Further properties of Practical Relativistic Zero-Knowledge Proofs for **NP**

Harmanpreet Singh Grover, School of Computer Science

McGill University, Montreal

October, 2021

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

Master of Computer Science

Abstract

Zero-knowledge protocols provide a means by which a prover(s) can convince a verifier(s) that some statement is true without disclosing anything else. These zero-knowledge proofs provide an elegant solution to the problem of identifying oneself without disclosing any secrets. In this work, our primary focus is on the multi-prover practical relativistic zero-knowledge protocols for separated verifier-prover pairs. Initially, we illustrate that the experimental multi-prover zero-knowledge protocol described in the recent work [ABC$^+$20] is secure against classical provers. Then, we prove that this same protocol constitutes a proof of knowledge for the same language. Further, we show that the same protocol achieves a stronger zero-knowledge property by a pair of no-signalling simulators rather than signalling ones, as is usually the case. Security of the protocol is enforced via physical principle of special relativity.

Abrégé

Les protocoles à connaissance nulle nous donnent une façon par laquelle un prouver(s) peut convaincre un vérificateur(s) qu'un énoncé est vrai sans lui dévoiler quoi que ce soit d'autre. Ces preuves à connaissance nulle nous apportent une solution élégante au problème de s'identifier sans pour autant révéler un quelconque secret. Dans ce travail, notre point de mire porte sur les protocoles multi-prouveurs relativistes à connaissance nulle pour paires distanciées de prouveurs-vérificateurs. Initialement, nous démontrons que le protocole expérimental multi-prouveurs relativiste à connaissance nulle décrit dans le papier récent de [ABC$^+$20] est sécuritaire face à des prouveurs classiques. Ensuite, nous prouvons que ce même protocole constitue une preuve de connaissance pour le même langage. Enfin, nous démontrons que ce même protocole satisfait une forme plus forte de « à connaissance nulle » en exhibant une paire de simulateurs non-signalant contrairement aux simulateurs habituels qui sont signalants. La sécurité du protocole est obtenue grâce au principe physique de la relativité restreinte.

Acknowledgements

I wish to thank, first and foremost, my supervisor Professor Claude Crépeau, who put his faith in me as his graduate student. I was constantly inspired by his expertise on the subject matter, innovative insights and boundless optimism. His timely suggestions and guidance were unparallel towards the completion of this work. Merci Claude.

Finally, my heartfelt thanks to my dearest family and friends for always being there for me, and to whom I owe everything.

Table of Contents

Abstract i

Abrégé ii

Acknowledgements iii

List of Figures vi

1 INTRODUCTION **1**

2 BACKGROUND AND PRELIMINARIES **4**

2.1 Basic Notations and Terminology 5

2.1.1 Sets and strings 5

2.2 Theory of Computation 5

2.3 Review of Quantum Computation 7

2.3.1 Vector Spaces 7

2.3.2 Quantum Information Theory 10

2.4 Relativity 17

2.5 Classical Complexity Theory 18

2.5.1 Problems and Languages 18

2.5.2 Proof Complexity 19

2.6 Graph Colouring 25

2.7 Bit commitment scheme 25

2.8 PR box 28

2.9 Zero Knowledge Proof Systems 29

2.9.1 Perfect and Computational Zero-knowledge 29
2.9.2 Proof of Knowledge 31
2.10 Review of Practical Relativistic Zero-Knowledge for NP Protocol [CMS+19] . 36
2.10.1 Summary of the protocols given in [CMS+19] 37
2.10.2 How are protocols given in section-2.10.1 [CMS+19] useful? 40

3 MAIN RESULT **42**

3.1 Analysis and Proof of Knowledge of [ABC+20] 42
3.1.1 Distribution of questions 42
3.1.2 The Protocol 43
3.1.3 Proof of knowledge 47
3.1.4 Why our Proof of Knowledge does not follow the Unruh's Quantum Proof of Knowledge [Unr12]? 50
3.2 No signalling simulation for the protocol $\Pi_{\text{lhv}}[G]$ (protocol-3.1.2) [ABC+20] . 51

4 CONCLUSION AND FUTURE WORK **57**

List of Figures

2.1 Speed of Light seems to be slowed down in the non-vacuum medium due to collisions with the particles, but the photon's velocity is c 18

2.2 Proof System for NP [Ibr20] 21

2.3 Equivalence of NP and Deterministic Interaction [Ibr20] 22

2.4 Proof System for IP [Ibr20] 23

2.5 Proof System for MIP [Ibr20] 24

2.6 Example of bit commitment scheme using safe. During commitment phase, Prover (shown on the left) sends the message M $\in \{0, 1\}$ in a closed case to the Verifier (shown on the right). Later, during the unveil phase, Prover sends the combination to Verifier so that he can retrieve the message [ZKL] 27

2.7 a **PR**-box satisfying the CHSH condition, that $a \wedge b = x \oplus y$, uniformly among solutions 29

3.1 The **3-COLZK**-box 54

Chapter 1

INTRODUCTION

The Internet can be a hazardous place to visit. Since, our day-to-day activities that occur online are increasing, therefore our exposure to online privacy risks imposed on us by fraudsters, oppressive governments and identity thieves are also increasing. Privacy-enhancing technologies (PETs) are technologies that help to reduce such online privacy threats by giving control to the users about the collection, use and dissemination of their information and day-to-day activities. Modern PETs use advanced cryptographic primitives in order to embody fundamental data protection principles [Hen14].

One such important technique is zero-knowledge proofs of knowledge. Informally, a zero-knowledge proof of knowledge is a protocol between two mutually distrusting parties, a verifier and a prover, in which the prover tries to convince the verifier that an element belongs to a language and nothing else. The prover possesses "evidence" that proves its claim in the traditional sense (E.g. a NP-witness); however, the prover never reveals the evidence or any non-trivial knowledge (formally defined later) about it, during the entire protocol, to the verifier.

The idea behind zero-knowledge proofs was first introduced in [GMR89] and it formalises the method to demonstrate existence of witness without actually revealing it. One of the most important applications of zero-knowledge proofs is the task of identification of a user, in which the user can reveal their identity by demonstrating knowledge of a secret proof of

a mathematical statement created and published by them. In this thesis we consider the problem of 3-colouring of graphs where an instance is a graph and a proof of 3-colourability allots one out of the three possible colours to each vertex of that graph, such that no two vertices connected by an edge have the same colour. One of the main reasons for choosing 3-colourability problem is that it is NP-complete (described later in 2.5.1).

A zero-knowledge proof for 3-colourability was first given in [GMW91] by assuming the existence of one-way functions, i.e., functions that can be easily calculated but for which finding an inverse image is not feasible. Under the assumption of one-way functions, the zero-knowledge proof ensures that upon participation in such a protocol, a honest prover will persuade a verifier of the validity of the claim when it is indeed valid (completeness), will not persuade the verifier when it is invalid (soundness), without improving the latter's ability of finding a 3-colouring (zero-knowledge). Such assumptions weaken the long term security of zero-knowledge protocols, that are used in various applications like cryptocurrencies [SCG+14]. As zero-knowledge property of the protocol would be compromised if the one-way function used in that protocol is (later) found to be efficiently invertible. This characteristic is very important given the advances in quantum computing [BL17, AAB+19].

Fortunately, it is feasible to construct a zero-knowledge protocol without the requirement of any computational assumption. The main idea, as introduced in [BOGKW88], is to have several provers in the interactive proofs, that try to convince the verifier of the 3-colourability of a graph without any computational assumptions. The main difference, between the original definition of interactive proof and the multi-prover scenario stands in the possibility to limit several provers to local computations, where a single prover can always talk to itself. This proposes the use of spatial separation to impose the impossibility to communicate [Kil90, Ken99] at least for short spell of time: presuming the principle of special relativity and sending queries to the different provers with precise timings, there is a short period of time in which they are physically unable to communicate with each other. But, this idea of relativistic zero-knowledge proofs has purely been of theoretical interest so far, because

known protocols require large information exchange between the verifiers and the provers. Hence, this prohibited their practical implementation.

But in [CMS+19, ABC+20], an efficient relativistic zero-knowledge proofs for a NP-complete problem was developed and implemented. In this work, we first describe the practical relativistic bit commitment scheme [ABC+20] and show that it is sound against classical provers. We then show that the same protocol is a Proof of knowledge not just a proof of membership. Therefore, it can be used for various applications like identification of a user. Further, we also comment that the protocol does not follow the approach taken by Unruh [Unr12] for showing it to be a Quantum proof of knowledge. Furthermore, we show that the same protocol possess a stronger zero-knowledge property compared to other existing zero-knowledge protocols as it requires weaker no-signalling simulators. We then provide an efficient PR box (described later in 2.8) simulation for the same protocol.

The remainder of this thesis is organised as follows. Chapter 2 covers the broad range of concepts needed from theoretical computer science and physics. Further, chapter 3 provides the main results of our thesis. Lastly, in chapter 4 we conclude our thesis and provide some future work direction.

Main Contributions

1. We show that the protocol given in [ABC+20] is sound against classical provers.
2. We show that the protocol given in [ABC+20] is a proof of knowledge.
3. We explain why our proof of knowledge does not follow Quantum proof of knowledge technique given in [Unr12].
4. We provide an efficient PR box (described later in section-2.8) simulation for the [ABC+20] protocol. Therefore, showing that the protocol possesses stronger zero-knowledge property due to use of weaker no-signalling simulators.

Chapter 2

BACKGROUND AND PRELIMINARIES

In this chapter we give a brief introduction of the various concepts from different fields including Theoretical Computer Science and the Einstein's Special Relativity that are necessary for the understanding of this thesis. This includes various basic notations, definitions and mathematics used throughout this thesis. We start with the definitions of bits and string, an introduction to the theory of computation, a review of quantum computation, relativity and classical complexity theory. We then do a quick overview of the graph colouring problem. Further, we give the details of bit commitment scheme and PR box. Next we describe in detail the concept of zero knowledge proof systems including zero knowledge proofs of knowledge. Finally, we provide the explanation of the first version of the protocol for which proof of knowledge is provided in this work and also compare it's computation cost to prior similar protocol.

2.1 Basic Notations and Terminology

2.1.1 Sets and strings

For the empty and finite sets, we use upper case Greek or Latin letters, where as, for elements of those sets, we use lower case letters. For example, if E is the set of edges in a graph, then e $\in$ E is one of the edge of that graph. We use $\emptyset$ to denote an empty set. We represent the sets of natural numbers (including 0), integers, real numbers and complex numbers by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{C}$ respectively. We denote the natural number set $\{1, 2, ..., n\}$ by $[n]$. Also, a bit is the most basic unit of information, that belongs to set $\{0, 1\}$, in computer science and a bit string is a sequence of zero or more bits.

2.2 Theory of Computation

In this section, we will summarize the most essential concepts from this subject matter. For readers interested to explore this topic further can refer to [Sip96].

Definition 2.2.1. *(Turing Machine): A Turing machine, $\mathcal{M}$, is defined by $\langle Q, \Gamma, \Sigma, \delta, q_0, F\rangle$, together with a one directional infinite tape such that,*

- $Q \neq \emptyset$ *is a finite set of states*
- $\Gamma \neq \emptyset$ *is a finite set of alphabet symbols*
- $b \in \Gamma$ *denotes an empty/blank symbol*
- $\Sigma \subseteq (\Gamma$ - $\{b\})$ *is the input alphabet to $\mathcal{M}$, this includes the symbols that are allowed to appear in the initial tape configuration*
- $q_0 \in Q$ *is the starting state of $\mathcal{M}$*
- $F \subseteq Q$ *are the final accepting states of $\mathcal{M}$*
- $\delta : (Q$ - $F) \times \Gamma \nrightarrow Q \times \Gamma \times \{L, R\}$ *is a partial function representing the transition table, where R and L instructs the tape head to move right and left respectively.*

A Turing Machine (TM) is a mathematical model of computation that describes an abstract machine, which manipulates symbols $\sigma \in \Sigma$ on a strip of tape according to a partial function δ. The term 'state' in the above means the name/designator of the current instruction to be performed. The machine has a tape head that points to some cell on the tape strip and can read the value of that cell. The machine moves the head (right or left) according to δ rules or writes a new value to the current cell. A Turing machine can simulate any computer algorithm. A TM is one of the most important models of computation that describes the computation of an output of a function corresponding to an input.

Various types of Turing Machines (TM) exist including the ones having multiple tapes or two-way infinite tapes with read-only, write-only or read/write capabilities. All these single machine models can simulate each other with a polynomial time overhead. These variations exist just for simplicity and convenience (called as universality of TMs).

Definition 2.2.2. *(Non-Deterministic Turing Machine): A non-deterministic Turing Machine is a TM having a Transition relation δ in place of a function*

$$\delta : ((Q - F) \times \Gamma) \times Q \times \Gamma \times \{L, R\} \tag{2.1}$$

Now, the outcome of the δ is a set of possibilities instead of a single outcome. A non-deterministic TM accepts an input string if any of the computational paths starting from that input leads the TM to an accepting state.

Definition 2.2.3. *(Probabilistic Turing Machine): A probabilistic Turing Machine is a TM having an extra tape which is filled with new random symbols from Σ during each initialization. Therefore, probabilistic Turing Machine is a TM that chooses between the available transitions at each point according to the random tape content.*

Definition 2.2.4. *(Reducibility): A reduction from problem A to problem B is a function f: $\Sigma_A^* \to \Sigma_B^*$, such that, $\forall a \in \Sigma_A^*$ we have*

$$a \in A \iff f(a) \in B \tag{2.2}$$

If the function f can be implemented in polynomial time then we call it to be Karp/polynomial reduction (we don't care about other types of reductions in this work). If B can be reduced to A in polynomial time, we denote it by $B \leq_P A$ and say that A is at least as hard as B.

2.3 Review of Quantum Computation

In this section, we will try to understand the concepts from quantum computation that are necessary for this thesis.

2.3.1 Vector Spaces

A vector is a collection of elements that belong to a set (such as $\mathbb{R}$ or $\mathbb{C}$) and the set of such vectors is called a vector space. Here, we consider only the vector spaces that are finite and are over $\mathbb{C}$ (complex vector space). For representing the space of all vectors, composed of n complex numbers, we use $\mathbb{C}^n$. Elements of such space can be represented as (v_1, v_2, ..., v_n) or more frequently

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

Definition 2.3.1. *(Inner product): Inner product of two vectors u and v, where u, $v \in \mathbb{C}^n$ for some $n \in \mathbb{N}$ is defined as*

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

where u_i^* represents the complex conjugate of i^{th} element of u. If $c = a + ib$ is a complex number then its complex conjugate $c^* = a - ib$, where a, $b \in \mathbb{R}$ and $i \stackrel{def}{=\!=} \sqrt{-1}$.

A Hilbert space is a complex vector space with a map of inner product.

Definition 2.3.2. *(Norm): Norm of a vector $v \in \mathbb{C}^n$ is defined as*

$$||v|| \stackrel{def}{=\!=} \sqrt{\langle v, v \rangle}$$

A vector u having $||u|| = 1$ is called a unit vector. Also, a normalized vector u has a unit norm and we can normalize a vector u, where $||u|| \neq 0$, by performing $\frac{u}{||u||}$. Lastly, if two vectors $u, v \in \mathbb{C}^n$ have $\langle u, v \rangle = 0$, we call them orthogonal if they are not unit vectors and orthonormal if they are unit vectors.

Dirac Notation and Linear Operator

In this section, we describe the Dirac notation that is frequently used in quantum information theory. Dirac notation is also called as the bra-ket notation and a vector in this notation is represented as

$$|\psi\rangle$$

where ψ is a unit vector in Hilbert space and the $|\cdot\rangle$ symbol is known as a ket. Similarly, the symbol for bra, dual element of $|\cdot\rangle$, is as follows

$$\langle\cdot|$$

Now, for any vector $|\psi\rangle$, we have its dual vector, denoted by $\langle\psi|$. Dual vector $\langle\psi|$ is the conjugate transpose, represented by $\dagger$, of $|\psi\rangle$ with each of its element being its complex conjugate. So, we have

$$\langle\psi| \stackrel{def}{=\!=} |\psi\rangle^\dagger = [\psi_1^* \cdots \psi_n^*]$$

Definition 2.3.3. *(Inner product): The inner product of two vectors $|\psi\rangle$ and $|\phi\rangle$ is defined as*

$$\langle\psi|\phi\rangle \stackrel{def}{=\!=} \sum_i \psi_i^* \phi_i = [\psi_1^* \cdots \psi_n^*] \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{bmatrix}$$

From above, we can notice that inner product of two vectors is the multiplication of bra of one vector with the ket of another vector, that is why it is named as "bra-ket". For any vector $|\psi\rangle$, its norm is simply $\sqrt{\langle\psi|\psi\rangle}$. We now define one more linear operation called as the outer product. Outer product of two n $\times$ 1 dimension vectors produces an n $\times$ n dimension matrix as a result.

Definition 2.3.4. *(Outer product): The outer product of two vectors $|\psi\rangle$, $|\phi\rangle \in \mathbb{C}^n$ is defined as*

$$|\psi \rangle\!\langle \phi| \stackrel{def}{=\!=} \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_n \end{bmatrix} [\phi_1^* \cdots \phi_n^*] = \begin{bmatrix} \psi_1\phi_1^* & \cdots & \psi_1\phi_n^* \\ \vdots & \ddots & \vdots \\ \psi_n\phi_1^* & \cdots & \psi_n\phi_n^* \end{bmatrix}$$

We now define linear operators.

Definition 2.3.5. *(Linear Operator): For any two given vector spaces X and Y, a linear operator between these spaces is any function f: $X \to Y$ s.t. f is linear in its input*

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

We denote the set of all linear operators that map from $X \to Y$ by $\mathcal{L}(X,Y)$ and $\mathcal{L}(X,X) = \mathcal{L}(X)$

Linear operators are usually represented in a matrix form for convenience. Identity operator $\mathbb{1}$ which satisfies $\mathbb{1}|\psi\rangle = |\psi\rangle$ is one of the most important linear operator and can be represented as the identity matrix.

2.3.2 Quantum Information Theory

Quantum mechanics is a mathematical framework, used for developing physical theories, that overcomes the shortcomings of the classical mechanics. Classical theories failed to predict the result of a number of experiments. The Stern-Gerlach experiment is one of the most notable experiment that made physicists to think outside the classical mechanics. We will not go into the details of such experiments and how classical mechanics failed to predict their results. Instead, we will cover the most important concepts of quantum information theory from a computer scientist's perspective. For readers that are interested about learning the development of quantum information, we recommend [NC02, Wil13].

Qubits

In classical computer science, the fundamental unit of information is a binary digit, also called as a bit. A bit can have two values which is either 0 that signifies false or 1 that signifies true in mathematical logic. A binary digit can represent a two-state system like "on/off" of a switch. The quantum analog of a classical bit is known as a quantum bit, or qubit. It can represent any fundamental two-level quantum system like polarization of a photon, spin of an electron, or ground and excited state of an atom.

A qubit is a basic unit of quantum information. It is a two-level quantum mechanical system. We can use linear combination of orthonormal basis $|0\rangle$, $|1\rangle$ to represent any arbitrary pure qubit:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

where α and β are complex numbers that satisfy the following condition:

$$|\alpha|^2 + |\beta|^2 = 1$$

and $|c|^2 = a^2 + b^2$ if $c = a + ib$. The complex numbers α, β are probability amplitudes. The above equation of $|\psi\rangle$ indicate that $|\psi\rangle$ has $|\alpha|^2$ probability of being in state $|0\rangle$ and $|\beta|^2$ probability of being in state $|1\rangle$ when measured. It is often convenient to represent states as their vector representation. The expression for qubit $|\psi\rangle$ is as follows:

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

where $|0\rangle = [1\ 0]^T$ and $|1\rangle = [0\ 1]^T$. The special states $|0\rangle$ and $|1\rangle$, known as the computational basis, are the quantum analogs of classical 0 and 1 states. The linear combination of any pair of orthonormal vectors can represent a qubit.

Unlike classical bits, a qubit can be in states other than $|0\rangle$ or $|1\rangle$, known as superposition of states. Even though there can be infinite number of linear combinations of states, but we cannot take out infinite information from one qubit. When a qubit is measured, the state of the qubit collapses to a basis state $|0\rangle$ or $|1\rangle$, that corresponds to classical bits 0 or 1. Also, the result will be the same even if that qubit is measured again. This suggests that the probability amplitudes α and β of a qubit cannot be determined, unless there are many identical qubits. In spite of the fact that by measuring qubit we only learn one classical bit of information, there are many advantages of quantum computing over classical case because of entanglement and quantum gates.

Unitary Transformation

A unitary transformation is the rotation of axes in the Hilbert space. This transformation is a reversible linear operator that do not leak classical information. It can be expressed as a unitary matrix U that has to satisfy the below condition:

$$U^\dagger U = UU^\dagger = \mathbb{1}$$

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

$$X \stackrel{def}{=\!=} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y \stackrel{def}{=\!=} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z \stackrel{def}{=\!=} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.3}$$

where the Pauli-X is the quantum counterpart of the NOT gate. Hadamard gate H is another frequently used single-qubit quantum gate

$$H \stackrel{def}{=\!=} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Application of the quantum gates to a qubit is exactly same as the application of the unitary matrix to a qubit. Output of applying Hadamard gate on $|0\rangle$ gives

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

where $H|1\rangle = \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$. We use $|+\rangle = H|0\rangle$ and $|-\rangle = H|1\rangle$. The pair $\{|+\rangle, |-\rangle\}$, called as the Hadamard basis or the diagonal basis, forms an orthonormal basis.

Now, we define tensor product, which is used for the transformation of the multiple qubit states. Even though we call it tensor product, it is actually called a Kronecker product in the specific context of matrices as in quantum computation. In this text, we will give only describe tensor product in operational sense.

Definition 2.3.6. *(Tensor product): Tensor product of two matrices A and B of dimensions $m \times n$ and $p \times q$ respectively is as follows:*

$$A \otimes B \stackrel{def}{=\!=} \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}B & A_{n2}B & \cdots & A_{mn}B \end{bmatrix}$$

Since, $A_{ij}B$ has dimensions $p \times q$. Therefore, above belongs to $nq \times mp$ dimension Hilbert space.

The tensor product of $|0\rangle$ and $|0\rangle$ is a two qubit state $|00\rangle$,

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

Similarly, tensor product of Pauli-X gate and the Hadamard gate is as follows

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

We now provide the controlled-not gate or CNOT gate, which is the analog of the classical XOR gate and is one of the most important gates along with the Hadamard gate. Since, it

is responsible for most non-local results of quantum computation.

$$CNOT \stackrel{def}{=\!=} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Lastly, It is worthwhile to mention that $H^{\otimes n}$ is used to represent Hadamard gates tensored with itself n times.

Measurement

The measurement of a quantum system is defined as a collection of measurement operators $\{M_i\}_i$ in quantum mechanics. The index i represents the possible outcomes. The measurement operators are subjected to

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

which means that the sum of probabilities of measurement outcomes is one. Also, the probability of occurrence of outcome i after measurement of state $|\psi\rangle$, is

$$p_i = \langle\psi|M_i^\dagger M_i|\psi\rangle$$

and the resultant state after the quantum system collapses, when the result is i, is

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

Let us now calculate the probability amplitudes of a state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, represented in computational basis. So, the probability of the resultant state to be $|0\rangle$, when measurement

operators are $M_0 = \langle 0|^\dagger \langle 0|$ and $M_1 = \langle 1|^\dagger \langle 1|$, is

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

Similarly, for the result $|1\rangle$ we have,

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

Above results show why α and β are called probability amplitudes.

Entanglement

In this section, we describe one of the most important concepts of quantum computing that gives it various advantages over classical computing. Early quantum protocols like quantum teleportation [BBC$^+$93] and super dense coding [BW92] are mainly dependent on this concept. Before describing entanglement, we first need to understand the product composite quantum systems.

Definition 2.3.7. *(Product states): Product states are pure quantum states that can be expressed as the tensor products of single quantum states. Assuming two Hilbert Spaces $\mathcal{H}_A$, $\mathcal{H}_B$, we call $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ a product state only if there exist states $|\phi\rangle \in \mathcal{H}_A$ and $|\xi\rangle \in \mathcal{H}_B$ such that*

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

Now, we define entanglement as below:

Definition 2.3.8. *(Entanglement): A pure quantum state that is not a product is entangled. An example of entangled two qubit state is*

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

Actually, the state $|\Psi^{-}\rangle$ is the famous EPR state [EPR35] and is part of the four Bell states.

Definition 2.3.9. *(Bell states): Bell states are the four maximally entangled bipartite quantum states, and they are as follows*

$$\left|\Phi^{+}\right\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$\left|\Phi^{-}\right\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

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

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

Now, in order to understand the properties of the entanglement, consider two physicists Alice and Bob such that, they first construct a Bell state $|\Psi^{-}\rangle$ and then go to different labs with one qubit each. Later, both of them decide to measure their qubits in $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$ computational basis. From 2.3.2, we know that whosoever measures first has $||\frac{1}{\sqrt{2}}||^2 = \frac{1}{2}$ probability of getting 0 and $||\frac{1}{\sqrt{2}}||^2 = \frac{1}{2}$ probability of getting 1. But the peculiar thing here is that, once one party measures his/her qubit and gets a classical output $b \in \{0, 1\}$, he/she will know the other parties outcome, in this case $\bar{b} = 1$ - b, independent of the fact whether other party has performed the measurement or not. In terms of information,

density matrix (not covered in this text, but interested readers can refer to [NC02, Wil13]) representations of the Alice's qubit and Bob's qubit is same and given by

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

So, the state for Bob's qubit does not change before measurement, by the measurement of Alice's qubit. Hence, no information is being transmitted by measurement and therefore, the causality constraint is not violated.

2.4 Relativity

For understanding terms like signalling we cover concepts such as frame of reference, no-communication theorem and few others from the Einstein's special theory of relativity. We recommend Einstein's book [She16] for further reading.

Definition 2.4.1. *(Frame of Reference): A frame of reference is a set of coordinates relative to which measurement of physical properties are performed.*

Definition 2.4.2. *(Inertial Frame of Reference): An inertial frame of reference is a reference frame in which a physical object moves with a constant speed, which may be zero, unless acted upon by an external force.*

Postulate 2.4.0.1. *(The Principle of Relativity): The laws of nature are the same for all inertial frames of reference.*

Postulate 2.4.0.2. *(Invariance of the Speed of Light c): The speed of light, c, is a constant, independent of the state of motion of the source.*

Definition 2.4.3. *(No-communication theorem): No-communication theorem states that information cannot move from one place to another faster than the speed of Light.*

The above two postulates can help to solve some complications in the practical implementation of the multi-prover interactive proof system described in 2.5.13. Since, in the

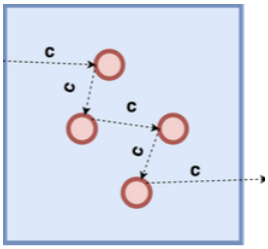


Figure 2.1: Speed of Light seems to be slowed down in the non-vacuum medium due to collisions with the particles, but the photon's velocity is c

implemention of the MIP the provers are separated by a "distance" such that they cannot collude with each other while interacting with the verifiers. So, the "distance" should be large enough to prevent the travel of information between the provers throughout the entire duration of their interaction with the verifiers.

2.5 Classical Complexity Theory

For readers interested to go deeper in complexity theory we recommend [AB09]. In this section, we cover only the basic definitions from classical complexity theory.

2.5.1 Problems and Languages

Computational problems can be expressed as a either an optimization problem, a decision problem, a counting problem or a search problem. We briefly describe all four problems below.

Definition 2.5.1. *(Binary Relation): A binary relation, $\mathcal{R}$, consisting of a domain set A and a codomain set B, is a subset of the cartesian product $A \times B$.*

Definition 2.5.2. *(Decision Problem): A decision problem is a boolean function f: $\{0,1\}^n \rightarrow \{0,1\}$ that maps the binary encoding of an input to the problem to a YES/NO answer.*

Definition 2.5.3. *(Formal Language): A language is the set of inputs to a decision problem that outputs 1(i.e, YES).*

$$\mathcal{L}_f = \{x : f(x) = 1\} \tag{2.4}$$

Decision problems are the main problems in the field of complexity theory. Computational complexity is mainly concerned about deciding, using an algorithm $\mathcal{A}$, whether a given string belongs to a formal language under consideration. If $\mathcal{A}$ returns an output YES, then the input string is considered a member of the formal language, otherwise, $\mathcal{A}$ can behave differently like stopping and rejecting membership, looping forever or having an undefined behaviour.

Now, we'll discuss two very important concepts: (1) Hard language, (2) Complete language

Definition 2.5.4. *(C-Hard): A language L is C-Hard, for some complexity class C, iff $\forall$ language L' $\in$ C we have L' $\leq_p$ L.*

Definition 2.5.5. *(C-Complete): A language L is C-Complete, for some complexity class C, iff L is C-Hard and L $\in$ C.*

Why NP-Complete problems are important?

1. NP-complete problems can be thought of as capturing the entire difficulty of NP.
2. Every problem in NP can be polynomially reduced to any NP-complete problem. Therefore, if a deterministic polynomial time algorithm can be found to solve one of them, every NP problem would become solvable in polynomial time.

2.5.2 Proof Complexity

In this section, we first define a proof system which is used to study the computational resources required to prove or disprove statement.

Definition 2.5.6. *(Proof System): A propositional proof system is described by a proof-verification algorithm $\mathcal{A}(x,t)$ with two inputs, where t is the transcript of the alleged proof and x is the proposition. The alleged proof is provided by some prover(s). If t is a proof of x, then proof-verification algorithm accepts, i.e. $\mathcal{A}(x,t) = 1$. $\mathcal{A}$ is required to be efficient with a low false acceptance rate (soundness) and a low false rejection rate (completeness). A proof-verification algorithm is commonly called a verifier.*

Completeness measures how good a proof system is at creating proofs for valid statements, whereas soundness means that every statement that is provable, by the verifier, is in fact true.

Definition 2.5.7. *(Completeness): A probabilistic proof system for language L, has completeness α, where $0 \leq \alpha \leq 1$, if:*

$$\forall x \in L \; \exists \; t \; Pr[\mathcal{V}(x,t) = 1] \geq \alpha$$

where $\mathcal{V}$ is a honest verifier and x, of length poly(x), is a proof generated by a prover for a valid statement that an honest verifier accepts.

When $\alpha = 1$, we call it Perfect Completeness.

Definition 2.5.8. *(Soundness): Ideally, for a false statement a prover should not be able to convince an honest verifier, $\mathcal{V}$, that the statement is valid. So, a probabilistic proof system for a language L has soundness 1 - β, where $0 \leq \beta \leq 1$, if:*

$$\forall x \notin L \; and \; \forall t \; Pr[\mathcal{V}(x,t) = 1] \leq \beta$$

When $\beta = 0$, we call it Perfect Soundness.

Now, we'll provide brief explanation to some of the most important canonical proof systems of complexity theory. Let us assume an input set of functions, $I = \{0,1\}^*$, and $I_n = \{x \in I || x| = n\}$

Definition 2.5.9. *(The class P): A language $L \subseteq I$ is said to belong to class P, if there exists an algorithm A for computing membership in L, s.t. $\exists$ a positive constant c s.t. for every n and every $x \in L$, A calculates $x \in L$ in $O(n^c)$ time.*

In class P, a verifier just requires the input x, for some language L. Therefore, it is the simplest form of proof system since there is no need of proof transcript t and it can just use a polynomial time computation on input x to verify its membership in L.

Definition 2.5.10. *(The class NP): A language L is said to be in class NP, if there exists a deterministic polynomial time verifier Turing Machine, V_c, and a constant c s.t.*

- *If $x \in L$, then $\exists t$ with $|t| \in O(|x|^c)$ and $V_c(x, t) = 1$*
- *If $x \notin L$, then $\forall t\ V_c(x, t) = 0$*

NP is also described by the languages that can be solved by a non-deterministic Turing Machine in polynomial time. By making use of both definitions, we can consider a proof system 2.2 in which the prover has access to an exponential deterministic Turing Machine as a resource and can find a solution for any language L ∈ NP. The prover and the deterministic polynomial-time verifier get the input x. The prover then shares the proof t of instance x with the verifier in order to convince him that x ∈ L. Therefore, the verifier will accept the input if x ∈ L. But, for x ∉ L, if the malicious prover gives any proof t' to the verifier, then the verifier would always reject then we say L belongs to class NP.

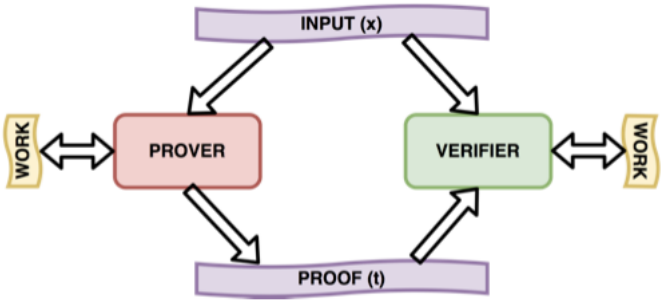
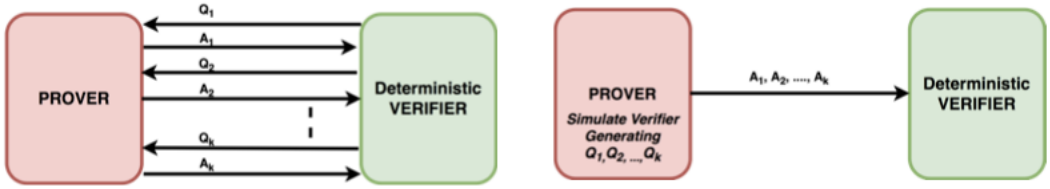


Figure 2.2: Proof System for NP [Ibr20]

Now, we provide the interaction in proof systems with two different definitions, provided in two independent different works [Bab85, GMR89]. Let us first consider a Verifier, a deterministic TM, and a Prover having a back-and-forth interaction. Now, we'll sketch why it is equivalent to NP. To see this, consider a verifier that deterministically chooses on a first question and shares it with the prover and then the prover provides a response back. The verifier then deterministically chooses the next question (that might depend on prover's response) and shares it again with the prover which responds back. Imagine this back-and-forth communication happens for polynomial rounds (since verifiers are polynomial time machines).

If you were an all powerful prover, then you could simply simulate the verifier, produce the first question, make up what verifier could ask next, given your response to the generated question, then again simulate the verifier to produce the next questions. You can repeat these steps to have a transcript filled with this interaction. Then, you can simply begin by sharing this transcript with the verifier and the verifier could check that the transcripts indeed show what verifier would have done (shown in Fig 2.3).



(a) Deterministic Interactive Proof (b) Unrolling of Deterministic Interaction

Figure 2.3: Equivalence of NP and Deterministic Interaction [Ibr20]

Definition 2.5.11. *(Bounded-Error Probabilistic Polynomial Time (BPP)): A language L ∈ BPP, if there exists a probabilistic Turing Machine, $\mathcal{M}$, that runs in polynomial time on all inputs and satisfy:*

- *(Completeness)* $x \in L \implies Pr[\mathcal{M}(x) = 1] \geq \frac{2}{3}$
- *(Soundness)* $x \notin L \implies Pr[\mathcal{M}(x) = 1] \leq \frac{1}{3}$

Definition 2.5.12. *(Interactive Proofs (IP)): A language L is said to belong to IP[k], if there exists a deterministic polynomial-time TM $\mathcal{V}$, the verifier, s.t. given a problem instance x, generated randomness tape r for $\mathcal{V}$, and some prover $\mathcal{P}$ communicating with $\mathcal{V}$ for k(|x|) rounds, for some polynomial time computable function k: $\mathbb{N} \to \mathbb{N}$ and $\mathcal{V}$ runs in polynomial time in |x| s.t.*

- *(Completeness):* $\exists \mathcal{P}$ *s.t. if* $x \in L \longrightarrow Pr[output_{\mathcal{V}}(\mathcal{V}(r,x) \overset{k}{\longleftrightarrow} \mathcal{P}(x)) = 1] \geq \frac{2}{3}$
- *(Soundness):* $\forall \mathcal{P}$ *s.t. if* $x \notin L \longrightarrow Pr[output_{\mathcal{V}}(\mathcal{V}(r,x) \overset{k}{\longleftrightarrow} \mathcal{P}(x)) = 1] \leq \frac{1}{3}$

In proof system for IP (Fig. 2.4), the verifier can be thought as a TM with read-once-access to a randomness tape where as the prover is an all-powerful TM. Both the verifier and

the prover read the input problem instance x, from the input tape, and then they interact back and forth via a shared communication tape. The verifier usually starts this interaction. Each TM has access to its own read-write work tape. Many attempts have been made to try to restrict the verifier's powers, which include restricting the amount of extra space used in verifying the proof, or number of bits read from prover's proof, or allowing only a number of random bits to be used by the verifier, or restricting time.

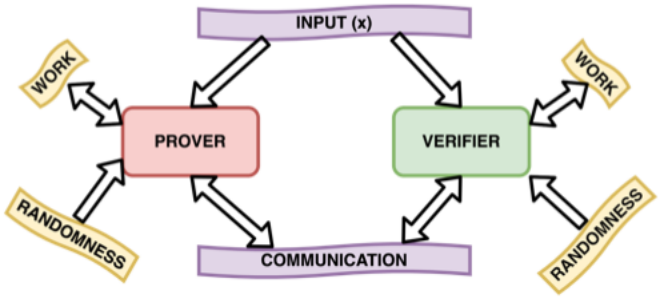


Figure 2.4: Proof System for IP [Ibr20]

We will now describe an extension to IP which gives us more powerful proof system . The extension is to add more provers in the proof system. It is trivial to understand that since the single prover in IP was considered all-powerful, adding more provers naively would not help in recognizing more languages. The main idea here is to separate the provers such that they are unable to interact. Now, we will describe a multi-prover interactive proof system (Fig 2.5) in which we have a probabilistic polynomial-time TM, denoting the verifier, interacting with $n \in \mathbb{N}$ provers. Before the interaction process begins, the provers can negotiate among themselves and decide on an optimal strategy (which maximize their chances of winning against the verifier) and also the strategy's randomness is hidden from the verifier. But, once the provers start interacting with the verifier, they can no longer communicate with each other. Therefore, the verifier can verify proofs to stronger languages by asking various questions to these non-interacting provers.

Definition 2.5.13. *(Multi-Prover Interactive Proofs (MIP)): Consider a polynomial time computable function k:* $\mathbb{N} \to \mathbb{N}$ *which denotes the upper bound on the number of rounds*

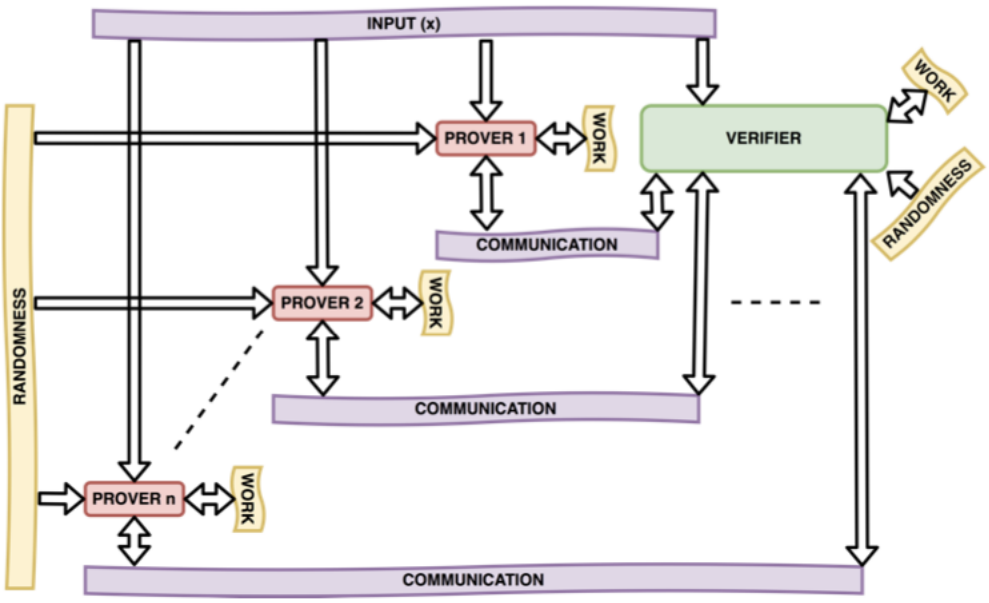


Figure 2.5: Proof System for MIP [Ibr20]

sufficient for each input size. A language L is said to belong to MIP[n] if there exists a deterministic polynomial-time verifier $\mathcal{V}$ s.t. given an instance x of a problem, generated randomness tape r for $\mathcal{V}$, n non-interacting-provers $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n$ sharing an infinite read-only random tape and $\forall i (1 \leq i \leq n)$, $P_i \overset{k(|x|)}{\longleftrightarrow} \mathcal{V}$, we have:

- *(Completeness): $\exists \mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n$ s.t. if $x \in L$ then $Pr[output_{\mathcal{V}}(\mathcal{V}(r,x) \overset{k}{\longleftrightarrow} \langle \mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n \rangle(x)) = 1] \geq \frac{2}{3}$*
- *(Soundness): $\forall \mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n$ s.t. if $x \notin L$ then, $Pr[output_{\mathcal{V}}(\mathcal{V}(r,x) \overset{k}{\longleftrightarrow} \langle \mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n \rangle(x)) = 1] \leq \frac{1}{3}$*

With this we complete our survey of important proof systems and classes from classical complexity theory.

Now, we will shift our focus to the NP-complete problem on which the protocol of [ABC+20] is based.

2.6 Graph Colouring

In graph k-colouring problem we try to assign colours (using at most k colours) to the vertices of a graph $G = (V, E)$, where V and E represent the sets of vertices and edges of G respectively, such that no two adjacent vertices have the same colour, i.e.,

$$\forall (u, v) \in E : c(u) \neq c(v)$$

where $c(u)$ and $c(v)$ represent the colours of vertices u and v respectively. A G is said to be k-proper coloured if each of its vertex is assigned one out of the k colours such that no two adjacent vertices have the same colour.

Definition 2.6.1. *(Colour class): A subset of vertices that receive the same colour in colouring problem is called a colour class.*

Definition 2.6.2. *(Chromatic number): It is the smallest value of k for which the G admits a k-proper colouring.*

Definition 2.6.3. *(3-Colouring problem): A graph colouring problem in which we are allowed to use at most 3 colours, is called a 3-colouring problem.*

Theorem 2.6.1. ***3-Colouring is NP-Complete***

2.7 Bit commitment scheme

A commitment scheme is a vital cryptographic technique that acts as a building block for various other more complex primitives like secret sharing, signature schemes and zero knowledge proofs. It is first given by Blum [Blu83] in the scenario of fair coin flipping over telephone in which two parties wish to agree on the outcome of a coin flip over telephone but the parties don't necessarily trust each other. In his work, Blum used the idea of random hard problems to commit. However, one can also argue that earlier work on mental poker by Shamir et al. [SRA81] implicitly used commitments, since in order to generate a fair deal of cards, Alice

encrypts the card names under her own encryption key, which is the basic idea for implementing commitments. Later, Brassard and Crépeau [BC86] describe the commitments in the context of interactive zero knowledge proofs (described in detail in section 2.9) on NP problems. One thing that is common in various applications of the commitment schemes is that it can prove to be useful when untrusting parties wish to come to an agreement.

Commitment schemes permit a party to commit to a chosen value (or statement) digitally and later reveal it to the other involved parties. The commitment schemes are designed in such a manner that the committing party cannot change the value once it has committed to it, i.e, the commitment schemes are binding. Such schemes are called bit commitment schemes when the committed value is a single bit, let it be $b \in \{0, 1\}$. These schemes ensure security for all the involved parties like in a two-party commitment scheme the committing party is bound to their choice and the receiving party cannot extract any information prior to the revealing [DPP93, HM96]. An illustration for such a scheme is a safe where the committing party puts the message inside the case and locks it with a code and sends it to the other party. Later at the time of unveiling, the sender sends the code to the receiver who opens the case and get the message. Fig 2.6 gives the illustration of the locked case example.

Definition 2.7.1. *(Commitment Scheme): A commitment scheme is a two step process between two communicating parties. Let Alice be the party committing and sending the message and Bob be the party receiving the committed message. During the commit phase, Alice chooses a message m, encrypts it, let c be the encrypted message, and sends it to the Bob. Now, during the unveil phase, the Alice sends the information required by the Bob to decrypt the encrypted message c and get the message m. Two main properties that a commitment scheme should satisfy are given below:*

- ***hiding****: The property that receiver, i.e. Bob, should not be able to learn any information regarding message m just from the encrypted message c. Hiding provides security against Bob.*

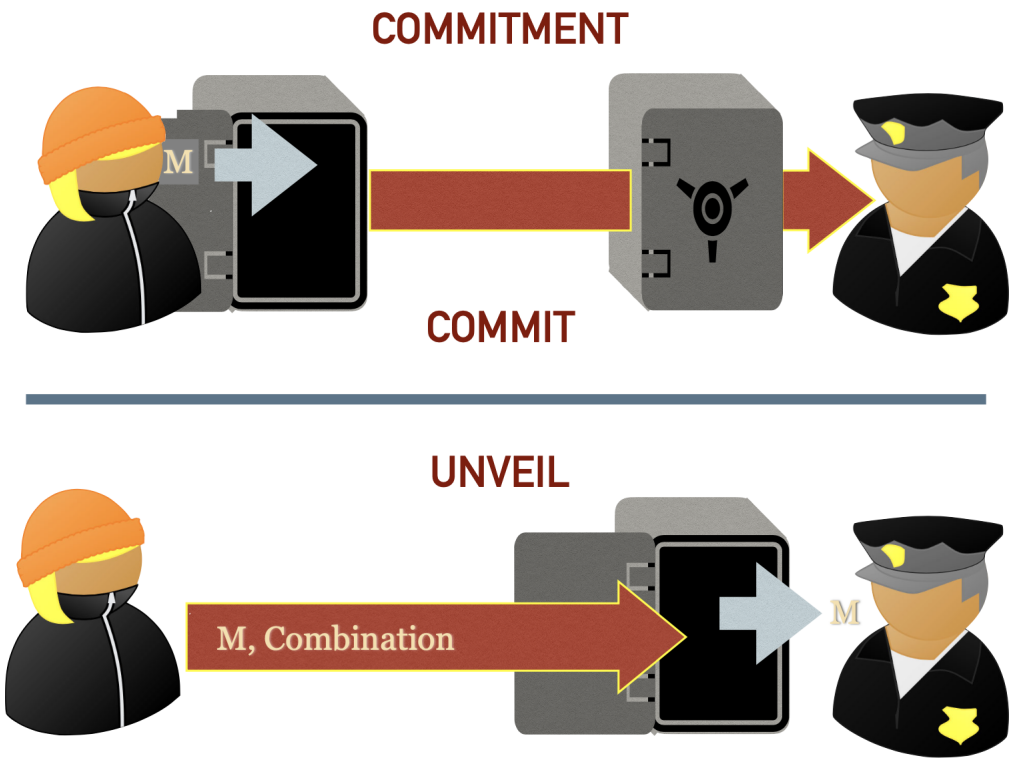


Figure 2.6: Example of bit commitment scheme using safe. During commitment phase, Prover (shown on the left) sends the message M $\in \{0, 1\}$ in a closed case to the Verifier (shown on the right). Later, during the unveil phase, Prover sends the combination to Verifier so that he can retrieve the message [ZKL]

- ***binding***: *Committed message c should not be able to be opened to more than one value of m by Alice. This property provides security against Alice.*

An unconditional and perfectly secure bit commitment scheme is well known to be impossible both classically and quantumly [LC97, LC98, May97]. For the classical case, there is an information theoretic argument and the intuition behind the idea is as follows. The commitment scheme can be unconditionally hiding, only if c can be produced by any message as c should not reveal any information regarding committed message m. On the other hand, for the commitment scheme to be unconditionally binding, the encrypted message c must contain enough information so that if Alice tries to change the original message m the Bob should be able to detect it. So it is evident that both these conditions are impossible to satisfy simultaneously. Therefore, we consider security of commitment schemes in computation

terms. But, for understanding the argument of the quantum case some basic knowledge of quantum information is required.

2.8 PR box

In this section, we provide the formal description of PR box (shown in 2.7) mentioned in section 1. A PR box was introduced by Popescu and Rohrlich in [PR94, PR98]. It is an imaginary gadget that can achieve Clauser-Horne-Shimony-Holt (CHSH) [CHSH69] correlation. The box takes two binary inputs x and y, and gives back two output bits a and b satisfying $a \oplus b = x \wedge y$. The correlation can be captured in the following with both inputs being randomly sampled from a uniform distribution,

$$\Pr(a, b|x, y) = \begin{cases} \frac{1}{2} & \text{if } a \oplus b = x \wedge y \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

The PR box outputs a as soon as it receives the first input x even when second input y is yet to be received, i.e., it works in an asynchronous manner. The box satisfies the relativity constraints as no information is communicated through its use. The players restricted to only local computations can successfully simulate the PR box with a maximum probability of 75%, whereas the quantum players sharing entanglement can do it 85% of the time. The PR box can be generalized to a more fundamental information theoretic concept known as no-signalling. No-signalling provers are allowed to make use of only no-signalling correlations and/or PR box. A restriction put on no-signalling provers is that they cannot communicate, which is also the least amount of restriction in terms of communicational power. One of the results of the PR box is that it can achieve trivial classical communication complexity [VD13] suggesting that a physical implementation of a PR box might be impossible. In spite of this, the PR box still finds its significance for cryptographers as no-signalling provers can break cryptographic protocols which are even secure against quantum adversaries [CSST11].

Now, we provide some important theorems and definitions [CRC19]:

Figure 2.7: a **PR**-box satisfying the CHSH condition, that $a \wedge b = x \oplus y$, uniformly among solutions

Definition 2.8.1. *(Strategy) : A strategy for Alice and Bob is a probability distribution Pr(x, y | a, b) describing exactly how they will answer (x, y) on every pair of questions (a, b) that are chosen according to a distribution.*

Definition 2.8.2. *(No-signalling) : A strategy under a certain distribution is no-signalling, if you can produce exactly the same input output relation by signalling in one direction (either left-signalling or right-signalling).*

Theorem 2.8.1. *If a strategy is achievable using one way signalling and symmetric then it is no-signalling [Theorem 17 from [CRC19]].*

With this we complete the description of PR box and also the important definitions and theorems related to it.

2.9 Zero Knowledge Proof Systems

Informally, a zero-knowledge protocol or a zero-knowledge proof is a cryptographic technique by which the prover can prove to a verifier that a witness of $x \in L$ exists, without actually disclosing any computational advantage about computing a witness, i.e, the verifier only learns the fact that a witness exists. In other words, zero knowledge proofs are the proofs that yield nothing beyond the existence of the witness.

Now, we describe the concept of zero-knowledge interactive proof system.

2.9.1 Perfect and Computational Zero-knowledge

In simple terms, an interactive proof system $(\mathcal{P}, \mathcal{V})$ for language L is considered zero-knowledge if whatever can be calculated after communicating with $\mathcal{P}$ on input $x \in$ L can

also be easily calculated from x (without any communication). Zero-knowledge captures $\mathcal{P}$'s ability to protect against attempts to gain computational advantage by communicating with it.

Definition 2.9.1. *(Perfect Zero-knowledge): Consider $(\mathcal{P}, \mathcal{V})$ be an interactive proof system for some Language L. We call $(\mathcal{P}, \mathcal{V})$ to be perfect zero-knowledge if $\forall$ probabilistic polynomial-time interactive machine $\mathcal{V}^*$ there exists a probabilistic polynomial-time algorithm $\mathcal{S}^*$ s.t. $\forall x \in L$ the following two random variables are identically distributed:*

- *$\langle \mathcal{P}, \mathcal{V}^* \rangle(x)$ (i.e., the output of the interactive machine $\mathcal{V}^*$ after communicating with the interactive machine $\mathcal{P}$ on common input x)*
- *$\mathcal{S}^*(x)$ (i.e., the output of machine $\mathcal{S}^*$ on input x)*

Machine $\mathcal{S}^*$ is called a simulator for the interaction of $\mathcal{V}^*$ with $\mathcal{P}$.

Definition 2.9.2. *(Zero-knowledge [GMR89] version of MIP [BOGKW88]): Consider $(\mathcal{P}_1, \cdots, \mathcal{P}_k, \mathcal{V})$ a k-prover interactive system for language L. We call $(\mathcal{P}_1, \cdots, \mathcal{P}_k, \mathcal{V})$ to be perfect zero-knowledge if $\forall$ probabilistic polynomial time (PPT) interactive Turing machine $\widetilde{\mathcal{V}}$ $\exists$ a PPT machine $\mathcal{S}^*$ (i.e. the simulator) which has blackbox access to $\widetilde{\mathcal{V}}$ s.t. $\forall$ $x \in L$ the following two random variables are identically distributed:*

- *$\langle \mathcal{P}_1, \cdots, \mathcal{P}_k, \widetilde{\mathcal{V}} \rangle(x)$ (i.e., the output of the interactive machine $\widetilde{\mathcal{V}}$ after communicating with the interactive machines $\mathcal{P}_1, \cdots, \mathcal{P}_k$ on common input x)*
- *$\mathcal{S}^*(x)$ (i.e., the output of machine $\mathcal{S}^*$ on input x)*

For better understanding of the zero-knowledge, we imagine another Turing machine called as a judge that tries to tell apart the probability distribution $\mathcal{S}^*(x)$ and $\langle \mathcal{P}_1, \cdots, \mathcal{P}_k, \widetilde{\mathcal{V}} \rangle(x)$. So we can say that zero-knowledge means that a judge should not be able to distinguish between samples drawn from probability distribution $\mathcal{S}^*(x)$ and $\langle \mathcal{P}_1, \cdots, \mathcal{P}_k, \widetilde{\mathcal{V}} \rangle$(x) with a significant probability. Also, we say that the proof system is perfect zero-knowledge against quantum verifiers if the zero-knowledge condition holds against quantum $\widetilde{\mathcal{V}}$.

Definition 2.9.3. *(Statistical Distance): Let $\mathcal{D}$ and $\mathcal{E}$ be the two distributions on a set Ω. The statistical distance between $\mathcal{D}$ and $\mathcal{E}$ is defined by*

$$\Delta(\mathcal{D}, \mathcal{E}) = \max_{Q \subseteq \Omega} \left| Pr_{\mathcal{D}}(Q) - Pr_{\mathcal{E}}(Q) \right|$$

Definition 2.9.4. *(Statistical Indistinguishability): Two probability ensembles $\{\mathcal{X}_n\}_{n\in\mathbb{N}}$ and $\{\mathcal{Y}_n\}_{n\in\mathbb{N}}$ are known as statistically indistinguishable, if for any positive polynomial $p(\cdot)$, and for all sufficiently large n's, the following holds:*

$$\Delta(\mathcal{X}_n, \mathcal{Y}_n) < \frac{1}{p(n)}$$

Definition 2.9.5. *(Statistical Zero-knowledge): Consider $(\mathcal{P},\ \mathcal{V})$ be an interactive proof system for some language L. We call $(\mathcal{P},\ \mathcal{V})$ to be zero-knowledge if $\forall$ probabilistic polynomial-time interactive machine $\mathcal{V}^*$ $\exists$ a probabilistic polynomial-time algorithm $\mathcal{S}^*$ s.t. $\forall x \in L$ the two ensembles given below are statistically indistinguishable:*

- *$\langle \mathcal{P}, \mathcal{V}^* \rangle(x)$ (i.e., the output of the interaction, on common input x, of interactive machine $\mathcal{V}^*$ with the interactive machine $\mathcal{P}$)*
- *$\mathcal{S}^*(x)$ (i.e., the output of the machine $\mathcal{S}^*$ on input x)*

Machine $\mathcal{S}^*$ is called a simulator for the interaction of $\mathcal{V}^*$ with $\mathcal{P}$. The above definition means that the statistical difference between $\mathcal{S}^*(x)$ and $\langle \mathcal{P}, \mathcal{V}^* \rangle(x)$ is negligible in terms of $|x|$.

2.9.2 Proof of Knowledge

Informally, in proofs of knowledge the prover asserts "knowledge" of some object and not merely its existence. Before giving the formal definition of proofs of knowledge, we'll provide some basic definitions and notations required for defining the proofs of knowledge formally.

Technical Preliminaries

Assume $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation. Let, $R(x) = \{w : (x,w) \in R\}$ and language $L_R = \{x : \exists w \ s.t. \ (x,w) \in R\}$. If $(x,w) \in R$, then we say that w is a witness for membership of x to L_R. We call R polynomially bounded if $\exists$ a polynomial p s.t. $|w| \leq p(|x|)\ \forall (x,w) \in R$. Also, we say R an NP-relation if R is polynomially bounded and also, $\exists$ a polynomial-time algorithm for deciding membership in R.

Definition 2.9.6. *(Message-Specification Function): The function $\mathcal{P}(x,y,r)$, called as the message-specification function of machine $\mathcal{P}$ on common input x, auxiliary input y and random input r, is the message sent by the machine $\mathcal{P}$ on x, y and r, after receiving messages $\overline{m}$ on message tape.*

Definition 2.9.7. *(Knowledge Extractor): A knowledge extractor is an oracle machine with access to the $\mathcal{P}(x,y,r)$ that represent the knowledge of machine $\mathcal{P}$ on common input x, auxiliary input y and random input r. The extractor tries to extract a witness for membership of x (i.e., $w \in R(x)$).*

Definition 2.9.8. *(System of Proofs of Knowledge): Assume R to be a binary relation and κ: $\mathbb{N} \to [0,1]$. An interactive function $\mathcal{V}$ is called a knowledge verifier for the relation R with knowledge error κ if the two conditions given below are satisfied:*

- *Non-triviality: $\exists$ an interactive machine $\mathcal{P}^*$ s.t. $\forall (x,\ y) \in R$ all possible interactions of $\mathcal{V}$ with $\mathcal{P}^*$ on common input x and auxiliary input y are accepting.*

- *Validity (with error κ): $\exists$ a polynomial $q(\cdot) > 0$, a constant $d > 0$ and a probabilistic oracle machine K s.t. for any interactive function $\mathcal{P}^*$, any $x \in L_R$ and $\forall\ y,\ r \in \{0,1\}^*$, it follows that:*

$$Pr[\langle \mathcal{P}^*(x,y,r), \mathcal{V}(x) \rangle = 1] \geq \kappa(|x|) \implies$$
$$Pr[(x,w) \in R : w \leftarrow K^{\mathcal{P}^*(x,y,r)}(x)] \geq \frac{1}{q(|x|)} \left(Pr\left[\langle \mathcal{P}^*(x,y,r), \mathcal{V}(x) \rangle = 1\right] - \kappa(|x|)\right)^d.$$

The oracle machine K is known as a universal knowledge extractor. Informally, the validity condition says that K produces a witness for any $x \in L_R$, by interacting with a $\mathcal{P}^*$, with a success probability polynomially related to the corresponding acceptance probability (of the $\mathcal{V}$ when interacting with $\mathcal{P}^*$ on common input x and when $\mathcal{P}^*$ has auxiliary input y and random input r.)

A system for proofs of knowledge for a relation R is an interactive pair $(\mathcal{P}, \mathcal{V})$ in which $\mathcal{V}$ is a knowledge verifier for R and $\mathcal{P}$ is a machine following the non-triviality condition.

Now, in order to reduce the knowledge error, we can do sequential repetitions of the proof system. The error reduces exponentially with the number of repetitions.

Proposition 2.9.1. *Assume R to be a polynomially bounded relation and $t : \mathbb{N} \to \mathbb{N}$ to be polynomially bounded function. Let $(\mathcal{P}, \mathcal{V})$ be a system for proof of knowledge for R with knowledge error κ. Then, the proof system that we get after repeating $(\mathcal{P}, \mathcal{V})$ sequentially $t(|x|)$ times on common input x is a system for proof of knowledge for R with knowledge error $\kappa^{'}(n) = \kappa(n)^{t(n)}$.*

There has been a lot of examples of proofs of knowledge in the previous literatures like proof of knowledge for Hamiltonian cycles with a knowledge error $\frac{1}{2}$ [Blu86]. But these protocols are mainly of theoretical interest due to their high one round communication cost (i.e., $\Omega(|V|^2)$ in the mentioned case).

Till now, (in definition-2.9.8) we refer to a knowledge extractor with expected running time. But for our main result (chapter-3), we need a more stringent definition of proof of knowledge in which the knowledge extractor is required to run in strict polynomial time, which is as follows:

Definition 2.9.9. *(System of Strong Proofs of Knowledge): Assume R to be a binary relation. An interactive function $\mathcal{V}$ is called a strong knowledge verifier for the relation R if the two conditions given below are satisfied:*

- *Non-triviality: $\exists$ an interactive machine $\mathcal{P}^*$ s.t. $\forall(x, y) \in R$ all possible interactions of $\mathcal{V}$ with $\mathcal{P}^*$ on common input x and auxiliary input y are accepting.*

- *Strong validity :* $\exists$ *a negligible function* $\mu : \mathbb{N} \to [0,1]$*, and a probabilistic (strict) polynomial-time oracle machine* K *s.t. for every interactive function* $\mathcal{P}^*$ *and* $\forall\ x, y, r \in \{0,1\}^*$*, machine* K *satisfies the following condition:*

$$Pr[\langle \mathcal{P}^*(x,y,r), V(x)\rangle = 1] > \mu(|x|) \implies$$
$$Pr[(x,w) \in R : w \leftarrow K^{\mathcal{P}^*(x,y,r)}(x)] \geq 1 - \mu(|x|).$$

The oracle machine K *is known as a strong knowledge extractor.*

A system for strong proofs of knowledge for a relation R is an interactive pair $(\mathcal{P}, \mathcal{V})$ in which $\mathcal{V}$ is a strong knowledge verifier for R and $\mathcal{P}$ is a machine that satisfies the non-triviality condition.

We will now provide the definition of the quantum proof of knowledge [Unr12]. But, before that we will introduce Interactive quantum machine briefly. For detailed explanation of the execution of interactive quantum machines please refer to [Unr12].

Definition 2.9.10. *(Interactive Quantum Machine (M)): Interactive quantum machine can be described as a family of quantum circuits* $(M_{\eta x})_{\eta\in\mathbb{N},x\in\{0,1\}^*}$ *and a family of integers* $(r^M_{\eta x})_{\eta\in\mathbb{N},x\in\{0,1\}^*}$*, where* $M_{\eta x}$ *determines the unitary operation that is performed on quantum registers* S *and* N*, and* $r^M_{\eta x}$ *determines the number of messages sent and received by the machine. Register* S *is for the internal state of machine and register* N *is for sending and receiving messages. Also,* η *is the security parameter and* x *is the classical input.*

Definition 2.9.11. *(Quantum Proof of knowledge): Let* R *be a relation and* κ *be the knowledge error. We call an interactive proof system* $(\mathcal{P}, \mathcal{V})$*, where* $(\mathcal{P}, \mathcal{V})$ *can be quantum machines, quantum extractable for* R *with* κ*, iff there exists a polynomially-bounded function* $q(\cdot) > 0$*, a constant* $d > 0$ *and a quantum-polynomial-time oracle machine*[1] K *such that for any interactive quantum prover* $\mathcal{P}^*$*, any* $x \in \{0,1\}^*$ *and any quantum state* $|\psi\rangle$*, it follows*

[1]execution in superposition is not necessary for the results of [Unr12]

that :

$$Pr[\langle \mathcal{P}^*(x, |\psi\rangle), V(x)\rangle = 1] \geq \kappa(|x|) \implies$$
$$Pr[(x, w) \in R : w \leftarrow K^{P^*(x,|\psi\rangle)}(x)] \geq \frac{1}{q(|x|)} \left(Pr\left[\langle \mathcal{P}^*(x, |\psi\rangle), V(x)\rangle = 1\right] - \kappa(|x|)\right)^d .$$

A quantum proof of knowledge for R with knowledge error κ is a complete quantum extractable proof system for R with knowledge error κ.

(NOTE: In above definition-2.9.11 from [Unr12], auxiliary input y and random input r are not used. But, it does not affect our further analysis as we are not using them.)

For understanding the conditions under which a classical proof of knowledge is a quantum proof of knowledge (i.e, the protocol is secure against malicious quantum prover) according to [Unr12], we need to learn about the following definitions :

Definition 2.9.12. *(Σ-protocol): We call a proof system ($\mathcal{P}$, $\mathcal{V}$) a Σ-protocol iff $\mathcal{P}$ and $\mathcal{V}$ are classical, the communication comprises of three messages com, ch, resp, where ch is uniformly selected from the challenge space set $C_{\eta x}$ that may only depend on the security parameter η and the input x. Also, $\mathcal{V}$ accepts or not based on a deterministic polynomial-time computation on x, com, ch, resp. Furthermore, we need that it should be possible to sample uniformly from $C_{\eta x}$ up to a negligible probability in probabilistic polynomial time and the membership in $C_{\eta x}$ should be decidable in deterministic polynomial time in $\eta + x$.*

Definition 2.9.13. *(Special soundness): We say a Σ-protocol ($\mathcal{P}$, $\mathcal{V}$) for a relation R has special soundness iff there is a deterministic polynomial-time algorithm K_0, known as special extractor, s.t. the following condition holds: For any two accepting conversations (com, ch, resp) and (com, ch′, resp′) for statement x, s.t. ch $\neq$ ch′ and ch, ch′ $\in C_{\eta x}$, It follows that $w := K_0$(x, com, ch, resp, ch′, resp′) satisfies (x, w) $\in R$. The special extractor is not a normal extractor as it requires two accepting conversations instead of one.*

Definition 2.9.14. *(Strict soundness): We say a Σ-protocol ($\mathcal{P}$, $\mathcal{V}$) has strict soundness iff for any two accepting conversations (com, ch, resp) and (com, ch, resp′) for statement x, we have resp = resp′.*

A set of constraints under which a classical proof of knowledge is a quantum proof of knowledge [Unr12] are:

- The proof system's protocol should be in the form of Σ-protocol (Definition 2.9.12), in which there are three messages (commitment, challenge, and response) with a public coin verifier.

- Proof system should have special soundness (Definition 2.9.13), i.e, given two accepting conversations between $\mathcal{P}$ and $\mathcal{V}$ with same commitment but different challenges, we can efficiently calculate a witness.

- Proof system should have strict soundness (Definition 2.9.14), which means that given the commitment and the challenge of a conversation, there is at most one response that would make $\mathcal{V}$ accept. This condition is required to make sure that the response provided by $\mathcal{P}$ does not comprise too much information; measuring it will then not perturb the state of $\mathcal{P}$ too much.

For example, in [Unr12] Unruh's shows a construction of quantum proof of knowledge for Hamiltonian cycles from the classical proof of knowledge provided in [Blu86].

2.10 Review of Practical Relativistic Zero-Knowledge for NP Protocol [CMS$^+$19]

In this section, we summarize the main results from the [CMS$^+$19]. The findings of this paper serve as the basis for our new protocol (see Protocol-3.1.2) provided in the main result section.

2.10.1 Summary of the protocols given in [CMS$^+$19]

These protocols are Zero-knowledge protocols for the 3-COLourability problem (described in 2.6) that use two (local) provers. But, before explaining the protocols, we need to learn about classical and quantum value of interactive proofs.

Definition 2.10.1. ***Provers limited to local operations*** *means: the provers can apply functions on their own input with a pre-shared randomness.*

Definition 2.10.2. ***Classical value of k-prover interactive proof system*** $\Pi(x)$*, denoted by* $\omega(\Pi(x))$*, is the minimum value of parameter* $q(|x|)$*, s.t.* $\forall\, \mathcal{P}'_1, ..., \mathcal{P}'_k$, $Pr([\mathcal{P}'_1, ..., \mathcal{P}'_k, \mathcal{V}](x) = accept$ $\leq q(|x|)$ *when* $x \notin L$*. Here, the provers are limited to local operations during the execution of the protocol.*

Commitment Scheme

In the protocol, provers use $w = b_i \cdot r + c_i$ to commit to a colour c_i of a vertex $i \in V$, of graph $G = (V, E)$, asked by the verifier. The b_i is a specific random mask that is shared between provers before the start of the protocol and the r represents the randomness provided by the verifier while asking the query. Only the following three scenarios are possible for such commitment schemes, depending on the verifier's queries (see below for details):

1. (forever hiding): The verifier does not learn colours if the queries have no common vertex [i.e. $i \neq i'$, $j \neq j'$ in commit phase [2] below].

2. (consistency testing): The verifier can verify that the answers of the provers agree with each other when they both are given at least a common vertex and a common randomness [i.e. at least one of $i = i'$ or $j = j'$ is same and $r = r'$ in commit phase [2] below].

3. (implicit unveiling): The verifier can learn the entire colouring of one edge when queries with particular parameters are provided to the provers [i.e. both $i = i'$ and $j = j'$ are same and $r \neq r'$ in commit phase [2] below].

Perfect Zero-knowledge Two-Prover Protocol

1. The provers $\mathcal{P}_1$ and $\mathcal{P}_2$ start by agreeing on random masks b_i for each vertex and also a 3 Colouring for the graph $G = (V, E)$.

2. Commit Phase

   - Then the verifier chooses and sends queries $q_1 = (i, j, r)$ to $\mathcal{P}_1$ and $q_2 = (i', j', r')$ to $\mathcal{P}_2$.
   - Then $\mathcal{P}_1$ commits to the colours of vertices asked in the queries q_1 and $\mathcal{P}_2$ commits to colours of vertices asked in q_2 using the commitment scheme described in 2.10.1.

3. Check Phase: In this phase, the verifier checks whether the provers pass any one of the following tests, based on their responses.

   - Edge-Verification Test: In this test, the verifier checks whether the edge asked in the commit phase has end vertices of different colours, i.e., the graph three colouring condition is followed.
   - Well-definition Test: In this test, verifier checks whether the provers' responses for the queries are consistent on at least one vertex.

Theorem 2.10.1. *The two-prover interactive proof system (described in 2.10.1) is perfectly complete with classical value (ω) less than equal to 1 - $\frac{1}{9|E|}$ upon any graph $G = (V, E) \notin$ 3COL [CMS+19].*

The direct consequence of theorem-2.10.1 is that, we need $\Omega(|E|)$ sequential repetitions of protocol-2.10.1 to produce an interactive proof system for 3COL with negligible soundness.

Theorem 2.10.2. *The two-prover interactive proof system (described in 2.10.1) is perfect zero-knowledge [CMS+19].*

Protocol : Relativistic zero-knowledge protocol for Hamiltonian Cycle [CL17]

Input - The provers and the verifiers are provided a graph $G = (V, E)$.

Auxiliary Input - The provers $\mathcal{P}_1$ and $\mathcal{P}_2$ know a Hamiltonian cycle $\mathcal{C}$ of G.

Preprocessing - $\mathcal{P}_1$ and $\mathcal{P}_2$ pre-agree on a random permutation $\Pi : V \to V$ and a $n \times n$ matrix $A \in \mathcal{M}_n^{\mathbb{F}_Q}$ in which each element of A is selected uniformly at random in $\mathbb{F}_Q$. Here, n, $\mathbb{F}_Q$ and $\mathcal{M}_n^{\mathbb{F}_Q}$ represents the number of vertices in G, a field for a large prime power Q and the set of matrices of size $n \times n$ having elements in the field $\mathbb{F}_Q$ respectively.

1. Commit to each bit of adjacency matrix $M_{\Pi(G)}$ of $\Pi(G)$ as follows:
   - $\mathcal{V}_1$ shares a matrix $B \in \mathcal{M}_n^{\mathbb{F}_Q}$ in which each element of B is selected uniformly at random in $\mathbb{F}_Q$.
   - $\mathcal{P}_1$ outputs the matrix $Y \in \mathcal{M}_n^{\mathbb{F}_Q}$ where $\forall i, j \in [n]$, we have $Y_{i,j} = A_{i,j} + (B_{i,j} * (M_{\Pi(G)})_{i,j})$.
2. The verifier then sends a challenge (a random bit) $chall \in \{0, 1\}$ to the prover.
3. According to challenge value, we have:
   - If $chall = 0$, $\mathcal{P}_2$ sends all the elements of A to $\mathcal{V}_2$ and reveals Π, i.e. decommits to all the elements of $M_{\Pi(G)}$.
   - If $chall = 1$, $\mathcal{P}_2$ sends, $\forall$ edges (u, v) of $\mathcal{C}'$, $A_{u,v}$ as well as $\mathcal{C}'$. So, basically the $\mathcal{P}_2$ reveals the bit, having value 1, of the adjacency matrix that corresponds to the Hamiltonian cycle $\mathcal{C}'$ of $\Pi(G)$.
4. The verifier then verifies that the received decommitments are valid and relates to what provers have declared. He also verifies the correctness of the timing constraint of the bit commitment. So,

- if $chall = 0$, the prover's revealing A must satisfy $\forall i, j \in [n]$, $Y_{i,j} = A_{i,j} + (B_{i,j} * (M_{\Pi(G)})_{i,j})$.
- if $chall = 1$, the the prover's revealing A must satisfy $\forall (u, v) \in \mathcal{C}'$, $Y_{u,v} = A_{u,v} + B_{u,v}$.

2.10.2 How are protocols given in section-2.10.1 [CMS$^+$19] useful?

In this section, we show that the relativistic zero-knowledge protocols given in section-2.10.1 ([CMS$^+$19]) are very efficient compared to the previous similar protocols like relativistic 2-provers zero-knowledge protocol for Hamiltonian cycle protocol [CL17] (shown in 2.10.1). We use Hamiltonian cycle protocol given in [CL17] for comparison because it follows Unruh's [Unr12] proof of knowledge definition only with one difference, i.e. the authors use two-prover relativistic commitments instead of computational commitments. Also, in general the protocols that are proofs of knowledge are better compared to protocols that are proofs of membership as the proof of knowledge protocols can be used for identification of a user. A natural way to determine a person's identity is to ask him/her to supply a proof of knowledge of a fact that the person is supposed to know.

Now, we will shift our focus back to efficiency of protocol given in 2.10.1 compared to protocol-2.10.1. In protocol-2.10.1 having soundness error nearly $\frac{1}{2}$, for a graph of $|V|$ vertices an approximate $200|V|^2$ bits of communication is required before the verifier can announce his choice challenge, i.e. *chall*. $200|V|^2$ bits of communication is needed due to the transfer of adjacency matrices B and Y, of size $|V| \times |V|$ each, having elements in the field $\mathbb{F}_Q$ where Q is a large prime power [**REMARK**: If $Q = 64|V|!2^{3k}$ as suggested by the authors then for any cheating prover the verifier accepts with at most $\frac{1}{2} + 2^{-k}$ probability]. Therefore, for the implementation of this protocol either the separation between the provers or the communication speed between prover-verifier pairs have to be very high. Whereas, for 2-prover protocol given in 2.10.1 the provers have to communicate two trits each after

receiving an edge and two bits each from the verifier. And, to reduce the soundness error to nearly $\frac{1}{2}$ we need $|E|$, i.e., the number of edges, sequential repetitions of the same basic protocol. Therefore, this makes the later protocol much more practical as it can work for provers with short separation.

Chapter 3

MAIN RESULT

In this section, we provide the main results of our thesis. First, we provide bit commitment scheme used in [ABC$^+$20] and show that it is sound against classical provers. Then, we show that this protocol is also a proof of knowledge. Further, we show that our proof of knowledge doesn't follow the approach to demonstrate it is a quantum proof of knowledge (given in [Unr12]). Additionally, we show that the protocol satisfies a stronger zero-knowledge notion in a scenario of 2 provers, 2 verifiers and 2 judges by providing a no-signalling simulator.

3.1 Analysis and Proof of Knowledge of [ABC$^+$20]

The analysis of the protocol in this section is very similar to the one provided in [CMS$^+$19]. Firstly, we will start by explaining how verifiers select questions for the two provers.

3.1.1 Distribution of questions

In this section, we provide the probability distribution ($\mathcal{D}_G$) of the $\mathcal{V}$s' questions in the protocol $\Pi_{\text{lhv}}[G]$ (described in 3.1.2), where $G = (V, E)$ is a connected undirected graph. $\mathcal{D}_G$ comprises of one edge and one bit for each prover. Upon graph $G = (V, E)$, (e, b) and (e', b') are the questions asked to the provers $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. Also, the $\mathcal{V}$s never ask two

non-intersecting edges to the two different provers, i.e $p_{\mathcal{D}_G}(e, b, e', b') = 0$ if $e \cap e' = \emptyset$ and arbitrary b and b'. $e \cap e'$ means the common vertices between the two edges.

Firstly, an edge $e = (i, j) \in E$ and a variable $b \in \mathbb{F}_2$ are chosen uniformly at random. Then, with probability ϵ we perform edge verification test, in which we set $e' = e$ and $b' = \overline{b}$. Finally, with probability 1 - ϵ, we execute the well definition test. In well definition test, the second edge e' is chosen uniformly at random from the set of edges having vertex i with probability $\frac{1}{2}$ and from the set of edges having vertex j with probability $\frac{1}{2}$. Also, the variable b' is set to b. So, It follows that for well definition test, in which $e = (i, j) \in E$ and $e' \in \mathsf{Edges}(e) = \mathsf{Edges}(i) \cup \mathsf{Edges}(j)$, we have:

$$p_{\mathcal{D}_G}(e, b, e', b) = \frac{1-\epsilon}{4|E|}\left(\frac{|\{e'\} \cap \mathsf{Edges}(i)|}{|\mathsf{Edges}(i)|} + \frac{|\{e'\} \cap \mathsf{Edges}(j)|}{|\mathsf{Edges}(j)|}\right) \ , \tag{3.1}$$

Also, for edge verification test, we have:

$$p_{\mathcal{D}_G}(e, b, e, \overline{b}) = \frac{\epsilon}{2|E|} + \frac{1-\epsilon}{4|E|}\left(\frac{1}{|\mathsf{Edges}(i)|} + \frac{1}{|\mathsf{Edges}(j)|}\right) \geq \frac{\epsilon}{2|E|} \ , \tag{3.2}$$

3.1.2 The Protocol

Now, we provide the protocol given in [ABC+20] that is slightly simpler than the protocol described in [CMS+19] by using simpler commitments. In this protocol ($\Pi_{\text{lhv}}[G]$), we have an interactive system of two verifiers ($\mathcal{V}_1$ and $\mathcal{V}_2$) and two provers ($\mathcal{P}_1$ and $\mathcal{P}_2$), represented by $\mathcal{V}$ and $\mathcal{P}$ respectively. First of all, both $\mathcal{P}_1$ and $\mathcal{P}_2$ agree on a random 3-colouring of G and then for each round they randomly permute the three colors of the original 3-colouring. Let the colouring for each round be $c_i(n) \in \{0,\ 1,\ 2\}$ for every $i \in V$ and n identifying rounds. They also choose ℓ^0 and ℓ^1 at random in $\mathbb{F}_3$ for each $i \in V$ such that $\ell_i^0 + \ell_i^1 \equiv c_i \pmod 3$ holds. Note that, the labellings ℓ^0 and ℓ^1 can have same values even for adjacent vertices. Lastly, dependence on n is ignored for simplicity in the description of the protocol given below.

Protocol $\Pi_{\text{lhv}}[G]$: Two-prover, 3-COL [ABC$^+$20]

$\mathcal{P}_1$ and $\mathcal{P}_2$ pre-agree on a random 3-colouring of G: $\{(i, c_i)|c_i \in \mathbb{F}_3\}_{i\in V}$ such that $(i, j) \in E \implies c_j \neq c_i$. Also, they randomly select labellings ℓ_i^0 and ℓ_i^1 for $i \in V$ such that $\ell_i^0 + \ell_i^1 \equiv c_i \pmod 3$ holds.

Commit phase:

- $\mathcal{V}$ picks $(((i, j), b), ((i', j'), b')) \in_{\mathcal{D}_G} (E \times \mathbb{F}_2)^2$.
- $\mathcal{V}_1$ sends $((i, j), b)$ to $\mathcal{P}_1$ and $\mathcal{V}_2$ sends $((i', j'), b')$ to $\mathcal{P}_2$.
- If $(i, j) \in E$ and $b \in \mathbb{F}_2$ then $\mathcal{P}_1$ replies $a_i = \ell_i^b$ and $a_j = \ell_j^b$.
- If $(i', j') \in E$ and $b' \in \mathbb{F}_2$ then $\mathcal{P}_2$ replies $a'_{i'} = \ell_{i'}^{b'}$ and $a'_{j'} = \ell_{j'}^{b'}$.

Check phase:

Edge-Verification Test:

- if $(i, j) = (i', j')$ and $b' \neq b$ then $\mathcal{V}$ accept iff $a_i + a'_i \neq a_j + a'_j$.

Well-Definition Test:

- If $(i, j) = (i', j')$ and $b' = b$ then $\mathcal{V}$ accepts iff $(a_i = a'_i) \wedge (a_j = a'_j)$.
- if $(i, j) \cap (i', j') = i$ and $b' = b$ then $\mathcal{V}$ accepts iff $a_i = a'_i$.
- If $(i, j) \cap (i', j') = j$ and $b' = b$ then $\mathcal{V}$ accepts iff $a_j = a'_j$.

$\Pi_{\text{lhv}}[G]$ clearly satisfies perfect completeness. Also, the following theorem proves that $\Pi_{\text{lhv}}[G]$ is sound against classical provers.

Theorem 3.1.1. *The two-prover interactive proof system* Π_{lhv} *(Protocol-3.1.2) is perfectly complete with classical value* $\omega(\Pi_{\text{lhv}}[G]) \leq 1 - \frac{1}{5|E|}$ *upon any graph* $G = (V, E) \notin \mathsf{3COL}$.

Proof. Consider $G \notin \mathsf{3COL}$ and δ be the probability that the verifiers ($\mathcal{V}_1$ and $\mathcal{V}_2$) uncover an error in the check phase of the protocol when communicating with two local dishonest provers ($\widetilde{\mathcal{P}_1}$ and $\widetilde{\mathcal{P}_2}$). It follows that during the protocol, the two provers can neither interact directly with each other due to the distance between them nor indirectly via the verifiers as the verifiers' questions are independent of provers' responses. Hence, we can assume the strategy of the two provers to be deterministic without harming the soundness error [GO94] by allowing both $\mathcal{P}$s to choose answers that maximize their success probability given questions. Therefore, assume a deterministic strategy represented by a pair of arrays $W^\ell[i, j, b] \in \mathbb{F}_3^2$ which will be used by prover $\widetilde{\mathcal{P}}_\ell$ for $\ell \in \{1, 2\}$ (note: V always presents the question $((i, j), b) \in E \times \mathbb{F}_2$ to the provers in the order $i < j$.) In output pair $W^\ell[\cdot, \cdot, \cdot]$, $W^\ell_z[\cdot, \cdot, \cdot]$ represents it's z-component, where z $\in$ {1, 2}. We consider $W[i, b]$ for [i, b] $\in$ $E \times \mathbb{F}_2$ to be well-defined if for all j, k such that $(i, j), (i, k) \in E$, one of the following 4 equalities is correct based on conditions j > i or j < i, k > i or k < i

$$W_1^1[i, j, b] = W_1^2[i, k, b] = W_2^1[j, i, b], \text{ or } W_1^1[i, j, b] = W_2^2[k, i, b] = W_2^1[j, i, b] \tag{3.3}$$

So, if for all $i \in V$ and $b \in \mathbb{F}_2$, $W[i, b]$ is well defined, then we consider W to be well defined.

Now, we calculate the least probability $\delta_{\text{wdt}} > 0$ with which the well definition test will find an error, if $W[i, b]$ is not well defined for some $i \in V$ and $b \in \mathbb{F}_2$. Since (3.3) is unsatisfied, let $W_1^1[i, j, b] \neq W_1^2[i, k, b]$ for some $(i, j), (i, k) \in E$. Also, the other 3 cases are treated in a similar way. Consider $e = (i, j)$ and $e' = (i, k)$ be the two edges. From (3.1) we have that the well-definition test will uncover an error with probability

$$p_{\mathcal{D}_G}(e, b, e', b) \geq \frac{1 - \epsilon}{4|E||Edges(i)|}$$

But, we notice that we can detect whether $W[i, b]$ is not well defined, in at least $|Edges(i)|$ places. Let any $(i, m) \in E$, such that $m >$ i (The other case of $m < i$ is treated in a similar

way). Also, it is clear that one of the following three conditions have to be correct:

$$W_1^1[i,j,b] \neq W_1^2[i,m,b], W_1^1[i,m,b] \neq W_1^2[i,m,b], \text{ or } W_1^1[i,m,b] \neq W_1^2[i,k,b]. \tag{3.4}$$

Since, $W[i,b]$ is not well defined and $\mathcal{V}$s have $|\mathsf{Edges}(i)|$ places to catch the provers and each one of these is selected with a minimum probability of $\frac{1-\epsilon}{4|E||\mathsf{Edges}(i)|}$. So,

$$\delta_{\text{wdt}} \geq \frac{(1-\epsilon)\cdot|\mathsf{Edges}(i)|}{4|E|\cdot|\mathsf{Edges}(i)|} = \frac{1-\epsilon}{4|E|} \ . \tag{3.5}$$

Now, consider that W is well defined and the answers returned by the provers are consistent. As we discussed earlier, when the $\mathcal{P}$s' return consistent values, then we can calculate the colours as $c_i := W[i,b] + W[i,\bar{b}]$ for $b \in \mathbb{F}_2$. Since, we know that $G \notin 3COL$, therefore, at least one edge (i^*, j^*) would have both its vertices of the same colour. So, when the same edge (i^*, j^*) along with $b \in \mathbb{F}_2$ and $\bar{b} \in \mathbb{F}_2$ is announced to provers $\widetilde{\mathcal{P}_1}$ and $\widetilde{\mathcal{P}_2}$ respectively, the edge-verification test will detect it with the following probability:

$$\delta_{\text{evt}} \geq \sum_b \min_{e\in E}(p'_{\mathcal{D}_G}(e,b,e,\bar{b})) \geq \frac{\epsilon}{|E|} \ .$$

Hence, δ of any deterministic strategy for $G \notin 3COL$ follows:

$$\delta \geq \min(\delta_{\text{wdt}}, \delta_{\text{evt}}) \geq \frac{1}{5|E|} \ \text{(maximized at } \epsilon = 1/5) \ .$$

The above result satisfies the classical value of the game $\omega(\Pi_{\text{lhv}}[G]) \leq 1-\delta$. □

The direct consequence of above theorem-3.1.1 is that, we require $\Omega(|E|)$ sequential repetitions of protocol-3.1.2 to produce an interactive proof system for 3COL with negligible soundness error. Therefore, the protocol-3.1.2 is sound against classical provers.

3.1.3 Proof of knowledge

In this section, we show that our protocol $\Pi_{\text{lhv}}[G]$ is a proof of knowledge, based on the proof of knowledge definition-2.9.9 .

Consider δ to be the maximum probability with which the dishonest $\mathcal{P}$s can pass the check phase even when their tables don't contain a 3-colouring. Let $W^{\ell}[i, j, b](r)$, for a particular provided randomness r, be the table that define the complete deterministic behaviour of the prover $\mathcal{P}_{\ell}$, where $\ell \in \{1, 2\}$. So, the $\mathcal{P}$s answer $\mathcal{V}$s' queries according to their tables $W^{\ell}[i, j, b](r)$. But, in case they don't wish to respond, they can send ϵ value. Now, in order to calculate δ, we first compute the probability of catching dishonest $\mathcal{P}$s while they try to convince the honest $\mathcal{V}$s that their tables contain a 3-colouring of the undirected graph G = (V, E). There are 2 ways for $\mathcal{V}$s to catch the dishonest $\mathcal{P}$s: (a) if the graph is not properly 3-coloured (b) if their tables have inconsistencies. First in edge verification test, there has to be at least one edge, let it be (i', j'), that would have same coloured vertices on both its ends so, i.e, $W_1^1[i', j', b](r) + W_1^2[i', j', \overline{b}](r) = W_2^1[i', j', b](r) + W_2^2[i', j', \overline{b}](r)$ holds. Hence, the probability of $\mathcal{V}$s sending that edge to the $\mathcal{P}$s during edge verification test is $\frac{1}{2|E|}$, where $|E|$ represents the total number of edges in G. Therefore, the maximum probability of dishonest $\mathcal{P}$s winning the edge verification test is $\left(1 - \frac{1}{2|E|}\right)$. For the second case, that is the well definition test, the minimum probability of catching dishonest $\mathcal{P}$s is $\frac{1}{4|E|}$ (From (3.5)), which means at least one of $W_1^1[i', \cdot, b](r) \neq W_1^2[i', \cdot, b](r)$ or $W_2^1[\cdot, i', b](r) \neq W_1^2[i', \cdot, b](r)$ or $W_1^1[i', \cdot, b](r) \neq W_2^2[\cdot, i', b](r)$ or $W_2^1[\cdot, i', b](r) \neq W_2^2[\cdot, i', b](r)$ holds for vertex i'. So, the maximum probability for dishonest $\mathcal{P}$s to pass the well-definition test is $\left(1 - \frac{1}{4|E|}\right)$.

Since, we are computing the maximum probability δ of dishonest $\mathcal{P}$s passing the check phase even when having inconsistent tables or badly coloured edges, we have:

$$\delta = max\left(\left(1 - \frac{1}{2|E|}\right), \left(1 - \frac{1}{4|E|}\right)\right) = \left(1 - \frac{1}{4|E|}\right) \tag{3.6}$$

Protocol : Strong Knowledge Extractor for 3-COL protocol

Let x be the input, w be the witness, K be the knowledge extractor and r be the randomness of the provers. Now, to extract the witness (3COL), K runs different copies of the basic protocol, (provided in 3.1.2), $[Pro_1, Pro_2, \cdots, Pro_n]$, where $n \leq 5|E|\ell$, sequentially using randomness r_i from the same provided r and perform the following steps during each copy i:

1. It asks all the possible questions, which are $(((i,j),b),((i',j'),b')) \in_{\mathcal{D}_G} (E \times \mathbb{F}_2)^2$, to $\mathcal{P}_1$ and $\mathcal{P}_2$ to obtain the whole deterministic behaviour of the provers in the context of $Pro_{1..i}$.
2. If both provers $\mathcal{P}_1$ and $\mathcal{P}_2$ answer $(a_i, a_j, a'_{i'}, a'_{j'})$ all the possible questions correctly from well definition and edge verification tests then K can extract the three colouring. Otherwise, K chooses a pair of questions (q_i, q'_i) for $(\mathcal{P}_1, \mathcal{P}_2)$ and get answers (a_i, a'_i) from $(W_i^1[\cdot,\cdot,\cdot](r_i), W_i^2[\cdot,\cdot,\cdot](r_i))$.
3. Then, K moves to the next copy of the protocol, i.e, Pro_{i+1}.

Proposition 3.1.1. *For knowledge error κ greater than $3^{-|V|}$, for all provers there exist trivial extractors that can extract a 3-colouring of G without even interacting with provers.*

Proof. Such trivial extractors can generate a 3-colouring by randomly choosing a colouring for G using only 3 colours and then checking whether the end points of every edge are indeed of different colours. □

From proposition-3.1.1, it is safe to assume that κ is never greater than $3^{-|V|}$. Now, for proving that proof of knowledge method follows the definitions-2.9.9, we need to compute κ and $p(G, y, r)$.

Let $\kappa(\ell) = 2^{-\ell}$ with $\kappa(\ell) \leq 3^{-|V|}$. So, for $L > 8|E|\ell$, we have :

$$\begin{aligned} \left(1 - \frac{1}{4|E|}\right)^L &\leq \frac{1}{e^{2\ell}} \\ &< \frac{1}{e^{\ell}} \\ &< \frac{1}{2^{\ell}} \end{aligned} \tag{3.7}$$

where $\left(1 - \frac{1}{4|E|}\right)^L$ is the probability of $\mathcal{V}$s accepting the $\mathcal{P}$s, when $\mathcal{P}$s never use a 3-colouring.

Using the constraints given above, we now compute the bounds on ℓ,

$$2^{-\ell} \leq 3^{-|V|} \implies 3^{|V|} \leq 2^{\ell} \implies |V| \log_2(3) \leq \ell \tag{3.8}$$

Considering $p(G, y, r)$ be the average probability of $\mathcal{V}$s accepting $\mathcal{P}$s, given graph G, 3-colouring y and randomness r, it follows:

$$p(G, y, r) = \sum_{|s|} \frac{1}{|s|} p(G, y, r, s) \tag{3.9}$$

where $p(G, y, r, s) \in \{0, 1\}$ is the probability of $\mathcal{V}$s accepting $\mathcal{P}$s during one iteration given graph G, 3-colouring y, randomness r and question s. Also, $|s| \geq (4|E| + 2|E|)^L$, because $|s|$ represent the set of questions from which the pair of questions are chosen where $4|E|$ and $2|E|$ are the maximum possible questions that the verifiers can ask the provers in case of well-definition test and edge-verification test respectively.

Using 3.7 and 3.9, for $p(G, y, r) \geq \kappa(\ell)$, we show that :

$$Pr[K_{p(G,y,r,s)} \in R(G)] \geq \frac{1}{2} \tag{3.10}$$

i.e., for $p(G, y, r) \geq \kappa(\ell)$ the $\mathcal{P}$s tables must give an actual 3-colouring at least $4|E|\ell$ times, where $|V| \log_2(3) \leq \ell$, i.e, $\frac{1}{2}$ times of the total number of iterations L, s.t., $L > 8|E|\ell$. Because for the $p(G, y, r)$ to be greater than κ at least half the rounds the probability of

prover getting accepted by verifier has to be 1, i.e. $p(G, y, r) \geq \left(1 - \frac{1}{4|E|}\right)^{L/2} \times 1^{L/2}$, as the maximum probability for dishonest provers to pass the test without actually using a 3-colouring is $\left(1 - \frac{1}{4|E|}\right)$.

From the above discussion, we can conclude that if $\mathcal{P}$s use three colouring at least half the time, therefore $p(G, y, r) > \kappa(x)$, then K would be able to extract a 3-colouring from the $\mathcal{P}$s tables with probability 1 which is greater than 1 - $\kappa(x)$, given it runs for strict polynomial time. Therefore, our protocol follows the proof of knowledge definitions-2.9.9.

3.1.4 Why our Proof of Knowledge does not follow the Unruh's Quantum Proof of Knowledge [Unr12]?

We now reason why our protocol's classical PoK doesn't follow the [Unr12]'s methodology for proving a QPoK (summarised in section-2.9.2):

- Our protocol doesn't follow the definition of Σ-Protocol. As, in our case, there are no explicit commitments provided to the $\mathcal{V}$s by the $\mathcal{P}$s. The $\mathcal{V}$s directly provide the challenges to the $\mathcal{P}$s in the form of $((i, j), b)$, where $(i, j) \in E$ and $b \in \mathbb{F}_2$, and get the respective responses (ℓ_i^b, ℓ_j^b). So, our protocol follows a system of (challenge, response).

- We cannot extract colour of more than two vertices during one iteration of our protocol. So, we need more than two accepting conversations to extract the witness. Hence, our protocol doesn't have special soundness.

- Our protocol doesn't follow the strict soundness requirement, because of the following reasons:

  - In well definition test, if the response value(s), by the provers, is(are) same for the intersecting vertex(s). Then, we have an accepting conversation. Therefore, there are more than one accepting responses for the same challenge.

– In edge definition test, as long as the colours of the vertices in the challenge edge (responded by $\mathcal{P}$s) are different, $\mathcal{V}$s accepts that. Therefore, there are more than one response that leads to an accepting conversation for the same challenge.

From the above discussion, we can conclude that our protocol doesn't follow the technique used by Unruh to show that a protocol is a QPoK [Unr12]. Therefore, the proof technique used by Unruh to show a proof of knowledge to be a quantum proof of knowledge has to be a more generic. Since, it requires special conditions which need not be satisfied by all protocols. We leave this as an open question.

3.2 No signalling simulation for the protocol $\Pi_{\text{lhv}}[G]$ (protocol-3.1.2) [ABC$^+$20]

In this section, we show that in multi-prover zero-knowledge protocol, shown in 3.1.2, for 3-Colourability problem the zero-knowledge property is attained by a pair of no-signalling simulators. Therefore, just like no-signalling simulators, the judges can be separated in such a way that the verifiers cannot communicate but still can simulate as they are no-signalling. Such judges would be able to make a difference between signalling simulators and no-signalling verifiers. So, we have a stronger notion of zero-knowledge in this new protocol, as in usual protocols the simulators are allowed signalling. Also, we provide another definition of zero-knowledge proofs, in terms of two provers, two verifiers and two judges, with stronger notion of zero-knowledge as it uses two no-signalling simulators. The definition is as follows:

Definition 3.2.1. *(Strong Zero-Knowledge using No-signalling Simulators): Assume* ($\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{V}_1, \mathcal{V}_2$) *be a 2-prover 2-verifier interactive system for language L. We call* ($\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{V}_1$, $\mathcal{V}_2$) *to be strongly perfect zero-knowledge if* $\forall$ *no-signalling probabilistic polynomial time (PPT) interactive Turing machines* $\widetilde{\mathcal{V}_1}$ *and* $\widetilde{\mathcal{V}_2}$ $\exists$ *no-signalling-PPT machines* $\mathcal{S}_1^*$ *and* $\mathcal{S}_2^*$ *(i.e. the simulators), which have blackbox access to* $\widetilde{\mathcal{V}_1}$ *and* $\widetilde{\mathcal{V}_2}$ *respectively, s.t.* $\forall$ *input* $x \in L$ *and* $\forall$ *auxiliary inputs* z_1, z_2 *provided by judges the following conditions are satisfied:*

The following two random variables are identically distributed:

- $\langle\langle \mathcal{P}_1, \widetilde{\mathcal{V}_1}\rangle(x, z_1),\ \langle \mathcal{P}_2, \widetilde{\mathcal{V}_2}\rangle(x, z_2)\rangle$ *(i.e. the joint distribution of the output of the interactive machine* $\widetilde{\mathcal{V}_1}$ *after communicating with interactive machine* $\mathcal{P}_1$ *on common input* x *and auxiliary input* z_1 *and the output of the interactive machine* $\widetilde{\mathcal{V}_2}$ *after communicating with interactive machine* $\mathcal{P}_2$ *on common input* x *and auxiliary input* z_2*)*

- $\langle \mathcal{S}_1^*(x, z_1),\ \mathcal{S}_2^*(x, z_2)\rangle$ *(i.e. the joint distribution of the output of machine* $\mathcal{S}_1^*$ *on common input* x *and auxiliary input* z_1 *and the output of machine* $\mathcal{S}_2^*$ *on common input* x *and auxiliary input* z_2*)*

Informally, the above definition-3.2.1 suggests that the pair of no-signalling simulators is trying to simulate the interaction between both (prover, verifier) pairs. So, like the simulators the judges can also be separated such that the no-signalling verifiers can perform the protocol.

Firstly, we informally describe the PR box simulation of the protocol, shown in 3.1.2. The simulators ($\mathcal{S}_1^*$ and $\mathcal{S}_2^*$) perform the simulation in two steps. In first step of the simulation, the simulators try to figure out the relation between vertex nodes received from the verifiers, i.e. whether the nodes receive are identical, using only PR boxes and local operations. Therefore at the end of the first step, the simulators generate four boolean pairs $\langle (z_{00}, z'_{00}), (z_{01}, z'_{01}), (z_{10}, z'_{10}), (z_{11}, z'_{11})\rangle$ for each of the four different possible relations between vertex nodes: $i = i'$, $i = j'$, $j = i'$, $j = j'$, i.e. if a pair has two unequal values then its corresponding vertices are equal. Then in the second step of the simulation the simulators generate the commitments $(\ell_0, \ell_1, \ell'_0, \ell'_1)$, using a no-signalling box (proved in theorem-3.2.1) that we call 3-COLZK box (shown in 3.1), for the verifiers ($\mathcal{V}_1^*$ and $\mathcal{V}_2^*$) based on the outputs of the first step and the inputs from the verifiers such that the commitments follow all the constraints described in the protocol (shown in 3.1.2). That is, for well-definition test, one or more ℓ_i would have same value(s) as ℓ'_j where $i, j \in \mathbb{F}_2$. Similarly, the commitments for edge-verification test would follow the property of proper 3-colouring that two vertices of an edge have different colours.

Theorem 3.2.1. *3-COLZK box is no-signalling*

Proof. Given: 3-COLZK box has inputs $\langle b, b\rangle$ from the verifiers and $\langle (z_{00}, z'_{00})\ , (z_{01}, z'_{01}),$ $(z_{10}, z'_{10}), (z_{11}, z'_{11})\rangle$ representing the vertices relationships.

1. The 3-COL box takes input $(z_{00}, z_{01}, z_{10}, z_{11}, b)$ from $\mathcal{S}_1^*$ and generate outputs (ℓ_0, ℓ_1) uniformly at random.

2. Then, it takes input $(z'_{00}, z'_{01}, z'_{10}, z'_{11}, b')$ from $\mathcal{S}_2^*$ and generate (ℓ'_0, ℓ'_1) based on the following constraints:

   - (Well definition test): $\forall u, v, b, b' \in \{0, 1\}$ $(b = b' \wedge z_{uv} \neq z'_{uv}) \rightarrow \ell'_v = \ell_u$
   - (Edge verification test): $\forall u, v, b, b' \in \{0, 1\}$ $(b \neq b' \wedge z_{uv} \neq z'_{uv} \wedge z_{\bar{u}\bar{v}} \neq z'_{\bar{u}\bar{v}}) \rightarrow \ell_u$ $+ \ell'_v \neq \ell_{\bar{u}} + \ell'_{\bar{v}}$ uniformly among solutions, conditioned on $\forall u, v \in \{0, 1\}$ $(z_{uv} \neq z'_{uv}) \wedge (z_{u\bar{v}} = z'_{u\bar{v}}) = (z_{uv} \neq z'_{uv}) \wedge (z_{\bar{u}v} = z'_{\bar{u}v}) = 0$ meaning that only two of the four vertex relationships can be satisfied at the same time.

The commitments $(\ell_0, \ell_1, \ell'_0, \ell'_1)$ generated by above steps have the exact same distribution as the commitments provided by the provers and these commitments are generated using one-way signalling as variables are sent only in one direction i.e. left to right. Also due to the symmetry of the method we can even start from the inputs of $\mathcal{S}_2^*$ and then proceed similarly as above, making the box symmetric. So, the 3-COLZK box is symmetric and can be implemented via one-way signalling. Therefore, using the theorem-17 from [CRC19] it is clear that the 3-COLZK box is no-signalling. □

Now, we will formally describe the no-signalling simulator for the protocol $\Pi_{\text{lhv}}[G]$.

No signalling simulation for the protocol $\Pi_{\text{lhv}}[G]$ (protocol-3.1.2) [ABC$^+$20]

STEP ONE : The four boolean pairs $(z_{00}, z'_{00}) \in \{0, 1\}$, $(z_{01}, z'_{01}) \in \{0, 1\}$, $(z_{10}, z'_{10}) \in \{0, 1\}$ and $(z_{11}, z'_{11}) \in \{0, 1\}$ are generated by $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ using PR boxes and local operations, corresponding to the cases $i = i'$, $i = j'$, $j = i'$ and $j = j'$ respectively where, i, i', j and j' $\in V$. If any of the case is true, then its corresponding pair has two unequal values. Also,

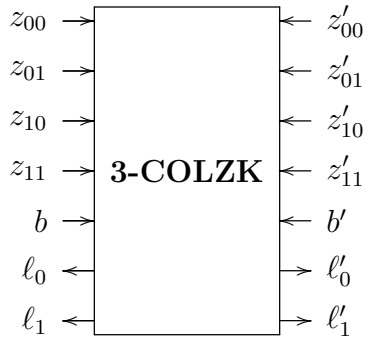


Figure 3.1: The **3-COLZK**-box

we can have more than one case satisfied at the same time. Now, for computation of these pairs the simulators ($\mathcal{S}_1^*$ and $\mathcal{S}_2^*$) do as follows:

1. $\mathcal{S}_1^*$ receives $((i,j),b) \in_{\mathcal{D}_G} E \times \mathbb{F}_2$ from $\mathcal{V}_1$ and $\mathcal{S}_2^*$ receives $((i',j'),b') \in_{\mathcal{D}_G} E \times \mathbb{F}_2$ from $\mathcal{V}_2$.

2. Now, $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ compute values of z_{00} and z'_{00} respectively in such a way that $z_{00} \neq z'_{00}$, or $z_{00} \oplus z'_{00} = 1$, only if $i = i'$. For that, suppose two variables x and x' of ℓ bits each where ℓ denotes the number of bits required to represent the larger value among vertices i and i'. Assuming x_j to be the j^{th} bit of x and $\wedge$ be the AND operator, we set:

$$x_j \oplus x'_j = \bigwedge_{k=1}^{j} (\overline{i_k} \oplus i'_k)$$

Also, $x_{j-1} \oplus x'_{j-1} = \bigwedge_{k=1}^{j-1} (\overline{i_k} \oplus i'_k)$ and $x_1, x'_1 = \overline{i_1}, i'_1$. For $j > 1$,

$$x_j \oplus x'_j = (x_{j-1} \oplus x'_{j-1}) \wedge (\overline{i_j} \oplus i'_j)$$

$$x_j \oplus x'_j = (x_{j-1} \wedge \overline{i_j}) \oplus (x_{j-1} \wedge i'_j) \oplus (x'_{j-1} \wedge \overline{i_j}) \oplus (x'_{j-1} \wedge i'_j) \tag{3.11}$$

Considering the above equation, it is evident that $(x_{j-1} \wedge \overline{i_j})$ and $(x'_{j-1} \wedge i'_j)$ can be calculated, by $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ respectively, using local operations whereas the computation of $(x_{j-1} \wedge i'_j)$ and $(x'_{j-1} \wedge \overline{i_j})$ can be achieved using PR boxes. Thus, using PR box

equation (2.5) we have:

$$y \oplus y' = x_{j-1} \wedge i'_j$$

$$z \oplus z' = x'_{j-1} \wedge \overline{i_j}$$

Now, using above two equations in equation-3.11 we get,

$$x_j \oplus x'_j = (x_{j-1} \wedge \overline{i_j}) \oplus (y \oplus y') \oplus (z \oplus z') \oplus (x'_{j-1} \wedge i'_j)$$

$$x_j \oplus x'_j = ((x_{j-1} \wedge \overline{i_j}) \oplus y \oplus z) \oplus (y' \oplus z' \oplus (x'_{j-1} \wedge i'_j)) \tag{3.12}$$

From above equation-3.12, we can notice that $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ can compute $x_j = ((x_{j-1} \wedge \overline{i_j}) \oplus y \oplus z)$ and $x'_j = (y' \oplus z' \oplus (x'_{j-1} \wedge i'_j))$ respectively using two PR boxes and local operations which then is further used to compute x_ℓ and x'_ℓ for setting $z_{00} = x_\ell$ and $z'_{00} = x'_\ell$.

3. Using the same method as described in step (2), the $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ calculate the values of (z_{01}, z'_{01}), (z_{10}, z'_{10}) and (z_{11}, z'_{11}).

Once, all four boolean pairs are computed, the simulators move to the next step, i.e. computing the commitments.

STEP TWO: In this step of the simulation, the simulators $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ generate the commitments (ℓ_0, ℓ_1) and (ℓ'_0, ℓ'_1), using the 3-COLZK box (described earlier), for $\mathcal{V}_1$ and $\mathcal{V}_2$ respectively based on the constraints of the protocol-3.1.2. And, the generated commitments have the exact same distribution as the provers' commitments.

Finally the simulators send the commitments generated above to the verifiers.

(Remark: Having a non-rewinding no-signalling simulator implies that the protocol is unsound against no-signalling provers because they could act just like the simulators.)

Complexity of the no-signalling simulator in terms of number of PR Boxes required

- During STEP ONE of the simulator, the simulators need to produce 4 boolean pairs corresponding to 4 possible relations between vertices received and each vertex can be of maximum $\log(|V|)$ bits where $|V|$ denotes the number of vertices of G. And also the generation of each boolean pairs require use of 2 PR Boxes. Therefore, the total number of PR Boxes required in this simulation step is $4 * 2 * \log(|V|) = 8\log(|V|)$.

- According to [FW11], for any ϵ, we can approximate the 3-COLZK-box (shown in 3.1), needed in STEP TWO of the simulation, within ϵ with a finite number of PR Boxes which is of the order of $O\left(\log\left(\frac{1}{\epsilon}\right)\right)^9 = O\left(-(\log(\epsilon))^9\right)$.

So, the total number of PR boxes required by the simulators are $O(-(\log(\epsilon))^9) + 8\log(|V|)$, which is polynomial in terms $\frac{1}{\epsilon}$ and $|V|$. Therefore, making the simulation efficient.

Chapter 4

CONCLUSION AND FUTURE WORK

The objective of this thesis has been to study interactive zero-knowledge proofs under the relativistic assumptions. We provided a proof that an existing protocol is a practical proof of knowledge. This new practical proof of knowledge is significant because most of the earlier work on proof of knowledge has been of theoretical interest due to high communication costs of the protocol. Also, proof of knowledge in general are of importance as proof of membership is not sufficient for various applications like proving identity of a user. So, the user must provide proof of knowledge in order to show that it indeed knows a witness. Further, we show that our proof of knowledge does not follow the proof technique that Unruh used for showing it to be quantum proof of knowledge [Unr12].

Secondly, we show the existing protocol posses a stronger zero-knowledge property by providing no-signalling simulator for the same. This result is useful because like the no-signalling simulators the judges can also be separated such that the no-signalling verifiers are able to perform the protocol. Also, this stronger zero-knowledge property makes the protocol special as usually the simulators are signalling which are stronger compared to no-signalling simulators.

Further, the research done in this thesis, raised some more interesting questions which are as follows:

1. Is the protocol of [ABC$^+$20] a quantum Proof of Knowledge? If yes, would that quantum proof of knowledge be practical, i.e., the number of repetitions be sufficiently small for practical use?

2. Can we find an entangled simulator for zero-knowledge aspect of the protocol of [ABC$^+$20]?

3. Can we find a variation of the protocol of [ABC$^+$20] that is sound against no-signalling provers?

Bibliography

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

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

[ABC+20] Pouriya Alikhani, Nicolas Brunner, Claude Crépeau, Sébastien Designolle, Raphaël Houlmann, Weixu Shi, and Hugo Zbinden. Experimental relativistic zero-knowledge proofs. *arXiv preprint arXiv:2012.10452*, 2020.

[Bab85] László Babai. Trading group theory for randomness. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 421–429, 1985.

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

[BC86] Gilles Brassard and Claude Crépeau. Non-transitive transfer of confidence: A perfect zero-knowledge interactive protocol for sat and beyond. In *27th Annual*

Symposium on Foundations of Computer Science (sfcs 1986), pages 188–195. IEEE, 1986.

[BL17] Daniel J Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017.

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

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. Citeseer, 1986.

[BOGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Killian, and Avi Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceeding of the 20th STOC*, pages 113–131, 1988.

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

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

[CL17] André Chailloux and Anthony Leverrier. Relativistic (or 2-prover 1-round) zero-knowledge protocol for NP secure against quantum adversaries. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 369–396. Springer, 2017.

[CMS+19] Claude Crépeau, Arnaud Massenet, Louis Salvail, Lucas Stinchcombe, and Nan Yang. Practical relativistic zero-knowledge for np. *arXiv preprint arXiv:1912.08939*, 2019.

[CRC19] Xavier Coiteux-Roy and Claude Crépeau. The rgb no-signalling game. *arXiv preprint arXiv:1901.05062*, 2019.

[CSST11] Claude Crépeau, Louis Salvail, Jean-Raymond Simard, and Alain Tapp. Two provers in isolation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 407–430. Springer, 2011.

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

[EPR35] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935.

[FW11] Manuel Forster and Stefan Wolf. Bipartite units of nonlocality. *Physical Review A*, 84(4):042112, 2011.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

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

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

[Hen14] Ryan Henry. Efficient zero-knowledge proofs and applications. 2014.

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

[Ibr20] Aly Tarek Ibrahim. *A Study of Non-Local Strategies for Zero-Knowledge Proof Systems*. PhD thesis, McGill University (Canada), 2020.

[Ken99] Adrian Kent. Unconditionally secure bit commitment. *Physical Review Letters*, 83(7):1447, 1999.

[Kil90] J Kilian. Strong separation models of multi-prover interactive proofs. In *DIMACS Workshop on Cryptography*, 1990.

[LC97] Hoi-Kwong Lo and Hoi Fung Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410, 1997.

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

[May97] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical review letters*, 78(17):3414, 1997.

[NC02] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

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

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

[SCG+14] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

[She16] Eric Sheldon. Relativity: The special and the general theory, 100th anniversary edition, by albert einstein: Scope: general interest, edited book. level: general readership, non-specialists, pre-university, undergraduate, advanced undergraduate, postgraduate, early career researcher, researcher, teacher, specialist, scientist, 2016.

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

[SRA81] Adi Shamir, Ronald L Rivest, and Leonard M Adleman. Mental poker. In *The mathematical gardner*, pages 37–43. Springer, 1981.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 135–152. Springer, 2012.

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

[Wil13] Mark M Wilde. *Quantum information theory*. Cambridge University Press, 2013.

[ZKL] Zklux-2-2019. `https://sikoba.com/docs/zklux1/ZKLux1_Crepeau_RelativisticZK.pdf`. (Accessed on 06/13/2021).