Pontifícia Universidade Católica
do Rio de Janeiro

Camila Lima de Sousa

Post Processing in Quantum Cryptography Systems

Dissertação de Mestrado

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica, do Departamento de Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

Advisor: Prof. Guilherme Penello Temporão

Rio de Janeiro
September 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Camila Lima de Sousa

Post Processing in Quantum Cryptography Systems

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica. Approved by the Examination Committee:

Prof. Guilherme Penello Temporão
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Thiago Barbosa dos Santos Guerreiro
Departamento de Física – PUC-Rio

Prof. Marco Antonio Grivet Mattoso Maia
Departamento de Engenharia Elétrica – PUC-Rio

Rio de Janeiro, September the 18th, 2024

Camila Lima de Sousa

Bachelor of Science in Electrical Engineering with emphasis in Electric Power Systems, from the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), 2021. Member of the Laboratory of Optoelectronics since 2021.

Bibliographic data

Sousa, Camila Lima de

Post Processing in Quantum Cryptography Systems / Camila Lima de Sousa; advisor: Guilherme Penello Temporão. – 2024.

101 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2024.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Distribuição de Chaves Quânticas. 3. Taxa de Erro de Bits Quânticos. 4. Estimação de Erro. 5. Correção de Erro. 6. Amplificação de Privacidade. I. Temporão, Guilherme Penello. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

To Breno, Karine and Nadja.

Acknowledgments

First, I extend my deepest appreciation to Guilherme P. Temporão, my advisor, whose invaluable guidance and expertise have been crucial throughout this research. His insights and constructive feedback have influenced the development of this work.

I would like to express my sincere gratitude to my husband, Breno, for his unwavering support and encouragement throughout this journey. His belief in me has been a constant source of motivation.

I extend my heartfelt thanks to my family for their enduring support and encouragement throughout this process. Specially my mother, Karine, her understanding and patience have been a source of strength.

I am grateful to my friends at the university for their valuable contributions to my academic journey. Their shared knowledge and the fellowship we enjoyed enriched my experience and made the research process more enjoyable.

I am thankful to Pontifical Catholic University of Rio de Janeiro (PUC-Rio), for providing the essential resources and environment necessary for conducting this research.

Finally, I wish to thank CAPES, CNPq, FAPERJ and PUC-Rio for the financial support.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Sousa, Camila Lima de; Temporão, Guilherme Penello (Advisor). **Post Processing in Quantum Cryptography Systems**. Rio de Janeiro, 2024. 101p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Quantum communication protocols are essential for secure transmission of information, utilizing the principles of quantum mechanics to achieve security unattainable by classical cryptographic systems. Unlike traditional methods that rely on conventional cryptographic keys, quantum protocols exploit unique properties of quantum systems to ensure communication security. However, the practical implementation of quantum key distribution (QKD) is challenged by errors introduced during the generation and transmission of quantum states and the potential presence of eavesdroppers. This thesis explores some of the most commonly used strategies for error estimation, error reconciliation, and privacy amplification within QKD systems. Through a literature review and comprehensive simulations, the study evaluates the most effective techniques in each area. The ultimate goal of this analysis is to develop a method to be implemented on Rede Rio Quântica, a metropolitan quantum communication network interlinking the institutions PUC-Rio, CBPF and UFRJ via optical fibers and UFF through a free-space channel. The findings underscore the importance of optimizing error correction and privacy measures to enhance the reliability and security of quantum communication networks.

Keywords

Quantum Key Distribution; Quantum Bit Error Rate; Error Estimation; Error Reconciliation; Privacy Amplification.

Resumo

Sousa, Camila Lima de; Temporão, Guilherme Penello. **Pós Processamento em Sistemas de Criptografia Quântica**. Rio de Janeiro, 2024. 101p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Os protocolos de comunicação quântica são essenciais para a transmissão segura de informações, utilizando os princípios da mecânica quântica para alcançar uma segurança inatingível por sistemas criptográficos clássicos. Diferentemente dos métodos tradicionais que dependem de chaves criptográficas convencionais, os protocolos quânticos exploram propriedades únicas dos sistemas quânticos para garantir a segurança da comunicação. No entanto, a implementação prática da distribuição quântica de chaves (QKD) é desafiada por erros introduzidos durante a geração e transmissão de estados quânticos e pela possível presença de espiões. Esta dissertação explora algumas das estratégias mais usadas para estimativa de erros, correção de erros e amplificação de privacidade em sistemas de QKD. Por meio de uma revisão bibliográfica e simulações abrangentes, o estudo avalia as técnicas mais eficazes em cada área. O objetivo final desta análise é desenvolver um método a ser implementado na Rede Rio Quântica, uma rede de comunicação quântica metropolitana que interliga as instituições PUC-Rio, CBPF e UFRJ por meio de fibras ópticas e a UFF através de um canal de espaço livre. Os resultados destacam a importância de otimizar as medidas de correção de erros e privacidade para melhorar a confiabilidade e a segurança das redes de comunicação quântica.

Palavras-chave

Distribuição de Chaves Quânticas; Taxa de Erro de Bits Quânticos; Estimação de Erro; Correção de Erro; Amplificação de Privacidade.

Table of contents

List of figures

List of tables

List of Abreviations

BSM – Bell State Measurement

CBPF – Centro Brasileiro de Pesquisas Físicas

CHSH – Clauser-Horne-Shimony-Holt

DWDM – Dense Wavelength Division Multiplexing

GLLP – Gottesman, Lo, Lütkenhaus and Preskill

ILM – Inamori, Lütkenhaus and Mayers

IM – Intensity Modulator

LS – Light Source

LDPC – Low-Density Parity-Check

MZI – Mach–Zehnder Interferometer

MDI-QKD – Measurement-Device-Independent Quantum Key Distribution

PM – Phase Modulator

PUC-Rio – Pontifícia Universidade Católica do Rio de Janeiro

QBER – Quantum Bit Error Rate

Qubits – Quantum Bits

QC – Quantum Cryptography

QKD – Quantum Key Distribution

RNG – Random Number Generator

RRQ – Rede Rio Quântica

SPA – Sum-Product Algorithm

TF-QKD – Twin-field Quantum Key Distribution

UFRJ – Universidade Federal do Rio de Janeiro

UFF – Universidade Federal Fluminense

VOA – Variable Optical Attenuator

WDM – Wavelength Division Multiplexing

WCP – Weak Coherent Pulse

WCS – Weak Coherent State

"The world of quantum mechanics is not the world of your intuition. Quantum mechanics is the way the world really is."

Peter Shor, *MIT News.*

1
Introduction

Cryptography transforms readable messages into ciphertext, an encrypted format unintelligible without the correct decryption key. It derived from the Greek terms "crypto" and "graphy", which means hidden/secret and writing respectively. It is fundamental for securing communication to protect sensitive data from unauthorized access [1]. Encryption algorithms define how messages are encoded, ensuring that the resulting ciphertext remains secure from potential eavesdroppers. The key security requirement is that decryption without the corresponding key is infeasible. Although achieving absolute security is challenging, cryptographic systems strive to be highly resistant to unauthorized access. Initially focused on confidentiality, modern cryptography now also addresses authentication, digital signatures and non-repudiation [2].

Regardless of classical cryptography's strengths, the evolution of quantum computing has challenged its security assumptions. Digital computers are generally effective at simulating physical computing devices, leveraging their efficiency in handling complex calculations and processes. However, quantum mechanics introduces a paradigm shift that may challenge this conventional understanding. Limitations imposed by time and memory in digital simulations become more pronounced when addressing large-scale problems. Quantum mechanics alters these constraints by providing new approaches that could surpass the capabilities of classical simulations [3].

As the 19th century discovery of electrodynamics significantly influenced 20th century technological advancements, quantum mechanics is increasingly shaping contemporary science and technology. Quantum mechanics, forms the basis for Quantum Cryptography (QC), which has transitioned from theoretical foundations in the 1970s to practical applications in information security today [4]. This transition reflects a broader shift in how physicists view quantum mechanics, moving from theoretical puzzles to practical engineering tools.

The application of quantum mechanics promises to enhance the speed and security of computing. Quantum computers harness three fundamental principles that distinguish them from classical systems: superposition, interference, and entanglement. Superposition enables quantum memory to exist

in multiple states simultaneously, vastly expanding the computational possibilities compared to classical memory, which is limited to a single state at a time. Interference allows quantum systems to combine and manipulate these superpositions in complex ways, enabling more efficient problem-solving. Entanglement, perhaps the most profound of these effects, allows different parts of a quantum computer, or even separate quantum computers, to become correlated in ways that classical systems cannot replicate. This correlation enhances the computational power of quantum systems, enabling them to solve problems that are intractable for classical computers [5].

The concept of Quantum Cryptography was formulated with notable contributions by Wiesner in 1983 [6], and further advancements by Charles H. Bennett and Gilles Brassard in 1984 [7]. This new approach leverages the principles of quantum mechanics to further enhance security. Unlike classical methods, quantum cryptography addresses vulnerabilities exposed by potential future quantum computers, providing advanced levels of protection. This ensures the integrity of information against increasingly complex attacks, marking a major leap in securing communications.

Advancements in quantum cryptography have made quantum communication possible, enabling the transmission of quantum bits (qubits) over networks. The security of these systems is rooted in the fundamental quantum phenomena of superposition and entanglement [8], which allow qubits to exist in multiple states simultaneously and to be deeply interconnected across distances in a way that classical systems cannot replicate. However, a critical element of quantum security is further reinforced by the no-cloning theorem [9], a foundational principle in quantum mechanics. The no-cloning theorem asserts that it is impossible to create an identical copy of an arbitrary unknown quantum state. Since no unitary transformation can map two distinct quantum states onto the same state, exact cloning becomes impossible.

This principle stands in direct contrast to classical information, where perfect duplication of data is routine. Any attempt to clone a quantum state would disturb it due to wavefunction collapse, making eavesdropping inherently detectable. As a result, quantum key distribution (QKD) protocols leverage this principle to safeguard the transmission of secret keys, as the theorem ensures that any interception of qubits introduces errors that expose an eavesdropper's presence. Nevertheless, challenges remain, including maintaining qubit coherence over long distances and developing efficient quantum repeaters to extend the range and reliability of quantum communication.

With current technology, the realistic error rates on the sifted key are a few percent, compared to the 10^{-9} error rate typical in optical communi-

cation [4]. Still quantum communication protocols are vulnerable to errors from technical imperfections in the channel or equipment and potential eavesdropping. To ensure secure communication, additional steps such as error rate estimation, error reconciliation, and privacy amplification are necessary. The following chapters will explore these topics in detail and include simulations to clarify these concepts.

1.1 Motivation

Quantum Networks are designed to enable the transmission, distribution and sharing of quantum states among users spread across different geographic locations. Such networks would offer many applications. In the realm of quantum computing, they enable remote access to quantum computers, allowing users to leverage computational power from afar. In quantum communications, these networks support distributed cryptographic protocols between different nodes, enhancing security. Additionally, quantum metrology benefits from quantum networks in applications such as interferometric telescopes [10], clock synchronization [11] and conducting tests of quantum physics, like loophole-free Bell inequality violations [12].

Commonly, this type of networks are confined to specific geographic locations. To achieve their full potential, it is crucial for these networks to be globally interconnected, enabling seamless exchange, sharing and measurement of qubits between any two nodes. This worldwide integration defines the vision of the Quantum Internet, facilitating enhanced communication and computing capabilities beyond local constraints [13, 14].

In a collaborative effort to advance quantum technology in Brazil, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), alongside Universidade Federal Fluminense (UFF), Universidade Federal do Rio de Janeiro (UFRJ) and Centro Brasileiro de Pesquisas Físicas (CBPF), developed a specialized center focusing on quantum networks and quantum internet. This involves the creation of a quantum communication network, Rede Rio Quântica (RRQ), in Rio de Janeiro. The central hub at PUC-Rio in Gávea is interconnected with CBPF in Urca and UFRJ in Ilha do Fundão via optical fibers from the Rede-Rio de Computadores/FAPERJ and links to UFF in Gragoatá/Niterói through a free-space optical link, as illustrated in Figure 1.1.

In Figure 1.2, users of RRQ are represented by shaded boxes: UFRJ (Alice), PUC-Rio (Charlie), and UFF (Debbie). Bob is split into CBPF1 and CBPF2, representing two different laboratories. The network's topology

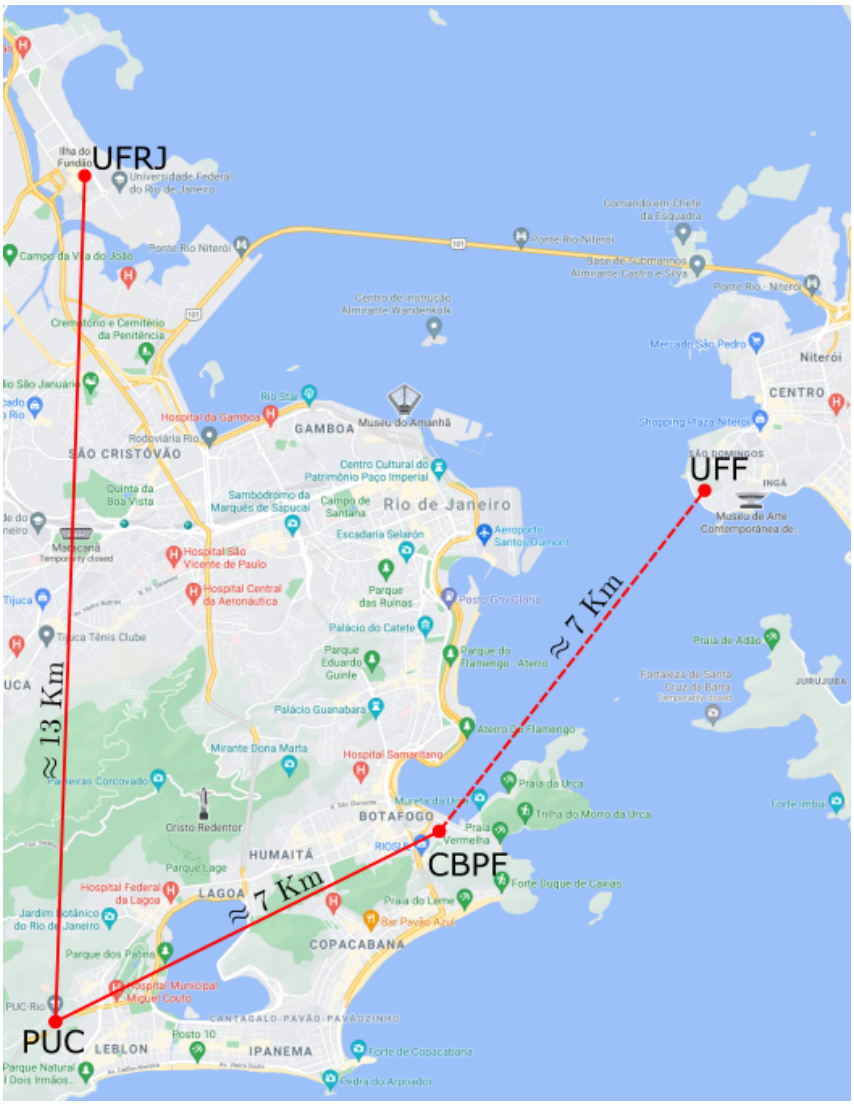

Figure 1.1: Basic scheme of Rede Rio Quântica structure. Image: Google Maps.

employs the CAL19 variant of the Twin-Field Quantum Key Distribution (TF-QKD) protocol [15], incorporating a large Sagnac interferometer [16] to connect all users. The developed scheme, offers several advantages over protocols used in commercially available systems today.

In the CAL19 TF-QKD scheme, the qubit is encoded in the direction of propagation through the Sagnac, forming two orthogonal states $|\circlearrowleft\rangle$ and $|\circlearrowright\rangle$. Charlie (PUC-Rio) prepares the transmission of a fixed state given by $|\psi\rangle = \frac{1}{\sqrt{2}}(|\circlearrowright\rangle + i\,|\circlearrowleft\rangle)$.

Alice and Bob then introduce relative phase shifts to the clockwise component, affecting the returned state to Charlie as $|\psi'\rangle = \frac{1}{\sqrt{2}}(e^{i(\phi_A+\phi_B)}\,|\circlearrowright\rangle + i\,|\circlearrowleft\rangle)$. The Sagnac interferometer's automatic phase stabilization removes the need for active phase control, unlike other TF-QKD systems. Synchronization is achieved using a Dense Wavelength Division Multiplexing (DWDM) channel, with Wavelength Division Multiplexing (WDM) elements ensuring proper

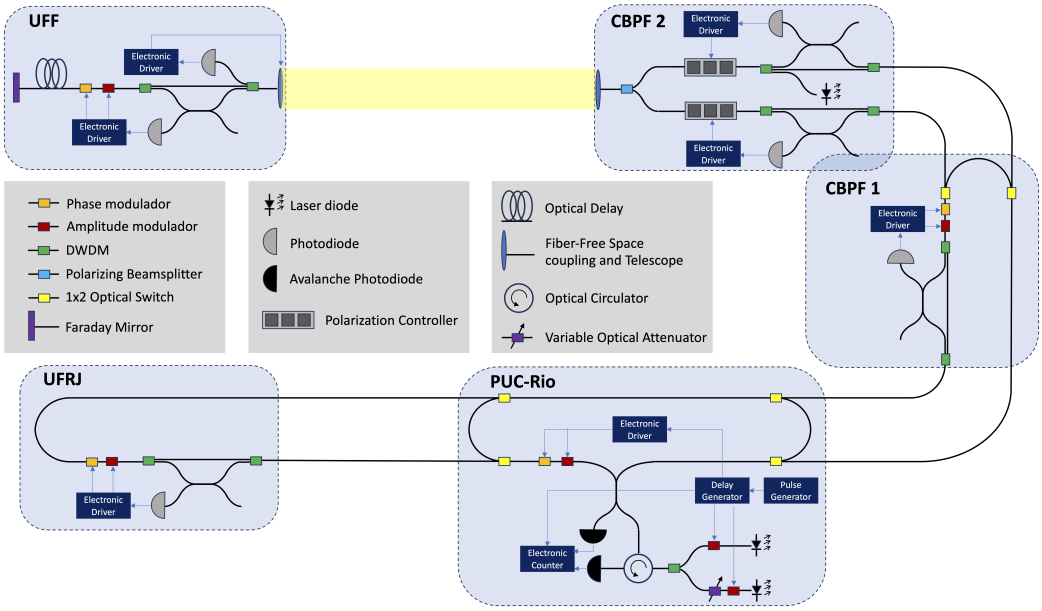

Figure 1.2: Illustration of RRQ scheme.

timing.

To incorporate Debbie (UFF) into the network, a polarization beam splitter at CBPF2 and a Faraday mirror at UFF are used, enabling effective operation of the Sagnac interferometer in free-space connections. Photons traveling from CBPF to UFF exhibit vertical polarization in a clockwise direction and horizontal in a counterclockwise direction. Polarization controllers are installed before coupling with the telescope at CBPF2 to address time-induced polarization state changes.

Finally, 1x2 optical switches at PUC-Rio and CBPF1 reconfigure the Sagnac interferometer to bypass non-participating users, reducing losses and improving key generation rates during specific key exchanges. This setup is particularly useful when Charlie and Debbie engage in key exchange, as the switches at PUC-Rio facilitate the bypass, optimizing system efficiency.

This work proposes a post-processing protocol designed to enhance the security of information transmission within Rede Rio Quântica. By focusing on optimizing both data integrity and confidentiality, the proposed protocol offers a framework that strengthens the network's resilience against potential interception and data loss. It ensures that transmitted information remains secure by incorporating advanced cryptographic techniques and error correction methods. This dual approach safeguards the data from unauthorized access and also minimizes the risk of data corruption, thereby maintaining the overall reliability of the quantum communication network.

2
Theoretical Background

This chapter provides the fundamental concepts for the understanding of this work. It begins with an introduction to quantum mechanics, emphasizing its foundational postulates, which are essential for comprehending the behavior of quantum systems. Furthermore, the chapter delves into the concept of Quantum Bit Error Rate (QBER), a parameter for assessing the integrity and security of quantum communication systems and concludes with principles of optical interferometry.

2.1
Postulates of Quantum Mechanics

This section provides a summary of quantum mechanics, highlighting its role as the mathematical framework used to develop physical theories. Although quantum mechanics does not dictate specific physical laws, it provides the mathematical and conceptual tools necessary for their development.

The postulates of quantum mechanics emerged from a process filled with assumptions and guesswork. This iterative approach involved exploring ideas and refining them over time to establish the foundational principles of the theory [17]. While the sequence and number of quantum mechanics postulates can differ among various sources, they convey the same fundamental ideas. This work references [18] due to its detailed approach.

1º Postulate: a physical system's state is represented by a complex vector known as the state vector, which belongs to a mathematical structure called the Hilbert space [19] or state space. Using Dirac's notation [20], a quantum state is symbolized as $|\psi(t)\rangle$.

2º Postulate: physical quantities that can be measured are represented by observable operators within the corresponding Hilbert space.

3º Postulate: measurements yield only the eigenvalues of the observable operator. For an observable $\hat{A}$ and an eigenstate $|u_n\rangle$, the measurement result is the eigenvalue a_n, as shown in equation 2-1.

$$\hat{A}\,|u_n\rangle = a_n\,|u_n\rangle \tag{2-1}$$

4º Postulate: the probability of obtaining a specific eigenvalue when measuring an observable is given by the square modulus of the projection of the quantum state onto the corresponding eigenstate of that observable. For the case of a non-degenerated spectrum, the probability $P(a_n)$ of obtaining the eigenvalue a_n when measuring an observable is determined by the expression 2-2.

$$P(a_n) = |\langle u_n|\psi\rangle|^2 \tag{2-2}$$

5º Postulate: following a measurement, the quantum state collapses to the eigenstate associated with the observed eigenvalue. This collapse corresponds to the projection of the initial state onto the relevant eigenstate, as formally described in equation 2-3).

$$|\psi'\rangle = \frac{P_{a_n}|\psi\rangle}{\sqrt{\langle\psi| P_{a_n}|\psi\rangle}} \tag{2-3}$$

6º Postulate: the Schrödinger equation 2-4 [21], determines the time evolution of the state $|\psi(t)\rangle$, with $\hat{H}(t)$ representing the Hamiltonian operator.

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = \hat{H}(t)|\psi(t)\rangle \tag{2-4}$$

In cases where the Hamiltonian does not depend on time, the Schrödinger equation simplifies to the expression given in equation 2-5.

$$|\psi(t)\rangle = e^{-\frac{i\hat{H}}{\hbar}t}|\psi_0\rangle \tag{2-5}$$

Through the use of these mathematical tools, one can conduct an analysis of classical systems, such as the harmonic oscillator, within the domain of quantum mechanics. This approach enhances the understanding of states that are commonly used in quantum communications, thereby deepening the comprehension of their practical applications.

2.2 Quantum Harmonic Oscillator

The harmonic oscillator is a foundational concept in physics, with applications across classical and quantum mechanics. It is essential for understand-

ing the quantization of energy levels and the behavior of quantum fields. In the context of a one-dimensional cavity containing an electromagnetic field, the Hamiltonian of this system is expressed through equation 2-6, as derived from classical principles [22]. Within this framework, the operators $\hat{p}$ and $\hat{q}$ represent the momentum and position, respectively, offering a detailed representation of the system's dynamics.

$$\hat{H} = \frac{1}{2}\left(\hat{p}^2 + \omega^2\hat{q}^2\right) \tag{2-6}$$

To simplify the determination of permissible energy levels and their corresponding quantum states, illustrated in figure 2.1, one can employ the annihilation ($\hat{a}$) and creation ($\hat{a}^\dagger$) operators. By employing these non-observable operators, defined in equations 2-7 and 2-8, the calculations become more straightforward, making it easier to comprehend the system's quantum behavior. Within this context, the Hamiltonian can be reformulated in terms of these operators, as shown in Equation 2-9.

$$\hat{a} = (2\hbar\omega)^{-1/2}(\omega\hat{q} + i\hat{p}) \tag{2-7}$$

$$\hat{a}^\dagger = (2\hbar\omega)^{-1/2}(\omega\hat{q} - i\hat{p}) \tag{2-8}$$

$$\hat{H} = \hbar\omega\left(\hat{a}^\dagger\hat{a} + \frac{1}{2}\right) \tag{2-9}$$

The number operator $\hat{a}^\dagger\hat{a}$, denoted as $\hat{n}$, is fundamental to the analysis of quantum systems. Its eigenstates, labeled $|n\rangle$, also serve as eigenstates of the Hamiltonian. Physically, this operator characterizes Fock states [23], which represent quantum states with a well-defined number of photons within the cavity. Each Fock state is associated with an energy level E_n, which corresponds to an eigenvalue of the Hamiltonian $\hat{H}$. Therefore, investigating the interaction of the creation operator with the Hamiltonian is essential for understanding their respective functions and implications.

$$\hat{H}(\hat{a}^\dagger|n\rangle) = (E_n + \hbar\omega)(\hat{a}^\dagger|n\rangle) \tag{2-10}$$

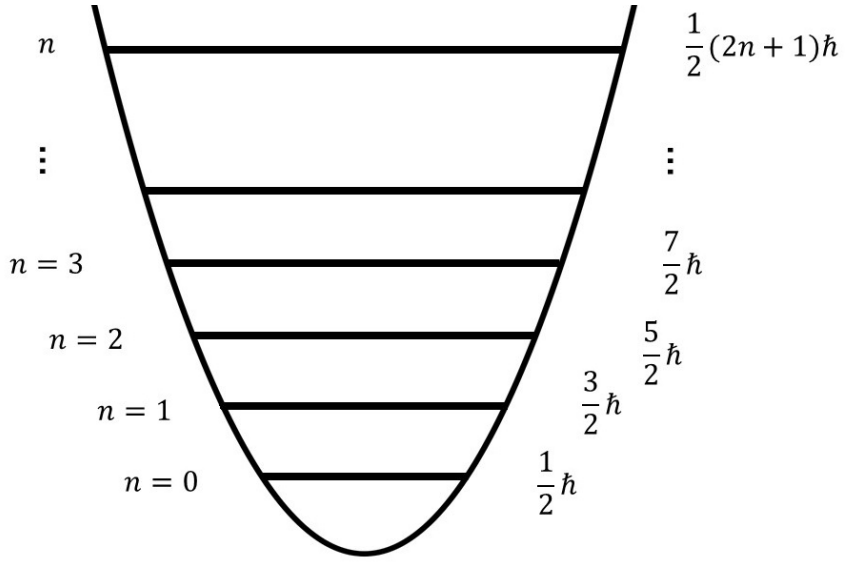

Figure 2.1: The energy levels of a general harmonic oscillator.

$$\hat{H}(\hat{a}\,|n\rangle) = (E_n - \hbar\omega)(\hat{a}\,|n\rangle) \tag{2-11}$$

The creation operator increases the system's energy by $\hbar\omega$, effectively adding one photon with frequency ω. In contrast, the annihilation operator reduces the system's energy by $\hbar\omega$ by removing a photon of the same frequency. When applied to the Fock state, these operators lead to the expressions given in equations 2-12 and 2-13. The following derivations will provide the explicit values for the allowed energy levels, E_n, represented in equation 2-14.

$$\hat{a}\,|n\rangle = \sqrt{n}\,|n-1\rangle \tag{2-12}$$

$$\hat{a}^\dagger\,|n\rangle = \sqrt{n+1}\,|n+1\rangle \tag{2-13}$$

$$E_n = \hbar\omega\left(n + \frac{1}{2}\right) \tag{2-14}$$

In the analysis of single-mode fields, Fock states play a crucial role. They constitute an orthogonal and complete set, serving as basis for describing photon systems in quantum mechanics. This orthogonality and completeness are fundamental, allowing a precise description of quantum states in a given mode. The significance of Fock states extends to practical applications, including

quantum information processing, quantum computing and quantum cryptography, where they enable high-precision experiments and the development of critical technologies like single-photon sources and detectors [22].

2.3 Weak Coherent States

In quantum optics, a coherent state is characterized by a well-defined phase and amplitude, closely resembling a classical electromagnetic field. Weak coherent states extend this concept by considering states with relatively low photon numbers, where quantum effects remain significant, yet the states are not as fundamentally quantum as Fock states [23, 24].

Weak Coherent States (WCSs) [25] offer a practical and cost-effective approach for the probabilistic generation of single-photon pulses. Weak Coherent Pulses (WCPs) represent practical instances of WCSs, where theoretical understandings of WCSs are applied to generate and use low-photon-number pulses in quantum communication systems. This technique, involving a faint laser, is widely employed in quantum cryptography systems designed for QKD. Due to their laser-like nature, WCPs are characterized by Fock states and are modeled by a Poissonian distribution [26]. In the Fock state basis, a WCP can be represented as follows:

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle \,, \tag{2-15}$$

Where α denotes the complex coherent amplitude and $|n\rangle$ represents the Fock state with n photons. The probability of observing n photons in the weak coherent pulse follows a Poissonian distribution:

$$P(n) = \frac{|\alpha|^{2n} e^{-|\alpha|^2}}{n!}. \tag{2-16}$$

This distribution describes the likelihood of different photon number outcomes, where $|\alpha|^2$ represents the average photon number. The inherent probabilistic nature of photon counts in WCSs within a given time interval prevents the deterministic generation of single-photon pulses. This stochastic behavior requires management of the probabilities associated with the emission of both multi-photon and vacuum pulses. The strong correlation between these probabilities underscores the need for precise control to preserve the integrity of the quantum communication system.

In the context of QKD systems, the presence of multi-photon pulses must be avoided due to the potential vulnerability to eavesdropping through a photon-number splitting attack [27]. Such attacks exploit the additional photons to gain unauthorized access to the key information. To mitigate this vulnerability, one can implement substantial attenuation of the source, ensuring that the average photon count per pulse remains significantly below one. This strategy effectively reduces the probability of multi-photon emissions, thereby enhancing the security and reliability of the QKD system.

2.4 QBER

Quantum Bit Error Rate (QBER) is a metric used to assess the integrity and security of the transmitted quantum keys. It quantifies the proportion of erroneous bits detected during the key generation process, reflecting the presence of errors or potential eavesdropping activities. Accurate measurement of QBER is essential for evaluating the overall performance of QKD systems, as it directly influences the feasibility of secure communication.

Formally, it is defined as the ratio of incorrect bits to the total number of bits exchanged between users [4]. Typically, QBER is expected to be low, with an ideal target being below 11% [28]. The QBER can be expressed as follows:

$$\mathrm{QBER} = \frac{N_{\mathrm{wrong}}}{N_{\mathrm{right}} + N_{\mathrm{wrong}}} = \frac{R_{\mathrm{error}}}{R_{\mathrm{sift}} + R_{\mathrm{error}}} \approx \frac{R_{\mathrm{error}}}{R_{\mathrm{sift}}} \tag{2-17}$$

The *raw rate*, denoted as R_{raw} represents the total bit rate received by Alice and Bob before any basis reconciliation has been performed. This rate can be calculated by multiplying the pulse rate f_{rep}, the average number of photons per pulse μ, the probability of a photon reaching the receiver t_{link}, and the probability of photon detection η.

The *sift rate* is defined as half of the *raw key* rate. This is due to the fact that the *sift key* includes only those instances in which Alice and Bob have selected compatible measurement bases. Consequently, the *sift rate* can be expressed as follows:

$$R_{\mathrm{sift}} = \frac{1}{2} R_{\mathrm{raw}} = \frac{1}{2} q f_{\mathrm{rep}} \mu t_{\mathrm{link}} \eta \tag{2-18}$$

The factor q assumes values between 0 and 1, typically being $\frac{1}{2}$ or 1, depending on the specific phase encoding configurations employed to correct for non-interfering path combinations.

The error rate, denoted as R_{error}, as presented in Equation 2-17, can be divided into three components. The first component is the rate R_{opt}, which quantifies the proportion of photons detected by the incorrect detector, with a probability p_{opt}, due to factors such as interference or polarization misalignment. This rate is expressed as:

$$R_{\text{opt}} = R_{\text{sift}} p_{\text{opt}} = \frac{1}{2} q f_{\text{req}} \mu t_{\text{link}} p_{\text{opt}} \eta \tag{2-19}$$

The second component, denoted as R_{det}, arises from spurious detections, commonly known as dark counts. These are erroneous signals detected due to internal noise in the detectors rather than actual photon events and are independent of the bit rate. However, errors are only generated when these dark counts occur within a specific time window where a photon detection is expected.

Therefore, R_{det} is determined by p_{dark}, which represents the probability of registering a dark count within a given time window for each detector, and by n, the number of detectors. Additionally, R_{det} incorporates two $\frac{1}{2}$ factors. One accounts for the probability of a dark count occurring when Alice and Bob choose different bases, and the other represents the probability of a dark count being registered by the correct detector. Thus, it can be written as:

$$R_{\text{det}} = \frac{1}{2}\frac{1}{2} f_{\text{req}} p_{\text{dark}} n \tag{2-20}$$

The final component, R_{acc}, is relevant exclusively in protocols employing entanglement. This error rate arises due to the presence of uncorrelated photons, which can occur when the photon sources generate multiple pairs. Thus, it can be described as:

$$R_{\text{acc}} = \frac{1}{2}\frac{1}{2} p_{\text{acc}} f_{\text{req}} t_{\text{link}} n \eta \tag{2-21}$$

Where p_{acc} represents the probability of additional photons being present within a specific time window. By integrating these components, the QBER can be formulated as:

$$\begin{aligned} \text{QBER} &= \frac{R_{\text{opt}} + R_{\text{det}} + R_{\text{acc}}}{R_{\text{sift}}} \\ &= QBER_{\text{opt}} + QBER_{\text{det}} + QBER_{\text{acc}} \end{aligned} \tag{2-22}$$

In quantum communication systems, QBER is a useful measure of accuracy. By identifying and correcting errors through error correction protocols, and by employing error estimation techniques to optimize these protocols, the integrity and security of quantum transmissions are maintained. This combined approach ensures that quantum communication systems remain both secure and reliable.

2.5 Optical Interferometry

Interferometry is a technique used in physics and optics that involves the superposition of waves to produce interference patterns, thereby providing detailed insights into the properties of light and other waves. This technique finds applications across various scientific disciplines, including physics, astronomy, chemistry, and engineering.

The precision and resolution offered by interferometry render it a valuable tool for optical sensors. The interference patterns generated can reveal information about wave sources, distances and variations in physical properties. Moreover, these patterns can be controlled through external perturbations, such as modulating light intensity using electrical signals. Hence, this discussion will focus on three principal configurations of light interferometers and will include calculations of electromagnetic field interactions [29, 30].

2.5.1 Sagnac

The Sagnac interferometer, a device rooted in the principles of interference, is particularly notable for its application in ring interferometry. Initially described by French physicist Georges Sagnac in 1913 [16], this interferometer is used in rotation sensors and gyroscopes due to its capacity to measure angular rotation by correlating it with phase shifts in the beams.

Typically, it is composed of a beam splitter, a ring-shaped optical path and multiple mirrors. The light from a source is divided into two counter-propagating beams that traverse the closed loop of the ring. Considering the input electromagnetic field as U_1, as figure 2.2 shows, one can right the the split fields as follows:

$$U_2 = \frac{1}{\sqrt{2}} U_1 e^{j\frac{\pi}{2}} \tag{2-23}$$

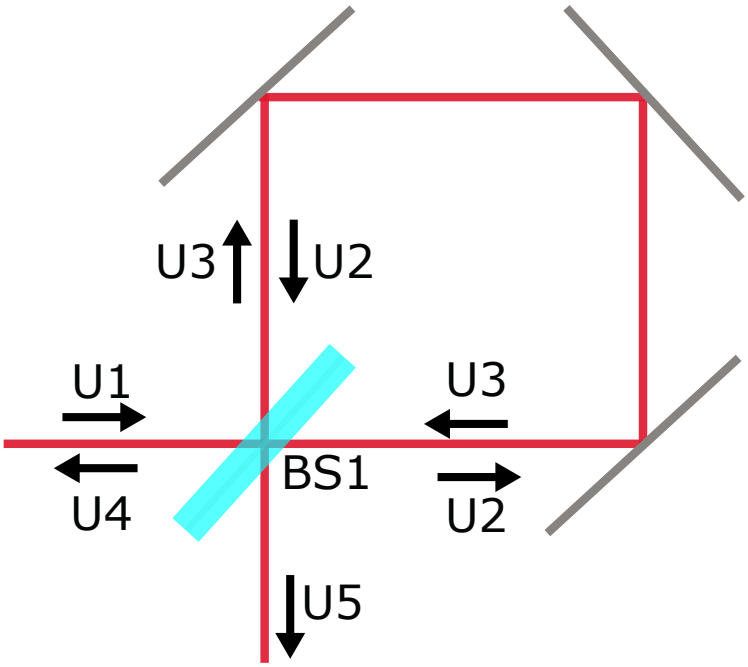

Figure 2.2: Sagnac Interferometer Scheme.

$$U_3 = \frac{1}{\sqrt{2}} U_1 \tag{2-24}$$

In this context, U_2 and U_3 represent the beams that pass through and reflect off the beam splitter, respectively. The factor $1/\sqrt{2}$ refers to the power splitting at the beam splitter, as power is proportional to the absolute square of the field amplitude. The phase shift of the transmitted beam arises from energy and momentum conservation considerations [31, 32]. The resulting expressions for the beams can be written as follows:

$$U_4 = \frac{1}{\sqrt{2}} U_2 e^{j\phi} + \frac{1}{\sqrt{2}} U_3 e^{j\frac{\pi}{2}+j\phi} \tag{2-25}$$

$$U_5 = \frac{1}{\sqrt{2}} U_2 e^{j\frac{\pi}{2}+j\phi} + \frac{1}{\sqrt{2}} U_3 e^{j\phi} \tag{2-26}$$

Where ϕ represents the additional phase acquired along the interferometer path. Furthermore, using equation 2-23 and 2-24, the output fields can be expressed in terms of the input fields as follows:

$$U_4 = \frac{1}{2} U_1 e^{j\frac{\pi}{2}+j\phi} + \frac{1}{2} U_1 e^{j\frac{\pi}{2}+j\phi} = U_1 e^{j\frac{\pi}{2}+j\phi} \tag{2-27}$$

$$U_5 = \frac{1}{2}U_1 e^{j\pi+j\phi} + \frac{1}{2}U_1 e^{j\phi} = \frac{1}{2}(U_1 - U_1)e^{j\phi} = 0 \tag{2-28}$$

Therefore, output at the opposite port of the beam splitter is zero. However, this result can vary if an additional phase difference is introduced between the internal beams, such as in applications for gyroscopes [33, 34]. In contrast, the output along the same path as the incoming beam retains the same power but acquires an additional phase of $\pi/2 + \phi$.

In Rede Rio Quântica, the Sagnac interferometer is chosen for its automatic phase stabilization, which eliminates the need for active phase control typically required in other QKD systems. This feature simplifies the network's architecture and enhances its reliability, making it an optimal choice for a large-scale, secure quantum communication network like Rede Rio Quântica.

2.5.2 Mach-Zehnder

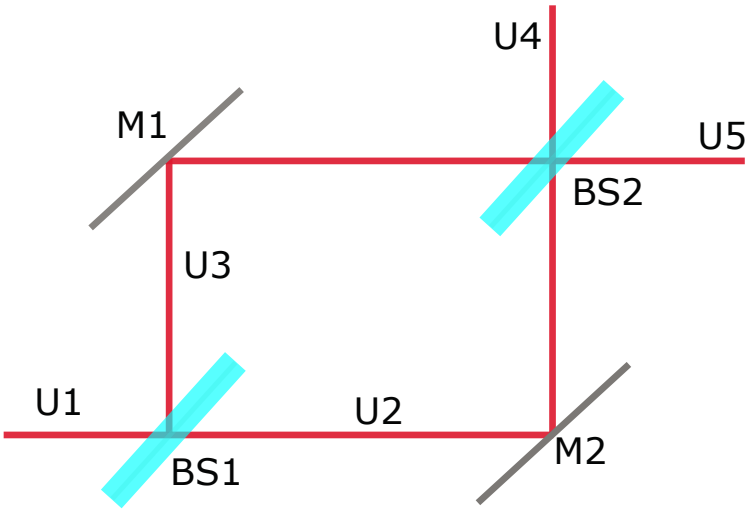

Figure 2.3: Mach-Zehnder Interferometer Scheme.

Originally developed by physicists Ludwig Mach and Ludwig Zehnder, the Mach-Zehnder interferometer is a tool in wave optics, used to generate and analyze interference patterns. This device operates by splitting a beam of light with two beam splitters and redirecting the light paths using mirrors, which recombine to produce interference effects. Its design and functionality make it indispensable in various scientific fields, including quantum mechanics, optical communications, and precision metrology [35, 36].

Figure 2.3 illustrates a balanced Mach-Zehnder interferometer, where both beams traverse identical optical paths. Consider U_1, the input electromagnetic field. The beam is divided into two paths, producing fields U_2 and

U_3. Since the balanced nature of the interferometer, both beams acquire the same phase shift, denoted as ϕ. One can write these fields as in equation 2-23 and 2-24. Upon reaching the final beam splitter, these fields result in output fields U_4 and U_5, which can be described as functions of the initial input field.

$$U_4 = \frac{1}{\sqrt{2}} U_2 e^{j\frac{\pi}{2}+j\phi} + \frac{1}{\sqrt{2}} U_3 e^{j\phi} = \frac{1}{2} U_1 e^{j\pi+j\phi} + \frac{1}{2} U_1 e^{j\phi} = 0 \tag{2-29}$$

$$U_5 = \frac{1}{\sqrt{2}} U_2 e^{j\phi} + \frac{1}{\sqrt{2}} U_3 e^{j\frac{\pi}{2}+j\phi} = \frac{1}{2} U_1 e^{j\frac{\pi}{2}+j\phi} + \frac{1}{2} U_1 e^{j\frac{\pi}{2}+j\phi} = U_1 e^{j\frac{\pi}{2}+j\phi} \tag{2-30}$$

From the equations presented, it is clear that in one of the output ports of the beam splitter, the fields are out of phase, resulting in destructive interference, which effectively cancels the fields. In the other output port, the fields are in phase, leading to constructive interference. The resultant output field in this port matches the input field but with an additional phase shift of $\pi/2 + \phi$.

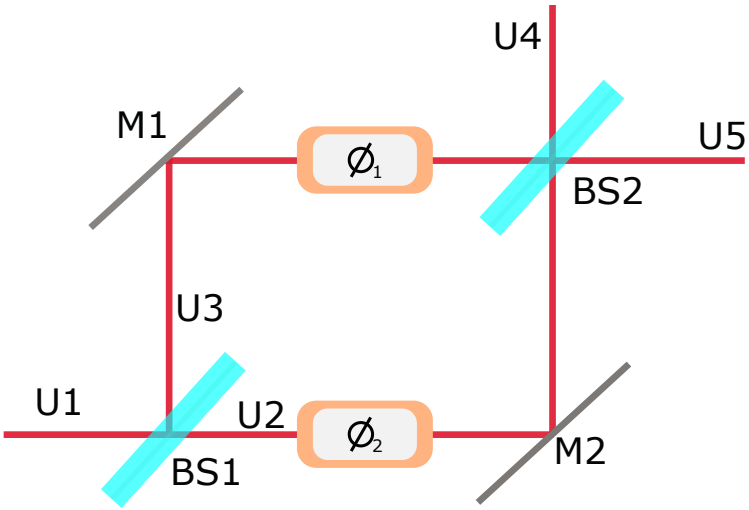

Figure 2.4: Unbalenced Mach-Zehnder interferometer.

Now, let us examine the scenario where a phase difference is introduced between the beams. In this case, the field in the upper arm, denoted as U_3, acquires an additional phase shift ϕ_1, while the field in the lower arm gains a phase shift ϕ_2. This configuration is illustrated in figure 2.4, which includes two phase shifters to illustrate the phase difference. The resulting fields U_4 and U_5 can then be expressed as follows.

$$U_4 = \frac{1}{\sqrt{2}} U_2 e^{j\frac{\pi}{2}+j\phi_2} + \frac{1}{\sqrt{2}} U_3 e^{j\phi_1} = \frac{1}{2} U_1 \left(e^{j\phi_1} - e^{j\phi_2} \right) \tag{2-31}$$

$$U_5 = \frac{1}{\sqrt{2}} U_2 e^{j\phi_2} + \frac{1}{\sqrt{2}} U_3 e^{j\frac{\pi}{2}+j\phi_1} = \frac{1}{2} U_1 \left(e^{j\phi_1} + e^{j\phi_2} \right) \tag{2-32}$$

In this scenario, the output fields are observed to depend on the additional phases introduced along the two paths. The power of these fields can be expressed in terms of the input optical power, ($P_1 = |U_1|^2$), as shown bellow.

$$P_4 = |U_4|^2 = \frac{P_1}{2} \left[1 - cos(\Delta\phi) \right] \tag{2-33}$$

$$P_5 = |U_5|^2 = \frac{P_1}{2} \left[1 + cos(\Delta\phi) \right] \tag{2-34}$$

Here, $\Delta\phi = \phi_1 - \phi_2$ represents the phase difference between the two paths. This dependency of output power on the phase difference enables the Mach-Zehnder interferometer to serve in various applications, such as sensors [37] and light modulators [38]. Furthermore, Mach-Zehnder modulators are commonly employed in QKD for generating WCPs [39].

2.5.3 Michelson

The Michelson interferometer is a fundamental optical instrument designed to produce interference patterns of light waves. Invented by the American physicist Albert A. Michelson in the late 19th century, the device gained prominence through its use in the Michelson–Morley experiment [40], which provided one of the strongest challenges to the Aether theory. Today, the Michelson interferometer is widely used across various fields, including spectroscopy, metrology, and the testing of optical materials.

Figure 2.5 illustrates a standard Michelson interferometer. An input field U_1 enters a beam splitter, which divides it into two fields, U_2 and U_3. These fields, as described by equations 2-23 and 2-24, travel along different paths, are reflected by mirrors and then return to the beam splitter. The resulting fields, U_4 and U_5, follow the same equations as those for the Mach-Zehnder interferometer, represented by equations 2-31 and 2-32. Therefore, the power

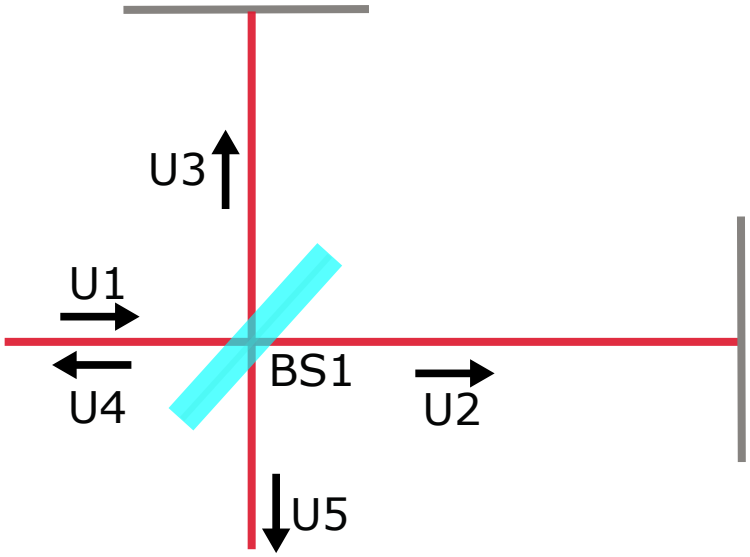

Figure 2.5: Michelson Interferometer Scheme.

of the output fields, P_4 and P_5, depends on the input field and the phase difference between the two paths.

Adjusting the position of the mirrors alters the path difference, a feature employed to investigate interference phenomena or measure physical quantities such as refractive index changes in gases [41]. The Michelson interferometer also finds applications in numerous areas such as medicine [42] and astronomy [43].

3
Quantum Communication Protocols

As advancements in quantum computing pose new challenges to traditional cryptographic systems, Quantum Key Distribution (QKD) has emerged as a groundbreaking solution. QKD offers unparalleled security by ensuring that once a quantum transmission is completed, no classical transcript remains available for eavesdroppers to analyze or decrypt. Unlike conventional cryptographic methods, where intercepted data might be stored and later decrypted with advanced computational resources, QKD provides long-term protection by fundamentally eliminating the risk of future data breaches [44].

In a QKD protocol, two parties establish a secure key through the use of a quantum channel for transmitting quantum signals and a classical channel for authentic communication. Although the quantum channel can be vulnerable to eavesdropping, any attempt to intercept the transmission introduces detectable disturbances. Consequently, both parties must estimate the extent of potential information leakage to ensure the key's security, as any degradation in the quantum channel indicates possible information loss [45].

The evolution of QKD has led to the development of a range of quantum communication protocols, each designed to address specific challenges in both quantum and classical cryptography. This chapter will explore several of these protocols, detailing their mechanisms and advantages.

3.1
BB84

The BB84 protocol, introduced by Charles Bennett and Gilles Brassard in 1984 [7], represents one of the pioneering algorithms in the field of Quantum Key Distribution. This protocol can be categorized into three distinct phases: the preparation and transmission of quantum states, the reconciliation of measurement bases, and the estimation of the quantum bit error rate. To simplify the analysis and focus on the core principles, it is assumed that all components involved, the photon source, detectors and optical devices, are ideal and operate with perfect fidelity. These assumptions facilitate the understanding of the protocol's theoretical framework.

Initially, one of the parties, commonly referred to as Alice, prepares the

quantum states by selecting a random bit sequence, the key, and encoding it into the polarization modes of photons. She the randomly chooses a basis for encoding, either the computational basis ($|0\rangle$ and $|1\rangle$) or the Hadamard basis ($|+\rangle$ and $|-\rangle$), which correspond to the rectilinear and diagonal bases, respectively. After preparing each photon, Alice transmits it through a quantum channel, such as an optical fiber, to the other party, known as Bob. This quantum channel can be modeled as a random polarization rotator due to variations in the medium's birefringence, which can be influenced by factors such as temperature and strain [46, 47].

Before receiving the photons, Bob randomly selects the basis in which he will measure each incoming photon. If Bob selects the same basis as Alice, he will correctly determine the bit with 100% probability. Alternatively, if he chooses a different basis, his probability of obtaining the correct bit is 50%. For example, if Alice transmits a bit encoded as $|+\rangle$, which corresponds to 0 in the Hadamard basis, and Bob measures using the rectilinear basis, the probability of success is:

$$P_s = |\langle 0|+\rangle|^2 = |\langle 0| \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|^2 = \frac{1}{2} \tag{3-1}$$

Therefore, to reduce uncertainties over the mismatched basis, Alice and Bob perform a basis reconciliation process. They use a public classical channel to exchange information about the bases they used for each transmission. When they find that different bases were chosen, they discard the corresponding data. On average, half of the key is rejected due to the random nature of the base selection.

Alice's polarization basis	↔	↗	↕	↔	↖	↕	↕	↔	↗	↕
Alice's bit sequence	0	0	1	0	1	1	1	0	0	1
Bob's chosen basis	×	×	+	×	+	+	×	+	+	+
Bob's polarization result	↖	↗	↕	↖	↔	↕	↗	↔	↕	↕
After basis reconciliation		↗	↕			↕		↔		↕
Shared key		0	1			1		0		1

Table 3.1: Example of BB84 QKD. The table illustrates the polarization bases selected by Alice, where ↔ and ↕ correspond to the bits 0 and 1, respectively, in the horizontal basis. Similarly, the Hadamard basis is represented by ↗ and ↖. Bob then chooses between these two bases, denoted here as + and ×. Finally, Alice and Bob publicly compare their chosen bases and discard any measurements where the bases do not match.

Following basis reconciliation, Alice and Bob proceed to estimate the QBER. They randomly select a portion of the remaining key and exchange

the bit information (0 or 1) through a public classical channel. These bits are discarded to prevent potential interception by eavesdroppers. This stage is required for assessing the quality of the quantum channel, particularly if it exhibits any loss or noise. Additionally, the estimated QBER helps detect the presence of an eavesdropper, often referred to as Eve. If the QBER exceeds or equals 11%, it indicates a possible eavesdropping attempt [28]. This threshold is based on the no-cloning theorem, which states that quantum states cannot be copied or amplified without introducing noise [48]. In such scenarios, it is advisable to discard the entire key and restart the protocol to ensure the security of the communication.

3.2 E91

In 1991, Artur K. Ekert introduced the E91 protocol [49]. This protocol represented a major advancement by incorporating quantum entanglement as a fundamental component for secure key distribution. Entanglement can be defined as a quantum phenomenon where two or more particles become interconnected in such a way that the quantum state of each particle cannot be described independently of the others. Consequently, even if the entangled particles are separated by large distances, measuring the state of one particle instantaneously determines the state of the other particle [22, 50, 51].

Given this definition, the steps of the E91 protocol can be outlined as follows: Initially, Alice and Bob each randomly select a measurement basis. They then perform measurements on their respective entangled photons using these chosen bases. Upon completion of their measurements, Alice and Bob exchange information about the bases they used and categorize their measurement results into two distinct groups.

The first group comprises results obtained using different measurement bases, while the second group consists of results obtained with the same basis. Measurements where one or both parties fail to register are discarded. Alice and Bob then publicly disclose the measurement results from the first group, where different bases were used, while keeping the results from the second group confidential.

The results obtained from measurements performed with different bases are used to compute the Clauser-Horne-Shimony-Holt (CHSH) correlation function, defined by equation 3-2, a specific Bell inequality [52]. The violation of this inequality implies that the measured state exhibits non-local correlations, a hallmark of entangled states [53]. This demonstration of non-locality can only be achieved with entangled states.

$$S = E(a_1, b_1) - E(a_1, b_3) + E(a_3, b_1) + E(a_3, b_3) \tag{3-2}$$

Where $E(a_i, b_j)$ represents the correlation coefficient between the measurements conducted by Alice along a_i and those performed by Bob along b_j, given by:

$$E(a_i, b_j) = -a_i \cdot b_j \tag{3-3}$$

If $S = 2\sqrt{2}$ [54, 55], the subset of measurements where the same basis was used can be reliably employed to form a secure key. This ensures that the results obtained from these measurements are anti-correlated. Conversely, if the value of S deviates from this threshold, it suggests that the photons are not entangled, which could indicate the presence of an eavesdropper, Eve.

In the scenario where Eve attempts to disrupt the key distribution by replacing Alice and Bob's measurements with her own, the absence of knowledge about the chosen bases prevents her from avoiding detection. Thus, any attempt by Eve to intercept or manipulate the measurements will be revealed through the violation of the Bell inequality. Consequently, the E91 protocol provides a mechanism for verifying the security of key distribution by leveraging entanglement to detect eavesdropping attempts.

3.3 Decoy States

Although QKD is theoretically secure [56, 57], practical implementations often encounter significant challenges that can compromise this security. One prominent issue arises from the use of highly attenuated lasers as photon sources. These sources, while typically effective, may occasionally emit pulses containing multiple photons, creating a vulnerability to advanced eavesdropping techniques, such as photon splitting attacks. The intended security of QKD systems can be compromised by these real-world imperfections [58].

In essence, within the standard BB84 protocol, only the signals originating from single-photon pulses emitted by Alice can be assured of security. According to the approach outlined by GLLP [59], the secure key generation rate, per signal state emitted by Alice, can be expressed as:

$$S \geq Q_\mu \left\{ -H_2(E_\mu) + \Omega \left[1 - H_2 \left(\frac{E_\mu}{\Omega} \right) \right] \right\} \tag{3-4}$$

The variable Q_μ indicates the gain, i.e., the ratio of detection events by Bob to the total signals emitted by Alice when both use the same basis, and E_μ represents the associated QBER. The parameter Ω refers to the fraction of detection events by Bob that originate from single-photon signals emitted by Alice, while e_N indicates the QBER of an n-photon signal. Additionally, H_2 is the binary Shannon entropy [60], which measures the uncertainty in the system's information content.

In a photon splitting attack, an eavesdropper can measure the number of photons in each pulse emitted by the sender. By suppressing single-photon signals and intercepting the multi-photon pulses, Eve can potentially gain information about the key without being detected. This type of attack poses a significant threat to the security of QKD protocols, as it compromises the integrity of the key distribution process.

To address this issue, decoy state protocols have been developed. Hwang's introduction of decoy states in [61] represented a significant breakthrough, although his initial security analysis was heuristic. Subsequent advancements have refined this method, enabling its implementation with current technology [58]. These protocols introduce decoy states, signal states with varying photon number distributions that are indistinguishable from standard BB84 states to an eavesdropper.

Decoy states are used exclusively for detecting eavesdropping attempts, while standard signal states serve for key generation. The parameters Y_n and e_n are defined by the photon number n of the state. Specifically, Y_n represents the probability that an n-photon signal results in a detection event, including contributions from background noise such as dark counts and stray light. Similarly, the QBER, e_n, also depends exclusively on the photon number n of the state:

$$Y_n(signal) = Y_n(decoy) = Y_n \tag{3-5}$$

$$e_n(signal) = e_n(decoy) = e_n \tag{3-6}$$

By randomly varying the intensity of these pulses, Alice and Bob can more effectively detect the presence of an eavesdropper. Comparing the detection rates between decoy states and signal states allows for a more accurate estimation of the quantum channel's loss and error rates.

Given that the relations between the variables Q_μ and Y_n, as well as between E_μ and e_n, are linear, Alice and Bob can use the experimental values

of Q_μ and E_μ to accurately deduce the corresponding values of Y_n and e_n with high confidence. This allows them to set acceptable ranges for Y_n and e_n across all photon numbers n. Consequently, any eavesdropping attempt that significantly alters these values will be detected through the decoy state method.

To address the inefficiencies in practical error correction protocols, [58] incorporates a correction factor $f(e) > 1$ in equation 3-4, which is based on the ideal Shannon limit. This adjustment compensates for deviations from the theoretical performance, ensuring a more accurate representation of the key generation rate.

$$S \geq q\left\{-Q_\mu f(E_\mu) H_2(E_\mu) + Q_1\left[1 - H_2\left(e_1\right)\right]\right\} \tag{3-7}$$

The work presented by the authors [58] demonstrate that their decoy state method enhances the key generation rate in QKD when compared to the GLLP [59] framework. This method improves efficiency by producing higher key rates and also extends the distance over which secure QKD can be achieved, surpassing previous limitations.

3.4 MDI-QKD

The Measurement-Device-Independent Quantum Key Distribution (MDI-QKD) protocol enhances the security of Quantum Key Distribution (QKD) systems by addressing vulnerabilities associated with measurement devices, which are treated as untrusted and potentially compromised. In conventional QKD protocols, the security is heavily reliant on the assumption that the measurement devices are secure, which can be a substantial risk. MDI-QKD overcomes this limitation by employing principles of entangled photon pairs and quantum correlations to securely establish a key between remote parties, independent of the trustworthiness of the measurement apparatus.

A fundamental aspect of MDI-QKD is the elimination of the necessity for users to fully trust their measurement devices. Instead, the protocol operates on the premise that even if the measurement devices are compromised, the integrity of the key exchange can still be maintained. By utilizing a set of entangled particles and comparing measurement results between the parties, MDI-QKD effectively detects and mitigates potential eavesdropping attempts. This approach ensures that any interference from an eavesdropper can be identified, thus reinforcing the overall security of the key exchange process.

Consequently, MDI-QKD represents a significant advancement in quantum communication, offering a robust framework for secure key distribution in a complex threat landscape.

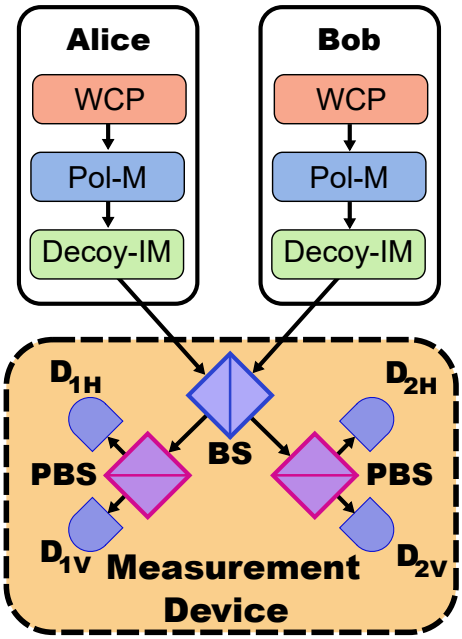

Figure 3.1: Setup of MDI-QKD.

To illustrate the protocol, the authors provide a simple example in [62]. Initially, Alice and Bob randomly prepare WCPs in one of four different polarization modes, similar to the BB84 protocol [7]. They then modulate the pulse intensity to implement decoy state techniques [58], which help estimate the relay's success probability and QBER. Alice and Bob transmit their WCPs to an untrusted relay, Charlie (or Eve), who is located between them. Charlie then performs a Bell state measurement (BSM) using linear optical elements, as shown in figure 3.1.

To enhance the accuracy of their key distribution, Alice and Bob apply decoy state techniques to estimate the gain, defined as the probability that the relay successfully outputs a result, and the QBER for different input photon numbers. After all the states have been transmitted, Charlie communicates the successful measurements through a public channel. Alice and Bob then discard any data not associated with a successful measurement, comparing the bases of the remaining states and retaining only those that match. Additionally, a bit flip is applied to the data by either Alice or Bob, except in instances where the diagonal basis was used and Charlie's successful measurement corresponds to a triplet state. This final step ensures that the shared key is correctly correlated.

The following table outlines the possible measurement outcomes, their associated probabilities and respective uses. This table clarifies the necessity

of the bit flipping process while also helping to elucidate the possible paths that Alice and Bob's pulses can take, as well as the overall performance of the protocol.

Alice Polarization	Bob Polarization	Possible Detected States	Probability of Generating a Usable Detection State	Shared Bit
↕	↔	Usable: $c_V c_H$, $d_V d_H$,$c_V d_H$,$c_H d_V$	100%	1
↔	↕	Unusable: none	100%	0
↕	↕	Unusable: $c_V c_V$, $d_V d_V$ Usable: none	0%	none
↔	↔	Unusable: $c_H c_H$, $d_H d_H$ Usable: none		
↘	↘	Usable: $c_H c_V$, $d_V d_H$	50%	0
↗	↗	Unusable: $c_H c_H$,$d_H d_H$,$c_V c_V$,$d_V d_V$		1
↘	↗	Usable: $c_H d_V$, $c_V d_H$		0
↗	↘	Unusable: $c_H c_H$,$d_H d_H$,$c_V c_V$,$d_V d_V$		1

Table 3.2: Detection outcomes and their utilities for each scenario, with d_V and d_H representing events on detectors D_{1H} and D_{1V}, and c_H and c_V on detectors D_{2H} and D_{2V}, respectively.

This protocol offers a significant security enhancement by eliminating all detector side-channel vulnerabilities, surpassing traditional methods like ILM [63] and GLLP [59]. It can potentially double the transmission distance of conventional QKD systems using weak coherent pulses, with a key generation rate that that is comparable to those achieved by standard security proofs using entangled photon pairs. Although MDI-QKD requires nearly perfect state preparation, this challenge can be effectively managed, making it a highly secure and practical solution for quantum communication.

3.5 TF-QKD

Recent advancements in optical QKD have demonstrated impressive achievements, such as key rates of 1.26 megabits per second over 50 kilometers of standard fiber and 1.16 bits per hour over 404 kilometers of ultralow-loss fiber using MDI-QKD configurations. Despite these successes, overcoming the fundamental rate-distance limit of QKD, continues to be a formidable challenge. This limit defines the maximum achievable secret key rate over a given distance, determined by the quantum channel's secret-key capacity [64, 65, 66, 67, 68].

In Twin-Field Quantum Key Distribution (TF-QKD) [69], optical fields are phase-randomized at two distant locations and then combined at a central station, where they interfere to form "twin" fields. These twins, sharing the same random phase, enable secure key generation. The key rate in TF-QKD scales with the square-root of the channel transmittance, akin to quantum repeaters but without their complex technology, making TF-QKD a viable method for extending secure quantum communications.

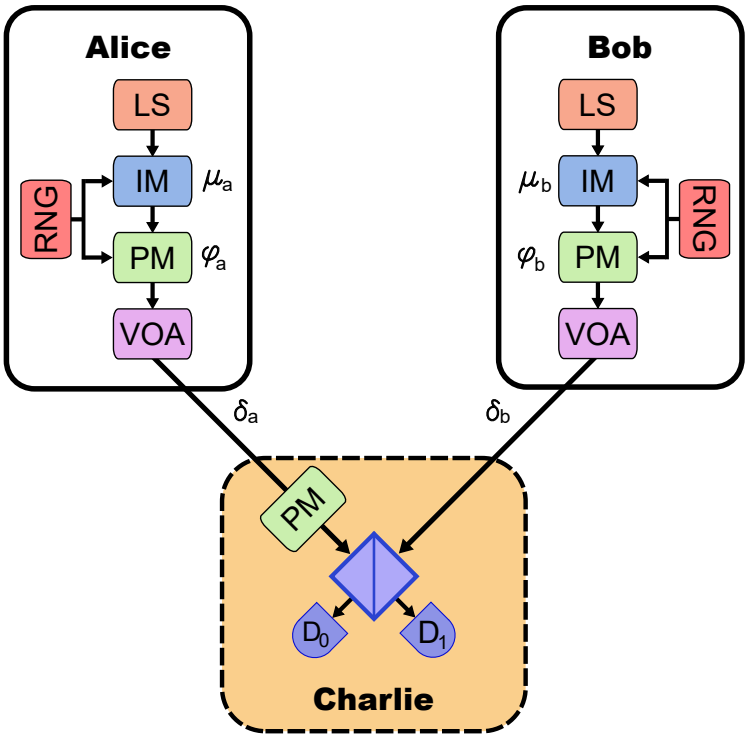

Figure 3.2: Schematic of the TF-QKD setup. Alice and Bob each use light sources (LSs) to generate pulses, which are then modulated by intensity modulators (IMs) to adjust their intensities, μ_a and μ_b, in accordance with the decoy-state technique. Phase encoding is achieved through phase modulators (PMs) combined with random number generators (RNGs), resulting in pulses with phases ϕ_a and ϕ_b. These pulses, which are either bright or dim, are regulated by variable optical attenuators (VOAs) and accumulate phase noise, denoted as δ_a and δ_b, during transmission. At Charlie's beam splitter, the pulses interfere and are detected by single-photon detectors D_0 and D_1. Charlie uses the bright pulses for phase alignment, while the dim pulses are employed for key bit extraction.

In this protocol, Alice and Bob encode their data into dim, phase-randomized optical pulses, which are sent to a central station, Charlie. At Charlie's station, these pulses interfere on a beam splitter, allowing him to determine if the secret bits are the same or different without knowing their exact values, thereby ensuring security against eavesdropping. The protocol also uses phase randomization and decoy states to improve range and performance. By publicly disclosing phase slices, matching twins can be identified, which minimizes the QBER and enhances efficiency. This approach significantly advances secure quantum communication over long distances [69].

The Rede Rio Quântica employs the CAL19 variant of the Twin-Field Quantum Key Distribution (TF-QKD) protocol [15]. This protocol employs a Sagnac interferometer to link all network users, capitalizing on the interferometer's phase stabilization and interference properties. In the CAL19 scheme,

qubits are encoded based on their propagation direction through the Sagnac interferometer, forming two orthogonal states. This approach ensures secure QKD, enhances the network's performance and scalability, making it well-suited for the Rede Rio Quântica's extensive quantum communication infrastructure.

4
Error Rate Estimation

Error rate estimation is the process of quantifying the occurrence rate of errors in a predictive system. In the context of Quantum Key Distribution, it ensures the security and reliability of the generated keys. Precise error rate estimation is indispensable for detecting and mitigating potential eavesdropping activities, enhancing the security of the communication channel. Errors in practical applications may also be caused by factors such as optical misalignment, disruptions in the quantum channel, or noise in Bob's detectors [70]. Moreover, it plays a fundamental role in optimizing QKD protocols to achieve better performance. This chapter explores an algorithm for error rate estimation, providing an analysis of its methodologies and applications within a QKD system.

The first step of the process involves generating a raw key using a specified QKD protocol. Alice prepares and transmits qubits to Bob, who measures these qubits using randomly chosen bases. Following the generation of the raw key, Alice and Bob publicly share their basis choices over a classical channel. This comparison step ensures that only the qubits measured with matching bases are preserved, as any discrepancies in basis choices lead to the corresponding bits being discarded.

The code developed in this work assumes that Alice generates and transmits a random key consisting of 1 million bits to Bob. Considering that this key was obtained after Alice and Bob have compared their measurement bases and discarded the mismatched ones, this preliminary step ensures that both parties are working with a mutually agreed upon subset of bits. It is expected that Bob's key will contain a certain percentage of errors, which is randomly distributed along the key.

To accurately estimate the error rate in Bob's key, part of the key must be sacrificed. This involves performing comparisons in random parts of Alice's and Bob's key. By analyzing this subset, the percentage of errors in the overall key can be estimated, thereby enabling the implementation of error correction protocols to ensure the security and reliability of the QKD system. The estimated QBER is calculated as follows:

$$QBER_{EST} = \frac{number\ of\ sacrificed\ mismatched\ bits}{total\ number\ of\ sacrificed\ bits} \tag{4-1}$$

The first parameter of interest is the amount of bits required to sacrifice in order to determine the error rate percentage. In the following simulation, it is considered that 5% of the key bits are flipped. These errors are randomly distributed over Bob's key. The initial simulation indicated that every iteration of error estimation resulted into different estimated error, though they were consistently close to the actual error rate. Given this variability, it is more informative to consider statistical measures of the error estimations by running multiple iterations and evaluating the outcomes.

For the following analysis, the code runs 1000 events for each sacrifice rate. Figure 4.1 illustrates the distribution of these estimations using a box plot. Each box represents essential statistical parameters, the central mark denotes the median, while the lower and upper edges indicates the 25th and 75th percentiles, respectively. The whiskers extend to encompass the most extreme non-outlier data points and outliers are distinctly marked using the cross symbol [71].

As illustrated in figure 4.1, it is notable that as the sacrifice rate increases, the standard deviation decreases, and the median tends towards the expected error. Therefore, there is a compromise relation between the amount of the key that can be sacrificed and the precision and accuracy of the error measurement. Consequently, users must balance the need for a minimum key distribution rate with the requirement for security, as both are influenced by the error estimation.

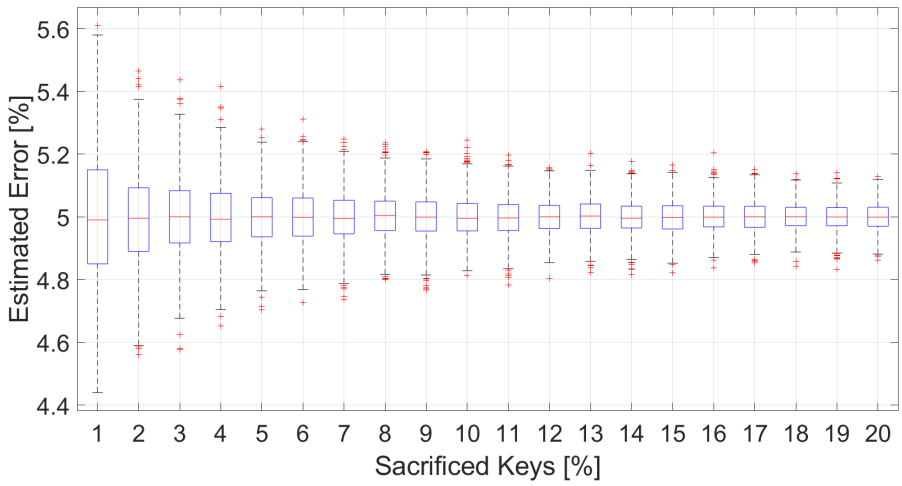

Figure 4.1: Box plot illustrating the statistical outcomes of a simulation with a 5% error rate across different sacrifice rate values.

To validate the observed compromise relation, additional simulations were performed for different error rates. It was considered error rates of 1%, 10% and 20%, with the latter representing the worst case scenario. The following figures, 4.2, 4.3 and 4.4, show the box plot for these error rates, respectively.

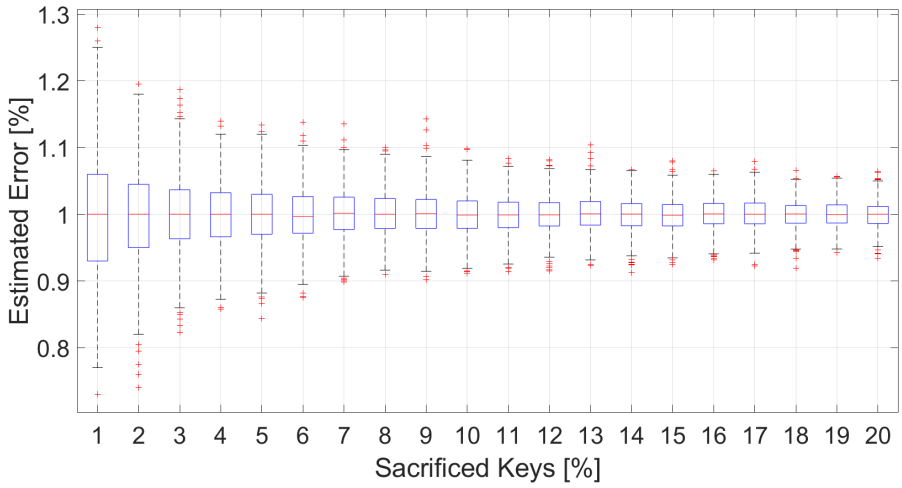

Figure 4.2: Box plot illustrating the statistical outcomes of a simulation with a 1% error rate across different sacrifice rate values.

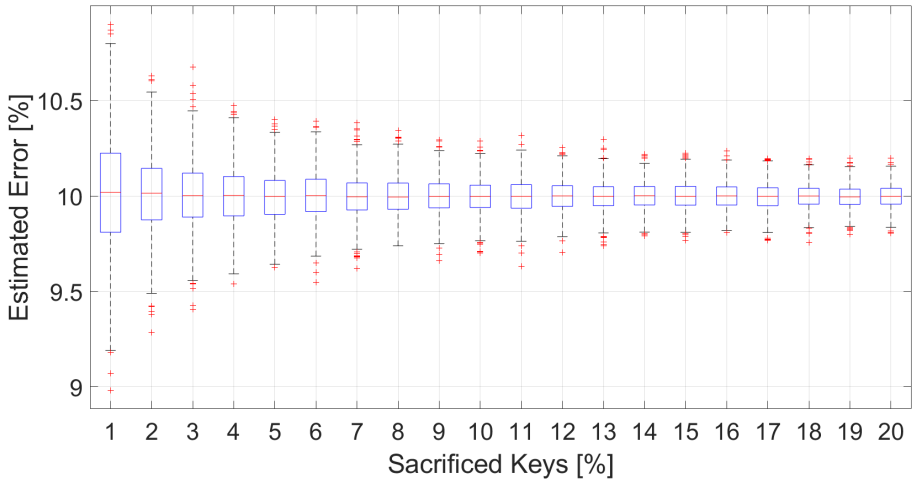

Figure 4.3: Box plot illustrating the statistical outcomes of a simulation with a 10% error rate across different sacrifice rate values.

To summarize, the coefficient of variation for each data set is analyzed and plotted, as shown in figure 4.5. Although the standard deviation increases with higher error rates, the relative uncertainty, defined as the standard deviation divided by the mean value, tends to decrease. This is evident from the box plots, which demonstrate that while deviations are larger at higher error rates, their impact relative to the mean value diminishes. Thus, while

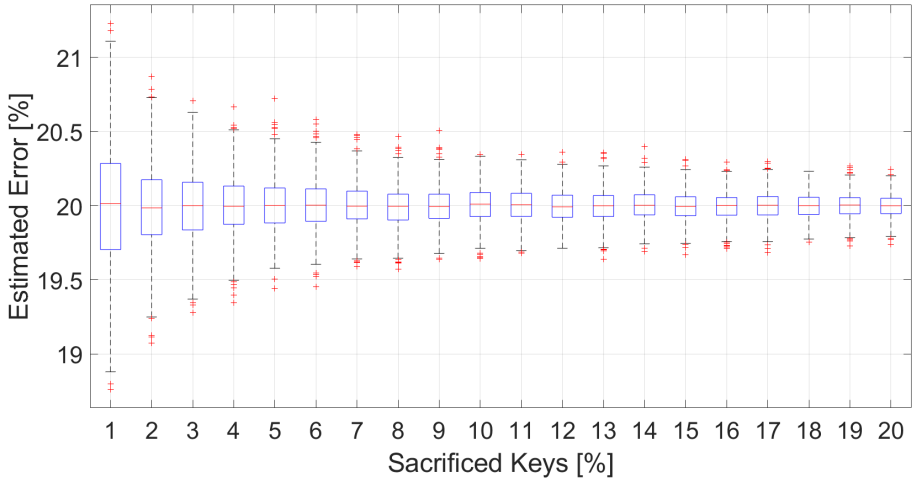

Figure 4.4: Box plot illustrating the statistical outcomes of a simulation with a 20% error rate across different sacrifice rate values.

absolute deviations grow with increasing error rates, their relative effect on overall uncertainty is reduced.

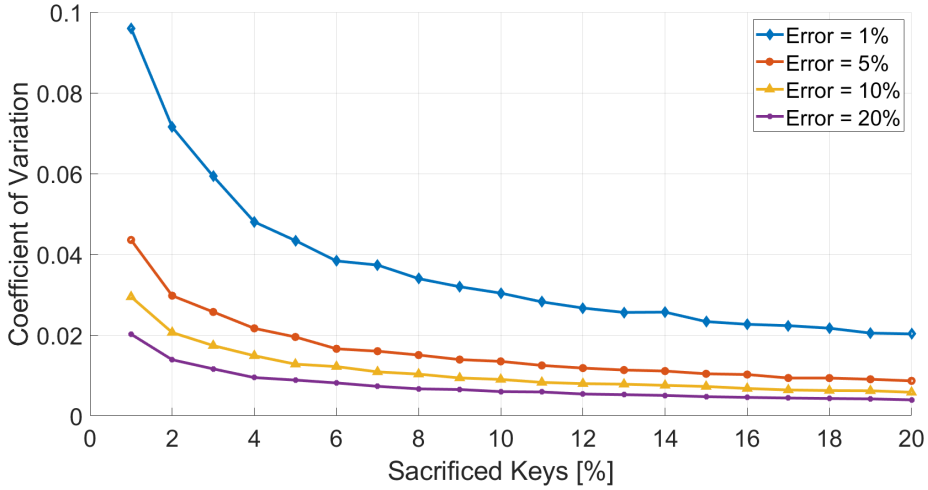

Figure 4.5: Simulation of the Coefficient of Variation across different error rate scenarios (1%, 5%, 10%, and 20%) as presented in the preceding box plots.

Examining the extreme error rate values, it is observed that the standard deviation for a 1% error rate is approximately 0.10, whereas for a 20% error rate, it decreases to around 0.02. This highlights the importance of setting reliability thresholds when designing QKD systems to define acceptable tolerance levels. Should the error rate exceed this threshold, Alice and Bob are expected to discard the key and initiate a new process. In the case of the BB84 protocol, the threshold for acceptable error rates is approximately 11%, as established by Shor and Preskill in [28].

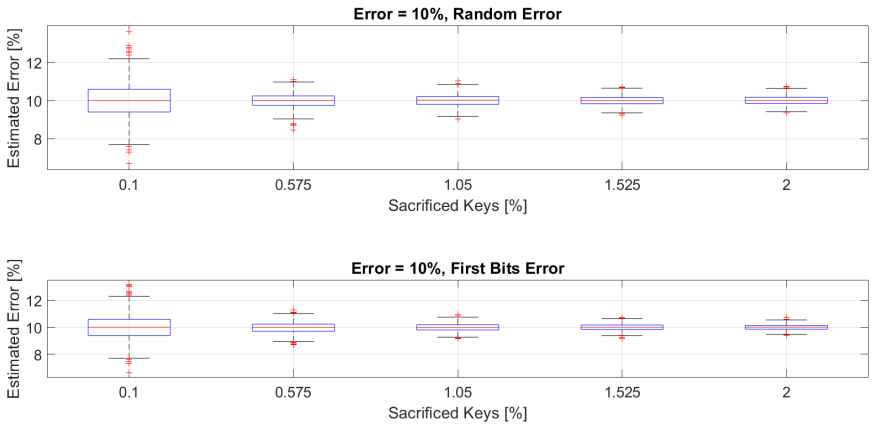

Figure 4.6: Box plot comparing two scenarios with a 10% error rate: the top graph represents scattered errors, while the bottom graph depicts burst errors.

To this point, errors have been assumed to be randomly and uniformly distributed across the key. However, if an eavesdropper, Eve, manipulates the key by targeting specific sequences of bits, this could lead to burst errors rather than random errors. In an intercept-resend attack, Eve intercepts and measures specific bits using her chosen bases, then fabricates and sends replacement pulses to Bob. This method introduces an error probability of at least 25% [70] for each bit measured by Bob in the correct basis. For simplicity, it is assumed that Eve's interference affects the initial segments of the key, leading to a concentration of errors in that specific portion. Figure 4.6 presents the box plot for a 10% error rate under both random and burst error scenarios.

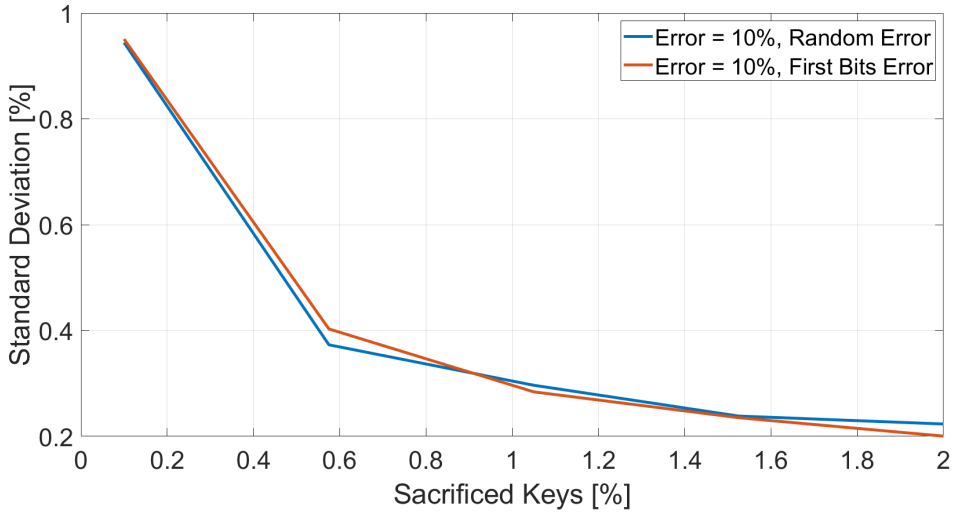

Figure 4.7: Illustration of the standard deviation calculated from statistical results under a 10% error rate, comparing scattered errors and burst errors.

In figure 4.6, it is challenging to discern significant differences between the random error and burst error scenario. However, it indicates that burst errors do not mislead the estimation process. For a more detailed comparison, the standard deviations of both scenarios are plotted together in figure 4.7. Since the result mean values are practically identical, there is no need to compare this parameter.

Based on this analysis, it can be concluded that the presence of burst errors does not impact the error estimation technique employed in this study. This indicates the robustness of the estimation method against different error distribution patterns, ensuring reliable performance under various attack strategies by an eavesdropper.

5
Error Reconciliation

As discussed in the previous chapter, if the error rate exceeds a specific threshold, the presence of an eavesdropper is detected, leading to the discarding of all measured values and the restarting of the process. If the estimated error rate is below this threshold, the process continues. However, even with an error rate below the threshold, measurement errors must be identified and corrected or discarded. This procedure of detecting and correcting discrepancies between the secret key sent by Alice and the one received by Bob is known as Error Reconciliation and is conducted over a public classical channel [72].

In the context of QKD, error reconciliation serves two primary purposes. It corrects the errors that naturally occur during the transmission of quantum states through noisy channels. Furthermore, it enables the detection and correction of errors that might be introduced by Eve, thereby reinforcing the protocol's security against potential attacks. It is recognized as a highly time-consuming and computationally intensive part of the QKD process. As demonstrated in traffic analysis experiments [73, 74], the key reconciliation step can impact the quantum channel and the key generation rate, depending on the specific implementation.

Error reconciliation protocols employ classical error correction techniques adapted for the quantum domain. Methods such as the Cascade protocol, Winnow and Low-Density Parity-Check (LDPC) are used to manage and rectify errors effectively. Each method offers unique advantages and is selected based on the specific needs of the system, such as the error rate, computational resources and the desired security level [75].

The following sections will review the most used error reconciliation approaches and conduct a comparative evaluation. This analysis will emphasize the error reconciliation in bridging the theoretical quantum cryptography and secure communication systems. Ultimately, a simulation of the selected approach will be conducted to demonstrate its application and efficacy.

5.1 Cascade

The Cascade protocol defines an iterative method for error detection and correction through the application of parity checks and binary search techniques, making it particularly effective for managing low to moderate error rates. Although this technique is employed within QKD protocols, the error correction process itself is entirely classical. The error correction relies on the binary search algorithm, also known as the binary search method [72], which is a recursive code capable of precisely correcting a single error within an odd number of errors.

The protocol operates by dividing the raw key into blocks and employing parity checks to identify discrepancies. It consists of multiple passes, each refining the error detection and correction process. In the first pass, Alice and Bob divide their keys into blocks of a predetermined size and publicly compare the parities of these blocks. When a discrepancy in parity is found, indicating an error within the block, they perform a binary search to locate and correct the erroneous bit [76].

In [72], the first pass uses blocks sized $\frac{0.73}{\text{QBER}}$, thereby increasing the probability of a block containing a single error. Alice and Bob calculate the parity for each block. Alice sends Bob her parity and Bob uses this information to calculate the parity error through a XOR operation. For even results, Bob cannot extract much information, since the block may either contain no errors or an even number of errors. However, for odd results, Bob can infer that there is at least one bit error, which is sufficient for him to apply the binary search algorithm to locate and correct the error.

The Binary Algorithm is a recursive error correcting code capable of correcting a single error within a block of bits. As a consequence, Bob can only operate on blocks with odd parity errors during this pass. The algorithm consists of dividing the parent block, identified by an odd parity error, into two sub-blocks: the left block and the right block. Then, Bob requests the parity of Alice's left block. Since the parent block's parity error is odd, the left and right blocks will have opposite error parity. This means Bob only needs the parity information for one of these blocks to determine the parity of the other. Bob then applies the binary algorithm recursively to the block with the odd parity error. This process continues until they are left with a single-bit block. At this point, Bob inverts the remaining bit, thereby correcting the single bit error [77].

For a better overview of this algorithm, consider an example where Alice and Bob have used a QKD protocol to establish a shared key and are now

performing error reconciliation using the Cascade protocol. They have divided their key into blocks of 8 bits each. To illustrate, figure 5.1 demonstrates the procedures of this example. Suppose Bob's block is 10110101, and Alice's is 10010101. Therefore, the third bit, highlighted in red, indicates the need for correction.

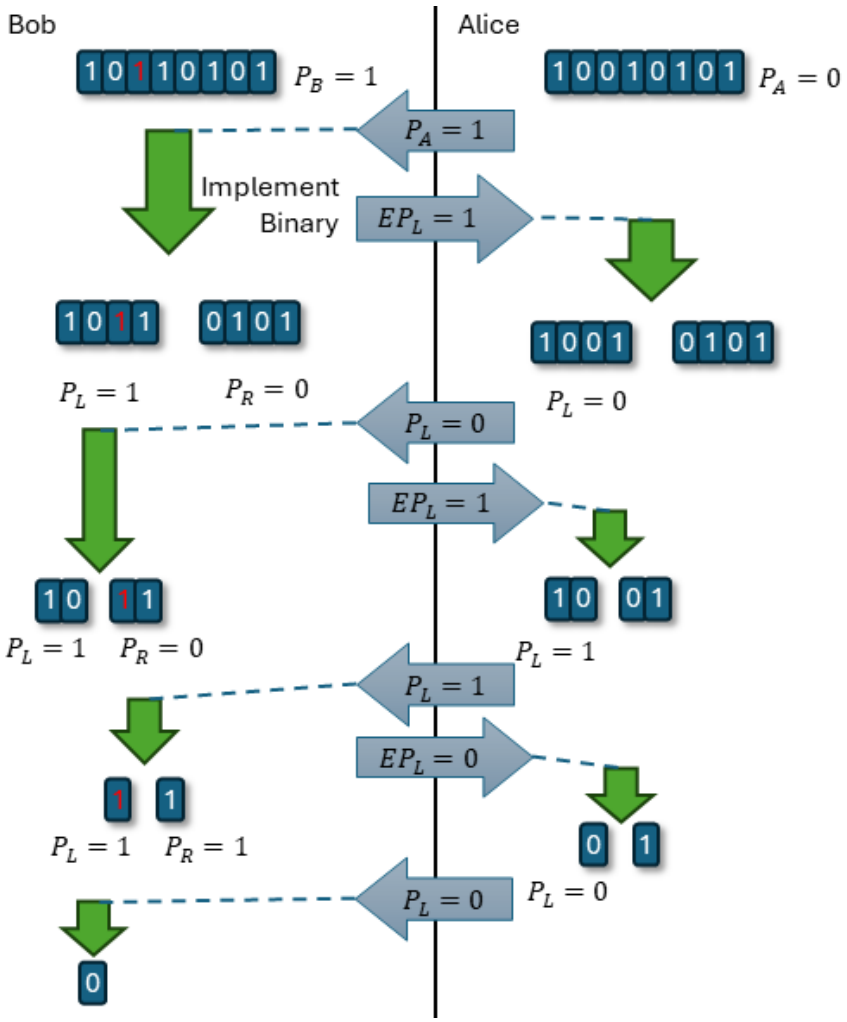

Figure 5.1: Illustration of an example of Alice and Bob using the Binary algorithm.

Both parties calculate the parity of their respective blocks. Bob finds an odd parity ($P_B = 1$), while Alice obtains an even parity ($P_A = 0$). Alice then communicates her result to Bob, who calculates the error parity as $EL = P_A$ XOR $P_B = 1$. This indicates that Bob's block contains at least one error, enabling him to initiate the Binary Algorithm for error correction. Bob informs Alice of his intention to begin error correction, splits his block into two halves, and calculates the parity of the left block (P_L). Alice performs the same calculation for her corresponding block and shares the result with Bob. Bob then computes the error parity of the left block (EP_L) and communicates the outcome to Alice.

If the error parity is 1, both Alice and Bob understand that the left block contains at least one erroneous bit. If $EP_L = 0$, while the status of the left block remains uncertain, they can conclude that the right block contains at least one error. In this scenario, since $EP_L = 1$, the process is repeated in the left block. Bob and Alice split the left block in half, calculate the parity of the new left block, and Alice communicates her result to Bob. Upon determining that the parity is 0, Bob informs Alice, and they shift their focus to the right block, as the left block's error parity is now even. In the final step, Bob identifies and corrects the erroneous bit within the block or addresses one of the odd errors, thereby aligning his key with Alice's.

After completing the first pass of error correction, Bob and Alice obtain blocks with even parity, making it impossible to apply the error correction procedure again within the same blocks. To initiate the next iteration, they randomly shuffle their keys using a shared mapping that is agreed upon through a public classical channel. With the error count reduced, they can increase the block sizes for subsequent passes. As suggested in [72], doubling the block size after each pass and conducting a total of four passes enhances the error correction capacity.

Despite the efficiency of the Cascade protocol in correcting errors, the high number of iterations required for communication between Alice and Bob causes security concerns. Although these iterations are performed rapidly, the high frequency of exchanges can lead to significant information leakage. While it ensures error correction with high accuracy, it also increases the risk of compromising the key's security. Therefore, when implementing the Cascade protocol, it is essential to balance its error correction capabilities with the associated communication overhead and potential privacy risks.

5.2 Winnow

One of the significant challenges of the Cascade algorithm for QKD error reconciliation is the unplanned leakage of small portions of the key during the iterative process. To address these concerns, the Winnow protocol was introduced in 2003, as presented in [78]. This protocol is designed to require only two rounds of communication between the communicating parties, Alice and Bob. Its objective is to increase throughput and reduce the interactivity in the Cascade protocol by eliminating the binary search step. Additionally, the Winnow protocol incorporates a privacy maintenance step, which involves discarding bits that have been leaked.

The Winnow protocol bases its error correction in Hamming Code

[79, 80]. It initiates with a parity comparison on blocks of size $N = 2^m$, where $m \in \{3, 4, 5, 6, \ldots\}$. Following this comparison, one bit is removed from each block to ensure the privacy of the remaining bits. For blocks where the parities did not match, a Hamming hash function is applied to the remaining $N-1$ bits to correct single-bit errors. Finally, m bits are discarded from the blocks where the Hamming algorithm has been applied. This step, referred to as privacy maintenance, ensures that any information potentially leaked during the error correction process is eliminated, thereby preserving the confidentiality of the key [78].

Consider a scenario where Alice generates a random key and sends it to Bob through a noisy quantum channel, which introduces errors into Bob's measurements and the sifted key. Alice and Bob then divide their respective keys into blocks of size N. According to the optimal initial block size suggested in [75], N is set to 8 bits. Alice and Bob use a parity matrix H, a specific form of the hash function [81], to calculate their respective syndromes S_A and S_B for each block. Alice transmits S_A to Bob, who compares it with his syndrome by performing a XOR operation. If the block contains no errors, the result is the zero vector. In the event of a single error within the block, the XOR operation reveals the error's position. Bob then applies a NOT gate to the identified bit position to correct the error. Below is an example of how the Hamming protocol identifies an error:

$$\begin{aligned} S_A &= H \cdot M_A \\ &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{5-1}$$

$$\begin{aligned} S_B &= H \cdot M_B \\ &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \end{aligned} \tag{5-2}$$

$$S_d = S_A \times S_B = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \tag{5-3}$$

Consequently, Bob determines that the erroneous bit is located at position 110, which corresponds to the 6th bit in the block.

It is important to note that the Hamming code's capacity is restricted to correcting blocks with a single bit error, which necessitates the use of the smallest possible block sizes at the start of the Winnow protocol. Moreover, a significant drawback of the Winnow protocol arises from its dependence on Hamming codes, this reliance can potentially introduce errors during the error correction process [78]. Although Winnow operates considerably faster than the Cascade protocol, its efficiency diminishes for error rates below 10%, a common scenario in practical QKD implementations [82].

5.3 Low Density Parity Check

Low-Density Parity-Check (LDPC) codes represent a class of linear block codes characterized by a sparse parity-check matrix, i.e., a matrix composed of many zeros and a few ones [83]. These codes provide error correction near-ideal performance that approaches the limits established by Shannon [60].

In parity-check codes, codewords are generated by combining a block of binary information digits with a block of check digits. The check digits are calculated as a binary sum of a predefined set of information digits [84]. The sparsity of the parity-check matrix enables efficient encoding and decoding processes. Decoding is typically performed using iterative algorithms, such as Sum-Product Algorithm (SPA).

After the initial raw key generation and basis reconciliation phases, Alice and Bob share a sifted key that includes bits measured with matching bases. However, this sifted key still contains errors due to quantum noise and potential eavesdropping. A portion of the sifted key is sacrificed to estimate the average QBER. This step helps in determining the amount of errors present in the key.

Alice and Bob agree on a parity check matrix, H, before starting the protocol, such that only they know. Therefore, the message length M and code length N are defined. During the reconciliation stage, they split the remaining key into M-bit blocks. Each block is then encoded using G, the generator matrix.

$$\begin{aligned} c_A &= m_A G \\ c_B &= m_B G \end{aligned} \tag{5-4}$$

Where c_A and c_B are $1 \times N$ coded block, and m_A and m_B are $1 \times M$ message blocks from Alice and Bob, respectively. The Generator matrix G is a $M \times N$ matrix calculated as follows:

$$G = \left[H_2 H_1^{-1} | I_{M \times M} \right], \tag{5-5}$$

H_2 and H_1 are obtained by splitting the $N \times N - M$ parity check matrix H.

$$H^T = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \tag{5-6}$$

H_1 is a $N - M \times N - M$ matrix and H_2 is $M \times N - M$. The ratio $\frac{M}{N}$ determines the code rate.

After encoding, Alice transmits the codewords through a classical channel. The confidentiality of the parity-check matrix, known only by Alice and Bob, ensures the protocol's reliability [85]. Upon receiving Alice's codewords, Bob employs the sum-product algorithm in the logarithmic domain. This technique begins with initializing the logarithmic coefficients $|LQ_{ij}^x|$ to the initial logarithmic probabilities of the symbols $|Lf_{ij}^x|$. Where x represent either symbol 0 or 1.

Next, Bob initiates the horizontal steps, calculating the logarithmic coefficients $|LR_{ij}^x|$ for each pair (i, j) using the following equations:

$$|LR_{ij}^0| = \ln(2) \mp |\ln(1 \pm |e^{L\delta R_{ij}}|)| \tag{5-7}$$

$$|LR_{ij}^1| = \ln(2) \pm |\ln(1 \mp |e^{L\delta R_{ij}}|)| \tag{5-8}$$

The signal in the expression is dependent on whether δR_{ij} is even or odd. this parameter is obtained by:

$$|L\delta R_{ij}| = \sum_{j' \in N(i) \setminus j} |L\delta Q_{ij'}| \tag{5-9}$$

In vertical steps, similar to horizontal steps, but the values of the logarithmic coefficients $|LQ_{ij}^{x}|$ are calculated for each pair (i,j). This process uses the following equations:

$$|LQ_{ij}^{x}| = |Lc_{ij}^{x}| - \min(Lc_{ij}^{0}, LC_{ij}^{1}) + \left|\ln\left(1 + |e^{-|LQ_{ij}^{0} - LQ_{ij}^{1}|}\right)\right| \tag{5-10}$$

Here, $|Lc_{ij}^{x}|$ is calculated by:

$$|Lc_{ij}^{x}| = |Lf_{ij}^{x}| + \sum_{i' \in M(j) \setminus i} |LR_{ij}^{x}|. \tag{5-11}$$

Subsequently, an estimation of each symbol $\hat{d}_j$ is obtained. This iteration repeats until the predefined maximum number of iterations is reached. The explanation provided here represents a basic simplification of the Sum-Product Algorithm (SPA) in the logarithmic domain [83].

Therefore, after the maximum number of iterations, Bob expects to have the original codewords that Alice transmitted. He then compares his received codewords to those expected and corrects any mismatches. If the SPA decoding technique has successfully corrected the errors, the resulting average QBER should ideally be zero.

Although decoding LDPC codes involves substantial computational and memory requirements, it manages large matrices while delivering high precision and computational efficiency. Moreover, the process necessitates only a single information exchange and does not require excessive equipment or computational resources.

5.4 Simulation of Cascade Protocol

In general, the optimal error reconciliation protocol would fix all bit errors within each block, avoid creating new errors, and disclose as little information as possible about the key bits to an eavesdropper during public communication [78]. Each error reconciliation protocol used in QKD systems presents its unique advantages and disadvantages. Protocols such as Cascade, Winnow and LDPC codes vary in their approach to error detection and correction, computational complexity, and communication overhead. These

differences make them suitable for different scenarios and requirements of implementations.

Decoding LDPC codes involves greater computational and memory demands than the Cascade or Winnow protocols. However, this increased complexity is offset by a significant reduction in communication overhead, as LDPC codes require only a single round of information exchange. This is particularly advantageous in network environments where bandwidth and latency are constrained, making efficient communication essential.

The Cascade protocol is simple and effective at low to moderate error rates but is vulnerable to information leakage due to its iterative process. The Winnow protocol improves throughput by reducing the number of interactions required but also faces potential leakage problems.

For this study, the Cascade protocol was selected due to its proven efficacy in managing low to moderate error rates and its straightforward implementation. Despite the potential risk of information leakage, this concern will be addressed using privacy amplification techniques. These techniques are designed to mitigate the risk of leakage and enhance the security of the key against potential eavesdropping attempts.

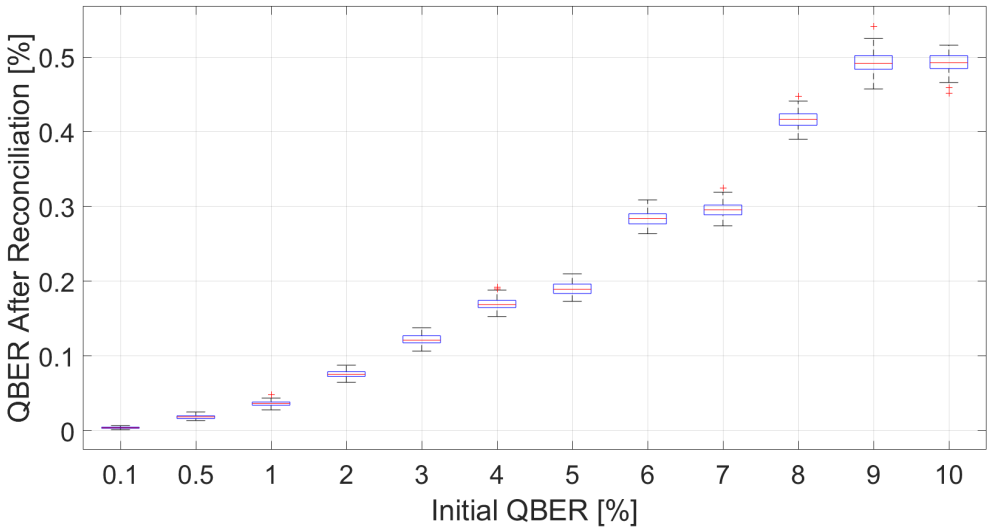

Figure 5.2: Simulation of the Cascade protocol for different values of QBER.

To simulation purposes, it is assumed that Alice generates and sends the key to Bob through a quantum channel. The basis reconciliation and error estimation have already been completed, meaning that both parties share a sifted key and have an estimated QBER. Following [72], the protocol undergoes four passes for error reconciliation. This scenario is simulated for various QBER values, ranging from 0.1 to 10. The simulation executes 100 events for each QBER value, as illustrated in figure 5.2.

Additionally, it is interesting to investigate the cascading effect inherent in this protocol by simulating different numbers of passes and revisiting previously completed passes to determine how these adjustments enhance error correction efficiency. For this simulation, a fixed QBER value of 5% is assumed, and multiple scenarios are evaluated. Figure 5.3 illustrates the resulting average QBER after implementing different orders and numbers of passes. This analysis aims to demonstrate the impact of the iterative process on error reduction.

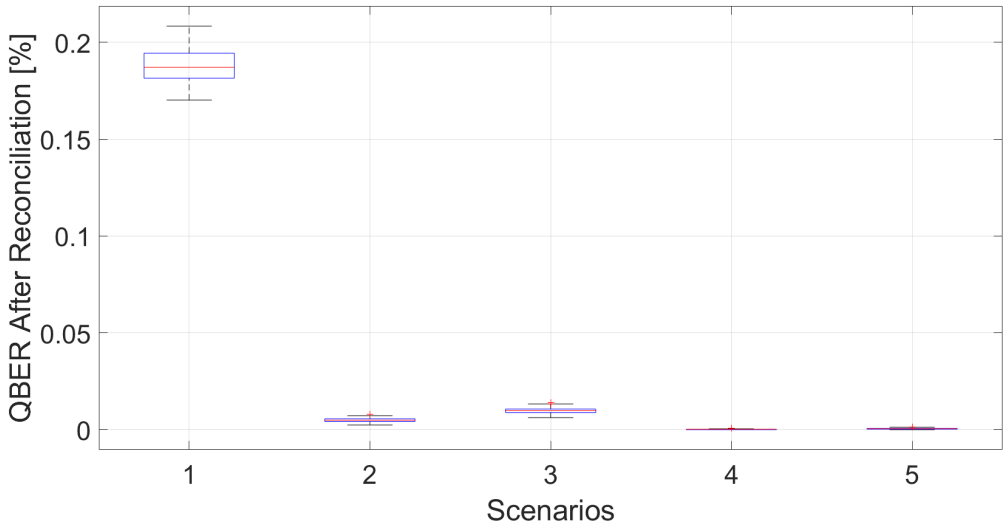

Figure 5.3: Simulation of the Cascade protocol for different scenarios of passes considering a QBER of 5%.

Scenario 1 follows the standard procedure of passes 1 through 4 as detailed in Section 5.1. In Scenario 2, the standard procedure is followed, after which pass 1 is repeated. This addition notably reduces the mean error value by almost tenfold. However, it significantly increases the number of iterations due to the smaller block size in pass 1. In Scenario 3, where pass 2 is repeated after the standard procedure, the reduction in error is less effective compared to the repetition of pass 1. This observation suggests that adding passes 3 or 4 would be progressively less beneficial, given their larger block sizes. Nonetheless, introducing additional passes after these larger blocks can enhance error correction while maintaining a lower number of iterations.

In Scenario 4, the addition of passes 3 and 2 was tested, respectively. This configuration yielded a mean error rate close to zero, outperforming Scenario 2 and demonstrating greater efficiency due to fewer iterations compared to repeating pass 1. Scenario 5 explores another variation by running passes 3 and 4 after the standard procedure. Although the mean error rate in this scenario is close to that in Scenario 4, it is slightly higher. While it requires fewer iterations.

In this section, the Cascade protocol was simulated and tested under various parameters. The results indicated a trade-off between the protocol's performance and the permissible number of iterations. Furthermore, as discussed in Section 5.1, the frequency of interactions between Alice and Bob during this protocol correlates with the amount of key leakage. Consequently, the allowable number of iterations is directly related to the privacy amplification process discussed in the next chapter, which will determine the extent of Eve's knowledge about the key, ensuring its reliability.

6
Privacy Amplification by Public Discussion

In Quantum Key Distribution, privacy amplification is essential for eliminating any residual information that an eavesdropper, might have obtained through various attacks or noise in the communication channel. Furthermore, even after error reconciliation, some information may be compromised due to the exchange of key information between Alice and Bob over a public channel [86].

Introduced in 1988 [87], this technique transforms a partially secure key into a fully secure one by reducing the information available to the eavesdropper to insignificant levels. Eve might gain partial information about the secret key by intercepting the quantum communication channel and eavesdropping on the public reconciliation discussions. To eliminate this compromised information, a privacy amplification algorithm is employed, ensuring the secrecy and integrity of the final key [72].

The process involves Alice and Bob agreeing on a public hashing function to distill a shorter, but highly secure, key from the initially shared partially secure key. The theoretical basis relies on the properties of universal hash functions [88, 89], which guarantee that any partial information Eve has about the original key will be exponentially reduced after the hashing process.

This chapter explores the principles of privacy amplification and the concept how it can protect the key against an eavesdropping. Moreover, it examines the implementation of a most common privacy amplification technique. This simulation combines all concepts from the previous chapters and allow us to obtain the final final goal for the QKD protocol, the secret key sharing between two parties.

6.1
Introduction

The main principles behind privacy amplification involve correctness, secrecy and the eavesdropper's uncertainty. To grasp these ideas, it is essential to understand the broader context of a QKD protocol, which provides the framework for their application. Consider a scenario in which Alice and Bob aim to share a secret key via a QKD protocol. Their quantum and classical

channel may be intercepted by an eavesdropper. It is assumed that Eve possesses unlimited technological capabilities and access to both quantum and classical channels, striving to extract as much information as possible from their key.

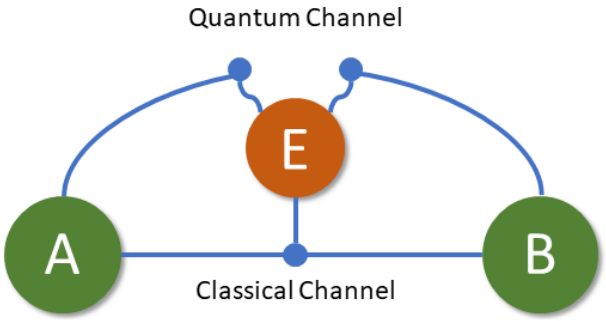

Figure 6.1: Basic QKD scheme where parties A and B aim to share a secret key, while an eavesdropper E attempts to intercept information during the key exchange.

After Alice and Bob have completed all steps of the QKD protocol, Bob holds a sifted and corrected key initially prepared and sent by Alice. To minimize any potential eavesdropper's knowledge, they must apply a method adhering to three essential parameters: correctness, secrecy and eavesdropper uncertainty. Correctness ensures that Alice and Bob have the same result. Secrecy dictates that Eve's information about the key is reduced compared to the initial state. Eavesdropper uncertainty sets a limit on Eve's minimum knowledge required for process success [86]. This final step must enhance the key's entropy and the conditional entropy of the shared key and Eve's key without introducing errors. Consequently, Alice and Bob cannot use a predefined function, as Eve could anticipate this and select her attacks to maximize information gain.

In computer science, the concept of a randomness extractor, or simply an extractor, is critical for transforming weak sources of randomness into outputs that approximate a uniform distribution. This transformation effectively increases the entropy of the original input [90]. Specifically, in the context of privacy amplification, focus is directed towards a subset of randomness extractors that leverage universal$_2$, or dual universal$_2$, hash functions [91]. These hash functions constitute a special family characterized by the property that for a given family of functions $H = \{f|X \to Y\}$ any two distinct inputs x_1 and x_2 yield distinct outputs for more than $1/dim(Y)$ of the functions within the family. This characteristic makes universal$_2$ hash functions particularly valuable in cryptographic applications, including privacy amplification. By reducing the dimensionality of the output to slightly less than the eavesdropper's knowledge

of the key, these functions enhance the security of the cryptographic process [86].

6.2 Simulation

The concept of privacy amplification is very broad, since it can be used for any QKD protocol. As QKD technology progresses, the implementation of privacy amplification becomes increasingly vital. Having this in mind, this section delves into the application of privacy amplification techniques within the context of previously conducted simulations, thereby completing the comprehensive post-processing of a QKD protocol.

In the scenario under consideration, represented in Figure 6.1, Alice and Bob employ the BB84 protocol and are currently engaged in the privacy amplification phase. After the initial key exchange, Alice has prepared and transmitted quantum states to Bob. They have shared their basis choices and discarded any incompatible selections. Additionally, a portion of the key has been sacrificed to estimate the error rate, and an error correction protocol has been employed to correct any discrepancies in the key bits [92].

To enhance the secrecy of the protocol, it is crucial for Alice and Bob to select a family of universal hash functions. They utilize these functions by sharing a small seed, which is shorter than their current shared key. Research indicates that modified Toeplitz matrices serve as an efficient family of dual universal hash functions suitable for QKD applications [93, 94, 95]. The structure of a Toeplitz matrix [96] is defined by the rule $T_{ij} = t_{i-j}$, resulting in the following matrix representation:

$$T = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & \cdots & a_{1-n} \\ a_1 & a_0 & a_{-1} & \ddots & \ddots & \vdots \\ a_2 & a_{-1} & a_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_{-2} \\ \vdots & \ddots & \ddots & \ddots & a_0 & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_2 & a_1 & a_0 \end{bmatrix} \tag{6-1}$$

The modified Toeplitz matrix is defined by concatenating the original matrix with the identity matrix, represented as $T' = [I\,T]$. To randomly generate this matrix, Alice and Bob must share a seed. This seed is used to create the first row and column of the matrix. Once the matrix is constructed, they apply a mathematical operation to derive their final secret key, denoted

as K:

$$K_{k\times 1} = I_{k\times k} R_{k\times 1} \bigoplus T(S)_{k\times r-k} R_{r-k\times 1} \tag{6-2}$$

The variables in the expression are defined as follows: R represents the column vector containing the keys after error reconciliation and r is the size of this vector. The $\bigoplus$ operation, commonly referred to as the XOR operation in binary arithmetic, represents a modulo 2 addition, represents a modulo 2 addition. The variable S denotes the seed used to generate the Toeplitz matrix, while k signifies the size of the secret key. In this study, the size of the random seed is determined by the following expression:

$$S = max(\{r - k, k\}) \tag{6-3}$$

It is essential to define the secret key length or the resulting dimension of the hashing. This parameter has been the subject of considerable debate in numerous studies. In this research, the secret key rate is determined according to the definitions provided by [28, 97]:

$$k = r[1 - H(Q) - \eta_{ec} H(Q)] \tag{6-4}$$

In this context, $H(\cdot)$ refers to the Shannon entropy [60] modulo 2 and η_{ec} represents the efficiency of the error correction protocol. It is important to highlight that $\eta_{ec} \leq 0$. For the simulations conducted, since it is not feasible to compute this efficiency during the protocol, the worst-case scenario for the Cascade protocol was considered, with $\eta_{ec} = 1.24$ [98].

The simulation employed the previously described processes to analyze the secret key ratio across a range of initial QBER values. For each QBER, 50 events of the BB84-based QKD protocol were executed. The results, shown in Figure 6.2, present a box plot of the ratio between the secret key and the initial key. This analysis allows for theoretical calculations of the secret key ratio using predefined equations and considering that, on average, half of the initial key is discarded during basis reconciliation. The calculation of a theoretical secret key ratio is given by the equation bellow.

$$\frac{K_{est}}{R_{init}} = \frac{1}{2}(1 - \eta_{sac})[1 - H(Q) - \eta_{ec} H(Q)] \tag{6-5}$$

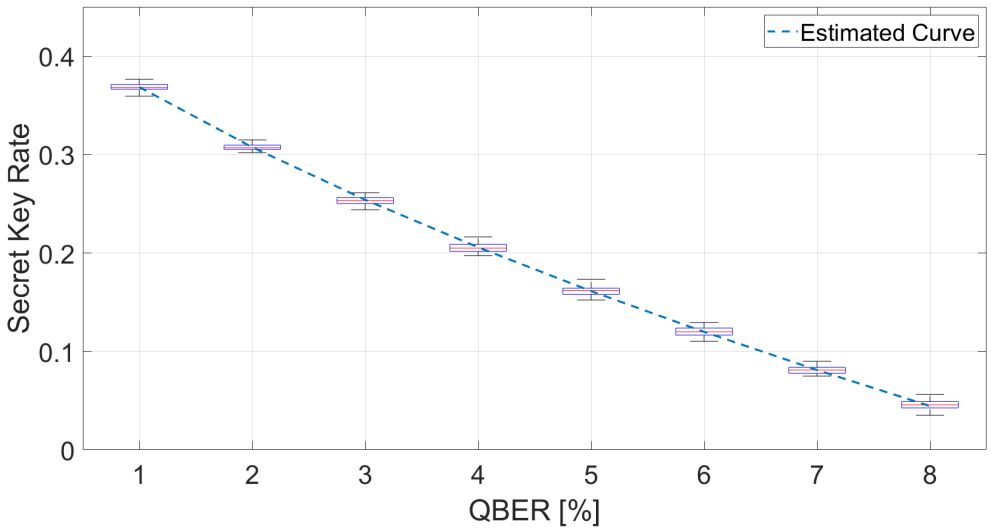

Figure 6.2: Simulation of the secret key rate for various initial QBER values, with a 10% sacrifice rate applied for error estimation.

In these calculations, the parameter η_{sac} represents the percentage of the key sacrificed for error estimation. The theoretical curve depicted in Figure 6.2 aligns closely with the mean values of the box plots, demonstrating the simulation's effectiveness as a tool for estimating the secret key length in a QKD protocol. One can calculate the QBER using data and measurements from the channel, as described in section 2.4, and further adjust the simulation parameters to explore potential enhancements in the secret key rate.

This approach validates the theoretical predictions and also provides a practical framework for experimenting with different parameters and strategies to optimize the secret key rate. Through this process, researchers and practitioners can better understand the impacts of various protocol configurations and channel conditions, ultimately leading to more secure and efficient quantum communication systems.

7
Conclusion and Future Work

This work provides an analysis of post-processing in QKD, exploring the concepts and procedures involved at each stage. The analysis includes a detailed examination of error estimation and reconciliation processes, with particular emphasis on the Cascade protocol, chosen for its efficiency and speed in resolving discrepancies in the raw key. Furthermore, a privacy amplification algorithm was developed to enhance security by minimizing potential eavesdropper information. The research outlines the theoretical foundations and translates them into practical application through the development of a corresponding code. This implementation, developed in MATLAB, effectively integrates the discussed protocols and algorithms, offering a practical solution for ensuring secure quantum communication.

To ensure that the estimated key generation rate is both practical and secure, it is essential to perform post-processing steps. This post-processing phase involves the processes of error correction and privacy amplification. Error correction reconciles discrepancies in the raw key caused by noise and imperfections in the quantum channel, ensuring that both parties share an identical key. Privacy amplification, on the other hand, mitigates any potential knowledge that an eavesdropper might have gained, even after error correction, by further securing the key and reducing any partial information available to unauthorized parties.

By employing mathematical techniques such as universal hash functions, privacy amplification transforms the reconciled key into a secure form with negligible eavesdropper knowledge. This post-processing phase is essential for achieving the desired security levels in quantum cryptographic systems, as it protects the keys against both technical imperfections and potential adversarial attacks.

In the preceding chapters, equation 6-5 was adapted for the QKD protocol to estimate the average secret key generation rate. The final objective of this work was to use the tools developed in the previous chapters to estimate the key generation rate for Rede Rio Quântica. The first modification involved redefining R_{init} as R_{sift}, as outlined in equation 2-18. In this context, q is set to 1, f_{rep} is 50 MHz, μ is 1 photon per pulse, considering decoy states as in

[99], $\eta = 10\%$ and t_{link} for RRQ is given by the following equation:

$$t_{link} = 10^{\frac{-\alpha 2L}{10}} (t_a t_b)^{1/2} \tag{7-1}$$

Where, L represents the distance between Alice and Bob, α is the fiber attenuation coefficient (0.2 dB/km), t_a and t_b are the insertion losses in Alice's and Bob's systems, respectively. Empirical tests by the RRQ team measured these losses to be around 13 dB, approximately 5% transmission efficiency.

The final parameter necessary for the simulation is the QBER, which was estimated using equation 2-22. To simplify the calculations, $QBER_{opt}$ term was excluded due to the fact that $QBER_{det} \gg QBER_{opt}$. The system in question employs two detectors, each with a dark count rate of 3×10^{-6} counts per pulse. Using the developed tools and equations, the key generation rate was evaluated for various distances between Alice and Bob. The results are presented in figure 7.1. Given that the RRQ link has an extension of 27 km, this distance was used for the simulation. For this distance, the estimated secret key generation rate between Alice and Bob is approximately 2300 keys per second. This estimation reflects the effective use of the QBER and other parameters in determining the practical key rate achievable under the given conditions.

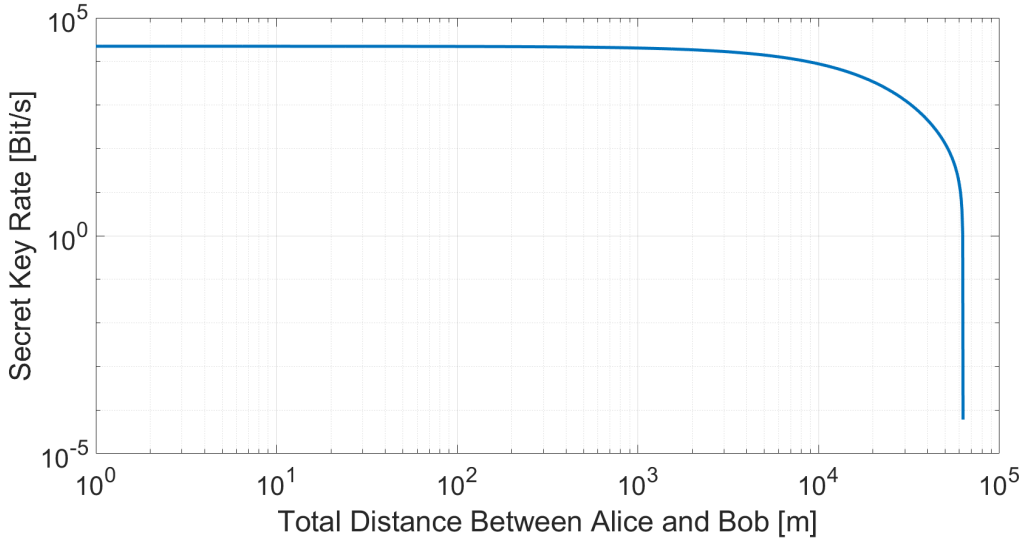

Figure 7.1: Simulation results for key generation in Rede Rio Quântica.

For future research, it would be valuable to explore alternative methods for privacy amplification, given that the final key size considered in this study's simulations is the smallest possible. This exploration could examine scenarios where an eavesdropper may have some information, but not enough to fully

compromise the key's security. Additionally, assessing the frequency at which keys can be reused without undermining security would provide important insights. The initial assumption that the encrypted message size is equivalent to the key size results in a relatively low transmission rate of 2 kbits per second, which warrants further investigation.

Moreover, comparing the theoretical results obtained in this work with empirical data from RRQ could provide validation. For instance, laboratory measurements over varying distances within the RRQ would help assess the practical accuracy of the presented simulations. Another key aspect for future research is to validate the proposed methods by measuring the actual key generation rate of the RRQ and determining whether it aligns with the simulation results discussed in this thesis, thereby ensuring that the theoretical models accurately reflect real-world performance.

Bibliography

[1] A. Gupta and N. Walia, "Cryptography algorithms: A review," *International Journal of Engineering Development and Research 2321-9939*, vol. 2, p. 1667, 01 2014.

[2] G. Brassard, *Modern Cryptology*, ser. Lecture Notes in Computer Science. Springer-Verlag, 1988, vol. 325.

[3] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, p. 1484–1509, Oct. 1997. [Online]. Available: http://dx.doi.org/10.1137/S0097539795293172

[4] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics*, vol. 74, no. 1, p. 145–195, Mar. 2002. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.74.145

[5] R. de Wolf, "The potential impact of quantum computers on society," 2017. [Online]. Available: https://arxiv.org/abs/1712.05380

[6] S. Wiesner, "Conjugate coding," *SIGACT News*, vol. 15, no. 1, p. 78–88, jan 1983. [Online]. Available: https://doi.org/10.1145/1008908.1008920

[7] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *the IEEE International Conference on Computers, Systems and Signal Processing*, Banglore, India, 1984.

[8] H. de Riedmatten, I. Marcikic, J. A. W. van Houwelingen, W. Tittel, H. Zbinden, and N. Gisin, "Long-distance entanglement swapping with photons from separated sources," *Phys. Rev. A*, vol. 71, p. 050302, May 2005. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.71.050302

[9] J. L. Park, "The concept of transition in quantum mechanics," *Found Phys*, vol. 1, 1970.

[10] D. Gottesman, T. Jennewein, and S. Croke, "Longer-baseline telescopes using quantum repeaters," *Physical Review Letters*, vol. 109, no. 7, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.109.070503

[11] E. O. Ilo-Okeke, L. Tessler, J. P. Dowling, and T. Byrnes, "Remote quantum clock synchronization without synchronized clocks," *npj Quantum Information*, vol. 4, no. 1, Aug. 2018. [Online]. Available: http://dx.doi.org/10.1038/s41534-018-0090-2

[12] B. Hensen, H. Bernien, A. Dréau, A. Reiserer, N. Kalb, M. Blok, J. Ruitenberg, R. Vermeulen, R. Schouten, C. Abellan, W. Amaya, V. Pruneri, M. Mitchell, M. Markham, D. Twitchen, D. Elkouss, S. Wehner, T. Taminiau, and R. Hanson, "Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres," *Nature*, vol. 526, 10 2015.

[13] C. Simon, "Towards a global quantum network," *Nature Photonics*, vol. 11, pp. 678–680, 2017. [Online]. Available: https://doi.org/10.1038/s41566-017-0032-0

[14] H. Kimble, "The quantum internet," *Nature*, vol. 453, pp. 1023–1030, 2008. [Online]. Available: https://doi.org/10.1038/nature07127

[15] M. Curty, K. Azuma, and H.-K. Lo, "Simple security proof of twin-field type quantum key distribution protocol," 2018. [Online]. Available: https://arxiv.org/abs/1807.07667

[16] G. Sagnac, "L'ether lumineux démontré par l'effet du vent relatif d'éther dans un interféromètre en rotation uniforme," *Comptes Rendus*, vol. 157, pp. 708–710, 1913.

[17] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge: Cambridge University Press, 2010.

[18] C. Cohen-Tannoudji, B. Diu, and F. Laloë, *Quantum Mechanics*. John Wiley & sons, 2005, vol. 1.

[19] P. Blanchard and E. Brüning, *Hilbert Spaces: A Brief Historical Introduction*. Cham: Springer International Publishing, 2015, pp. 201–212. [Online]. Available: https://doi.org/10.1007/978-3-319-14045-2_14

[20] P. A. M. Dirac, "A new notation for quantum mechanics," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 35, no. 3, p. 416–418, 1939.

[21] E. Schrödinger, "Quantisierung als eigenwertproblem (erste mitteilung)," *Annalen der Physik*, vol. 79, no. 4, pp. 361–376, 1926.

[22] C. Gerry and P. Knight, *Introductory Quantum Optics*. Cambridge University Press, 2005.

[23] M. O. Scully and M. S. Zubairy, *Quantum Optics*. Cambridge University Press, 1997.

[24] L. Mandel and E. Wolf, *Optical Coherence and Quantum Optics*. Cambridge University Press, 1995.

[25] T. F. da Silva, G. C. do Amaral, D. Vitoreti, G. P. Temporão, and J. P. von der Weid, "Spectral characterization of weak coherent state sources based on two-photon interference," *J. Opt. Soc. Am. B*, vol. 32, no. 4, pp. 545–549, Apr 2015. [Online]. Available: https://opg.optica.org/josab/abstract.cfm?URI=josab-32-4-545

[26] M. Fox, *Quantum Optics: An Introduction*. Oxford: Oxford University Press, 2006.

[27] N. Lütkenhaus and M. Jahma, "Quantum key distribution with realistic states: photon-number statistics in the photon-number splitting attack," *New Journal of Physics*, vol. 4, no. 1, p. 44, jul 2002. [Online]. Available: https://dx.doi.org/10.1088/1367-2630/4/1/344

[28] P. W. Shor and J. Preskill, "Simple proof of security of the bb84 quantum key distribution protocol," *Phys. Rev. Lett.*, vol. 85, pp. 441–444, Jul 2000. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.85.441

[29] P. Hariharan, *Optical Interferometry*. San Diego, CA: Academic Press, 2003.

[30] B. E. A. Saleh and M. C. Teich, *Fundamentals of photonics; 2nd ed.*, ser. Wiley series in pure and applied optics. New York, NY: Wiley, 2007. [Online]. Available: https://cds.cern.ch/record/1084451

[31] H. Fearn and R. Loudon, "Quantum theory of the lossless beam splitter," *Optics Communications*, vol. 64, no. 6, pp. 485–490, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0030401887902756

[32] A. Agnesi and V. Degiorgio, "Beam splitter phase shifts: Wave optics approach," *Optics & Laser Technology*, vol. 95, pp. 72–73, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030399217300634

[33] G. Pavlath, "Fiber-optic gyroscopes," in *Proceedings of LEOS'94*, vol. 2, 1994, pp. 237–238 vol.2.

[34] A. Lawrence, *Modern Inertial Technology: Navigation, Guidance and Control*. Springer, 1998.

[35] L. Zehnder, "Ein neuer interferenzrefraktor," *Zeitschrift für Instrumentenkunde*, vol. 11, pp. 275–285, 1891.

[36] L. Mach, "Ueber einen interferenzrefraktor," *Zeitschrift für Instrumentenkunde*, vol. 12, pp. 89–93, 1892.

[37] L. Li, L. Xia, Z. Xie, and D. Liu, "All-fiber mach-zehnder interferometers for sensing applications," *Opt. Express*, vol. 20, no. 10, pp. 11 109–11 120, May 2012. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-20-10-11109

[38] J. Švarný, "Analysis of quadrature bias-point drift of mach-zehnder electro-optic modulator," in *2010 12th Biennial Baltic Electronics Conference*, 2010, pp. 231–234.

[39] J. R. Da Silva, J. Batista Rosa Silva, and R. V. Ramos, "Approaching single-photon pulses with weak coherent states and nonlinear phase modulation," in *2021 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, 2021, pp. 1–3.

[40] A. Michelson and E. Morley, "On the relative motion of the earth and the luminiferous ether," *American Journal of Science*, vol. 34, no. 203, pp. 333–345, 1887.

[41] G. Chantry, J. Fleming, D. Fuller, H. Gebbie, and B. Steventon, "The use of a michelson interferometer for continuous gas analysis in the far infra-red," *Infrared Physics*, vol. 12, no. 1, pp. 35–44, 1972. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020089172900346

[42] J. Ma, X. Ma, and L. Xu, "Optical ultrasound sensing for biomedical imaging," *Measurement*, vol. 200, p. 111620, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224122008302

[43] B. P. Abbott *et al.*, "Gw151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence," *Physical Review Letters*, vol. 116, no. 24, Jun. 2016. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.116.241103

[44] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews*

of Modern Physics, vol. 81, no. 3, p. 1301–1350, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.81.1301

[45] H.-K. Lo, M. Curty, and K. Tamaki, "Secure quantum key distribution," *Nature Photonics*, vol. 8, no. 8, p. 595–604, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1038/nphoton.2014.149

[46] D. N. Payne, A. J. Barlow, and J. J. Ramskov Hansen, "Development of low- and high-birefringence optical fibers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 30, no. 4, pp. 323–334, 1982.

[47] W. Eickhoff, "Temperature sensing by mode–mode interference in birefringent optical fibers," *Opt. Lett.*, vol. 6, no. 4, pp. 204–206, Apr 1981. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-6-4-204

[48] W. Wootters and W. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, 1982.

[49] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Phys. Rev. Lett.*, vol. 67, pp. 661–663, Aug 1991. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.67.661

[50] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[51] S. Weinberg, *Foundations of Modern Physics*. Cambridge University Press, 2021.

[52] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, "Proposed experiment to test local hidden-variable theories," *Physical Review Letters*, vol. 23, pp. 880–884, 1969.

[53] A. Einstein, B. Podolsky, and N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?" *Phys. Rev.*, vol. 47, pp. 777–780, May 1935. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.47.777

[54] W. Tittel, J. Brendel, H. Zbinden, and N. Gisin, "Violation of bell inequalities by photons more than 10 km apart," *Physical Review Letter*, vol. 81, pp. 3563–3566, 1998.

[55] J. C. Howell, R. S. Bennink, S. J. Bentley, and R. W. Boyd, "Realization of the einstein-podolsky-rosen paradox using momentumand position-entangled photons from spontaneous parametric down conversion," *Physical Review Letter*, vol. 92, no. 21, 2004.

[56] D. Mayers, "Advances in cryptology," in *Proceedings of Crypto'96*, ser. Lecture Notes in Computer Science, N. Koblitz, Ed., vol. 1109. Springer, New York, 1996, pp. 343–357.

[57] H.-K. Lo and H. F. Chau, "Unconditional security of quantum key distribution over arbitrarily long distances," *Science*, vol. 283, no. 5410, p. 2050–2056, Mar. 1999. [Online]. Available: http://dx.doi.org/10.1126/science.283.5410.2050

[58] H.-K. Lo, X. Ma, and K. Chen, "Decoy state quantum key distribution," *Physical Review Letters*, vol. 94, 2005.

[59] D. Gottesman, H.-K. Lo, N. Lütkenhaus, and J. Preskill, "Security of quantum key distribution with imperfect devices," 2004. [Online]. Available: https://arxiv.org/abs/quant-ph/0212066

[60] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[61] W.-Y. Hwang, "Quantum key distribution with high loss: Toward global secure communication," *Physical Review Letters*, vol. 91, no. 5, Aug. 2003. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.91.057901

[62] H.-K. Lo, M. Curty, and B. Qi, "Measurement device independent quantum key distribution," *Physical Review Letters*, vol. 108, 2012.

[63] H. Inamori, N. Lütkenhaus, and D. Mayers, "Unconditional security of practical quantum key distribution," *The European Physical Journal D*, vol. 41, pp. 599–627, 03 2007.

[64] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, "Quantum repeaters: The role of imperfect local operations in quantum communication," *Phys. Rev. Lett.*, vol. 81, pp. 5932–5935, Dec 1998. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.81.5932

[65] L. Duan, M. Lukin, J. Cirac, and P. Zoller, "Long-distance quantum communication with atomic ensembles and linear optics," *Nature*, vol. 414, 2001.

[66] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin, "Quantum repeaters based on atomic ensembles and linear optics," *Rev. Mod. Phys.*, vol. 83, pp. 33–80, Mar 2011. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.83.33

[67] M. Takeoka, S. Guha, and M. M. Wilde, "Fundamental rate-loss tradeoff for optical quantum key distribution," *Nature Communications*, vol. 5, no. 1, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1038/ncomms6235

[68] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, "Fundamental limits of repeaterless quantum communications," *Nature Communications*, vol. 8, no. 1, Apr. 2017. [Online]. Available: http://dx.doi.org/10.1038/ncomms15043

[69] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and et al., "Overcoming the rate–distance limit of quantum key distribution without quantum repeaters," *Nature*, vol. 557, pp. 400–403, 2018.

[70] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," in *Advances in Cryptology — EUROCRYPT '90*, I. B. Damgård, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 253–265.

[71] R. B. G. L. Smith, *Statistical Methods in Practice: For Scientists and Technologists*. ohn Wiley & Sons, 2009.

[72] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Advances in Cryptology — EUROCRYPT '93*, T. Helleseth, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 410–423.

[73] M. Mehic, O. Maurhart, S. Rass, D. Komosny, F. Rezac, and M. Voznak, "Analysis of the public channel of quantum key distribution link," *IEEE Journal of Quantum Electronics*, vol. 53, no. 5, pp. 1–8, 2017.

[74] M. Mehic, P. Fazio, S. Rass, O. Maurhart, M. Peev, A. Poppe, J. Rozhon, M. Niemiec, and M. Voznak, "A novel approach to quality-of-service provisioning in trusted relay quantum key distribution networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 168–181, 2020.

[75] M. Mehic, M. Niemiec, H. Siljak, and M. Voznak, *Error Reconciliation in Quantum Key Distribution Protocols*. Cham: Springer International Publishing, 2020, pp. 222–236. [Online]. Available: https://doi.org/10.1007/978-3-030-47361-7_11

[76] D. Tupkary and N. Lütkenhaus, "Using cascade in quantum key distribution," *Physical Review Applied*, vol. 20, no. 6, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1103/PhysRevApplied.20.064040

[77] B. Rijsman, "The cascade information reconciliation protocol," 2020, accessed in 2024. [Online]. Available: https://cascade-python.readthedocs.io/en/latest/protocol.html

[78] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson, "Fast, efficient error reconciliation for quantum cryptography," *Physical Review A*, vol. 67, no. 5, May 2003. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.67.052303

[79] R. W. Hamming, "Error detecting and error correcting codes," *The Bell System Technical Journal*, no. 2, pp. 147–161, 1950.

[80] ——, *Coding and Information Theory*. Prentice Hall, 1980.

[81] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 2001. [Online]. Available: http://www.cacr.math.uwaterloo.ca/hac/

[82] D. Elkouss, A. Leverrier, R. Alléaume, and J. Boutros, "Efficient reconciliation protocol for discrete-variable quantum key distribution," *CoRR*, vol. abs/0901.2140, 2009. [Online]. Available: http://arxiv.org/abs/0901.2140

[83] J. Moreira and P. Farrel, *Essential of Error-Control Coding*. Wiley & Sons, 2006.

[84] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[85] I. L. Martínez, "Real world quantum cryptography," Ph.D. dissertation, University of Calgary, 2014.

[86] R. Renner, "Security of quantum key distribution," 2006. [Online]. Available: https://arxiv.org/abs/quant-ph/0512258

[87] C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM Journal on Computing*, vol. 17, no. 2, pp. 210–229, 1988. [Online]. Available: https://doi.org/10.1137/0217014

[88] J. Carter and M. N. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, vol. 18, no. 2, pp. 143–154, 1979. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022000079900448

[89] M. N. Wegman and J. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System*

Sciences, vol. 22, no. 3, pp. 265–279, 1981. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022000081900337

[90] S. Vadhan and L. Trevisan, "Extracting randomness from samplable distributions," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2000, p. 32. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SFCS.2000.892063

[91] T. Tsurumaru and M. Hayashi, "Dual universality of hash functions and its applications to quantum cryptography," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4700–4717, 2013.

[92] Y. Huang, X. Zhang, and X. Ma, "Stream privacy amplification for quantum cryptography," *PRX Quantum*, vol. 3, no. 2, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1103/PRXQuantum.3.020353

[93] M. Hayashi, "Exponential decreasing rate of leaked information in universal random privacy amplification," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3989–4001, 2011.

[94] Y. G. Yang, P. Xu, R. Yang *et al.*, "Quantum hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption," *Scientific Reports*, vol. 6, 2016. [Online]. Available: https://doi.org/10.1038/srep19788

[95] M. Hayashi and T. Tsurumaru, "More efficient privacy amplification with less random seeds via dual universal hash function," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 2213–2232, 2016.

[96] R. M. Gray, "Toeplitz and circulant matrices: A review," *Foundations and Trends® in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2006. [Online]. Available: http://dx.doi.org/10.1561/0100000006

[97] G. Bebrov, "On the (relation between) efficiency and secret key rate of qkd," *Scientific Reports*, vol. 14, p. 3638, 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-54246-y

[98] P. Jouguet and S. Kunz-Jacques, "High performance error correction for quantum key distribution using polar codes," 2013. [Online]. Available: https://arxiv.org/abs/1204.5882

[99] X. Ma, B. Qi, Y. Zhao, and H.-K. Lo, "Practical decoy state for quantum key distribution," *Physical Review A*, vol. 72, no. 1, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.72.012326

A
Error Estimation Code

```
%% Master Thesis - Error Estimation - Camila Lima
clc
clear all
close all

%% Initializing Alice's Key

key_leng = 1e6;
error_percent = 20;
sift_percent = 20;
Alice_key = randi([0,1],1,key_leng);

%% Bob's Key

Bob_key = bob_key_generator(key_leng,error_percent,Alice_key);

%% Error Estimation

num_iter = 1e3;

mean_est_error = mean(error_estimation_calculator(key_leng,
    ↪ sift_percent,num_iter,Alice_key,Bob_key));

%% Estimated Error x Sifted Percent

max_sift = 20;
min_sift = 1;
num_points = 20;
num_iter = 1e3;

sifted_percent_var = linspace(min_sift,max_sift,num_points);
```

```
est_error_var = zeros(num_iter,num_points);

for i = 1:num_points
    est_error_var(:,i) = error_estimation_calculator(key_leng,
       ↪ sifted_percent_var(i),num_iter,Alice_key,Bob_key);
end

%% Plotting Full Result (Box Plot)

figure(1)
boxplot(est_error_var,sifted_percent_var)
grid on
%plot(sifted_percent_var,ones(num_points)*error_percent,'--','
   ↪ Color','r','LineWidth',2,'DisplayName','Absolute Error')
xlabel('Sacrificed␣Keys␣[%]')
ylabel('Estimated␣Error␣[%]')
set(gca,'FontSize',26)

%% Standard Deviation

error_std_dev = zeros(1,num_points);

for i = 1:num_points
    error_std_dev(i) = std(est_error_var(:,i));
end

figure(2)
plot(sifted_percent_var,error_std_dev,'LineWidth',3)
grid on
xlabel('Sacrificed␣Keys␣[%]')
ylabel('Standard␣Deviation␣[%]')
%title('Standard Deviation For Error Estimation')
legend('Error␣=␣5%')
set(gca,'FontSize',26)

%% Stadard Deviation Multiple Errors

max_sift = 20;
min_sift = 1;
```

```
num_points = 20;
num_iter = 1e3;
error_std_dev = zeros(4,num_points);
error_mean = zeros(4,num_points);
j = 1;
figure(3)
hold on
for error_percent = [1,5,10,20]

    Bob_key = bob_key_generator(key_leng,error_percent,Alice_key
        ↪ );

    sifted_percent_var = linspace(min_sift,max_sift,num_points);

    est_error_var = zeros(num_iter,num_points);
    % error_std_dev = zeros(1,num_points);
    form = ["-diamond","-o","-^","-*"];
    leg = ["Error = 1%","Error = 5%","Error = 10%","Error =
        ↪ 20%"];

    for i = 1:num_points
        est_error_var(:,i) = error_estimation_calculator(key_leng
            ↪ ,sifted_percent_var(i),num_iter,Alice_key,Bob_key);
        error_std_dev(j,i) = std(est_error_var(:,i));
        error_mean(j,i) = mean(est_error_var(:,i));
    end
    % plot(sifted_percent_var,error_std_dev(j,:),form(j),'
        ↪ LineWidth',3)

    plot(sifted_percent_var,error_std_dev(j,:)./error_mean(j,:),
        ↪ form(j),'LineWidth',3)
    grid on
    xlabel('Sacrificed␣Keys␣[%]')
    ylabel('Coefficient␣of␣Variation')
    %title('Standard Deviation For Error Estimation')
    set(gca,'FontSize',26)
    j = j + 1;
end
legend(leg)
```

```
hold off

%% Simulation of Eve Attacking First Bits

error_percent = 10;

Bob_key = bob_key_generator(key_leng,error_percent,Alice_key);
Bob_key_2 = bob_key_generator_eve_attack(key_leng,error_percent,
    ↪ Alice_key);

max_sift = 2;
min_sift = 0.1;
num_points = 5;
num_iter = 1e3;

sifted_percent_var = linspace(min_sift,max_sift,num_points);

est_error_var_1 = zeros(num_iter,num_points);
est_error_var_2 = zeros(num_iter,num_points);
error_std_dev_1 = zeros(1,num_points);
error_std_dev_2 = zeros(1,num_points);

for i = 1:num_points
    est_error_var_1(:,i) = error_estimation_calculator(key_leng,
        ↪ sifted_percent_var(i),num_iter,Alice_key,Bob_key);
    est_error_var_2(:,i) = error_estimation_calculator(key_leng,
        ↪ sifted_percent_var(i),num_iter,Alice_key,Bob_key_2);
    error_std_dev_1(i) = std(est_error_var_1(:,i));
    error_std_dev_2(i) = std(est_error_var_2(:,i));
end

%% Plot Results For Eve's Attack

figure(4)
subplot(2,1,1);
boxplot(est_error_var_1,sifted_percent_var);
grid on
title("Error = 10%, Random Error")
xlabel('Sacrificed␣Keys␣[%]')
```

```
ylabel('Estimated␣Error␣[%]')
set(gca,'FontSize',16)

subplot(2,1,2);
boxplot(est_error_var_2,sifted_percent_var);
grid on
title("Error = 10%, First Bits Error")
xlabel('Sacrificed␣Keys␣[%]')
ylabel('Estimated␣Error␣[%]')
set(gca,'FontSize',16)

figure(5)
plot(sifted_percent_var,error_std_dev_1,sifted_percent_var,
    ↪ error_std_dev_2,'LineWidth',3);
grid on
legend("Error = 10%, Random Error","Error = 10%, First Bits
    ↪ Error")
xlabel('Sacrificed␣Keys␣[%]')
ylabel('Standard␣Deviation␣[%]')
set(gca,'FontSize',26)

%% Bob's Key Generator

function Bob_key = bob_key_generator(key_leng,error_percent,
    ↪ Alice_key)

error_index = randperm(key_leng,round(error_percent/100*key_leng
    ↪ ));

Bob_key = Alice_key;

for i = error_index
    Bob_key(i) = not(Bob_key(i));
end
end

%% Bob Key Generation (Given Eve is Attacking First Bits)
```

```
function Bob_key = bob_key_generator_eve_attack(key_leng,
    ↪ error_percent,Alice_key)

Bob_key = Alice_key;

for i = 1:key_leng*error_percent/100
    Bob_key(i) = not(Bob_key(i));
end
end

%% Error Estimation Calculator

function est_error = error_estimation_calculator(key_leng,
    ↪ sift_percent,num_iter,Alice_key,Bob_key)

est_error = zeros(1,num_iter);
for m = 1:num_iter

    error_count = 0;

    sifted_index = randperm(key_leng,round(sift_percent/100*
        ↪ key_leng));

    for i = sifted_index
        error_count = error_count + (Bob_key(i) ~= Alice_key(i));
    end

    est_error(m) = error_count/(key_leng*sift_percent/100)*100;
end

% mean_est_error = mean(est_error);

end
```

B
Error Reconciliation Code

```
%% Master Thesis - Error Reconciliation - Camila Lima
clc
clear all
close all

%% Initializing Alice's Key

key_len = 1e6;
error_percent = 5;
Alice_key = randi([0,1],1,key_len);

%% Bob's Key

Bob_key = bob_key_generator(key_len,error_percent,Alice_key);

%% Cascade Error Correction

Bob_corrected_key = Cascade(Bob_key,Alice_key,error_percent/100,
    ↪ key_len);

error_after = sum(xor(Bob_corrected_key,Alice_key))/key_len;

fprintf("Error after: %0.4f percent \n",error_after*100);

%% Cascade Error Correction For Multiple Errors

num_runs = 100;
QBER = [0.1,0.5,1,2,3,4,5,6,7,8,9,10];
QBER_after = zeros(num_runs,length(QBER));
i = 1;
for q = QBER
    for j = 1:num_runs
```

```
        Bob_key = bob_key_generator(key_len,q,Alice_key);

        Bob_corrected_key = Cascade(Bob_key,Alice_key,q/100,
            ↪ key_len);

        QBER_after(j,i) = sum(xor(Bob_corrected_key,Alice_key))/
            ↪ key_len*100;
    end
    i = i + 1;
end

figure(1)
boxplot(QBER_after,QBER)
grid on
xlabel('Initial␣QBER␣[%]')
ylabel('QBER␣After␣Reconciliation␣[%]')
set(gca,'FontSize',26)

%% Multiple Iteractions Cascade

n_sce = 5;
sce = [[1 2 3 4 0
    ↪ 0];[1,2,3,4,1,0];[1,2,3,4,2,0];[1,2,3,4,3,2];[1,2,3,4,3,4]];
    ↪
num_runs = 100;
QBER = 5;
QBER_after = zeros(num_runs,n_sce);

for i = 1:n_sce
    fprintf("Scenario %d start\n",i);
    for j = 1:num_runs
        Bob_key = bob_key_generator(key_len,QBER,Alice_key);

        Bob_corrected_key = Cascade_multi(Bob_key,Alice_key,QBER
            ↪ /100,key_len,sce(i,:));

        QBER_after(j,i) = sum(xor(Bob_corrected_key,Alice_key))/
            ↪ key_len*100;
    end
```

```
    fprintf("Scenario %d done\n",i);
end

figure(2)
boxplot(QBER_after)
grid on
xlabel('Scenarios')
ylabel('QBER After Reconciliation [%]')
set(gca,'FontSize',26)

%% FUNCTIONS %%
%% Bob's Key Generator

function Bob_key = bob_key_generator(key_leng,error_percent,
    ↪ Alice_key)

error_index = randperm(key_leng,round(error_percent/100*key_leng
    ↪ ));

Bob_key = Alice_key;

for i = error_index
    Bob_key(i) = not(Bob_key(i));
end
end

%% The Binary Algorithm

function Corrected_block = Binary_algorithm(parent_block,
    ↪ correct_parent_block,block_size)

left_block = parent_block(1:ceil(block_size/2));

right_block = parent_block(ceil(block_size/2)+1:end);

correct_left_block_parity = parity_check(correct_parent_block(1:
    ↪ ceil(block_size/2)));
```

```
is_left_odd = error_parity(correct_left_block_parity,
   ↪ parity_check(left_block));

switch is_left_odd

    case 1
        odd_parity_error_block = left_block;

        if isscalar(odd_parity_error_block)
            Corrected_block = [not(odd_parity_error_block)
               ↪ right_block];
            return
        else
            Corrected_block = [Binary_algorithm(left_block,
               ↪ correct_parent_block(1:ceil(block_size/2)),ceil
               ↪ (block_size/2)) right_block];
        end

    case 0
        odd_parity_error_block = right_block;
        if isscalar(odd_parity_error_block)
            Corrected_block = [left_block not(
               ↪ odd_parity_error_block)];
            return

        else
            Corrected_block = [left_block Binary_algorithm(
               ↪ right_block,correct_parent_block(ceil(
               ↪ block_size/2)+1:end),floor(block_size/2))];
        end
end

end

%% Parity Check Function

function parity = parity_check(binary_string)

parity = 0;
```

```
for b = binary_string
    parity = xor(parity,b);
end

end

%% Error Parity Function

function par = error_parity(b1,b2)

par_vec = xor(b1,b2);

par = parity_check(par_vec);

end

%% Cascade Reconciliation Protocol (Single Iteration)

function corrected_key = Cascade_single_iter(Bob_key,Alice_key,
    ↪ tl_block_size)

corrected_key = Bob_key;

num_blocks = floor(length(Alice_key)/tl_block_size);

for n = 1:num_blocks

    ib = 1 + (n-1)*tl_block_size;
    eb = ib - 1 + tl_block_size;

    if error_parity(corrected_key(ib:eb),Alice_key(ib:eb))

        corrected_key(ib:eb) = Binary_algorithm(corrected_key(ib:
            ↪ eb),Alice_key(ib:eb),tl_block_size);

    end
end
```

```
if error_parity(corrected_key(num_blocks*tl_block_size:end),
    ↪ Alice_key(num_blocks*tl_block_size:end))

    corrected_key(num_blocks*tl_block_size:end) =
        ↪ Binary_algorithm(corrected_key(num_blocks*
        ↪ tl_block_size:end), ...
        Alice_key(num_blocks*tl_block_size:end), ...
        length(Alice_key)-num_blocks*tl_block_size);
end

end

%% Cascade Reconciliation Protocol (Single Run)

function corrected_key = Cascade(Bob_key,Alice_key,QBER,key_len)

k1 = ceil(0.73/QBER);

corrected_key = Cascade_single_iter(Bob_key,Alice_key,k1);

k2 = 2*k1;

per = randperm(key_len);

corrected_key = Cascade_single_iter(shuffle(corrected_key,per),
    ↪ shuffle(Alice_key,per),k2);

corrected_key = unshuffle(corrected_key,per);

k3 = 2*k2;

per = randperm(key_len);

corrected_key = Cascade_single_iter(shuffle(corrected_key,per),
    ↪ shuffle(Alice_key,per),k3);

corrected_key = unshuffle(corrected_key,per);

k4 = 2*k3;
```

```
per = randperm(key_len);

corrected_key = Cascade_single_iter(shuffle(corrected_key,per),
    ↪ shuffle(Alice_key,per),k4);

corrected_key = unshuffle(corrected_key,per);

end


%% Cascade Reconciliation Protocol (Multiple Runs)

function corrected_key = Cascade_multi(Bob_key,Alice_key,QBER,
    ↪ key_len,order)

k = zeros(1,4);

k(1) = ceil(0.73/QBER);
k(2) = 2*k(1);
k(3) = 2*k(2);
k(4) = 2*k(3);

corrected_key = Bob_key;

i = 0;
for o = order
    if o == 0
        return
    elseif i == 0
        corrected_key = Cascade_single_iter(corrected_key,
            ↪ Alice_key,k(1));
        i = 1;
    else
        per = randperm(key_len);

        corrected_key = Cascade_single_iter(shuffle(corrected_key
            ↪ ,per),shuffle(Alice_key,per),k(o));
```

```
        corrected_key = unshuffle(corrected_key,per);
    end

end
end

%% Shuffle and Unshufle Function

function shuffled_key = shuffle(key,new_pos)

shuffled_key = key;

for i = 1:length(key)
    shuffled_key(i) = key(new_pos(i));
end

end

function unshuffled_key = unshuffle(shuffled_key,new_pos)

unshuffled_key = shuffled_key;

for i = 1:length(shuffled_key)
    unshuffled_key(new_pos(i)) = shuffled_key(i);
end

end
```

C
Post Processing Code

```
%% Master Thesis - Post Processing - Camila Lima
clc
clear all
close all

%% Pre-determined

key_init_len = 1e6;

QBER_sim = [1,2,3,4,5,6,7,8];

sacrifice_percent = 10;

cascade_order = [1,2,3,4,3,4];
i=1;

SIM_events = 50;

secret_key_len = zeros(SIM_events,length(QBER_sim));

for QBER_channel = QBER_sim
    for sim = 1:SIM_events

        %% Alice Preparation

        Alice_bases = randi([0,1],1,key_init_len); %% 0:
            ↪ horizontal/vertical ; 1: Diagonal/Anti-diagonal

        Alice_init_key = randi([0,1],1,key_init_len);

        Alice_QBIT = [Alice_bases ; Alice_init_key];
```

```
%% Bob Key Extraction

Bob_bases = randi([0,1],1,key_init_len);

Bob_init_key = key_extraction(Bob_bases,Alice_QBIT,
    ↪ QBER_channel,key_init_len);

%% Key Sifting

[sift_index,sift_key_len] = basis_reconciliation(
    ↪ Alice_bases,Bob_bases);

Alice_sift_key = Alice_init_key(sift_index);

Bob_sift_key = Bob_init_key(sift_index);

%% Error Estimation

[est_QBER,Alice_sac_key,Bob_sac_key,sac_key_len] =
    ↪ error_estimation(sift_key_len,sacrifice_percent,
    ↪ Alice_sift_key,Bob_sift_key);

%% Error Correction: Cascade Protocol

Bob_corrected_key = Cascade_multi(Bob_sac_key,
    ↪ Alice_sac_key,est_QBER/100,sac_key_len,
    ↪ cascade_order);

%% Privacy Amplification

f_EC = 1.24;

secret_key_len(sim,i) = ceil(sac_key_len*(1-(1+f_EC)*
    ↪ sh_entropy(est_QBER/100)));

% seed_size = max([sac_key_len - secret_key_len,
    ↪ secret_key_len]);

% seed = randi([0 1],1,seed_size);
```

```
        % T = toeplitz(seed(1:secret_key_len),seed(1:sac_key_len
            ↪  - secret_key_len));

        % Alice_secret_key = xor(Alice_sac_key(1:secret_key_len)
            ↪ ', T*Alice_sac_key(secret_key_len+1:end)')';

        % Bob_secret_key = xor(Bob_corrected_key(1:
            ↪ secret_key_len)', T*Bob_corrected_key(
            ↪ secret_key_len+1:end)')';

    end
    i = i + 1;
    fprintf("End of QBER = %f\n",QBER_channel);
end

%% Print Result

figure(1)
hold on
boxplot(secret_key_len/key_init_len,QBER_sim)
grid on

plot(QBER_sim, 1/2*(1-sacrifice_percent/100)*((1-(1+f_EC)*
    ↪ sh_entropy(QBER_sim/100))), "--",LineWidth=2)
grid on
hold off
ylim([0 0.45])
xlim([0.5 8.5])
xlabel("QBER [%]")
ylabel("Secret Key Rate")
set(gca,'FontSize',26)
legend('Estimated␣Curve')


%% FUNCTIONS %%

%% Key Extraction
```

```
function extracted_key = key_extraction(measuring_base,
    ↪ QBIT_array,QBER,key_len) %% QBIT_array is a matrix 2x(
    ↪ key_len) which the first row is the base and the colum
    ↪ the key bit

extracted_key = zeros(1,key_len);

wrong_index = randperm(key_len,ceil(key_len*QBER/100));

for i = 1:key_len
    if measuring_base(i) == QBIT_array(1,i)
        extracted_key(i) = QBIT_array(2,i);
    else
        extracted_key(i) = randi([0,1]);
    end
end

for i = wrong_index
    extracted_key(i) = ~ extracted_key(i);
end

end

%% Basis Reconciliation

function [sift_index,sift_key_len] = basis_reconciliation(
    ↪ Alice_bases,Bob_bases)

sift_index = find(~xor(Alice_bases,Bob_bases));

sift_key_len = length(sift_index);

end

%% Error Estimation

function [est_QBER,Alice_sac_key,Bob_sac_key,sac_key_len] =
    ↪ error_estimation(key_len,sacrifice_percent,Alice_key,
    ↪ Bob_key)
```

```
est_QBER = 0;

sifted_index = randperm(key_len,ceil(sacrifice_percent/100*
    ↪ key_len));

for i = sifted_index
    est_QBER = est_QBER + (Bob_key(i) ~= Alice_key(i));
end

est_QBER = est_QBER/(key_len*sacrifice_percent/100)*100;

Alice_key(sifted_index) = [];

Alice_sac_key = Alice_key;

Bob_key(sifted_index) = [];

Bob_sac_key = Bob_key;

sac_key_len = length(Alice_sac_key);

end

%% The Binary Algorithm

function Corrected_block = Binary_algorithm(parent_block,
    ↪ correct_parent_block,block_size)

left_block = parent_block(1:ceil(block_size/2));

right_block = parent_block(ceil(block_size/2)+1:end);

correct_left_block_parity = parity_check(correct_parent_block(1:
    ↪ ceil(block_size/2)));

is_left_odd = error_parity(correct_left_block_parity,
    ↪ parity_check(left_block));
```

```
switch is_left_odd

    case 1
        odd_parity_error_block = left_block;

        if isscalar(odd_parity_error_block)
            Corrected_block = [not(odd_parity_error_block)
                ↪ right_block];
            return
        else
            Corrected_block = [Binary_algorithm(left_block,
                ↪ correct_parent_block(1:ceil(block_size/2)),ceil
                ↪ (block_size/2)) right_block];
        end

    case 0
        odd_parity_error_block = right_block;
        if isscalar(odd_parity_error_block)
            Corrected_block = [left_block not(
                ↪ odd_parity_error_block)];
            return

        else
            Corrected_block = [left_block Binary_algorithm(
                ↪ right_block,correct_parent_block(ceil(
                ↪ block_size/2)+1:end),floor(block_size/2))];
        end
end

end

%% Parity Check Function

function parity = parity_check(binary_string)

parity = 0;

for b = binary_string
    parity = xor(parity,b);
```

```
end

end

%% Error Parity Function

function par = error_parity(b1,b2)

par_vec = xor(b1,b2);

par = parity_check(par_vec);

end

%% Cascade Reconciliation Protocol (Single Iteration)

function corrected_key = Cascade_single_iter(Bob_key,Alice_key,
    ↪ tl_block_size)

corrected_key = Bob_key;

num_blocks = floor(length(Alice_key)/tl_block_size);

for n = 1:num_blocks

    ib = 1 + (n-1)*tl_block_size;
    eb = ib - 1 + tl_block_size;

    if error_parity(corrected_key(ib:eb),Alice_key(ib:eb))

        corrected_key(ib:eb) = Binary_algorithm(corrected_key(ib:
            ↪ eb),Alice_key(ib:eb),tl_block_size);

    end
end

if error_parity(corrected_key(num_blocks*tl_block_size:end),
    ↪ Alice_key(num_blocks*tl_block_size:end))
```

```
    corrected_key(num_blocks*tl_block_size:end) =
        ↪ Binary_algorithm(corrected_key(num_blocks*
        ↪ tl_block_size:end), ...
        Alice_key(num_blocks*tl_block_size:end), ...
        length(Alice_key)-num_blocks*tl_block_size);
end

end


%% Cascade Multi Passes

function corrected_key = Cascade_multi(Bob_key,Alice_key,QBER,
    ↪ key_len,order)

k = zeros(1,4);

k(1) = ceil(0.73/QBER);
k(2) = 2*k(1);
k(3) = 2*k(2);
k(4) = 2*k(3);

corrected_key = Bob_key;

i = 0;
for o = order
    if o == 0
        return
    elseif i == 0
        corrected_key = Cascade_single_iter(corrected_key,
            ↪ Alice_key,k(1));
        i = 1;
    else
        per = randperm(key_len);

        corrected_key = Cascade_single_iter(shuffle(corrected_key
            ↪ ,per),shuffle(Alice_key,per),k(o));

        corrected_key = unshuffle(corrected_key,per);
```

```
    end

end
end

%% Shuffle and Unshufle Function

function shuffled_key = shuffle(key,new_pos)

shuffled_key = key;

for i = 1:length(key)
    shuffled_key(i) = key(new_pos(i));
end

end

function unshuffled_key = unshuffle(shuffled_key,new_pos)

unshuffled_key = shuffled_key;

for i = 1:length(shuffled_key)
    unshuffled_key(new_pos(i)) = shuffled_key(i);
end

end

%% Shannon Entroy

function h = sh_entropy(x)

h = -x.*log2(x)-(1-x).*log2(1-x);

end
```

D
Post Processing in RRQ Code

```
%% Master Thesis - RRQ Simulation - Camila Lima
clc
clear all
close all

%% Initialization

L = linspace(1,1e6,61e6); % Distance [m]

F_rep = 50e6; % Pulse Repetition rate [Hz]
mu = 1; % Avarage number of photons per pulse
eta_det = 0.1; % Detectors quantum efficiency
alpha = 0.2; % Fiber loss [dB/km]
tatb = 0.01; % Alice and Bob total insertion loss
t_link = 10.^(-2*alpha/10.*L*1e-3)*(tatb); % Probability of a
    ↪ photon reaching the detectors
D = 3e-6; % Dark count per pulse per detectors
n = 2; % Number of detectors

%% Rates

R_sift = 1/2 * F_rep * mu * t_link * eta_det; % Sifted key rate

R_det = 1/4*F_rep*D*n* eta_det; % Error Detection rate

Q = R_det./R_sift;
Q((Q>1))=1;

eta_sac = 0.1; % Sacrificed percentual of the key for error
    ↪ estimation
eta_ec = 1.16; % Efficiency of error correction protocol
```

```
K = (1-eta_sac).*(1-(1+eta_ec).*sh_entropy(Q)).*R_sift;
K(find(K<0,1):end) = 0;

%% Result

figure(1)
loglog(L,K,'LineWidth',3);
xlabel("Total Distance Between Alice and Bob [m]");
ylabel("Secret Key Rate [Bit/s]");
set(gca,'FontSize',26);
grid on

fprintf("Expected Secure key per second for Rede Rio: %.0f\n",K(
    ↪ find(L>=24e3,1)));

%% Shannon Entroy

function h = sh_entropy(x)

h = -x.*log2(x)-(1-x).*log2(1-x);

end
```