m}) \otimes \ldots \otimes X^{s_{n,m}} HX^{s^X_{n,m}} R_z(\phi_{n,m})\right) \cdot U' \\
&= \left(X^{s_{1,m}} Z^{s^X_{1,m}} HR_z(\phi_{1,m}) \otimes \ldots \otimes X^{s_{n,m}} Z^{s^X_{n,m}} HR_z(\phi_{n,m})\right) \cdot U' \\
&= \left(X^{s_{1,m}} Z^{s^X_{1,m}} \otimes \ldots \otimes X^{s_{n,m}} Z^{s^X_{n,m}}\right) \cdot U'' \\
&= \left(X^{s_{1,m}} Z^{s^X_{1,m}} \otimes \ldots \otimes X^{s_{n,m}} Z^{s^X_{n,m}}\right) \cdot U \\
&= \left(X^{s^X_{1,m+1}} Z^{s^Z_{1,m+1}} \otimes \ldots \otimes X^{s^X_{n,m+1}} Z^{s^Z_{n,m+1}}\right) \cdot U,
\end{aligned}$$

where in the first equation, $XZ = -ZX$ was used; in the second and third equations, definitions of updated angles were expanded, and the identities in (4.8) were applied multiple times; in the fourth equation, the unitary U'' was introduced by merging the $HR_z(\phi_{i,m})$ with U'; in the fifth equation, it was observed that $U'' = U$, as no CZ operators appear in the $(m+1)$-th layer; finally, in the last equation, it was observed that the exponents of the Pauli operators are identical.

Let us move on to the first case, i.e., $a_i = 1$ for odd i. We will perform another case distinction based on the parity of n. For now, let us assume that n is even. Following the same approach as above, we reach the point where our unitary equals

$$\left(\mathrm{CZ}^{a_1}_{1,2} \cdot \mathrm{CZ}^{a_2}_{2,3} \cdots \mathrm{CZ}^{a_{n-1}}_{n-1,n}\right) \cdot \left(X^{s_{1,m}} Z^{s^X_{1,m}} \otimes \ldots \otimes X^{s_{n,m}} Z^{s^X_{n,m}}\right) \cdot U''.$$

What remains is to propagate the local Pauli operators through the CZ operators using Lemma 4.5. To this end, let i be an odd index. Since $i \leq n$ is odd and n is even, we still have $i+1 \leq n$. We then consider the following chain of equations:

$$\begin{aligned}
&\mathrm{CZ}_{i,i+1} \cdot \left(X^{s_{i,m}} Z^{s^X_{i,m}} \otimes X^{s_{i+1,m}} Z^{s^X_{i+1,m}}\right) \\
&= \left(X^{s_{i,m}} Z^{s^X_{i,m} \oplus s_{i+1,m}} \otimes X^{s_{i+1,m}} Z^{s^X_{i+1,m} \oplus s_{i,m}}\right) \cdot \mathrm{CZ}_{i,i+1}
\end{aligned}$$

$$= \left(X^{s^X_{i,m+1}} Z^{s^Z_{i,m+1}} \otimes X^{s^X_{i+1,m+1}} Z^{s^Z_{i+1,m+1}} \right) \cdot \mathrm{CZ}_{i,i+1},$$

where the first equality follows from Lemma 4.5, and the second by observing that the exponents of the Pauli operators are identical. Putting this together for all i, we obtain the unitary

$$\begin{aligned} &\left(X^{s^X_{1,m+1}} Z^{s^Z_{1,m+1}} \otimes \ldots \otimes X^{s^X_{n,m+1}} Z^{s^Z_{n,m+1}} \right) \cdot \left(\mathrm{CZ}^{a_1}_{1,2} \cdot \mathrm{CZ}^{a_2}_{2,3} \cdots \mathrm{CZ}^{a_{n-1}}_{n-1,n} \right) \cdot U'' \\ = &\left(X^{s^X_{1,m+1}} Z^{s^Z_{1,m+1}} \otimes \ldots \otimes X^{s^X_{n,m+1}} Z^{s^Z_{n,m+1}} \right) \cdot U, \end{aligned}$$

where the equation holds because U'' represents the unitary before incorporating the CZ operators that appear in the brickwork state in the $(m+1)$-th layer, and together they form U.

The same argument clearly applies to the case where n is odd. In this case, the local Pauli operator $X^{s_{n,m}} Z^{s^X_{n,m}}$ can be propagated to the left without passing through a CZ. All other local Pauli operators are handled as in the case where n is even.

Finally, the third case, where $a_i = 1$ for even i, follows the same reasoning. We omit the details, as all the necessary insights to understand the argument have already been provided. □

The proposition can be further generalized by allowing the n qubits in the first layer to be part of a larger quantum state, similar to the argument illustrated with the circuit in (4.4). For concreteness, let $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n'}$ be an n'-qubit quantum state with $n' \geq n$, and use the first n qubits of $|\psi\rangle$ in Proposition 4.6. Define

$$V := \left(X^{s^X_{1,m}} Z^{s^Z_{1,m}} \otimes \ldots \otimes X^{s^X_{n,m}} Z^{s^Z_{n,m}} \right) \cdot U,$$

then the quantum state at the end is given by

$$(V \otimes I) |\psi\rangle .$$

4.2.3 Proof of Universality

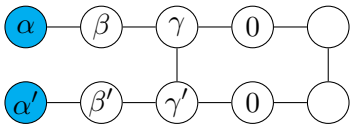
We are now ready to prove the universality of the brickwork state [BFK09]. Loosely speaking, we aim to show that for an arbitrary $n \times n$ unitary U, there exists a measurement pattern $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ that implements a unitary V, which approximates U.

To formalize the task, we fix the universal quantum gate set $S := \{\mathrm{CX}, H, T\}$, which was shown to be universal in [BMPRV00]. Any $n \times n$ unitary U can then be efficiently approximated by a quantum circuit C operating on n qubits and consisting of gates chosen from S (see Definition 2.8). Our goal is to construct a measurement pattern $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ on $\mathcal{G}_{n\times m}$ that implements the same unitary up to local Pauli operators, as these can be handled in the classical post-processing step already described in Section 4.1. Furthermore, m will scale linearly with the size of C, and we will require only measurement angles in the set Θ (as defined in Chapter 2). Note that the updated measurement angles also remain in Θ, as considering them modulo 2π does not change the measurement.

To achieve this, we first present measurement patterns on the unit cell that implement $T \otimes I$ and $H \otimes I$. We then argue that T and H can be implemented in a similar manner. For formal reasons, we also describe how to implement the identity I. Finally, we provide a measurement pattern that implements CX. With these foundational patterns established, we proceed to prove universality by demonstrating how to combine these patterns on a general brickwork state to implement any quantum circuit using S as a universal gate set.

We begin with the following lemma, which uses a notation slightly different from that described earlier in Fig. 4.1: The angle is now written inside the circle, and the measurement outcome is omitted.

Lemma 4.7. *We are given the following measurement pattern on the unit cell for arbitrary angles* $\alpha, \alpha', \beta, \beta', \gamma, \gamma' \in \mathbb{R}$*:*



By executing the updated measurement pattern, the implemented unitary corresponds, up to local Pauli operators, to

$$R_z(\gamma)R_x(\beta)R_z(\alpha) \otimes R_z(\gamma')R_x(\beta')R_z(\alpha').$$

Proof. By Proposition 4.6 and Corollary 4.2, this measurement pattern implements the unitary

$$\mathrm{CZ} \cdot (HR_z(0) \otimes HR_z(0)) \cdot (HR_z(\gamma) \otimes HR_z(\gamma')) \cdot \mathrm{CZ} \cdot (HR_z(\beta) \otimes HR_z(\beta')) \cdot (HR_z(\alpha) \otimes HR_z(\alpha')).$$

Another way to see this is by writing out the quantum circuit and repeatedly apply Corollary 4.2. This method follows exactly the same steps as those in the single-qubit case described in the example corresponding to Fig. 4.2. Using the identities in (4.8), along with the fact that

$$\mathrm{CZ}(R_z(\theta) \otimes R_z(\theta')) = (R_z(\theta) \otimes R_z(\theta'))\,\mathrm{CZ}$$

for all $\theta, \theta' \in \mathbb{R}$, as both unitaries are diagonal matrices in the standard basis, and that $\mathrm{CZ}^2 = I$, the expression can be simplified to the desired form. □

By plugging in explicit values for the angles, we can implement various unitaries, as shown in Figs. 4.7 and 4.8.

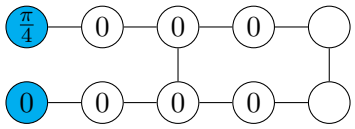


Figure 4.7: Implementation of $T \otimes I$.

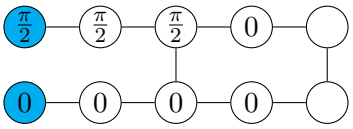


Figure 4.8: Implementation of $H \otimes I$.

The calculation for multiplying the matrices will be omitted, as it is trivial to verify. One only needs the identities $\cos(\frac{\pi}{4}) = \sin(\frac{\pi}{4}) = \frac{1}{\sqrt{2}}$ and $e^{i\pi/2} = i$. Note that, by symmetry, we also obtain $I \otimes T$ and $I \otimes H$ by switching the upper and lower angles. Furthermore, if we use the brickwork state $\mathcal{G}_{1\times 5}$ with the angles in the first row, we obviously implement the T and H unitaries. If all angles are set to 0, we then obviously implement the identity.

Lastly, we want to show that the CX unitary can be implemented as follows:

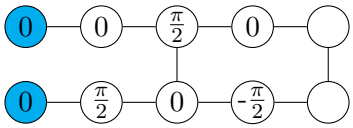
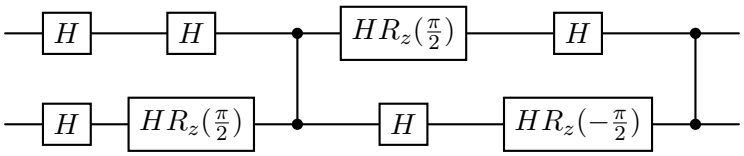


Figure 4.9: Implementation of CX.

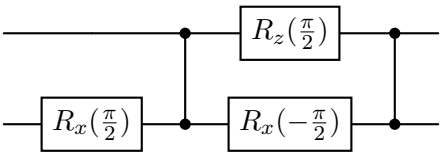
To prove this, we take use the following simple identity:

$$Z \cdot R_x(-\tfrac{\pi}{2}) \cdot Z \cdot R_x(\tfrac{\pi}{2}) = -iX, \tag{4.13}$$

which can be easily verified through direct computation. The measurement pattern implements the quantum computation described by the following quantum circuit (up to local Pauli operators):



This circuit is equivalent to:



Since global phases are irrelevant, we can replace $R_z(\frac{\pi}{2})$ with $P(\frac{\pi}{2}) = \left(\begin{smallmatrix} 1 & 0 \\ 0 & i \end{smallmatrix}\right)$. To confirm that this new circuit implements CX, we need only verify that it behaves identically to CX when acting on the standard basis states. For input $|0\rangle \otimes |b\rangle$, where $b \in \{0, 1\}$, both unitaries leave the state unchanged. For input $|1\rangle \otimes |b\rangle$, where $b \in \{0, 1\}$, the circuit maps the state to:

$$i\,|1\rangle \otimes \left(Z \cdot R_x(-\tfrac{\pi}{2}) \cdot Z \cdot R_x(\tfrac{\pi}{2})\right)|b\rangle = i\,|1\rangle \otimes -iX\,|b\rangle = |1\rangle \otimes X\,|b\rangle = \mathrm{CX}(|1\rangle \otimes |b\rangle),$$

where we used the identity from Eq. (4.13) in the first equality. Thus, the measurement pattern shown in Fig. 4.9 successfully implements CX.

We simply need to put the components together to prove universality. Note that all the measurement angles in the presented pattern lie in Θ, as considering them modulo 2π does not change the measurement.

Theorem 4.8 (Universality [BFK09, Theorem 1])**.** *The brickwork state $\mathcal{G}_{n\times m}$ is universal for quantum computation. Furthermore, we only require single-qubit measurements under angles in Θ, and measurements can be done layer-by-layer.*

Proof. Let C be a quantum circuit operating on n qubits, consisting of gates from the set $S := \{\mathrm{CX}, H, T\}$. Denote the size of C, i.e., the number of gates in C, by g. We define $m := 8g + 1$. Our goal is to construct a measurement pattern $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ on $\mathcal{G}_{n\times m}$ that implements the same unitary as C (up to local Pauli operators). Remember that the i-th row of physical qubits in the brickwork state can be viewed as representing the logical qubit in the i-th row of the quantum circuit. We now explain how to implement the gates in C sequentially on $\mathcal{G}_{n\times m}$. To simplify the procedure, we defer the gates in C such that no gates are implemented in parallel. This reordering does not affect the overall unitary transformation. For better understanding, we recommend keeping the illustration of the general brickwork state in Fig. 4.5 handy.

We start with the first gate, assuming it is a T gate or H gate acting on the i-th row. If n is even, the first five qubits in the i-th row of the brickwork state correspond to either the upper or lower part of a unit cell. In this case, the measurement patterns in Figs. 4.7 and 4.8 (or their mirrored counterparts) can be used to implement the unitary, while all other unit cells implement the identity using angles set to 0. If n is odd but $i < n$, the same method applies. For $i = n$, it suffices to use the same angles as in the first row in Fig. 4.7 or Fig. 4.8 to implement the unitary, while all other unit cells again implement the identity. Therefore, we can always implement the T and H gates using the first five layers of the brickwork state. For technical reasons, we use a measurement pattern that implements the identity (all angles are zero) for layers 5 through 9.

Now, assume the first gate is a CX gate with control on the i-th row and target on the $(i + 1)$-th row. If i is odd, the CX gate can be implemented using a unit cell in the first five

layers of the brickwork state, as shown in Fig. 4.9, while all other rows in these layers implement the identity. Additionally, the identity is implemented in layers 5 through 9. If i is even, we instead implement the identity in the first five layers and use a unit cell in layers 5 through 9 to implement the CX gate.

Our procedure ensures that the first gate can always be implemented using the first nine layers of the brickwork state. To implement the second gate, we repeat the same process, now using layers 9 through 17. These layers have the same structure as layers 1 through 9, allowing the same reasoning to apply. This procedure is repeated for all subsequent gates, leveraging the periodic structure of the brickwork state. By Corollary 4.2, this procedure implements exactly the unitary C (up to local Pauli operators).

In conclusion, the family of brickwork states can efficiently simulate any quantum circuit. Conversely, it is straightforward to see that any computation using the brickwork state can be efficiently simulated in the quantum circuit model. Thus, the two models are computationally equivalent, showing that the family of brickwork states is universal for quantum computation. □

Note that we could also add four additional layers that implement the identity at the end, to return to the setting where $m \equiv 5 \mod 8$.

Chapter 5

Half-Blind Quantum Computation

In this chapter, we present a new protocol that extends the *universal blind quantum computation* (UBQC) protocol by Broadbent, Fitzsimons, and Kashefi [BFK09]. In the UBQC protocol, the client requests the server to blindly apply an $n \times n$ unitary U to the fixed quantum state $|+\rangle^{\otimes n}$. In this section, we generalize this protocol to a setting where the unitary is blindly applied to an arbitrary quantum state held by the server, which may be entangled with some internal register of the server. More formally, we aim to blindly implement the computation $(U \otimes I)|\psi\rangle$, where $|\psi\rangle$ is an arbitrary state held by the server. Henceforth, we refer to this task as *half-blind quantum computation* (HBQC). We also provide a proof of correctness and information-theoretical blindness for this protocol. In the next section, we show how to make the interaction purely classical, thereby removing the requirement for the client to send qubits to the server. This is achieved by combining the HBQC protocol with our blind RSP protocol from Chapter 3. We refer to this task as *classical half-blind quantum computation* (CHBQC). Once again, we provide a proof of correctness and computational blindness for the protocol.

In the context of this thesis, this chapter introduces the CHBQC protocol, which serves as the core component of our compiler.

5.1 Half-Blind Quantum Computation

The concept of secure *delegated quantum computing* (DQC) protocols is motivated by highly practical considerations. It is reasonable to anticipate that, when quantum computers become available, they will be hosted by large institutions offering their services in a cloud-based model, making them accessible to a significant portion of the human population. Secure protocols are designed to enable clients with limited or no quantum technology to access the full power of quantum computers while ensuring the privacy of their information.

A parallel can be drawn to the current scenario involving supercomputers. An entire research area, known as *cloud computing*, focuses on delegating heavy computations to powerful computers owned by large companies to speed up calculations. However, there is no guarantee that these companies will not access the data sent to them; in other words, they can potentially view the computations being performed. In the classical world, we try to overcome these obstacles by using techniques such as *fully homomorphic encryption*, which allows computations to be performed on encrypted data without the need to decrypt it first.

In the quantum world, the *universal blind quantum computation* (UBQC) protocol, introduced by Broadbent, Fitzsimons, and Kashefi in [BFK09], was the first to ensure the secure delegation of quantum computations between a client and a server, assuming the client has specific, limited quantum capabilities. Without delving too deeply into the UBQC protocol, let us briefly outline how it works. Throughout this discussion, the terms 'client' and 'verifier' will be used

interchangeably, as will 'server' and 'prover'. Additionally, it is worth noting that this section heavily relies on Chapter 4.

The UBQC setting is as follows: The server is assumed, for simplicity, to have full quantum computational power, while the client can only prepare single qubits in the state

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

where $\theta \in \Theta := \{k \cdot \pi/4 \mid k = 0, \ldots, 7\}$. In the protocol, the client holds a classical description of a measurement pattern that implements an $n \times n$ unitary U on the brickwork state $\mathcal{G}_{n\times m}$. In the first phase, the client prepares $|+_\theta\rangle$ states with uniformly random $\theta \leftarrow\$ \Theta$ and sends them to the server, keeping θ secret. The server then entangles these qubits using the CZ operator according to the structure of the brickwork state $\mathcal{G}_{n\times m}$. In the second phase, the two parties interact classically. The client instructs the server to measure the qubits in specific bases, while the server reports the measurement outcomes. The bases are chosen adaptively based on the server's prior measurement outcomes. At the end of the protocol, the server holds the state $U|+\rangle^{\otimes n}$, up to local Pauli operators. Through this process, the client successfully delegates the application of the $n \times n$ unitary U to the fixed quantum state $|+\rangle^{\otimes n}$ within the MBQC model.

The most naive solution to this task would be for the server to create $\mathcal{G}_{n\times m}$, the client to send over the description of the measurement pattern, and the server to execute the pattern on $\mathcal{G}_{n\times m}$. However, in this approach, the server learns which unitary U is being applied, compromising the client's privacy. In the actual protocol, a different approach is taken, as hinted by the use of random z-rotated $|+\rangle$ states used in the brickwork state. It is crucial to note that the UBQC protocol relies on the fact that all qubits in the resource state are prepared by the client. This preparation allows the client to introduce randomness, which effectively hides the measurement angles and prevents the server from deducing information about the measurement pattern used to implement U.

In this work, however, we want the client not to operate on the fixed state $|+\rangle^{\otimes n}$, but rather on a quantum state $|\psi\rangle$ held by the server, which may be entangled with some internal register of the server. More formally, we aim to blindly implement the computation $(U \otimes I)|\psi\rangle$, where $|\psi\rangle$ is an arbitrary state held by the server. Naturally, we need to incorporate the server's qubits into the brickwork state, but this gives rise to another challenge when attempting to apply a modified version of the UBQC protocol in a naive way. Specifically, in UBQC, it is crucial that all the qubits in the brickwork state are prepared by the client to introduce randomness through z-rotations. This is no longer possible, as the qubits belonging to the server are never in the client's possession. Nevertheless, we can overcome this limitation with a simple trick. The intuition behind our solution is that by teleporting $|\psi\rangle$ into the original brickwork state prepared by the client, we achieve the same effect as applying random rotations to $|\psi\rangle$, thereby resolving the problem of the client lacking direct access to $|\psi\rangle$. A more formal description will follow.

In Chapter 4, particularly Proposition 4.6, we observed that instead of using the $|+\rangle^{\otimes n}$ state as the first layer in the brickwork state, we can use any n-qubit quantum state $|\psi\rangle$. This substitution results in $U|\psi\rangle$ (up to some local Pauli operators). This fact is well-known in the MBQC literature and has been utilized, for instance, in [MDF17] in the context of cluster states. Furthermore, if $|\psi\rangle$ consists of more than n qubits, and the first n qubits are used when executing the measurement pattern, then the overall computation is simply $(U \otimes I)|\psi\rangle$ (up to some local Pauli operators), as the MBQC procedure implements gates independently of the input state, as explained in the discussion following the proof of Proposition 4.6. Consequently, the first step is for the server to use its qubits in the first layer of the brickwork state, while the remaining qubits are prepared by the client.

The remaining challenge is to demonstrate how the client can hide his measurement angles while allowing the verifier to execute the measurement pattern. To solve this, we extend the

regular brickwork state $\mathcal{G}_{n\times m}$, to a larger brickwork state $\mathcal{G}_{n\times(m+8)}$ by introducing eight layers of dummy $|+\rangle^{\otimes n}$ states between the input layer and the remaining $m-1$ layers (see Figs. 5.1 and 5.2 for examples). The first eight layers are then measured in the 0-basis to implement the identity, while the remaining qubits are measured according to the original measurement pattern. Note that after these first eight layers, the server measures qubits that were prepared by the client with the injected randomness. In this procedure, loosely speaking, we teleport the $|\psi\rangle$ state to the point where the randomness has already been injected, achieving the same effect as directly injecting randomness into $|\psi\rangle$. While this could be achieved with just two additional layers instead of eight (as two layers suffice to implement an identity gate), using eight layers preserves the topology of the brickwork state, making the write-up more convenient.

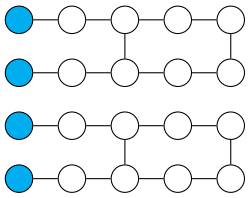


Figure 5.1: The brickwork state $\mathcal{G}_{4\times 5}$.

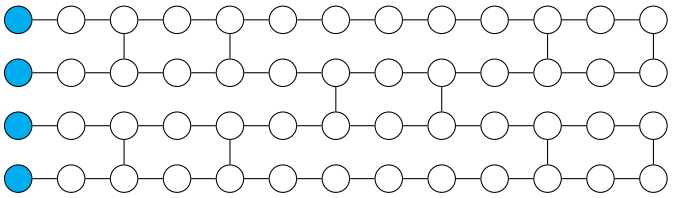


Figure 5.2: The brickwork state $\mathcal{G}_{4\times 13}$.

In the following, we formally describe our *half-blind quantum computation* (HBQC) protocol and refer the reader to Chapters 2 and 4 for background information and notational conventions.

Our HBQC Protocol. The input and output of the protocol are:

- (Input) The client V has an n-qubit unitary map U, represented as a sequence of measurement angles $\{\phi_{x,y} : \phi_{x,y} \in \Theta\}_{x\in[n],y\in[m-1]}$ of a measurement-based quantum computation over a brickwork state $\mathcal{G}_{n\times m}$.
  The server P inputs the first n qubits of a quantum state $|\psi\rangle$.
- (Output) At the end of the interaction, the client holds the measurement outcome of measuring the first n qubits of $(U\otimes I)|\psi\rangle$ in the standard basis and the server holds the post-measurement state of the remaining qubits.

For the interaction, we define $m' := m+8$ and the new measurement pattern

$$\begin{aligned} \varphi_{x,y} &:= 0 && \forall x \in [n], y \in [8] \\ \varphi_{x,y} &:= \phi_{x,y-8} && \forall x \in [n], y \in [\![9, m'-1]\!] \end{aligned}$$

for the larger brickwork state $\mathcal{G}_{n\times m'}$. The interaction between V and P proceeds as follows:

- (State Preparation)
  1. For the column $y=1$, and each row $x \in [n]$, P uses his input qubits (the first n qubits from his quantum state $|\psi\rangle$).
  2. For each column $y \in [\![2,8]\!]$, and each row $x \in [n]$, P creates qubits in the $|+\rangle$ state.
  3. For each column $y \in [\![9, m'-1]\!]$, and each row $x \in [n]$, V prepares the state $\left|+_{\theta_{x,y}}\right\rangle$, where $\theta_{x,y} \leftarrow\!\!\$\ \Theta$, and sends the qubit to P.
  4. For the column $y=m'$, and each row $x \in [n]$, P creates qubits in the $|+\rangle$ state, which are used as the final output layer.
  5. P entangles the qubits by applying CZ operators between the pairs of qubits specified by the pattern of the brickwork state $\mathcal{G}_{n\times m'}$.

- (Computation)

  For column $y = 1, \ldots, 8$:
  For row $x = 1, \ldots, n$:

  1. V computes the updated measurement angle $\varphi'_{x,y}$, to take previous measurement outcomes received from P into account.
  2. V transmits $\delta_{x,y} := \varphi'_{x,y}$ to P.
  3. P measures in the $\delta_{x,y}$-basis and transmits the result $b_{x,y} \in \{0,1\}$ to V.
  4. V sets $s_{x,y} := b_{x,y}$.

  For column $y = 9, \ldots, m' - 1$:
  For row $x = 1, \ldots, n$:

  1. V computes the updated measurement angle $\varphi'_{x,y}$, to take previous measurement outcomes received from P into account.
  2. V computes $\delta_{x,y} := \varphi'_{x,y} - \theta_{x,y} + r_{x,y}\pi$, where $r_{x,y} \leftarrow\$ \{0,1\}$, and transmits it to P.
  3. P measures in the $\delta_{x,y}$-basis and transmits the result $b_{x,y} \in \{0,1\}$ to V.
  4. V calculates $s_{x,y} := b_{x,y} \oplus r_{x,y}$.

- (Measurement)

  1. P measures the remaining n qubits in the standard basis and sends the outcome $a' \in \{0,1\}^n$ to V.
  2. V computes the actual outcome $a := \left(s^X_{1,m'} \parallel \ldots \parallel s^X_{n,m'}\right) \oplus a'$.

5.1.1 Correctness

We prove that the protocol implements the desired functionality.

Theorem 5.1. *The HBQC protocol as described above is correct, i.e., if both parties honestly follow the protocol, the output will be correct.*

Proof. The measurement pattern $\{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ implements, by definition, the unitary U. The 0-basis measurements in the first eight layers implement an identity gate (as described in Section 4.2.3), meaning that $\{\varphi_{x,y}\}_{x\in[n],y\in[m'-1]}$ still implements U.

We now argue that the added randomness in the measurement angles δ and the quantum states $|+_\theta\rangle$ cancels out during the computation, so that we still perform the same quantum computation according to the standard execution of the updated measurement pattern $\{\varphi'_{x,y}\}_{x\in[n],y\in[m'-1]}$. Note that the CZ operator commutes with both $R_z(\theta) \otimes I$ and $I \otimes R_z(\theta)$, as all are diagonal matrices in the standard basis. Thus, the state preparation phase is equivalent to first preparing the brickwork state and then applying the z-rotations to the specific qubits, rather than doing it the other way around.

Moreover, a φ'-basis measurement on a state $|\gamma\rangle$ is the same as a $(\varphi' - \theta)$-basis measurement on a state $R_z(\theta)|\gamma\rangle$, as can be verified by explicitly writing out the definition. In the protocol, we measure in the δ-basis, where $\delta := \varphi' - \theta + r\pi$. If $r = 0$, P's measurement has the same effect as V's target φ'-basis measurement; if $r = 1$, all V needs to do is flip the outcome to get again the target φ'-basis measurement, since $R_z(r\pi)$ equals Z^r up to a global phase and $Z|+_\beta\rangle = |-_\beta\rangle$ for all $\beta \in \mathbb{R}$. This shows that the protocol yields the same outcome as directly executing the updated measurement pattern $\{\varphi'_{x,y}\}_{x\in[n],y\in[m'-1]}$, without any added randomness.

Therefore, by Proposition 4.6, after the computation phase, the server holds the quantum state $(U' \otimes I)|\psi\rangle$, where

$$U' := \left(X^{s^X_{1,m'}} Z^{s^Z_{1,m'}} \otimes \ldots \otimes X^{s^X_{n,m'}} Z^{s^Z_{n,m'}} \right) U.$$

Obtaining the outcome a' by measuring the first n qubits of $(U' \otimes I)|\psi\rangle$ is equivalent to obtaining outcome a by measuring the first n qubits of $(U \otimes I)|\psi\rangle$, since the $X^{s^X_{i,m'}}$ operators only flip the bits at the corresponding positions, depending on the values of $s^X_{i,m'}$, while the $Z^{s^Z_{i,m'}}$ operators have no effect (just introducing a global phase). In other words:

$$\left(\langle a'| \otimes I\right)\left(U' \otimes I\right)|\psi\rangle = \pm \left(\langle a| \otimes I\right)\left(U \otimes I\right)|\psi\rangle .$$

This also immediately shows that the post-measurement state of the remaining qubits of $(U' \otimes I)|\psi\rangle$, given the measurement outcome a', is the same as that of the remaining qubits of $(U \otimes I)|\psi\rangle$, given the measurement outcome a, up to a global phase. □

5.1.2 Information-Theoretical Blindness

We begin by providing an intuition of what should be encapsulated by the *blindness* property before presenting a formal definition. Intuitively, for blindness, the following should hold: A malicious server should be unable to distinguish between the possible computations chosen by the client based on the information it receives during the protocol. However, it is important to note that the server does learn the dimensions of the brickwork state, (n, m), which provide an upper bound on the size of the client's computation. This information will be modeled as a leakage to the server.

To formalize this intuition, recall that any quantum adversary can be modeled as a sequence of unitaries, acting on the message registers along with an internal register containing the adversary's workspace and sufficiently-many ancillas. Thus, when defining blindness we can without loss of generality consider only the state that the adversary holds at the end of the execution. More precisely, for a given input $W = \{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ (encoded as a measurement pattern), we define $\sigma_{W,a}$ to be the (subnormalized) state held by the prover in the end of the protocol, corresponding to the output of the verifier being a, and conditioned on the input of the protocol being W. We define information-theoretical blindness in the following.

Definition 5.2 (Information-Theoretical Blindness)**.** *The HBQC protocol is* information-theoretically blind while leaking at most $L(\cdot)$*, the dimensions of the used brickwork state, if for all provers and for all possible inputs W_0 and W_1 with $L(W_0) = L(W_1)$, we have that*

$$\sum_a \sigma_{W_0,a} = \sum_a \sigma_{W_1,a}.$$

First, note that the prover can be computationally unbounded, which gives rise to the information-theoretic version of blindness. Second, observe that this definition differs from the one presented in [BFK09]. An equivalent version of their definition was provided in [FK17], which states that the protocol with input W is blind while leaking at most $L(W)$ if the distribution of messages obtained by the prover during the protocol depends only on $L(W)$. Our definition is implied by theirs, but we choose this alternative formulation as it will be more convenient to generalize to the computational setting.

Let us now prove the following helpful lemma, before moving on to prove blindness.

Lemma 5.3. *For all $\theta \in \mathbb{R}$, we have $|+_\theta\rangle\langle+_\theta| + |+_{\theta+\pi}\rangle\langle+_{\theta+\pi}| = I$.*

Proof. Follows by direct calculation:

$$\begin{aligned}|+_\theta\rangle\langle+_\theta| + |+_{\theta+\pi}\rangle\langle+_{\theta+\pi}| &= \frac{1}{2}\begin{pmatrix}1 & e^{-i\theta}\\ e^{i\theta} & 1\end{pmatrix} + \frac{1}{2}\begin{pmatrix}1 & e^{-i\theta-i\pi}\\ e^{i\theta+i\pi} & 1\end{pmatrix}\\ &= \frac{1}{2}\begin{pmatrix}1 & e^{-i\theta}\\ e^{i\theta} & 1\end{pmatrix} + \frac{1}{2}\begin{pmatrix}1 & -e^{-i\theta}\\ -e^{i\theta} & 1\end{pmatrix}\\ &= I.\end{aligned}$$

□

Theorem 5.4. *The HBQC protocol is information-theoretically blind while leaking at most the dimensions of the brickwork state.*

Proof. The proof follows the same argument as in [BFK09], up to minor syntactical adjustments. Let $W = \{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ be an arbitrary input with $L(W) = (n, m)$. Note that throughout the execution of the protocol the server receives (n, m), along with the following information

$$\{\varphi'_{x,y}\}_{x\in[n],y\in[8]}, \left\{\left|+_{\theta_{x,y}}\right\rangle, \varphi'_{x,y} - \theta_{x,y} + r_{x,y}\pi\right\}_{x\in[n],y\in[\![9,m'-1]\!]}.$$

The first tuple can be ignored for the analysis, as this information is something the server can compute on its own, since

$$\varphi'_{x,y} = (-1)^{s^X_{x,y}} \cdot \varphi_{x,y} + s^Z_{x,y} \cdot \pi = s^Z_{x,y} \cdot \pi \qquad \forall x \in [n], y \in [8]$$

and the server knows $s_{x,y} = b_{x,y}$ for all $x \in [n], y \in [8]$, hence also $s^Z_{x,y}$. We are left with

$$\left\{\left|+_{\theta_{x,y}}\right\rangle, \varphi'_{x,y} - \theta_{x,y} + r_{x,y}\pi\right\}_{x\in[n],y\in[\![9,m'-1]\!]} = \{|+_{\tau_{x,y}+r_{x,y}\pi}\rangle, \varphi'_{x,y} - \tau_{x,y}\}_{x\in[n],y\in[\![9,m'-1]\!]}$$

by defining $\tau_{x,y} := \theta_{x,y} - r_{x,y}\pi$. Now, consider $\tau_{x,y}$ to be sampled uniformly at random from Θ instead of $\theta_{x,y}$. The distribution remains unchanged.

We now argue that, from the server's perspective, each qubit is independently in the maximally mixed state, and that each angle is independently and uniformly distributed in Θ. To do this, we begin by considering the information from the last layer, i.e.,

$$\{|+_{\tau_{x,y}+r_{x,y}\pi}\rangle, \varphi'_{x,y} - \tau_{x,y}\}_{x\in[n],y=m'-1}.$$

Note that $r_{x,m'-1}$ only appears in the quantum state $\left|+_{\tau_{x,m'-1}+r_{x,m'-1}\pi}\right\rangle$, and for example, not in

$$\varphi'_{i,m'-1} = (-1)^{s^X_{i,m'-1}} \cdot \varphi_{i,m'-1} + s^Z_{i,m'-1} \cdot \pi$$

for $i \in [n]$, since only the measurement outcomes $s_{j,k} = b_{j,k} \oplus r_{j,k}$ from the previous layers appear in the formula. Thus, $r_{x,m'-1}$ for $x \in [n]$ is independent of everything else and hidden from the server, meaning the server receives the maximally mixed state $I/2$ by Lemma 5.3. Therefore, only $\varphi'_{x,m'-1} - \tau_{x,m'-1}$ depends on $\tau_{x,m'-1}$, which is then also uniformly random and independent of everything else. Consequently, the qubits in layer $y = m' - 1$ are maximally mixed, and the corresponding angles are independently uniformly distributed.

We can now inductively move on to the previous layer, say layer y_i, and apply the same reasoning, where the key observation is that r_{x,y_i} no longer depends on the angles defined in subsequent layers. To summarize, we have shown that the view of the server consists of the classical messages:

$$\{\tau^*_{x,y} : \tau^*_{x,y} \leftarrow\$\ \Theta\}_{x\in[n],y\in[\![9,m'-1]\!]}$$

and all qubits are in the maximally mixed state.

We can conclude that the view of the server is perfectly independent of W, which proves the desired implication. □

5.2 Classical Half-Blind Quantum Computation

Finally, we show how to make the verifier in the HBQC protocol completely classical, at the cost of introducing computational assumptions. We refer to this task as *classical half-blind quantum computation* (CHBQC). The protocol is identical to the one presented in Section 5.1, except for the following two modifications:

- We replace step 3 in the State Preparation phase with any blind RSP protocol that satisfies the properties outlined in Definition 3.1. For all (x, y), repeat the blind RSP protocol until it terminates successfully. Denote the verifier's output as $(t_{x,y}, \theta_{x,y})$.
- In step 2 of the Computation phase, for $y \in [\![9, m' - 1]\!]$, we instead define

$$\delta_{x,y} := \varphi'_{x,y} - (\theta_{x,y} + t_{x,y}\pi) + r_{x,y}\pi.$$

We remark that since the security parameter was introduced in the protocol, all inputs (including the brickwork dimensions) will implicitly depend on λ. However, this dependency is omitted when it is clear from the context. Furthermore, regarding termination in the first bullet point, we still terminate after a polynomial number of attempts with a probability negligibly close to 1, using a standard argument based on the Chernoff bound.

5.2.1 Correctness

Next, we show that the protocol is still correct.

Theorem 5.5. *The CHBQC protocol as described above is correct, i.e., if both parties honestly follow the protocol, the output will be correct.*

Proof. This follows directly from the correctness of both the HBQC protocol and the blind RSP protocol. Note that the RSP protocol prepares states in

$$Z^{t_{x,y}} \left|+_{\theta_{x,y}}\right\rangle = \left|+_{\theta_{x,y}+t_{x,y}\pi}\right\rangle.$$

Now, let $\theta^*_{x,y} := \theta_{x,y} + t_{x,y}\pi$ be the regular angle used in the HBQC protocol, which only appears in the above quantum state and the measurement angle $\delta_{x,y}$. The $\delta_{x,y}$ is in the CHBQC protocol also appropriately modified and so correctness follows directly from the correctness of the HBQC protocol. □

5.2.2 Computational Blindness

Before proving blindness against QPT adversaries, we present a formal definition of computational blindness. Analogous to the information-theoretic version of the definition, for a given family of inputs $W = \{W_\lambda\}_{\lambda\in\mathbb{N}}$, we denote by $\sigma^\lambda_{W,a}$ the (subnormalized) state of the prover at the end of the protocol run with security parameter λ, corresponding to the verifier's output being a_λ, and conditioned on the input of the protocol being W_λ.

Definition 5.6 (Computational Blindness)**.** *The CHBQC protocol is* computationally blind while leaking at most $L(\cdot)$*, the dimensions of the used brickwork state, if for all families of inputs $W_0 = \{W_{\lambda,0}\}_{\lambda\in\mathbb{N}}$ and $W_1 = \{W_{\lambda,1}\}_{\lambda\in\mathbb{N}}$ such that $L(W_{\lambda,0}) = L(W_{\lambda,1})$ and any family of QPT-implementable POVMs $\{M_\lambda, I - M_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function* negl *such that for all $\lambda \in \mathbb{N}$ it holds that:*

$$\left|\sum_{a_\lambda} \mathrm{tr}(\sigma^\lambda_{W_0,a} M_\lambda) - \sum_{a_\lambda} \mathrm{tr}(\sigma^\lambda_{W_1,a} M_\lambda)\right| \leq \mathsf{negl}(\lambda).$$

The definition captures the behavior that an efficient adversary cannot distinguish between the two inputs.

Theorem 5.7. *The CHBQC protocol is computationally blind while leaking at most the dimensions of the brickwork state.*

Proof. The proof follows the same structure as that of Theorem 5.4. Let $W = \{\phi_{x,y}\}_{x\in[n],y\in[m-1]}$ be an arbitrary input with $L(W) = (n, m)$. The view of the distinguisher consists of the transcript of the RSP protocol, along with the classical variables

$$\{\delta_{x,y} := \varphi'_{x,y}\}_{x\in[n],y\in[8]} \text{ and } \left\{\delta_{x,y} := \varphi'_{x,y} - (\theta_{x,y} + t_{x,y}\pi) + r_{x,y}\pi\right\}_{x\in[n],y\in[\![9,m'-1]\!]},$$

where the first tuple does not depend on W and it only depends on public information that the server has, and therefore it can be ignored.

We proceed via a hybrid argument where, starting from the last layer, we substitute each $\delta_{x,y}$ with a uniformly sampled $\delta^*_{x,y} \leftarrow\!\$\ \Theta$. To see why each hybrid is computationally indistinguishable from the previous one, it suffices to observe that

$$\begin{aligned}\delta_{x,y} &\equiv \varphi'_{x,y} - (\theta_{x,y} + t_{x,y}\pi) + r_{x,y}\pi \\ &\equiv \varphi'_{x,y} - \theta_{x,y} + r^*_{x,y}\pi \\ &\approx_c \varphi'_{x,y} - \theta^*_{x,y} + r^*_{x,y}\pi \\ &\equiv \delta^*_{x,y}\end{aligned}$$

where $r^*_{x,y} \leftarrow\!\$\ \{0, 1\}$ and $\theta^*_{x,y} \leftarrow\!\$\ \Theta$. The second equivalence follows since $r_{x,y}$ is sampled uniformly and independently of $t_{x,y}$ and thus $r_{x,y} \oplus t_{x,y} \in \{0, 1\}$ is uniformly distributed as well. The computational indistinguishability follows by the blindness of the RSP protocol.

Finally, in the last hybrid, we can see that the view of the adversary consists of some transcripts of the RSP protocol and a set of randomly sampled $\{\delta^*_{x,y}\}_{x,y}$, and in particular is perfectly independent of W. Thus, no computationally bounded distinguisher can tell apart two executions for W_0 and W_1 such that $L(W_0) = L(W_1)$, concluding our proof. □

Chapter 6

A New Compiler for Nonlocal Games

In this chapter, we present a novel *compiler* for transforming nonlocal games into an interactive protocol involving only a single computationally bounded player. We begin with a brief introduction to *nonlocal games* and take a closer look at the inner workings of the *KLVY compiler*, named after Kalai, Lombardi, Vaikuntanathan, and Yang in [KLVY23]. In the next section, we introduce our new compiler, whose core component is outlined in Section 5.2, and we provide proofs for quantum completeness and quantum soundness. Subsequently, we present a concrete example of the compilation process by applying our compiler to the famous *CHSH game* [CHSH69]. Finally, we compare our compiler with the KLVY compiler in terms of efficiency, computational assumptions, and properties.

The description of our compiler, together with the proofs for quantum completeness and quantum soundness, forms the highlight of this thesis. In comparison to the KLVY compiler, which relies on the existence of *quantum homomorphic encryption* (QHE) schemes, our compiler solely relies on the existence of plain TCFs. Moreover, our compiler is built from a black-box blind RSP protocol, satisfying Definition 3.1, which, in turn, was constructed in Section 3.3 from a black-box plain TCF. Thus, our approach provides a highly modular framework for instantiating the compiler. Furthermore, there already exist various constructions based on a variety of computational assumptions to implement TCFs, as discussed in Section 3.2. This flexibility allows our compiler to be implemented, for example, using post-quantum assumptions in isogeny-based cryptography.

6.1 Nonlocal Games

In the previous chapters, we explored some principles and applications of quantum information—a field grounded in the laws of quantum mechanics, which describe the behavior of nature at the smallest scales. The development of quantum mechanics was significantly advanced by many physicists, including Niels Bohr [Boh13], Werner Heisenberg [Hei25], Erwin Schrödinger [Sch26], Max Born [Bor26], Paul Dirac [Dir28], among others. When it first emerged, quantum mechanics was highly controversial, as it introduced deeply counterintuitive concepts, such as its inherently probabilistic nature. In 1935, physicists Albert Einstein, Boris Podolsky, and Nathan Rosen published a paper outlining the Einstein–Podolsky–Rosen (EPR) paradox [EPR35]. The paper presents a thought experiment that questions whether quantum mechanics provides a complete description of physical reality, leading to the conclusion that it should be supplemented by *hidden variables*. Loosely speaking, a *hidden-variable theory*, is a physical model that seeks to explain the probabilistic outcomes of quantum mechanics by introducing additional (possibly inaccessible) variables that predetermine the outcomes of measurements. This means that, before the measurement is performed, the result is already determined. Another term we will encounter

is the *principle of locality*, which loosely states that an object is influenced directly only by its immediate surroundings.

The mathematical implications of local hidden-variable theories in the context of quantum mechanics were explored by physicist John Stewart Bell, who, in 1964, showed that such theories satisfy a constraint now known as Bell's inequality [Bel64]. Bell further argued that the predictions of quantum mechanics violate these inequalities, showing the incompatibility between local hidden-variable theories and quantum mechanics. Subsequent work, such as that by Clauser, Horne, Shimony, and Holt (CHSH), refined Bell's theorem into a more experimentally testable form [CHSH69]. These theoretical insights were validated through experimental tests, commonly referred to as Bell tests. The first such test was conducted by Clauser and Freedman in 1972 [FC72].

Modern formulations of such experimental setups are often framed using the language of computer science and are referred to as *nonlocal games*. These games are extensively studied in quantum information theory because they provide profound insights into the foundational aspects of quantum mechanics and have practical implications for fields such as cryptography. One significant application of this framework is *Device-Independent Cryptography*, which enables cryptographic protocols, such as *Quantum Key Distribution*, that do not require trust in the internal workings of the devices involved—a concept first introduced by Mayers and Yao in [MY98]. In these protocols, one verifies that the output statistics satisfy certain properties, forcing the devices to behave in a desired manner without knowing what happens in their inner workings.

Before providing a formal mathematical definition of a nonlocal game, we will describe it in simple terms to build an intuitive understanding of the concept. Loosely speaking, a nonlocal game is a hypothetical interaction involving a referee and two spatially separated cooperating players, Alice and Bob. Each player is assigned a fixed set of possible questions and a fixed set of possible answers. The referee samples a pair of questions according to a predetermined probability distribution defined over the product space of the question sets and sends one question to each player. The players then respond with answers chosen from their respective answer sets. The referee evaluates whether the answers, in conjunction with the questions, satisfy a predefined correlation, referred to as a predicate. If they do, the players win; otherwise, they lose.

The interesting aspect of these games is that all sets, the probability distribution, and the winning condition (or predicate) are fixed and known to the players before the game begins. This allows the players to agree beforehand on a strategy for how to respond based on the questions they receive, aiming to maximize their winning probability. Additionally, they are permitted to share resources prior to the game, such as shared classical randomness or quantum resources like entangled qubits. However, no communication between the players is allowed during the game.

The analysis of such games typically involves determining the maximal winning probability under specific constraints, such as whether the players are classical or quantum—that is, whether they are restricted to sharing classical or quantum resources. The maximal winning probability is then referred to as the *classical value* or the *quantum value* of the game, depending on the resources allowed. By comparing the performance of classical and quantum strategies in these games, we will see that there are games where the quantum value is indeed strictly greater than the classical value. This enables nonlocal games to experimentally demonstrate the presence of quantum entanglement (assuming that no physical theory beyond quantum mechanics describes our reality).

In the following, we review the formal definition of nonlocal games, which we restrict to two players for simplicity, and present related quantities of interest. We will closely follow the works of [KMPSW24, CHTW04, Slo19] for this section.

Definition 6.1 (Nonlocal Game)**.** *A* (two-player) nonlocal game *is a tuple*

$$\mathcal{G} = (\mathcal{I}_A, \mathcal{I}_B, \mathcal{O}_A, \mathcal{O}_B, \mu, V),$$

which describes a game involving two non-communicating players, Alice and Bob, who interact with a referee. The sets $\mathcal{I}_A, \mathcal{I}_B, \mathcal{O}_A$, and $\mathcal{O}_B$ are finite. The elements of $\mathcal{I}_A$ (resp. $\mathcal{I}_B$) are referred to as the questions for Alice *(resp.* questions for Bob*), while the elements of $\mathcal{O}_A$ (resp. $\mathcal{O}_B$) are called the* answers of Alice *(resp.* answers of Bob*). Moreover,*

$$\mu : \mathcal{I}_A \times \mathcal{I}_B \to [0, 1]$$

is a probability distribution, and

$$V : \mathcal{O}_A \times \mathcal{O}_B \times \mathcal{I}_A \times \mathcal{I}_B \to \{0, 1\}$$

is the verification function*, also called the* predicate*. In the game, the referee samples a question pair $(x, y) \leftarrow \mu$, sending x to Alice and y to Bob. Alice and Bob then return answers $a \in \mathcal{O}_A$ and $b \in \mathcal{O}_B$, respectively. The referee evaluates $V(a, b, x, y)$ to determine the outcome: The players win if the result is 1 and lose if the result is 0.*

A k-player nonlocal game is defined similarly, simply by extending the definition in a natural way. We may also use the notation $V(a, b|x, y)$ instead of $V(a, b, x, y)$ to emphasize that this represents the value of answers a, b given questions x, y. Moreover, we emphasize that these games are described within the information-theoretical model, meaning that the players are computationally unbounded and not restricted to any specific computational model. This perspective will change when we consider compiled games in Section 6.2, as computational assumptions will be introduced to emulate spatial separation. A general nonlocal game is often illustrated as in Fig. 6.1.

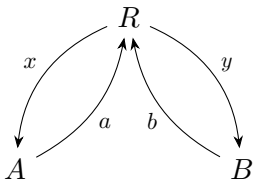


Figure 6.1: A nonlocal game played between the referee R and two players A and B.

All information about the game $\mathcal{G}$ is available to the players before the game begins, allowing them to agree on a strategy in advance. However, once the game starts, the players are not allowed to communicate. Later, we will define specific types of strategies that Alice and Bob can follow. For now, let S denote a general strategy. The key object in determining the winning probability of the game is the conditional probability distribution of the answers (a, b) given the questions (x, y), denoted by $p(a, b|x, y)$, which is implicitly determined by the strategy S. The collection $\{p(a, b|x, y)\}_{a,b,x,y} \in \mathbb{R}^{\mathcal{O}_A \times \mathcal{O}_B \times \mathcal{I}_A \times \mathcal{I}_B}$ is often referred to as a *correlation matrix*, which models the behavior of the players. The general formula for the winning probability when following strategy S in the nonlocal game $\mathcal{G}$ is given by

$$\omega(\mathcal{G}, S) := \sum_{x,y} \mu(x, y) \sum_{a,b} V(a, b|x, y) \cdot p(a, b|x, y), \tag{6.1}$$

where the first sum ranges over all possible question pairs, weighted by their probabilities of occurrence, and the second sum aggregates the probability terms for which the answers a and b allow the two players to win the game, conditioned on the questions being x and y.

We now describe the specific types of strategies that we are interested in.

Definition 6.2 (Deterministic Classical Strategy)**.** *A* deterministic classical strategy *S for a nonlocal game $\mathcal{G}$ consists of the following:*

- *A function $f : \mathcal{I}_A \to \mathcal{O}_A$.*
- *A function $g : \mathcal{I}_B \to \mathcal{O}_B$.*

Alice's answer is then given by $a := f(x)$, while Bob's answer is given by $b := g(y)$.

Thus, Alice and Bob must deterministically choose their answers based on the questions they receive. Note that this definition implicitly assumes that Alice cannot see Bob's question and vice versa. This is reasonable, as no communication during the game is allowed, which is ensured through spatial separation. In such a deterministic classical strategy S, the probability of Alice and Bob answering a and b, when receiving x and y is given by

$$p(a,b|x,y) = \begin{cases} 1 & \text{if } a = f(x) \text{ and } b = g(y), \\ 0 & \text{else.} \end{cases}$$

The winning probability can thus be expressed as

$$\omega(\mathcal{G}, S) = \sum_{x,y} \mu(x,y) \cdot V(f(x), g(y)|x,y).$$

It is also possible to define a randomized classical strategy, where the two players may use (possibly shared) randomness in their decisions. However, we will omit this definition, as introducing randomness does not provide any advantage in these games. It can be formally shown that classical randomness does not increase the winning probability. This is because a randomized classical strategy can be expressed as a convex combination of deterministic classical strategies. Consequently, the winning probability of a randomized classical strategy is upper-bounded by that of the best deterministic strategy in the convex combination. Therefore, we can restrict ourselves to examining deterministic classical strategies.

One is typically interested in the value of the game, which is the maximal winning probability achievable by the players.

Definition 6.3 (Classical Value)**.** *Let $\mathcal{G}$ be a nonlocal game, and denote the set of deterministic classical strategies for $\mathcal{G}$ by $\mathcal{S}$. Then, the* classical value *of $\mathcal{G}$ is given by*

$$\omega_c(\mathcal{G}) := \max_{S \in \mathcal{S}} \omega(\mathcal{G}, S).$$

Note that we are taking the maximum (not the supremum), as the maximum is always achievable due to the finiteness of $\mathcal{S}$.

In the next type of strategy we want to describe, the two players can behave quantumly and share quantum resources. Specifically, they can prepare a quantum state $|\psi\rangle$ of their choice before the game begins, and during the game, they can use any POVM dependent on their respective questions to measure their respective quantum systems. This can be illustrated as shown in Fig. 6.2.

Definition 6.4 (Quantum Strategy)**.** *A* quantum strategy *for a nonlocal game $\mathcal{G}$ consists of the following:*

- *Two finite-dimensional Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$.*
- *A bipartite state $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$.*

- *For every $x \in \mathcal{I}_A$, a POVM $\{A_{xa}\}_{a \in \mathcal{O}_A}$ acting on $\mathcal{H}_A$ with outcomes $a \in \mathcal{O}_A$.*
- *For every $y \in \mathcal{I}_B$, a POVM $\{B_{yb}\}_{b \in \mathcal{O}_B}$ acting on $\mathcal{H}_B$ with outcomes $b \in \mathcal{O}_B$.*

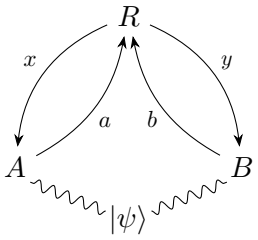


Figure 6.2: A general quantum strategy with both players sharing a state in $|\psi\rangle$.

Note that, as in the classical case, this definition implicitly accounts for spatial separation. In quantum mechanics, spatially separated subsystems are often represented by the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$ of their Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$. One can think of $\mathcal{H}_A$ as representing Alice's quantum system, on which she can operate, and $\mathcal{H}_B$ as representing Bob's quantum system. Their joint quantum state is $|\psi\rangle$. After receiving their respective questions, Alice and Bob can measure their system with respect to a POVM of their choice. In such a quantum strategy, the probability of Alice and Bob answering a and b, respectively, when receiving x and y, is given by $p(a, b|x, y) = \langle\psi| A_{xa} \otimes B_{yb} |\psi\rangle$. Furthermore, note that one could also allow mixed states in the definition, which might initially seem to make the strategy more powerful. However, using the fact that every density operator admits a purification, we can instead consider the corresponding pure state together with appropriately adjusted POVMs that produce the same probabilities. Thus, randomizing the choice of quantum state does not yield any advantage, making pure states sufficient for consideration. Moreover, when $|\psi\rangle$ is not entangled (i.e., it is separable and of the form $|\psi_1\rangle \otimes |\psi_2\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$), the strategy becomes equivalent to a randomized classical strategy. Therefore, entanglement is the feature that makes this type of strategy more powerful than classical strategies, as we will see.

The value of a nonlocal game using quantum strategies is defined similarly to the classical case.

Definition 6.5 (Quantum Value)**.** *Let $\mathcal{G}$ be a nonlocal game, and denote the set of quantum strategies for $\mathcal{G}$ by $\mathcal{S}$. Then, the* quantum value *of $\mathcal{G}$ is given by*

$$\omega_q(\mathcal{G}) := \sup_{S \in \mathcal{S}} \omega(\mathcal{G}, S).$$

Note that we are taking the supremum (not the maximum), as there are infinitely many possible strategies. In fact, there exists a nonlocal game where the quantum value is never exactly achieved by a quantum strategy [Slo19]. The intuitive explanation is that the strategy can be improved by increasing the dimensions of the spaces, enabling progressively better performance.

To conclude this section, we will examine an explicit example of a nonlocal game, including computations for its classical value and quantum value. We consider the previously mentioned famous *CHSH game* $\mathcal{G}_{\text{CHSH}}$, named after Clauser, Horne, Shimony, and Holt [CHSH69]. It is a nonlocal game in which the questions and answers are binary values, i.e.,

$$\mathcal{I}_A = \mathcal{I}_B = \mathcal{O}_A = \mathcal{O}_B = \{0, 1\},$$

the probability distribution μ is uniform

$$\mu(0,0) = \mu(0,1) = \mu(1,0) = \mu(1,1) = \frac{1}{4},$$

and the verification function V is defined by

$$V(a,b|x,y) = \begin{cases} 1 & \text{if } a \oplus b = x \cdot y, \\ 0 & \text{else.} \end{cases}$$

The setup for this nonlocal game is therefore fully specified by these declarations.

Let us now compute the classical value of the CHSH game. We begin by proposing a simple deterministic classical strategy defined via

$$\begin{aligned} f : \mathcal{I}_A &\to \mathcal{O}_A \\ x &\mapsto 0 \end{aligned}$$

and

$$\begin{aligned} g : \mathcal{I}_B &\to \mathcal{O}_B \\ y &\mapsto 0. \end{aligned}$$

In words, Alice and Bob, regardless of what they receive, always respond with 0. It is evident that the winning probability is 75%, as three out of four question pairs are answered correctly. This implies that $\omega_c(\mathcal{G}_{\text{CHSH}}) \geq 75\%$.

We now argue that no other deterministic classical strategy can achieve a higher winning probability. Assume, for the sake of contradiction, that such a strategy exists. That is, there exist functions f and g that can answer all four question pairs correctly. This would mean they satisfy the following equations:

$$\begin{aligned} f(0) \oplus g(0) &= 0 \\ f(0) \oplus g(1) &= 0 \\ f(1) \oplus g(0) &= 0 \\ f(1) \oplus g(1) &= 1. \end{aligned}$$

However, given these equalities, we can deduce

$$\begin{aligned} 1 &= 0 \oplus 0 \oplus 0 \oplus 1 \\ &= (f(0) \oplus g(0)) \oplus (f(0) \oplus g(1)) \oplus (f(1) \oplus g(0)) \oplus (f(1) \oplus g(1)) \\ &= 0, \end{aligned}$$

where the last equality follows because each value appears twice in the XOR terms. This leads to an obvious contradiction. Hence, our assumption that such functions exist is incorrect, proving that $\omega_c(\mathcal{G}_{\text{CHSH}}) \leq 75\%$. Consequently, we conclude

$$\omega_c(\mathcal{G}_{\text{CHSH}}) = 75\%.$$

The interesting part happens now, where we show that a quantum strategy exists that can outperform this 75% bound. Consider the following quantum strategy, where Alice and Bob meet beforehand and prepare an EPR pair, i.e., two qubits in the state

$$|\psi\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Alice will keep the first qubit, and Bob keeps the second. Consequently, we have the following Hilbert spaces associated with their respective quantum systems: $\mathcal{H}_A = \mathcal{H}_B = \mathbb{C}^2$. Let us now

describe how to define their POVMs with respect to the questions they receive. Recall that performing unitary transformations on a quantum system followed by a measurement in the computational basis can be modeled by a POVM; hence, we will describe their behavior in the first setting. Since we need the y-rotation $R_y(\theta)$ for this strategy, we denote the received bits by $\tilde{x}$ and $\tilde{y}$ to avoid confusion with the y in the index.

If Alice receives $\tilde{x} \in \{0,1\}$, she first applies $R_y(\tilde{x} \cdot \frac{\pi}{2})$ to her qubit and then measures it in the computational basis. If Bob receives $\tilde{y} \in \{0,1\}$, he first applies $R_y((-1)^{\tilde{y}} \cdot \frac{\pi}{4})$ to his qubit and then measures it in the computational basis.

If one goes through the straightforward calculations, one can see that the winning probability will be given by

$$\cos^2\left(\frac{\pi}{8}\right) = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 85\%.$$

This shows that using quantum strategies, we can indeed outperform the bound for classical strategies. Let us pause for a moment and think about the consequences of this. If we observe two players achieving a winning probability that exceeds 75%, we can be sure that their correlations cannot be described by classical physics. In other words, we can expect the presence of quantum entanglement in their strategy, assuming that no physical theory beyond quantum mechanics describes our reality, as such physical theories may also allow for better-performing strategies.

The above strategy henceforth leads to the inequality $\omega_q(\mathcal{G}_{\text{CHSH}}) \geq \frac{1}{2} + \frac{1}{2\sqrt{2}}$. One can, in fact, even show that no other quantum strategy can outperform this bound, i.e., $\omega_q(\mathcal{G}_{\text{CHSH}}) \leq \frac{1}{2} + \frac{1}{2\sqrt{2}}$, which is well known as *Tsirelson's bound* [Cir80]. Consequently, we conclude

$$\omega_q(\mathcal{G}_{\text{CHSH}}) = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 85\%.$$

Before moving on to the next section, we would like to briefly provide an outlook on results related to the CHSH game, which form the foundation for many cryptographic applications. However, we will only touch on the surface of these theorems, as this thesis does not aim to provide a comprehensive introduction to these results.

Earlier, we presented an optimal quantum strategy to win the CHSH game with the highest possible probability $\omega_q(\mathcal{G}_{\text{CHSH}}) = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 85\%$. An important task now is to classify optimal quantum strategies to better understand these nonlocal games from a theoretical perspective. In the CHSH game, there are, obviously, infinitely many optimal strategies because the two players can add a Hilbert space to their respective quantum systems and act trivially on it. Alternatively, they could start with a different quantum state, which can arise by applying single-qubit unitaries to both qubits of an EPR pair, allowing the players to undo these transformations during the game. Without getting too formal, one can show that these are the only two operations that can produce other optimal strategies. That is, any optimal strategy can be expressed in this way—by taking the strategy presented earlier and applying the two operations described above. This shows that optimal quantum strategies for $\mathcal{G}_{\text{CHSH}}$ are essentially unique. This property is known as the *rigidity property* of the CHSH game or, alternatively, stated as the EPR pair state being *self-tested* by playing the CHSH game. The history of self-testing goes back to Popescu and Rohrlich, who characterized the optimal quantum strategies for $\mathcal{G}_{\text{CHSH}}$ [PR92].

One can go even further by analyzing quantum strategies that are not optimal but are close to optimal. If the winning probability of a quantum strategy in $\mathcal{G}_{\text{CHSH}}$ is close to the quantum value $\omega_q(\mathcal{G}_{\text{CHSH}})$, then one can show that the strategy itself is close to one of the optimal quantum strategies presented earlier. While we will not formally define what 'closeness' means between two strategies, this closeness implies that the joint quantum state is in some sense close to the EPR pair with respect to the regular norm. This concept is referred to as the *robustness* of the CHSH game. A proof for this is given in [MYS12].

Finally, we emphasize that there are also other types of strategies commonly studied in the nonlocal game community. We will introduce one final strategy that generalizes quantum strategies and will be necessary to state one of our main results later in Section 6.3.1.

To do so, we briefly digress into functional analysis involving infinite-dimensional Hilbert spaces $\mathcal{H}$. A linear operator $f \in \mathrm{Lin}(\mathcal{H})$ is called *bounded* if there exists some $C > 0$ such that $\|f(x)\| \leq C$ for all unit-norm vectors $x \in \mathcal{H}$. We denote the set of bounded linear operators $f : \mathcal{H} \to \mathcal{H}$ by $\mathcal{B}(\mathcal{H})$. It is well known in functional analysis that the adjoint exists for every bounded linear operator. Another well-known theorem in functional analysis states that if $\mathcal{H}$ is finite-dimensional, then every linear operator $f \in \mathrm{Lin}(\mathcal{H})$ is automatically bounded; that is, $\mathcal{B}(\mathcal{H}) = \mathrm{Lin}(\mathcal{H})$. This observation allows us to extend the definition of a POVM provided earlier in Section 2.1 to the infinite-dimensional case. A POVM is defined exactly as before, with the additional assumption that the operators are bounded.

Let us now describe the next type of strategy.

Definition 6.6 (Commuting Operator Strategy)**.** *A* commuting operator strategy *for a nonlocal game $\mathcal{G}$ consists of the following:*

- *A (possibly infinite-dimensional) Hilbert space $\mathcal{H}$.*
- *A pure quantum state $|\psi\rangle \in \mathcal{H}$.*
- *For every $x \in \mathcal{I}_A$, a POVM $\{A_{xa}\}_{a \in \mathcal{O}_A}$ acting on $\mathcal{H}$ with outcomes $a \in \mathcal{O}_A$.*
- *For every $y \in \mathcal{I}_B$, a POVM $\{B_{yb}\}_{b \in \mathcal{O}_B}$ acting on $\mathcal{H}$ with outcomes $b \in \mathcal{O}_B$.*

Moreover, we require that $A_{xa}B_{yb} = B_{yb}A_{xa}$ for all $x \in \mathcal{I}_A, a \in \mathcal{O}_A, y \in \mathcal{I}_B, b \in \mathcal{O}_B$.

Here, we have $p(a, b|x, y) = \langle\psi| A_{xa}B_{yb} |\psi\rangle$. The motivation for this definition arises from the fact that the no-communication assumption can be modeled in two ways:

1. Spatially separating the players so that they act on tensor product subsystems.
2. Requiring that the players' actions commute on the joint system.

In the finite-dimensional case, these conditions are equivalent; however, in the infinite-dimensional case, they are not. This discrepancy motivates the study of commuting operator strategies. Note that every quantum strategy is also a commuting operator strategy.

The value of a nonlocal game using commuting operator strategies is defined similarly to the quantum strategy case.

Definition 6.7 (Commuting Operator Value)**.** *Let $\mathcal{G}$ be a nonlocal game, and denote the set of commuting operator strategies for $\mathcal{G}$ by $\mathcal{S}$. Then, the* commuting operator value *of $\mathcal{G}$ is given by*

$$\omega_{qc}(\mathcal{G}) := \sup_{S \in \mathcal{S}} \omega(\mathcal{G}, S).$$

Based on our previous observations, the following inequalities are evident:

$$\omega_c(\mathcal{G}) \leq \omega_q(\mathcal{G}) \leq \omega_{qc}(\mathcal{G}).$$

6.2 The KLVY Transformation

In the previous section, we formally defined nonlocal games with multiple players and a referee. In this section, we discuss how to transform nonlocal games into an interactive protocol involving a single computationally bounded player and one referee. In this context, the player and referee are typically referred to as the prover and verifier, respectively. We refer to a generic procedure for converting any nonlocal game into a single-prover protocol as a *compiler*. The first such compiler was introduced by Kalai, Lombardi, Vaikuntanathan, and Yang (KLVY) in [KLVY23].

One motivation for building such a compiler is that, in practice, the assumption that players do not communicate is difficult to enforce. In a single-prover protocol, however, this assumption is trivially satisfied. In a nonlocal game, the players are computationally unbounded, whereas in the compiled game produced by the KLVY compiler, the prover is computationally bounded. This computational assumption is important for emulating spatial separation (i.e., the no-communication assumption), which will become clearer later, when we describe the KLVY compiler.

Spatial separation is not the only reason to consider compilers. Recall from the previous section that the CHSH game provided a way to classically verify whether two players demonstrate quantum behavior by observing their winning probability. In the jargon of quantum cryptography, this is referred to as *classically testing two quantum devices*, meaning that the devices claim to be quantum, and a classical verifier can confirm this by interacting with them and observing behavior that cannot be explained classically. For example, surpassing the 75% winning probability in the CHSH game demonstrates quantum behavior, assuming no physical theory exists beyond quantum mechanics. Extending this concept to a scenario involving only one quantum device requires new protocols, which are collectively referred to as *Proofs of Quantumness*. The KLVY compiler offers such a proof of quantumness by transforming the CHSH game into a single-prover protocol [KLVY23]. In fact, this approach works for any nonlocal game where the classical value is strictly smaller than the quantum value. The reasons for this will also become clearer later. Furthermore, a subsequent work by Natarajan and Zhang [NZ23] leveraged this compiler to develop a new protocol for the *Classical Verification of Quantum Computations*.

As these applications demonstrate, compilers provide a modular framework for constructing quantum cryptographic protocols. Researchers can focus on the information-theoretic multi-player setting, which is typically simpler and well-studied, and then compile these nonlocal games into single-prover protocols. The established theorems about the compiled games will take care of the rest.

We now turn to the description of the KLVY compiler, which also provides a better understanding of our compiler presented in Section 6.3. The central component of the KLVY compiler is a so-called *quantum homomorphic encryption* scheme. For the sake of self-containedness, we present the definition here; however, the primary reason is that the author feels some pressure from the imaginary reader, who is excited to know more about it and expects the definition due to its appearance in the title of the thesis. This definition is taken from [NZ23], a subsequent work based on the original work of [KLVY23]. The reason for taking the definition from [NZ23] is that it has been rephrased using terms that already align with our notation and definitions. We emphasize that the definition is solely needed to understand the KLVY compiler and will not be used in any form for our compiler presented later.

Definition 6.8 (Quantum Homomorphic Encryption [NZ23, Definition 5])**.** *A* quantum homomorphic encryption (QHE) *scheme* $\mathsf{QHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *for a class of quantum circuits* $\mathcal{C}$ *is a tuple of algorithms with the following syntax:*

- Gen *is a PPT algorithm that takes as input the security parameter* 1^λ *and outputs a (classical) secret key* sk *of* $\mathsf{poly}(\lambda)$ *bits;*

- Enc *is a PPT algorithm that takes as input a secret key* sk *and a classical input* x*, and outputs a ciphertext* ct*;*

- Eval *is a QPT algorithm that takes as input a tuple* $(C, |\Psi\rangle, \mathsf{ct}_{\mathrm{in}})$*, where* $C : \mathcal{H} \times (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes m}$ *is a quantum circuit,* $|\Psi\rangle \in \mathcal{H}$ *is a quantum state, and* $\mathsf{ct}_{\mathrm{in}}$ *is a ciphertext corresponding to an* n*-bit plaintext.* Eval *computes a quantum circuit* $\mathsf{Eval}_C(|\Psi\rangle \otimes |0\rangle^{\otimes \mathsf{poly}(\lambda,n)}, \mathsf{ct}_{\mathrm{in}})$ *which outputs a ciphertext* $\mathsf{ct}_{\mathrm{out}}$*. If* C *has classical output, we require that* Eval_C *also has classical output.*

- Dec *is a QPT algorithm that takes as input a secret key* sk *and ciphertext* ct*, and outputs a state* $|\phi\rangle$*. Additionally, if* ct *is a classical ciphertext, the decryption algorithm outputs a classical string* y*.*

We require the following two properties from $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$*:*

- ***Correctness with auxiliary input:*** *For every security parameter* $\lambda \in \mathbb{N}$*, any quantum circuit* $C : \mathcal{H}_{\mathsf{A}} \times (\mathbb{C}^2)^{\otimes n} \to \{0,1\}^*$ *(with classical output), any quantum state* $|\Psi\rangle_{\mathsf{AB}} \in \mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}_{\mathsf{B}}$*, any message* $x \in \{0,1\}^n$*, any secret key* $\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$ *and any ciphertext* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, x)$*, the following states have negligible trace distance:*

  Game 1*. Start with* $(x, |\Psi\rangle_{\mathsf{AB}})$*. Evaluate* C *on* x *and register* A*, obtaining classical string* y*. Output* y *and the contents of register* B*.*

  Game 2*. Start with* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, x)$ *and* $|\Psi\rangle_{\mathsf{AB}}$*. Compute* $\mathsf{ct}' \leftarrow \mathsf{Eval}_C(\cdot \otimes |0\rangle^{\mathsf{poly}(\lambda,n)}, \mathsf{ct})$ *on register* A*. Compute* $y' = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}')$*. Output* y' *and the contents of register* B*.*

  In words, "correctness with auxiliary input" requires that if QHE evaluation is applied to a register A *that is a part of a joint (entangled) state in* $\mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}_{\mathsf{B}}$*, the entanglement between the QHE evaluated output and* B *is preserved.*

- ***IND-CPA security against quantum distinguishers:*** *For any two messages* x_0, x_1 *and any QPT adversary* $\mathcal{A}$*:*

$$\left| \Pr\left[\mathcal{A}^{\mathsf{Enc}_{\mathsf{sk}}(\cdot)}(\mathsf{ct}_0) = 1 \;\middle|\; \begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{sk}, x_0) \end{array} \right] - \Pr\left[\mathcal{A}^{\mathsf{Enc}_{\mathsf{sk}}(\cdot)}(\mathsf{ct}_1) = 1 \;\middle|\; \begin{array}{l} \mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{sk}, x_1) \end{array} \right] \right| \leq \mathsf{negl}(\lambda).$$

A quantum fully homomorphic encryption (QFHE) *scheme is a QHE scheme for the class of all poly-size quantum circuits.*

In simple terms, this describes an encryption scheme that allows quantum circuits to be applied to messages that are encrypted without requiring them to be decrypted first. The authors of [KLVY23] identify two schemes, presented in [Mah18a, Bra18], that satisfy their QHE definition under the assumption of the hardness of the learning with errors (LWE) problem. Notably, these two schemes even satisfy the QFHE definition.

The KLVY compiler transforms any two-player nonlocal game into a two-round protocol, where the basic idea is as follows: In the first round, the prover simulates the computation of the first player. In the second round, the prover simulates the computation of the second player. To emulate spatial separation—meaning that the answer in the second round should not depend on the first round—the question for the first player is encrypted, which then in turn uses the security of the encryption scheme. To enable the simulation of the computation in the

first round, the encryption scheme must be appropriately homomorphic, allowing the player to compute on the encrypted question. Hence, a secure QHE will heuristically handle all potential issues.

The formal description proceeds as follows. In the KLVY compiler, a quantum homomorphic encryption scheme $\mathsf{QHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ is fixed for a class of quantum circuits $\mathcal{C}$, which includes all quantum circuits the first player would have used in the nonlocal game. The compiler then transforms any two-player nonlocal game $\mathcal{G}$ into a computationally bounded single-prover interactive protocol $\mathcal{G}_{\text{comp}}$ (associated with the security parameter λ) with a classical verifier, defined as follows:

- The verifier samples two questions (x, y) from the underlying (two-player) nonlocal game, generates a secret key $\mathsf{sk} \leftarrow \mathsf{Gen}(1^\lambda)$, and sends an encryption of Alice's question $\mathsf{Enc}(\mathsf{sk}, x)$ to the prover.
- The prover responds to the verifier with an encrypted answer α, which can be thought of as $\mathsf{Enc}(a)$. In the honest case, this would homomorphically evaluate Alice's quantum circuit.
- The verifier decrypts α to recover a, then sends y to the prover in plaintext.
- The prover outputs a response b. In the honest case, this corresponds to Bob's response.
- The verifier holds a transcript (a, b, x, y) and determines whether the prover wins by evaluating the predicate of the nonlocal game.

Note that this compiled game consists of the same question sets, answer sets, question sampling probability distribution, and verification function. Moreover, the Eval algorithm from the QHE scheme is not explicitly used in the protocol itself. However, it is utilized when transforming a general strategy for a nonlocal game into a strategy for the compiled game. Notably, the compiler also works for the general case of compiling k-player nonlocal games by simply performing the same procedure for k rounds. Specifically, it encrypts the questions of the first $(k-1)$ players under different keys and proceeds as described above for each player that needs to be simulated.

To analyze such a single-prover interactive game between a PPT verifier and a prover, both of which take as input the security parameter in unary 1^λ, we adopt the definitions of the classical CS value and the quantum CS value of a single-prover protocol from [NZ23], originally derived from [KLVY23, Definition 3.1]. However, we refer to these simply as the classical value and the quantum value, which, as is clear from the context, will not be confused with the value of the corresponding nonlocal game.

Definition 6.9 ([NZ23, Definition 19])**.** *A single-prover interactive protocol G, specified by an interactive verifier Turing machine V, has classical CS value $\geq \omega$ if and only if there exists an interactive PPT Turing machine P such that for every $\lambda \in \mathbb{N}$,*

$$\Pr\Big[\langle P, V\rangle(1^\lambda) = 1\Big] \geq \omega,$$

where the probability is taken over the random coin tosses of V, and where $\langle P, V\rangle$ denotes the output bit of $V(1^\lambda)$ after interacting with P.

Definition 6.10 ([NZ23, Definition 20])**.** *A single-prover interactive protocol G, specified by an interactive verifier Turing machine V, has quantum CS value $\geq \omega^*$ if and only if there exists an interactive QPT Turing machine P such that for every $\lambda \in \mathbb{N}$,*

$$\Pr\Big[\langle P, V\rangle(1^\lambda) = 1\Big] \geq \omega^*,$$

where the probability is taken over the random coin tosses of V, and where $\langle P, V\rangle$ denotes the output bit of $V(1^\lambda)$ after interacting with P.

The work in [CMMNP+24] defined the latter quantity more concretely by specifying what a quantum strategy S for the KLVY-compiled game $\mathcal{G}_{\text{comp}}$ is, and expressed the winning probability $\omega_q(\mathcal{G}_{\text{comp}}, S)$ as a formula similarly to Eq. (6.1). We omit this definition here but use this notation as it is more concise and aligns better with the results we will present later. However, we include a similar definition tailored to the case of our compiler in Section 6.3.

One of the main results in [KLVY23] regarding the compiled game $\mathcal{G}_{\text{comp}}$ is as follows, using the above notation:

Theorem 6.11 ([KLVY23, Theorem 3.2])**.** *Let $\mathcal{G}$ be a two-player nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the compiled game under the KLVY compiler. Then the following two statements hold:*

1. *For every quantum strategy S for $\mathcal{G}$, there exists a quantum strategy S_{comp} for $\mathcal{G}_{\text{comp}}$ such that the following inequality holds*

$$\omega_q(\mathcal{G}_{\text{comp}}, S_{\text{comp}}) \geq \omega_q(\mathcal{G}, S) - \mathsf{negl}(\lambda).$$

2. *The classical value of $\mathcal{G}_{\text{comp}}$ is at most $\omega_c(\mathcal{G}) + \mathsf{negl}(\lambda)$.*

The first statement, which we refer to as *quantum completeness*, can, loosely speaking, be achieved by running the strategy S sequentially, i.e., evaluating Alice's circuit on the encrypted question using the homomorphic property of the QHE, and then performing Bob's circuit in the clear on the remaining state. The second statement, which we refer to as *classical soundness*, can, loosely speaking, be achieved by considering any single classical prover in the compiled game and constructing two provers for the nonlocal game by rewinding the classical prover. This approach crucially relies on the prover being classical rather than quantum (because of the *no-cloning theorem*). Spatial separation is then ensured by the security of the encryption scheme.

Thus, the compiler preserves the gap (if it exists) between the classical value $\omega_c(\mathcal{G})$ and the quantum value $\omega_q(\mathcal{G})$, meaning that, in the compiled game, we can also distinguish a classical prover from a quantum prover by observing the winning probability. This result enables the production of a wide variety of protocols for proofs of quantumness for a single device by compiling nonlocal games where $\omega_c(\mathcal{G}) < \omega_q(\mathcal{G})$, as noted in [KLVY23].

However, the authors of [KLVY23] did not establish any results regarding quantum soundness; that is, they could not bound the winning probability of a QPT prover in the compiled game in terms of the values of the underlying nonlocal game.

Subsequently, the work in [NZ23] established such a quantum soundness result for the compiled CHSH game, allowing them to produce a new protocol for the classical verification of BQP using a QFHE scheme as a black box. Loosely speaking, this was achieved by compiling the two-device protocol of Grilo from [Gri19] using the KLVY transformation. Furthermore, a more recent study presented a protocol with a *succinct verifier* [MNZ24], improving upon prior work [BKLMM+22], which relied on stronger cryptographic assumptions.

Further works in [CMMNP+24, BVBDM+24, MPW24] established similar results for more general classes of nonlocal games, the so-called XOR games. An XOR game is a restricted type of nonlocal game where the answers are bits, and the verification function checks whether the XOR of the answers equals some function f applied to the questions as input. The CHSH game is an example of an XOR game, where $f(x, y) := x \cdot y$.

Finally, in a recent work, [KMPSW24] established a bound on the quantum value of all compiled nonlocal games. While in the other works mentioned so far the upper bound was given by the quantum value $\omega_q(\mathcal{G})$ of the underlying nonlocal game, in this paper, the upper bound was given by the commuting operator value $\omega_{qc}(\mathcal{G})$, as defined in Definition 6.7.

6.3 A New Compiler

We now present the highlight of this thesis: Our novel compiler, which transforms nonlocal games into an interactive protocol involving a single computationally bounded player and provide proofs for quantum completeness and quantum soundness. For simplicity, we focus on the special case of two-player nonlocal games. However, our compiler can be straightforwardly adapted to k-player nonlocal games, similar to [KLVY23]. Conceptually, our compiler shares the same structure as the KLVY compiler. However, we employ distinct cryptographic primitives, which can be constructed from different cryptographic assumptions, as explained in Section 3.2.

In the KLVY compiler, the primary cryptographic primitive is a QHE scheme, which ensures a form of 'blind computation'. We extend this vague idea by replacing the QHE scheme with the CHBQC protocol introduced in Section 5.2.

To illustrate how our compiler works, let

$$\mathcal{G} = \{\mathcal{G}_\lambda\}_{\lambda\in\mathbb{N}} = \{\mathcal{I}_{\lambda,A}, \mathcal{I}_{\lambda,B}, \mathcal{O}_{\lambda,A}, \mathcal{O}_{\lambda,B}, \mu_\lambda, V_\lambda\}_{\lambda\in\mathbb{N}}$$

be a family of two-player nonlocal games indexed by the security parameter. Additionally, consider the family of unitaries

$$U = \{U_\lambda\}_{\lambda\in\mathbb{N}} = \{U_{\lambda,x}\}_{\lambda\in\mathbb{N}, x\in\mathcal{I}_{\lambda,A}},$$

where $\{U_{\lambda,x}\}_{x\in\mathcal{I}_{\lambda,A}}$ represents the unitaries corresponding to Alice's strategy for the game $\mathcal{G}_\lambda$.

To use the CHBQC protocol, these unitaries must be expressed as measurement patterns. Without loss of generality, for any given λ, we assume the measurement patterns for all $U_{\lambda,x}$ have the same size. This can be achieved by padding smaller patterns with identities (i.e., zero measurement angles) to match the size of the largest pattern. We emphasize that this measurement pattern is known to both the verifier and the prover because the prover provides it to the verifier to follow his strategy.

Given the security parameter in unary 1^λ, the prover and verifier execute the following interactive protocol:

- The verifier samples a question pair $(x, y) \leftarrow \mu_\lambda$.
- The verifier and prover engage in the CHBQC protocol. The verifier's input is $U_{\lambda,x}$, while the prover's state $|\psi\rangle$ is arbitrary. Let a' denote the prover's output, and let a be the verifier's output derived from a'.
- The verifier sends y to the prover in plaintext.
- The prover responds with some b.
- The verifier accepts if $a \in \mathcal{O}_{\lambda,A}$, $b \in \mathcal{O}_{\lambda,B}$, and $V_\lambda(a, b|x, y) = 1$.

Notably, step two is the only aspect that differs from the KLVY compiler, where we substitute the QHE scheme with our CHBQC protocol. For the nonlocal game $\mathcal{G}$, we use $\mathcal{G}_{\text{comp}}$ to denote the compiled game, even though this notation was previously used for the KLVY compiler. It is clear from the context which compiler is used. Note that the verifier is, in some sense, blindly operating on 'half' of the quantum state held by the prover, motivating the name for our HBQC protocol, as can be clearly seen in the example provided in Section 6.4.

When we say we compile a nonlocal game $\mathcal{G}$, we mean compiling a family of nonlocal games where all games are identical to $\mathcal{G}$, i.e., $\mathcal{G}_\lambda = \mathcal{G}$. We also refer to this as a *constant game*.

As in the case of nonlocal games, we will define how a general quantum strategy can be described for an efficient prover.

Definition 6.12 (QPT Strategy)**.** *A* QPT strategy *for a family of compiled games $\mathcal{G} = \{\mathcal{G}_\lambda\}_\lambda$ is a QPT algorithm $\{W_\lambda\}_\lambda$. The quantum prover behaves as follows: When receiving the question $y \in \mathcal{I}_{\lambda,B}$, the prover applies W_λ to $|y\rangle$ along with the post-measurement state of the CHBQC protocol. The prover measures a suitable number of qubits and responds with the measurement outcome b.*

We could also have used POVMs as in the definition of a quantum strategy for nonlocal games in Definition 6.4; however, we stick to this definition as the QPT assumptions are easier to state in this way. Moreover, the prover's behavior can be modeled by POVMs $\{B^\lambda_{yb}\}_{b\in\mathcal{O}_{\lambda,B}}$, where

$$B^\lambda_{yb} = (\langle b| \otimes I)W^\dagger_\lambda(|y\rangle\langle y| \otimes I)W_\lambda(|b\rangle \otimes I).$$

Lastly, for a QPT strategy S in the compiled game, we denote the winning probability, when playing the game $\mathcal{G}_\lambda$, as $\omega_\lambda(\mathcal{G}_{\text{comp}}, S)$.

6.3.1 Quantum Completeness

Now, we will prove the quantum completeness of our compiler.

Theorem 6.13 (Quantum Completeness)**.** *Let $\mathcal{G}$ be any two-player nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the compiled game under our compiler. For every quantum strategy S for $\mathcal{G}$, there exists a QPT strategy S_{comp} for $\mathcal{G}_{\text{comp}}$ such that the following inequality holds*

$$\omega_\lambda(\mathcal{G}_{\text{comp}}, S_{\text{comp}}) \geq \omega_q(\mathcal{G}, S) - \mathsf{negl}(\lambda).$$

Proof. To describe the QPT strategy S_{comp}, assume that in the nonlocal game, the quantum strategy uses the quantum state $|\psi\rangle$ along with Alice's unitaries $\{U_x\}_{x\in\mathcal{I}_A}$. In the compiled game, the single prover uses this quantum state together with these unitaries in the CHBQC protocol during step two of the game. After the CHBQC protocol, the single prover applies Bob's unitary to the remaining qubits of the original $|\psi\rangle$ state and proceeds as Bob would in the nonlocal game. If the CHBQC protocol does not abort (which happens with negligible probability, as our blind RSP protocol succeeds with probability $1 - \mathsf{negl}(\lambda)$), the output statistics of a and b of the single prover are exactly the same as in the nonlocal game, by the correctness of the CHBQC protocol. Therefore, this QPT strategy succeeds with probability at least $\omega_q(\mathcal{G}, S) - \mathsf{negl}(\lambda)$. □

We refer to Section 6.4 for a more detailed explanation of the single-prover construction.

6.3.2 Quantum Soundness

Now, we will prove the quantum soundness of our compiler.

Theorem 6.14 (Quantum Soundness)**.** *Let $\mathcal{G}$ be any two-player nonlocal game, and let $\mathcal{G}_{\text{comp}}$ be the corresponding compiled game under our compiler. Let S be any QPT strategy for $\mathcal{G}_{\text{comp}}$. Then it holds that*

$$\limsup_{\lambda\to\infty} \omega_\lambda(\mathcal{G}_{\text{comp}}, S) \leq \omega_{qc}(\mathcal{G}).$$

The quantum soundness of the KLVY compiler for constant games was established in [KMPSW24]. The statement is identical to our above theorem, except that $\mathcal{G}_{\text{comp}}$ refers to the compiled game under the KLVY transformation. To prove the above theorem, we do not need to do much since we can leverage many results from previous works in [NZ23, KMPSW24]. We only need to reprove the theorems where the security properties of the QHE scheme are used and replace them with the computational blindness properties of this compiler.

Let $\sigma^\lambda_{x,a}$ denote the (subnormalized) state of the prover after executing the CHBQC protocol with security parameter λ, corresponding to the verifier's output $a \in \mathcal{O}_{\lambda,A}$, and conditioned on the protocol's input being $x \in \mathcal{I}_{\lambda,A}$. By the computational blindness of the CHBQC protocol (Theorem 5.7), we can immediately deduce the following.

Lemma 6.15. *For all $x, x' \in \mathcal{I}_{\lambda,A}$ and any family of QPT-implementable POVMs $\{M_\lambda, I - M_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$ it holds that:*

$$\left| \sum_{a\in\mathcal{O}_{\lambda,A}} \mathrm{tr}(\sigma^\lambda_{x,a} M_\lambda) - \sum_{a\in\mathcal{O}_{\lambda,A}} \mathrm{tr}(\sigma^\lambda_{x',a} M_\lambda) \right| \leq \mathsf{negl}(\lambda). \tag{6.2}$$

For the case of constant games, quantum soundness of the compiler can be proven using the same analysis as in [KMPSW24]. The only step that has to be slightly generalized is that [NZ23, Lemma 8] has to be proven for more general states of the form

$$\sigma^\lambda_x := \sum_{a\in\mathcal{O}_{\lambda,A}} \sigma^\lambda_{x,a},$$

instead of states of the form

$$\rho^\lambda_x := \mathop{\mathbb{E}}_{c_1,\ldots,c_m=\mathrm{Enc}(x_\lambda)} \sum_{\alpha_1,\ldots,\alpha_m} (A^{c_1}_{\lambda,\alpha_1}) \otimes \cdots \otimes (A^{c_m}_{\lambda,\alpha_m}) (|\psi_\lambda\rangle\langle\psi_\lambda|)^{\otimes m} (A^{c_1}_{\lambda,\alpha_1})^\dagger \otimes \cdots \otimes (A^{c_m}_{\lambda,\alpha_m})^\dagger,$$

where A denotes Alice's POVM in the KLVY compiler (we refer to [KMPSW24] for precise definitions of these operators). This generalization will be proven now.

Lemma 6.16 ([NZ23, Lemma 8])**.** *Let $\lambda \in \mathbb{N}$ be a security parameter. For any two efficiently sampleable distributions $\{D_{\lambda,1}\}, \{D_{\lambda,2}\}$ over plaintext Alice questions, for any efficiently preparable state σ^λ_x (where σ^λ_x arises from this new compiler), and for any two-outcome measurement $\{M_\lambda, I - M_\lambda\}$ that can be implemented by a circuit with size $\mathsf{poly}(\lambda)$ acting on $m = \mathsf{poly}(\lambda)$ copies of σ^λ_x, there exists a negligible function $\mathsf{negl}(\lambda)$ such that, for all $\lambda \in \mathbb{N}$ it holds that*

$$\left| \mathop{\mathbb{E}}_{x\leftarrow D_{\lambda,1}} \mathrm{tr}((\sigma^\lambda_x)^{\otimes m} M_\lambda) - \mathop{\mathbb{E}}_{x\leftarrow D_{\lambda,2}} \mathrm{tr}((\sigma^\lambda_x)^{\otimes m} M_\lambda) \right| \leq \mathsf{negl}(\lambda). \tag{6.3}$$

Proof. Note that the statement can be reduced to Lemma 6.15 by a simple hybrid argument. Let $\{M_\lambda, I - M_\lambda\}$ be a two-outcome measurement that can be implemented by a circuit with size $\mathsf{poly}(\lambda)$ acting on m copies of σ^λ_x such that Ineq. (6.3) does not hold, i.e.

$$m_\lambda := \left| \mathop{\mathbb{E}}_{x\leftarrow D_{\lambda,1}} \mathrm{tr}((\sigma^\lambda_x)^{\otimes m} M_\lambda) - \mathop{\mathbb{E}}_{x\leftarrow D_{\lambda,2}} \mathrm{tr}((\sigma^\lambda_x)^{\otimes m} M_\lambda) \right| > \mathsf{negl}(\lambda).$$

Then we can construct a two-outcome measurement $\{N_\lambda, I - N_\lambda\}$ that can be implemented by a circuit with size $\mathsf{poly}'(\lambda)$ acting on σ^λ_x such that Ineq. (6.2) does not hold as follows. Given input σ^λ_x with $x \leftarrow D_{\lambda,1}$ or $x \leftarrow D_{\lambda,2}$, choose an index $i \in \{1, \ldots, \mathsf{poly}(\lambda)\}$ uniformly random, prepare the state $(\sigma^\lambda_{x_1})^{\otimes i-1} \otimes (\sigma^\lambda_x) \otimes (\sigma^\lambda_{x_2})^{\otimes \mathsf{poly}(\lambda)-i}$ where $x_1 \leftarrow D_{\lambda,1}$ and $x_2 \leftarrow D_{\lambda,2}$, and apply M_λ to this prepared state. Then, we have

$$\begin{aligned}
&\left| \mathop{\mathbb{E}}_{x\leftarrow D_{\lambda,1}} \mathrm{tr}(\sigma^\lambda_x N_\lambda) - \mathop{\mathbb{E}}_{x\leftarrow D_{\lambda,2}} \mathrm{tr}(\sigma^\lambda_x N_\lambda) \right| \\
&= \frac{1}{\mathsf{poly}(\lambda)} \left| \sum_{i=1}^{\mathsf{poly}(\lambda)} \mathop{\mathbb{E}}_{x_1\leftarrow D_{\lambda,1}} \mathop{\mathbb{E}}_{x_2\leftarrow D_{\lambda,2}} \mathrm{tr}((\sigma^\lambda_{x_1})^{\otimes i} \otimes (\sigma^\lambda_{x_2})^{\otimes \mathsf{poly}(\lambda)-i} M_\lambda) \right.
\end{aligned}$$

$$
\begin{aligned}
&\quad - \mathop{\mathbb{E}}_{x_1 \leftarrow D_{\lambda,1}} \mathop{\mathbb{E}}_{x_2 \leftarrow D_{\lambda,2}} \mathrm{tr}((\sigma^\lambda_{x_1})^{\otimes i-1} \otimes (\sigma^\lambda_{x_2})^{\otimes \mathsf{poly}(\lambda)-i+1} M_\lambda) \Big| \\
&= \frac{1}{\mathsf{poly}(\lambda)} \left| \mathop{\mathbb{E}}_{x \leftarrow D_{\lambda,1}} \mathrm{tr}((\sigma^\lambda_x)^{\otimes \mathsf{poly}(\lambda)} M_\lambda) - \mathop{\mathbb{E}}_{x \leftarrow D_{\lambda,2}} \mathrm{tr}((\sigma^\lambda_x)^{\otimes \mathsf{poly}(\lambda)} M_\lambda) \right| \\
&\geq \frac{1}{\mathsf{poly}^*(\lambda)}.
\end{aligned}
$$

This contradicts Lemma 6.15 for $x \leftarrow D_{\lambda,1}, x' \leftarrow D_{\lambda,2}$. □

Once this fact is established, the proofs of [NZ23, Lemma 15-17] (see also [CMMNP+24, Lemma 2.21]) follows identically. This in turn is the only result in the proof of quantum soundness [KMPSW24], where IND-CPA security of the QFHE scheme is used. By proving [NZ23, Lemma 8] for this compiler, the following proposition—and consequently, the quantum soundness of this proposed compiler for *constant* games—follows as an immediate corollary.

Proposition 6.17 ([KMPSW24, Proposition 4.6])**.** *Consider any nonlocal game $\mathcal{G}$ and a QPT strategy for the compiled game $\mathcal{G}_{\mathrm{comp}}$ (which is the same for all λ). Let $x, x' \in \mathcal{I}_A$, and let $P = P(\{B_{yb}\})$ be a polynomial in noncommuting variables $\{B_{yb}\}_{y \in \mathcal{I}_B, b \in \mathcal{O}_B}$. Then there exists a negligible function η such that, for all $\lambda \in \mathbb{N}$,*

$$
\left| \mathrm{tr}\left(\sigma^\lambda_x \, P(\{B^\lambda_{yb}\})\right) - \mathrm{tr}\left(\sigma^\lambda_{x'} \, P(\{B^\lambda_{yb}\})\right) \right| \leq \eta(\lambda),
$$

and where $\{B^\lambda_{yb}\}_{b \in \mathcal{O}_B}$ are POVMs for $y \in \mathcal{I}_B$, corresponding to the measurements that lead to the prover's second reply.

6.4 Compiling the CHSH Game

In this section, we provide a concrete transformation of our compiler applied to the famous CHSH game $\mathcal{G}_{\mathrm{CHSH}}$ described earlier in Section 6.1. Recall that the two players share an EPR pair

$$
|\psi\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),
$$

i.e., Alice has the first qubit, and Bob has the second. Alice's action is described by applying the unitary $A_{\tilde{x}} := R_y(\tilde{x} \cdot \frac{\pi}{2})$ to her qubit upon receiving $\tilde{x} \in \{0, 1\}$, followed by a measurement in the computational basis. Similarly, if Bob receives $\tilde{y} \in \{0, 1\}$, he first applies $B_{\tilde{y}} := R_y((-1)^{\tilde{y}} \cdot \frac{\pi}{4})$ to his qubit and then measures it in the computational basis.

To engage in the CHBQC protocol, we must first provide a measurement pattern for Alice's unitaries, which can be derived from the discussion of Lemma 4.7 and are illustrated in Figs. 6.3 and 6.4.

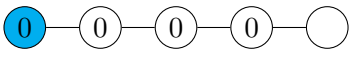


Figure 6.3: Implementation of $A_0 = I$.

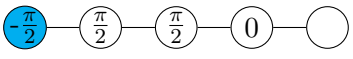


Figure 6.4: Implementation of $A_1 = R_y(\frac{\pi}{2})$.

To describe the compilation process, we will use the more 'qubit-optimized' version already mentioned in Section 5.1. In this version, instead of introducing eight additional layers, we introduce only two. This means we transition from $\mathcal{G}_{1\times 5}$ to $\mathcal{G}_{1\times 7}$ instead of $\mathcal{G}_{1\times 13}$. Every

subsequent step is essentially the same as in the 'non-optimized' version, providing the same level of insight but saving time.

We are now ready to describe the compiled CHSH game. The measurement pattern is known to both the verifier and the prover. Both parties have the security parameter in unary 1^λ as input and the prover prepares the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

- The verifier samples two questions $(\tilde{x}, \tilde{y}) \in \{0,1\}^2$ uniformly at random.
- The verifier and the prover engage in the CHBQC protocol (Section 5.2). The verifier's input is the measurement pattern of $A_{\tilde{x}}$, and the prover's input is the first qubit of his EPR pair. Executing the CHBQC protocol proceeds as follows: The prover creates two $|+\rangle$ states and then interacts with the verifier in the blind RSP protocol four times to prepare four qubits in the prover's possession. These qubits are in the state $Z^{t_i}|+_{\theta_i}\rangle = |+_{\theta_i + t_i\pi}\rangle$, where $t_i \in \{0,1\}$ and $\theta_i \in \Theta$ are known to the verifier for $1 \leq i \leq 4$. The prover then entangles these qubits according to $\mathcal{G}_{1\times 7}$, as illustrated in Fig. 6.5, where the blue circle in the figure represents the first qubit of the EPR pair.

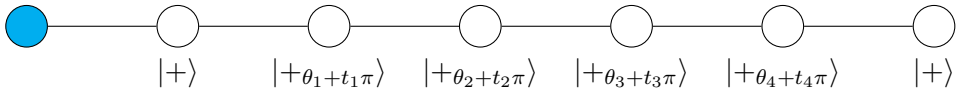


Figure 6.5: The prover entangles his qubits according to $\mathcal{G}_{1\times 7}$.

  They now proceed through the Computation phase using a measurement pattern that consists of a measurement angle of zero for the first two qubits, concatenated with the pattern of $A_{\tilde{x}}$ for the last five qubits. After the Computation phase in the CHBQC protocol, the prover holds two qubits in the state

$$\left(X^{s^X_{1,7}} Z^{s^Z_{1,7}} \otimes I\right)(A_{\tilde{x}} \otimes I)\,|\psi\rangle\,,$$

  where $s^X_{1,7} \in \{0,1\}$ and $s^Z_{1,7} \in \{0,1\}$ are known to the verifier. The prover then measures the first qubit in the computational basis, obtaining the outcome $a' \in \{0,1\}$ and sending it to the verifier. The verifier computes $a := s^X_{1,7} \oplus a'$.

- The verifier sends $\tilde{y}$ to the prover in plain.
- The prover applies $B_{\tilde{y}}$ to the second qubit, measures it in the computational basis, obtains the outcome b, and sends it to the verifier.
- Lastly, the verifier computes $V(a, b|\tilde{x}, \tilde{y})$.

This example beautifully illustrates that the verifier takes over the role of Alice (and that of the referee) by knowing which unitary operation to apply to the respective qubits, as in the original game. The second step (the second bullet point) can be conceptually understood as the prover sending the first qubit to the verifier, who then applies Alice's corresponding unitary operation and measures it, just as Alice would in the nonlocal game. At this point, the quantum state of the prover equals that in the original game. When the verifier sends the second question to the prover, the prover can apply Bob's unitary operation and respond with the same answer that Bob would provide.

6.5 Comparison

In this section, we compare our compiler with the KLVY compiler. For this analysis, we consider only constant games, i.e., $\mathcal{G}_\lambda = \mathcal{G}$ for some fixed nonlocal game $\mathcal{G}$. Let U_x denote the unitaries corresponding to Alice's strategy for the game $\mathcal{G}$, dependent on the question x. Since we are considering constant games, the measurement pattern implementing U_x is also independent of λ, i.e., the dimensions of the brickwork state remain constant with respect to λ.

We now investigate the round complexity with respect to the security parameter λ. The KLVY compiler runs in 2 rounds, independent of the security parameter, resulting in a round complexity of $O(1)$. To analyze the round complexity of our compiler, we note that the blind RSP protocol in Section 3.3 consists of six rounds with a probability of successful termination equal to $\frac{1}{64}$. Let $c > 0$ be any constant. By repeating the protocol $\lfloor \lambda^c \rfloor$ times, we can boost the success probability to $1 - \mathsf{negl}(\lambda)$. Consequently, the number of rounds increases to $O(\lambda^c)$. As the brickwork state consists of $O(1)$ qubits, the State Preparation phase in the CHBQC protocol requires $O(\lambda^c)$ rounds, while the Computation phase consists of $O(1)$ rounds. This results in an overall round complexity of $O(\lambda^c)$. Thus, the KLVY compiler is more efficient than our compiler in terms of round complexity. Since the round complexity of our compiler is determined by the round complexity of the blind RSP protocol it uses, this is the component that requires improvement.

Regarding the (classical) computational overhead of the verifier, in our compiler, it grows with the size of the (quantum) computation performed by the prover, which is, however, independent of λ. The above observation, however, indicates that the verifier might perform specific computations multiple times depending on λ. This is not a significant concern, as in the Computation phase of the CHBQC protocol, only simple arithmetic calculations are required to update the measurement angles and outcomes, which remain constant for fixed games. During the blind RSP protocol, the verifier must call Invert a total of $O(\lambda^c)$ times. Compared to the quantum computations involved, and given the current advancements in classical computing, this is likely to be a minor concern, although still important to consider. In the KLVY compiler, there is only one call to the encryption scheme Enc, which does not pose any problem, as encryption schemes are built very efficiently nowadays.

Next, we compare the number of ancilla qubits required to perform the compiled game. By ancilla qubits, we mean the additional qubits needed beyond those used to prepare the shared state $|\psi\rangle$, as both compilers are designed such that, in both transformations, the prover prepares $|\psi\rangle$. In [KLVY23], no concrete analysis is provided for general compiled games, only for the compiled CHSH game. In that context, the number of ancilla qubits is upper-bounded by the number of bits required to represent the elements in the domain and range of the underlying TCF in Mahadev's QFHE scheme, i.e., $O(\log |D| + \log |R|)$, where D is the domain and R is the range of the TCF (both implicitly dependent on λ). This is because the Mahadev QFHE scheme creates a uniform superposition over the TCF domain and evaluates the function in superposition. The same is true in our case. The only qubits required in our compiler are those for the brickwork state, which are $O(1)$, and those used in the computation within the blind RSP protocol, exactly as in the KLVY case.

We now turn to the modularity of the two compilers. As seen earlier, the KLVY compiler works for any QFHE scheme, as its results are based on a black-box QHE scheme. Our compiler uses, in some sense, two black-box cryptographic primitives: It is built from a black-box blind RSP protocol, satisfying Definition 3.1, which, in turn, is constructed in Section 3.3 from a black-box plain TCF. Thus, our approach provides a highly modular framework for instantiating the compiler. Additionally, as discussed in Section 3.2, there exist various constructions based on diverse computational assumptions to implement TCFs. This flexibility allows our compiler to

be implemented using, for instance, post-quantum assumptions in isogeny-based cryptography.

We now compare the cryptographic assumptions. The KLVY compiler relies on the existence of a QHE scheme, which must be defined over the class of quantum circuits $\mathcal{C}$ corresponding to the quantum circuits Alice would use in the two-player nonlocal game. For practical reasons, it is desirable to use a single QHE scheme for the compilation process, rather than switching schemes depending on the nonlocal game and the quantum strategy to be compiled. Thus, it is reasonable to assume that the QHE should indeed be a QFHE, which, however, makes it potentially more challenging to instantiate the compiler based on other cryptographic assumptions. While this is technically incomparable with the existence of TCFs, we can discuss concrete instances to compare the underlying computational assumptions. To the best of our knowledge, there are two approaches to building QFHE schemes (with a classical client): One assuming the hardness of the learning with errors (LWE) problem [Mah18a, Bra18], and another assuming the existence of indistinguishability obfuscation plus any dual-mode TCF [GV24]. As the latter work was published during the preparation of this thesis, the author has not explored Gupte and Vaikuntanathan's work in detail to determine whether their QFHE scheme satisfies the correctness with auxiliary input property given in Definition 6.8. This indeed requires justification, similar to how the authors of [KLVY23] had to justify that Mahadev's QFHE satisfies this property. Thus, prior to our work, compiled nonlocal games were known to exist under at most either of these two sets of assumptions. For a discussion of the cryptographic assumptions for our compiler, see Section 3.2.

Lastly, we discuss the properties that the compilers satisfy in terms of the value of the compiled game. Both compilers satisfy the properties of quantum completeness and quantum soundness. However, the KLVY compiler also satisfies the classical soundness property, which enables certain nonlocal games to be compiled into proof of quantumness protocols, as explained in Section 6.2. Whether our compiler satisfies classical soundness remains an open problem for future work.

Bibliography

[AC01] Mark Adcock and Richard Cleve. *A quantum Goldreich-Levin theorem with cryptographic applications*. 2001. arXiv: quant-ph/0108095 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0108095.

[AMR22] Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. "Candidate Trapdoor Claw-Free Functions from Group Actions with Applications to Quantum Protocols". In: *TCC 2022, Part I*. Ed. by Eike Kiltz and Vinod Vaikuntanathan. Vol. 13747. LNCS. Chicago, IL, USA: Springer, Heidelberg, Germany, Nov. 2022, pp. 266–293. DOI: 10.1007/978-3-031-22318-1_10.

[AMMW24] Yusuf Alnawakhtha, Atul Mantri, Carl A Miller, and Daochen Wang. "Lattice-based quantum advantage from rotated measurements". In: *Quantum* 8 (2024), p. 1399.

[ABCC24] Atul Singh Arora, Kishor Bharti, Alexandru Cojocaru, and Andrea Coladangelo. *A computational test of quantum contextuality, and even simpler proofs of quantumness*. 2024. arXiv: 2405.06787 [quant-ph]. URL: https://arxiv.org/abs/2405.06787.

[ADR82] Alain Aspect, Jean Dalibard, and Gérard Roger. "Experimental test of Bell's inequalities using time-varying analyzers". In: *Physical review letters* 49.25 (1982), p. 1804.

[AGR82] Alain Aspect, Philippe Grangier, and Gérard Roger. "Experimental realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: a new violation of Bell's inequalities". In: *Physical review letters* 49.2 (1982), p. 91.

[BVBDM+24] Matilde Baroni, Quoc-Huy Vu, Boris Bourdoncle, Eleni Diamanti, Damian Markham, and Ivan Šupić. "Quantum bounds for compiled XOR games and d-outcome CHSH games". In: *preprint arXiv:2403.05502* (2024).

[BKLMM+22] James Bartusek, Yael Tauman Kalai, Alex Lombardi, Fermi Ma, Giulio Malavolta, Vinod Vaikuntanathan, Thomas Vidick, and Lisa Yang. "Succinct Classical Verification of Quantum Computation". In: *CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 2022, pp. 195–211. DOI: 10.1007/978-3-031-15979-4_7.

[Bel64] John S. Bell. "On the Einstein Podolsky Rosen paradox". In: *Physics Physique Fizika* 1.3 (1964), p. 195.

[Boh13] Niels Bohr. "I. On the constitution of atoms and molecules". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 26.151 (1913), pp. 1–25.

[Bor26] Max Born. "Quantenmechanik der stoßvorgänge". In: *Zeitschrift für physik* 38.11 (1926), pp. 803–827.

[BMPRV00] P. Oscar Boykin, Tal Mor, Matthew Pulver, Vwani Roychowdhury, and Farrokh Vatan. "A new universal and fault-tolerant quantum basis". In: *Information Processing Letters* 75.3 (2000), pp. 101–107.

[Bra18] Zvika Brakerski. "Quantum FHE (Almost) As Secure As Classical". In: *CRYPTO 2018, Part III*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10993. LNCS. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 2018, pp. 67–95. DOI: 10.1007/978-3-319-96878-0_3.

[BCMVV18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. "A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device". In: *59th FOCS*. Ed. by Mikkel Thorup. Paris, France: IEEE Computer Society Press, Oct. 2018, pp. 320–331. DOI: 10.1109/FOCS.2018.00038.

[BGKPV23] Zvika Brakerski, Alexandru Gheorghiu, Gregory D. Kahanamoku-Meyer, Eitan Porat, and Thomas Vidick. "Simple Tests of Quantumness Also Certify Qubits". In: *CRYPTO 2023, Part V*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14085. LNCS. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 2023, pp. 162–191. DOI: 10.1007/978-3-031-38554-4_6.

[BKVV20] Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. *Simpler Proofs of Quantumness*. 2020. arXiv: 2005.04826 [quant-ph]. URL: https://arxiv.org/abs/2005.04826.

[BFK09] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. "Universal Blind Quantum Computation". In: *50th FOCS*. Atlanta, GA, USA: IEEE Computer Society Press, Oct. 2009, pp. 517–526. DOI: 10.1109/FOCS.2009.36.

[Cir80] Boris S. Cirel'son. "Quantum generalizations of Bell's inequality". In: *Letters in Mathematical Physics* 4 (1980), pp. 93–100.

[CHSH69] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. "Proposed experiment to test local hidden-variable theories". In: *Physical review letters* 23.15 (1969), p. 880.

[CHTW04] Richard Cleve, Peter Hoyer, Benjamin Toner, and John Watrous. "Consequences and limits of nonlocal strategies". In: *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004*. IEEE. 2004, pp. 236–249.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. "Hidden Cosets and Applications to Unclonable Cryptography". In: *CRYPTO 2021, Part I*. Ed. by Tal Malkin and Chris Peikert. Vol. 12825. LNCS. Virtual Event: Springer, Heidelberg, Germany, Aug. 2021, pp. 556–584. DOI: 10.1007/978-3-030-84242-0_20.

[CMMNP+24] David Cui, Giulio Malavolta, Arthur Mehta, Anand Natarajan, Connor Paddock, Simon Schmidt, Michael Walter, and Tina Zhang. *A Computational Tsirelson's Theorem for the Value of Compiled XOR Games*. 2024. arXiv: 2402.17301 [quant-ph]. URL: https://arxiv.org/abs/2402.17301.

[DK06] Vincent Danos and Elham Kashefi. "Determinism in the one-way model". In: *Phys. Rev. A* 74 (5 Nov. 2006), p. 052310. DOI: 10.1103/PhysRevA.74.052310. URL: https://link.aps.org/doi/10.1103/PhysRevA.74.052310.

[Dir28] Paul Adrien Maurice Dirac. "The quantum theory of the electron". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 117.778 (1928), pp. 610–624.

[EPR35] Albert Einstein, Boris Podolsky, and Nathan Rosen. "Can quantum-mechanical description of physical reality be considered complete?" In: *Physical review* 47.10 (1935), p. 777.

[FK17] Joseph F. Fitzsimons and Elham Kashefi. "Unconditionally verifiable blind quantum computation". In: *Phys. Rev. A* 96 (1 July 2017), p. 012303. DOI: 10.1103/PhysRevA.96.012303. URL: https://link.aps.org/doi/10.1103/PhysRevA.96.012303.

[FC72] Stuart J. Freedman and John F. Clauser. "Experimental test of local hidden-variable theories". In: *Physical review letters* 28.14 (1972), p. 938.

[GV19] Alexandru Gheorghiu and Thomas Vidick. "Computationally-Secure and Composable Remote State Preparation". In: *60th FOCS*. Ed. by David Zuckerman. Baltimore, MD, USA: IEEE Computer Society Press, Nov. 2019, pp. 1024–1033. DOI: 10.1109/FOCS.2019.00066.

[GMR84] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. "A "Paradoxical" Solution to the Signature Problem (Extended Abstract)". In: *25th FOCS*. Singer Island, Florida: IEEE Computer Society Press, Oct. 1984, pp. 441–448. DOI: 10.1109/SFCS.1984.715946.

[Gri19] Alex B. Grilo. "A Simple Protocol for Verifiable Delegation of Quantum Computation in One Round". In: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Ed. by Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi. Vol. 132. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 28:1–28:13. ISBN: 978-3-95977-109-2. DOI: 10.4230/LIPIcs.ICALP.2019.28. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2019.28.

[GV24] Aparna Gupte and Vinod Vaikuntanathan. "How to Construct Quantum FHE, Generically". In: *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part III*. Santa Barbara, CA, USA: Springer-Verlag, 2024, pp. 246–279. ISBN: 978-3-031-68381-7. DOI: 10.1007/978-3-031-68382-4_8. URL: https://doi.org/10.1007/978-3-031-68382-4_8.

[Hei25] Werner Heisenberg. "Quantum-theoretical re-interpretation of kinematic and mechanical relations". In: *Z. Phys* 33 (1925), pp. 879–893.

[Joz05] Richard Jozsa. *An introduction to measurement based quantum computation*. 2005. arXiv: quant-ph/0508124 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0508124.

[KCVY22] Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. "Classically verifiable quantum advantage from a computational Bell test". In: *Nature Physics* 18.8 (Aug. 2022), pp. 918–924. ISSN: 1745-2481. DOI: 10.1038/s41567-022-01643-7. URL: http://dx.doi.org/10.1038/s41567-022-01643-7.

[KLVY23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. "Quantum Advantage from Any Non-local Game". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. STOC 2023. Orlando, FL, USA: Association for Computing Machinery, 2023, pp. 1617–1628. ISBN: 9781450399135. DOI: 10.1145/3564246.3585164. URL: https://doi.org/10.1145/3564246.3585164.

[Kit95] A Yu Kitaev. "Quantum measurements and the Abelian stabilizer problem". In: *arXiv preprint quant-ph/9511026* (1995).

[KMPSW24] Alexander Kulpe, Giulio Malavolta, Connor Paddock, Simon Schmidt, and Michael Walter. *A bound on the quantum value of all compiled nonlocal games*. 2024. arXiv: 2408.06711 [quant-ph]. URL: https://arxiv.org/abs/2408.06711.

[Mah18a] Urmila Mahadev. "Classical Homomorphic Encryption for Quantum Circuits". In: *59th FOCS*. Ed. by Mikkel Thorup. Paris, France: IEEE Computer Society Press, Oct. 2018, pp. 332–338. DOI: 10.1109/FOCS.2018.00039.

[Mah18b] Urmila Mahadev. "Classical Verification of Quantum Computations". In: *59th FOCS*. Ed. by Mikkel Thorup. Paris, France: IEEE Computer Society Press, Oct. 2018, pp. 259–267. DOI: 10.1109/FOCS.2018.00033.

[MDF17] Atul Mantri, Tommaso F. Demarie, and Joseph F. Fitzsimons. "Universality of quantum computation with cluster states and (X, Y)-plane measurements". In: *Scientific reports* 7.1 (2017), p. 42861. DOI: 10.1038/srep42861.

[MY98] Dominic Mayers and Andrew Yao. "Quantum cryptography with imperfect apparatus". In: *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*. IEEE. 1998, pp. 503–509.

[MYS12] Matthew McKague, Tzyh Haur Yang, and Valerio Scarani. "Robust self-testing of the singlet". In: *Journal of Physics A: Mathematical and Theoretical* 45.45 (2012), p. 455304.

[MPW24] Arthur Mehta, Connor Paddock, and Lewis Wooltorton. "Self-testing in the compiled setting via tilted-CHSH inequalities". In: *preprint arXiv:2406.04986* (2024).

[MNZ24] Tony Metger, Anand Natarajan, and Tina Zhang. *Succinct arguments for QMA from standard assumptions via compiled nonlocal games*. 2024. arXiv: 2404.19754 [quant-ph]. URL: https://arxiv.org/abs/2404.19754.

[NZ23] Anand Natarajan and Tina Zhang. "Bounding the Quantum Value of Compiled Nonlocal Games: From CHSH to BQP Verification". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 1342–1348. DOI: 10.1109/FOCS57990.2023.00081.

[NC10] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[Nie06] Michael A. Nielsen. "Cluster-state quantum computation". In: *Reports on Mathematical Physics* 57.1 (2006), pp. 147–161. ISSN: 0034-4877. DOI: https://doi.org/10.1016/S0034-4877(06)80014-5. URL: https://www.sciencedirect.com/science/article/pii/S0034487706800145.

[Out22] Nobel Prize Outreach. *The Nobel Prize in Physics 2022*. https://www.nobelprize.org/prizes/physics/2022/summary/. 2022.

[PR92] Sandu Popescu and Daniel Rohrlich. "Which states violate Bell's inequality maximally?" In: *Physics Letters A* 169.6 (1992), pp. 411–414.

[RB01] Robert Raussendorf and Hans J. Briegel. "A One-Way Quantum Computer". In: *Phys. Rev. Lett.* 86 (22 May 2001), pp. 5188–5191. DOI: 10.1103/PhysRevLett.86.5188.

[Sch26] Erwin Schrödinger. "Quantisierung als eigenwertproblem". In: *Annalen der physik* 385.13 (1926), pp. 437–490.

[Slo19] William Slofstra. “The set of quantum correlations is not closed”. In: *Forum of Mathematics, Pi*. Vol. 7. Cambridge University Press. 2019, e1.

[Tur36] Alan M. Turing. “On computable numbers, with an application to the Entscheidungs problem”. In: *Proceedings of the London Mathematical Society Series/2 (42)* (1936), pp. 230–265.

[WJSWZ98] Gregor Weihs, Thomas Jennewein, Christoph Simon, Harald Weinfurter, and Anton Zeilinger. “Violation of Bell’s inequality under strict Einstein locality conditions”. In: *Physical Review Letters* 81.23 (1998), p. 5039.