Simulating the BB84 QKD protocol: A QuTiP-based study in ideal quantum channels

Henrique Sobrinho Ghizoni[1*], Anderson. R. C. Buarque [1]
[1] *QuIIN - Quantum Industrial Innovation, EMBRAPII CIMATEC Competence Center in Quantum Technologies, SENAI CIMATEC, Av. Orlando Gomes 1845, 41650-010, Salvador, BA, Brazil.*
**Corresponding author: institution; addresses; hsghizoni@gmail.com*

Abstract: This work addresses the implementation of the BB84 quantum key distribution protocol via QuTiP, a specialized Python library for quantum system modeling. The objective lies in providing a didactic yet rigorous account of the protocol, simulated over an ideal noise-free channel. The methodology comprises a detailed, stepwise simulation covering state preparation, basis selection, key sifting, error correction, and eavesdropping detection. Results indicate that the simulation effectively reproduces the protocol's expected behavior, enabling quantitative detection of intrusions through measures such as quantum bit error rate (QBER) and mutual information. This work enhances the understanding of quantum cryptographic principles by transforming abstract theoretical constructs into observable phenomena. It demonstrates that QuTiP-based modeling constitutes a powerful pedagogical tool for teaching quantum key distribution in academic settings. The findings contribute to both methodology and education in quantum cryptography by offering a reproducible simulation framework and reinforcing the physical foundations of BB84 security.

Keywords: Quantum Key Distribution, BB84, Quantum Technologies, Teaching, Quantum Communication.

Abbreviations: QKD, Quantum Key Distribution. P&M, Prepare and Measure. HWP, Half Wave Plate. PBS, Polarized Beam Splitter. RNG, Random Number Generation. SP, Single Photon. PP, Post-Processing. RKL, Raw Key Length. kB, Kilobytes. B, Bytes. ms, milliseconds. s, seconds.

1. Introduction

Classical cryptography relies on the computational difficulty of efficiently solving problems such as integer factorization or discrete logarithms and underpins widely used protocols such as RSA and Diffie–Hellman. The advancement of quantum computing and the existence of algorithms such as Shor's threaten this traditional framework, making it urgent to seek solutions based on physical principles [2].

In this context, quantum cryptography emerges as a secure alternative. The BB84 protocol, conceived by Bennett and Brassard in 1984 [1], introduces a paradigm that exploits non-orthogonal quantum states distributed in conjugate bases to share secret keys. The protocol relies on two fundamental principles of quantum mechanics: the no-cloning theorem, which prevents copying unknown quantum states, and the unavoidable disturbance caused by any measurement, which introduces detectable errors. This approach grants BB84 physical security and enables the detection of intrusions that cannot be concealed, an advantage that classical cryptography does not offer.

This work presents simulations performed with QuTiP, a Python library dedicated to modeling quantum systems, which enables the step-by-step implementation of the BB84 protocol. The implementation was carried out over an ideal, noise-free quantum channel, in order to clearly observe each stage of the protocol, rom preparation and measurement to

post-processing. This controlled setting not only facilitates a detailed analysis of BB84's operation but also makes it a valuable tool for educational purposes, supporting the teaching and understanding of fundamental concepts in quantum cryptography. Finally, the work investigates how eavesdropping on the quantum channel can be detected through the estimation of the QBER parameter and/or the analysis of the mutual information between the parties involved in the protocol.

2.1. Quantum Key Distribution-BB84

The BB84 protocol, proposed by Charles Bennett and Gilles Brassard in 1984 [1] is the first and most well-known QKD protocol. It enables two parties, traditionally named Alice (sender) and Bob (receiver), to establish a shared secret key with security guaranteed by the laws of quantum mechanics. The protocol leverages fundamental quantum principles such as the no-cloning theorem and the fact that quantum measurements disturb the system, thereby making any eavesdropping attempt detectable [2].

To implement these principles in practice, BB84 relies on two types of communication channels: a quantum channel, inherently vulnerable to eavesdropping, used for transmitting quantum states, and a classical public channel that, despite being unsecured, requires authentication to prevent attacks.

While the quantum channel ensures the uniqueness and non-replicability of the transmitted qubits, the classical channel is used for basis reconciliation and error correction. By monitoring the quantum bit error rate, the protocol allows the detection of potential eavesdropping attempts, thereby ensuring the integrity and confidentiality of the final secret key.

2.1.1. BB84: protocol essential parts

The protocol operationalizes three essential needs: a quantum channel for transmitting qubits encoded in photon polarization states; an authenticated classical channel for public communication; trust in the validity of quantum mechanical postulates, especially the no-cloning theorem, superposition, and measurement-induced collapse.

In practical implementations, BB84 uses the polarization states of single photons to encode information [2]. The protocol employs two mutually non-orthogonal bases: the rectilinear basis, $\boldsymbol{B_z}$, it contains the horizontal (0º), $|\boldsymbol{H}\rangle$, and vertical polarization (90º), $|\boldsymbol{V}\rangle$, i.e:

$$\boldsymbol{B_z} = \{|\boldsymbol{H}\rangle, |\boldsymbol{V}\rangle\} \quad (1)$$

and the diagonal basis, $\boldsymbol{B_x}$, it contains the diagonal (45º), $|\boldsymbol{D}\rangle$, and the anti-diagonal (-45º) polarization states:

$$\boldsymbol{B_x} = \{|\boldsymbol{A}\rangle, |\boldsymbol{D}\rangle\} \quad (2)$$

where

$$|\boldsymbol{A}\rangle = \frac{|\boldsymbol{H}\rangle+|\boldsymbol{V}\rangle}{\sqrt{2}}, \quad (3)$$

$$|\boldsymbol{D}\rangle = -\frac{|\boldsymbol{H}\rangle+|\boldsymbol{V}\rangle}{\sqrt{2}} \quad (4)$$

Shannon entropy, introduced by Claude E. Shannon [3], quantifies the uncertainty associated with random variable and is given by

$$\boldsymbol{H(X) = -\Sigma_i\, p(x_i) log_2[p(x_i)]}, \qquad (5)$$

Where $\boldsymbol{p(x_i)}$ represents the probability of occurrence of event $\boldsymbol{x_i}$. In the context of quantum key distribution, entropy measures the unpredictability of the generated keys and serves as a key parameter for assessing security and the rate of useful information transmitted.

Closely related to this concept is mutual information, defined between two random variables $\boldsymbol{X}$ e $\boldsymbol{Y}$ as

$$\boldsymbol{I(X{:}Y) = H(X) - X(X|Y)}. \qquad (6)$$

Mutual information represents the amount of knowledge one variable contains about the other, and in quantum key distribution, it is used to compare the information shared between legitimate parties (Alice and Bob) and that potentially accessible to an eavesdropper (Eve).

2.1.2. Prepare and measure

The P&M operation of the BB84 protocol can be divided into a few main steps:

P&M-I. Alice generates two uniformly random bit strings:

$$a = (a_1, \dots, a_{4n}), \qquad (7)$$

$$b = (b_1, \dots, b_{4n}), \qquad (8)$$

where $a_i \in \{0,1\}$ denotes the bit value and $b_i \in \{0,1\}$ determines the basis $(0{:}\,Bz,\ 1{:}\,Bx)$. The use of an oversized block (e.g., 4n) anticipates the need for sifting and error estimation.

For each pair (a_i, b_i) Alice prepares a qubit:

$$|\psi_i\rangle = \begin{cases} |0\rangle, & a_i = 0,\ b_i = 0 \\ |1\rangle, & a_i = 1,\ b_i = 0 \\ |A\rangle, & a_i = 0,\ b_i = 1 \\ |D\rangle, & a_i = 1,\ b_i = 1 \end{cases} \qquad (9)$$

She then sends the sequence of 4n qubits to Bob via the quantum channel. The no-cloning theorem ensures that any attempt by Eve to duplicate these states will fail [2-3].

P&M-II. Bob independently chooses a random basis string, given by

$$b' = (b'_1, \dots, b'_{4n}) \qquad (10)$$

and measures each incoming qubit in the corresponding basis. He records the outcomes a_i. If $b_i = b'_i$, then Bob retrieves a_i with certainty (ignoring noise); otherwise, the outcome is statistically uncorrelated with a_i.

2.1.3. Post-processing

The post-processing operation of the BB84 protocol can be divided into a few main steps:

PP-I. Alice divulges her basis string b_i over the authenticated classical channel. Bob compares with his b_i' and retains only those indices where $b_i = b'_i$. The remaining bits, roughly 2n in number, form the sifted key. As shown by [2], the expected matching probability is 0.5 due to the uniform random choice of bases.

PP-II. Alice and Bob publicly sample a subset of the sifted key to estimate the Quantum Bit Error Rate (QBER):

$$QBER = {}^{N_b}/_{L_{Sk}} \qquad (10)$$

where N_b is the number of bitflips and L_{Sk} is the length of the sifted key, obtained after the sifting process. An intercept-resend attack introduces a theoretical QBER of 25%, since Eve's random basis choice leads to a 50% error rate half the time [2-3].

PP-III. Provided the QBER is below a critical threshold (typically~ 11%) for individual attacks) [2], Alice and Bob engage in an error correction protocol (e.g., Cascade or LDPC codes) to reconcile discrepancies in their sifted keys.

PP-IV. To remove any partial knowledge Eve may have acquired, Alice and Bob apply a privacy amplification procedure, compressing the reconciled key into a shorter string of length L_{Rk}. The key rate is approximately:

$$r = 1 - H(QBER - I(A:E)), \qquad (11)$$

where $H(p)$ is the binary entropy function and $I(A:E)$ is Eve's estimated mutual information. This ensures a secure final key under general attacks.

2.1.4 Eve's Attack

There are different types of attacks [2] that Eve can perform during the execution of the BB84 protocol, such as: Intercept-Resend (IR), Photon Number Splitting (PNS), Blinding Attack, etc. In the intercept-resend attack, Eve intercepts the quantum channel between Alice and Bob and follows the preparation and measurement steps, attempting to impersonate Bob to Alice, and Alice to Bob, and randomly selects the basis to be used in the measurement in each round. In this attack, Eve aims to extract as much information as possible from the keys generated during the protocol stages. However, as Eve intercepts the quantum channel by performing measurement and subsequently preparing a state equivalent to the one obtained, due to the non-orthogonality between the states of different bases, Eve begins to introduce errors in the key bits, which can be estimated during the parameter estimation stage. Intercept-Resend is one of the first attacks to be discussed in the context of quantum key distribution teaching, and therefore this work will focus on it.

There are more sophisticated attacks, such as Photon Splitting Number (PNS) [2], in which Eve takes advantage of possible variations in the number of photons generated by the source per round and stores these photons in a quantum memory, performing measurements after the splitting stage. Another famous attack in the literature involves exploiting loopholes in APD detectors, the Blinding Attack [2], in which Eve manages to manipulate the triggering of Bob's detector.

2.1.5. Implementation

The BB84 protocol was designed to be implemented in photonic systems, ideally in the single-photon (SP) regime. Figure 1 shows an implementation of the BB84 protocol, such that the choice of quantum channel can be arbitrary, free space or optical fiber, and the classical authenticated channel (CAC) also has an arbitrary choice, and can even be a public channel, as long

as it ensures that access is only granted to the respective user (Alice/Bob).

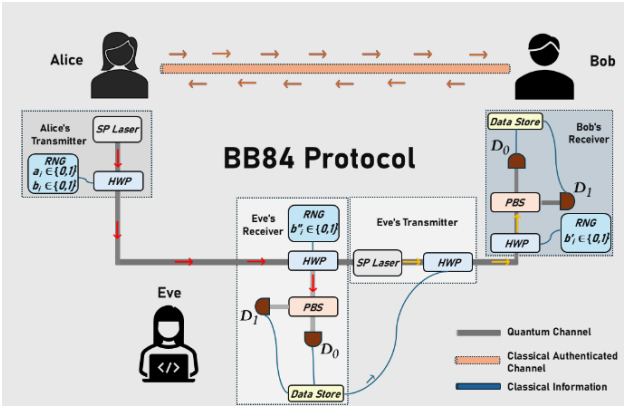
In the context of BB84, the presence of a spy, called Eve, must always be considered. In this work, we will consider the Intercept-Resend type of attack, in which Eve has access to the information disclosed in the classical channel, without being able to alter it, and, in addition, Eve intercepts the quantum channel, performing a measurement on the state sent by Alice, $|\psi_A\rangle$, and resent an equivalent state to the measured state to Bob, $|\psi_E\rangle$. It is worth noting that, due to the Non-Cloning Theorem [2], Eve cannot clone Alice's state.

As shown in Figure 1, Alice prepares the state using three main components: a random number generator (RNG), used to generate the bit to be sent a_i, and the base chosen to encrypt the bit, b_i; a single-photon laser; and a Half-Wave Plate (HWP), responsible for generating polarization in the photon, which can be 0°, 90°, -45°, or 45°. Alice prepares the state $|\psi_A\rangle$ using these tools and sends it to Bob through the quantum channel.

To perform the IR attack, Eva needs to take a measurement in the state $|\psi_A\rangle$, and she can perform this measurement using: a random number generator (RNG), to generate b_i''; a PBS, which associates the polarization of the photon with a direction of propagation; and two detectors, D_0 and D_1, associated with measurements 0 and 1 (computational basis), respectively; a data center, to store the results obtained. For Eve to try to go unnoticed and take advantage of the espionage, she will always send Bob a state $|\psi_E\rangle$, which is constructed by measuring Alice's state. The state is prepared and sent to Bob via the quantum channel.

Bob's measurement can be performed using: an RNG, to generate b'_i, the basis to be used for the measurement; a PBS, which associates the polarization of the photon with a direction of propagation; and two detectors, D_0 and D_1, associated with measurements 0 and 1 (computational basis) and a data center, to store the result obtained.

Figure 1: (Color online) Representation of one of the ways to implement the BB84 protocol. Ideally, Alice prepares a state (a_i, b_i), with a_i and b_i chosen randomly, encrypting the state produced by the laser through the polarization of light in the Half-Wave Plate (HWP) and sends the state to Bob through the quantum channel. Bob then randomly selects a basis b'_i and configures the HWP according to the selected basis before sending it to Alice. The Polarizing Beam Splitter (PBS) introduces a path difference for the light according to the polarization, which is then associated with detectors D_0 and D_1. Considering the Intercept-Resend attack, Eve has the same receiver apparatus as Bob and the same transmitter apparatus as Alice.



3. Simulation Methodology

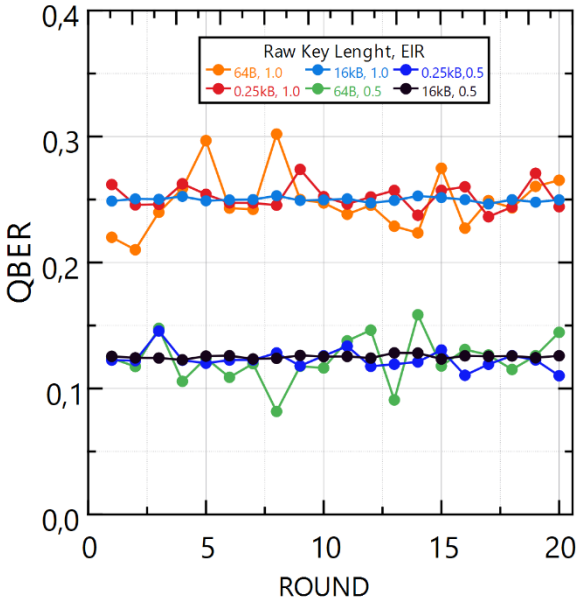
The main objective of the BB84 protocol is to generate a secure key between Alice and Bob, such that if Eve attempts to intercept the quantum channel, she can be detected in the PP part, specifically in steps PP-II and PP-III, by estimating the QBER and classifying whether to maintain or abort the protocol. In order to simulate a quantum key distribution using the implementation proposed in Figure 1, we chose to use QuTip[4], a Python package for quantum physics and its applications, for the simulation.

There are two main investigations in this work, one in which the Eve interception rate (EIR) is fixed, 0.5 or 1.0, for a certain number of protocol rounds, in which the main objective is to investigate the QBER rate and its relationship with the raw key size; another investigation in which Eve's interception rate is analyzed in relation to mutual information between Alice and Bob, $\boldsymbol{I(A:B)}$, the mutual information between Alice and Eve, $\boldsymbol{I(A:E)}$, and the difference between these quantities, given by $\boldsymbol{I(A:B) - I(A:E)}$.

4. Results

The QBER rate is of great importance for the BB84 protocol, since it can be used to detect Eve's presence during protocol execution. In this context, Figure 2 contains the QBER results for 20 distinct rounds of BB84 protocol execution, for three RKL values: 64B, 0.25kB, and 16kB; and two values for EIR: 0.5 e 1.0, corresponding to 50% and 100% Eve interception during protocol executions, respectively. The main result obtained is the impact that the RKL value has on the QBER statistics, obtaining a direct relationship that the higher the RKL value, the lower the QBER variation, leading to the idea that the larger the key size, the more accurately the QBER value can be obtained, which is of great importance for real implementations.

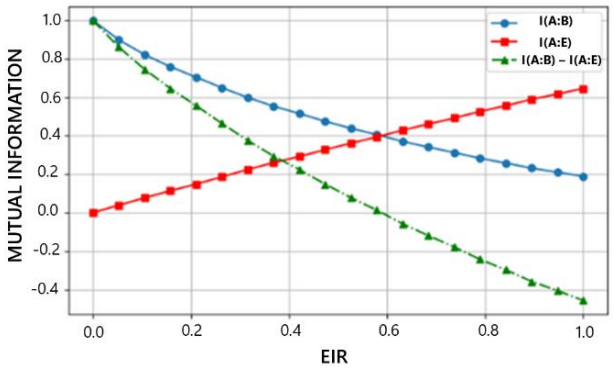
Figure 2: (Color online) The QBER rate vs round in the BB84 protocol simulation using QuTiP. The orange, red, and light blue curves given EIR=1.0 (Eve's Intercept Rate), for three key sizes, 64B, 0.25kB, and 16kB, respectively. The green, dark blue and black curves given EIR=0.5, for three key sizes, 64B, 0.25kB, and 16kB, respectively. It can be seen that the larger the key size, the smaller the variation in the QBER value.



The mutual information between Alice and Bob, $\boldsymbol{I(A:B)}$ is a quantity that directly correlates the amount of information shared between both parties and is therefore a quantifier for QKD. In this same scenario, the mutual information between Alice and Eve, $\boldsymbol{I(A:E)}$ is associated with the amount of information Eve can obtain through the attack, that is, the higher the value of $\boldsymbol{I(A:E)}$, the more successful

Eve is in her attack. Figure 3 contains an analysis between the EIR and the three maximum values of the different mutual information: $\boldsymbol{I(A:B)}$, $\boldsymbol{I(A:E)}$ e $\boldsymbol{I(A:B) - I(A:E)}$, for a key with RKL equal to 64kB. The results obtained point to a direct correlation between EIR and mutual information values. As EIR increases, we have two results: $\boldsymbol{I(A:B)}$ decays exponentially, as does $\boldsymbol{I(A:B) - I(A:E)}$, and, on the other hand, $\boldsymbol{I(A:E)}$ increases linearly. An important thing to note is when $\boldsymbol{EIR \approx 0.58}$, leading to the case that $\boldsymbol{I(A:E) \approx I(A:B)}$, meaning Eve can get the same amount of info as Bob about Alice, reinforcing why the protocol should be shut down based on the critical QBER estimate. As shown in Figure 2, for $\boldsymbol{EIR = 1.0}$, the average QBER value is $\boldsymbol{\approx 0.25}$, which is considered the maximum average value for a lossless quantum channel. For $\boldsymbol{EIR = 0.5}$, the average QBER value is $\boldsymbol{\approx 0.125}$, which can be considered an alert value, since it is very close to the limit at which Eve can have mutual information equivalent to Bob's, in relation to Alice.

Figure 3: (Color online) Mutual Information vs Eve's Interception Rate (EIR). Note that as EIR increases, $\boldsymbol{I(A:E)}$ increases and, consequently, $\boldsymbol{I(A:B)}$ decreases, in line with what is proposed in the literature [2].



5. Conclusions

Quantum key distribution uses the principles of quantum mechanics to generate keys for communication between parties, with the advantage of attack detection. In this context, the BB84 protocol proposes quantum key distribution between Alice and Bob, using light polarization states in single-photon mode, with preparation and measurement and post-processing steps. The implementation of the BB84 protocol in physical hardware is challenging and requires considerable resources, considering the necessary experimental apparatus.

In this work, we use the Python language package QuTip [4] to simulate the implementation of the protocol between two parties, allowing us to explore crucial quantities for the protocol, such as the QBER rate and mutual information values for different key sizes and in different scenarios of Eve's performance during the execution of the protocol. The results obtained agree with the literature and show the potential of using simulations for the study of quantum communication and quantum key distribution, allowing valuable insights and a practical approach to the BB84 protocol. In terms of performance, for the data obtained in Figure 1, the average execution times obtained for $EIR = 1.0$ were $137.6\ ms$, $423.3\ ms$, and $24.67 s$ for RKL of 64B, 0.25 kB, and 16 kB, respectively. For $EIR = 0.5$, the average values obtained were $110.43\ ms$, $349.4\ ms$ and $19.7\ s$, respectively, such that simulations with $EIR = 0.5$ take on average approximately 0.81% of the execution time for cases with $EIR = 1.0$ (values obtained with a notebook with i7-9750h processor). These execution times are

favourable for the educational tool, as they show the low computational cost of obtaining baseline relationships and results for interpreting the BB84 protocol. Finally, it is hoped that this work will serve as inspiration for future work on quantum key distribution, to compare results obtained via simulation with results found experimentally, considering real quantum channels, and therefore with the presence of noise.

Acknowledgement

This work has been fully funded by the project "Circuito formativo em Tecnologias Quânticas-QIN-FCRH-2025-2-1-1" in supported by QuIIN - Quantum Industrial Innovation, EMBRAPII CIMATEC Competence Center in Quantum Technologies, with financial resources from the PPI IoT/Manufatura 4.0 of the MCTI grant number 053/2023, signed with EMBRAPII.

References

[1] C. H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing. Theoretical computer science. 2014; 560 (7).
[2] R. Wolf and R. Wolf, Quantum key distribution protocols, Quantum Key Distribution: An Introduction with Exercises. Cham: Springer Nature Switzerland; 2021; p. 93-98.
[3] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information. Cambridge: Cambridge University Press; 2010; p. 587-589.
[4] Johansson, J. Robert, Paul D. Nation, and Franco Nori. "QuTiP: An open-source Python framework for the dynamics of open quantum systems." Computer physics communications. 2012; 183(8): 1760-1772. Avaliable from:. https://doi.org/10.1016/j.cpc.2012.02.021
[5] Author(s). Title of article. Journal Title. Year;Volume(Issue):Pages. Available from:.