Post-Quantum Zero-Knowledge Proofs using Homomorphic Bit Commitment

John Stuart, School of Computer Science

McGill University, Montreal

December, 2022

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

Master of Computer Science

Abstract

With quantum computers rapidly becoming more powerful, proving that cryptographic protocols are secure against quantum adversaries is more important than ever. Moreover, it is the goal of every cryptographer to loosen assumptions, in particular, restrictions on the adversary's ability to perform long computations. In this thesis, a multi-party structure is used to create two polynomial time Zero-Knowledge Proofs (ZKPs) that are secure with no computational restrictions on any party, even for adversaries that share quantum entanglement. A ZKP allows one party to convince another party of a fact without disclosing any extra knowledge except the validity of the fact. For example, it could be used to allow a customer to prove their identity to a bank machine without giving away private information such as a personal identification number. An important tool in many ZKPs is bit commitment, which is essentially a digital way for a sender to put a message in a lockbox, lock it, and send it to the receiver. Then, when the sender would like the receiver to read the message, the key is sent for the receiver to open the box. This way, the message is hidden from the receiver until they receive the key, and the sender is unable to change their mind after sending the lockbox. In the thesis, the homomorphic properties of a certain multi-party bit commitment scheme will be exploited to allow the receiver to perform operations on commitments, resulting in ZKPs for two NP-Complete problems: the Subset Sum problem and 3SAT.

Abrégé

Avec la montée en puissance des ordinateurs quantiques, il est plus important que jamais de démontrer que les protocoles cryptographiques sont sécuritaires contre des adversaires quantiques. De plus, l'objectif de tout cryptographe est d'assouplir certaines hypothèses, en particulier celles qui restreignent la capacité de l'adversaire à effectuer de longs calculs. Dans cette thèse, un système multi-parties est utilisé pour créer deux preuves Zéro-Knowledge (ZK), celles-ci opèrent en temps polynomial et sont sécurisées. De plus, elles n'ont aucune restriction calculatoire pour aucune partie, même pour les adversaires qui partagent une intrication quantique. Une preuve ZK permet à une partie de prouver à une autre partie d'un fait sans révéler aucune connaissance autre que la véracité du fait. Par exemple, une telle preuve pourrait être utilisée pour permettre à un client de prouver son identité à un guichet bancaire sans donner des renseignements confidentiels tel que son numéro d'identification personnel. Un des principaux outils utilisés lors d'une preuve ZK est le schéma de mise en gage. Celui-ci est essentiellement un outil numérique permettant à un expéditeur de sceller un message dans un coffre-fort et de l'envoyer à un destinataire. Ensuite, quand l'expéditeur veut que le destinataire lise le message, la clé du coffre fort lui est envoyée. Ceci permet au message de rester inconnu du receveur jusqu'à ce qu'il reçoive la clé de l'expéditeur. De plus, l'expéditeur ne peut pas changer le message une fois le coffre-fort envoyé. Au cours de cette thèse, les propriétés homomorphiques d'un schéma multi-parties de mise en gage sont utilisées pour permettre au destinataire d'effectuer des opérations sur le gage. Ceci donne ainsi des preuves ZK pour deux problèmes NP-complets, les problèmes de la somme de sous-ensembles et 3-SAT.

Acknowledgements

I would like to thank Professor Claude Crépeau for his help and guidance in research.

Table of Contents

List of Figures

List of Tables

0.1 List of Abbreviations

ZKPs Zero-Knowledge Proofs i

HAM-ZKP Hamiltonian Cycle ZKP 20

REL-HAM-ZKP Relativistic Hamiltonian Cycle ZKP 30

Chapter 1

Introduction

Imagine being in a foreign country needing to buy food, but your bank card is not accepted for payment. Luckily, there is a bank machine around the corner, however you are unsure whether it can be trusted - after all, there are many instances of fraud in which criminals plant fake bank machines or hide discrete cameras on legitimate machines to obtain the private banking information of the users. Ideally, interacting with a bank machine would not leak any private information that could be used in a malicious way by the creators of the machine at a later time. This is where zero-knowledge proofs (ZKPs) come in.

Zero-knowledge proofs were first introduced by [GMR85]. A couple years later, Blum presented an elegant ZKP for the Hamiltonian cycle problem in [Blu87]. A Hamiltonian cycle is a cycle in a graph which passes through each vertex exactly once. After a graph G is fixed, Blum's protocol allows a prover to convince a verifier that G contains a Hamiltonian cycle without giving away any knowledge of the cycle. Like many other ZKPs, Blum's protocol uses bit commitment, which is essentially a cryptographically secure way for Alice to send Bob a message in a locked box so that he cannot see the message until she sends the key at a later time, but she also cannot change the message after she sends the box. Bit commitment schemes typically rely on computational assumptions such as existence of one-way functions as in [GMW91], however there is a protocol that was proposed in [BCMS98] that only uses spatial separation between parties as its sole assumption to prove security.

In [CL17], the authors adapted Blum's ZKP from [Blu87] to use the bit commitment protocol from [BCMS98]. Chailloux and Leverrier managed to prove that this new protocol is secure against quantum adversaries. One interesting aspect of the bit commitment scheme that they used is its homomorphic properties. In particular, the contents of two commitments can be added together before unveiling, and in general, it is possible to form a new commitment containing any linear combination of the contents of the commitments. For this reason, along with rising public interest in homomorphic encryption, it would be an interesting result to have a ZKP that explicitly uses this homomorphic property while still being secure against quantum adversaries. Similar ideas have been considered for classical adversaries in [BC86] and [BC87], where a perfectly hiding and computationally binding bit commitment scheme is used to create ZKPs that rely on the commitment scheme's homomorphic properties for many steps.

In addition to the homomorphic properties of [BCMS98], the commitment scheme can be used to commit to values other than bits as long as the field $\mathbb{F}_Q$ is chosen carefully. An excellent NP-Complete problem to showcase this homomorphic property of the bit commitment scheme is the Subset Sum problem, which asks the following: Given a set S of positive integers and a target integer k, is there a subset of S which sums to k [KT05]? In Section 3.1, a novel ZKP is given for this problem, and then in Section 3.2, another new ZKP is given for the well-known 3SAT problem which is also NP-Complete [AHU74]. Both proofs use a technique similar to that used in [CL17] in order to prove soundness against quantum adversaries.

Before the protocols are introduced, the relevant quantum information theory and cryptography background material is covered. In addition, to demonstrate the need for quantum proofs of security to newer readers, the magic square game is discussed in Section 2.3. Following this is an introduction to the relativistic bit commitment scheme in Section 2.5. Finally, the two protocols are presented, accompanied by proofs of security.

Chapter 2

Background and Preliminaries

2.1 Quantum Information Theory

2.1.1 The Qubit

The content of this chapter has been adapted from [Wil16]. Classically, information is stored in bits which can take on one of two possible values, either 0 or 1. However, in quantum theory, information can also be stored in qubits, and the state of a qubit is represented by a vector. In particular, any qubit can be expressed as a two-dimensional unit vector with complex entries, meaning it can be written as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \text{ where } |\alpha|^2 + |\beta|^2 = 1.$$

We often call the above vector a *pure* state. Later, we will see *mixed* states, which are represented as matrices as opposed to vectors. In quantum information theory, *Dirac notation* tends to be useful for representing states. It involves using the *ket* $|\cdot\rangle$, which is defined as follows

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Using Dirac notation, the above pure state can be written as $\alpha|0\rangle + \beta|0\rangle$. Typically, the state of an arbitrary qubit is abbreviated as $|\psi\rangle = \begin{bmatrix} \psi_0 \\ \psi_1 \end{bmatrix}$.

A Hilbert space is a vector space with an inner product. In this thesis, we will only deal with finite-dimensional Hilbert spaces and the inner product will always be the usual dot product. Note that any single qubit state lives in the Hilbert space $\mathbb{C}^2$. However, not all vectors in $\mathbb{C}^2$ are valid states; only those vectors with unit norm. In Dirac notation, the conjugate transpose is abbreviated as a *bra*:

$$\langle\psi| := |\psi\rangle^\dagger.$$

The reason for the particular naming of this notation is that the inner product of two vectors can be written as a *bra-ket* $\langle\psi||\phi\rangle$. This allows us to characterize vectors with unit norm as those $|\psi\rangle \in \mathbb{C}^2$ such that $\langle\psi||\psi\rangle = 1$.

2.1.2 Composite Quantum Systems

While computations using only a single qubit can be useful, we will often consider computations with multiple qubits. In the classical case, the state of two bits b_1 and b_2 is usually stored as the pair (b_1, b_2), meaning that the space of the composite system is the Cartesian product $\mathbb{Z}_2 \times \mathbb{Z}_2$. However, in the quantum case, the state of two pure qubit states is described as the tensor product $|\psi_1\rangle \otimes |\psi_2\rangle$ of the two individual pure states, meaning that the space of the composite quantum system is the tensor product $\mathbb{C}^2 \otimes \mathbb{C}^2$, to be defined below. As a reminder, the tensor product of two matrices is given below:

Definition 2.1.1. For any $m \times n$ matrix A and any $p \times q$ matrix B, we define the tensor product $\otimes$ as follows

$$A \otimes B := \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix}$$

where $A_{ij}B$ is a submatrix of size $p \times q$, hence $A \otimes B$ has size $mp \times nq$.

In Dirac notation, we abbreviate the tensor product $|\psi\rangle \otimes |\phi\rangle$ by concatenating the strings $|\psi\phi\rangle$. For example, we can write

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Using the definition of tensor products of matrices, we can now define the tensor product of vector spaces. Note that throughout this section, the vector spaces are over the complex field $\mathbb{C}$.

Definition 2.1.2. If V and W are m and n dimensional vector spaces, then $V \otimes W$ is the vector space of column vectors of size mn.

By equipping $V \otimes W$ with the usual dot product, we obtain a Hilbert space. In general, an n-qubit quantum state is represented as a unit vector in the Hilbert space $(\mathbb{C}^2)^{\otimes n} := \mathbb{C}^2 \otimes ... \otimes \mathbb{C}^2$ with n copies of $\mathbb{C}^2$. Note that $(\mathbb{C}^2)^{\otimes n}$ is the set of all 2^n dimensional vectors with complex entries equipped with the usual dot product.

We will often abbreviate an n-qubit space by H. In addition, if Alice and Bob both have one or more qubits, then we will denote the Hilbert space containing Alice's qubits as H_A and the Hilbert space containing Bob's qubits as H_B. Note that H_A will take the form of $(\mathbb{C}^2)^{\otimes n}$ for some $n \in \mathbb{N}$, and H_B will take the form of $(\mathbb{C}^2)^{\otimes m}$ for some $m \in \mathbb{N}$. To make reading easier, we often label quantum states with subscripts. For example, if $|\psi\rangle \in H_A$ and $|\phi\rangle \in H_B$, we would write the state of the composite system as $|\psi\rangle_A \otimes |\phi\rangle_B$. This state lives in $H_A \otimes H_B$.

We've seen how tensoring two single-qubit states leads to a state in $(\mathbb{C}^2)^{\otimes 2}$. However, one special property is that there are unit vectors in $(\mathbb{C}^2)^{\otimes 2}$ that are not simply the result of tensoring two unit vectors in $\mathbb{C}^2$. These vectors describe entangled states and have very remarkable qualities. Before we describe more about entanglement of quantum states, we give a precise definition below.

Definition 2.1.3. Let H_A and H_B be finite dimensional Hilbert spaces. We say that $|\psi\rangle_{AB} \in H_A \otimes H_B$ is a *separable* state with respect to H_A and H_B when there exist states $|\phi\rangle_A \in H_A$ and $|\gamma\rangle_B \in H_B$ such that $|\psi\rangle_{AB} = |\phi\rangle_A \otimes |\gamma\rangle_B$.

Definition 2.1.4. A state $|\psi\rangle_{AB} \in H_A \otimes H_B$ that is not separable with respect to H_A and H_B is said to be *entangled* with respect to H_A and H_B.

For example, consider the entangled two-qubit state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. We can prove that it is entangled by first rewriting it in vector form $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T$. Next, if it were separable, then by Definition 2.1.3 there would exist two states $\begin{bmatrix} \alpha & \beta \end{bmatrix}^T$ and $\begin{bmatrix} \gamma & \delta \end{bmatrix}^T$ such that

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \otimes \begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix}$$

where the second equality comes from using the definition on the tensor product. Since $\mathbb{C}$ is a field, then in particular $\alpha = 0$ or $\delta = 0$. However, in the first case $\alpha\gamma = 0 \neq \frac{1}{\sqrt{2}}$ and in the second case $\beta\delta = 0 \neq \frac{1}{\sqrt{2}}$. Hence no such states $\begin{bmatrix}\alpha & \beta\end{bmatrix}^T$ and $\begin{bmatrix}\gamma & \delta\end{bmatrix}^T$ exist, meaning that $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is not separable. By Definition 2.1.4, we conclude that $|\psi\rangle$ is an entangled state. This state is one of the four Bell states and is used throughout quantum information theory.

2.1.3 Unitaries

In the same way that there are gates that can be applied to bits, like the NOT gate on a single bit or an AND gate on two bits, there are gates that can be applied to qubits as well. In the classical case, any function from $\{0,1\}^n$ to $\{0,1\}$ can be made into a gate, however in the quantum case there are several constraints. Firstly, a quantum gate must be linear, so its action on an n-qubit state can be represented by multiplication on the left with a $2^n \times 2^n$ matrix. By quantum theory, such a matrix must be a unitary so that the output of a quantum gate is also a valid quantum state. As a reminder, a matrix U is unitary when $UU^\dagger = U^\dagger U = I$, and this means that $U|\psi\rangle$ is a valid quantum state since it has unit norm:

$$(U|\psi\rangle)^\dagger U|\psi\rangle = \langle\psi|U^\dagger U|\psi\rangle = \langle\psi||\psi\rangle = 1.$$

An example of a common two-qubit gate is the CNOT gate

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

The effect of this gate on a two-qubit state can be seen clearly by using Dirac notation with the usual basis of $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. The CNOT gate flips the second qubit conditioned on the first qubit being $|1\rangle$. For example, $\text{CNOT}|00\rangle = |00\rangle$ and $\text{CNOT}|10\rangle = |11\rangle$. Interestingly, if the basis is changed, then the effect of the CNOT gate appears to be reversed. For example, consider the alternate basis $\{| + +\rangle, | + -\rangle, | - +\rangle, | - -\rangle\}$, where $|+\rangle := \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle := \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. If the second qubit is $|+\rangle$, the CNOT acts as the identity on the first qubit, whereas if the second qubit is $|-\rangle$, the CNOT gate flips $|+\rangle$ to $|-\rangle$ and vice versa. For instance, $\text{CNOT}| + +\rangle = | + +\rangle$ and $\text{CNOT}| + -\rangle = | - -\rangle$. In this way, one's perspective heavily influences the apparent action of a quantum gate.

2.1.4 Measurements

Performing a measurement to a quantum state allows us to extract classical information about the state. For example, many quantum algorithms consist of inputting a state into a quantum circuit, then performing a measurement at the end to decide on the outcome of the algorithm. In our representation, a state is just a vector. If we are given an arbitrary state and we would like to learn about its vector representation, then we can perform a measurement. To model an n-qubit measurement, we use a finite collection of $2^n \times 2^n$ matrices $\{M_i\}_{i\in I}$ called measurement operators. These matrices must satisfy

$$\sum_i M_i^\dagger M_i = \mathbb{1}$$

where $\mathbb{1}$ is the identity matrix. The probability of obtaining outcome i when measuring state $|\psi\rangle$ with the measurement operators $\{M_i\}_{i\in I}$ is

$$\Pr\Big[i \Bigm| |\psi\rangle\Big] = \langle\psi|M_i^\dagger M_i|\psi\rangle.$$

One important point is that performing a measurement can alter the state. In particular, after obtaining outcome i from the measurement, the resulting state is

$$\frac{M_i|\psi\rangle}{\sqrt{\Pr\left[i\,|\,|\psi\rangle\right]}}.$$

Remark 2.1.5. Note that two states can not be distinguished by a measurement if they only differ by a constant. In other words, if $|\phi\rangle = a|\psi\rangle$ for some complex number a, then we say that $|\phi\rangle$ is the same state as $|\psi\rangle$. This is because we must have $|a| = 1$, so $a^\dagger a = 1$ hence the outcomes of measurements are the same:

$$\Pr\left[i \,\middle|\, a|\psi\rangle\right] = \langle\psi|a^\dagger M_i^\dagger M_i a|\psi\rangle = \langle\psi|M_i^\dagger M_i|\psi\rangle = \Pr\left[i \,\middle|\, |\psi\rangle\right].$$

For this reason, we consider two states to be equal if they only differ by a constant.

Next is a thought provoking example of quantum measurements. Consider again the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. It is a two-qubit state, so assume Alice has one qubit and Bob has the other. They would both like to perform a measurement on their qubit using the operators $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. First Alice will measure, then Bob will measure. We must therefore describe the action of the measurements to the entire two-qubit state. The measurement operators for Alice's measurement on this two-qubit state become $A_0 = |0\rangle\langle 0| \otimes \mathbb{1}$ and $A_1 = |1\rangle\langle 1| \otimes \mathbb{1}$ since Alice will not apply any measurement to Bob's qubit. To calculate the resulting state and probabilities, we start with the following calculation.

$$\begin{aligned}
A_i|\psi\rangle =& (|i\rangle\langle i| \otimes \mathbb{1})\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
&= \frac{1}{\sqrt{2}}(|i\rangle\langle i| \otimes \mathbb{1})(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle) \\
&= \frac{1}{\sqrt{2}}(|i\rangle\langle i| \otimes \mathbb{1})(|0\rangle \otimes |0\rangle) + \frac{1}{\sqrt{2}}(|i\rangle\langle i| \otimes \mathbb{1})(|1\rangle \otimes |1\rangle) \\
&= \frac{1}{\sqrt{2}}|i\rangle\langle i||0\rangle \otimes \mathbb{1}|0\rangle + \frac{1}{\sqrt{2}}|i\rangle\langle i||1\rangle \otimes \mathbb{1}|1\rangle
\end{aligned}$$

$$= \frac{1}{\sqrt{2}}|i\rangle \otimes |i\rangle = \frac{1}{\sqrt{2}}|ii\rangle$$

The last line follows since $\langle i||j\rangle = \delta_{i,j}$. Then the probability of Alice obtaining outcome i is

$$\Pr\Big[i \Big| \; |\psi\rangle\Big] = \langle\psi|A_i^\dagger A_i|\psi\rangle = \frac{1}{\sqrt{2}}\langle ii|\frac{1}{\sqrt{2}}|ii\rangle = \frac{1}{2}$$

and the resulting state after her measurement is

$$\frac{A_i|\psi\rangle}{\sqrt{\Pr[i\,||\psi\rangle]}} = \frac{\frac{1}{\sqrt{2}}|ii\rangle}{\sqrt{\frac{1}{2}}} = |ii\rangle.$$

Next, Bob measures his qubit. Since his measurement will not apply any gate to Alice's qubit then the measurement operators are $B_j = \mathbb{1} \otimes |j\rangle\langle j|$ for $j \in \{0, 1\}$.

$$\begin{aligned} B_j|ii\rangle &= (\mathbb{1} \otimes |j\rangle\langle j|)(|i\rangle \otimes |i\rangle) \\ &= \mathbb{1}|i\rangle \otimes |j\rangle\langle j||i\rangle \\ &= |i\rangle \otimes |j\rangle\delta_{i,j} = |ii\rangle\delta_{i,j} \end{aligned}$$

The probability that he obtains outcome j is

$$\Pr\Big[j \Big| \; |ii\rangle\Big] = \langle ii|B_j^\dagger B_j|ii\rangle = \langle ii|\delta_{i,j}|ii\rangle\delta_{i,j} = \delta_{i,j}.$$

Therefore he will always obtain outcome i, the same result as Alice. After this measurement, the state will then become

$$\frac{B_i|ii\rangle}{\sqrt{\Pr[i \mid |ii\rangle]}} = \frac{|ii\rangle\delta_{i,i}}{\sqrt{1}} = |ii\rangle.$$

This means that Bob's qubit does not change as a result of him measuring it. This example leads us to locality, which will be described next.

2.1.5 Locality

Working with qubits allows for more than just fast computation. Distant qubits can also be correlated in ways not permitted by classical physics. For example, the experiment in the previous section could be realized by having Alice and Bob meet up to prepare their two-qubit state, then they could travel far apart before Alice performs her measurement on her qubit. As soon as she measures her qubit to obtain outcome i, the second qubit will instantly "collapse" to the same state $|i\rangle$. When Bob measures his qubit, he will of course also get the outcome i. It appears that Alice's qubit is communicating with Bob's qubit, however one fundamental principle in physics is that signalling cannot exceed the speed of light. Indeed, this experiment does not allow for instant communication between Alice and Bob at a distance despite their qubits having a strong correlation.

For many of the protocols that will be given in the thesis, we will refer to *local* parties. By local, we mean that the parties are classical, and they are allowed to share random strings before the protocol begins, then make computations using only resources available in their lab during the protocol. Formally, the parties can be modelled by Turing machines, each of which receiving its own input as well as a shared common random tape. If instead the parties are quantum, then they may share a quantum state before the protocol, than make quantum operations and measurements in their lab during the protocol. Formally, the parties would be modelled as quantum circuits, with each receiving its own quantum input and a shared finite dimensional quantum state. This multi-prover quantum model is described in more detail in [VW16]. These constraints are meant to capture the spirit of not communicating, however the distinction is more subtle than one might expect. For more on this, see [CY17].

2.1.6 Mixed States

Often, the exact representation of a quantum state is not known. For example, imagine Bob plans to send Alice $|0\rangle$ with probability $\frac{1}{4}$, and $|1\rangle$ with probability $\frac{3}{4}$. Knowing this, how can Alice represent this unknown quantum state that she will receive? Well, we can capture the notion of a mixed state using the density operator.

Definition 2.1.6. Let J be a random variable and for each $i \in J$, let $|\psi_i\rangle$ be a state. Then the density operator ρ of the ensemble $\{p(i), |\psi_i\rangle\}_{i\in J}$ is defined as

$$\rho := \sum_{i \in J} p(i)|\psi_i\rangle\langle\psi_i|.$$

For example, the density matrix ρ_A corresponding to Alice's state above would be

$$\rho_A = \frac{1}{4}|0\rangle\langle 0| + \frac{3}{4}|1\rangle\langle 1| = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{3}{4} \end{bmatrix}.$$

One important density matrix that finds endless use in quantum information theory is $\frac{\mathbb{1}}{d}$, often called the maximally mixed state. Note that here, $\mathbb{1}$ is the $2^n \times 2^n$ identity matrix on n qubits and d is the dimension of the Hilbert space, so the maximally mixed state on n qubits is $\frac{\mathbb{1}}{2^n}$. For example, if Bob had instead decided to send Alice $|0\rangle$ or $|1\rangle$ with equal probability, then ρ_A would be $\frac{\mathbb{1}}{2}$. In fact, for any choice of orthonormal basis, if Bob chooses a random state from that basis and sends it to Alice, then her density matrix will be the maximally mixed state.

2.1.7 No-Cloning Theorem

One important principle in quantum information theory is that it is impossible to perfectly copy an arbitrary quantum state. In fact, even arbitrary close approximations cannot be copied. Observe that this contrasts classical information theory since a string of bits can easily be written down and copied. This no-go theorem makes proving the zero-knowledge property much more difficult than in the classical case since a simulator can rewind a classical verifier, however rewinding can not always be performed on a quantum verifier. Many researchers are devising techniques to rewind a quantum verifier under specific conditions such as in [Unr10], however the zero-knowledge protocols presented here involve two provers and rewinding is not necessary. While this multi-prover structure allows authors to easily prove the Zero-Knowledge property, the soundness property is still difficult to prove due to the inability to freely rewind. However, Chailloux and Leverrier showed a technique in [CL17] that bypasses rewinding for proving soundness. These techniques will be used in Sections 3.1 and 3.2.

2.2 Cryptographic Background

In this section, we define two cryptographic objects that will appear throughout the thesis: bit commitment schemes and zero knowledge proofs. We start with a definition of the complexity class NP from [Sip13].

Definition 2.2.1. A verifier for a language A is an algorithm V such that

$$A = \{w \mid V \text{ accepts } \langle w, c\rangle \text{ for some string } c\}$$

The running time of a verifier is measured relative to the length of w, meaning that V is a polynomial time verifier if it runs in polynomial time relative to the length of w.

Definition 2.2.2. **NP** is the class of languages that have polynomial time verifiers.

An example of a language in NP is Hamiltonian Cycle, which we'll denote by HAM. An undirected graph is in HAM if it contains a cycle that passes through every vertex exactly once. For example, the graph below is in HAM since it contains a cycle from $1 \to 3 \to 2 \to 4 \to 1$ shown below with the edges in bold.

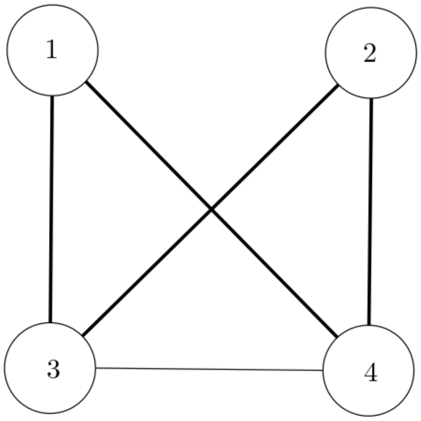

Figure 2.1: Graph with Hamiltonian cycle

2.2.1 Commitment schemes

Cryptographic protocols often require a sort of digital lockbox. These are called commitment schemes. They are analogous to Alice placing a message in a box, locking it, and sending the box to Bob while maintaining possession of the key. Then, when Alice would like Bob to see her message, she sends Bob the key. We give the following definition adapted from Chailloux and Leverrier in [CL17].

Definition 2.2.3. A commitment scheme is an interactive protocol between Alice and Bob with a Commit phase and a Reveal phase.

Commit Phase: Alice chooses a message m that she wants Bob to see only after the unveil phase. Then, they perform a communication protocol that corresponds to this commit phase.

Unveil Phase: In order for Alice to reveal the message m to Bob, they complete a communication protocol that corresponds to this unveil phase. At the end of the protocol, Bob outputs either Accept or Reject.

The commitment scheme is a description (COM, UNV) of the protocol that is followed by the honest parties during the commit and unveil phases of the protocol. In order to be useful, commitment schemes must satisfy two properties: hiding and binding. Roughly speaking, a commitment scheme is hiding if before the reveal phase, Bob can not figure out Alice's message. Below is a formal definition of perfect hiding, but first we introduce some terminology. Generally, Alice and Bob are modelled as Turing machines A and B. If B^* is a (possibly malicious) probabilistic Turing machine, then the random variable $\langle A(m), B^* \rangle$ denotes the output of B^* after A commits the message m to B^*. Finally, M is the message set of possible messages m.

Definition 2.2.4. We say that a commitment scheme is *perfectly hiding* if for any probabilistic machine B^*, the random variables $\{\langle A(m), B^* \rangle\}_{m \in M}$ are identically distributed.

In practice, this means that despite Bob's best efforts, he will not be able to deduce Alice's message before she has revealed it to him. On the other hand, a commitment scheme is binding if Alice cannot change her message after the commit phase. Defining the binding property is a subtle matter. Here, we give a definition of the sum-binding property of commitment schemes, given also in [CL17]. We begin with a cheating strategy (COM', UNV') for Alice when interacting with an honest Bob. Ideally, once Alice has chosen her commitment strategy, there should be only one message to unveil. In a perfect world, the probability of Alice successfully unveiling that message is 1, whereas the probability of Alice deceptively unveiling any other message is less than ϵ. However, note that Alice could just as well choose a probabilistic approach to committing. For example, suppose COM' either commits to 0 or 1, each with probability $\frac{1}{2}$. Then she would be able to successfully unveil a 0 with probability at least a half, and the same would go for unveiling a 1. For this reason, a particular commitment made by a malicious Alice does not necessarily have a message associated with it. With all this in mind, consider the following definition:

Definition 2.2.5. (Sum-Binding) A commitment protocol is ϵ-sum-binding if

$$\forall COM', \quad \sum_{m \in M} \max_{UNV'}(\Pr[\text{Alice successfully unveils } m \mid (COM', UNV')]) \leq 1 + \epsilon$$

In Section 2.5, we will see an example of a commitment scheme with $M = \{0, 1, ..., P-1\}$ which is $1+\frac{4P}{Q^{1/3}}$-sum-binding where Q is the size of the field that will be used for the protocol.

An example of a simple problem with a solution that uses a bit commitment scheme is given in [Blu81] and will be outlined next. Suppose Alice and Bob are getting divorced and must decide how to split all their belongings. They have already settled on who gets the furniture, the house, the kids, but they are unable to decide who gets the car. Eventually, they agree to flip a coin to determine the new owner of the car. However, since Alice and Bob are unable to be in the same room together, they can only communicate over the phone, hence neither is willing to let the other one flip the coin. They need a way to randomly decide who gets the car by communicating only over the phone.

An easy way to solve this using a bit commitment scheme is to have Alice toss a coin to obtain a random bit b, then send Bob a commitment of b over the phone. Bob then tosses a coin to obtain a random bit b', and sends b' to Alice over the phone. Finally, Alice unveils b to Bob over the phone, and if $b = b'$, then Alice walks away with the car, otherwise Bob is the lucky owner. Notice that for any strategies that Alice and Bob use, the two bits will be equal with probability 50%, meaning that this is a fair way to flip a coin over the phone. This is because Bob's bit is chosen after Alice's bit was decided, but since he is unable to see her bit, then their choice will be independent.

2.2.2 Zero-Knowledge Protocols

We saw in the example above of flipping a coin over the telephone that commitment schemes can be very useful building blocks in cryptography. One area that they find endless usage is in zero-knowledge protocols (ZKPs). A ZKP is a communication protocol between two parties: a prover and a verifier. The prover's goal is to convince the verifier of a fact without giving away any more information than the validity of that fact. For example, suppose a prover wants to convince a red-green colourblind verifier that two marbles are not the same colour. The verifier does not trust the prover, so he gets the prover to answer a series of questions. The verifier holds a marble in each hand, then hides his hands behind his back and randomly decides whether or not to switch the marbles. When he puts his hands out in front again, the prover must say whether the marbles were swapped or not. This is repeated multiple times with the prover guessing whether or not the marbles were swapped in each round. If the prover succeeds for many rounds, the verifier will be convinced that the marbles are indeed different, however the verifier will not know where that difference lies. For example, after many rounds of the protocol, the verifier will still not know which marble is red and which one is green, or whether those are even the colours of the marbles.

In many ZKPs, a single verifier will interrogate multiple provers. Generally, we require that the provers cannot communicate, however they can use quantum resources such as entangled states and local operations. The provers are represented as Turing machines $P_1, ..., P_k$ and the verifiers are represented as one Turing machine V. The following model is given in [BOGKW88]

Definition 2.2.6. Let $P_1, ..., P_k$ be Turing machines which are computationally unbounded and V be a probabilistic polynomial time Turing machine. All machines have a read-only input tape, a work tape and a random tape. In addition, $P_1, ..., P_k$ share an infinite read-only random tape of 0's and 1's. Every P_i has one write-only communication tape on which it writes messages for V. V has k write-only communication tapes. On communication tape i, V writes messages to P_i. In addition, V has one output tape, generally used to write Accept or Reject, however no restriction is placed on the contents of this tape. We call $(P_1, ..., P_k, V)$

a k-prover interactive protocol, and $\langle P_1, .., P_k, V\rangle(x)$ is a random variable referring to the content of the output tape of V, upon its termination, after interacting with $P_1, ..., P_k$ on input x.

As an example, we will introduce Blum's ZKP for HAM given in [Blu87]. First, we give notation that will be used in the protocol. Suppose we have a graph $G = (V, E)$.

Notation	Meaning
V	set of vertices $\{v_1, ..., v_n\}$
E	set of edges of the graph
Π	random permutation on vertices
$\Pi(G)$	graph $(V, \{(\Pi(u), \Pi(v)) \mid (u, v) \in E\})$
$M_{\Pi(G)}$	adjacency matrix of $\Pi(G)$
C	a Hamiltonian cycle $\{(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), ..., (v_{i_n}, v_{i_1})\}$
$\Pi(C)$	permuted Hamiltonian cycle $\{(\Pi(u), \Pi(v)) \mid (u, v) \in C\}$

Table 2.1: Notation used in ZKP for Hamiltonian Cycle

The following protocol allows a prover that knows a Hamiltonian cycle of G to convince a verifier that G has a Hamiltonian cycle without giving away any knowledge of its whereabouts. It assumes the existence of a bit commitment protocol.

The ZKPs considered in this thesis must satisfy three conditions: completeness, soundness, and zero-knowledge. For each of these conditions, the above HAM-ZKP example will be discussed. The first two conditions are given below from [BOGKW88].

Definition 2.2.7. We say that a language L has a k-prover interactive proof-system (IPS) if there exists an interactive BPP machine V such that:

1. (Completeness) $\exists P_1, ..., P_k$ such that $(P_1, ...P_k, V)$ is a k-prover interactive protocol and $\forall x \in L$, prob(V accepts input x) $\geq 2/3$.

2. (Soundness) $\forall P_1, ..., P_k$ such that $(P_1, ...P_k, V)$ is a k-prover interactive protocol and $\forall x \notin L$, prob(V accepts input x) $\leq 1/3$.

Hamiltonian Cycle ZKP (HAM-ZKP) using bit commitment
1. The prover picks a random permutation $\Pi : V \to V$. He then sends the verifier a commitment of each bit of $M_{\Pi(G)}$.
2. The verifier replies with a challenge bit $chall \in \{0, 1\}$.
3. If $chall = 0$ then the prover unveils $M_{\Pi(G)}$ and sends Π to the verifier. If $chall = 1$ then the prover unveils only the bits of $M_{\Pi(G)}$ that correspond to $\Pi(C)$ in $\Pi(G)$.
4. If $chall = 0$, the verifier confirms that applying Π to G yields the unveiled permuted adjacency matrix $M_{\Pi(G)}$. If $chall = 1$, the verifier confirms that each unveiled bit is a 1 and that they form a Hamiltonian cycle. If any check fails, then the verifier rejects, otherwise the verifier accepts.

Table 2.2: ZKP for Hamiltonian Cycle

If the bound of 2/3 is changed to 1, then the condition is known as perfect completeness, which essentially states that if both the provers and the verifier follow the protocol, then the verifier will accept. The HAM-ZKP has perfect completeness. Indeed, if the honest prover P is given a graph with a Hamiltonian Cycle, then since P has unbounded computation, P can find the cycle, then proceed with the steps of the protocol and the checks by the honest verifier V will succeed.

On the other hand, soundness asserts that no provers should be able to successfully cheat. Suppose that a graph without a Hamiltonian cycle is given to a dishonest prover P'. Then since this graph has no Hamiltonian cycle, the commitment of P' cannot be both a permuted version of G and containing a Hamiltonian cycle at the same time. If the dishonest prover's commitment is neither of the above, then the probability that the unveil successfully passes the verifier's check is essentially 0. However, if the prover committed to one of the above, then since the honest verifier V randomly chooses which question to ask, then with probability essentially $\frac{1}{2}$, the commitment that prover P' unveils will not pass the check of the honest verifier V.

Note that the upper bound in Definition 2.2.7 is $\frac{1}{3}$, whereas our analysis obtained a value of $\frac{1}{2}$. This may appear to not satisfy soundness, but by repeating the protocol twice, then the probability that any prover P' correctly passes in two rounds is $(\frac{1}{2})^2 = \frac{1}{4}$, which is below $\frac{1}{3}$. In this way, sequential repetition can reduce the soundness error of any protocol by repeating a polynomial number of times since we require the parties to communicate for at most a polynomial number of steps. To avoid having to repeat protocols, we generally replace the $\frac{1}{3}$ in Definition 2.2.7 with $1 - \frac{1}{p(|x|)}$ for some polynomial p that takes as input the size of the instance x. Repeating such a protocol $2p(|x|)$ times yields a probability of acceptance at most $(1 - \frac{1}{p(|x|)})^{2p(|x|)} \leq e^{-\frac{1}{p(|x|)}2p(|x|)} = e^{-2} < \frac{1}{3}$ as desired, still in polynomial time.

Finally, we say that a protocol has the Perfect Zero-Knowledge property if it satisfies the following definition from [Gol01].

Definition 2.2.8. Definition 4.3.6 (Perfect Zero-Knowledge, Liberal Formulation): We say that $(P_1, ..., P_k, V)$ is perfect zero-knowledge in the liberal sense if for every probabilistic polynomial-time interactive machine V^* there exists an expected polynomial-time algorithm S^* such that for every $x \in L$, the random variables $\langle P_1, ..., P_k, V^* \rangle(x)$ and $S^*(x)$ are identically distributed.

The above definition needs some unpacking. Firstly, what does it mean intuitively for a protocol to be perfect ZK? It means that regardless of the questions asked by any verifier, malicious or not, the verifier cannot gain additional knowledge about the witness. However, quantifying *additional knowledge* can be tricky. One way to circumvent this is to require that the entire transcript between the prover(s) and verifier should be easy to simulate without any prover. For this reason, Definition 2.2.8 refers to the output of the verifier, which may be a transcript since no restriction is placed on the output of the verifier. In other words, for a protocol to be zero-knowledge, the content of any transcript between a potentially dishonest verifier and the honest prover(s) needs to be producible by only using the potentially dishonest verifier and polynomial extra work. If this is the case, then the transcript cannot contain any additional knowledge.

Formally, V refers to the Turing machine describing the actions of the honest verifier, hence a dishonest verifier is another Turing machine, V^*. Importantly, since V^* is programmed to communicate with the prover(s), then simulating a transcript of V^* and $P_1, ..., P_k$ must be done by another algorithm S^*. For obvious reasons, this algorithm S^* is referred to as the Simulator, and its program depends on V^*, but not $P_1, ..., P_k$.

We will now briefly explain why the HAM-ZKP has perfect ZK. Intuitively, the verifier either sees a random permutation of the graph or a random cycle with no context, both giving no additional information. This is captured in the formal proof outlined briefly below. Suppose we have a possibly malicious verifier V^*. We define a simulator S^* that starts by guessing a random challenge $chall' \in \{0, 1\}$, then sends a commitment to V^* that will pass the check if the challenge $chall$ ends up being equal to $chall'$. More precisely, if $chall' = 0$, then S^* commits to a permuted adjacency matrix of the graph G, and if $chall' = 1$, then S^* commits to a matrix of all 1's. This way, if $chall = chall'$, then the simulator is fortunate since it will be able to unveil a response that appears correct to V^* in order to imitate an honest prover's response. If $chall \neq chall'$, which will happen with probability $\frac{1}{2}$, then S^* will rewind V^* to the start of the protocol to reattempt the process. After rewinding, S^* can make a new random choice of $chall'$, then proceed with committing as before. Each time that $chall \neq chall'$, the simulator must rewind the verifier.

The two challenges are expected to match with probability $\frac{1}{2}$, hence the simulator runs in expected polynomial time since an average of 2 attempts will be needed for the simulator to succeed in a round. Since the simulator will erase the failed attempts and not record the rewinding on the transcript, then the final transcript outputted by S^* will only show the successful rounds. For this reason, the distribution of transcripts output by S^* will be identical to the distribution of transcripts output by interactions between V^* and P.

Note that this ZKP used only one prover and the proof of zero-knowledge required rewinding the verifier. In the two-prover ZKPs presented in this thesis, the simulator will not need to rewind the verifier(s), hence proving zero-knowledge in the quantum setting will be easier for reasons described in Section 2.1.7.

2.3 Pseudo-Telepathy games

In this section, we introduce a multi-party game, also presented in [BBT05], that is perfectly winnable with quantum players but not with classical players. The purpose of including this example is to highlight the importance of proving soundness against quantum adversaries. In the magic square game that will be presented, it will be clear to the reader that a classical winning strategy is not possible. However, there exists a winning quantum strategy.

The magic square game is a two-player game in which Alice and Bob work together to answer certain questions about a 3×3 matrix. In particular, Alice will be asked for the entries in a certain row, while Bob will be asked for the entries in a certain column. The entries can only be 0 or 1, and they win if the following three conditions are satisfied:

(a) Their common entry is the same.

(b) The sum of the Alice's row entries is even.

(c) The sum of the Bob's column entries is odd.

Even more, Alice and Bob are separated during the game. For details on this, refer to the section on Locality (2.1.5). For this reason, they may not send each other messages during the game and therefore must agree on a strategy beforehand. However, we'll now explain why no local classical winning strategy exists using a proof by contradiction.

2.3.1 Impossibility of a Classical Winning Strategy

Since the winning probability of any randomized strategy cannot exceed that of the best deterministic strategy, then we consider a winning local deterministic strategy. This is because any randomized strategy can be viewed as a combination of deterministic strategies, and so the winning probability of the randomized strategy is the average of the winning probabilities of the deterministic strategies. The responses of Alice and Bob correspond to 3×3 matrices. By condition (a), their two matrices must be identical since any entry could be common depending on the row and column asked. However, the conditions (b) and (c) are contradictory since summing all the entries in the matrix will be even by (b), but odd by (c). For example, if Alice and Bob's strategy corresponds to the matrix below, then in order to satisfy (b), the last bit would have to be 0, but to fulfill (c), the last bit would have to be 1.

0	1	1
1	0	1
0	0	?

Table 2.3: Impossibility of winning classical strategy for Magic Square

2.3.2 Quantum Winning Strategy

This is a surprising result, demonstrating the strength of quantum correlations. The strategy starts with a shared quantum state. The first two qubits of the following state belong to Alice, whereas the last two belong to Bob.

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

When Alice is asked for the three entries of row x, she applies the operation A_x to her two qubits. Similarly, when Bob is asked for the three entries of column y, he applies the operation B_y to his two qubits.

$$A_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} i & 0 & 0 & 1 \\ 0 & -i & 1 & 0 \\ 0 & i & 1 & 1 \\ 1 & 0 & 0 & i \end{bmatrix} \qquad A_2 = \frac{1}{2}\begin{bmatrix} i & 1 & 1 & i \\ -i & 1 & -1 & i \\ i & 1 & -1 & -i \\ -i & 1 & 1 & -i \end{bmatrix} \qquad A_3 = \frac{1}{2}\begin{bmatrix} -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix}$$

$$B_1 = \frac{1}{2}\begin{bmatrix} i & -i & 1 & 1 \\ -i & -i & 1 & -1 \\ 1 & 1 & -i & i \\ -i & i & 1 & 1 \end{bmatrix} \qquad B_2 = \frac{1}{2}\begin{bmatrix} -1 & i & 1 & i \\ 1 & i & 1 & -i \\ 1 & -i & 1 & i \\ -1 & -i & 1 & -i \end{bmatrix} \qquad B_3 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}$$

Next, Alice and Bob both measure their two qubits in the standard basis. This will give Alice two bits a_1, a_2, and Bob two bits b_1, b_2. Alice will output $(a_1, a_2, a_1 \oplus a_2)$, and Bob will output $(b_1, b_2, 1 \oplus b_1 \oplus b_2)$. It is clear that this strategy will satisfy conditions (b) and (c) since

$$a_1 \oplus a_2 \oplus (a_1 \oplus a_2) = 0 \quad \text{and} \quad b_1 \oplus b_2 \oplus (1 \oplus b_1 \oplus b_2) = 1.$$

On the other hand, it is more work to show that condition (a) is satisfied. One example will be shown here. Suppose Alice was asked for row 3 and Bob was asked for column 3. We first compute the state $A_3 \otimes B_3|\psi\rangle$.

$$\frac{1}{2}\begin{bmatrix} -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix} \frac{1}{2}(|0011\rangle - |0110\rangle - 1001\rangle + 1100\rangle)$$

$$= \frac{1}{4\sqrt{2}} \left(\begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} - \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$= \frac{1}{4\sqrt{2}} \begin{bmatrix} 0 & -2 & 2 & 0 & 2 & 0 & 0 & 2 & 2 & 0 & 0 & -2 & 0 & 2 & 2 & 0 \end{bmatrix}$$

$$= \frac{1}{2\sqrt{2}} (-|0001\rangle + |0010\rangle + |0100\rangle + |0111\rangle + |1000\rangle - |1011\rangle + |1101\rangle + |1110\rangle)$$

Next, Alice measures her two qubits of $A_3 \otimes B_3|\psi\rangle$ to obtain the bits a_1, a_2 and Bob measures his two qubits to get the bits b_1, b_2. We check that in each case, $a_3 = b_3$. In Table 2.4 we show the possible measurement outcomes, where the third bits are computed as described above to satisfy conditions (b) and (c).

Alice's Results	000	000	011	011	101	101	110	110
Bob's Results	010	100	001	111	001	111	010	100

Table 2.4: Measurement results of quantum strategy for Magic Square

Therefore Alice and Bob's answers will match in the third positions and win the game in this case.

In a world where quantum computation is quickly advancing, any new cryptographic protocol must pass stringent security measures. This thought-provoking pseudo-telepathy game highlights the importance of proving that such protocols are indeed secure against quantum adversaries, even if they appear to be secure in the classical case.

2.4 Related Works

The protocol in 3.1 is mainly based off of the Relativistic 2-prover protocol given in [CL17]. In the same paper, the authors prove a strong lower bound regarding consecutive measurements of quantum states. They proceed to apply this theorem to bound the soundness error of their two-prover ZKP for HAM, originating from Blum's protocol in [Blu87]. This protocol has a structure common to many other ZKPs, called a Σ-protocol. It is defined by Unruh in [Unr10] where the author discusses proofs of knowledge.

Definition 2.4.1. A Σ-protocol is a 3-message ZKP between a prover and a verifier consisting of a commitment *comm* from the prover, then a challenge *chall* from the verifier, and finally a response *resp* from the prover.

The security of Σ-protocols often rely on a property called special soundness.

Definition 2.4.2. A Σ-protocol has special soundness when there exists a polynomial time algorithm K that computes a witness given two accepting conversations of the same commitment with different challenges.

For example, the ZKP for the Hamiltonian cycle problem (2.2) has special soundness since the response to $chall = 0$ gives away the permutation and the response to $chall = 1$ indicates the location of the Hamiltonian cycle in the permuted graph. By combining both responses, the Hamiltonian cycle of the original graph can be easily reconstructed. On the other hand, an example of a Σ-protocol without special soundness would be the following protocol from [GMW91]. It is a ZKP for the 3-Colouring problem, which concerns labelling each vertex of a graph with an element of $\{0, 1, 2\}$ such that no two adjacent vertices have the same label. The protocol goes as follows:

This protocol does not have special soundness because if two different questions are answered, then the colours of at most 4 vertices are revealed, but it is unclear how the

ZK Protocol for 3-COL using a commitment scheme
1. The prover starts with a 3-colouring and picks a random permutation $\Pi : \{0, 1, 2\} \to \{0, 1, 2\}$ to permute the colours in the graph. He then sends the verifier a commitment for each vertex indicating the colour after applying Π.
2. The verifier replies with a random edge $(i, j) \in E$.
3. The prover unveils the permuted colours of the ith and jth vertices.
4. The verifier confirms that the colours of the endpoints are different. If not, then the verifier rejects, otherwise the verifier accepts.

Table 2.5: ZKP for 3COL

rest of the vertices should be coloured. For example, if there is a triangle between v_i, v_j, v_k in the graph, then having the colours of both endpoints of (i, j) and (i, k) does not give any new information since all three colours must be different. For a protocol to have special soundness, every pair of responses should allow an extractor to construct a witness, assuming the commitments were the same and the challenges were different.

It is known that the Hamiltonian cycle problem is NP-complete, therefore the existence of Blum's ZKP (2.2) immediately guarantees that any problem in NP has a ZKP. However, Blum's construction relies on the existence of a bit commitment scheme. If we choose only a computationally hiding commitment scheme, defined in [Gol01], then the protocol is only zero-knowledge for a certain period of time. In fact, even if the verifier could not compute the committed bits during the protocol, the verifier can continue to compute long after the protocol has finished. This means that with enough time and possibly faster computers, all of the committed information in the protocol could potentially be revealed. In many contexts, this information should always stay private, even after the protocol is finished, hence this is not a desirable property for zero-knowledge protocols.

For the reason described above, we prefer to use perfectly-hiding protocols. However, there is a balance between the hiding and binding properties of any bit commitment scheme. If we restrict ourselves to the typical model of a single prover and single verifier communicating through a perfect channel, then if a bit commitment scheme is perfectly hiding, the binding property can be broken with a computationally unbounded prover. This can be seen intuitively as follows. Note this is merely a rough idea of the proof since it is lacking rigour for the sake of brevity. Assume the commitment scheme is perfectly hiding, and Alice sends a commitment c to Bob. Then a computationally unbounded Bob could find two message-reveal string pairs $(m_1, r_1) \neq (m_2, r_2)$ that both produce c as a commitment. This follows because if Bob could only find one message-reveal string pair that produced c as a commitment, then the commitment scheme would not be perfectly hiding. The issue is that a computationally unbounded Alice could also find $(m_1, r_1) \neq (m_2, r_2)$ for the reveal phase and break the binding property.

This means that if a commitment scheme is chosen to be perfectly hiding, then the scheme can be computationally binding at best, where the computationally binding property is defined in [Gol01]. Note that the definitions of computational hiding and binding are omitted from this thesis, however one can think of them as assumptions that certain problems cannot be solved in polynomial time. As always, cryptographers try to limit their assumptions when designing protocols, so choosing a computationally binding commitment scheme is not ideal either. For example, the long standing factoring problem is soon to be insecure with the arrival of efficient quantum computers that use Shor's algorithm [Sho94]. Fortunately, these issues can be avoided by instead using a relativistic commitment scheme like the one in [LKB$^+$15]. It requires Alice and Bob to each split into two parties: A_1, A_2 and B_1, B_2, where each party is local. Then, they must follow the protocol given in 2.5. This change of structure from that of typical commitment schemes results in perfect hiding and statistical binding for the relativistic bit commitment scheme. The adjective, *relativistic*, refers to the assumption of distanced local provers. This assumption is in fact possible to enforce by distancing the verifiers, then requiring each response from the provers to be within a certain time period. This way, the verifiers can be sure that there was not enough time for any question to travel to the other prover before the response was received from the appropriate prover.

In [CL17], the authors adapted Blum's HAM-ZKP to use a relativistic bit commitment scheme requiring multiple provers, described in 2.5. We will denote the protocol of [CL17] by Relativistic Hamiltonian Cycle ZKP (REL-HAM-ZKP). Even though Blum's HAM-ZKP and the relativistic bit commitment scheme are both secure on their own, combining the two protocols may not necessarily be secure against quantum adversaries, hence it requires its own proof. Fortunately, the authors of [CL17] managed to prove this by relating the probability of provers successfully cheating their protocol with the probability of provers winning a slightly different game. They called this related game G_{coup}. Here is the formal definition of the game that dishonest provers play when analyzing soundness of REL-HAM-ZKP.

Definition 2.4.3. A game $G = (I_A, I_B, O_A, O_B, V)$ is defined by

- 2 input sets I_A,I_B which are respectively Alice's and Bob's input sets.
- 2 output sets sets O_A, O_B which are respectively Alice's and Bob's output sets.
- A valuation function $V : I_A \times I_B \times O_A \times O_B \rightarrow \{0, 1\}$ which indicates whether the game is won for some fixed inputs and outputs. The game is won if the value of V is 1.

Next is the definition of the related game. G_{coup}.

Definition 2.4.4. For any game $G = (I_A, I_B, O_A, O_B, V)$ on the uniform distribution we define G_{coup} as follows:

- Alice receives a random $x \in I_A$. Bob receives a random pair of different inputs (y, y') from I_B.
- Alice outputs $a \in O_A$. Bob outputs $b, b' \in O_B$.
- They win the game if $V(x, y, a, b) = V(x, y', a, b') = 1$.

The twist with G_{coup} is that the second prover must answer two questions at once, making it more difficult for Alice and Bob to win G_{coup}. In fact, the maximum probabilities of winning the two games are related. If we denote the maximum probability of winning a game H among all local classical strategies by $\omega(H)$, then we have the following result.

Proposition 2.4.5. *For any game G with questions asked uniformly at random and $I_B = \{0,1\}$, the classical winning probabilities have the following relationship: $2\omega(G) - 1 \leq \omega(G_{coup})$.*

We will present our own proof of this result. It is a classical version of the quantum result proved in [CL17].

Proof. Suppose Alice and Bob can win G with probability $\omega(G)$. Then we will design a strategy for G_{coup} that wins with probability at least $2\omega(G) - 1$. On input $(x, (y, y'))$, Alice and Bob do the following:

- Alice runs on x and outputs a
- Bob runs on y and outputs b
- Bob is re-winded
- Bob runs on y' and outputs b'.

Note that since Bob is entirely classical, he can be re-winded without any issues. Now, we analyze the probability that both answers are correct. In other words, what is the probability that $V(x, y, a, b) = 1$ and $V(x, y', a, b') = 1$?

Letting $n := |I_A|$, we denote $v, w \in [0,1]^n$ to be the probability vectors such that

$$v_x = \mathbb{E}[V(x, 0, A(x), B(0))] \quad \text{and} \quad w_x = \mathbb{E}[V(x, 1, A(x), B(1))]$$

where $A(x), B(y)$ are Alice and Bob's outputs on input x and y. The expectation is taken over Alice and Bob's random coin flips. This allows us to write the probability of winning

G as

$$\omega(G) = \mathop{\mathbb{E}}_{x,y}[V(x, y, A(x), B(y))] = \frac{v \cdot \vec{1} + w \cdot \vec{1}}{2n}.$$

Then since winning G_{coup} requires succeeding for both inputs, we get

$$\omega(G_{coup}) \geq \mathop{\mathbb{E}}_{x}[V(x, 0, A(x), B(0))V(x, 1, A(x), B(1))] = \frac{v \cdot w}{n}.$$

Then by noting that each entry of v and w is between 0 and 1, we can obtain the result as follows:

$$\begin{aligned} &0 \leq (\vec{1} - v) \cdot (\vec{1} - w) = v \cdot w - v \cdot \vec{1} - w \cdot \vec{1} + \vec{1} \cdot \vec{1} = v \cdot w - 2n\omega(G) + n \\ \implies &2\omega(G) - 1 \leq \frac{v \cdot w}{n} \leq \omega(G_{coup}). \end{aligned}$$

□

The above proposition for the classical case relies on rewinding Bob. In the quantum case, rewinding cannot always be done after measuring the output b since this would allow for copying of quantum states. However, the authors of [CL17] managed to prove a theorem that bounds the amount of error that arises from making two consecutive measurements on a quantum state. They applied this theorem to a strategy for G_{coup} that involves running Bob on both inputs consecutively with no rewinding in between. This gave them an analogous quantum result to Proposition 2.4.5. However, before stating their result, we quickly introduce a definition.

Definition 2.4.6. Let $S \in \mathbb{N}$. We say that a game is S-projective if Bob has at most S possible outputs to win the game for any possible x, y, a. So $\max_{x,y,a} |\{b \mid V(x, y, a, b) = 1\}| \leq S$.

Now we can state the result of [CL17].

Proposition 2.4.7. *For any game G on the uniform distribution which is S-projective, we have*

$$\omega^*(G_{coup}) \geq \frac{1}{64S}\left(\omega^*(G) - \frac{1}{|I_B|}\right)^3.$$

The proof begins in the same way as the proof above of Proposition 2.4.5, building a strategy of G_{coup} using a strategy for G. Their proof then uses the bound on consecutive measurements of quantum states in order to lower bound the success of their G_{coup} strategy.

The purpose of deriving Proposition 2.4.7 is to upper bound the success probability of cheating quantum provers in REL-HAM-ZKP. This allowed the authors of [CL17] to prove that their protocol was sound. The convenience of G_{coup} is that even in the quantum case, upper bounding $\omega^*(G_{coup})$ can be relatively simple, then it can be translated into an upper bound for $\omega^*(G)$ with the help of their proposition. Their proof technique will be used to prove soundness for the ZK protocols proposed in Sections 3.1 and 3.2.

Remark 2.4.8. When analyzing the proof, it becomes apparent that it can be applied to any protocol with a similar enough structure. In particular, the proof should extend to any Σ-protocol with special soundness. For example, very similar ZK-protocols to Blum's exist for the k-clique problem, the k-independent set problem, the k-longest path problem, and the subgraph problem. Using a similar contruction to that of the REL-HAM-ZKP, multi-prover ZK-protocols for all of these problems can be directly proven to be sound against quantum provers.

2.5 Properties of the Relativisitic Commitment Scheme

In [BCMS98] the authors introduce a bit commitment scheme involving two provers and two verifiers. Several variations of the protocol have had their security examined in [Ken05], [CSST11], and [LKB+15]. Here, we will present the scheme, a brief explanation to why it is hiding and binding, then finally explore some of its useful properties. The scheme requires two separated provers P1 and P2, as well as two verifiers V1 and V2. The operations will be over the field $\mathbb{F}_Q$ for some large prime power Q. Suppose that the provers want to commit to an element $b \in \mathbb{F}_Q$. They choose a random $c \in \mathbb{F}_Q$, and the protocol goes as follows:

Commit Phase

1. V1 chooses a random $a \in \mathbb{F}_Q$ and sends a to P1.

2. P1 replies with $w := ab + c$ to V1.

Now the provers have committed to b. When the provers would like the verifiers to know their hidden value b, then they complete the unveil phase:

Unveil Phase

1. P2 sends b, c to V2.

2. V1 and V2 verify that $w = ab + c$.

This protocol is illustrated in Figure 2.2.

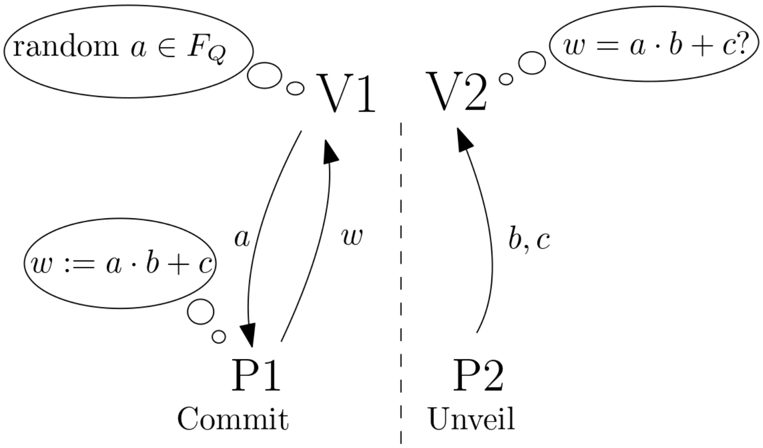

Figure 2.2: Relativistic Two-Prover Bit Commitment Scheme

We'll now explain why the protocol is perfectly hiding. Once V1 has sent a and received w, then for each possible b that P1 could be committing to, there is a unique c such that $c = w - ab$. Since b and c are inaccessible to V1 and V2 before the unveil phase, this means that in Definition 2.2.4, B^* is sending an a then receiving a completely random w, hence

B^*'s outputs for any message b must be identically distributed. For this reason, the protocol is perfectly hiding.

Now we discuss why the protocol is binding. In other words, what keeps the provers from changing their mind from one secret b to another secret b' after they have committed? The formal proof of the sum-binding property against quantum parties is given in [CL17], therefore we will merely give an intuition. If the commitment was w, then to open to the secret b, P2 would have to send $(b, w - ab)$ to V2, whereas to open to the secret b', then P2 would have to send $(b', w - ab')$ to V2. This means that if the provers could unveil both to b and to b', then P2's outputs could be combined in order to guess a since

$$((w - ab') - (w - ab))\,(b - b')^{-1} = a.$$

However, P2 should only be able to guess a with probability $\frac{1}{Q}$ since it was chosen randomly by the verifiers and only sent to P1 who is separated from P2. As mentioned, the quantum case is analyzed in [CL17]. If the set of messages is restricted to $[P] = \{0, 1, ..., P-1\}$, then the protocol is ϵ-sum-binding with $\epsilon = 1 + \frac{4P}{Q^{1/3}}$. For this reason, when using this protocol, the set of messages $[P]$ should be chosen first, then a sufficiently large field $\mathbb{F}_Q$ can be chosen to make ϵ small. The authors arrived at this bound by comparing it with the well-known $CHSH_Q(P)$ game and using existing work on its quantum winning probability.

One can notice that the value w calculated by P1 is an affine function of b. This turns out to be very useful when designing protocols. Imagine a prover commits to b and b', but they want to have the option to unveil the sum $b + b'$ and nothing else. Then by combining the commitments, the provers and the verifiers are able to transform the commitments of b and b' into a commitment of the sum $b + b'$. More specifically, if they complete the Commit phase so that the verifiers possess the values

$$w = a \cdot b + c \text{ and } w' = a' \cdot b' + c'$$

then as long as $a = a'$, the verifiers can add the values they receive to obtain a commitment $w + w'$ of $b + b'$ since

$$w + w' = (a \cdot b + c) + (a \cdot b' + c') = a(b + b') + (c + c').$$

From the expression above, one can see that the key needed to unveil $b + b'$ is $c + c'$. This property is exploited in Section 3.1 where it is the primary mechanism for a simple ZKP for the subset sum problem. Note that this behaviour can be generalized to any linear combination of commitments. If the provers have committed to b and b' but want to unveil $db + d'b$ for some $d, d' \in \mathbb{F}_Q$, then they can have the verifiers combine the commitments in the same way, $dw + d'w$, and then unveil using the key $dc + d'c'$ since the following holds:

$$dw + d'w' = d(a \cdot b + c) + d'(a \cdot b' + c') = a(db + d'b') + (dc + d'c').$$

This can be useful in the case of taking $d = 1$ and $d' = -1$ to prove that two commitments are equal. Indeed, if P1 commits to b and b' and then P2 decides to convince the verifiers that $b = b'$, then he can send $0, c - c'$ for the verifiers to check that $w - w' = a(0) + (c - c')$. This trick is the main tool used in the ZKP of Section 3.2 for 3SAT.

Chapter 3

Main Results

3.1 A ZKP for the Subset Sum Problem

3.1.1 General Idea

In this section, we'll introduce a two-prover ZK proof of the Subset Sum Problem that relies on the affine properties of the relativistic bit commitment. We'll start by stating the problem:

Subset sum problem: Given a set $\{s_1, ..., s_n\}$ of positive integers, is there a subset that sums to k? A solution is a binary vector $v \in \mathbb{F}_2^n$ for which $\sum_{i=1}^{n} v_i s_i = k$. This version of the subset-sum problem is NP-complete [KT05].

The intuition for why the following protocol works is quite simple. Imagine Alice is trying to convince Bob that the set $\{1, 4, 5, 7, 8\}$ has a subset that sums to 14, without giving away the subset. She sets up two rows of 5 upside-down cups on a table, and in the ith column, one cup has no marbles and the other cup has s_i marbles. This is done randomly and independently so that Bob does not know which cup is empty in each column. At this stage, the upside-down cups may have an arrangement of marbles similar to Figure 3.1.

Next, Bob can ask Alice either to lift each cup up one-by-one in order for him to check that Alice set it up properly, or he can ask her to slide one cup from each column to the edge

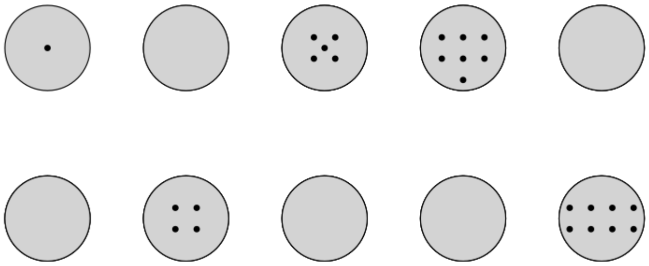

Figure 3.1: Original arrangement of marbles

of the table, then knock those cups into a bowl in order for him to count the 14 marbles. This second option is represented in Figure 3.2.

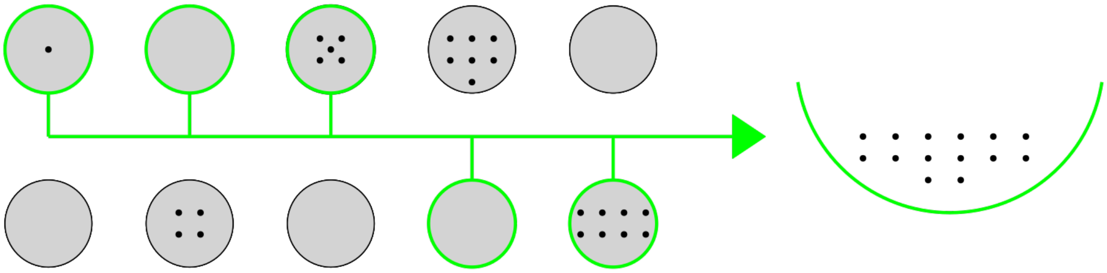

Figure 3.2: Action of Alice if Bob asks to see that the marbles sum to k

Since the cups are knocked into the bowl quickly, he has no idea which marbles came from which cups, but he can count the number of marbles in total. By repeating this experiment multiple times with new random arrangements in each round, Bob will gain confidence that there is a subset that sums to 14, but he will never get any clues to which subset it may be.

3.1.2 Protocol

We describe the notation in Table 3.1 that will appear in the protocol. The operations of the protocol will be over the field $\mathbb{F}_Q$ for some large prime Q that exceeds the sum of all the elements in the set s. This way, the integers of the set s and their sums can be interpreted as values of $\mathbb{F}_Q$. The operation $\cdot$ is scalar multiplication defined entry-wise: $a \cdot (x_1, ..., x_n) := (ax_1, ..., ax_n)$.

Notation	Purpose	Data-Type
c_0, c_1	keys for rows of cups	$\mathbb{F}_Q^n$
w_0, w_1	encryption of rows of cups	$\mathbb{F}_Q^n$
a	randomly chosen by V	$\mathbb{F}_Q$
s	input set	$\mathbb{F}_Q^n$
k	subset sum target	$\mathbb{F}_Q$
v	solution to the problem	$\mathbb{F}_2^n$
x	indicates cups for solution	$\mathbb{F}_2^n$
z	indicates empty cups	$\mathbb{F}_2^n$
c'	sum of keys of cups of x	$\mathbb{F}_Q$
$chall$	randomly chosen by V	$\mathbb{F}_2$
$\cdot$	scalar multiplication	operation
$*$	entry-wise multiplication	operation

Table 3.1: Notation used in ZKP for Subset Sum

We now introduce a ZKP between two provers and verifiers. Assuming the two provers know a witness v, they will be able to convince two verifiers that there exists a solution to the given subset sum problem instance. Before engaging in a round of the protocol, P1 and P2 share random vectors $c_0, c_1 \in \mathbb{F}_Q^n$ and $z \in \mathbb{F}_2^n$. The protocol is described in 3.2.

Two-Prover, Two-verifier Subset Sum ZK protocol
1. V1 sends P1 a random value $a \in \mathbb{F}_Q$. 2. P1 replies with $w_0 = a \cdot (s * z) + c_0$ and $w_1 = a \cdot (s * \overline{z}) + c_1$. 3. V2 sends P2 $chall \in \{0, 1\}$. 4. If $chall = 0$, then P2 sends V2 z, c_0, c_1. If $chall = 1$, then P2 will send to V2 the binary vector $x = v \oplus z$ and the value $c' = \sum\limits_{i=1}^{n} (c_{x_i})_i$. 5. After the round, if $chall = 0$, then the verifiers confirm that $w_0 = a \cdot (s * z) + c_0$ and $w_1 = a \cdot (s * \overline{z}) + c_1$. If instead $chall = 1$, then the verifiers check that $\sum\limits_{i=1}^{n} (w_{x_i})_i = ak + c'$. If a check fails, they reject, otherwise they accept.

Table 3.2: ZKP for Subset Sum

3.1.3 Proof of Security

Proposition 3.1.1. *The ZK Subset Sum protocol has perfect completeness.*

Proof. Suppose the provers have a solution $v \in \mathbb{F}_2^n$ and random shared vectors $c_0, c_1 \in \mathbb{F}_Q^n$ and $z \in \mathbb{F}_2^n$. Then it is clear that if steps (2) and (4) are followed properly by the provers, then the verification in (5) for $chall = 0$ will pass. On the other hand, if $chall = 1$, then the checks will still pass since we have

$$\begin{aligned}
\sum_{i=1}^{n}(w_{x_i})_i &= \sum_{i=1}^{n}\overline{x_i}(w_0)_i + x_i(w_1)_i \\
&= \sum_{i=1}^{n}\overline{x_i}(as_iz_i + (c_0)_i) + x_i(as_i\overline{z_i} + (c_1)_i) \\
&= \sum_{i=1}^{n} as_i(\overline{x_i}z_i + x_i\overline{z_i}) + (\overline{x_i}(c_0)_i + x_i(c_1)_i) \\
&= \sum_{i=1}^{n} as_iv_i + \sum_{i=1}^{n}(\overline{x_i}(c_0)_i + x_i(c_1)_i) \\
&= a\sum_{i=1}^{n} s_iv_i + \sum_{i=1}^{n}(c_{x_i})_i \\
&= ak + c'.
\end{aligned}$$

Therefore all the checks by the verifiers will pass, so the probability that the provers are accepted by the verifiers is 1.

□

Proposition 3.1.2. *The Subset Sum ZKP is sound against malicious quantum provers with soundness exponentially close to $\frac{1}{2}$ in a single round.*

The proof uses a technique similar to the proof of soundness in [CL17]. Here are the steps that will be taken in the formal proof.

- Formalize the game that cheating provers play. This allows G_{coup} to be defined by Definition 2.4.4 where the second prover must answer both challenges at once. Note that G_{coup} is not zero-knowledge.

- Since the subset sum problem instance has no solution, combining the answers for both challenges must not yield a solution. Using this, we construct a strategy for P2 to guess a using basic modular arithmetic, assuming P2 can successfully answer both challenges.
- By no-signalling, the probability of P2 successfully guessing a is not more than $\frac{1}{Q}$. This yields an upper bound on the probability that P2 can answer both challenges, hence upper bounding $\omega^*(G_{coup})$.
- We use Proposition 1 from [CL17] to relate the winning probability of G_{coup} with the winning probability of the ZKP for the Subset Sum Problem for cheating provers.

Proof. We define the game G^{SS} so that it satisfies Definition 2.4.3.

- P1 receives value $a \in \mathbb{F}_Q$, and P2 receives *chall* $\in \{0, 1\}$.
- P1 outputs values $w_0, w_1 \in \mathbb{F}_Q^n$. If *chall* $= 0$, then P2 outputs $z \in \mathbb{F}_2^n$ and $c_0, c_1 \in \mathbb{F}_Q^n$. If *chall* $= 1$, then P2 outputs a value $c' \in \mathbb{F}_Q$ and a vector $x \in \mathbb{F}_2^n$.
- If *chall* $= 0$, then the two players win if $w_0 = a \cdot (s * z) + c_0$ and $w_1 = a \cdot (s * \overline{z}) + c_1$. If *chall* $= 1$, then players win if $\sum\limits_{i=1}^{n} (w_{x_i})_i = ak + c'$.

Recall Definition 2.4.6. The game G^{SS} is 2^n-projective since after z or x is chosen by P2, then the winning values of c_1, c_2 or c' are fixed. This can be seen by rearranging the equations in the last bullet-point. In order to upper bound $\omega^*(G^{SS})$, we consider the game G^{SS}_{coup}, given in Definition 2.4.4. Our goal is to show that if cheating provers can win the game G^{SS}_{coup}, then P2 has a strategy to perfectly guess a, which should only happen with negligible probability. Fix an input/output pair $(a, (w_0, w_1))$ for P1, and consider the outputs for P2 for both inputs. For *chall* $= 0$, we have

$$w_0 = a \cdot s * z + c_0 \text{ and } w_1 = a \cdot s * \overline{z} + c_1$$

$$\implies w_b = a(\overline{b} \cdot s * z + b \cdot s * \overline{z}) + c_b.$$

For *chall* $= 1$, we have

$$\sum_{i=1}^{n} (w_{x_i})_i = ak + c'.$$

Substituting the left hand side, then rearranging, we get

$$\begin{aligned} ak + c' = \sum_{i=1}^{n} (w_{x_i})_i &= \sum_{i=1}^{n} a \cdot (\overline{x_i} s_i z_i + x_i s_i \overline{z_i}) + (c_{x_i})_i \\ \implies a \left(\sum_{i=1}^{n} (\overline{x_i} s_i z_i + x_i s_i \overline{z_i}) - k \right) &= c' - \sum_{i=1}^{n} (c_{x_i})_i. \end{aligned}$$

CLAIM: $\sum\limits_{i=1}^{n} (\overline{x_i} s_i z_i + x_i s_i \overline{z_i}) \neq k.$

Assume by contradiction that we have equality. Then we will construct a solution to the subset sum problem, however since we are proving soundness in the case of a dishonest prover, we have implicitly assumed this to not be possible. The candidate solution is $v' := x \oplus z$. Indeed, we have

$$\sum_{i=1}^{n} v'_i s_i = \sum_{i=1}^{n} (x \oplus z)_i s_i = \sum_{i=1}^{n} (\overline{x_i} z_i + x_i \overline{z_i}) s_i = \sum_{i=1}^{n} \overline{x_i} z_i s_i + x_i \overline{z_i} s_i = k.$$

Therefore we have constructed a solution v', hence k cannot be equal to the sum.

By the result of the claim, we can divide and obtain

$$\implies a = \left(c' - \sum_{i=1}^{n} (c_{x_i})_i \right) \left(\sum_{i=1}^{n} (\overline{x_i} s_i z_i + x_i s_i \overline{z_i}) - k \right)^{-1}.$$

However, since the value on the right hand side is completely determined by the output of P2 and the value on the lefthand side is chosen uniformly at random for P1, this can happen with probability at most $\frac{1}{Q}$ by no-signalling. This means that $\omega^*(G^{SS}_{coup}) \leq \frac{1}{Q}$. Now, we can apply Proposition 1 from [CL17] to obtain

$$\omega^*(G^{SS}) \leq \frac{1}{2} + \left(\frac{64 \cdot 2^n}{Q}\right)^{1/3}.$$

If we take $Q \geq 64 \cdot 2^{n+3K}$, then the protocol has soundness $\frac{1}{2} + 2^{-K}$. Lastly, we verify that there is only a polynomial amount of communication needed for the protocol. In (1), V1 sends $\log(Q)$ bits, then in (2), P1 sends $2n\log(Q)$ bits, then in (3), only 1 bit is sent, and finally in (4), $n + 2n\log(Q)$ bits are sent if $chall = 0$, and if $chall = 1$, then $n + \log(Q)$ bits are sent. Therefore each step requires only a polynomial number of bits in n and k since $\log(Q) \in O(n + k)$. □

Proposition 3.1.3. *The ZK Subset Sum protocol has perfect zero knowledge against malicious quantum verifiers*

Proof. We'll show that this protocol is zero knowledge in the model of 2-provers and a single quantum verifier. We will give the verifier the freedom to query the provers in any order, without needing to respect relativistic timing restraints. Proving the ZK property in this model will be even stronger since the model gives as much power to the malicious verifier as possible. We model a cheating verifier V^* as two families of circuits (V_1^*, V_2^*), where V_i^* takes as input a sub-view and outputs the message to prover i for an instance of size n. Since the verifier can ask the questions in either order, we have two cases:

- Case 1: V_2^* depends on the interaction with P1.
- Case 2: V_1^* depends on the interaction with P2.

Since the cases are treated very similarly, we will just present the proof of Case 1. In other words, we'll consider a verifier that queries P1 and waits for a response before querying

P2. Our first step is to describe the view when V^* interacts with two honest provers. Then, we will define a simulator that can create the same view using only query access to V^* despite not having any access to provers. Since these two views will be the same, then the protocol will be zero-knowledge.

The view will consist of classical registers Q_1 and Q_2 that will hold the questions that will be asked to P1 and P2. Also, the responses of the two provers will be stored in classical registers R_1 and R_2. In addition, the verifier will hold a private quantum register V. In addition, we will adopt the notation $D(\psi) := |\psi\rangle\langle\psi|$ for quantum states to avoid the need to write ψ twice.

Case 1 with Honest Prover

We assume that the operation of V_2^* will depend on the interaction with P1. At the beginning of the protocol, the verifier's view is an auxiliary state $\sigma_0 := \rho_V$.

Next, after the verifier's first message, the view is

$$\sigma_1 := V_1^*(\rho_V) = \sum_{a \in \mathbb{F}_Q} p_a D(a)_{Q_1} \otimes \rho(a)_V.$$

Here, p_a is the probability of a being the query for P1, and $\rho(a)_V$ is the verifier's private quantum state after sending a to P1. Following the response of P1, the verifier's view is

$$\sigma_2 := \frac{1}{Q^{2n}} \frac{1}{2^n} \sum_{c_0, c_1 \in \mathbb{F}_Q^n} \sum_{z \in \mathbb{F}_2^n} \sum_{a \in \mathbb{F}_Q} p_a D(Y(z, c_0, c_1))_{R_1} \otimes D(a)_{Q_1} \otimes \rho(a)_V.$$

where $Y(z, c_0, c_1) := (a \cdot s * z + c_0, a \cdot s * \overline{z} + c_1)$. Next, the verifier sends the challenge, which can be influenced by all that has happened up to this point. In other words, the verifier applies the circuit V_2^* to the view σ_2. The view becomes

$$\sigma_3 := \frac{1}{Q^{2n}}\frac{1}{2^n}\sum_{c_0,c_1\in\mathbb{F}_Q^n}\sum_{z\in\mathbb{F}_2^n}\sum_{a\in\mathbb{F}_Q}\sum_{chall\in\{0,1\}} p_{a,chall}D(Y(z,c_0,c_1))_{R_1}$$

$$\otimes D(chall)_{Q_2}\otimes D(a)_{Q_1}\otimes\rho(a,chall,Y(z,c_0,c_1))_V.$$

After the final message, we have two cases. If the challenge is 0, then P2 sends z, c_0, c_1 in the second response register. On the other hand, if the challenge is 1, then that register should instead contain a value c' and a binary vector x. In particular, an honest prover will choose $x = v \oplus z$ and $c' = \sum_{i=1}^{n}(c_{x_i})_i$, yielding

$$\sigma_4 := \frac{1}{Q^{2n}}\frac{1}{2^n}\sum_{c_0,c_1\in\mathbb{F}_Q^n}\sum_{z\in\mathbb{F}_2^n}\sum_{a\in\mathbb{F}_Q} D(Y(z,c_0,c_1))_{R_1}\otimes D(a)_{Q_1}\otimes$$

$$\Big(p_{a,0}D(0)_{Q_2}\otimes D(z,c_0,c_1)_{R_2}\otimes\rho(a,0,Y(z,c_0,c_1))_V$$

$$+ p_{a,1}D(1)_{Q_2}\otimes D(v\oplus z,c')_{R_2}\otimes\rho(a,1,Y(z,c_0,c_1))_V\Big).$$

This is the final view for an honest prover. Now, before moving onto the simulator, we will rewrite this final state so that it will resemble the simulator's final state later on. The first step is to let $(w_0,w_1) := Y(z,c_0,c_1)$ and sum over w_0, w_1 instead of c_0, c_1. This means that $c_0 = w_0 - a\cdot s * z$ and $c_1 = w_1 - a\cdot s*\overline{z}$. So far, this gives us

$$\sigma_4 = \frac{1}{Q^{2n}}\frac{1}{2^n}\sum_{w_0,w_1\in\mathbb{F}_Q^n}\sum_{z\in\mathbb{F}_2^n}\sum_{a\in\mathbb{F}_Q} D(w_0,w_1)_{R_1}\otimes D(a)_{Q_1}\otimes$$

$$\Big(p_{a,0}D(0)_{Q_2}\otimes D(z,w_0-a\cdot s*z,w_1-a\cdot s*\overline{z})_{R_2}\otimes\rho(a,0,w_0,w_1)_V$$

$$+ p_{a,1}D(1)_{Q_2}\otimes D(v\oplus z,c')_{R_2}\otimes\rho(a,1,w_0,w_1)_V\Big).$$

The next step is to move the sum over z past the terms on which it does not act. We obtain

$$\begin{aligned}\sigma_4 = \frac{1}{Q^{2n}} \sum_{w_0,w_1\in\mathbb{F}_Q^n} \sum_{a\in\mathbb{F}_Q} D(w_0,w_1)_{R_1} \otimes D(a)_{Q_1} \otimes \Big(& p_{a,0}|0\rangle\langle 0|_{Q_2}\otimes \\ & \frac{1}{2^n}\sum_{z\in\mathbb{F}_2^n} D(z, w_0 - a\cdot s * z, w_1 - a\cdot s * \overline{z})_{R_2} \otimes \rho(a,0,w_0,w_1)_V \\ & + p_{a,1}D(1)_{Q_2} \otimes \frac{1}{2^n}\sum_{z\in\mathbb{F}_2^n} D(v\oplus z, c')_{R_2} \otimes \rho(a,1,w_0,w_1)_V\Big).\end{aligned}$$

The final step is to rename $x := v \oplus z$ in the third line and sum over x instead. We can also use the relation from Proposition 3.1.1 to obtain $c' = \sum_{i=1}^{n}(w_{x_i})_i - ka$. Putting this together, we get

$$\begin{aligned}\sigma_4 = \frac{1}{Q^{2n}} \sum_{w_0,w_1\in\mathbb{F}_Q^n} \sum_{a\in\mathbb{F}_Q} D(w_0,w_1)_{R_1} \otimes D(a)_{Q_1} \otimes \Big(& p_{a,0}D(0)_{Q_2}\otimes \\ & \frac{1}{2^n}\sum_{z\in\mathbb{F}_2^n} D(z, w_0 - a\cdot s * z, w_1 - a\cdot s * \overline{z})_{R_2} \otimes \rho(a,0,w_0,w_1)_V \\ & + p_{a,1}D(1)_{Q_2} \otimes \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} D(x, \sum_{i=1}^{n}(w_{x_i})_i - ka)_{R_2} \otimes \rho(a,1,w_0,w_1)_V\Big).\end{aligned}$$

Case 1 with Simulator

Now, we describe how to simulate the views of the verifier without the help of any provers. We will denote the ith simulated view as σ_i'. Since the simulator has access to V_1^* and ρ_V, then σ_0 and σ_1 are straightforward to simulate. Intuitively, no effort is required at this stage because the prover has not acted yet. For σ_2, the response from P1 is two uniformly random vectors w_0, w_1 since c_0 and c_1 act as one-time pads. Then

$$\sigma_2' = \frac{1}{Q^{2n}} \sum_{w_0,w_1\in\mathbb{F}_Q^n} \sum_{a\in\mathbb{F}_Q} p_a D(w_0,w_1)_{R_1} \otimes D(a)_{Q_1} \otimes \rho(a)_V.$$

This can be created from σ_1' by tensoring with the maximally mixed state in register R_1. Next, the simulator applies V_2^* to σ_2' to get σ_3':

$$\sigma_3' = \frac{1}{Q^{2n}} \sum_{w_0, w_1 \in \mathbb{F}_Q^n} \sum_{a \in \mathbb{F}_Q} \sum_{chall \in \{0,1\}} p_{a,chall} D(w_0, w_1)_{R_1} \otimes D(chall)_{Q_2} \otimes D(a)_{Q_1} \otimes \rho(a, chall, w_0, w_1)_V.$$

The final step is to simulate σ_4. Again, this depends on the challenge bit *chall*.

- For $chall = 0$ in register Q_2, the simulator chooses a random $z \in \mathbb{F}_2^n$ and then puts $D(z, w_0 - a \cdot s * z, w_1 - a \cdot s * \bar{z})$ in register R_2.
- For $chall = 1$ in register Q_2, the simulator chooses a random $x \in \mathbb{F}_2^n$ and then places $D(x, \sum_{i=1}^{n} (w_{x_i})_i - ka)$ in register R_2.

This gives a final view of

$$\sigma_4' = \frac{1}{Q^{2n}} \sum_{w_0, w_1 \in \mathbb{F}_Q^n} \sum_{a \in \mathbb{F}_Q^n} D(w_0, w_1)_{R_1} \otimes D(a)_{Q_1} \otimes \Big(p_{a,0} D(0)_{Q_2} \otimes \frac{1}{2^n} \sum_{z \in \mathbb{F}_2^n} D(z, w_0 - a \cdot s * z, w_1 - a \cdot s * \bar{z})_{R_2} \otimes \rho(a, 0, w_0, w_1)_V + p_{a,1} D(1)_{Q_2} \otimes \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} D(x, \sum_{i=1}^{n} (w_{x_i})_i - ka)_{R_2} \otimes \rho(a, 1, w_0, w_1)_V \Big).$$

Now, it is clear that the two views σ_4 and σ_4' are equal, hence our simulator succeeded.

Case 2 The proof of Case 2 is very similar and is left out. □

3.1.4 Analysis of Efficiency

Now we analyze the efficiency of the Subset Sum ZKP. The best known quantum algorithm for the Subset Sum Problem has time complexity $O(2^{n/3})$ [AHJ+22]. In order to take roughly time 2^{100} to solve, the instance must therefore be of size $n \geq 300$. The prime Q is chosen such that $Q > 64 \cdot 2^{n+3K}$ with $K = 5$ so that soundness is $\frac{1}{2} + 2^{-K} \approx 0.53$. Then the expected number of bits sent in each round is

$$\log_2(Q) + 1 + 2n\log_2(Q) + (n + \frac{2n\log_2(Q) + \log_2(Q)}{2}) \approx 290,000$$

where the terms correspond to the number of bits sent by V1, V2, P1, P2, in that order. This means that each round requires roughly 36KB of communication. This is far less than the 1.89MB required by [CL17], and comparable to the 17KB required by the protocol of [CB21]. Note however that the Subset Sum ZKP has a lower soundness error than [CB21] (roughly $\frac{1}{2}$ compared to $\frac{2}{3}$), meaning that less rounds are required. To reduce the total soundness error to 2^{-100}, our Subset Sum ZKP would require only 110 rounds ($0.53^{110} \approx 2^{-100}$), whereas [CB21] would require 170 rounds ($0.67^{170} \approx 2^{-100}$). Putting this together, the total information sent in our protocol is roughly 3.96MB (36KB per round with 110 rounds), compared to 2.89MB of total communication for [CB21] (17KB per round with 170 rounds). This means that our ZKP is very comparable to [CB21] in terms of total information sent.

One important consideration is the complexity of computation for the provers. We will first motivate this by explaining the experimental setup, which is depicted in Figure 3.3.

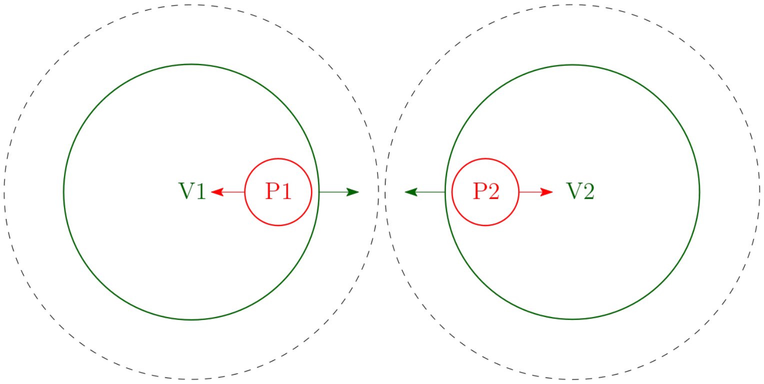

Figure 3.3: Message propagation in experimental setup

Before the rounds, the verifiers position themselves a known distance apart. Then in every round, they each send their message to their respective prover. The provers each receive a message, perform some computation, then reply back to their respective verifier. In order to ensure that the provers are separated, the verifiers time the responses to ensure that the provers reply within fixed time limits. These time limits depend on the distance between the verifiers, and ensure that the provers are inside the dotted circles, hence separated. Importantly, the response of P2 is not influenced by the question of V1 and similarly, the response of P1 is not influenced by the question of V2. For example, after V1 sends the message a intended for P1, then if P2 receives a as well, then P2 will still not have enough time to reply to V2 with a message influenced by a. This is crucial to fulfill the assumption that each prover only has access to the question intended for them. For more details on the experimental setup, see [ABC$^+$21] or [CB21].

In theory, if all the messages travel towards the other parties at the speed of light, and each party can instantly reply when they receive a message, then as long as the provers stay within the dotted circles, then the protocol should run smoothly. However in practice, the messages may not follow such a direct path to their intended recipients, and the provers must take some time to perform internal computations once they have received their message before they can reply. Effectively, this means that the provers cannot be close to the edges of the dotted circles since it will not give them enough time to respond, and slower protocols require the verifiers to further increase their separation. Experimentally, large distances have been required by the verifiers in order to give the provers enough time to respond. For example, the small-scale setup in [CB21] required the verifiers to position themselves 400km apart, whereas in [ABC$^+$21], the verifiers were only 60m apart. These differences are a result of the amount of communication required in each round and the time complexity of the operations required by the provers. For this reason, we also seek to speed up the computations required by the honest provers in order to allow for more practical, small-scale uses of ZKPs.

With the experimental setup explained, we can now compare multiplication complexities. While only three multiplications must be performed by the prover in [CB21], their field $\mathbb{F}_Q$ is far larger than ours, chosen to have $2^{23209} - 1$ elements. To their benefit, they only need to perform three multiplications per round, whereas in our protocol, $2n$ multiplications must be performed, albeit in a much smaller field of size roughly 2^{321}. As an added bonus, our multiplications are all by the same constant a, hence parallelism is very straightforward.

Multiplication can have varying time complexities depending on the implementation, and some algorithms contain large hidden constants. To simplify the comparison, tests were run using Python with and without parallelism (array multiplication in NumPy). In our brief experiments, our multiplications were roughly 3 times faster without parallelism and roughly 7 times faster with parallelism when compared to [CB21]. We leave a more thorough analysis for future work.

In short, faster multiplication can allow the verifiers in the protocol to be at closer distances, making it more practical, and allowing the protocol to run faster. In Table 3.3 we present a summary of the results by comparing our Subset Sum ZKP with other known quantum-secure ZKPs in the literature.

Protocol	#Bytes/Round	#Rounds	Multiplication complexity
[ABC$^+$21]	2B	10^{19}	negligible
[CL17]	1.89MB	177	40 to 300× slower
[CB21]	17.03KB	170	3 to 7× slower
Subset Sum ZKP	36KB	110	Baseline

Table 3.3: Comparison of efficiency of known ZKPs

Note that although the protocol of [ABC$^+$21] has very efficient rounds, three provers are required to achieve quantum security, and the number of rounds is impractical. Note that the numbers presented here differ slightly from in [CL17][1] and in [CB21][2].

[1] [CL17] requires 177 rounds because in [CB21], Q is chosen to be $10000n! = 64n!2^{3k}$ so $k \approx 2.4$, hence soundness is $\frac{1}{2} + 2^{-2.4} \approx 0.69$

[2] In [CB21], 340 rounds were chosen for loss tolerance

One last point is that generating positive instances of the Subset Sum Problem is straightforward by sampling n random numbers between 1 and Q/n, then determining the target k by randomly choosing a subset of the n numbers.

3.2 A ZKP for 3-SAT

3.2.1 General Idea

3-SAT problem: Given a Boolean expression ϕ of the variables $x_1, ..., x_n$ in conjunctive normal form with m clauses of size 3 (3-CNF), determine whether there is an assignment for the variables that satisfies ϕ. For example, the variables may be $x_1, ..., x_5$, and ϕ may be

$$\phi' \equiv (x_3 \vee \neg x_2 \vee x_5) \wedge (\neg x_1 \vee \neg x_4 \vee \neg x_5) \wedge (x_1 \vee \neg x_2 \vee x_5) \wedge (x_1 \vee x_4 \vee x_2).$$

Then ϕ' has a satisfying assignment of $x = (1, 0, 1, 0, 0)$. This problem is known to be NP-complete [AHU74].

Note that a witness to the a 3SAT instance is typically viewed as an assignment x of the variables, but instead for this section, we will construct a witness differently. An alternate way to present a solution to the 3SAT problem is to provide a vector $e \in \{1, 2, 3\}^m$ that indicates the position of a 1 in each clause of ϕ. This way, one could perform a linear scan through ϕ and determine the variable assignments. The only required check would be that the same variable does not obtain two different assignments based on two different clauses. For example, an $e \in \{1, 2, 3\}^4$ for ϕ' could be $e' := (1, 2, 1, 1)$, though it is not unique.

Many ZKPs require a randomized step to ensure that the verifier does not learn anything about the witness when the prover unveils an answer. Here, the randomization will be independent cyclic permutations of the variables in each clause.

Definition 3.2.1. Let ϕ be a CNF with m clauses, each of size 3. We define a cyclic CNF permutation Π to be a collection of m cyclic permutations on $\{1, 2, 3\}$. By $\Pi(\phi)$, we denote

ϕ after each clause has been permuted by the corresponding permutation of Π. We define CP to be the set of all cyclic CNF permutations on ϕ.

For example by choosing a random $\Pi \in CP$, with ϕ' as above, then $\Pi(\phi')$ could be the following:

$$\Pi(\phi') \equiv (x_5 \vee x_3 \vee \neg x_2) \wedge (\neg x_4 \vee \neg x_5 \vee \neg x_1) \wedge (x_5 \vee x_1 \vee \neg x_2) \wedge (x_1 \vee x_4 \vee x_2).$$

Note that simply re-arranging the contents of the CNF does not change the satisfiability of the formula since $\vee$ is commutative. In fact, the values of $e \in \{1,2,3\}^m$ must get adjusted according to Π. We write $\Pi(e)$ to denote the witness in $\{1,2,3\}^m$ for $\Pi(\phi)$ where each coordinate of e has undergone the corresponding permutation in Π. For example, $\Pi(e') = (2,1,2,1)$. The idea with the protocol below is that the provers will commit to the satisfying assignment, as well as the formula $\Pi(\phi)$ with the satisfying assignment substituted in for the variables. For example, with the running example of ϕ', the provers would make the following commitment:

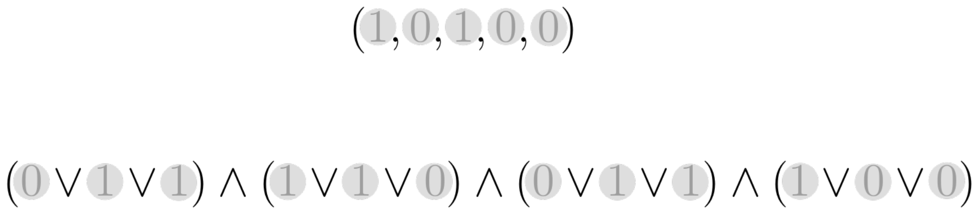

Figure 3.4: Commitment by provers

Next, the verifiers can either challenge the provers to unveil a 1 in each clause, or to prove that the variable assignments were consistent. If the challenge was the former, then the verifiers would see the following:

Note that the unveiled 1's are in the positions specified by $\Pi(e')$. On the other hand, if the verifiers asked to see that the variable assignments were consistent, then the provers would go through each variable of $\Pi(\phi)$ and prove that it's value is consistent with the satisfying assignment. The diagram below depicts this step for only the third clause to avoid clutter.

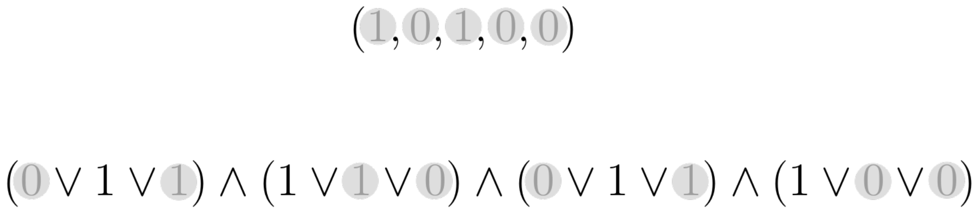

Figure 3.5: Provers unveil a 1 in each clause

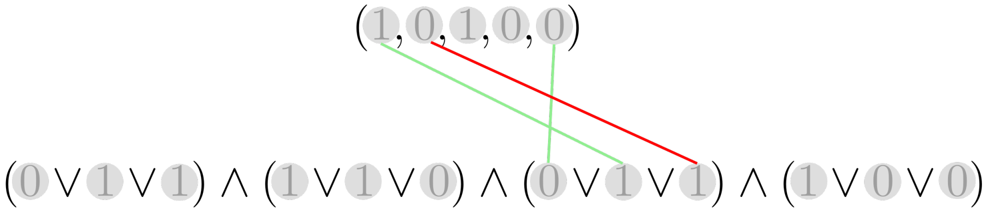

Figure 3.6: Provers show consistent variable assignments

Note that the provers will use their ability to unveil the difference of commitments for the green lines, and use their ability to unveil the sum of commitments for the red lines. In particular, the difference of two equal bits is 0, and the sum of two different bits is 1.

3.2.2 Protocol

First, we introduce notation that will be used throughout the protocol. The operations will be mainly addition and multiplication over the field $\mathbb{F}_Q$, however, we will also multiply elements of $\mathbb{F}_Q$ by bits, hence Q may be any large prime power since every field has elements 0,1.

Next, we introduce a ZKP between two provers and two verifiers. Assuming the two provers know a witness $s' \in \{0,1\}^n$, they will be able to convince two verifiers that there exists a solution to the given 3SAT problem instance. First, they compute an $e \in \{1,2,3\}^m$ from s' that indicates the position of a 1 in each clause. Then, before each round, the provers share random $\Pi \in CP$, $c \in \mathbb{F}_Q^{3m}$, $c' \in \mathbb{F}_Q^n$.

Notation	Purpose	Data-Type
Π	cyclic 3-CNF permutation	CP
ϕ	3-CNF formula	CP
s'	satisfying assignment of ϕ	$\mathbb{F}_2^n$
p	bits of permuted formula $\Pi(\phi)$	$\mathbb{F}_2^{3m}$
c	key for p	$\mathbb{F}_Q^{3m}$
c'	key for s'	$\mathbb{F}_Q^n$
w	encryption of p	$\mathbb{F}_Q^{3m}$
w'	encryption of s'	$\mathbb{F}_Q^n$
e	indicates a 1 in each clause of ϕ	$\{1,2,3\}^m$
chall	random challenge bit chosen by verifier	$\{0,1\}$

Table 3.4: Notation used in ZKP for 3SAT

Two-Prover, Two-verifier 3-SAT ZKP

1. V1 sends P1 a random value $a \in \mathbb{F}_Q$.

2. P1 replies with $w' = a \cdot s' + c'$ and $w = a \cdot p + c$.

3. V2 sends P2 $chall \in \{0,1\}$.

4. If $chall = 0$, P2 first sends Π to V2. Then for each $i = 1, ..., 3m$: let x_j be at position i of $\Pi(\phi)$. P2 sends to V2:

- $c_i + c'_j$ if variable x_j is negated at position i.
- $c_i - c'_j$ if variable x_j is not negated at position i.

Instead if $chall = 1$, then for each $i = 1, ..., m$, P2 sends $(\Pi(e)_i, c_{3(i-1)+\Pi(e)_i})$ to V2, unveiling a 1 in each clause.

5. After the round, if $chall = 0$, then the verifiers confirm that for each $i = 1, ..., 3m$, if variable x_j is at position i of $\Pi(\phi)$, then

- $w_i + w'_j = a + (c_i + c'_j)$ if variable x_j is negated at position i.
- $w_i - w'_j = c_i - c'_j$ if variable x_j is not negated at position i.

If instead $chall = 1$, then the verifiers check that for each $i = 1, ..., m$, it is the case that $w_{3(i-1)+\Pi(e)_i} = a + c_{3(i-1)+\Pi(e)_i}$.

Table 3.5: ZKP for 3SAT

3.2.3 Proof of Security

Proposition 3.2.2. *The 3-SAT ZKP has perfect completeness.*

Proof. Assume that the two honest provers have a witness s'. Then if they follow the steps in the protocol with two honest verifiers, then when $chall = 0$, the checks that the verifiers make will pass. Indeed, if $i \in \{1, ..., 3m\}$ and variable x_j is negated at position i in $\Pi(\phi)$, then $s'_j + p_i = 1$, yielding

$$w_i + w'_j = (ap_i + c_i) + (as'_j + c'_j) = a(p_i + s'_j) + (c_i + c'_j) = a + (c_i + c'_j).$$

If instead variable x_j is not negated at position i in $\Pi(\phi)$, then $s'_j = p_i$, meaning that

$$w_i - w'_j = (ap_i + c_i) - (as'_j + c'_j) = a(p_i - s'_j) + (c_i - c'_j) = c_i - c'_j.$$

Next we confirm that the checks will pass when $chall = 1$. The provers are honest, meaning each clause contains a 1, therefore $p_{3(i-1)+\Pi(e)_i} = 1 \ \forall i \in \{1, ..., m\}$, hence

$$w_{3(i-1)+\Pi(e)_i} = ap_{3(i-1)+\Pi(e)_i} + c_{3(i-1)+\Pi(e)_i} = a + c_{3(i-1)+\Pi(e)_i}.$$

Therefore all the checks by the verifiers will pass for both values of *chall*.

□

Proposition 3.2.3. *The 3-SAT ZKP is sound against malicious quantum provers with soundness exponentially close to $\frac{1}{2}$ in a single round.*

Proof. We define the game G^{3SAT}.

- P1 receives value $a \in \mathbb{F}_Q$, and P2 receives $chall \in \{0, 1\}$.
- P1 outputs values $w \in \mathbb{F}_Q^{3m}, w' \in \mathbb{F}_Q^n$. If $chall = 0$, then P2 outputs a $\Pi \in CP$ and a value $\delta \in \mathbb{F}_Q^{3m}$. If $chall = 1$, then P2 outputs $f \in \{1, 2, 3\}^m$, $\gamma \in \mathbb{F}_Q^m$.
- After the round, if $chall = 0$, then the provers win if for each $i = 1, ..., 3m$, it holds that

- $w_i + w'_j = a + \delta_i$ where variable x_j appears negated at position i of $\Pi(\phi)$,
- $w_i - w'_j = \delta_i$ where variable x_j appears not negated at position i of $\Pi(\phi)$.

If instead *chall* = 1, then the provers win if for each $i = 1, ..., m$, it is the case that $w_{3(i-1)+f_i} = a + \gamma_i$.

Recall Definition 2.4.6. The game G^{3SAT} is 3^m-projective where since after P1 has output values w, w', then P2 has 3^m choices for Π and 3^m choices for f, and in both cases the second value (δ or c) will be uniquely determined. This can be seen by rearranging the equations in the last bullet-point. In order to upper bound $\omega^*(G^{3SAT})$, we consider the game G^{3SAT}_{coup}, given in Definition 2.4.4. From a winning strategy for G^{3SAT}_{coup}, we'll devise a strategy for P2 to guess a based solely on P2's local input *chall* and output.

Fix an input/output pair $(a, (w_0, w_1))$ for P1, and suppose P2 successfully answers both challenges. Note that constructing a satisfying assignment is implicitly assumed to be impossible since soundness is currently being considered. Then there must be a variable x_j and two conflicting clauses $i, i' \in \{1, ..., m\}$ such that x_j is negated at position $3(i-1) + f_i$ of $\Pi(\phi)$ but not negated at position $3(i'-1) + f_{i'}$ of $\Pi(\phi)$. Otherwise, a solution could be formed as discussed in the introduction, and there would not be any conflicts between any variable assignments. This situation is illustrated below.

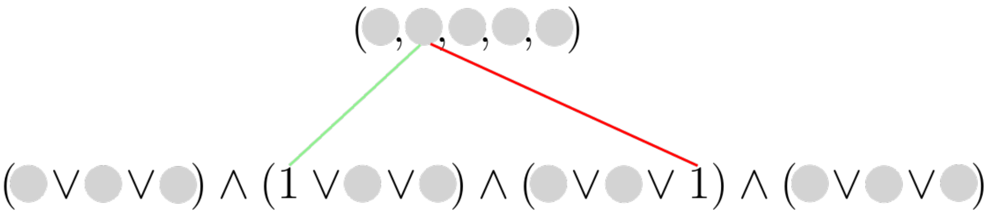

Figure 3.7: Conflicting variable assignments. The green line represents that the difference of the two bits was unveiled to be 0, and the red line means that the sum was unveiled to be 1

Then the conflict between clauses $i, i' \in \{1, ..., m\}$ for variable x_j leads to the following.

$$w_{3(i-1)+f_i} + w'_j = a + \delta_{3(i-1)+f_i} \quad \text{and} \quad w_{3(i'-1)+f_{i'}} - w'_j = \delta_{3(i'-1)+f_{i'}}$$

Summing the contents of the two equations cancels out w'_j.

$$w_{3(i-1)+f_i} + w_{3(i'-1)+f_{i'}} = a + \delta_{3(i-1)+f_i} + \delta_{3(i'-1)+f_{i'}}$$

Next, we use the equality from satisfying *chall* $= 1$, and then finally isolate for a.

$$\begin{aligned} (a+\gamma_i) + (a+\gamma_{i'}) &= a + \delta_{3(i-1)+f_i} + \delta_{3(i'-1)+f_{i'}} \\ \implies a &= \delta_{3(i-1)+f_i} + \delta_{3(i'-1)+f_{i'}} - \gamma_i - \gamma_{i'} \end{aligned}$$

Note that the values on the right hand side are all decided by P2, hence P2 has a strategy to guess a. However, $P2$ can achieve this correlation with probability at most $\frac{1}{Q}$ by no-signalling, thus $\omega^*(G^{3SAT}_{coup}) \leq \frac{1}{Q}$. Now, we can apply Proposition 1 from [CL17] to obtain

$$\omega^*(G^{3SAT}) \leq \frac{1}{2} + \left(\frac{64 \cdot 3^m}{Q}\right)^{1/3}.$$

If we take $Q \geq 64 \cdot 2^{\log(3)m+3k}$, then the protocol has soundness $\frac{1}{2} + 2^{-k}$. Next, there is a polynomial amount of communication in total since each step requires only a polynomial number of bits in n and m since $\log(Q) \in O(m)$. Indeed, in (1), V1 sends $\log(Q)$ bits, then in (2), P1 sends $(n+3m)\log(Q)$ bits, then in (3), only 1 bit is sent. Finally in (4), if *chall* $= 0$ then sending Π takes at most $2m$ bits and sending the rest is $3mQ$ bits. On the other hand, if *chall* $= 1$, then at most $m(Q+2)$ bits are sent.

□

Proposition 3.2.4. *The 3SAT ZKP has perfect zero-knowledge against malicious quantum verifiers.*

Proof. We'll again use the model of 2-provers and a single quantum verifier, giving the verifier the freedom to query the provers in any order, without needing to respect relativistic timing restraints. As before, we model a cheating verifier V^* as two families of circuits (V_1^*, V_2^*), where V_i^* takes as input a sub-view and outputs the message to prover i for an instance of size n. Since the verifier can ask the questions in either order, we have two cases:

- Case 1: V_2^* depends on the interaction with P1.
- Case 2: V_1^* depends on the interaction with P2.

Since the cases are treated very similarly, we will just present the proof of Case 1. In other words, we'll consider a verifier that queries P1 and waits for a response before querying P2. Our first step is to describe the view when V^* interacts with two honest provers. Then, we will define a simulator that can create the same view using only query access to V^* despite not having any access to provers. Since these two views will be the same, then the protocol will be zero-knowledge.

The view will consist of classical registers Q_1 and Q_2 that will hold the questions that will be asked to P1 and P2. Also, the responses of the two provers will be stored in classical registers R_1 and R_2. In addition, the verifier will hold a private quantum register V. In addition, we will adopt the notation $D(\psi) := |\psi\rangle\langle\psi|$ for quantum states to avoid the need to write ψ twice.

Case 1 with Honest Prover

We assume that the operation of V_2^* will depend on the interaction with P1. At the beginning of the protocol, the verifier's view is an auxiliary state $\sigma_0 := \rho_V$.

Next, after the verifier's first message, the view is

$$\sigma_1 := V_1^*(\rho_V) = \sum_{a \in \mathbb{F}_Q} p_a D(a)_{Q_1} \otimes \rho(a)_V.$$

Here, p_a is the probability of a being the query for P1, and $\rho(a)_V$ is the verifier's private quantum state after sending a to P1. Following the response of P1, the verifier's view is

$$\sigma_2 := \frac{1}{Q^{n+3m}}\frac{1}{3^m}\sum_{c'\in\mathbb{F}_Q^n, c\in\mathbb{F}_Q^{3m}}\sum_{\Pi\in CP}\sum_{a\in\mathbb{F}_Q} p_a D(a\cdot s'+c', a\cdot p+c)_{R_1}\otimes D(a)_{Q_1}\otimes\rho(a)_V.$$

Note that p is uniquely determined by ϕ, Π, and s', so it is not part of the sum. Also, $|CP| = 3^m$. Next, the verifier sends the challenge, which can be influenced by all that has happened up to this point. In other words, the verifier applies the circuit V_2^* to the view σ_2. The view becomes

$$\sigma_3 := \frac{1}{Q^{n+3m}}\frac{1}{3^m}\sum_{c'\in\mathbb{F}_Q^n, c\in\mathbb{F}_Q^{3m}}\sum_{\Pi\in CP}\sum_{a\in\mathbb{F}_Q}\sum_{chall\in\{0,1\}} p_{a,chall} D(a\cdot s'+c', a\cdot p+c)_{R_1}$$
$$\otimes D(chall)_{Q_2}\otimes D(a)_{Q_1}\otimes\rho(a, chall, a\cdot s'+c', a\cdot p+c)_V.$$

After the final message, we have two cases. If the challenge is 0, then P2 sends the CNF permutation Π and a vector $\delta\in\mathbb{F}_Q^{3m}$ in the second response register. On the other hand, if the challenge is 1, then that register should instead contain vectors $\Pi(e)\in\{1,2,3\}^m$,$\gamma\in\mathbb{F}_Q^m$. In particular, an honest prover will choose $\delta_i = c_i \pm c'_j$ as described in step 4 of the protocol, and $\gamma_i = c_{3(i-1)+\Pi(e)_i}$ where $\Pi(e)_i$ indicates the position of a 1 in the i-th clause of $\Pi(\phi)$.

$$\sigma_4 := \frac{1}{Q^{n+3m}}\frac{1}{3^m}\sum_{c'\in\mathbb{F}_Q^n, c\in\mathbb{F}_Q^{3m}}\sum_{\Pi\in CP}\sum_{a\in\mathbb{F}_Q} D(a\cdot s'+c', a\cdot p+c)_{R_1}\otimes D(a)_{Q_1}\otimes$$
$$\Big(p_{a,0}D(0)_{Q_2}\otimes D(\Pi,\delta)_{R_2}\otimes\rho(a,0,a\cdot s'+c', a\cdot p+c)_V$$
$$+ p_{a,1}D(1)_{Q_2}\otimes D(\Pi(e),\gamma)_{R_2}\otimes\rho(a,1,a\cdot s'+c', a\cdot p+c)_V\Big)$$

This is the final view when interacting with an honest prover. Now, before moving onto the simulator, we will rewrite this final state so that it will resemble the simulator's final state later on. The first step is to let $w' := a\cdot s'+c'$ and $w := a\cdot p+c$ and sum over

w', w instead of c', c. Rewriting the sum this way may seem problematic since δ and γ were explicitly defined using c and c', however using the relations $c' = w' - a \cdot s'$ and $c = w - a \cdot p$, the definition of γ becomes

$$\gamma_i = c_{3(i-1)+\Pi(e)_i} = w_{3(i-1)+\Pi(e)_i} - a \cdot p_{3(i-1)+\Pi(e)_i} = w_{3(i-1)+\Pi(e)_i} - a.$$

since the honest prover gives $\Pi(e)_i$ such that $p_{3(i-1)+\Pi(e)_i} = 1$. The definition of δ becomes

$$\delta_i = \begin{cases} w_i + w'_j - a & \text{if variable } x_j \text{ is negated at position } i \text{ of } \Pi(\phi). \\ w_i - w'_j & \text{if variable } x_j \text{ is not negated at position } i \text{ of } \Pi(\phi). \end{cases}$$

where x_j is the variable at position i of $\Pi(\phi)$. Now, we can replace the summation over c', c with a summation over w', w in σ_4 to obtain:

$$\begin{aligned}\sigma_4 := \frac{1}{Q^{n+3m}} \frac{1}{3^m} \sum_{w' \in \mathbb{F}_Q^n, w \in \mathbb{F}_Q^{3m}} \sum_{\Pi \in CP} \sum_{a \in \mathbb{F}_Q} & D(w', w)_{R_1} \otimes D(a)_{Q_1} \otimes \\ & \Big(p_{a,0} D(0)_{Q_2} \otimes D(\Pi, \delta)_{R_2} \otimes \rho(a, 0, w', w)_V \\ & + p_{a,1} D(1)_{Q_2} \otimes D(\Pi(e), \gamma)_{R_2} \otimes \rho(a, 1, w', w)_V \Big).\end{aligned}$$

The next step is to move the sum over Π past the terms on which it does not act. We obtain

$$\begin{aligned}\sigma_4 := \frac{1}{Q^{n+3m}} \sum_{w' \in \mathbb{F}_Q^n, w \in \mathbb{F}_Q^{3m}} \sum_{a \in \mathbb{F}_Q} & D(w', w)_{R_1} \otimes D(a)_{Q_1} \otimes \\ & \Big(p_{a,0} D(0)_{Q_2} \otimes \frac{1}{3^m} \sum_{\Pi \in CP} D(\Pi, \delta)_{R_2} \otimes \rho(a, 0, w', w)_V \\ & + p_{a,1} D(1)_{Q_2} \otimes \frac{1}{3^m} \sum_{\Pi \in CP} D(\Pi(e), \gamma)_{R_2} \otimes \rho(a, 1, w', w)_V \Big).\end{aligned}$$

At this point, we can rename $\Pi(e)$ to f and sum over all $f \in \{1,2,3\}^m$ instead. We obtain

$$\sigma_4 := \frac{1}{Q^{n+3m}} \sum_{w' \in \mathbb{F}_Q^n, w \in \mathbb{F}_Q^{3m}} \sum_{a \in \mathbb{F}_Q} D(w', w)_{R_1} \otimes D(a)_{Q_1} \otimes \Big(p_{a,0} D(0)_{Q_2} \otimes \frac{1}{3^m} \sum_{\Pi \in CP} D(\Pi, \delta)_{R_2} \otimes \rho(a, 0, w', w)_V + p_{a,1} D(1)_{Q_2} \otimes \frac{1}{3^m} \sum_{f \in \{1,2,3\}^m} D(f, \gamma)_{R_2} \otimes \rho(a, 1, w', w)_V\Big).$$

Case 1 with Simulator

Now, we describe how to simulate the views of the verifier without the help of any provers. We will denote the ith simulated view as $\tilde{\sigma}_i$. Since the simulator has access to V_1^* and ρ_V, then σ_0 and σ_1 are straightforward to simulate. Intuitively, no effort is required at this stage because the prover has not acted yet. For σ_2, the response from P1 is two uniformly random vectors $w \in \mathbb{F}_Q^{3m}, w' \in \mathbb{F}_Q^n$ since c and c' act as one-time pads. Then

$$\tilde{\sigma_2} = \frac{1}{Q^{n+3m}} \sum_{w \in \mathbb{F}_Q^{3m}, w' \in \mathbb{F}_Q^n} \sum_{a \in \mathbb{F}_Q} p_a D(w', w)_{R_1} \otimes D(a)_{Q_1} \otimes \rho(a)_V.$$

This can be created from $\tilde{\sigma_1}$ by tensoring with the maximally mixed state in register R_1. Next, the simulator applies V_2^* to $\tilde{\sigma_2}$ to get $\tilde{\sigma_3}$:

$$\tilde{\sigma_3} = \frac{1}{Q^{n+3m}} \sum_{w \in \mathbb{F}_Q^{3m}, w' \in \mathbb{F}_Q^n} \sum_{a \in \mathbb{F}_Q} \sum_{chall \in \{0,1\}} p_{a,chall} D(w', w)_{R_1} \otimes D(chall)_{Q_2} \otimes D(a)_{Q_1} \otimes \rho(a, chall, w', w)_V.$$

The final step is to simulate σ_4. Again, this depends on the challenge bit $chall$.

- For *chall* $= 0$ in register Q_2, the simulator chooses a random CNF permutation Π and then puts $D(\Pi, \tilde{\delta})$ in register R_2 where $\tilde{\delta} \in \mathbb{F}_Q^{3m}$ is defined below

$$\tilde{\delta}_i = \begin{cases} w_i + w'_j - a & \text{if variable } x_j \text{ is negated at position } i \text{ of } \Pi(\phi) \\ w_i - w'_j & \text{if variable } x_j \text{ is not negated at position } i \text{ of } \Pi(\phi) \end{cases}$$

  where x_j is the variable at position i of $\Pi(\phi)$.

- For *chall* $= 1$ in register Q_2, the simulator chooses a random $\tilde{f} \in \{1, 2, 3\}^m$ and then places $D(\tilde{f}, \tilde{\gamma})$ in register R_2, where $\tilde{\gamma} \in \mathbb{F}_Q^m$ is defined below:

$$\tilde{\gamma}_i = w_{3(i-1)+\tilde{f}_i} - a.$$

This gives a final view of

$$\begin{aligned}\tilde{\sigma_4} = \frac{1}{Q^{n+3m}} \sum_{w \in \mathbb{F}_Q^{3m}, w' \in \mathbb{F}_Q^n} \sum_{a \in \mathbb{F}_Q} & D(w', w)_{R_1} \otimes D(a)_{Q_1} \otimes \\ & \Big(p_{a,0} D(0)_{Q_2} \otimes \frac{1}{3^m} \sum_{\Pi \in PERM} D(\Pi, \tilde{\delta})_{R_2} \otimes \rho(a, 0, w', w)_V \\ & + p_{a,1} D(1)_{Q_2} \otimes \frac{1}{3^m} \sum_{\tilde{f} \in \{1,2,3\}^m} D(\tilde{f}, \tilde{\gamma})_{R_2} \otimes \rho(a, 1, w', w)_V \Big).\end{aligned}$$

Now, it is clear that the two views σ_4 and $\tilde{\sigma_4}$ are equal since $\tilde{\delta}, \tilde{\gamma}$ are defined identically to δ, γ in the honest case, hence our simulator succeeded.

The proof of case 2 follows the same logic. □

Chapter 4

Conclusion and Future Work

The zero-knowledge protocols presented in this thesis highlight the usefulness of homomorphic bit commitment. Using the relativistic commitment scheme described in Section 2.5 contents of commitments and be easily combined to allow for new protocols. For example, a protocol that allows provers to prove that two commitments are equal is extremely simple with this approach by unveiling the difference of the contents. In addition, the proof technique used for soundness in [CL17] is extremely powerful and can be applied to a wide range of protocols.

Note that in both protocols from Sections 3.1 and 3.2, the commitments that were combined were commitments made only by the provers. It would be interesting to devise a protocol that verifies whether two strings are equal when one string is in the possession of one party and the other string is kept by the other party. Indeed, this could allow for novel zero-trust identification protocols that have no computational assumptions, only relativistic ones. One difficulty of directly applying the existing approach to this problem is that at the end of the protocol, the parties should not necessarily know the difference of their two strings. They should only learn one bit of information from the protocol: whether the two strings are equal or not.

Finally, in [CB21], the authors managed to prove a similar bound as in [CL17], except that it can be applied to protocols with three challenges. If these bounds could be extended to prove soundness for protocols with a polynomial number of challenges, then perhaps a similar proof technique could be applied to prove soundness of the ZKP for QMA presented in [BG22] when using the relativistic commitment scheme seen in Section 2.5. This would allow for a simple two-prover ZKP for QMA using only relativistic assumptions.

Bibliography

[ABC$^+$21] Pouriya Alikhani, Nicolas Brunner, Claude Crépeau, Sébastien Designolle, Raphaël Houlmann, Weixu Shi, Nan Yang, and Hugo Zbinden. Experimental relativistic zero-knowledge proofs. *Nature*, 599(7883):47–50, nov 2021.

[AHJ$^+$22] Jonathan Allcock, Yassine Hamoudi, Antoine Joux, Felix Klingelhöfer, and Miklos Santha. Classical and quantum algorithms for variants of subset-sum via dynamic programming. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[AHU74] Alfred Aho, John Hopcroft, and Jeffrey Ullman. *The design and analysis of computer algorithms*. Addison-Wesley series in computer science and information processing. Addison-Wesley Pub. Co., 1974.

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

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

[BC87] Gilles Brassard and Claude Crepeau. Zero-knowledge simulation of boolean circuits. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 223–233, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[BCMS98] Gilles Brassard, Claude Crépeau, Dominic Mayers, and Louis Salvail. Defeating classical bit commitments with a quantum computer, 1998.

[BG22] Anne Broadbent and Alex Bredariol Grilo. QMA-hardness of consistency of local density matrices with applications to quantum zero-knowledge. *SIAM Journal on Computing*, 51(4):1400–1450, aug 2022.

[Blu81] Manuel Blum. Coin flipping by telephone. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981*, pages 11–15. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No 82-04, 1981.

[Blu87] Manuel Blum. How to prove a theorem so no one else can claim it. In *In: Proceedings of the International Congress of Mathematicians*, pages 1444–1451, 1987.

[BOGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multiprover 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.

[CB21] André Chailloux and Yann Barsamian. Relativistic zero-knowledge protocol for np over the internet unconditionally secure against quantum adversaries, 2021.

[CL17] André Chailloux and Anthony Leverrier. Relativistic (or 2-prover 1-round) zero-knowledge protocol for NP secure against quantum adversaries. In *Lecture Notes in Computer Science*, pages 369–396. Springer International Publishing, 2017.

[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.

[CY17] Claude Crépeau and Nan Yang. Multi-prover interactive proofs: Unsound foundations. In Raphaël C.-W. Phan and Moti Yung, editors, *Paradigms in Cryptology – Mycrypt 2016. Malicious and Exploratory Cryptology*, pages 485–493, Cham, 2017. Springer International Publishing.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, 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. *J. ACM*, 38(3):690–728, jul 1991.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001.

[Ken05] Adrian Kent. Secure classical bit commitment using fixed capacity communication channels. *J. Cryptology*, 18:313–335, 2005.

[KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., USA, 2005.

[LKB$^+$15] Tommaso Lunghi, Jędrzej Kaniewski, Felix Bussières, Raphael Houlmann, Marco Tomamichel, Stephanie Wehner, and Hugo Zbinden. Practical relativistic bit commitment. *Physical Review Letters*, 115(3), jul 2015.

[Sho94] Peter Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[Sip13] Michael Sipser. *Introduction to the Theory of Computation.* Course Technology, Boston, MA, third edition, 2013.

[Unr10] Dominique Unruh. Quantum proofs of knowledge, 2010.

[VW16] Thomas Vidick and John Watrous. Quantum proofs. *Foundations and Trends® in Theoretical Computer Science*, 11(1-2):1–215, 2016.

[Wil16] Mark Wilde. Preface to the second edition. In *Quantum Information Theory*, pages xi–xii. Cambridge University Press, nov 2016.