Bit Commitment Schemes from Non-Local Games

Justin Li, School of Computer Science
McGill University, Montreal
April, 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Computer Science

Résumé

Le protocole de mise en gage est une primitive cryptographique qui agit comme un coffre-fort numérique pour les personnes mutuellement méfiantes. Normalement, la sécurité de ce protocole est liée aux hypothèses de calcul sous-jacentes. Nous nous sommes plutôt intéressés à l'exploration des protocoles de mise en gage dont la sécurité est garantie par les lois physiques. Spécifiquement, nous nous concentrons sur les protocoles de mise en gage bâtis à partir de corrélations non locales, qui sont les distributions de probabilité conjointes dérivées de la mesure de systèmes quantiques intriqués. Ces corrélations transgressent les inégalités de Bell tout en respectant la causalité relativiste qui ne permet pas les communications supraluminiques. Un jeu de pseudo-télépathie nous donne une façon intuitive de comprendre l'intrication et sa nature non locale. Dans un tel jeu, plusieurs joueurs répondent conjointement aux défis lancés par un vérificateur sans communiquer entre eux. Il n'est pas possible pour les joueurs classiques, qui ne connaissent pas les questions des autres, de gagner à tous les coups. Cependant, ceux qui partagent les intrications appropriées peuvent gagner avec probabilité 1. Dans ce travail, nous présentons un protocole pour transformer n'importe quel jeu de pseudo-télépathie en un protocole de mise en gage qui est classiquement sécuritaire. Les joueurs qui partagent les ressources intriquées peuvent briser le caractère liant de la mise en gage construite avec le protocole mentionné, tandis que les joueurs qui utilisent des stratégies strictement locales n'y parviendront pas. Cette propriété ouvre la voie à des protocoles à divulgation nulle pour les simulateurs quantiques sans avoir besoin de signaler. Nous introduisons également une nouvelle définition du caractère liant des protocoles de mise en gage que nous appelons *le jeu non-liant*.

Abstract

A bit commitment scheme is a cryptographic primitive that acts as a digital safe for mutually mistrusted parties. Classically the security of the commitment is tied to the underlying computational assumption. We are instead interested in exploring commitment schemes whose security is guaranteed by physical laws. More specifically, we focus on bit commitment schemes built using nonlocal correlations, which are the joint probability distributions derived from measuring entangled quantum systems. These correlations can violate the Bell inequalities and still respect the relativistic causality of no faster-than-light communication. A pseudo-telepathy game provides an intuitive way to understand the nonlocal nature of entanglement, where multiple non-communicating players cooperate to answer challenges given by a verifier. The game cannot be won all the time for classical players without knowing each other's inputs, but players that share the appropriate entanglements can do so. In this work, we present a protocol to transform any pseudo-telepathy game into a classically secure bit commitment scheme. Players sharing nonlocal resources can cheat the binding property of the bit commitment scheme built using this protocol, while players that use strictly local strategies will not. This property paves the way for zero-knowledge protocols for quantum simulators without the need of signalling. We also introduce a new binding definition of bit commitment schemes that we call the *non-binding game*.

Acknowledgements

I would like to thank my supervisor Claude Crépeau for his unwavering support and guidance. His intellectual curiosity and intense passion for Theory of Computer Science and Cryptography greatly influenced me and made me discovered my own love for this field.

A special thanks for Xavier for proofreading my thesis and for his insightful comments. Finally, I would like to thank my friends and family for their extraordinary and unconditional support. This work would not be possible without their weekly motivation and their constructive criticisms.

For my mother who taught me the meanings of
perseverance, sacrifices, and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> Whether you can observe a thing or not depends on the theory which you use. It is the theory which decides what can be observed.
>
> Albert Einstein

Secure cryptographic protocols are the foundation of our digital world and have enabled accelerated growth in all domains of information technology. Modern cryptography is built upon the complexity of computational assumptions. Following the advancement of modern physics in the past decades, cryptographers proposed novel cryptographic protocols that rely on physical laws instead of computationally intensive problems like prime factoring a huge number. The security of these protocols is not tied to the vulnerability of underlying computational assumptions.

Newtonian physics, or what has become known as classical mechanics, can accurately model and predict observable events for everyday large body objects. Due to its accuracy and simplicity, it is still prevalent and widely used for many engineering efforts across all fields of studies. However, as physical measurement devices became more and more refined and precise, experiments at the atomic scale produced results that did not match

with theoretical predictions. Just like the quote from Einstein at the beginning of this chapter suggests, the inconsistency can be explained by the limitation of the theory. In what is known to be the golden age of physics in the early 1900s, many brilliant physicists contributed to the birth of quantum theory which bridged the gap between unexplained natural phenomena observed by new experimental devices and its theory.

Entanglement remains to this day one of the most unintuitive component of quantum physics. Einstein, Podolsky and Rosen described the first pair of entangled photons in a thought experiment in [EPR35] in 1935, and questioned the completeness of quantum mechanics due to the ramification of the result. Two photons generated from the same source are entangled and sent in two opposite directions, far away from each other, but the measurement outcomes of entangled states are correlated [1] regardless of the distance separating them. Einstein's skepticism for quantum entanglement laid in the fact that at first glance, the correlated results can be used to transmit information at a superluminal [2] speed which violates the relativistic constraint. It took around 30 years, until 1964 when John Bell solved this apparent paradox in his seminal paper [Bel64] where he proved that no existing local hidden variable theories [3] are responsible for this physical phenomenon through the use of the famous Bell inequality test. From that point on, it has been widely accepted that quantum mechanics is the most accurate physical theory of nature at atomic and subatomic scales. The puzzling nature of entanglement was one of the key ingredients for the field of quantum computing; a field sitting at the intersection of physics, mathematics, and computer science that is more and more prevalent in this information age.

In this chapter, we will present some of the core concepts at a superficial level that leads to our main results in chapter 3. The goal is to help readers gain a high-level intuition for the important ideas before we dive into complex concepts in chapter 2. We first clarify the

1. By correlated, we mean that the variables share a statistical relationship. In the case where we measure two binary variables, a correlated result means that the measurement outcomes are either always the same, or opposite.

2. Faster than the speed of light

3. A theory that is consistent with local realism and implies that any probabilisitic outcomes of quantum mechanics are the result of underlying unobservable variables.

nonlocal nature of quantum mechanics, and then we introduce the nonlocal game which is a model for nonlocality studies. Finally, we present the bit commitment scheme, a fundamental building block of cryptography that we aim to construct using nonlocal games.

1.1 Nonlocality

Quantum nonlocality is a characteristic feature of quantum mechanics. It generally refers to the joint probability distribution of the measurement statistics of entangled multipartite quantum systems. The resultant probability correlation is nonlocal because it cannot be modelled nor explained by local hidden variable theories as demonstrated by Bell's theorem. This means that the correlated measurement outcomes of entangled systems are not a result of hidden parameter settings or other inaccessible variables but rather the nonlocal nature of the physical world. Although correlated, the measurement results of quantum systems do not permit the transmission of information, and thus do not violate the faster-than-light communication constraint of special relativity. Nevertheless, the intrinsic non-classical nature of nonlocal correlations has been identified as one of the core resources for quantum information processing. Quantum computing has seen significant breakthroughs in recent decades due to the use of entanglement, and produced practical protocols like *super dense coding* [BW92] and *quantum teleportation* [BBC$^+$93].

Following the introduction of quantum entanglement, other stronger-than-quantum nonlocal correlations have been discovered as well. These nonlocal correlations all share the common characteristic of violating the Bell inequality more than entanglement while still not permiting superluminal communication. One notable example is the Popescu-Rohrlich (PR) box (introduced in section 2.6) which maximally violates the Clauser–Horne–Shimony–Holt (CHSH) inequality [CHSH69], a generalized version of the original Bell's inequality, to the maximum algebraic sum of 4 as opposed to the Tsirelson bound of $2\sqrt{2}$ with arbitrary quantum states [PR94]. Theorists have since found a simple way to demonstrate nonlocality in the form of games which will be introduced next.

1.2 Nonlocal games

Nonlocal games naturally exhibit the power of nonlocal correlations in the form of incomplete information games where players solve a computational task cooperatively without any form of communication. This generally means that players can only know their own input from a referee and nothing else. Players can share resources and elaborate strategies beforehand, but all inputs are usually required to produce outputs that can complete the task successfully. As a consequence, classical players using deterministic strategies can only satisfy the winning conditions most of the time, but players sharing non-classical resources such as quantum entanglement have a clear advantage over their classical counterparts [BBT05, ABB$^+$10, BFS13, RV15, DSV15, CRC19]. The aforementioned PR box has a corresponding nonlocal game (CHSH game) which shows the existence of nonlocality stronger than quantum.

Two players, Alice and Bob are physically separated far away so that no information can be communicated between them for a certain time. A verifier (same as a referee) gives each of the players a single binary digit, or simply a bit, $x, y \in \{0, 1\}$, respectively. Alice and Bob then each answers in a timely fashion with a bit a, b, respectively to the verifier. The two players win the game if the following predicate is satisfied:

$$a \oplus b = x \wedge y,$$

where $\oplus$ is the logical *exclusive or* operator and $\wedge$ is the logical *and* operator. In other words, if their inputs are both 1, then their outputs have to be different, and otherwise, their outputs have to be the same. Clearly, if Alice and Bob cannot know the other's input, they cannot answer correctly every single time in the classical setting. In fact, the best classical strategy can help them win on average 75% of time. However, they can improve their winning probability by approximately 10% if they can share entangled states. Furthermore, they have a winning strategy for the CHSH game, meaning they can win the game 100%

of the time, if they share the PR box prior to the start of the game [CHTW04]. This is the first instance where the use of quantum nonlocality cannot achieve a winning strategy even if players share up to infinite entangled states. In this thesis, we restrict our focus to a subset of nonlocal games called quantum pseudo-telepathy games that have a quantum winning strategy with the use of entanglement.

1.3 Bit commitment scheme

A commitment scheme is an important cryptographic primitive that serves as a building block for more complex protocols such as zero-knowledge proofs, secret sharing and signature schemes. It is first described by Blum in the context of fair coin flipping over telephone [Blu83], and later on by Brassard and Crépeau on interactive zero knowledge proofs on NP problems [BC86]. Coin flipping over the telephone is an interesting problem where two parties that do not necessarily trust each other want to agree on the result of a coin flip announced over the telephone. On the other hand, zero knowledge proof is a more elaborate protocol such that a prover wants to convince a verifier that he possesses the answer to a problem without revealing any parts of the actual answer. These are just two of the many applications of bit commitment schemes. We can easily extract from these applications that bit commitment schemes can prove useful for untrusting parties that need to reach an agreement.

Commitment schemes allows parties to commit to a choice or a value digitally and reveal it to the other parties at a later time while being forcefully faithful to the committed value. It is called a bit commitment scheme when the committed word is a single bit $b \in \{0, 1\}$. The protocol is conducted in a way that is trustworthy for both parties. It enforces that the party committing to the message is bound to their choice and at the same time the party receiving the message cannot discover any information on the committed message prior to the revealing [DPP93, HM96]. An illustrative example of such a scheme is a locked box, where the committing party puts the intended message in the box, locks it with a key and

then gives the box to the receiving party. At a later time, the sender gives the key to the receiver and the receiver can retrieve the message without it being tampered with between the time that the box is sent and opened. This example is illustrated below in fig 1.1.

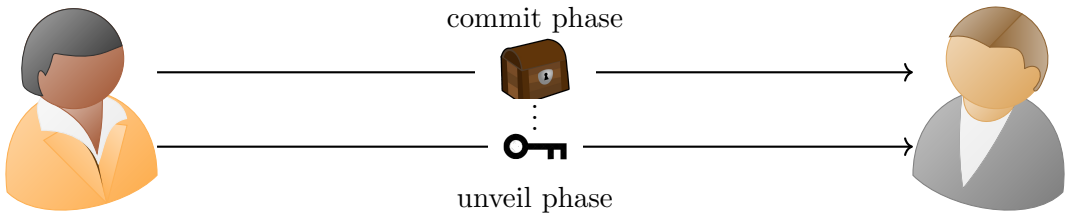


Figure 1.1 – An example of a commitment scheme using physical lock box. Alice (shown on the left) sends a locked box with a committed message concealed in it to Bob (shown on the right) during the commit phase. Later, at the unveil phase, Alice sends the key to the locked box to Bob for him to retrieve the message.

More formally, a commitment scheme is a two stage process between two parties Alice, the sender, and Bob, the receiver. During the commit phase, Alice commits to a message m by sending c, an encoded version of the original message, to Bob. At a later time, of Alice's choosing, she sends the necessary information to Bob to retrieve the original message m from c at the unveil phase. The following are the two main security properties that a commitment scheme should satisfy.

- ***hiding***: the protocol is secure against Bob if he cannot learn any information about the original message m from just the committed message c.
- ***binding***: the protocol is secure against Alice if she cannot open the commitment to more than 1 value of m.

An unconditionally and perfectly secure bit commitment scheme is known to be impossible both classically and quantumly [LC97, LC98, May97]. The argument for the classical case is an information theoretic one, and the intuition behind the idea is as follows. For the commitment scheme to be unconditionally binding, the committed message c must hold

enough information such that whenever Alice attempts to change the original message m, Bob can detect it. In other words, c can only be produced from some values of m during the commit phase, while other values of the message m' will produce a different c'. On the other hand, if the commitment is also unconditionally hiding, then c should not reveal any information about the committed message m. This means that c can be produced from any values of m. Clearly, these two criteria cannot be satisfied at the same time. Hence, the security of a commitment scheme is always modulo some computational assumptions. The argument for the quantum case is more involved, and requires some basic understanding of quantum information.

1.4 Related Work and Motivation

With advances in modern computing, the security of traditional bit commitment schemes are brought into question. This motivates the exploration of alternative strategies for constructing secure bit commitments. Many bit commitment schemes using physical phenomena have been proposed. The most promising of them all, quantum bit commitment schemes inspired by the success of quantum key distribution, were thought to be unconditionally secure [BCJL93, BC96, BC90]. However, Mayer and subsequently Lo and Chau proved that unconditional security of these protocols is impossible[LC97, LC98, May97]. Since then, other bit commitment schemes were invented using nonlocal games, such as the Popescu and Rohrlich (PR) box [BCU$^+$06], [WWW11], [AMPS16], the Greenberger-Horne-Zeilinger (GHZ) paradox [SCA$^+$11], and magic square games [CSST11]. Unlike what have been introduced in section 1.3, these bit commitment schemes are conducted in the multi-prover setting with multiple provers that want to commit to a bit value and a verifier that validates the commitment to accept or reject it. These concepts will be properly introduced in later chapters in section 2.4 and in section 3.1. Despite the subtle differences among the bit commitment schemes built using nonlocal games, they all are secure against classical provers, but provers that share the appropriate nonlocal resources can break the binding property

of these protocols. The starting point of this thesis is the magic square bit commitment scheme proposed by Crépeau, Salvail, Simard and Tapp in [CSST11].

We will define a protocol that can construct classically secure bit commitment schemes using any pseudo-telepathy game, with the additional property that quantum provers sharing the appropriate entanglements are not binding in these protocols. This effectively eliminates the underlying computational assumptions on the existing implementations of commitment schemes. It provides different level of securities in different computational models and additional insights into the application of nonlocal correlation in cryptography.

Bit commitment schemes are typically used in zero-knowledge proofs such as the famed result of [GMW91]. The bit commitment resulting from this thesis have the remarkable property that they allow their receivers to fake their unveiling as soon as they can share entanglement and perform quantum processing. Traditionally, these multi-provers bit commitments are simulated by a sole verifier (signalling to himself). However, in the context of Relativistic Zero-Knowledge [CY19] where each prover is talking to a nearby verifier, a stronger notion of zero-knowledge arises: Quantum Simulatable Zero-Knowledge. In this context, the simulators can produce their individual part of a global transcript so fast that signalling between them is made impossible while simulating. They still however manage to simulate due to their slight quantum advantage over the local provers. This property is achieved with the help of the bit commitments of this thesis: they are locally binding but non-binding to quantum simulators.

The remainder of this thesis will be organized as follows: Chapter 2 introduces the necessary mathematics and the basic notions of quantum information that are needed for this work. We will also introduce formally the framework of nonlocal games as well as some definitions and theorems related to games. In addition, we will introduce a new binding definition of a bit commitment scheme that is appropriate in a multi-provers and nonclassical setting. Chapter 3 will present the protocol for constructing bit commitment schemes using pseudo-telepathy games. Chapter 4 will discuss the security definitions and

proofs of the proposed protocol, and present concrete applications of the protocol. Finally, chapter 5 will conclude with some open problems.

Chapter 2

Preliminaries and Background

> Nature's imagination far surpasses our own
>
> Richard P. Feynman

In this chapter we will introduce the basic notations and mathematics used throughout this work. Building on that, we will also introduce the relevant subject matter for quantum information theory, and properly introduce the nonlocal game formalism which provides excellent testimony to the power of nonlocal correlations. We will describe the different settings of a nonlocal game, and also introduce a neat technique that transforms the nonlocal game and simplifies its analysis. To give it more context, we will introduce definitions and theorems relating to nonlocal games as well as the strategies that provers employ to maximize their chance of winning. Finally, we introduce the security definition of a bit commitment scheme and present the non-binding game, our new binding definition for a commitment scheme in a more general setting.

2.1 Basic notation and terminology

2.1.1 Sets and strings

We use capital English or Greek letters to denote finite and nonempty sets. For the elements of sets, we use the corresponding lower case letter. For example, if X is the set of n vertices in a graph, then $x \in X$ is one of the vertices. We denote the empty set as $\emptyset$. We use $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ and $\mathbb{C}$ to denote the sets of natural numbers (including 0), integers, real numbers and complex numbers respectively. We use the short hand notation $[n]$ to denote the set of natural numbers $\{1, 2, \ldots, n\}$, and $t \in [n]$ means t is an element of the set $[n]$. We assume the reader is familiar with elementary set operations.

In computer science, a *bit* is a unit of information that is either 0 or 1. We typically work with more than one bit in which case we call it a bit string.

Definition 2.1.1 (bit string)**.** *A bit string of length $n \in \mathbb{N}$ is a sequence of n bits, and is an element of $\{0,1\}^n$. We also denote the set of all bit strings of finite length as $\{0,1\}^*$, where*

$$\{0,1\}^* \stackrel{\text{def}}{=} \bigcup_{n \in \mathbb{N}} \{0,1\}^n.$$

For any bit string $s \in \{0,1\}^*$, the length of s, or the number of bits in s, is denoted as $|s|$.

We define below a binary operation that we will see often throughout this work.

Definition 2.1.2 (Exclusive or)**.** *The exclusive-or, known also as "xor", is a logical operation on two binary variables a, b such that it evaluates to true if and only if they differ $(a \neq b)$. Another way to evaluate it with only logical operators is the following*

$$a \oplus b \stackrel{\text{def}}{=} (a \vee b) \wedge \neg(a \wedge b).$$

This is equivalent to addition modulo 2*.*

2.2 Vector spaces

A vector space is composed of basic objects called vectors, which themselves can be composed of scalars from $\mathbb{R}$ or $\mathbb{C}$. We assume here that all vector spaces in this section are finite and over $\mathbb{C}$ which is referred to as the complex vector space. We use $\mathbb{C}^n$ to denote the space of all vectors v with n complex numbers which can be expressed as $(v_1, v_2, \ldots, v_n)$ or more commonly

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

Definition 2.2.1 (inner product)**.** *For some $n \in \mathbb{N}$, and any vectors $u, v \in \mathbb{C}^n$, we define the inner product of u, v as*

$$\langle u, v \rangle \stackrel{\text{def}}{=} \sum_{i=1}^{n} u_i^* v_i, \tag{2.1}$$

where u_i^* is the complex conjugate of the i^{th} element of vector u. The complex conjugate of a complex number $c = a + \imath b$ is simply $c^* = a - \imath b$ with $a, b \in \mathbb{R}$, with $\imath \stackrel{\text{def}}{=} \sqrt{-1}$.

A complex vector space with a map of inner product is called an *inner product space*, and more commonly referred to as *Hilbert space* in quantum mechanics. Each pair of vectors in the Hilbert space is associated with a complex number which is their inner product. This vector space is especially significant to our study because all quantum computations occur within it.

Definition 2.2.2 (norm)**.** *We define the norm of a vector $v \in \mathbb{C}^n$ by*

$$||v|| \stackrel{\text{def}}{=} \sqrt{\langle v, v \rangle}. \tag{2.2}$$

A vector u is called a *unit vector* if $||u|| = 1$. We also say that u is normalized if it has unit norm. We can simply normalize a vector u by doing $u/||u||$. We say two vectors $u, v \in \mathbb{C}^n$ are *orthogonal* if and only if $\langle u, v \rangle = 0$, and they are *orthonormal* if they are also

unit vectors.

2.2.1 Dirac notation and linear operator

We will now present the *Dirac notation*, also known as the *bra-ket notation*, which is commonly used in quantum information theory. A vector represented in this notation is

$$|\psi\rangle\,,$$

where ψ is a unit vector in Hilbert space and the symbol $|\cdot\rangle$ is called a *ket*. Similarly, we define the *bra*, identified with the symbol

$$\langle\cdot|$$

to be the dual element of $|\cdot\rangle$. For any vector $|\psi\rangle$, $\langle\psi|$ is its *dual vector* and is the conjugate transpose, denoted †, of $|\psi\rangle$ with each of its elements being its complex conjugate. This means

$$\langle\psi| \stackrel{\text{def}}{=} |\psi\rangle^\dagger = \begin{bmatrix}\psi_1^* & \dots & \psi_n^*\end{bmatrix}. \tag{2.3}$$

With this, the inner product of vectors $|\psi\rangle$ and $|\phi\rangle$ is denoted as

$$\langle\psi|\phi\rangle \stackrel{\text{def}}{=} \sum_i \psi_i^*\phi_i = \begin{bmatrix}\psi_1^* & \dots & \psi_n^*\end{bmatrix}\begin{bmatrix}\phi_1\\ \vdots \\ \phi_n\end{bmatrix}.$$

Notice that the inner product is just the bra of a vector multiplied by the ket of another vector, which is where the name "bra-ket" comes from. The norm of $|\psi\rangle$ is simply $||\,|\psi\rangle\,|| = \sqrt{\langle\psi|\psi\rangle}$. We now introduce another linear operation called the *outer product* which when performed on two vectors of dimension $n \times 1$ produces a matrix of dimension $n \times n$.

Definition 2.2.3 (outer product)**.** *The outer product of two vectors* $|\psi\rangle\,, |\phi\rangle \in \mathbb{C}^n$ *is defined*

as

$$|\psi\rangle\langle\phi| \stackrel{\text{def}}{=} \begin{bmatrix}\psi_1\\ \vdots \\ \psi_n\end{bmatrix}\begin{bmatrix}\phi_1^* & \dots & \phi_n^*\end{bmatrix} = \begin{bmatrix}\psi_1\phi_1^* & \dots & \psi_1\phi_n^* \\ \vdots & \ddots & \vdots \\ \psi_n\phi_1^* & \dots & \psi_n\phi_n^*\end{bmatrix}. \tag{2.4}$$

The outer product is a very useful way to represent *linear operators*. Suppose we have two Hilbert spaces V, W and their corresponding vectors $|v\rangle$ and $|w\rangle$. We define $|w\rangle\langle v|$ to be the linear operator from V to W which when performed on a vector $|\psi\rangle$ can be represented as

$$(|w\rangle\langle v|)|\psi\rangle = |w\rangle\langle v|\psi\rangle = \langle v|\psi\rangle|w\rangle.$$

The last part of the operation is allowed because of the scalar produced by $\langle v|\psi\rangle$ and the linearity of vector multiplication. We introduce linear operators more formally below.

Definition 2.2.4 (linear operator)**.** *For any vector spaces V, W, a linear operator between these two vector spaces is any function $A : V \to W$ such that A is linear in its inputs,*

$$A|v\rangle = A\left(\sum_i c_i|v_i\rangle\right) = \sum_i c_i A(|v_i\rangle).$$

We use $\mathcal{L}(V, W)$ to denote the set of all linear operators that maps from vector space V to W, and $\mathcal{L}(V, V) = \mathcal{L}(V)$.

A most convenient way to represent linear operator is by its matrix form. Suppose A is a $m \times n$ matrix with entries a_{ij} then, A is a linear operator that maps vectors in vector space $\mathbb{C}^n$ to vector space $\mathbb{C}^m$. One of the most important linear operators is the *identity operator* $\mathbb{1}$ which satisfies $\mathbb{1}|\psi\rangle = |\psi\rangle$, and can be represented as the identity matrix.

2.3 Quantum Information Theory

As mentioned in the introduction, quantum mechanics is a mathematical framework for the development of physical theories that overcomes what was lacking in the classical ones.

A series of experiments produced results that were not predicted by classical theories. Most notably, the Stern-Gerlach experiment is one of the first experiments that made physicists question the validity of classical mechanics. We will not cover such experiments and why the classical predictions did not match with the experimental data. Instead, we will introduce the core concepts of quantum information theory from the perspective of a computer science theorist. For readers that are interested in the exciting history of the development of quantum mechanics, [NC11] and [Wil13] provide an excellent overview of this.

2.3.1 Qubits

The fundamental unit of measurement in classical computer science is a binary digit, also known as a *bit*, which can contain either a 0 to signify *false* or a 1 to signify *true* in mathematical logic. A classical bit can represent a two state system such as the "on/off" of a switch, or whether or not a transistor allows the electric current to flow. The quantum analog of a classical bit is called a quantum bit or *qubit* for short. It represents any fundamental two-level quantum system. For example, it can model the spin of an electron, the polarization of a photon, or the excited state and the ground state of an atom.

A qubit is the simplest quantum mechanical system and lives in a two-dimensional Hilbert space $\mathcal{H}^2$. In mathematical form, any arbitrary pure qubit can be represented by the linear combination of orthonormal basis $|0\rangle, |1\rangle$ as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle , \tag{2.5}$$

where α and β are complex numbers and subject to

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.6}$$

The complex numbers α, β can be thought of as the probability amplitude of the associated states. The qubit $|\psi\rangle$ in the above equation has probability $|\alpha|^2$ to be in state $|0\rangle$ and

probability $|\beta|^2$ to be in state $|1\rangle$ once measured. For computational purposes, it is often easier to think of states in terms of their vector representation. For example the qubit $|\psi\rangle$ from above can be expressed as,

$$|\psi\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

where $|0\rangle = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, and $|1\rangle = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. The special $|0\rangle$ and $|1\rangle$ states are the classical counterparts of 0 and 1 states. They are generally referred to as the *computational basis*. A qubit can be expressed as the linear combination of any other pairs of orthonormal vectors.

The main difference between a classical bit and a qubit is that a qubit can also be in any state other than $|0\rangle$ or $|1\rangle$ called a *superposition* of states. In principle, there exists infinite linear combinations of states, however, one cannot extract an infinite amount of information from a qubit. When a qubit expressed in terms of computational basis is measured, the state of the qubit will collapse from its superposition of $|0\rangle$ and $|1\rangle$ to the state that is consistent with the classical result of 0 or 1. Moreover, any other further measurement of the qubit will yield the same result. This means that a scientist can never fully quantify the complex amplitudes α and β of a single qubit, unless there is a large amount of identical qubits. Despite the fact that we only learn one classical bit of information when measuring a qubit, the true advantage of quantum computing as opposed to the classical one will become apparent through the use of quantum gates and entanglement.

2.3.2 Unitary transformation

Quantum gates are reversible linear operators that are called *unitary transformations*, and describe the evolution of a closed quantum system. A unitary transformation does not leak any classical information and is reversible. It can be represented mathematically as a unitary matrix U that satisfies the following

$$U^\dagger U = U U^\dagger = \mathbb{1}, \tag{2.7}$$

where the dimension of the identity matrix $\mathbb{1}$ and that of the unitary matrix U are the same. Some of the most important unitary transformations are the Pauli operators

$$X \stackrel{\text{def}}{=} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}, \quad Z \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{2.8}$$

where the Pauli-X gate is the quantum analog of the *NOT* gate. Another single-qubit quantum gate that we will see very often is the *Hadamard* gate

$$H \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.9}$$

Applying a quantum gate on a qubit is as simple as applying the unitary matrix to the vector representation of a qubit. For example, the Hadamard gate on qubit $|0\rangle$ yields

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}},$$

and similarly, $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. We denote $|+\rangle = H|0\rangle$, and $|-\rangle = H|1\rangle$. The pair $\{|+\rangle, |-\rangle\}$ forms an orthonormal basis, and is referred to as the *diagonal basis* or simply the *Hadamard basis*.

So far, we have only seen single qubit states and quantum gates, but the same ideas apply for the multiple qubit states and their transformation through the use of tensor product. Although called tensor product, what we use in quantum computation is *Kronecker product*, a small variant of the former. There is an entire branch of mathematics that studies tensors and their operations. We will only introduce tensor product in an operational sense.

Definition 2.3.1 (tensor product)**.** *Suppose we have two matrices A of dimension $m \times n$ and B of dimension $p \times q$, then the matrix representation of their tensor product is as*

follows.

$$A \otimes B \stackrel{\text{def}}{=} \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}B & A_{n1}B & \dots & A_{mn}B \end{bmatrix}, \tag{2.10}$$

where $A_{ij}B$ is of dimension $p \times q$. Hence this lives in a Hilbert space of dimension $nq \times mp$.

For example, the two qubit state $|00\rangle$ is simply the tensor product of $|0\rangle$ and $|0\rangle$,

$$|00\rangle \stackrel{\text{def}}{=} |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The tensor product of the Pauli-X gate and the Hadamard gate is then

$$X \otimes H = \begin{bmatrix} 0H & 1H \\ 1H & 0H \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}.$$

One of the most important two-qubit gates is the *controlled-not gate* or *CNOT gate* in short, which is represented as

$$CNOT \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.11}$$

This is the quantum analog of the classical *XOR gate*, and is responsible for most of the nonlocal results of quantum computation along with the Hadamard gate as shown in the next sections. Finally, it is worth mentioning that $H^{\otimes n}$ is a short hand notation for the Hadamard gate tensored with itself n times.

2.3.3 Measurement

In quantum mechanics, the measurement of a quantum system is described by a collection $\{M_i\}_{i\in I}$ of *measurement operators*. The index i indicates the measurement outcomes that may occur. The measurement operators satisfy

$$\sum_i M_i^\dagger M_i = \mathbb{1}, \tag{2.12}$$

which expresses the fact that the probabilities of measurement outcomes sum to one. For a state $|\psi\rangle$, the probability that the result i occurs after measuring is expressed as

$$p_i = \langle\psi|M_i^\dagger M_i|\psi\rangle, \tag{2.13}$$

and when the outcome is i, the resultant quantum system collapses to state

$$\frac{M_i\,|\psi\rangle}{\sqrt{p_i}}. \tag{2.14}$$

When the qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, is expressed in computational basis, the measurement operators are $M_0 = |0\rangle\langle 0|$, and $M_1 = |1\rangle\langle 1|$. If the result is $|0\rangle$, then the probability is

$$p_0 = \langle\psi|\,(|0\rangle\langle 0|\,|\psi\rangle) = \begin{bmatrix}\alpha^* & \beta^*\end{bmatrix}\left(\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\right) = |\alpha|^2,$$

and similarly for the result $|1\rangle$, we have

$$p_1 = \langle\psi|\,(|1\rangle\langle 1|\,|\psi\rangle) = \begin{bmatrix}\alpha^* & \beta^*\end{bmatrix}\left(\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\right) = |\beta|^2,$$

as mentioned in section 2.3.1. For our purposes, these notions of quantum measurement suffice.

2.3.4 Entanglement

We conclude our introduction of quantum computation with entanglement. It has served as a key resource in early quantum protocols such as *super dense coding* [BW92] and *quantum teleportation* [BBC$^+$93]. Before explaining the concept of entanglement, we need to understand separable composite quantum systems.

Definition 2.3.2 (separable states)**.** *Pure quantum states that can be expressed as a tensor product of single quantum states are called separable. For example, consider Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$, and state $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$. The pure state $|\psi\rangle$ is separable if and only if there exist states $|\phi\rangle \in \mathcal{H}_A$ and $|\xi\rangle \in \mathcal{H}_B$ such that*

$$|\psi\rangle_{AB} = |\phi\rangle_A \otimes |\xi\rangle_B . \tag{2.15}$$

We then define entanglement naturally as the following.

Definition 2.3.3 (entanglement)**.** *A pure quantum state that is not separable is entangled. For example the state*

$$\left|\Psi^-\right\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}},$$

is an entangled two qubit state.

In fact, the state $|\Psi^-\rangle$ from the above definition is the famous *EPR state* introduced by Einstein, Podolsky and Rosen, and is one of the four *Bell states*.

Definition 2.3.4 (Bell states)**.** *The four maximally entangled bipartite quantum states are*

called the Bell states and are defined as follows

$$\begin{aligned}
\left|\Phi^{+}\right\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\
\left|\Phi^{-}\right\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\
\left|\Psi^{+}\right\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\
\left|\Psi^{-}\right\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}.
\end{aligned} \tag{2.16}$$

The intriguing properties of entanglement can be illustrated by the following example. Two physicists, Alice and Bob prepared the Bell state $|\Psi^{-}\rangle$ in a laboratory, and each parted ways with one of the two qubits to their own labs. At a later time, both agreed to measure their own qubit in the computational basis $\{M_0 = |0\rangle\langle 0|, M_1 = |1\rangle\langle 1|\}$. As shown previously in section 2.3.3, whoever measures first will obtain 0 with probability $||\frac{1}{\sqrt{2}}||^2 = \frac{1}{2}$, and 1 with the same probability. But the strange thing is that, as soon as Alice measures her own qubit and gets a classical result $b \in \{0, 1\}$, she will know immediately that Bob will get $\bar{b} = 1 - b$ as a result whether Bob has performed the measurement or not. This is also true the other way around if Bob measures his qubit first. This is the property that perplexed many when they were first introduced to quantum entanglement. However, if we look at it in terms of information, from Alice's point of view, her qubit represented in so-called *density matrix* (we did not cover density matrices, but interested readers can consult [NC11] and [Wil13] for more details) is

$$\frac{1}{\sqrt{2}}(|0\rangle\langle 0| + |1\rangle\langle 1|).$$

This is identical from Bob's point of view. Whether Alice measures her qubit or not, it does not change the state for Bob's qubit in his point of view prior to the measurement. That is, Bob will still get $|0\rangle$ or $|1\rangle$ with probability $\frac{1}{2}$. It is exactly because of this that no information can be transmitted between parties through measurement and hence it does not violate the causality constraint.

The advantage of quantum computation as opposed to the classical one will be displayed evidently in section 2.5 when we discuss quantum pseudo-telepathy games.

2.4 Multiprover Interactive Proofs

We now take a small detour to introduce multiprover interactive proof systems which laid the foundation for nonlocal games and multiprover bit commitment schemes. In theoretical computer science, a proof is a static sequence of logical symbols that serves to convince a verifier of a mathematical statement. This notion is closely related to two fundamental complexity classes, P and NP , which revolve around the ease to produce and to verify a proof. At the risk of oversimplifying, we mention briefly the concepts of languages and complexity classes below before describing further interactive proof systems. Interested readers can consult [Sip96] and [AB09] for more detailed backgrounds on complexity theories, one of the cornerstones of theoretical computer science.

A language $L \in \{0,1\}^*$ is a bit string of any length that can represent all decisional problem instances with an affirmative answer. A classic example of such a language is $\mathsf{3COL}$ that contains all strings x such that x encodes the description of a 3-colorable graph. A 3-colorable graph depicted in figure 2.1 is a graph with vertices that can be filled with only three distinct colours such that no two adjacent vertices have the same colour. In complexity theories, a proof confirms the membership of a string to a language. In the case of $\mathsf{3COL}$, given a graph G, a proof can be the complete colouring of G with only 3 colours, namely figure 2.1 is a proof that the graph with 5 vertices connected in this way is 3 colourable.

Informally, a language L is in class P if a proof that confirms the membership of a string x to L can be efficiently provided by a prover, whereas L is in class NP when a verifier can efficiently certify that a given proof of string x belongs to L is valid. In NP, the verifier simply reads the proof presented by the prover and verifies its validity. However, a more natural way for proving the soundness of a statement is to allow interactions between the prover and the verifier. By allowing the verifier to interrogate the prover, repeatedly

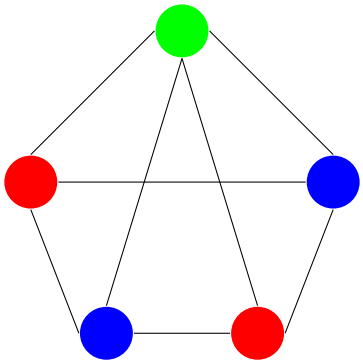


Figure 2.1 – An example of a 3-colorable graph.

questioning and recording the responses produced by the prover round after round, it allows the verifier to certify more complex problems.

This is the intuition behind an interactive proof system. It is also referred to as the complexity class IP which was first introduced in 1985 [GMR85] and independently in the same year as the complexity class Arthur-Merlin (AM) [Bab85] . An interactive proof system is a protocol between a polynomial time [1] bounded verifier and an arbitrarily powerful prover (can have access to unlimited resources), where upon receiving a common input, the two parties exchange a polynomial number of messages and the verifier either accepts or rejects the input in the end. We say a language L admits an interactive proof system if the following two requirements are satisfied:

- **Completeness**: for any $x \in L$, an honest prover can provide a valid proof that can convince a verifier with high probability.
- **Soundness**: if $x \notin L$, no provers can come up with a proof to convince a verifier that $x \in L$ except with some small probability, even if the prover is dishonest or does not follow the protocol.

It has been shown that IP = PSPACE in [LFKN92] and [Sha92], where the class PSPACE is the set of languages recognizable by a Turing machine using polynomial amount of tape

1. An algorithm is said to run in *polynomial time* if the number of steps required to complete the algorithm is upper bounded by a polynomial expression in terms of the size of the input to the algorithm.

space, and contains the class NP.

Merely 3 years later, the notion of interactive proof system was extended to that of multi-prover interactive proof system (MIP) in [BOGKW88] where the concept of two or more non-communicating provers jointly attempt to convince a verifier the soundness of a statement was first introduced. This has been shown to be extremely valuable both in theoretical computer science and cryptography. Although the provers cannot communicate during the protocol, they can share classical resources prior to the protocol and decide on common optimal strategies to collude against the verifier. This new setting can be easily illustrated by the following example where a police officer (verifier) individually interrogates the alibi of all suspects (provers) in separate rooms, and can check their answers against each other. This extra property nontrivially leads to the result that MIP = NEXP [BFL92] in 1992, where NEXP is the class of non-deterministic exponential time, and proved that MIP is believed to be more powerful than IP. The notion of MIP is further expanded with the introduction of quantum information theory, where the class MIP^* introduced in [CHTW04] consists of all powerful quantum provers that share unlimited amount of entanglement. In 2020, MIP^* was proved to have the same computational power as the class RE of recursively enumerable languages in which the halting problem can be solved [JNV$^+$20]. This is one of the most recent pieces of evidence that using entanglement as a computational resource provides a significant advantage over classical ones. The analysis of interactive proof systems requires significant theoretical background in computer science which is not covered in this thesis, but those that are familiar with it will see the stark similarities between nonlocal games and multiprover interactive proof systems.

2.5 Nonlocal games

Multiprover interactive proof systems can be easily reformulated as nonlocal games. What we call provers in the MIP setting are now referred to as players and they want to convince the verifier that they possess a sound strategy to always win the game. Just

like in MIP, provers are usually unbounded computationally while the verifier is strictly polynomial time. From here on, we will use players and provers interchangeably. In this section, we will first formally introduce the definition of a canonical nonlocal game that is mainly adopted from [JNV$^+$20] and [BBT05]. We then introduce other variations of the nonlocal game such as multiplayer nonlocal games and repeated nonlocal games. Finally, we present a transformation of a nonlocal game called anchoring transformation adapted from [BVY15] that can simplify our analysis. We have already seen an example of a nonlocal game in section 1.2 where we introduced the CHSH game. Other nonlocal games such as the Magic Square game and the Mermin-GHZ game will be introduced later on in chapter 3 and chapter 4, respectively.

A two-player nonlocal game is a cooperative game of incomplete information with players Alice and Bob and a verifier defined as follows.

Definition 2.5.1 (Two-player one-round nonlocal game)**.** *A two-player one-round nonlocal game G is specified by a tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \pi, W)$, where*

- *$\mathcal{X}, \mathcal{Y}$ are finite sets that represent the inputs or the questions from the verifier.*
- *$\mathcal{A}, \mathcal{B}$ are finite sets for the answers from players.*
- *π is called the promise and is a probability distribution over $\mathcal{X} \times \mathcal{Y}$.*
- *$W : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \times \mathcal{B} \to \{0, 1\}$ is a function called the winning predicate.*

Prior to the game, players Alice and Bob can share classical or quantum resources such as a series of random bits or entanglements and form strategies. When the game starts, Alice and Bob are not permitted to communicate. This can be accomplished with relativistic constraints. For example, they are separated at a sufficiently large distance such that a signal from one party traveling at the speed of light would not reach the other party in time to affect the other party's action during the game. In the beginning of the game, a pair of questions $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is sampled randomly according to the distribution π by the verifier. Alice is given the question x, and y is sent to Bob. Upon receiving the questions, Alice and

Bob answer with $a \in \mathcal{A}$ and $b \in \mathcal{B}$, respectively. They win the game if $W(x, y, a, b) = 1$ and lose otherwise.

We can also generalize the above two-party one-round game to the n-party case.

Definition 2.5.2 (Multiparty one-round nonlocal game)**.** *A n-party one-round nonlocal game is defined by the tuple $(\mathcal{X}, \mathcal{A}, \pi, W)$, where*

- *$\mathcal{X} = \times_{t=1}^{n} \mathcal{X}_t = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n$ is the set of all questions, and each $\mathcal{X}_t$ is the finite set of questions for player t.*
- *Similarly, $\mathcal{A} = \times_{t=1}^{n} \mathcal{A}_t = \mathcal{A}_1 \times \mathcal{A}_2 \ldots \times \mathcal{A}_n$ is the set of all answers with $\mathcal{A}_t$ being the finite set of answers from player t.*
- *π is again the probability distribution over $\mathcal{X}$.*
- *$W : \mathcal{X} \times \mathcal{A} \to \{0, 1\}$ is the winning predicate.*

Usually, the input set and output set for each player are identical with $\mathcal{X}_i = \mathcal{A}_i = \{0, 1\}$ for $i \in [n]$, and the winning predicate is simply a condition that the value of a binary operation on all inputs is the same as the value of another binary operation on all outputs. This is exactly the case for the CHSH game as we have seen earlier. We now define the notion of strategies and what it means to have a winning strategy for a nonlocal game.

Definition 2.5.3 (Deterministic strategy)**.** *A deterministic strategy is an injective deterministic function that produces an output for every input. It can be further broken down into a set of individual deterministic functions for each player. For the n party case, S denotes the set of deterministic strategies for all players such that*

$$S = \times_{t=1}^{n} S_t = S_1 \times \ldots \times S_n,$$

and S_t denote the set of strategies available for player t. A local deterministic strategy $s \in S$ is then

$$\begin{gathered} s : \mathcal{X} \to \mathcal{A} \\ s(x) = s(x_1, \ldots x_n) = (s_1(x_1), \ldots, s_n(x_n)) = (a_1, \ldots, a_n) = a. \end{gathered} \tag{2.17}$$

This corresponds to the case where players agree on their individual actions before receiving their respective inputs. If players share randomness $r \in R$ in the classical case, and each player computes their output with their own input and the shared value: $a_t = s_t(x_t, r)$ for all $t \in [n]$, then the strategy is called *local*. The following is an example of the local deterministic strategy that a single player can employ to produce all the possible output bits as a function of the input bits. Note that we can reformulate the following in terms of other local deterministic strategies that have the same effect.

Definition 2.5.4 (A single player local deterministic strategy)**.** *Given two random bits r_0, r_1, and an input bit x, a player can use the following local deterministic strategy to answer back an output bit:*

$$s : \mathcal{X} \times R_0 \times R_1 \to \mathcal{A}$$

$$s(x, r_0, r_1) = x \cdot r_0 \oplus r_1 = a.$$

When $(r_0, r_1) = (0, 0)$ or $(r_0, r_1) = (0, 1)$, the above strategy will produce constant outputs $s(x, 0, 0) = 0$ and $s(x, 0, 1) = 1$ respectively, regardless of the input bit x. When $(r_0, r_1) = (1, 0)$, we just output back the input bit $s(x, 1, 0) = x \oplus 0 = x$, and when $(r_0, r_1) = (1, 1)$, we answer with the complement bit of the input bit $s(x, 1, 1) = x \oplus 1 = \overline{x}$. We use this single player deterministic strategy in our analysis of the Mermin-GHZ game.

The difference between a deterministic strategy and a local deterministic one is that the former can be a function of variables from all players, whereas the latter one can only be composed of local variables from a single player. Despite this distinction, we will use the term deterministic strategy to refer to local deterministic strategy from this point on. We use the term classical strategies in a broad sense to include all strategies from players that do not share nonlocal resources. The next definition describes optimal classical strategy that can achieve the highest possible winning probability using deterministic strategies alone.

Definition 2.5.5 (optimal deterministic strategy)**.** *We denote the classical value of a game G to be $\omega_c(G)$ which is the maximum probability with which classical players can win over*

the set of deterministic strategies S.

$$\omega_c(G) = \max_{s \in S} \sum_{x \in \mathcal{X}} \pi(x) W(x, s(x)). \tag{2.18}$$

We say a classical strategy $\sigma \in S$ is an optimal deterministic strategy if it achieves $\omega_c(G)$; that is players using σ can win a game G with questions chosen according to π with probability $\omega_c(G)$.

It is without loss of generality that we restrict our attention to deterministic strategies for the optimal classical strategies. This is because a randomized strategy is equivalent to a convex combination of the deterministic ones. We can also fix the shared random variable R to the value of the best strategy, and transform the randomized strategy to a deterministic one. Hence, it is reasonable for us to make the assumption that the best course of action for classical players during a nonlocal game is to find an optimal deterministic strategy and simply follow the strategy deterministically to answer any queries given by the verifier. This is because if any player uses a different strategy compared to the rest of the players, it can lead to an overall unsatisfying answer for the given query with high probability.

Definition 2.5.6 (winning strategy)**.** *A strategy is called a winning strategy if players using it can win any instance of the game G with probability* 1 *as long as the questions are sampled according to the distribution π of the game.*

A game is called nonlocal when there is no classical winning strategy, but players that share the appropriate nonlocal correlations can reach a winning strategy. We denote $\omega^*(G)$ to be the maximum winning probability of the game G when players are allowed to share entanglements and perform local unitary transformation on their respective qubits.

A large subset of nonlocal games is called pseudo-telepathy games where no classical winning strategy exists, but players that share entanglements can form a quantum winning strategy. As a consequence, the game should be played many times until either players lose one round of the game, or they win consistently. This way, it can convince observers that

something classically impossible is happening. The difficulty of the game is quantified by how bad the optimal classical strategies do at the game. The harder the pseudo-telepathy game, the more convincing to observers that the players appear to communicate somehow, and thus have "telepathic" powers.

2.5.1 Parallel repeated nonlocal game

We can easily extend our definition of a multiparty nonlocal game to that of the parallel repeated game which is simply a multiparty game played with multiple instances in parallel.

Definition 2.5.7 (Parallel multiparty nonlocal games)**.** *Let $G = (\mathcal{X}, \mathcal{A}, \pi, W)$ be a $n-$player one-round game as defined previously, then we say the game $G^k = (X, A, \pi^k, W^k)$ is the $k-$fold parallel repetition of the game G, where*

- *the verifier samples the question*

$$x = \begin{pmatrix} x^1 \\ \vdots \\ x^i \\ \vdots \\ x^k \end{pmatrix} = \begin{pmatrix} x_1^1 & \dots & x_t^1 & \dots & x_n^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^i & \dots & x_t^i & \dots & x_n^i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^k & \dots & x_t^k & \dots & x_n^k \end{pmatrix}.$$

We use subscripts to denote which player we are referring to, and superscript to denote which instance of the game the player is in. This means that the single input x_t^i for $i \in [k], t \in [n]$ is the question for the t^{th} player in the i^{th} instance of the game. We use $x^i = (x_1^i, \dots, x_n^i) \in X^i \in \mathcal{X}$ to represent the questions for the i^{th} instance of the game for players 1 to n sampled according to π. Similarly, $x_t = (x_t^1, \dots, x_t^k) \in X_t \in \bigtimes_{i=1}^k \mathcal{X}_t$ is the question sampled for player t in all k instances of the game.

- *The players answer with*

$$a = \begin{pmatrix} a^1 \\ \vdots \\ a^i \\ \vdots \\ a^k \end{pmatrix} = \begin{pmatrix} a_1^1 & \dots & a_t^1 & \dots & a_n^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_1^i & \dots & a_t^i & \dots & a_n^i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_1^k & \dots & a_t^k & \dots & a_n^k \end{pmatrix}.$$

- π^k *is the product probability distribution over* X*, meaning* $\pi^k(x) = \prod_{i=1}^{k} \pi(x^i)$.

- $W^k : X \times A \to \{0,1\}, W^k(x,a) = \prod_{i=1}^{k} W(x^i, a^i)$.

It is interesting to note that the winning predicate of the $k-$fold parallel multiparty game returns 0 if players fail to answer properly in any of the k parallel executions of the game. This makes winning the repeated version of the nonlocal game much harder. Naturally, one would think that the maximum optimal winning probability of the repeated nonlocal game decreases exponentially with respect to the number of repetitions, namely, $\omega_c(G^k) \leq \omega_c(G)^k$. It turns out this is close to the reality for any two party games as proven by Raz in his seminal paper in 1998 [Raz98], where the exponential decay happens slower. However, 20 years later, a multiparty flavour of this result has yet to be proven. The main difficulty is that we cannot simply assume that players will execute each parallel repetition of the game independently. That means their answers to a specific instance of the game can depend on their questions from any other instances of the game. To side step this obstacle and simplify our analysis, we will introduce a trivial transformation to nonlocal games called the *anchoring transformation* proposed recently in 2015 in [BVY15], which preserves the value of the game and has exponential decay in its value in the parallel execution of the transformed game.

2.5.2 Anchored Games

We adapt the definition from [BVY15] and define a multiparty anchored game $G^{\perp}$ with a parameter α as follows.

Definition 2.5.8 (Multiparty Anchored Games)**.** *Let $G = (\mathcal{X}, \mathcal{A}, \pi, W)$ be a $n-$player one-round game as defined previously. We denote the game $G^{\perp} = (\mathcal{X}^{\perp}, \mathcal{A}, \pi^{\perp}, W^{\perp})$ to be an $\alpha-$anchored game.*

- *Let $\perp$ denote an anchored question, and for all $t \in [n]$, let $\mathcal{X}_t^{\perp} = \mathcal{X}_t \cup \{\perp\}$ be the set of questions to player t. Then, $\mathcal{X}^{\perp} = \times_{t=1}^{n} \mathcal{X}_t^{\perp} = \mathcal{X}_1^{\perp} \times \cdots \times \mathcal{X}_n^{\perp}$*
- *$\forall t \in [n]$, $\pi_t^{\perp}(x_t = \perp) \geq \alpha$, where $\pi_t^{\perp}$ is the marginal probability distribution on the t^{th} player's questions.*
- *$\forall x = (x_1, \ldots, x_n) \in \mathcal{X}^{\perp}, F_x \subseteq [n]$ denotes the set of coordinates of x that are anchored, meaning*

$$F_x = \{t \in [n] \ : \ x_t = \perp\},$$

then let $\pi(x|_{\overline{F}_x})$ to be the marginal probability of the question x restricted to the coordinates $\overline{F}_x = [n] \setminus F_x$, and

$$\begin{aligned} \pi^{\perp}(x) &= \pi^{\perp}(x|_{\overline{F}_x}) \cdot \pi^{\perp}(x|_{F_x}) \\ &= \pi(x|_{\overline{F}_x}) \cdot \prod_{t \in F_x} \pi^{\perp}(x_t), \end{aligned}$$

where the first equality signifies that the probability distribution on the questions that are anchored are independent of those that are not. The second equality shows $\pi^{\perp}(x|_{\overline{F}_x}) = \pi(x|_{\overline{F}_x})$ since $x|_{\overline{F}_x}$ are the questions from the original game and more importantly, $\pi^{\perp}(x|_{F_x}) = \prod_{t \in F_x} \pi^{\perp}(x_t)$ means that the probability distributions of the anchored questions are independent of each other as well.

- *Finally, the winning condition of the anchored game is modified trivially as follows*

$$W^{\perp}(x,a) = \begin{cases} W(x,a) & \text{if } \forall t \in [n], x_t \neq \perp \\ 1 & \text{otherwise} \end{cases},$$

where the verifier evaluates the answers just like in the original game if none of the sampled questions is $\perp$, otherwise, he simply accepts it.

Even though the definition allows the probability of an input being anchored to be at least α, we fix it to be exactly α when we apply the transformation. Otherwise, when this probability is large, it becomes rare for the players to actually play the original game. Another way to interpret the above definition is that the verifier samples the question $x = (x_1, x_2, \ldots, x_n) \in \mathcal{X}$ according to π, and replaces each x_t for $t \in [n]$ with the symbol $\perp$ independently with probability α. The verifier then accepts regardless of players' answers if any of their questions is anchored. In other words, in the $\alpha-$anchored game, the provers play the original game with probability $(1-\alpha)^n$, and the rest of the time, they play a trivial game. This transformation appears to make the game easier than the original one, but it actually facilitates dealing with the technical difficulties of the parallel repeated games where players can use non-product strategies as mentioned before. The anchoring transformation leads to three important results that we will be using when analyzing the security of our protocols.

Theorem 2.5.1 ([BVY15])**.** *The polynomial-time anchoring transformation takes a description of an arbitrary $n-$player one-round game G and returns a game $G^{\perp}$ with the following properties:*

1. $\omega_c(G^{\perp}) = 1 - (1-\alpha)^n \cdot (1 - \omega_c(G))$.
2. $\omega^*(G^{\perp}) = 1 - (1-\alpha)^n \cdot (1 - \omega^*(G))$.

3. *if* $\omega_c(G) \leq 1 - \epsilon$, *then*

$$\omega_c((G^{\perp})^k) \leq \exp\big(-\Omega(\alpha^{2n} \cdot \epsilon^3 \cdot k)\big),$$

where k *is the amount of time the game* $G^{\perp}$ *is executed in parallel, and* $\Omega(\cdot)$ *depends on* k *and the rest of the terms showed can be reduced to a constant factor.*

A more relevant result for us is that there exists a constant α *such that the above can be reduced to the following expressions:*

1. $\omega_c(G^{\perp}) = \frac{1}{4} + \frac{3}{4}\omega_c(G)$.
2. $\omega^*(G^{\perp}) = \frac{1}{4} + \frac{3}{4}\omega^*(G)$.
3. *If* $\omega_c(G) \leq 1 - \epsilon$, *then*

$$\omega_c((G^{\perp})^k) \leq \exp\big(-\Omega(\epsilon^3 \cdot k)\big).$$

To achieve the simplified expressions in the latter part of the theorem, we can derive an explicit formula for the probability α in terms of n, the number of players.

$$\alpha = 1 - \sqrt[n]{\frac{3}{4}}. \tag{2.19}$$

There is another important result related to the value of the repeated anchored game with quantum players $\omega^*((G^{\perp})^k)$, but we chose to omit it here. This is because the games we analyze all possess a quantum winning strategy, which means their values will remain unchanged:

$$\omega^*((G^{\perp})^k) = \omega^*(G^k) = \omega^*(G)^k = 1.$$

We can observe from the above theorem that the value of the anchored game $\omega_c(G^{\perp})$ improves if the original value $\omega_c(G)$ is less than 1, and the same applies for the quantum case. This makes sense since the players have a significant probability of playing a free game. On the other hand, $\omega_c(G^{\perp})$ remains unchanged otherwise. The main takeaway is that the

value of the repeated anchored game using classical strategies follows an exponential decay in terms of k, the number of times the game is repeated.

2.6 PR box

We formally describe the PR box mentioned earlier in the introduction. A PR box, introduced by Popescu and Rohrlich in ([PR94], [PR98]), is an imaginary device that can achieve the CHSH correlation. The box accepts two binary inputs x, y, and outputs two bits a and b, respectively, such that $a \oplus b = x \wedge y$. The correlation can be captured in the following with both inputs being randomly sampled from a uniform distribution,

$$\Pr(a, b|x, y) = \begin{cases} \frac{1}{2}, \text{ if } a \oplus b = x \wedge y \\ 0, \text{ otherwise} \end{cases} .$$

The PR box is operated in an asynchronous manner, meaning that the box outputs a as soon as it receives the input x even if y has not yet been received, and vice versa. The box is consistent with relativity since no information is communicated through its use. Local players can simulate the PR box successfully with a maximum probability of 75%, while quantum players sharing entanglements can do so about 85% of the time. We can generalize the PR box to a more fundamental information theoretic concept called *no-signalling*. No-signalling provers are those that make use of the PR box or any other no-signalling correlations. The only restriction for them is that no communication can take place, which is the least restrictive in terms of computational power. One of the consequence of the PR box is that it can achieve trivial classical communication complexity [VD13] which suggests heavily that a physical implementation of a PR box is impossible. Despite this, the PR box is still interesting to cryptographers since no-signalling provers can break cryptographic protocols that are secure even against quantum adversaries as demonstrated in [CSST11].

2.7 Security definition

In this section, we briefly introduce the notion of security in terms of bit commitment schemes. After that, we introduce a new binding definition of bit commitment schemes that we call the non-binding game. For a more complete view on the security of cryptosystems and protocols, readers can consult [KL14].

It is sometimes unreasonable to require "perfect security" in cryptographic protocols. Though ideal, we typically define the security of a cryptosystem in terms of a parameter, called the security parameter, such that the failure probability can be made arbitrarily small. Let $n \in \mathbb{N}$ be the security parameter. What we want is an inverse relationship with n and the failure probability, meaning that as n grows larger and larger, the failure probability becomes negligibly small asymptotically.

Definition 2.7.1 (negligible function)**.** *We say a function $\epsilon : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if for all positive polynomials p, there exists an integer N such that*

$$n > N \Rightarrow \epsilon(n) < \frac{1}{p(n)}. \tag{2.20}$$

For a bit commitment scheme between Alice and Bob, the following are the security definitions of the scheme.

Definition 2.7.2 (hiding)**.** *A bit commitment scheme is said to be statistically hiding if the receiver Bob gets only a negligible amount of information about the bit b prior to the opening of the commitment. The scheme is unconditionally hiding if Bob learns zero information.*

Definition 2.7.3 (historical binding)**.** *A bit commitment scheme is statistically binding if Alice cannot succeed in unveiling both values of the bit b' with a probability non-negligibly greater than 1. More formally,*

$$\Pr\big[\ \textit{Bob accepts} \mid \textit{Alice unveils } b' = 0\big] + \Pr\big[\ \textit{Bob accepts} \mid \textit{Alice unveils } b' = 1\big] \leq 1 + \epsilon(n).$$

Here Pr[*Bob accepts* | *Alice unveils* b'] *is the probability that Alice succeeds in unveiling* b' *to Bob, and* $\epsilon(n)$ *is negligible in terms of the security parameter* n*. In the case that* $\epsilon(n)$ *vanishes to* 0*, this is the same as Alice is forced to commit to a fixed bit.*

We introduce this binding definition for historical context, but we do not think it is still suitable for the commitment schemes that we examine in this work. This is the reason why we will use the binding game introduced below to analyze the binding property of our bit commitment schemes.

2.7.1 Non-binding game

With the discovery of nonlocal games, application of them in bit commitment schemes also started appearing in [BCU$^+$06], [WWW11],[AMPS16], [SCA$^+$11], [CSST11]. These commitment schemes remain classically secure without relying on extra computational assumptions such as the existence of one way functions and collision-free hash functions. The binding condition as defined above is also no longer satisfactory. Provers with the appropriate nonlocal resources can cheat the binding properties of these commitment schemes. We know the commitment scheme is not secure against non classical provers, but we don't know how easy it is for them to break it. Furthermore, the above definition for binding is not applicable to commitment schemes with multiple provers. Hence, it is natural to redefine the binding definition for commitment schemes constructed with nonlocal games in order to quantify how secure the commitment schemes are against different types of provers. It is worth mentioning that this new definition of binding presented below does not necessarily exist in the literature yet. We will refer back to this section when we analyze the binding property of our proposed commitment schemes in chapter 4.

Since our discussions in this work center around nonlocal games, we will formulate the binding condition of a commitment scheme in the form of a game as well. The redefined binding property known as the non-binding game can be conducted in the same setting as the commitment scheme that we are analyzing. That is, it can be played with the same

amount of provers, and against provers having different kind of resources. The intuition behind the new definition is that honest provers in a commitment scheme that is binding will be stuck with the value that they committed to, and cannot unveil other values successfully with non negligible probability. To reflect this, the non-binding game quantifies the success probability of provers unveiling a value that is not necessarily the one committed by their peers. The main difference between the non-binding game and the commitment protocol is that the unveiler will be forced to reveal a random value given by the verifier instead of the predetermined value committed by other provers. Obviously, if a prover succeeds in convincing the verifier that the value he is forced to reveal is the correct one all the time, then the commitment scheme is not binding at all. The non-binding game is described informally as below.

The non-binding game is conducted in two phases called the query phase and the challenge phase. Prior to the start of the non-binding game, the provers are allowed to discuss strategies and share resources. In particular, they do not have to decide on which value to commit to and can pick a series of strategies that allow them to commit any of the possible values. After this, the provers are separated and cannot communicate among each other. The query phase is performed in the same way as the commit phase in the commitment protocol. During the challenge phase, the verifier chooses a value c uniformly at random that the provers can commit to, and sends it to the prover responsible for unveiling [2]. Then the unveiler and the verifier interact the same way as the unveil phase of the original commitment protocol. The unveiler responds adaptively with the necessary information to open the commitment to the value c chosen by the verifier. The verifier collects all the exchanges from provers and performs a consistency check for the validity of the unveiling. Let $q \in \mathbb{N}$ denote the number of possible values that provers can commit to. Then, the commitment scheme is defined to be binding as follows.

2. We restrict our focus on single party unveiling since the bit commitment schemes built in this work all have one single prover responsible of opening the commitment.

Definition 2.7.4 (binding[3])**.** *A commitment scheme is binding if the probability that the provers win the corresponding non-binding game $\omega(G_{nb})$ with q possible values to unveil satisfies*

$$\omega(G_{nb}) \leq \frac{1}{q} + \epsilon(n), \tag{2.21}$$

where $\epsilon(n)$ is negligible in n.

On the other hand, the commitment scheme is not binding when the provers can win the non-binding game with probability non-negligibly better than $\frac{1}{q}$. We say a commitment scheme is fully non-binding when provers can win the non-binding game with probability 1. We present the non-binding game in the context of the bit commitment scheme constructed with the protocol from this work in section 4.1.1.

3. We believe our current binding definition is the most appropriate. The proof that our binding definition is equivalent to the historical binding definition 2.7.3 is left as an exercise to the readers.

Chapter 3

Pseudo-telepathy Game Bit Commitment Scheme

In this chapter, we study the subject of our main contribution in this thesis: bit commitment schemes from pseudo-telepathy games. The application of nonlocal games in bit commitment schemes is a relatively new concept, and we aim to present a generalized protocol to construct classically secure bit commitment schemes using any pseudo-telepathy game in this chapter. This is in an effort to use the laws of physics to achieve cryptographic tasks instead of relying on extra assumptions such as the existence of one way functions or collision free hash functions.

Commitment protocols that make use of the PR box introduced in section 2.6 have been proposed in [BCU$^+$06], [WWW11], and [AMPS16]. A bit commitment scheme constructed with the Mermin-GHZ game, which will be introduced in 4.2.1, also appeared in [SCA$^+$11]. These research efforts demonstrated that we can realize secure cryptographic primitives using nonlocal correlations. The protocols share the same characteristic that they rely on the existence of physical implementation of nonlocal boxes that contain the respective nonlocal correlations of the games. The verifier in these protocols can delegate most of the computation and verification steps to the nonlocal boxes since the provers rely on their own

boxes to output their answers. As long as the verifier can guarantee the integrity of the boxes, they can achieve classically secure bit commitments. They can go even a step further and obtain device-independent bit commitment schemes which guarantee the security of the cryptographic protocol regardless of the trustworthiness of the underlying physical devices. However, the commitment protocols that this work is trying to achieve is different from these results. While we do use nonlocal games in our protocol, the security of our protocol do not rely on the physical realization of the nonlocal boxes.

The direction of our research is heavily inspired by another protocol implemented using only the Magic Square game in [CSST11]. This bit commitment scheme is special since its implementation does not need to rely on the existence of physical nonlocal boxes as opposed to the others mentioned above. One common characteristics of all of these bit commitment schemes implemented using nonlocal correlations is that they are binding against classical provers, but provers that share the proper non local correlations can break the binding property using existing winning strategies for the respective games. This is because no classical winning strategies exist for nonlocal games. The Magic Square game and its bit commitment scheme are introduced below. We will then introduce the remaining ingredients required for our result. Finally, we present a protocol to construct a bit commitment scheme using any pseudo-telepathy games.

3.1 Magic Square bit commitment scheme

3.1.1 Magic Square game

The Magic square game is a two-player pseudo-telepathy game that is first introduced by Aravind in [Ara02], which is built on earlier work by Mermin in [Mer90]. A magic square is defined as a 3×3 matrix of bits. The condition for the square is that the sum of each row is even, and the sum of each column is odd. In other words, each row of this square has even parity, and each column of the square has odd parity. It can be easily verified that

this square cannot exist since it is impossible to satisfy all of the above constraints at the same time. Consider the square below with binary entries.

a	b	c
d	e	f
g	h	i

According to the row parity condition, the following equation should hold.

$$a \oplus b \oplus c \oplus d \oplus e \oplus f \oplus g \oplus h \oplus i = 0.$$

Meanwhile the following equation should also hold to satisfy the column parity condition.

$$a \oplus d \oplus g \oplus b \oplus e \oplus h \oplus c \oplus f \oplus i = 1.$$

Clearly, this is a contradiction. This concludes the proof that no such magic square exists.

The game is as follows. Two players, namely Alice and Bob are asked to provide the following information. Alice is asked to give the entries of a row $x \in \{0, 1, 2\}$, and as for Bob, the entries of a column $y \in \{0, 1, 2\}$ of the magic square matrix. Alice and Bob win the game if the parity conditions of the rows and columns are met, and the intersection of the given row and column agree. Figure 3.1 shows an example of an instance of the Magic Square game where Alice and Bob share a classical strategy.

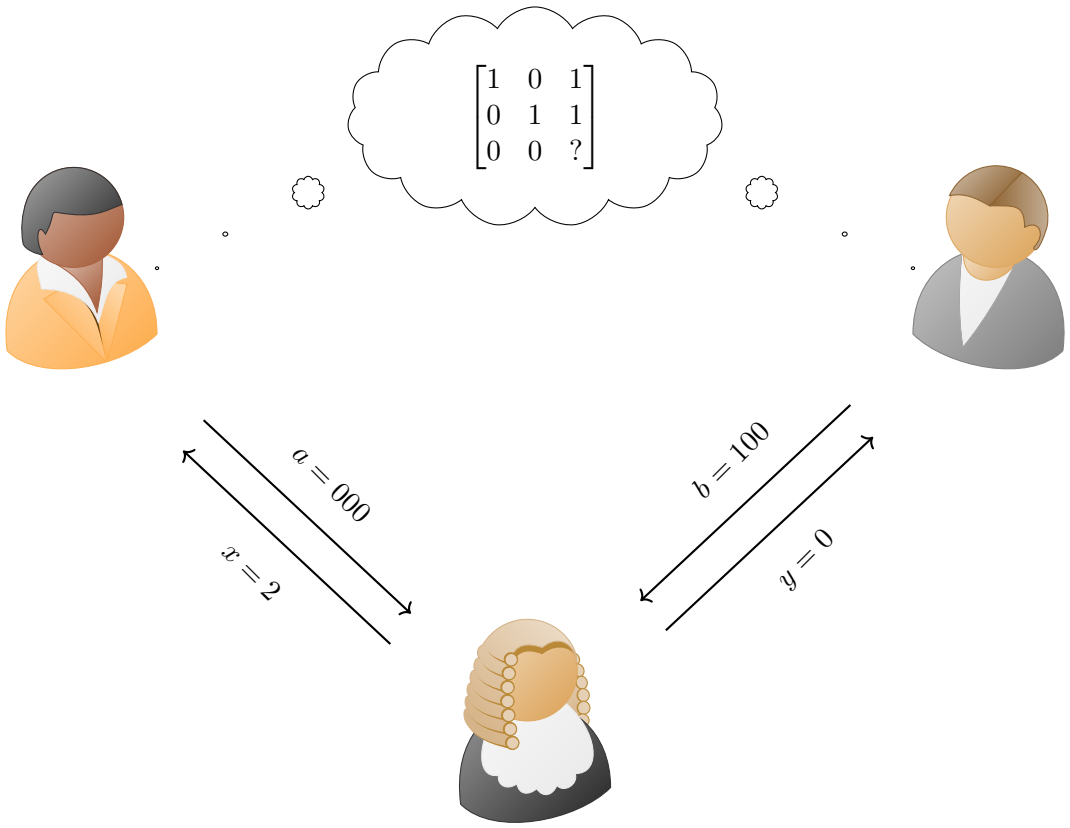


Figure 3.1 – An illustration of the Magic Square game. Alice (depicted on the left) and Bob (depicted on the right) share a classical optimal strategy before the game begins. Verifier (depicted as the judge on the bottom) sends the question $(x, y) = (2, 0)$ to Alice and Bob respectively. To satisfy the even parity condition of a row, Alice answers with $a = 000$ by changing the last entry in the 3^{rd} row to be 0. Bob answers with the entries in the first column $b = 100$. The answer from Alice and Bob agree on the intersecting element: the 3^{rd} element of the first column and the first element of the 3^{rd} row both equal to 0. They win this round of the game.

The quantum winning strategy [BBT05] of this game requires that Alice and Bob share

the following entangled state:

$$|\psi\rangle = \frac{1}{2}|0011\rangle - \frac{1}{2}|0110\rangle - \frac{1}{2}|1001\rangle + \frac{1}{2}|1100\rangle . \tag{3.1}$$

The first two qubits belong to Alice, and the last ones belong to Bob. Then, upon receiving inputs x and y, Alice and Bob apply the following unitary transformation $A_x \otimes B_y$, where x corresponds to the row number and y corresponds to the column number.

$$A_0 = \frac{1}{\sqrt{2}}\begin{bmatrix} i & 0 & 0 & 1 \\ 0 & -i & 1 & 0 \\ 0 & i & 1 & 0 \\ 1 & 0 & 0 & i \end{bmatrix},\ A_1 = \frac{1}{2}\begin{bmatrix} i & 1 & 1 & i \\ -i & 1 & -1 & i \\ i & 1 & -1 & -i \\ -i & 1 & 1 & -i \end{bmatrix},\ A_2 = \frac{1}{2}\begin{bmatrix} -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix},$$
$$B_0 = \frac{1}{2}\begin{bmatrix} i & -i & 1 & 1 \\ -i & -i & 1 & -1 \\ 1 & 1 & -i & i \\ -i & i & 1 & 1 \end{bmatrix},\ B_1 = \frac{1}{2}\begin{bmatrix} -1 & i & 1 & i \\ 1 & i & 1 & -i \\ 1 & -i & 1 & i \\ -1 & -i & 1 & -i \end{bmatrix},\ B_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}.$$

Players perform a measurement on their respective qubits in the computational basis to obtain the first two qubits of their answers a and b. Then, the third bit of their answers are computed to satisfy the parity condition. For example, if Alice and Bob receive $x = 0, y = 1$ respectively, after applying the unitary transformations, their shared quantum states evolve to:

$$\begin{aligned}(A_0 \otimes B_1)|\psi\rangle = \frac{1}{2\sqrt{2}}[&-|0000\rangle + |0001\rangle + i|0110\rangle + i|0111\rangle \\ &- i|1000\rangle - i|1001\rangle + i|1110\rangle - i|1111\rangle].\end{aligned}$$

Alice and Bob measure their respective qubits and obtain a result with equal probability $\frac{1}{8}$. For instance, Alice could get 01, and Bob could get 11 from the state $|0111\rangle$. To satisfy the parity condition, Alice would complete with bit 1 resulting in $a = 011$, and similarly, Bob would obtain the result $b = 111$. Alice and Bob win the game since the parity condition is met and the second entry of first row is the same as the first entry of the second column.

The detail of the possible outputs for Alice and Bob after performing the above quantum computation is listed in the following table.

a	b
000	001
000	010
011	100
011	111
101	001
101	010
110	100
110	111

Table 3.1 – Table for the possible outputs for players on input $(x, y) = (0, 1)$, where the first column is the output for Alice and the second column is the output for Bob. The blue bits are the intersection entries for Alice and Bob. The red bits are the last bits that Alice and Bob complete to satisfy the row and column parity conditions.

We can easily see that there exists no classical winning strategies since this would imply the existence of a deterministic magic square. Any optimal classical strategy can win the game with probability $\frac{8}{9}$ because there will always be one entry where Alice and Bob cannot agree.

3.1.2 Commitment scheme

We present below the Magic Square game bit commitment scheme introduced in [CSST11], where the Magic Square game used is a slight variation from the one introduced above, but the core ideas remain the same. The only difference is that Bob is now asked all three entries in either a row or a column, and Alice is asked only one of the three values from the row or the column that Bob answered. They win the game if the answer from Bob respects the row and column parity condition as defined before, and Alice's single entry agrees with Bob's answer as well. This version of the Magic square game is presented in [CHTW04] along with a quantum winning strategy that is similar in nature to that introduced previously. We define a particular classical strategy for the Magic Square game that will be used in the commitment protocol.

Definition 3.1.1 (valid matrix)**.** *To meet the parity condition of the Magic Square game, a matrix S is said to be valid for zero, if all rows of S xor to 0, and similarly, S is valid for one, if all columns xor to 1. For example, consider the following matrices S_0 and S_1:*

$$S_0 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

where S_0 is valid for zero and S_1 is valid for one.

The bit commitment scheme with provers Alice and Bob, and verifier Vic is as follows.

Before Commitment:

1. Alice and Bob agree on a random bit v, and n random squares S_i such that S_i is valid for v for each $i \in [n]$. They are then separated without communicating to each other throughout the protocol.

Commitment phase:

1. Alice computes $x := v \oplus b$, and sends it to Vic.
2. Vic randomly samples a pair of trits [1] (r_i, c_i) and sends the pair to Alice.
3. Alice answers with the entry at the intersection of row r_i and column c_i of the square S_i.

Unveil phase:

1. Alice sends b to Vic.
2. If $b = x$, Vic asks Bob for the entries of row number r_i of S_i, and otherwise, Vic asks for the column number c_i of S_i.
3. Vic accepts b if, for each i, the row or column that should *xor* to b does, and if the entry returned by Alice matches with Bob's answer. Vic rejects otherwise.

The following two theorems are proven in [CSST11].

Theorem 3.1.1 ([CSST11]). *Any classical strategy successfully cheats the binding property of the above bit commitment scheme with a probability of at most* $\left(\frac{17}{18}\right)^{n/6}$ [2]*, except with exponentially small probability.*

Theorem 3.1.2 ([CSST11]). *There exists a quantum strategy that successfully cheats the scheme with probability 1.*

1. ternary equivalent of bits, a trit can take any value in $\{0, 1, 2\}$

2. The original paper claimed that this probability is $\left(\frac{8}{9}\right)^{(n/6)}$, but that is only true if the bit commitment scheme uses the Magic Square game that we defined earlier. The optimal winning probability of the Magic Square game used here for classical players is $\frac{17}{18}$.

As mentioned earlier, a quantum winning strategy exists for this version of the Magic Square game as well. It suffices for players to use it to unveil an alternate bit value b', and still convince the verifier.

The remarkable property about this bit commitment scheme is the following. Despite the fact that the players share classical strategies that only satisfy half of the parity conditions of the Magic Square game, by restricting the queries of the verifier, they can still achieve classically secure bit commitment. The verifier only asks the questions that the players have the correct answer to according to their shared strategies. This is the key observation that inspired our result of Section 3.3. The scheme achieves hiding since x does not reveal anything about the committed bit b without the knowledge of v. The binding condition follows from the fact that no classical winning strategy exists.

3.2 Quantum secure commitment scheme

The following two-prover commitment scheme denominated **sBGKW** originated from [CSST11] and is simpler than the Magic Square commitment scheme. Despite of its simplicity, it is provably binding against both classical and quantum provers, but non-signalling provers can use the PR box introduced in section 2.6 to break the binding property.

Before Commitment:

1. Alice and Bob agree on a random bit string $w \in \{0,1\}^n$, and are then separated without communicating to each other.

Commitment phase:

1. The verifier samples a random bit string $r \in \{0,1\}^n$ and gives it to Alice.
2. Alice answers back with $x \coloneqq (b \cdot r) \oplus w$, where $b \cdot r \coloneqq b \wedge r_1 \,||\, b \wedge r_2 \,||\, \ldots \,||\, b \wedge r_n$, is the bit wise *and* between the string $\underbrace{bb\ldots b}_{n}$ and the string r.

Unveil phase:

1. Bob announces the n bit string w he shared with Alice to the verifier.
2. The verifier deduces b from the following. He computes $y \coloneqq x \oplus w$. If $y = 0^n$, then he accepts $b = 0$, and if $y = r$, he accepts $b = 1$. Otherwise, he rejects the commitment.

We can observe that Bob from the above commitment does not need to know b to unveil it. The commitment scheme is clearly hiding since the string x that the verifier receives is uniformly random. In the case that $b = 0$, he receives $r \oplus w$ which is the xor of two uniformly random bit strings, and otherwise he receives the uniformly random bit string w. The argument for the binding property is as follows. In order for Bob to open $b = 0$, he has to announce $w_0' = x$, and for him to open a commitment of $b = 1$, he has to announce $w_1' = x \oplus r$. If Bob is able to open the commitment in both ways, then it implies that he knows $w_0' \oplus w_1' = r$. However, this is a contradiction since Alice doesn't communicate to Bob during the protocol, and therefore Bob will have to guess correctly what the random bit string is with probability at most $\left(\frac{1}{2}\right)^n$. Interested readers can consult [CSST11] for a more complete proof of security.

On the other hand, if Alice and Bob have access to a pair of correlated PR boxes such that the outputs of the two boxes satisfy the CHSH condition with their individual inputs,

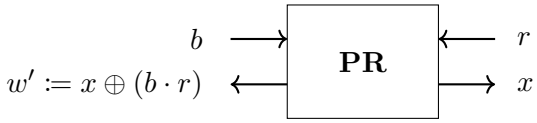


Figure 3.2 – A strategy for no-signalling provers using the PR box to cheat the binding property of the commitment scheme by unveiling any value of b correctly.

then Bob can easily cheat the above commitment scheme. The strategy is illustrated in figure 3.2. Upon receiving r from the verifier, Alice inputs sequentially each r_i in her PR box, and obtains x_i. She sends x to the verifier after inputing all n bits of r. During the unveil phase, Bob decides on a bit b to open. He inputs b to his PR box, and each time he obtains the output bit $w'_i \coloneqq x_i \oplus (b \cdot r_i)$. After n times, he obtains $w' \coloneqq x \oplus (b \cdot r)$, which is exactly what he needs to output to open a commitment of b. If $b = 0$ then $b \cdot r = 0^n$, and he receives $w' = x$, and if $b = 1$ then $b \cdot r = r$, and he receives $w' = x \oplus r$ which are the correct values for him to disclose.

We will use the above commitment scheme for our purpose of building commitment schemes from pseudo-telepathy games that are binding only against local provers.

3.3 Bit commitment scheme from pseudo-telepathy games

We now present the main result of this thesis which is a protocol to transform any pseudo-telepathy game to a classically secure bit commitment scheme that satisfies the binding and hiding conditions that we have defined in section 2.7.1. Before diving into the result we need to introduce a final key ingredient in our protocol: exclusion sets.

3.3.1 Exclusion sets

In analyzing the bit commitment scheme presented in section 3.1.2, it is important to note that the verifier is restricted in terms of his queries. This allows honest classical provers to win the underlying nonlocal game if they follow through an agreed upon deterministic

strategy. Naturally, this does not affect players that have access to the proper nonlocal correlations, since they can win any challenges that the verifier comes up with. We will formalize this idea with the introduction of the exclusion set.

Definition 3.3.1 (exclusion set)**.** *Given a nonlocal game $G = (\mathcal{X}, \mathcal{A}, \pi, W)$, and a deterministic strategy s, we define E_s to be the corresponding exclusion set such that*

$$E_s = \{x \mid W(x, s(x)) = 0\}. \tag{3.2}$$

In simpler terms, an exclusion set E_s of a nonlocal game G is the set of inputs such that using the strategy s will fail the game G systematically. If we focus our attention on any classical optimal strategy σ for game G, we have the following relationship:

$$|E_\sigma| = (1 - \omega_c(G)) \cdot |\mathcal{X}|, \tag{3.3}$$

where $\omega_c(G)$ is the optimal classical winning probability as defined in definition 2.5.5. This is true since all optimal classical strategies achieve the same winning probability. To help visualize this notion, consider the following classical strategy μ presented in fig. 3.1 for the Magic Square game.

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & ? \end{bmatrix}$$

Using this strategy, Alice and Bob can win 8 out of 9 times except for the entry $(2, 2)$. To illustrate how it works, consider the question $(x, y) = (2, 1)$, Alice will have to change the last entry in the 3^{rd} row to be 0 and answer with $a = 000$ in order to make the sum even. Bob will simply answer with the entries in the second column $b = 010$ since it already satisfies the odd parity condition. Alice and Bob also agree on their intersection, and they win this round of the game. On the other hand, with query $(x, y) = (2, 2)$, Alice will answer with $a = 000$ like previously to satisfy the even parity condition of a row. Bob will have to answer with $b = 111$ by changing the last entry of the column to comply with the odd

parity condition of a column. However, their intersection does not agree. This means that $E_\mu = \{(2,2)\}$.

With the definition of an exclusion set, we can also consider the relationship among optimal deterministic strategies. Consider another optimal strategy η for the Magic Square game:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & ? \end{bmatrix}$$

Even though the majority of the entries in η are different compared to those in μ, we can easily observe that η fails also at input $(2,2)$ which means that $E_\eta = E_\mu = \{(2,2)\}$. From this example, it can be observed that many optimal deterministic strategies derive the same exclusion set. This is because for an input x, there exists more than one output a that can fail the winning predicate. We introduce below a notion of equivalence in optimal classical strategies.

Definition 3.3.2 (equivalence relation)**.** *For a nonlocal game G, and two optimal deterministic strategies σ, σ', we say $\sigma \sim \sigma'$ if*

$$E_\sigma = E_{\sigma'}.$$

This means that $\forall x \in \mathcal{X}$, if $W(x, \sigma(x)) = 0$, then $W(x, \sigma'(x)) = 0$ as well, and vice versa. Note that each strategy has only one exclusion set. With this notion of equivalence of strategies, we can use a given exclusion set to refer to all optimal deterministic strategies that derive it. To do so, we can enumerate all of the optimal deterministic strategies and compute their corresponding exclusion sets. We can then group strategies together according to their equivalence relation with relabelling, which resembles a hash map with exclusion sets being the keys and the strategies being the values. This is illustrated below.

We make a simple assumption that for a finite size game G, the computation needed to

E_{σ_1}	$\sigma_1, \ldots, \sigma_j$
$\vdots$	$\vdots$
E_{σ_k}	$\sigma_k, \ldots, \sigma_n$

produce the above relation with exclusion sets and the corresponding optimal deterministic strategies can be done in constant time. This will serve as a starting point for our proposed strategies to build classically secure bit commitment schemes using any pseudo-telepathy games.

3.3.2 The protocol

We propose here a recipe to turn any pseudo-telepathy game into a bit commitment scheme such that the commitment scheme is secure against classical provers but not against provers that share entanglements. We assume that provers participating in the protocol are arbitrarily powerful, and their goal is to prove to a probabilistic polynomial verifier that their commitment is legitimate. To simplify our analysis, we consider only deterministic strategies for the classical provers and do not concern ourselves with random ones. We are allowed to make this assumption since any optimal probabilistic strategy is just a linear combination of optimal deterministic strategies and is upper-bounded by the same optimal winning probability as mentioned in section 2.5.

Our strategy is composed of 2 main transformations of a pseudo-telepathy game G with n players. The first transformation aims to convert G into an anchored game $G^{\perp}$, as introduced in section 2.5.2, which has exponential decay in its value when executed in parallel. We then transform $G^{\perp}$ into a bit commitment protocol with the addition of an extra prover and a constant parameter k that we can fine tune. The detail of the second transformation is summarized below and the entire protocol for transforming a pseudo-telepathy game to a bit commitment scheme follows after it.

Most of the computations for the provers in the bit commitment scheme happen before

the commit phase. From the original game G, provers will determine the optimal winning probability $\omega(G)$, as well as all optimal deterministic strategies, denoted $\vec{\sigma}$, that achieve $\omega(G)$. For each $\sigma \in \vec{\sigma}$, provers will compute their corresponding exclusion set E_σ as defined previously in section 3.3.1. Let $E_{\vec{\sigma}}$ denote the set of all exclusion sets. Provers construct the table of exclusion sets and their corresponding strategies according to the equivalence relation detailed in definition 3.3.2. For a limited size game, we assume that all of this can be done efficiently in constant time.

With the table of exclusion sets and optimal deterministic strategies for the original game G, provers divide the exclusion sets $E_{\vec{\sigma}}$ into $E^0_{\vec{\sigma}}$ and $E^1_{\vec{\sigma}}$ in any way of their choosing as long as the partitions satisfy the following two constraints. The first one is that the two sets are disjoint without any overlapping elements. The second one requires that the two partitions are non empty. These conditions can be formalized as below.

- $E^0_{\vec{\sigma}} \cap E^1_{\vec{\sigma}} = \emptyset$ and $E^0_{\vec{\sigma}} \cup E^1_{\vec{\sigma}} = E_{\vec{\sigma}}$.
- $E^0_{\vec{\sigma}} \neq \emptyset \neq E^1_{\vec{\sigma}}$

This separation of exclusion sets is made public and will be used throughout the bit commitment protocol.

In order to commit to a single bit value $\mathbf{b}$, provers will commit to a binary matrix $\mathbf{B}$ with k^2 randomly sampled independent entries during the commit phase, where

$$\mathbf{B} = \begin{pmatrix} b_{11} & \dots & \dots & \dots & b_{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ b_{k1} & \dots & \dots & \dots & b_{kk} \end{pmatrix},$$

and $i, j \in [k]$, such that each row is independent to each other but also satisfies the following

condition for the matrix $\mathbf{B}$ to be considered valid. For each row $i \in [k]$,

$$\bigoplus_{j=1}^{k} b_{ij} = b_{i1} \oplus \ldots \oplus b_{ik} = \mathbf{b}. \tag{3.4}$$

In other words, the binary sums of each row of the matrix $\mathbf{B}$ are identical and equal to the bit value to be committed. This also means that the values $b_{i_1}, \ldots, b_{ik-1}$ are independent to each other, but b_{ik} can be determined completely.

Essentially, to commit to the binary matrix $\mathbf{B}$, instead of the original game G, provers will actually play the anchored game $G^{\perp}$ with k^2 parallel repetition. This means that for the remainder of the protocol, the players and the verifier participate in the game $\left(G^{\perp}\right)^{k^2} = \left(X, A, \left(\pi^{\perp}\right)^{k^2}, \left(W^{\perp}\right)^{k^2}\right)$, where $X = \bigtimes_{i=1}^{k^2} \mathcal{X}^{\perp}$ and similarly $A = \bigtimes_{i=1}^{k^2} \mathcal{A}$. From here on, we use (i, j) to denote which instance of the game we are referring to, with $i, j \in [k]$. Namely, b_{ij} is the bit value committed in the $(i, j)^{th}$ repetition of the anchored game. We use $t, d \in [n]$ to index the players.

Having access to all the exclusion sets and their corresponding optimal deterministic strategies, provers sample k^2 exclusion sets from either $E^0_{\vec{\sigma}}$ or $E^1_{\vec{\sigma}}$, where we denote each exclusion set with $E^{b_{ij}}_{\sigma}$. The superscript b_{ij} of the exclusion set corresponds to the binary entries of the matrix $\mathbf{B}$ such that the eq. (3.4) is satisfied. For each $E^{b_{ij}}_{\sigma}$, provers agree on an optimal deterministic strategy σ^{ij} that derives it and will use this strategy to answer the challenge from the $(i, j)^{th}$ repetition of the anchored game. Next, we add an extra prover p_{n+1} who shares a uniformly random matrix $\mathbf{u}$ of bit strings with a chosen prover p_t, and no one else, where

$$\mathbf{u} = \begin{pmatrix} u^{11} & \ldots & \ldots & \ldots & u^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & u^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ u^{k1} & \ldots & \ldots & \ldots & u^{kk} \end{pmatrix}.$$

The special index t of prover p_t will be communicated to the verifier and is chosen in order

to satisfy the hiding criteria of definition 4.1.1 of the resultant commitment scheme. This means that for any $(i,j)^{th}$ instance of the anchored game, and for all $x^{ij} \in \mathcal{X}$ that is not anchored, if there exists a strategy $\sigma \in \vec{\sigma}$ and $E_\sigma \in E_\sigma^{b_{ij}}$ such that $\sigma(x^{ij}) = a^{ij} \in \mathcal{A}$, then there exists at least another strategy $\sigma' \in \vec{\sigma}$ such that $E_{\sigma'} \in E_\sigma^{\overline{b_{ij}}}$ and $\sigma'(x^{ij}) = a^{ij\prime} \in \mathcal{A}$ where $a^{ij\prime}$ may differ only from a^{ij} at index t. On top of this, it needs to satisfy the condition that $W(x^{ij}, \sigma(x^{ij})) = W(x^{ij}, a^{ij}) = 1 = W(x^{ij}, a^{ij\prime}) = W(x^{ij}, \sigma'(x^{ij}))$. Intuitively, this means that if player t's output is not known, then having the knowledge of $(x^{ij}, (a_1^{ij}, \ldots, a_{t-1}^{ij}, a_{t+1}^{ij}, \ldots, a_n^{ij})$ is not sufficient to determine b_{ij} since a^{ij} can be the result of a strategy derived from either $E_\sigma^{b_{ij}}$ or $E_\sigma^{\overline{b_{ij}}}$. We will discuss the hiding properties in further length in chapter 4.

Let a_t^{ij} denote player t's output for the $(i,j)^{th}$ instance of the game. Each entry u^{ij} is a random bit string with the property that $\left|u^{ij}\right| = k \cdot \left|a_t^{ij}\right|$. In other words, each u^{ij} is sampled randomly from the set $\{0,1\}^{k \cdot |a_t^{ij}|}$. The random bit string u^{ij} will be used between p_t and p_{n+1} for the commitment of player t's output using **sBGKW** introduced earlier in section 3.2. Player p_t will commit his answer at the same time as his peers output their answers during the commit phase, and player $n+1$ will disclose u^{ij} during the unveil phase for the verifier to infer player t's output during the $(i,j)^{th}$ instance of the game. With all of the above detailed, we are now ready to define the commit phase.

During the commit phase, the verifier samples k^2 questions . The questions are described as the matrix $\mathbf{x}^\perp$, where

$$\mathbf{x}^\perp = \begin{pmatrix} x^{11} & \ldots & \ldots & \ldots & x^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x^{k1} & \ldots & \ldots & \ldots & x^{kk} \end{pmatrix},$$

and is sampled according to the joint probability distribution $\left(\pi^\perp\right)^{k^2}$. Each entry $x^{ij} = (x_1^{ij}, \ldots, x_t^{ij}, \ldots, x_n^{ij}) \in \mathcal{X}$ is sampled with $\pi^\perp$. The subscript here denotes the index of the

players. Each player p_d for $d \in [n]$ then receives the question

$$x_d^{\perp} = \begin{pmatrix} x_d^{11} & \dots & \dots & \dots & x_d^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x_d^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_d^{k1} & \dots & \dots & \dots & x_d^{kk} \end{pmatrix}.$$

The verifier also samples the random bit string matrix R with the same dimension as the matrix $x_t^{\perp}$, and sends it to player p_t. Each entry $r^{ij} \in R$ is also sampled randomly from the set $\{0,1\}^{k \cdot |a_t^{ij}|}$ just like each $u^{ij} \in \mathbf{u}$. For each question x^{ij} of the $(i,j)^{th}$ instance of the anchored game, players respond by using the pre-agreed optimal deterministic strategy σ^{ij} except for player p_t. Player p_t still uses the shared optimal deterministic strategy and computes $a_t^{ij} = \sigma_t^{ij}(x_t^{ij})$. Then he commits each bit of a_t^{ij} by following the commit phase of **sBGKW**. This is the reason why both the random bit strings r^{ij}, u^{ij} have length $k \cdot \left|a_t^{ij}\right|$. In other words, for $l \in \left|a_t^{ij}\right|$, prover t computes the following:

$$a_{tl}^{ij} \cdot r_l^{ij} \oplus u_l^{ij},$$

where the subscript l in a_{tl}^{ij} denotes the l^{th} bit of a_t^{ij}, and the subscripts l in r_l^{ij} and in u_l^{ij} indicate the l^{th} set of k bits of r^{ij} and u^{ij}, respectively. To further simplify our notation, we use $\left(a_t^{ij} \cdot r^{ij}\right) \oplus u^{ij}$ to denote the following expression:

$$\left(a_{t1}^{ij} \cdot r_1^{ij}\right) \oplus u_1^{ij} \,||\, \left(a_{t2}^{ij} \cdot r_2^{ij}\right) \oplus u_2^{ij} \,||\, \dots \,||\, \left(a_{t|a_t^{ij}|}^{ij} \cdot r_{|a_t^{ij}|}^{ij}\right) \oplus u_{|a_t^{ij}|}^{ij}.$$

Let $\hat{a}^{ij}$ denote the output of the players in the $(i,j)^{th}$ parallel execution of the protocol,

where

$$\begin{aligned}\hat{a}^{ij} &= \left(\sigma_1^{ij}(x_1^{ij}), \ldots, \sigma_{t-1}^{ij}(x_{t-1}^{ij}),\ \left(\sigma_t^{ij}(x_t^{ij}) \cdot r^{ij}\right) \oplus u^{ij},\ \sigma_{t+1}^{ij}(x_{t+1}^{ij}), \ldots, \sigma_n^{ij}(x_n^{ij})\right) \\ &= \left(a_1^{ij}, \ldots, a_{t-1}^{ij},\ \left(a_t^{ij} \cdot r^{ij}\right) \oplus u^{ij},\ a_{t+1}^{ij}, \ldots, a_n^{ij}\right).\end{aligned}$$

This way, the verifier does not know the output from player p_t yet, since **sBGKW** is hiding. The verifier receives the answer

$$\hat{\mathbf{a}} = \begin{pmatrix} \hat{a}^{11} & \ldots & \ldots & \ldots & \hat{a}^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \hat{a}^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \hat{a}^{k1} & \ldots & \ldots & \ldots & \hat{a}^{kk} \end{pmatrix},$$

at the end of the commit phase.

Player p_{n+1} does not participate in the commit phase at all, but he plays a crucial role during the next part of the protocol. In the unveil phase, player p_{n+1} discloses the binary matrix $\mathbf{B}$ by unveiling all k^2 exclusion sets $E_\sigma^{b_{ij}}$ along with his shared secret matrix $\mathbf{u}$ with player p_t. With each u^{ij}, the verifier can deduce player p_t's output just like in the unveil phase of **sBGKW**. He computes the following:

$$y^{ij} = \hat{a}_t^{ij} \oplus u^{ij} = a_{t1}^{ij} \cdot r_1^{ij} \,||\, a_{t2}^{ij} \cdot r_2^{ij} \,||\, \ldots \,||\, a_{t|a_t^{ij}|}^{ij} \cdot r_{|a_t^{ij}|}^{ij} = y_1^{ij} \,||\, y_2^{ij} \,||\, \ldots \,||\, y_{|a_t^{ij}|}^{ij}.$$

For each position l of the original answer a_t^{ij}, he verifies the value of y_l^{ij}. If $y_l^{ij} = 0^k$, he accepts that $a_{tl}^{ij} = 0$, and if $y_l^{ij} = r_l^{ij}$, he accepts that $a_{tl}^{ij} = 1$. Otherwise, the verifier rejects the commitment if for any l, y_l^{ij} is not equal to r_l^{ij} or 0^k.

For each x^{ij}, he checks if the question is in the exclusion set of that instance, meaning $x^{ij} \in E_\sigma^{b_{ij}}$. If the condition is true, then the verifier simply accepts that instance of the game regardless of what a^{ij} is. This is because if the honest players follow any of the

strategies from the exclusion set, they cannot win. After the reconstruction of the answers, he confirms the validity of the players' answers according to the winning predicate $W^{\perp}$ of the anchored game $G^{\perp}$. Finally, the verifier accepts the commitment of $\mathbf{b}$ if and only if the following two conditions are both satisfied.

- the players succeed in winning all k^2 instances of the anchored game.
- the values of the binary matrix $\mathbf{B}$ satisfy eq. (3.4).

Contrarily, if players lose any instance of the game, or if any of the rows of $\mathbf{B}$ do not sum to $\mathbf{b}$, the commitment is rejected.

So far we have presented all the details of the recipe, we will now summarize it below in the form of a protocol. We will refer to the verifier from hereon as Victor for simplicity. We fix the indices $i, j \in [k]$, and $t, d \in [n]$, where k is the length of each entry $r^{ij} \in R, u^{ij} \in \mathbf{u}$, k^2 is the total number of repetitions of the game, and n is the total number of players of any particular instance of the game.

Anchoring transformation:

To transform a pseudo-telepathy game G to an anchored game $G^{\perp}$, it suffices to follow the 3 steps detailed below to obtain the interesting properties we want in theorem 2.5.1. To keep things simple, we want the properties from the latter part of the theorem.

1. Fix the parameter α according to eq. (2.19), which only requires the number of players n in G.
2. When the verifier samples the question, he obtains $x = (x_1, \ldots, x_n) \in \mathcal{X}$ using the original probability distribution π , and for each of the individual sub question x_d, he independently transforms it into the anchored question $\perp$ with probability α. The resulting set of questions $\mathcal{X}^{\perp}$ and the probability distribution $\pi^{\perp}$ that samples it are as described in definition 2.5.8
3. The winning predicate $W^{\perp}$ is detailed in definition 2.5.8, where the verifier will simply declare that the players have won the game if any of the sampled questions are anchored. Otherwise, he will evaluate the game by using the original winning predicate W such that the players win if $W(x, a) = 1$.

Before commitment:

1. Players determine all optimal deterministic strategies $\vec{\sigma}$, and the corresponding exclusion set $E_{\vec{\sigma}}$ for the original pseudo-telepathy game G. They group the exclusion sets using the equivalence relation. They further partition $E_{\vec{\sigma}}$ into two disjoint sets $E^0_{\vec{\sigma}}$ and $E^1_{\vec{\sigma}}$ according to the requirements stated previously.

2. Players decide on $\mathbf{b}$, the bit value to be committed, and the binary matrix $\mathbf{B}$,

$$\mathbf{B} = \begin{pmatrix} b_{11} & \dots & \dots & \dots & b_{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ b_{k1} & \dots & \dots & \dots & b_{kk} \end{pmatrix},$$

   such that each row satisfies the eq. (3.4). For each b_{ij}, players uniformly sample an exclusion set $E^{b_{ij}}_{\sigma}$ from the set $E^{b_{ij}}_{\vec{\sigma}}$.

3. For each $E^{b_{ij}}_{\sigma}$, players agree on an optimal deterministic strategy σ^{ij} that derives the exclusion set.

4. A special player p_t is chosen according to the hiding criteria of definition 4.1.1, and announce it to the verifier. Player p_t and p_{n+1} share a uniformly random bit string matrix $\mathbf{u}$,

$$\mathbf{u} = \begin{pmatrix} u^{11} & \dots & \dots & \dots & u^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & u^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ u^{k1} & \dots & \dots & \dots & u^{kk} \end{pmatrix},$$

   such that each entry u^{ij} is sampled from $\{0,1\}^{k\cdot|a^{ij}_t|}$ uniformly, and will be used for the **sBGKW** introduced in section 3.2 between them.

Commit phase:

1. Victor randomly samples a question $\mathbf{x}^{\perp}$ according to $\left(\pi^{\perp}\right)^{k^2}$, where

$$\mathbf{x}^{\perp} = \begin{pmatrix} x^{11} & \dots & \dots & \dots & x^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x^{k1} & \dots & \dots & \dots & x^{kk} \end{pmatrix},$$

and each $x^{ij} = \left(x_1^{ij}, \dots, x_n^{ij}\right) \in \mathcal{X}^{\perp}$, and sends $x_d^{\perp}$ to player p_d, where

$$x_d^{\perp} = \begin{pmatrix} x_d^{11} & \dots & \dots & \dots & x_d^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x_d^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_d^{k1} & \dots & \dots & \dots & x_d^{kk} \end{pmatrix}.$$

Victor samples also a bit string matrix R such that $\dim(R) = \dim(x_t^{\perp})$ with each entry r^{ij} sampled from $\{0,1\}^{k \cdot |a_t^{ij}|}$, and sends it to p_t.

2. Players answer with

$$\hat{\mathbf{a}} = \begin{pmatrix} \hat{a}^{11} & \dots & \dots & \dots & \hat{a}^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \hat{a}^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \hat{a}^{k1} & \dots & \dots & \dots & \hat{a}^{kk} \end{pmatrix},$$

$$\begin{aligned} \hat{a}^{ij} &= \left(\sigma_1^{ij}\left(x_1^{ij}\right), \dots, \sigma_{t-1}^{ij}\left(x_{t-1}^{ij}\right),\ \left(\sigma_t^{ij}\left(x_t^{ij}\right) \cdot r^{ij}\right) \oplus u^{ij},\ \sigma_{t+1}^{ij}\left(x_{t+1}^{ij}\right), \dots, \sigma_n^{ij}\left(x_n^{ij}\right)\right) \\ &= \left(a_1^{ij}, \dots, a_{t-1}^{ij},\ \left(a_t^{ij} \cdot r^{ij}\right) \oplus u^{ij},\ a_{t+1}^{ij}, \dots, a_n^{ij}\right). \end{aligned}$$

Unveil phase:

1. Player p_{n+1} sends both $\mathbf{B}$ and $\mathbf{u}$ to Victor, along with all k^2 different exclusion sets $E_\sigma^{b_{ij}}$.

2. Victor recovers the original answers from player t by performing the following procedures similar to that of the unveil stage in **sBGKW**. For each $\hat{a}_t^{ij}$, he computes

$$\begin{aligned} y^{ij} = \hat{a}_t^{ij} \oplus u^{ij} &= a_{t1}^{ij} \cdot r_1^{ij} \mid\mid a_{t2}^{ij} \cdot r_2^{ij} \mid\mid \ldots \mid\mid a_{t|a_t^{ij}|}^{ij} \cdot r_{|a_t^{ij}|}^{ij} \\ &= y_1^{ij} \mid\mid y_2^{ij} \mid\mid \ldots \mid\mid y_{|a_t^{ij}|}^{ij} \end{aligned}.$$

   Then, for each position $l \in [|a_t^{ij}|]$,

$$a_{tl}^{ij} = \begin{cases} 0, \text{ if } y_l^{ij} = 0^k \\ 1, \text{ if } y_l^{ij} = r_l^{ij} \end{cases}.$$

   Victor rejects the commitment if for any l and $i, j \in [k]$, the value of y_l^{ij} is not captured from the above equation.

3. For any $i, j \in [k]$, if $x^{ij} \in E_\sigma^{b_{ij}}$, Victor will accept the $(i,j)^{th}$ instance of the game, since the question is in the exclusion set. This means that the output of $W^\perp\left(x^{ij}, a^{ij}\right) = 1$ regardless of what the actual value of a^{ij} is.

4. Victor accepts the commitment of $\mathbf{b}$ if both of the following equations holds true

$$\begin{aligned} \left(W^\perp\right)^{k^2}\left(\mathbf{x}^\perp, \mathbf{a}\right) &= \prod_{i=1}^{k}\prod_{j=1}^{k} W^\perp\left(x^{ij}, a^{ij}\right) = 1, \\ \forall i \in [k],\ &\bigoplus_{j=1}^{k} b_{ij} = \mathbf{b}. \end{aligned}$$

   Otherwise, he rejects the commitment.

The protocol presented above follows a very nice structure where in the view of each prover including the extra prover p_{n+1}, they only interact with the verifier once just like in the parallel repeated game. This way, the theorem 2.5.1 of the anchored game applies directly. More precisely, the winning probability of all k^2 parallel instances of the anchored game for classical players is exponentially small in terms of k^2. This property will be useful when we prove that the bit commitment protocol is binding in the following chapter.

The fact that the unveiler does not know any of the questions asked during the protocol proves crucial in the analysis of the binding property as well. It actually forces the unveiler to disclose the matrix $\mathbf{u}$ exactly as it is, since the commitment protocol **sBGKW** conducted between him and player t is binding. In the occasion where he decides to disclose a different value $u^{ij'}$, there is a high probability that the unveiling of p_t's output is invalid. Even if the unveiling is successful, the output that the verifier uncovers by using $u^{ij'}$ can lead to an unsatisfying answer for the anchored game as well.

We can obtain variations of the protocol with how we apply the commitment protocol **sBGKW** with a tradeoff between efficiency and security. In an earlier iteration of this work, we simply hide player t's output a_t^{ij} by performing an exclusive-or of his output with the shared random bit string u^{ij} in each $(i,j)^{th}$ instance of the game. This makes the protocol more efficient, but the proof that the resulting protocol is binding against classical provers is not as straightforward and remains an open problem. On the other hand, instead of only hiding one player's output with the commitment scheme **sBGKW**, we can hide all of the players' output the same way. The resulting protocol is perfectly hiding without the need to satisfy the hiding condition detailed in section 4.1.2. This is simply because the verifier only receives what appears as uniformly random strings from each prover, and hence cannot deduce which optimal deterministic strategy is used to produce players' answers. Due to this, we can run the above protocol with only 1 instance of the original game G instead of k^2 instances of the anchored game $G^{\perp}$ and still obtain hiding. While we settled on this final version of the protocol, it is up to the readers of this work to decide which version is

best suited for their use cases.

It is also worth noting that we can make this proposed protocol completely secure even against provers that possess the necessary entanglements. We can instead commit to the optimal deterministic strategies used during the commit phase and make the unveiler p_{n+1} disclose them in each instance of the anchored game. With the knowledge of the deterministic strategies, players cannot provide any other answers that are not a result of the direct application of said strategies. This does not allow room for players to use alternative strategies including quantum winning ones to cheat the binding condition of the protocol because of this.

In the next chapter, we will discuss the hiding and binding properties for the proposed protocol and present proofs for their security. We will also present concrete applications of this protocol using the Magic Square game introduced in the beginning of this chapter as well as the Mermin-GHZ game.

Chapter 4

Security Analysis and Applications

As mentioned previously in section 1.3, the security of the bit commitment scheme is analyzed from two different angles. The scheme is secure against provers for the honest verifier if it is *binding*. That is, the provers cannot have a non-negligible probability of unveiling either values of the bit b at the same time according to the definition 2.7.3. This also means that the provers cannot delay choosing the bit value to be committed until the unveiling stage as in some common attack scenarios for quantum bit commitment schemes. In those scenarios, provers commit to a superposition of both $|0\rangle$ and $|1\rangle$, and then choose to reveal either state after the commitment. Traditionally, provers cheat the binding condition successfully if they convince the verifier that the commitment is for a bit value that provers did not agree to during the commit stage. As demonstrated in section 2.7.1, this definition of binding is not sufficient. Instead, we say a bit commitment scheme is binding if the provers cannot win the non-binding game with probability non-negligibly better than randomly guessing the value chosen by the verifier according to definition 2.7.4.

On the other hand, a bit commitment scheme is secure against a verifier for the honest provers if it is *hiding*. This means that the verifier does not extract any useful information about the committed bit b from the interactions with provers before the unveil phase. We declare that a verifier successfully cheats the hiding condition of the commitment scheme if

he acquires a non-negligible bias about b before the unveil stage.

In this chapter, we prove that a bit commitment scheme built following our proposed protocol presented in section 3.3.2 is both hiding and binding classically. We first show the non-binding game in the context of our protocol, and give a brief proof that the bit commitment scheme is indeed binding. For the scheme to be hiding, we present a simple criterion that needs to be satisfied. After that, we present applications of our protocol using the Magic Square game presented in detail in section 3.1.1 and the Mermin-GHZ game which will be presented in section 4.2.1.

4.1 Binding

We now analyze the security of a bit commitment scheme constructed using our proposed protocol against malicious classical provers. To do so, we define the non-binding game according to section 2.7.1 under the context of our protocol. The non-binding game and the bit commitment scheme share the same settings. As mentioned in section 2.5, we restrict our attention to deterministic strategies for classical provers. Hence, for a non-binding game built with a pseudo-telepathy game of n players, we assume that all classical provers behave deterministically as well. This assumption also includes player $n+1$ since he does not learn any of the questions given to the rest of the players during the games. Thus for any instance of the game, he cannot adapt his unveiling strategy in the hope to both correctly answer the verifier's query and unveil an exclusion set that belongs to the opposite side except with negligible probability. We will formalize this idea proving that classical players cannot win the non-binding game with probability significantly better than randomly guessing which bit value will be chosen by the verifier.

4.1.1 Non-binding game

The set up for the non-binding game is exactly the same as our proposed protocol in 3.3.2. The players first apply the anchoring transformation to the pseudo-telepathy game G with a parameter α that the bit commitment scheme is built upon. They then determine all the optimal deterministic strategies $\vec{\sigma}$ from the original game G and their corresponding exclusion sets $E_{\vec{\sigma}}$. They bipartition the exclusion sets such that the two sets $E^0_{\vec{\sigma}}$ and $E^1_{\vec{\sigma}}$ are disjoint and they are non-empty. A binary matrix $\mathbf{B}$ with k^2 entries is chosen by the provers such that $\mathbf{B}$ either corresponds to the unveiling of a single bit $\mathbf{b}$ by satisfying the eq. (3.4), or it corresponds to a strategy to unveil both values. Regardless of what $\mathbf{B}$ is, for each entry b_{ij}, the players choose an exclusion set from the side of $E^{b_{ij}}_{\vec{\sigma}}$, and subsequently an optimal deterministic strategy σ^{ij} that derives the same exclusion set. Player t and player $n+1$ sample a uniformly random bit string matrix $\mathbf{u}$ such that each entry $u^{ij} \in \{0,1\}^{k\cdot|a_t^{ij}|}$. The position t is chosen by the provers to satisfy the hiding condition of definition 4.1.1 and will be announced to the verifier.

Next, in the query phase, the verifier samples question according to the distribution $\left(\pi^{\perp}\right)^{k^2}$ of the k^2 repeated anchored game $(G^{\perp})^{k^2}$. He interrogates all provers except prover $n+1$. This is exactly the same as the commit phase in the protocol.

Finally, in the challenge phase, the prover p_{n+1} is given a bit $\mathbf{b}'$ to unveil. In order to win the non-binding game, he needs to reveal the appropriate matrix $\mathbf{B}$ and also unmask prover p_t's answers in such a way that all k^2 instances of the anchored games are won.

Both the query phase and the challenge phase are describe below in more details.

Query phase:

1. Victor randomly samples a question $\mathbf{x}^{\perp}$ according to $\left(\pi^{\perp}\right)^{k^2}$, where

$$\mathbf{x}^{\perp} = \begin{pmatrix} x^{11} & \dots & \dots & \dots & x^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x^{k1} & \dots & \dots & \dots & x^{kk} \end{pmatrix},$$

and each $x^{ij} = \left(x_1^{ij}, \dots, x_n^{ij}\right) \in \mathcal{X}^{\perp}$, and sends $x_d^{\perp}$ to player p_d, where

$$x_d^{\perp} = \begin{pmatrix} x_d^{11} & \dots & \dots & \dots & x_d^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x_d^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_d^{k1} & \dots & \dots & \dots & x_d^{kk} \end{pmatrix}.$$

Victor samples a random matrix R such that $\dim(R) = \dim(x_t^{\perp})$, and each $r^{ij} \in R$ is sampled from the set $\{0,1\}^{k \cdot |a_t^{ij}|}$, and sends it to p_t.

2. Players answer with

$$\hat{\mathbf{a}} = \begin{pmatrix} \hat{a}^{11} & \dots & \dots & \dots & \hat{a}^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \hat{a}^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \hat{a}^{k1} & \dots & \dots & \dots & \hat{a}^{kk} \end{pmatrix},$$

$$\begin{aligned} \hat{a}^{ij} &= \left(\sigma_1^{ij}\left(x_1^{ij}\right), \dots, \sigma_{t-1}^{ij}\left(x_{t-1}^{ij}\right),\ \left(\sigma_t^{ij}\left(x_t^{ij}\right) \cdot r^{ij}\right) \oplus u^{ij},\ \sigma_{t+1}^{ij}\left(x_{t+1}^{ij}\right), \dots, \sigma_n^{ij}\left(x_n^{ij}\right)\right) \\ &= \left(a_1^{ij}, \dots, a_{t-1}^{ij},\ \left(a_t^{ij} \cdot r^{ij}\right) \oplus u^{ij},\ a_{t+1}^{ij}, \dots, a_n^{ij}\right). \end{aligned}$$

Challenge phase:

1. The verifier randomly samples $\mathbf{b'} \in \{0,1\}$, and sends it to the prover $n+1$.
2. p_{n+1} announces $\mathbf{u}$ and all k^2 different exclusion sets $E_\sigma^{b_{ij}}$ to the verifier.
3. The verifier uncovers each of player p_t's original answers $\hat{a}_t^{ij}$ by computing

$$y^{ij} = \hat{a}_t^{ij} \oplus u^{ij} = a_{t1}^{ij} \cdot r_1^{ij} \,||\, a_{t2}^{ij} \cdot r_2^{ij} \,||\, \ldots \,||\, a_{t|a_t^{ij}|}^{ij} \cdot r_{|a_t^{ij}|}^{ij} = y_1^{ij} \,||\, y_2^{ij} \,||\, \ldots \,||\, y_{|a_t^{ij}|}^{ij}.$$

   Then, for each position $l \in [|a_t^{ij}|]$,

$$a_{tl}^{ij} = \begin{cases} 0, \text{ if } y_l^{ij} = 0^k \\ 1, \text{ if } y_l^{ij} = r_l^{ij} \end{cases}.$$

   The verifier rejects the commitment if for any l and $i,j \in [k]$, the value of y_l^{ij} is not captured from the above equation.
4. For any $i,j \in [k]$, if $x^{ij} \in E_\sigma^{b_{ij}}$, the verifier will accept the $(i,j)^{th}$ instance of the game, since the question is in the exclusion set. This means that the output of $W^\perp\left(x^{ij}, a^{ij}\right) = 1$ regardless of what the actual value of a^{ij} is.
5. The verifier accepts the opening of $\mathbf{b'}$ if both of the following equations hold true

$$\left(W^\perp\right)^{k^2}\left(\mathbf{x}^\perp, \mathbf{a}\right) = \prod_{i=1}^{k}\prod_{j=1}^{k} W^\perp\left(x^{ij}, a^{ij}\right) = 1,$$
$$\forall i \in [k],\ \bigoplus_{j=1}^{k} b_{ij} = \mathbf{b'},$$

   where $W^\perp$ is the winning predicate of the anchored game, meaning that if any of the individual questions in x^{ij} are anchored, then it evaluates to 1. Otherwise, the provers lose the non-binding game.

Remark that during the non-binding game, every player exchanges with the verifier only once. Each one obtains his inputs and nothing more, then outputs his own answers. As mentioned earlier, since the query phase of the non-binding game and the commit phase of the protocol are exactly the same, provers have no way of knowing whether they are executing the bit commitment scheme or playing the non-binding game. Only prover p_{n+1} knows if they have been participating in the non-binding game during the challenge phase when he is given a binary input from the verifier. However, player $n+1$ cannot communicate with the others during the unveiling and at this point of the non-binding game, players can no longer change their outputs. This implies that, due to the lack of information, classical players cannot adapt their strategy if they want to open a valid commitment in the bit commitment scheme and also win the non-binding game. Finally, the anchoring transformation does not affect players' success probability of the non-binding game as long as we keep the anchoring parameter α within a reasonable bound. For example, the following should be unlikely to occur: all k entries b_{ij} in a row of the matrix $\mathbf{B}$ are accepted freely because the questions of those instances are anchored.

Theorem 4.1.1. *Classical players win the non-binding game of a bit commitment scheme implemented using the protocol in 3.3.2 with probability at most $\frac{1}{2}+\epsilon(k)$, where $\epsilon(k)$ is negligible in terms of the number of repetitions of the anchored game during the bit commitment scheme.*

Proof. First we show how honest deterministic provers achieve a success probability of $\frac{1}{2}$ in the non-binding game. They choose a bit value $\mathbf{b}$ and a binary matrix $\mathbf{B}$ such that each row xors to $\mathbf{b}$ according to eq. (3.4). With probability $\frac{1}{2}$, the bit value that the verifier has chosen to force the unveiler to disclose is the same as the bit value $\mathbf{b}$ that his peers have committed. They win the non-binding game in this case by simply following their strategy deterministically. In the other case where the bit value chosen by the verifier is not the same as the one committed, the honest provers will lose. We show below that dishonest provers will not fare better than the honest provers with non-negligible probability in the

non-binding game in both cases.

To win the non-binding game, let us suppose that provers jointly would like to be able to unveil a bit value $\mathbf{b}$ both as 0 and as 1. To do so, assume that provers have two binary matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ that prover p_{n+1} can disclose to unveil $\mathbf{b} = 0$, and $\mathbf{b} = 1$, respectively. This includes the scenario where the provers honestly commit to $\mathbf{B_b}$, and disclose $\mathbf{B_{b'}}$ afterwards. It also includes the scenario where provers commit to nothing by sending a binary matrix that is neither $\mathbf{B}_0$ nor $\mathbf{B}_1$ and only decide on what $\mathbf{b}$ to unveil at the unveil stage by changing the necessary entries b_{ij} in order to disclose $\mathbf{B_b}$. Recall that the binary matrices $\mathbf{B}_0, \mathbf{B}_1$ have to satisfy eq. (3.4) in order to be accepted by the verifier. This means that each row of $\mathbf{B}_0$ needs to sum to 0, while each row of $\mathbf{B}_1$ has a binary sum of 1. Consequently, we can assume that each row of $\mathbf{B}_0$ differs from that of $\mathbf{B}_1$ in at least 1 position j, for $i, j \in [k]$, such that the value of b_{ij} in row i and position j changes the binary sum of that row in both matrices. Without loss of generality, we consider that the matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ differ in the same position j in each row i.

To prove that provers cannot win the non-binding game with probability much better than $\frac{1}{2}$, we keep the assumptions of the unveiler p_{n+1} to a minimum, and consider his actions as general as possible. For each $(i, j)^{th}$ instance of the protocol, the prover $n + 1$ discloses one of the following triples in order to unveil an entry b_{ij} of the binary matrix $\mathbf{B}_b$:

$$\begin{matrix} \left(u_0, b_0, E^0_{\sigma_0}\right)_{ij} \\ \left(u_1, b_1, E^1_{\sigma_1}\right)_{ij} \end{matrix},$$

where the subscript ij applies to all the elements including the optimal deterministic strategies σ_b. The elements u_0, u_1 are the values of the matrix $\mathbf{u}$ and similarly, $b_0 = 0$ and $b_1 = 1$ are the entries b_{ij} from $\mathbf{B}_b$. This means that prover p_{n+1} can change prover t's output adaptively with either u_0 or u_1 depending on his input. He also discloses the exclusion sets $E^0_{\sigma_0}$ or $E^1_{\sigma_1}$ accordingly, where $E^b_{\sigma_b} \in E^b_{\vec{\sigma}}$ with the implication that the strategy σ_b is used to produce the provers' output of that instance. This models the adaptive behaviour

of the unveiler who wants to maximize his chance of winning the non-binding game by always answering the triple that can satisfy the winning conditions. We show below that this adaptive strategy does not offer a significant advantage over the deterministic one for dishonest provers.

Recall that each entry of the matrix **u** is used as the shared random bit string between prover $n+1$ and prover t to execute the bit commitment protocol **sBGKW** for hiding prover t's output a_t^{ij} from the verifier until the unveil phase. To open a different commitment of a_t^{ij}, prover $n+1$ needs to break the binding property of **sBGKW** by disclosing either u_0 or u_1 that is different than u^{ij} used by p_t. But, this lowers the winning probability of the non-binding game since **sBGKW** is proven to be binding against both classical and quantum provers. As long as the unveiler does not share a correlation as powerful as the **PR** box with prover p_t, they cannot break binding of **sBGKW** except with exponentially small probability. Hence, it is in their best interest that prover p_{n+1} discloses his bit string matrix **u** shared with prover p_t exactly as it is. This is a reasonable assumption since we are only considering classical players in this proof. Consequently, we can safely assume that prover p_{n+1} discloses $u_0 = u_1 = u^{ij}$ regardless of his input and that the verifier will always uncover the provers' output in each instance of the protocol.

With this in mind, we can now focus our attention to the last two elements of the triple that the prover p_{n+1} discloses. In fact, we only need to consider the exclusion sets since the bipartition of the sets is known to the verifier. The value of b_{ij} in the matrix **B** is indicated by the superscript of the disclosed exclusion set.

Let us suppose that the provers $p_1, \ldots, p_n$ follow an agreed upon deterministic strategy s during the challenge phase to answer the query from the verifier. Note that the strategy s needs not be optimal as assumed in the actual bit commitment protocol. But, the strategy s has a corresponding exclusion set E_s since no classical winning strategy exists for a pseudo-telepathy game. Exclusion sets are optimal when they are derived from optimal deterministic strategies. All optimal exclusion sets have the same cardinality $(1-\omega_c(G))\cdot|\mathcal{X}|$.

If s is not optimal, then $|E_s| > (1 - \omega_c(G)) \cdot |\mathcal{X}|$. This means that for an optimal exclusion set to be different from another one, there must exist at least an input x that is in the former exclusion set and not in the latter one, and vice versa. This implies the following statement with S being the set of all classical strategies, and $\vec{\sigma}_b$ the set of all optimal deterministic strategies that have corresponding exclusion sets in $E^b_{\vec{\sigma}}$.

$$\forall s \in S, \exists b \in \{0,1\}, \forall \sigma_b \in \vec{\sigma_b}, \{x | x \in E_s \wedge x \notin E^b_{\sigma_b}\} \neq \emptyset$$

This is true as long as the strategy s is not the same as the optimal deterministic strategy σ_b. In other words, no matter which strategy s the provers used to answer their queries, there always exists an input x that belongs to the exclusion set of that same strategy, but not the disclosed exclusion set $E^b_{\sigma_b}$. This implies that with probability at least $\frac{1}{|\mathcal{X}|}$, the provers fail the winning condition of the game when the verifier samples an input x satisfying the above statement, since $\forall x \in E_s, W^{\perp}(x, s(x)) = 0$. Thus, each time that the unveiler decides to disclose an exclusion set that is not agreed upon with his peers, he wins the underlying game with probability strictly less than 1.

Per our assumptions that $\mathbf{B}_0$ and $\mathbf{B}_1$ differ in at least 1 entry in each row, in the case where provers commit to $\mathbf{B}_b$ and are forced to unveil $\mathbf{B}_{\bar{b}}$, prover $n+1$ needs to disclose at least k different entries. In the other case where provers commit to a binary matrix that contains some rows from $\mathbf{B}_0$ and some other from $\mathbf{B}_1$, there are at least $k/2$ entries that the unveiler needs to disclose differently when he wants to unveil one of the 2 values. Along with the fact that we are dealing with constant input size pseudo-telepathy games, this results in an exponential decrease in terms of $k/2$ for their overall winning probability of the non-binding game. ■

Theorem 4.1.2. *Quantum provers that share the appropriate entanglements for the underlying pseudo-telepathy game of the non-binding game can win with probability* 1.

Proof. With the shared entangled states, players use the quantum winning strategy of the

underlying pseudo-telepathy game to produce their outputs. Player $n+1$ discloses the shared bit string matrix $\mathbf{u}$ intact for the verifier to recover the outputs. Given the input $\mathbf{b}$ from the verifier, prover p_{n+1} chooses k^2 different exclusion sets that correspond to the binary matrix $\mathbf{B_b}$ and announces them back. This satisfies both of the conditions for the verifier to accept the unveiling of $\mathbf{b}$ since players' outputs always satisfy the winning predicate $\pi^{\perp}$, and matrix $\mathbf{B_b}$ satisfies eq. (3.4). ■

4.1.2 Hiding

We present below the criteria for a bit commitment scheme constructed using the protocol in 3.3.2 to be statistically hiding. That means a malicious verifier cannot learn enough information throughout his exchanges with provers during the protocol to determine the committed bit $\mathbf{b}$ exactly with negligibly higher chance than randomly guessing.

In the context of our bit commitment scheme, the provers commit to a binary matrix $\mathbf{B}$ that satisfies the eq. (3.4). This means that in order to learn the value of $\mathbf{b}$, the verifier just needs to obtain all the values $(b_{i1}, b_{i2}, \ldots, b_{ik})$ in $\mathbf{B}$ for any of the rows i, where $i \in [k]$. But this implies that the verifier can gain enough information on all k independent b_{ij} during k instances of the underlying anchored pseudo-telepathy game $G^{\perp}$. On the other hand, if the verifier fails to learn just one b_{ij} for each row i, he still cannot reconstruct $\mathbf{b}$ faithfully since the values are xored together, and the values in each rows are independent as well. This gives us a clear minimum criteria for the bit commitment scheme to be hiding.

Let us now turn our attention to how the verifier can actually obtain information about each committed value b_{ij} during the $(i,j)^{th}$ instance of the protocol. One way for the verifier to do so is to learn which optimal deterministic strategy $\sigma \in \vec{\sigma}$ the provers use during the $(i,j)^{th}$ instance of the game $G^{\perp}$. Provers share the knowledge of which exclusion set belongs to $E^0_{\vec{\sigma}}$ or $E^1_{\vec{\sigma}}$ with the verifier prior to the protocol. This means that learning the strategy σ can help identify the exclusion set that the strategy agrees with. But, the verifier needs to learn more than one input and output pair (x^{ij}, a^{ij}) during the game in order to filter

out which strategy can produce the same results. This proves to be difficult, given that the player p_t's output is masked by the secret string u^{ij} shared with player p_{n+1} as shown during the commit phase of the protocol. The players also choose an independent strategy per instance of the game, meaning that the chance of a strategy being reused in multiple instances of the game is low. Furthermore, if any of the questions during the $(i,j)^{th}$ instance of the game is anchored, the verifier cannot obtain a valid pair of $(x^{ij}, \hat{a}^{ij})$. Instead, for each pair of $(x^{ij}, \hat{a}^{ij})$ that is not anchored, the verifier can only narrow down a list of strategies that cohere with the resultant input and output relation.

The last observation leads to another way for the verifier to determine the committed value b_{ij}. There exists a scenario where he can find out b_{ij} with certainty if the list of strategies that can produce a specific pair of $(x^{ij}, \hat{a}^{ij})$ all belong to the same side of the partitions: $E_{\vec{\sigma}}^{b_{ij}}$. This means that there does not exist a single strategy σ' that agrees with the exclusion sets in $E_{\vec{\sigma}}^{\overline{b_{ij}}}$ such that $\sigma'(x^{ij}) = \hat{a}^{ij}$. This is the intuition behind the hiding criteria presented below, such that if the bipartition of exclusion sets for a game satisfy these criteria, then the resultant bit commitment scheme following the protocol in 3.3.2 is statistically hiding.

Definition 4.1.1 (hiding criteria)**.** *Given a pseudo-telepathy game G, and a bit commitment scheme $\mathbf{C}$ built using G by following the protocol 3.3.2. We define the hiding criteria for $\mathbf{C}$ for a specific index $t \in [n]$ to be the following.*

For any $(i,j)^{th}$ instance of the game in $\mathbf{C}$, such that for all $x^{ij} = (x_1^{ij}, \ldots, x_n^{ij}) \in \mathcal{X}$($x^{ij}$ is not anchored),

- *there exists $a^{ij} = (a_1^{ij}, \ldots, a_t^{ij}, \ldots, a_n^{ij}) \in \mathcal{A}$, and $a^{ij\prime} = (a_1^{ij}, \ldots, a_t^{ij\prime}, \ldots, a_n^{ij}) \in \mathcal{A}$ such that a^{ij} may only differ with $a^{ij\prime}$ at position t, and $W(x^{ij}, a^{ij}) = W(x^{ij}, a^{ij\prime}) = 1$,*
- *if there exists $\sigma \in \vec{\sigma}$, such that $E_\sigma \subseteq E_{\vec{\sigma}}^{b_{ij}}$ and $\sigma(x^{ij}) = a^{ij}$, then there exist at least another $\sigma' \in \vec{\sigma}$, such that $\sigma'(x^{ij}) = a^{ij\prime}$, and that $E_{\sigma'} \subseteq E_{\vec{\sigma}}^{\overline{b_{ij}}}$, where $b_{ij} \oplus \overline{b_{ij}} = 1$.*

In our proposed protocol where we use the commitment scheme **sBGKW** to hide the designated prover t's output, for $t \in [n]$, we find such an index t and a bipartition of the exclusion sets such that the above hiding criteria is satisfied. If the criteria is satisfied for more than 1 value of t, then the choice of t can be made arbitrary.

Theorem 4.1.3 (statistically hiding)**.** *For a bit commitment scheme built using a pseudo-telepathy game G following the protocol 3.3.2, if there exists an index $t \in [n]$ such that the hiding criteria of definition 4.1.1 is satisfied, then the resultant bit commitment scheme is statistically hiding.*

Proof. For a given index t, and a pair of input and output (x^{ij}, a^{ij}), we say a strategy σ exists if and only if $\sigma(x^{ij}) = a^{ij}$ or $\sigma(x^{ij}) = a^{ij'}$ (where $a^{ij'}$ is as defined above), and $W(x^{ij}, a^{ij}) = W(x^{ij}, \sigma(x^{ij})) = 1$. We also assume from here throughout the rest of the proof that exclusion sets E_σ and $E_{\sigma'}$ that agree with the strategies σ and σ' do not belong to the same side of the bipartition. Let us look at each of the following scenarios.

In the case where for a specific pair (x^{ij}, a^{ij}) during the $(i,j)^{th}$ parallel repetition of the protocol, σ exists while σ' does not, the verifier will repeatedly query x^{ij} in an effort to discover whether $b_{ij} = 0$ or $b_{ij} = 1$. Once the players output either a^{ij} or $a^{ij'}$, then the verifier can be sure of the value of b_{ij}. In the case where neither strategy exists for a pair (x^{ij}, a^{ij}), it is obvious that the verifier learns nothing about b_{ij} from them.

In the case where they both exist, regardless of the sizes of $|\{E_\sigma \subseteq E_{\vec{\sigma}}^{b_{ij}} \text{ s.t. } \sigma(x^{ij}) = a^{ij}\}|$ and $|\{E_{\sigma'} \subseteq E_{\vec{\sigma}}^{\overline{b_{ij}}} \text{ s.t. } \sigma'(x^{ij}) = a^{ij'}\}|$, as long as they are positive, we can obtain hiding. If $|\{E_\sigma \subseteq E_{\vec{\sigma}}^{b_{ij}} \text{ s.t. } \sigma(x^{ij}) = a^{ij}\}| = |\{E_{\sigma'} \subseteq E_{\vec{\sigma}}^{\overline{b_{ij}}} \text{ s.t. } \sigma'(x^{ij}) = a^{ij'}\}|$ for all $x^{ij} \in \mathcal{X}, a^{ij}, a^{ij'} \in \mathcal{A}$ such that $W(x^{ij}, a^{ij}) = W(x^{ij}, a^{ij'}) = 1$, then we get perfect hiding. However, if they are positive and different, the verifier learns a little bit about b_{ij} each time he uses such an x^{ij} and obtains either a^{ij} or $a^{ij'}$. Despite this, the information that he gains in this scenario is not sufficient for him to have an advantage in guessing correctly the actual committed bit value $\mathbf{b}$. Since in order to obtain $\mathbf{b}$, he has to guess correctly each and every b_{ij} in all of the rows of the binary matrix $\mathbf{B}$, for $\mathbf{b} = \bigoplus_{j=1}^{k} b_{ij} = b_{i1} \oplus \ldots \oplus b_{ik}$ for any $i \in [k]$.

The success probability for the verifier to correctly guess $\mathbf{b}$ in each row of $\mathbf{B}$ decays to $\frac{1}{2}$ exponentially as soon as there are $\Omega(k)$ b_{ij} that he does not know for certain in that row. This is simply due to the properties of the xor operation. In the case where this happens for all the rows, meaning the verifier only learns a negligible bias on what $\mathbf{b}$ is, then this will not be sufficient to help the verifier to guess correctly $\mathbf{b}$. ■

Hence, a bit commitment protocol constructed using our result is binding against classical provers according to theorem 4.1.1, and it is hiding according to theorem 4.1.3 if the partition of the exclusion sets satisfies the hiding criteria of definition 4.1.1. The protocol is fully non-binding against quantum provers according to theorem 4.1.2. We wish to investigate the consequence of building more complex cryptographic protocols using the bit commitment schemes resulting from this work with these properties. An immediate question in this nature is the application of these bit commitment protocols in Zero-Knowledge proofs. Furthermore, in this work we focused solely on pseudo-telepathy games that always have a quantum winning strategy. An interesting question is whether we can extend this work to other nonlocal games that do not have quantum winning strategies, whereas no-signalling provers that share nonlocal resources as powerful as the PR boxes can win with probability 1. This includes the CHSH game, the GYNI game [ABB$^+$10], and the RGB game [CRC19] where a quantum strategy always performs better than the classical counterpart. We believe that we have an affirmative answer to this question. It suffices to change the accepting criteria of the verifier for the bit commitment protocol. We do not require the provers to always win the underlying games in the commitment, instead they need to win at least $\omega^*(G) \cdot k^2$ instances of the game, where $\omega^*(G)$ is the quantum value. The resulting bit commitment schemes will be binding against both classical and quantum provers but not against non-signalling provers.

4.2 Applications

After analyzing the binding and hiding properties of the bit commitment schemes built using the recipe presented in section 3.3.2, we now present its applications using two well-known pseudo-telepathy games. One of which we have already seen in detail in the previous chapter. The other is the Mermin-GHZ game.

4.2.1 Mermin-GHZ game

The Mermin-GHZ game is a special instance of the parity game with three players: Alice, Bob and Charles. Each player gets a single bit input from a verifier, with the promise that all inputs have a parity of 0. If we label the players' input with x_1, x_2, x_3 then the promise is the following probability distribution

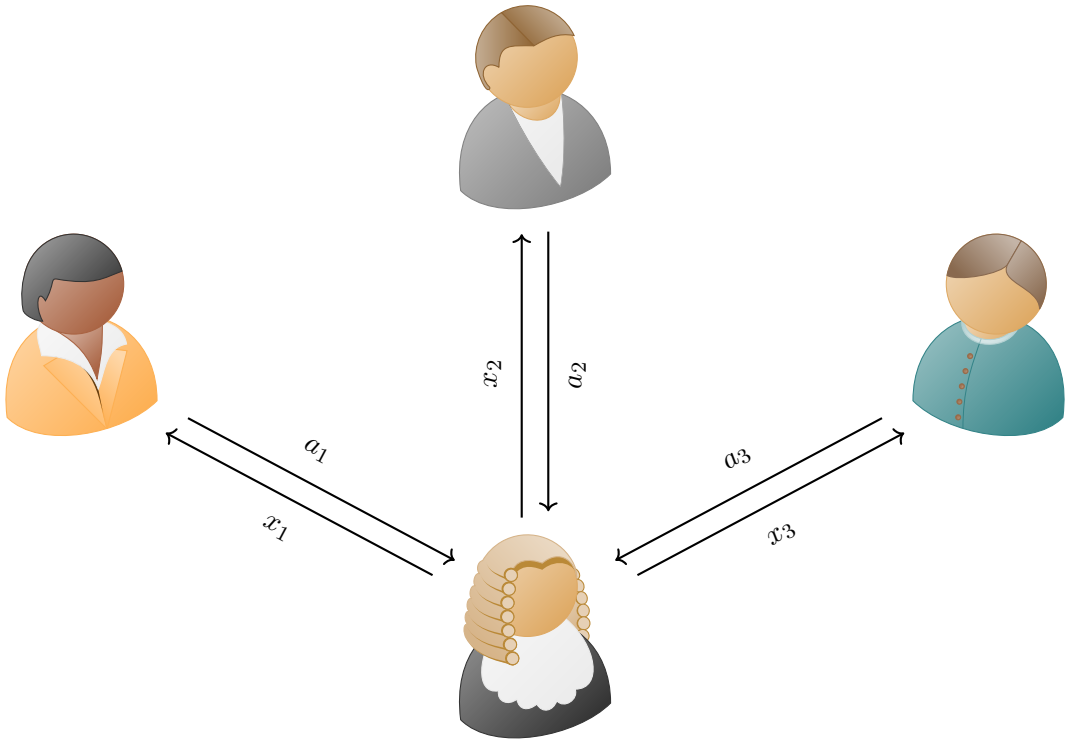
$$\Pr(x_1, x_2, x_3) = \begin{cases} \frac{1}{4}, \text{ if } x_1 \oplus x_2 \oplus x_3 = 0 \\ 0, \text{ otherwise} \end{cases},$$

which restricts the input sets to be $\mathcal{X} \in \{(0,0,0), (1,1,0), (1,0,1), (0,1,1)\}$. Each player ouputs a single bit response a_1, a_2, a_3, respectively. The winning condition of the game is that the parity of the output bits is equal to the disjunction of the input bits:

$$a_1 \oplus a_2 \oplus a_3 = x_1 \vee x_2 \vee x_3.$$

The game is depicted in fig. 4.1 on the following page, with Alice, Bob and Charles from left to right, and the winning condition highlighted below the judge.

At a first glance, it appears that this game can be won classically with just 3 binary inputs and outputs. But, just like the CHSH game and the Magic Square game, no classical winning strategy exists for this game. It can be demonstrated with the following analysis. Let us consider a deterministic strategy, where a mapping between $\mathcal{X}$ and $\mathcal{A}$ exists such



$$a_1 \oplus a_2 \oplus a_3 = x_1 \vee x_2 \vee x_3$$

Figure 4.1 – The Mermin-GHZ game.

that z_i^0 is the response for player i on input $x_i = 0$, and similarly for z_i^1. Then, for input $(0, 0, 0)$, the parity of the output (z_1^0, z_2^0, z_3^0) must be even. As for the rest of the inputs $(0, 1, 1), (1, 0, 1)$ and $(1, 1, 0)$, the exclusive-or of the corresponding outputs all need to be 1

in order to satisfy the winning condition. These sets of constraints are illustrated below.

$$z_1^0 \oplus z_2^0 \oplus z_3^0 = 0$$
$$z_1^0 \oplus z_2^1 \oplus z_3^1 = 1$$
$$z_1^1 \oplus z_2^0 \oplus z_3^1 = 1$$
$$z_1^1 \oplus z_2^1 \oplus z_3^0 = 1$$

A contradiction can then be obtained by summing the four constraints. The sum on the left-hand side is even, while the sum of the right-hand side is odd, which is impossible. The maximum value of the game is $\omega_c(G) = \frac{3}{4}$. This means that any optimal deterministic strategy will satisfy 3 out of the 4 constraints from above.

We can use the local deterministic strategy from definition 2.5.4 on page 27 to achieve the maximum value of the game. This family of strategies assigns two parameters $(c_i^0, c_i^1) \in \{0,\ 1\}$ to each player i, with $i \in \{1, 2, 3\}$ to represent Alice, Bob, and Charles. Upon receiving inputs x_i each player computes the following to find output a_i

$$a_i = s_i(x_i, c_i^0, c_i^1) = x_i \cdot c_i^0 \oplus c_i^1.$$

For example, the strategy $(0,0),(0,0),(0,1)$ means that Alice possesses the tuple $(c_1^0 = 0, c_1^1 = 0)$, Bob has $(c_2^0 = 0, c_2^1 = 0)$ and Charles has $(c_3^0 = 0, c_3^1 = 1)$, where on input $(1,1,0)$, they each get

$$a_1 = x_1 \cdot c_1^0 \oplus c_1^1 = 1 \cdot 0 \oplus 0 = 0,$$
$$a_2 = x_2 \cdot c_2^0 \oplus c_2^1 = 1 \cdot 0 \oplus 0 = 0,$$
$$a_3 = x_3 \cdot c_3^0 \oplus c_3^1 = 0 \cdot 0 \oplus 1 = 1.$$

This output satisfies the wining condition, which also means that the input $(1,1,0)$ is not in the exclusion set that this strategy belongs to.

We now describe the quantum winning strategy for this pseudo-telepathy game. Players

share the maximally entangled GHZ state $\frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|111\rangle$ prior to the start of the game, where the first qubit belongs to Alice, the second one to Bob and the third to Charles. After receiving their questions, each player performs the following quantum computations to their qubit to obtain the output:

1. Apply a unitary transformation U on their share of the entangled state with the input x_i, where U is defined as:

$$U|0\rangle \to |1\rangle$$
$$U|1\rangle \to e^{\imath\pi x_i/2}|1\rangle .$$

   The first symbol in the exponent is $\imath = \sqrt{-1}$.

2. Apply the Hadamard gate H on the resulting state.

3. Perform a measurement in the computational basis on their own qubit to obtain a_i, and output it.

Next, we will apply the protocol in section 3.3.2 to the Mermin-GHZ game to transform it into a bit commitment scheme.

4.2.2 Mermin-GHZ game bit commitment scheme

We simply follow the steps listed in section 3.3.2 to obtain the Mermin-GHZ game bit commitment scheme. We first apply the anchoring transformation to the Mermin-GHZ game. With n, the number of players, being 3, we obtain our anchoring parameter α by applying eq. (2.19)

$$\alpha = 1 - \sqrt[3]{\frac{3}{4}} \simeq 0.091. \tag{4.1}$$

This means that each input bit has roughly a 9% chance of being anchored. For any input $x \in \mathcal{X}$ and output $a \in \mathcal{A}$, the winning condition of the game is now:

$$W^{\perp}(x,a) = \begin{cases} 1, \text{ if } \exists d \in \{1,2,3\} \text{ s.t } x_d = \perp, \\ a_1 \oplus a_2 \oplus a_3 = x_1 \vee x_2 \vee x_3, \text{ otherwise} \end{cases} .$$

Next, we find and enumerate all optimal deterministic strategies described previously for the game, and their corresponding exclusion sets. This can be done with a *Python* program that exhaustively enumerates all 64 possible combinations of (c_i^0, c_i^1) for $i \in [3]$, and then filters them by evaluating the resulting outputs with the winning predicate. The result is presented in table 4.1 on the following page.

Due to the symmetry of the game, the index t to designate which player to execute the bit commitment scheme **sBGKW** with the extra player p_4 can be chosen arbitrarily. We observed that the bipartition of the exclusion sets can be done in any way to satisfy the hiding criteria of definition 4.1.1 as long as we have two exclusion sets in each side. Let us fix $t = 1$, and let the bipartition of the exclusion sets be as follows.

$$E_{\vec{\sigma}}^0 = \{E_1, E_2\}$$
$$E_{\vec{\sigma}}^1 = \{E_3, E_4\}$$

We show that this bipartition of the exclusion sets satisfies the hiding criteria of definition

exclusion set (x_1, x_2, x_3)	optimal deterministic strategies $(c_1^0, c_1^1), (c_2^0, c_2^1), (c_3^0, c_3^1)$	
$E_1 = \{(0,0,0)\}$	$(0,0), (0,0), (0,1)$	$(0,0), (0,1), (0,0)$
	$(0,1), (0,0), (0,0)$	$(0,1), (0,1), (0,1)$
	$(1,0), (1,0), (1,1)$	$(1,0), (1,1), (1,0)$
	$(1,1), (1,0), (1,0)$	$(1,1), (1,1), (1,1)$
$E_2 = \{(0,1,1)\}$	$(0,0), (1,0), (1,0)$	$(0,0), (1,1), (1,1)$
	$(0,1), (1,0), (1,1)$	$(0,1), (1,1), (1,0)$
	$(1,0), (0,0), (0,0)$	$(1,0), (0,1), (0,1)$
	$(1,1), (0,0), (0,1)$	$(1,1), (0,1), (0,0)$
$E_3 = \{(1,0,1)\}$	$(0,0), (1,0), (0,0)$	$(0,0), (1,1), (0,1)$
	$(0,1), (1,0), (0,1)$	$(0,1), (1,1), (0,0)$
	$(1,0), (0,0), (1,0)$	$(1,0), (0,1), (1,1)$
	$(1,1), (0,0), (1,1)$	$(1,1), (0,1), (1,0)$
$E_4 = \{(1,1,0)\}$	$(0,0), (0,0), (1,0)$	$(0,0), (0,1), (1,1)$
	$(0,1), (0,0), (1,1)$	$(0,1), (0,1), (1,0)$
	$(1,0), (1,0), (0,0)$	$(1,0), (1,1), (0,1)$
	$(1,1), (1,0), (0,1)$	$(1,1), (1,1), (0,0)$

Table 4.1 – Table listing all the exclusion sets and the optimal deterministic strategies that correspond to it for the Mermin-GHZ game.

4.1.1 for $t = 1$ in appendix A.1 with tables (A.2, A.3, A.4, A.5), and hence the resulting bit commitment scheme with this setup is statistically hiding from theorem 4.1.3. The provers and the verifier agree on the parameter k relating to the number of repetitions. Let the players be Alice, Bob, Charles, and let the extra player be Dave. With $i, j \in [k], d \in \{1, 2, 3\}$ and the verifier Victor, the protocol is as below.

Before commitment:

1. Players decide on $\mathbf{b}$, the bit value to be committed, and the binary matrix $\mathbf{B}$,

$$\mathbf{B} = \begin{pmatrix} b_{11} & \dots & \dots & \dots & b_{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ b_{k1} & \dots & \dots & \dots & b_{kk} \end{pmatrix},$$

   such that each row satisfies the eq. (3.4). For each b_{ij}, players uniformly sample an exclusion set $E_{\sigma}^{b_{ij}}$ from the set $E_{\vec{\sigma}}^{b_{ij}}$.

2. For each $E_{\sigma}^{b_{ij}}$, players agree on an optimal deterministic strategy σ^{ij} that derives the exclusion set. On the $(i,j)^{th}$ execution of the commitment protocol, player d will use local strategy $\sigma_d^{ij} = (c_d^0, c_d^1)_{ij}$.

3. Alice and Dave share a uniformly random bit string matrix $\mathbf{u}$ such that each u^{ij} is sampled from $\{0,1\}^k$,

$$\mathbf{u} = \begin{pmatrix} u^{11} & \dots & \dots & \dots & u^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & u^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ u^{k1} & \dots & \dots & \dots & u^{kk} \end{pmatrix},$$

Commit phase:

1. Victor randomly samples a question $\mathbf{x}^{\perp}$ according to $\left(\pi^{\perp}\right)^{k^2}$, where

$$\mathbf{x}^{\perp} = \begin{pmatrix} x^{11} & \dots & \dots & \dots & x^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x^{k1} & \dots & \dots & \dots & x^{kk} \end{pmatrix},$$

and each $x^{ij} = \left(x_1^{ij}, x_2^{ij}, x_3^{ij}\right) \in \mathcal{X}^{\perp}$, and sends $x_d^{\perp}$ to player p_d, where

$$x_d^{\perp} = \begin{pmatrix} x_d^{11} & \dots & \dots & \dots & x_d^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x_d^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_d^{k1} & \dots & \dots & \dots & x_d^{kk} \end{pmatrix}.$$

Victor samples a random bit string matrix R such that $\dim(R) = \dim(x_1^{\perp})$, and each $r^{ij} \in \{0,1\}^k$ and sends it to Alice.

2. Players answer with

$$\hat{\mathbf{a}} = \begin{pmatrix} \hat{a}^{11} & \dots & \dots & \dots & \hat{a}^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \hat{a}^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \hat{a}^{k1} & \dots & \dots & \dots & \hat{a}^{kk} \end{pmatrix},$$

$$\begin{aligned} \hat{a}^{ij} &= \left(\left(\sigma_1^{ij}\left(x_1^{ij}\right) \cdot r^{ij}\right) \oplus u^{ij},\ \sigma_2^{ij}\left(x_2^{ij}\right),\ \sigma_3^{ij}\left(x_3^{ij}\right)\right) \\ &= \left(\left(a_1^{ij} \cdot r^{ij}\right) \oplus u^{ij},\ a_2^{ij},\ a_3^{ij}\right). \end{aligned}$$

Unveil phase:

1. Dave sends both $\mathbf{B}$ and $\mathbf{u}$ to Victor, along with all k^2 different exclusion sets $E_\sigma^{b_{ij}}$.
2. Victor recovers the original answers from Alice by computing the following. For each $\hat{a}_1^{ij}$, he computes
$$y^{ij} = \hat{a}_1^{ij} \oplus u^{ij} = a_1^{ij} \cdot r^{ij}.$$
Then,
$$a_1^{ij} = \begin{cases} 0, \text{ if } y^{ij} = 0^k \\ 1, \text{ if } y^{ij} = r^{ij} \end{cases}.$$
Victor rejects the commitment if for any $i, j \in [k]$, the value of y^{ij} is not captured from the above equation.
3. For any $i, j \in [k]$, if $x^{ij} \in E_\sigma^{b_{ij}}$, Victor will accept the $(i,j)^{th}$ instance of the game such that $W^\perp\left(x^{ij}, a^{ij}\right) = 1$ regardless of what the actual value of a^{ij} is.
4. Victor accepts the commitment of $\mathbf{b}$ if both of the following equations hold true
$$\left(W^\perp\right)^{k^2}\left(\mathbf{x}^\perp, \mathbf{a}\right) = \prod_{i=1}^{k}\prod_{j=1}^{k} W^\perp\left(x^{ij}, a^{ij}\right) = 1,$$
$$\forall i \in [k], \bigoplus_{j=1}^{k} b_{ij} = \mathbf{b}.$$
Otherwise, he rejects the commitment.

4.2.3 Magic Square game bit commitment scheme

Similarly to the Mermin-GHZ bit commitment scheme shown previously, we first apply the anchoring transformation to the game. The anchoring parameter α is computed below with $n = 2$,

$$\alpha = 1 - \sqrt[2]{\frac{3}{4}} \simeq 0.134 \tag{4.2}$$

This means that each input bit has a probability of $\sim 13\%$ of being anchored. For any input $x \in \mathcal{X}$ and output $a \in \mathcal{A}$, the winning condition of the game is now:

$$W^{\perp}(x,a) = \begin{cases} 1, \text{ if } x_1 = \perp \vee x_2 = \perp, \\ W(x,a), \text{ otherwise,} \end{cases}$$

where $W(x,a)$ evaluates to 1 if and only if the parity of a_1 from Alice is even, the parity of a_2 from Bob is odd, and the intersecting entries of a_1 and a_2 agree.

As described in section 3.1.1, an optimal deterministic strategy for the Magic Square game is for the local provers to share a 3×3 binary matrix σ with a single entry labeled as '?' that satisfies the parity conditions in each row and column except for the row and column corresponding to '?'. Upon receiving the question x_1, a row number, Alice answers with $(\sigma(x_1,0), \sigma(x_1,1), \sigma(x_1,2))$, whereas Bob answers $(\sigma(0,x_2), \sigma(1,x_2), \sigma(2,x_2))$ when he receives the column number x_2. This is depicted in fig. 3.1 on page 42. Using this strategy, players can satisfy the winning condition for every row and column numbers except for the tuple (x_1, x_2) that correspond to '?', which results in $\omega_c(G) = \frac{8}{9}$. This means the exclusion sets of any optimal deterministic strategy of this game have exactly 1 element. With the aid of a program written in *Python*, we found all the squares that allow players to win with probability $\frac{8}{9}$. For each exclusion set, or simply each input, there exists 16 different squares that can win on all the other questions. This means that we have 9 exclusion sets and in total 144 optimal deterministic strategies for this game. We compile and show the exclusion sets along with all the corresponding optimal deterministic strategies in appendix A.2 with

tables (A.7, A.8, A.9, A.10, A.11, A.12, A.13, A.14, A.15).

We fix $t = 1$, and the bipartition of the exclusion sets is done in the following way.

$$E^0_{\vec{\sigma}} = \{(0,0),(0,1),(0,2),(1,0),(1,1)\}$$
$$E^1_{\vec{\sigma}} = \{(1,2),(2,0),(2,1),(2,2)\}$$

We show that this bipartition of the exclusion sets along with the index $t = 1$ satisfies the hiding criteria of definition 4.1.1 in appendix A.2, and hence the resulting bit commitment scheme is statistically hiding from theorem 4.1.3. The provers and the verifier agree on the parameter k relating to the number of repetitions. The Magic Square bit commitment protocol with players Alice, Bob, the extra player Charles, the verifier Victor, and along with the usual parameters $i, j \in [k], d \in \{1, 2\}$ is as below.

Before commitment:

1. Alice and Bob decide on $\mathbf{b}$, and the binary matrix $\mathbf{B}$,

$$\mathbf{B} = \begin{pmatrix} b_{11} & \dots & \dots & \dots & b_{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ b_{k1} & \dots & \dots & \dots & b_{kk} \end{pmatrix},$$

   such that each row satisfies the eq. (3.4). For each b_{ij}, they uniformly sample an exclusion set $E_{\sigma}^{b_{ij}}$ from the set $E_{\vec{\sigma}}^{b_{ij}}$.

2. For each $E_{\sigma}^{b_{ij}}$, Alice and Bob agree on an optimal deterministic strategy σ^{ij} that derives the exclusion set. On the $(i,j)^{th}$ execution of the commitment protocol, both Alice and Bob will use the shared optimal square (σ^{ij}) to answer the queries.

3. Alice and Charles share a uniformly random bit string matrix $\mathbf{u}$ such that each $u^{ij} \in \{0,1\}^{3k}$,

$$\mathbf{u} = \begin{pmatrix} u^{11} & \dots & \dots & \dots & u^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & u^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ u^{k1} & \dots & \dots & \dots & u^{kk} \end{pmatrix},$$

Commit phase:

1. Victor randomly samples the questions $x_1^{\perp}$ with $\left(\pi_1^{\perp}\right)^{k^2}$ and $x_2^{\perp}$ with $\left(\pi_2^{\perp}\right)^{k^2}$,

$$x_1^{\perp} = \begin{pmatrix} x_1^{11} & \dots & \dots & \dots & x_1^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x_1^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_1^{k1} & \dots & \dots & \dots & x_1^{kk} \end{pmatrix}, x_2^{\perp} = \begin{pmatrix} x_2^{11} & \dots & \dots & \dots & x_2^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & x_2^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x_2^{k1} & \dots & \dots & \dots & x_2^{kk} \end{pmatrix},$$

   and sends them to Alice and Bob, respectively. Victor samples a random bit string matrix R such that $\dim(R) = \dim(x_2^{\perp})$, and each $r^{ij} \in \{0,1\}^{3k}$ and sends it to Bob.

2. Players answer with

$$\hat{\mathbf{a}} = \begin{pmatrix} \hat{a}^{11} & \dots & \dots & \dots & \hat{a}^{1k} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \hat{a}^{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \hat{a}^{k1} & \dots & \dots & \dots & \hat{a}^{kk} \end{pmatrix},$$

$$\begin{aligned} \hat{a}_1^{ij} &= \left(\left(\sigma_1^{ij}\left(x_1^{ij}, 0\right) \cdot r_1^{ij}\right) \oplus u_1^{ij}, \left(\sigma_1^{ij}, \left(x_1^{ij}, 1\right) \cdot r_2^{ij}\right) \oplus u_2^{ij}, \left(\sigma_1^{ij}\left(x_1^{ij}, 2\right) \cdot r_3^{ij}\right) \oplus u_3^{ij}\right) \\ &= \left(\left(a_{11}^{ij} \cdot r_1^{ij}\right) \oplus u_1^{ij},\ \left(a_{12}^{ij} \cdot r_2^{ij}\right) \oplus u_2^{ij},\ \left(a_{13}^{ij} \cdot r_3^{ij}\right) \oplus u_3^{ij}\right) \\ a_2^{ij} &= \left(\sigma_2^{ij}\left(0, x_2^{ij}\right),\ \sigma_2^{ij}\left(1, x_2^{ij}\right),\ \sigma_2^{ij}\left(2, x_2^{ij}\right)\right) \\ \hat{a}^{ij} &= \left(\hat{a}_1^{ij}, a_2^{ij}\right) \end{aligned},$$

   where $a_{11}^{ij}, a_{12}^{ij}, a_{13}^{ij}$ are the 1^{st}, the 2^{nd} and the 3^{rd} bit of the output a_1^{ij} from Alice.

Unveil phase:

1. Charles sends both $\mathbf{B}$ and $\mathbf{u}$ to Victor, along with all k^2 different exclusion sets $E_\sigma^{b_{ij}}$.

2. Victor recovers the original answers from Alice by computing the following. For each $\hat{a}_1^{ij}$, he computes

$$\begin{aligned} y^{ij} &= \hat{a}_1^{ij} \oplus u^{ij} \\ &= \left(a_{11}^{ij} \cdot r_1^{ij},\ a_{12}^{ij} \cdot r_2^{ij},\ a_{13}^{ij} \cdot r_3^{ij}\right) \\ &= \left(y_1^{ij},\ y_2^{ij},\ y_3^{ij}\right) \end{aligned}$$

   Then, for $l \in \{1, 2, 3\}$,

$$a_{1l}^{ij} = \begin{cases} 0, \text{ if } y_l^{ij} = 0^k \\ 1, \text{ if } y_l^{ij} = r_l^{ij} \end{cases}.$$

   Victor rejects the commitment if for any $i, j \in [k]$, the value of y_l^{ij} is not captured from the above equation.

3. For any $i, j \in [k]$, if $x^{ij} \in E_\sigma^{b_{ij}}$, Victor will accept the $(i,j)^{th}$ instance of the game such that $W^\perp\left(x^{ij}, a^{ij}\right) = 1$ regardless of what the actual value of a^{ij} is.

4. Victor accepts the commitment of $\mathbf{b}$ if both of the following equations hold true

$$\left(W^\perp\right)^{k^2}\left(\mathbf{x}^\perp, \mathbf{a}\right) = \prod_{i=1}^{k}\prod_{j=1}^{k} W^\perp\left(x^{ij}, a^{ij}\right) = 1,$$

$$\forall i \in [k],\ \bigoplus_{j=1}^{k} b_{ij} = \mathbf{b}.$$

   Otherwise, he rejects the commitment.

Chapter 5

Conclusion

Nonlocal games remain an interesting mathematical model that provides an approachable gateway to more complex concepts in various fields of studies such as quantum nonlocality and computational complexity theories. We formally and thoroughly introduced the framework of the nonlocal game in section 2.5 along with all of its variants. In addition, we showed the anchoring transformation [BVY15] of a multiplayer nonlocal game which results in the extension of the famed Raz's parallel repetition theorem [Raz98] in the n players case despite the simplicity of the transformation. In this work, we introduced yet another application of the nonlocal game in cryptography by showing a protocol in section 3.3.2 that constructs a classically (local hidden variable model) secure bit commitment scheme from a pseudo-telepathy game. We employed the anchoring transformation, the **sBGKW** bit commitment scheme, as well as the structure and properties of the exclusion set of an optimal deterministic strategy as building blocks to achieve this. Our result is different from existing nonlocal game bit commitment schemes in that we do not need a physical implementation of a nonlocal box containing the nonlocal correlation of the respective game in order for quantum parties to obtain correlated outputs.

We analyzed and proved the security of the resulting bit commitment schemes following our protocol in Chapter 4. Provers can achieve hiding by carefully bipartitioning the

exclusion sets such that the hiding condition presented in section 4.1.2 is satisfied. We introduced a new binding definition of a bit commitment scheme called the non-binding game in section 2.7.1. A commitment protocol is binding against dishonest provers when they cannot win the non-binding game with probability much better than randomly guessing which value will be chosen by the verifier to unveil. Subsequently, we proved that classical players can only win the non-binding game in the context of the bit commitment scheme constructed following our protocol with probability at most $1/2 + \epsilon(k)$, whereas quantum provers that share entangled states used in the quantum winning strategy of the underlying pseudo-telepathy game can win with certainty. Together with theorem 4.1.1 and theorem 4.1.2, the resulting bit commitment scheme is binding against classical provers but is fully non-binding against quantum provers such that the verifier will accept any values that they unveil. To demonstrate that the protocol can be followed easily, we showed two concrete applications of the recipe with the Mermin-GHZ game and the Magic Square game in section 4.2.

We provided possible variations of our protocol with different efficiency and security requirements in the end of chapter 3. An immediate extension of this work is to formalize these variations of the protocol and analyze their security. Another interesting question is how we can use the special property that the quantum provers have an advantage over their classical counter parts in the bit commitment protocols resultant from this work to build Quantum Simulatable Zero-Knowledge proofs. Lastly, a reviewer suggested to extend our binding game definition to address the case for bit commitment schemes that have multiple provers involved in the opening of the commitment.

Appendix A

Hiding

A bit commitment scheme is statistically hiding when the verifier can only learn a negligible amount of information about the bit b prior to the opening of the commitment. Recall from our hiding definition in section 4.1.2 on page 74, in the settings of bit commitment schemes built using our protocol, it is statistically hiding when the following is true for the underlying pseudo-telepathy game G:

$$\begin{aligned}
&\forall x \in \mathcal{X}, \exists a, a' \in \mathcal{A} \text{ s.t } W(x,a) = W(x,a') = 1, \\
&\left|\{E_\sigma \subseteq E^b_{\vec{\sigma}} \mid \sigma(x) = a\}\right| > 0, \\
&\left|\{E_{\sigma'} \subseteq E^{\bar{b}}_{\vec{\sigma}} \mid \sigma'(x) = a'\}\right| > 0,
\end{aligned}$$

where a may differ with a' at position t. Note that we drop the ij indices to simplify the notation here. In simpler words, for any given query from the verifier, if there exists two valid answers that differ only at position t, then the number of exclusion sets with strategies that can produce one of the outputs from both sides of the bipartition has to be non-zero. This way, the verifier cannot be certain which side of the bipartition is responsible of the output. The bias he has for each output vanishes as we repeat the protocol k^2 times.

We show in the subsequent sections that the bit commitment schemes we have con-

structed as examples in section 4.2 have this property. More precisely, we show exhaustively that for each question, and for each corresponding winning output, there is always at least 1 exclusion set with an optimal deterministic strategy that can produce it from each side of the bipartition.

A.1 Mermin-GHZ bit commitment scheme is statistically hiding

As stated in section 4.2.2, the exclusion sets for the Mermin-GHZ game are partitioned as shown in table A.1, where the corresponding optimal deterministic strategies are listed in table 4.1 on page 83. We fixed $t = 1$, which means that the verifier does not know Alice's output during the commit phase.

$E^0_{\vec{\sigma}}$	$E^1_{\vec{\sigma}}$
$E_1 = \{(0,0,0)\}$	$E_3 = \{(1,0,1)\}$
$E_2 = \{(0,1,1)\}$	$E_4 = \{(1,1,0)\}$

Table A.1 – Bipartition of the exclusion sets of the Mermin-GHZ bit commitment scheme.

We do not have to worry about an output a' that is different from a for this game, since for any value of t, if we flip the bit at position t, the parity of the output will change as well. This leads to an unsatisfying answer since the winning condition of this game relies on the parity of the answers. This means that the choice of t can be made arbitrary for this bit commitment protocol. To confirm that our bipartition satisfies the hiding condition stated previously, it suffices to show the following: For each pair of (x, a), where $W(x, a) = 1$, we need to show an optimal deterministic strategy σ such that $E_\sigma \in E^b_{\vec{\sigma}}$, and $\sigma(x) = a$, and another optimal deterministic strategy σ' such that $E_{\sigma'} \in E^{\bar{b}}_{\vec{\sigma}}$, and $\sigma'(x) = a$.

To do so, for each question, we present a table with 3 columns that correspond to

a winning output, a strategy σ such that $E_\sigma \in E^0_{\vec{\sigma}}$, and another strategy σ' such that $E_{\sigma'} \in E^1_{\vec{\sigma}}$. We show this in the following tables (A.2, A.3, A.4, A.5)

a	$\sigma, E_\sigma \in E^0_{\vec{\sigma}}$	$\sigma', E_{\sigma'} \in E^1_{\vec{\sigma}}$
$(0,0,0)$	$((0,0),(1,0),(1,0)), E_2$	$((0,0),(1,0),(0,0)), E_3$
$(0,1,1)$	$((0,0),(1,1),(1,1)), E_2$	$((0,0),(0,1),(1,1)), E_4$
$(1,1,0)$	$((1,1),(0,1),(0,0)), E_2$	$((1,1),(0,1),(1,0)), E_3$
$(1,0,1)$	$((0,1),(1,0),(1,1)), E_2$	$((1,1),(1,0),(0,1)), E_4$

Table A.2 – Table showing strategies from both sides that can produce an answer a such that for $x = (0,0,0), W(x,a) = 1$.

We can verify one of the rows in table A.2 The rest of the verification is omitted. For the pair $(x,a) = ((0,0,0),(1,0,1)$, we have

$$a_1 = 0 \cdot 0 + 1 = 1 \quad a'_1 = 1 \cdot 0 + 1 = 1$$
$$a_2 = 1 \cdot 0 + 0 = 0 \quad a'_2 = 1 \cdot 0 + 0 = 0$$
$$a_2 = 1 \cdot 0 + 1 = 1 \quad a'_3 = 0 \cdot 0 + 1 = 1$$

This gives us $a = a' = (1,0,1)$, and $W(x,a) = 1$ since $x_1 \vee x_2 \vee x_3 = 0 \vee 0 \vee 0 \vee 0 = 0$ while $a_1 \oplus a_2 \oplus a_3 = 1 \oplus 0 \oplus 1 = 0$.

a	$\sigma, E_\sigma \in E^0_{\vec{\sigma}}$	$\sigma', E_{\sigma'} \in E^1_{\vec{\sigma}}$
$(0,0,1)$	$((0,0),(0,0),(0,1)), E_1$	$((0,0),(1,1),(0,1)), E_3$
$(0,1,0)$	$((0,0),(0,1),(0,0)), E_1$	$((0,0),(0,1),(1,1)), E_4$
$(1,0,0)$	$((1,1),(1,1),(1,1)), E_1$	$((1,1),(0,0),(1,1)), E_3$
$(1,1,1)$	$((1,1),(1,0),(1,0)), E_1$	$((1,1),(0,1),(1,0)), E_4$

Table A.3 – Table showing strategies from both sides that can produce an answer a such that for $x = (0,1,1), W(x,a) = 1$.

a	$\sigma, E_\sigma \in E^0_{\vec{\sigma}}$	$\sigma', E_{\sigma'} \in E^1_{\vec{\sigma}}$
$(0,0,1)$	$((1,1),(1,0),(1,0)), E_1$	$((0,0),(0,0),(1,0)), E_4$
$(0,1,0)$	$((1,1),(1,1),(1,1)), E_1$	$((1,1),(1,1),(0,0)), E_4$
$(1,0,0)$	$((0,1),(1,0),(1,1)), E_2$	$((0,1),(0,0),(1,1)), E_4$
$(1,1,1)$	$((0,1),(1,1),(1,0)), E_2$	$((1,0),(1,1),(0,1)), E_4$

Table A.4 – Table showing strategies from both sides that can produce an answer a such that for $x = (1,0,1), W(x,a) = 1$.

a	$\sigma, E_\sigma \in E^0_{\vec{\sigma}}$	$\sigma', E_{\sigma'} \in E^1_{\vec{\sigma}}$
$(0,0,1)$	$((0,0),(1,1),(1,1)), E_2$	$((0,0),(1,1),(0,1)), E_3$
$(0,1,0)$	$((1,1),(1,0),(1,0)), E_1$	$((1,1),(0,1),(1,0)), E_3$
$(1,0,0)$	$((0,1),(1,1),(1,0)), E_2$	$((0,1),(1,1),(0,0)), E_3$
$(1,1,1)$	$((1,0),(1,0),(1,1)), E_1$	$((1,0),(0,1),(1,1)), E_3$

Table A.5 – Table showing strategies from both sides that can produce an answer a such that for $x = (1,1,0), W(x,a) = 1$.

A.2 Magic Square bit commitment scheme is statistically hiding

The bipartition of the exclusion sets for the bit commitment scheme is shown in table A.6. Subsequently, for each exclusion set, we list all the optimal deterministic strategies that derive it in the following tables (A.7, A.8, A.9, A.10, A.11, A.12, A.13, A.14, A.15).

$E^0_{\vec{\sigma}}$	$E^1_{\vec{\sigma}}$
$E_1 = \{(0,0)\}$	$E_6 = \{(1,2)\}$
$E_2 = \{(0,1)\}$	$E_7 = \{(2,0)\}$
$E_3 = \{(0,2)\}$	$E_8 = \{(2,1)\}$
$E_4 = \{(1,0)\}$	$E_9 = \{(2,2)\}$
$E_5 = \{(1,1)\}$	

Table A.6 – Bipartition of the exclusion sets of the Magic Square bit commitment scheme.

$E_1 = \{(0,0)\}$			
$\begin{bmatrix} ? & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} ? & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} ? & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} ? & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$
$\begin{bmatrix} ? & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} ? & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} ? & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} ? & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} ? & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} ? & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} ? & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} ? & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} ? & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} ? & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} ? & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} ? & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

Table A.7 – Optimal deterministic strategies that all fail at $x = (0,0)$.

$E_2 = \{(0,1)\}$			
$\begin{bmatrix}0 & ? & 0\\0 & 0 & 0\\1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & ? & 0\\0 & 1 & 1\\1 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & ? & 0\\1 & 0 & 1\\0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & ? & 0\\1 & 1 & 0\\0 & 1 & 1\end{bmatrix}$
$\begin{bmatrix}0 & ? & 1\\0 & 0 & 0\\1 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & ? & 1\\0 & 1 & 1\\1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & ? & 1\\1 & 0 & 1\\0 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}0 & ? & 1\\1 & 1 & 0\\0 & 0 & 0\end{bmatrix}$
$\begin{bmatrix}1 & ? & 0\\0 & 0 & 0\\0 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}1 & ? & 0\\0 & 1 & 1\\0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}1 & ? & 0\\1 & 0 & 1\\1 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}1 & ? & 0\\1 & 1 & 0\\1 & 0 & 1\end{bmatrix}$
$\begin{bmatrix}1 & ? & 1\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}1 & ? & 1\\0 & 1 & 1\\0 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}1 & ? & 1\\1 & 0 & 1\\1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}1 & ? & 1\\1 & 1 & 0\\1 & 1 & 0\end{bmatrix}$

Table A.8 – Optimal deterministic strategies that all fail at $x = (0, 1)$.

$E_3 = \{(0,2)\}$			
$\begin{bmatrix}0 & 0 & ?\\0 & 0 & 0\\1 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0 & ?\\0 & 1 & 1\\1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 0 & ?\\1 & 0 & 1\\0 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 0 & ?\\1 & 1 & 0\\0 & 0 & 0\end{bmatrix}$
$\begin{bmatrix}0 & 1 & ?\\0 & 0 & 0\\1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1 & ?\\0 & 1 & 1\\1 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 1 & ?\\1 & 0 & 1\\0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 1 & ?\\1 & 1 & 0\\0 & 1 & 1\end{bmatrix}$
$\begin{bmatrix}1 & 0 & ?\\0 & 0 & 0\\0 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 0 & ?\\0 & 1 & 1\\0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0 & ?\\1 & 0 & 1\\1 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0 & ?\\1 & 1 & 0\\1 & 0 & 1\end{bmatrix}$
$\begin{bmatrix}1 & 1 & ?\\0 & 0 & 0\\0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}1 & 1 & ?\\0 & 1 & 1\\0 & 1 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 1 & ?\\1 & 0 & 1\\1 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 1 & ?\\1 & 1 & 0\\1 & 1 & 0\end{bmatrix}$

Table A.9 – Optimal deterministic strategies that all fail at $x = (0, 2)$.

$E_4 = \{(1,0)\}$			
$\begin{bmatrix}0&0&0\\?&0&0\\0&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\?&0&1\\1&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\?&1&0\\1&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\?&1&1\\0&0&0\end{bmatrix}$
$\begin{bmatrix}0&1&1\\?&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\?&0&1\\1&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&1\\?&1&0\\1&1&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\?&1&1\\0&1&1\end{bmatrix}$
$\begin{bmatrix}1&0&1\\?&0&0\\1&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\?&0&1\\0&1&1\end{bmatrix}$	$\begin{bmatrix}1&0&1\\?&1&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\?&1&1\\1&0&1\end{bmatrix}$
$\begin{bmatrix}1&1&0\\?&0&0\\1&0&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\?&0&1\\0&0&0\end{bmatrix}$	$\begin{bmatrix}1&1&0\\?&1&0\\0&1&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\?&1&1\\1&1&0\end{bmatrix}$

Table A.10 – Optimal deterministic strategies that all fail at $x = (1, 0)$.

$E_5 = \{(1,1)\}$			
$\begin{bmatrix}0&0&0\\0&?&0\\1&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&?&1\\1&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&?&0\\0&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&?&1\\0&0&0\end{bmatrix}$
$\begin{bmatrix}0&1&1\\0&?&0\\1&1&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\0&?&1\\1&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&?&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&?&1\\0&1&1\end{bmatrix}$
$\begin{bmatrix}1&0&1\\0&?&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\0&?&1\\0&1&1\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&?&0\\1&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&?&1\\1&0&1\end{bmatrix}$
$\begin{bmatrix}1&1&0\\0&?&0\\0&1&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\0&?&1\\0&0&0\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&?&0\\1&0&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&?&1\\1&1&0\end{bmatrix}$

Table A.11 – Optimal deterministic strategies that all fail at $x = (1, 1)$.

$E_6 = \{(1,2)\}$			
$\begin{bmatrix}0&0&0\\0&0&?\\1&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&?\\1&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&0&?\\0&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&1&?\\0&0&0\end{bmatrix}$
$\begin{bmatrix}0&1&1\\0&0&?\\1&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&1\\0&1&?\\1&1&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&0&?\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&1&?\\0&1&1\end{bmatrix}$
$\begin{bmatrix}1&0&1\\0&0&?\\0&1&1\end{bmatrix}$	$\begin{bmatrix}1&0&1\\0&1&?\\0&0&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&0&?\\1&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&1&?\\1&0&1\end{bmatrix}$
$\begin{bmatrix}1&1&0\\0&0&?\\0&0&0\end{bmatrix}$	$\begin{bmatrix}1&1&0\\0&1&?\\0&1&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&0&?\\1&0&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&1&?\\1&1&0\end{bmatrix}$

Table A.12 – Optimal deterministic strategies that all fail at $x = (1, 2)$.

$E_7 = \{(2,0)\}$			
$\begin{bmatrix}0&0&0\\0&0&0\\?&1&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&1\\?&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&0&1\\?&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&1&0\\?&0&1\end{bmatrix}$
$\begin{bmatrix}0&1&1\\0&0&0\\?&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\0&1&1\\?&1&1\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&0&1\\?&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&1&0\\?&1&0\end{bmatrix}$
$\begin{bmatrix}1&0&1\\0&0&0\\?&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\0&1&1\\?&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&0&1\\?&1&1\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&1&0\\?&0&0\end{bmatrix}$
$\begin{bmatrix}1&1&0\\0&0&0\\?&0&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\0&1&1\\?&1&0\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&0&1\\?&0&0\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&1&0\\?&1&1\end{bmatrix}$

Table A.13 – Optimal deterministic strategies that all fail at $x = (2, 0)$.

$E_8 = \{(2,1)\}$			
$\begin{bmatrix}0&0&0\\0&0&0\\1&?&1\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&1\\1&?&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&0&1\\0&?&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&1&0\\0&?&1\end{bmatrix}$
$\begin{bmatrix}0&1&1\\0&0&0\\1&?&0\end{bmatrix}$	$\begin{bmatrix}0&1&1\\0&1&1\\1&?&1\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&0&1\\0&?&1\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&1&0\\0&?&0\end{bmatrix}$
$\begin{bmatrix}1&0&1\\0&0&0\\0&?&0\end{bmatrix}$	$\begin{bmatrix}1&0&1\\0&1&1\\0&?&1\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&0&1\\1&?&1\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&1&0\\1&?&0\end{bmatrix}$
$\begin{bmatrix}1&1&0\\0&0&0\\0&?&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\0&1&1\\0&?&0\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&0&1\\1&?&0\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&1&0\\1&?&1\end{bmatrix}$

Table A.14 – Optimal deterministic strategies that all fail at $x = (2,1)$.

$E_9 = \{(2,2)\}$			
$\begin{bmatrix}0&0&0\\0&0&0\\1&1&?\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&1\\1&0&?\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&0&1\\0&1&?\end{bmatrix}$	$\begin{bmatrix}0&0&0\\1&1&0\\0&0&?\end{bmatrix}$
$\begin{bmatrix}0&1&1\\0&0&0\\1&0&?\end{bmatrix}$	$\begin{bmatrix}0&1&1\\0&1&1\\1&1&?\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&0&1\\0&0&?\end{bmatrix}$	$\begin{bmatrix}0&1&1\\1&1&0\\0&1&?\end{bmatrix}$
$\begin{bmatrix}1&0&1\\0&0&0\\0&1&?\end{bmatrix}$	$\begin{bmatrix}1&0&1\\0&1&1\\0&0&?\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&0&1\\1&1&?\end{bmatrix}$	$\begin{bmatrix}1&0&1\\1&1&0\\1&0&?\end{bmatrix}$
$\begin{bmatrix}1&1&0\\0&0&0\\0&0&?\end{bmatrix}$	$\begin{bmatrix}1&1&0\\0&1&1\\0&1&?\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&0&1\\1&0&?\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&1&0\\1&1&?\end{bmatrix}$

Table A.15 – Optimal deterministic strategies that all fail at $x = (2,2)$.

We will show how this bipartition of the exclusion sets along with $t = 1$ satisfies the hiding condition for $x = (0,0)$ in table A.16 on the following page. The remaining 8 tables

are left as an exercise for the readers.

$\sigma, E_\sigma \in E^0_{\vec{\sigma}}$	a	$\sigma', E_{\sigma'} \in E^1_{\vec{\sigma}}$	a'
$\begin{bmatrix}0 & ? & 0\\0 & 0 & 0\\1 & 0 & 1\end{bmatrix}, E_2$	$(0,0,0),(0,0,1)$	$\begin{bmatrix}0 & 1 & 1\\0 & 1 & ?\\1 & 1 & 0\end{bmatrix}, E_6$	$(0,1,1),(0,0,1)$
$\begin{bmatrix}0 & 1 & ?\\1 & 0 & 1\\0 & 0 & 0\end{bmatrix}, E_3$	$(0,1,1),(0,1,0)$	$\begin{bmatrix}0 & 1 & 1\\1 & 1 & 0\\? & 1 & 0\end{bmatrix}, E_7$	$(0,1,1),(0,1,0)$
$\begin{bmatrix}1 & 0 & 1\\? & 0 & 1\\0 & 1 & 1\end{bmatrix}, E_4$	$(1,0,1),(1,0,0)$	$\begin{bmatrix}1 & 0 & 1\\0 & 1 & 1\\0 & 0 & ?\end{bmatrix}, E_9$	$(1,0,1),(1,0,0)$
$\begin{bmatrix}1 & 0 & 1\\1 & ? & 1\\1 & 0 & 1\end{bmatrix}, E_5$	$(1,0,1),(1,1,1)$	$\begin{bmatrix}1 & 1 & 0\\1 & 1 & 0\\1 & ? & 1\end{bmatrix}, E_8$	$(1,1,0),(1,1,1)$

Table A.16 – Table showing strategies from both sides that can produce two answers a, a' such that for $x = (0,0), W(x,a) = W(x,a') = 1$.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ABB+10] Mafalda L Almeida, Jean-Daniel Bancal, Nicolas Brunner, Antonio Acín, Nicolas Gisin, and Stefano Pironio. Guess your neighbor's input: A multipartite nonlocal game with no quantum advantage. *Physical review letters*, 104(23):230404, 2010.

[AMPS16] Nati Aharon, Serge Massar, Stefano Pironio, and Jonathan Silman. Device-independent bit commitment based on the CHSH inequality. *New Journal of Physics*, 18(2):025014, 2016.

[Ara02] Padmanabhan K Aravind. Bell's theorem without inequalities and only two distant observers. *Journal of Genetic Counseling*, 15(4):397–405, 2002.

[Bab85] L Babai. Trading Group Theory for Randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 421–429, New York, NY, USA, 1985. Association for Computing Machinery.

[BBC+93] Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical review letters*, 70(13):1895, 1993.

[BBT05] Gilles Brassard, Anne Broadbent, and Alain Tapp. Quantum pseudo-telepathy. *Foundations of Physics*, 35(11):1877–1907, 2005.

[BC86] Gilles Brassard and Claude Crépeau. Non-transitive transfer of confidence: A perfect zero-knowledge interactive protocol for SAT and beyond. In *27th Annual Symposium on Foundations of Computer Science (FOCS 1986)*, pages 188–195. IEEE, 1986.

[BC90] Gilles Brassard and Claude Crépeau. Quantum bit commitment and coin tossing protocols. In *Conference on the Theory and Application of Cryptography*, pages 49–61. Springer, 1990.

[BC96] Gilles Brassard and Claude Crépeau. 25 Years of Quantum Cryptography. *SIGACT News*, 27(3):13–24, September 1996.

[BCJL93] G. Brassard, C. Crépeau, R. Jozsa, and D. Langlois. A quantum bit commitment scheme provably unbreakable by both parties. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 362–371, Nov 1993.

[BCU+06] Harry Buhrman, Matthias Christandl, Falk Unger, Stephanie Wehner, and Andreas Winter. Implications of superstrong non-locality for cryptography. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 462(2071):1919–1932, Feb 2006.

[Bel64] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika*, 1:195–200, Nov 1964.

[BFL92] L. Babai, L. Fortnow, and C. Lund. Non-Deterministic Exponential Time Has Two-Prover Interactive Protocols. *Comput. Complex.*, 2(4):374, December 1992.

[BFS13] Harry Buhrman, Serge Fehr, and Christian Schaffner. On the parallel repetition of multi-player games: The no-signaling case. *arXiv preprint*

arXiv:1312.7455, 2013.

[Blu83] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *ACM SIGACT News*, 15(1):23–27, 1983.

[BOGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-Prover Interactive Proofs: How to Remove Intractability Assumptions. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 113–131, New York, NY, USA, 1988. Association for Computing Machinery.

[BVY15] Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Anchoring games for parallel repetition. *arXiv preprint arXiv:1509.07466*, 2015.

[BW92] Charles H Bennett and Stephen J Wiesner. Communication via one-and two-particle operators on Einstein-Podolsky-Rosen states. *Physical review letters*, 69(20):2881, 1992.

[CHSH69] John F Clauser, Michael A Horne, Abner Shimony, and Richard A Holt. Proposed experiment to test local hidden-variable theories. *Physical review letters*, 23(15):880, 1969.

[CHTW04] Richard Cleve, Peter Hoyer, Benjamin Toner, and John Watrous. Consequences and Limits of Nonlocal Strategies. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, CCC '04, page 236–249, USA, 2004. IEEE Computer Society.

[CRC19] Xavier Coiteux-Roy and Claude Crépeau. The RGB No-Signalling Game. *arXiv preprint arXiv:1901.05062*, 2019.

[CSST11] Claude Crépeau, Louis Salvail, Jean-Raymond Simard, and Alain Tapp. Two Provers in Isolation. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 407–430, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[CY19] Claude Crépeau and Nan Yang. Non-Locality and Zero-Knowledge MIPs. *arXiv preprint arXiv:1907.12619*, 2019.

[DPP93] Ivan B Damgård, Torben P Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. In *Annual International Cryptology Conference*, pages 250–265. Springer, 1993.

[DSV15] Irit Dinur, David Steurer, and Thomas Vidick. A parallel repetition theorem for entangled projection games. *computational complexity*, 24(2):201–254, 2015.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[GMR85] S Goldwasser, S Micali, and C Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 291–304, New York, NY, USA, 1985. Association for Computing Machinery.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

[HM96] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Annual International Cryptology Conference*, pages 201–215. Springer, 1996.

[JNV+20] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*= RE. *arXiv preprint arXiv:2001.04383*, 2020.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition, 2014.

[LC97] Hoi-Kwong Lo and H. F. Chau. Is Quantum Bit Commitment Really Possible? *Phys. Rev. Lett.*, 78:3410–3413, Apr 1997.

[LC98] Hoi-Kwong Lo and H.F. Chau. Why quantum bit commitment and ideal quantum coin tossing are impossible. *Physica D: Nonlinear Phenomena*, 120(1-2):177–187, Sep 1998.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic Methods for Interactive Proof Systems. *J. ACM*, 39(4):859–868, October 1992.

[May97] Dominic Mayers. Unconditionally Secure Quantum Bit Commitment is Impossible. *Phys. Rev. Lett.*, 78:3414–3417, Apr 1997.

[Mer90] N David Mermin. Simple unified form for the major no-hidden-variables theorems. *Physical review letters*, 65(27):3373, 1990.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[PR94] Sandu Popescu and Daniel Rohrlich. Quantum nonlocality as an axiom. *Foundations of Physics*, 24(3):379–385, 1994.

[PR98] Sandu Popescu and Daniel Rohrlich. Causality and nonlocality as axioms for quantum mechanics. In *Causality and locality in modern physics*, pages 383–389. Springer, 1998.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[RV15] Oded Regev and Thomas Vidick. Quantum XOR games. *ACM Transactions on Computation Theory (ToCT)*, 7(4):1–43, 2015.

[SCA+11] J. Silman, A. Chailloux, N. Aharon, I. Kerenidis, S. Pironio, and S. Massar. Fully Distrustful Quantum Bit Commitment and Coin Flipping. *Physical Review Letters*, 106(22), Jun 2011.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, October 1992.

[Sip96] Michael Sipser. Introduction to the Theory of Computation. *ACM Sigact News*, 27(1):27–29, 1996.

[VD13] Wim Van Dam. Implausible consequences of superstrong nonlocality. *Natural Computing*, 12(1):9–12, 2013.

[Wil13] Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, USA, 1st edition, 2013.

[WWW11] S. Winkler, J. Wullschleger, and S. Wolf. Bit Commitment From Nonsignaling Correlations. *IEEE Transactions on Information Theory*, 57(3):1770–1779, 2011.