

Detection of Attacks in BB84 Quantum Key Distribution Using Qiskit-Aer and Machine Learning

^a Giovanna de Freitas Velasco, ^b Karel Mls

^a Escola de Engenharia de São Paulo/Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, São Carlos, São Paulo, Brazil, gf.velasco@usp.br

^b University of Hradec Králové, Hradec Králové, Czech Republic, karel.mls@uhk.cz

Abstract. Quantum Key Distribution (QKD) protocols like BB84 offer theoretically secure communication by using the principles of quantum mechanics. However, practical implementations are vulnerable to the noise in quantum channels and different attacks. In this paper, we simulate the BB84 protocol under both ideal and adversarial conditions using the Qiskit-Aer framework from IBM. We incorporate quantum channel imperfections such as depolarizing noise, readout errors, bit-flips and phase-flips. We also simulate the Intercept-Resend, Trojan Horse and Photon Number Splitting attacks. To detect the presence of such attacks, we analyse statistical data from measurement results and train a Random Forest machine learning model to classify the security of the channel. The results show promising accuracy in attack detection, even in noisy environments, highlighting the potential of AI-based methods to improve the security of practical QKD systems.

Keywords. Quantum Key Distribution, Quantum Cryptography, Noisy Channels, BB84, Qiskit, Qiskit-Aer, Quantum Security, QBER, Random Forest.

1. Introduction

Quantum computing has the potential to change the way computational problems are approached by using the quantum properties of particles. One of the fields going through significant transformation is cryptography, which is currently adapting to this new technology.

In 1994, Peter Shor proposed a quantum algorithm [1] capable of efficiently factoring large integers, a task considered computationally impractical for classical computers. Modern cryptographic systems, such as the RSA Protocol —introduced in 1977—rely on the difficulty of factoring large numbers into their prime components [2]. While generating a public key from two prime numbers is an easy task, the reverse process remains extremely time-consuming for classical systems, requiring thousands of years to break a 2048-bit key.

Shor's algorithm, however, showed that, with a sufficiently powerful quantum computer equipped with error correction, this operation could be made in minutes. Although quantum computers of this scale do not exist yet, rapid advancements in the quantum technologies field are a sign of the need

for quantum-resistant cryptographic protocols.

Before the proposal of Shor's algorithm, Bennett and Brassard [3] introduced the BB84 protocol in 1984, which laid the foundation for Quantum Key Distribution (QKD). This protocol uses the quantum states of photons to securely exchange the secret keys, providing a good level of confidentiality based on a few theorems of quantum mechanics. However, the practical implementations of BB84 may be vulnerable to imperfections in physical channels and to attacks in the quantum channels.

In this context, machine learning techniques can be a good way to improve the security of the QKD by performing an analysis in the quantum channel. This paper explores the simulation of the BB84 protocol under noisy conditions and attacks, using AI-based methods to verify the true level of privacy of the shared key.

2. Methodology

To illustrate the BB84 protocol's behavior and possible vulnerabilities, we created simulations using Qiskit, the quantum computing language developed by IBM. While Qiskit [4] provides the

tools for creating quantum circuits, Qiskit-Aer is used to model the noise in the quantum channel.

To decide if the shared key is truly safe to be used, we generate a dataset from multiple simulations under secure and compromised conditions. This dataset is used to train the machine learning model, in this case, a Random Forest classifier. Even though this paper focuses on a small-scale demonstration, the goal is to present the possibility of using machine learning to detect attacks in QKD systems.

3. The BB84 Protocol

The BB84 protocol uses two different channels to transmit information: a quantum channel and a classical channel. The two parties that will communicate with each other (we call them Alice and Bob) will use the quantum channel to send the polarized photons, which will be used to generate the key.

The protocol relies on the quantum property of polarization. When unpolarized light passes through a vertical polarizing filter, only the component oscillating in the vertical direction passes through. If this vertically polarized light is then passed through a horizontal filter, no light emerges on the other side. This is because vertical and horizontal polarizations are orthogonal, and thus mutually exclusive.

A similar phenomenon occurs with diagonal polarizations: when two diagonal filters are oriented orthogonally (at 45° and 135°), light that passes through the first is blocked by the second.

In quantum mechanics, polarization states can be modeled as superpositions of basis states. That is, before measurement, a photon's polarization can be described as a combination (superposition) of vertical and horizontal basis states or of the two diagonal bases. This indeterminacy is fundamental to the protocol.

The BB84 protocol proceeds as follows:

1. Alice generates a random binary string and a corresponding random sequence of the two possible polarization basis: rectilinear (0°/90°) or diagonal (45°/135°).
2. She encodes each bit as a polarized photon according to the chosen basis. Conventionally, a horizontally or 45° polarized photon represents binary 0, while a vertically or 135° polarized photon represents binary 1.
3. Alice sends the photons to Bob over the quantum channel.
4. Bob independently and randomly chooses a basis (rectilinear or diagonal) for each

incoming photon and measures their polarization accordingly, recording the outcomes as a binary string.

5. Alice and Bob then communicate over the classical channel to compare which basis they used (but not the bit values). They discard all bits where their bases didn't match.
6. The remaining bits — where both used the same basis — form the raw key, which is privately shared between Alice and Bob.

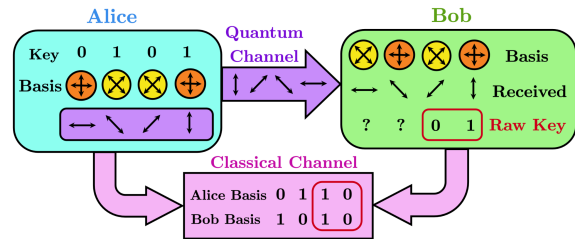


Fig. 1 - Illustration of the BB84 Protocol.

4. Security of Channels

4.1 Quantum Bit Error Rate (QBER)

In real-world applications, quantum channels are not perfect and naturally present some level of noise and errors. Noise can be viewed as the undesirable transformations suffered by a quantum system during the transfer from the sender to the receiver. Common sources of noise in quantum channels include depolarization, bit-flip, phase-flip, amplitude damping, and phase damping [6]. Additionally, there may be a non-negligible probability of measurement errors, known as *readout errors*, where the measured outcome is different from the actual quantum state.

It is important to take this noise into consideration when analysing the error rate in a shared key, since error can be introduced naturally without an eavesdropper in the channel. To measure the amount of error in Alice's and Bob's keys, the Quantum Bit Error (QBER) can be verified. It is defined as the ratio between the number of differing bits and the total number of bits compared. In the simulations presented in the paper, the QBER will be the parameters used to determine if the quantum channel is under attack.

4.2 The Intercept-Resend Attack

In the intercept-resend attack, there is an eavesdropper in the channel (usually named Eve) that attempts to measure the photon that Alice sends to Bob before he can measure it. In order to detect the presence of Eve, Alice and Bob publicly compare a part of their shared key after the process and verify the error rate between them. If the error rate exceeds a certain value, they may conclude that

the quantum channel has been compromised and discard the key.

This security is guaranteed by the No-Cloning Theorem [5], which states that it is impossible to create an identical copy of an arbitrary unknown quantum state without disturbing it. If Eve attempts to intercept and measure the photons, she must choose her own measurement bases. Since she cannot perfectly replicate the original quantum states, her interference introduces errors. So, when Bob performs his measurements and compares his results with Alice, the QBER is higher. Thus, quantum mechanics itself ensures the protocol's security [5].

4.3 The PNS Attack

In an ideal scenario, BB84 assumes that Alice sends single photons. However, in practice, many QKD systems use weak coherent pulses (laser pulses with a low average photon number) because true single-photon sources are technologically difficult to implement. These weak pulses may occasionally contain two or more photons.

In a PNS attack, Eve takes advantage of these multi-photon pulses by splitting off one photon and allowing the rest to continue to Bob undisturbed [6]. Eve can store the photon she captured in a quantum memory and, after Alice and Bob disclose the bases used via the classical channel, Eve measures it in the correct basis without introducing detectable errors. This makes the attack particularly difficult to detect because it does not increase the QBER in an expressive manner.

Since this attack is performed on the hardware, the simulation of the PNS attack will serve mainly as a way to illustrate the behaviour of the channels in this scenario and perform a security analysis.

4.4 The Trojan-Horse Attack

The Trojan-horse attack also targets the physical implementation of the QKD hardware. It extracts information by exploiting the optical components used in the quantum devices.

In this attack, Eve injects bright light pulses into Alice's or Bob's quantum device through the quantum channel. This light can reflect off internal optical components and return to Eve with information about the internal configuration of the device [6]. This means that Eve might learn which basis Alice or Bob is using, allowing her to perform more interceptions without introducing detectable errors.

5. Simulation in Qiskit-Aer

5.1 Noiseless BB84

To simulate the BB84 protocol and possible attacks, we used IBM's Qiskit and Qiskit-Aer quantum computing framework. In quantum computation, the states of qubits are manipulated using quantum logic gates, which act similarly to filters in the physical BB84 implementation. For a comprehensive introduction to quantum computation and information, we refer to Nielsen and Chuang [7], which also presents a good introduction to the basics of quantum mechanics.

In our simulation, qubits represent photons and exhibit the same property of superposition, which is fundamental to the protocol. We begin by creating qubits for Alice, as well as a random binary string representing the polarization bases to be used.

Since all qubits start in the $|0\rangle$ state, Alice has a 50% chance of flipping the bit using an X gate. If she chooses the diagonal basis for a given qubit, she applies a Hadamard (H) gate, effectively rotating the basis from rectilinear ($|0\rangle/|1\rangle$) to diagonal ($|+\rangle/|-\rangle$).

In Fig.2 we have an example of a BB84 quantum circuit using 4 qubits. In this case, Alice chooses to send the bit string 0010 and the basis string 1011. This means she applies the X gate to q_2 , and Hadamard gates to q_0 , q_2 and q_3 . The process of sending qubits to Bob is represented by a barrier, a visual separator that does not affect the computation but helps structure the circuit.

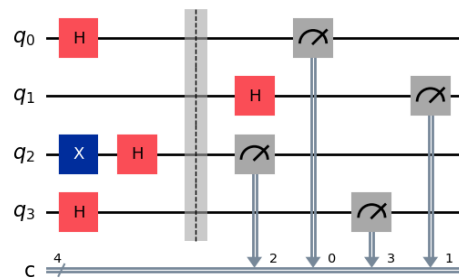


Fig. 2 - Example of simple BB84 quantum circuit.

Bob, independently, chooses the basis string 0100, applying a Hadamard gate only to q_1 . He then measures all qubits, and the results are stored in classical memory registers (denoted by c).

After exchanging basis information through the classical channel, Alice and Bob retain only the bits where their basis choices matched. In this example, the qubits q_0 and q_3 are measured using the same basis by both parties, and therefore their outcomes are kept as part of the shared secret key.

5.2 Noise Simulation

The Qiskit-Aer framework has a noise model that can be easily imported in the simulation. We can add multiple types of error that can be passed to the simulator backend.

Depolarization occurs in a quantum channel when a qubit, instead of maintaining its state through transmission, is randomly transformed into another state due to interactions with the environment. In Qiskit Aer, depolarizing error is modeled by applying a random Pauli operation X, Y, or Z to a qubit with some probability p . The circuit implementation of a depolarizing channel is present in [7].

Readout errors occur when a measurement device reads an incorrect value — for example, measuring a $|0\rangle$ state but recording a $|1\rangle$. In Qiskit, readout errors can be modeled by defining a probability matrix for flipping the measured value.

Bit-flip errors occur when a qubit flips from $|0\rangle$ to $|1\rangle$ or vice versa, analogous to classical bit errors. In Qiskit Aer, a bit-flip error is modeled by applying the Pauli-X gate to the affected qubit with a certain probability p [7].

Phase-flip errors affect the relative phase of a qubit's state by flipping the sign of the $|1\rangle$ component. In Qiskit Aer, phase-flip errors are simulated by applying a Pauli-Z gate with probability p , such as the previous errors.

Amplitude damping models energy dissipation processes and phase damping captures the loss of quantum coherence without energy exchange, simulating dephasing due to interactions with the environment. Although amplitude damping and phase damping offer realistic models of quantum systems, they significantly increase the complexity and memory consumption in simulations. So, to maintain performance, this paper focuses on simulating depolarizing noise, readout errors, bit-flip, and phase-flip errors, which provide a representative yet computationally feasible approximation of noise in quantum channels.

The depolarizing error, bit-flip error, and phase-flip error were simulated with a probability of 10%. These values were chosen to simulate a moderately noisy quantum channel where imperfections in quantum gates are likely to occur. A readout error was incorporated with asymmetric probabilities to reflect the inaccuracies during classical measurement (2% chance of reading a $|0\rangle$ as a $|1\rangle$ and 3% chance of the opposite).

5.3 Attacks Simulation

The Intercept-Resend attack was simulated by generating two quantum circuits: one in which Alice prepares her qubits and Eve measures them. The other circuit is between the basis of Eve and Bob. A

part of the final key obtained by Bob is compared with the corresponding part of Alice's key, and the QBER is computed.

The Trojan Horse attack was simulated by randomly flipping the states of the qubits before they were measured by Bob, altering the information Alice initially sent. This introduces additional errors in the measurement outcomes, resulting in a higher QBER.

In the simulation, the PNS attack was modeled by randomly flipping the qubit states with a low probability, mimicking the behavior of Eve intercepting and measuring the photons. This introduces a slight disturbance to the qubits, leading to errors in Bob's final measurement.

Each attack was introduced with subtle variations to avoid overly aggressive manipulation that might skew the QBER distributions unrealistically.

5.4 QBER Calculation

In a real-world implementation of the BB84 protocol, Alice and Bob must reveal and compare a random subset of their measurement bases through a classical authenticated channel. This process ensures that they only retain bits for the final key where their chosen measurement bases match. The same approach is adopted in our simulation to accurately reflect the BB84 protocol's behavior.

In the simulation, this selective comparison is implemented by filtering the bit positions where Alice and Bob used the same basis. Only these positions are used to extract the expected and received keys for QBER (Quantum Bit Error Rate) analysis. The parameter that determined the percentage of bits in the key to be shared was chosen to be 20%, in order to guarantee that Alice and Bob still have plenty of bits left for the shared key while still being able to check if the channel was safe.

6. Machine Learning Algorithm

Alice and Bob must decide whether to trust the keys they have obtained through the BB84 protocol. As discussed, noise in the quantum channel can lead to discrepancies between their respective keys, either due to natural errors or the presence of an attack on the channel. To help in this decision, we propose the use of machine learning to detect and classify the security of the channel.

For this analysis, we employed a Random Forest classifier, that is a widely used machine learning model for classification tasks. This model was chosen because it has a good capacity to handle large datasets and recognize patterns. The Random Forest constructs multiple decision trees and combines their outputs to make a prediction. Each

tree is trained on a random subset of the data, and the overall prediction is based on a majority vote across all trees. This approach helps reduce overfitting, which makes the model more robust to noise.

However, the Random Forest model demands sufficient training data to effectively learn meaningful patterns. Given that this project aims to demonstrate the practicality of using machine learning to evaluate quantum channel security, this classifier serves as an appropriate proof of concept.

7. Results

7.1 QBER Analysis

The simulation was repeated for 1,500 iterations, with 256 qubits per run, to ensure statistically accurate results. This setting provided a dataset for analyzing the effects of both natural quantum noise and different attack strategies. To verify the QBER in different scenarios, we analyse the box plot in **Fig 3**.

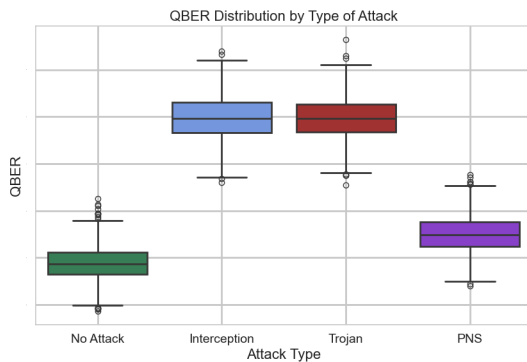


Fig. 3 - Box-plot of QBER in different attacks.

The box plot representation of the QBER values provides a good visualization of how the error rate varies in each condition. In the “No Attack” scenario, the interquartile range is narrow, with values concentrated between approximately 0.15 and 0.22, and a median close to 0.19. This indicates a stable error profile in the channel without attacks, even with the presence of noise.

In contrast, the "Intercept-Resend" and "Trojan Horse" attack scenarios exhibit QBER distributions centered around 0.5, with wider interquartile ranges and maximum values exceeding 0.7 in some cases. The high median and extended upper whiskers show that these attacks interfere strongly in the transmission of photons in the channel.

The “PNS” attack has a box plot distribution that shows that it is not as aggressive as the other two. Its median QBER is around 0.25, with values ranging from 0.11 to 0.43. This intermediate profile reflects the probabilistic nature of the PNS attack, which selectively manipulates multi-photon signals rather than disturbing all of the transmitted qubits.

7.2 Classification Results

The simulation is present in the Github repository listed in [9] and can be executed easily with the installation of Qiskit and Qiskit-Aer. We encourage the readers who want to learn how the model was made to alter different parameters and verify the changes. To analyse the performance of the classification, we verify the Confusion Matrix obtained, present in **Fig 4** and the classification report present in **Table 1**.

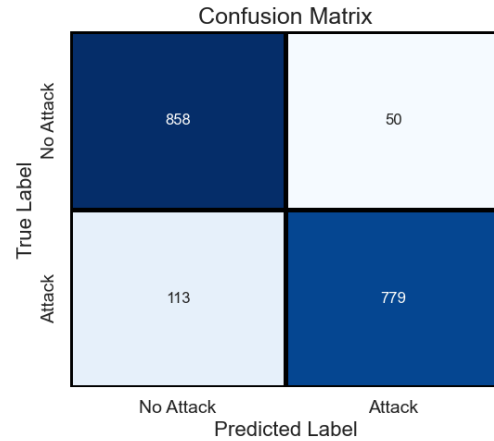


Fig. 4 - Confusion Matrix of the Random Forest.

Tab. 1 - Classification Report of the Random Forest.

Class	Precision	Recall	F1 Score	Support
No Attack	0.88	0.94	0.91	908
Attack	0.94	0.87	0.91	892
Accuracy	-	-	0.91	1800
Macro Avg	0.91	0.91	0.91	1800
Weighted Avg	0.91	0.91	0.91	1800

Using only the QBER as the classification feature, the Random Forest classifier obtained an accuracy of 91%. Precision and recall values were high for both classes: the model correctly identified *No Attack* instances with a precision of 0.88 and a recall of 0.94, and it detected *Attack* scenarios with a precision of 0.94 and a recall of 0.87. The F1-score for both classes was 0.91, a balanced and effective performance.

These results suggest that QBER is a strong indicator of security anomalies in the quantum communication process. Notably, the relatively high recall for the *No Attack* class indicates a low false positive rate, while the strong precision for the *Attack* class demonstrates the model’s reliability in detecting these attacks.

8. Conclusions

These results show that the attacks in the quantum channel can be simulated in an effective manner that models the real behaviour in Qiskit-Aer. The QBER results were close to what was theoretically expected and the model was overall successful in detecting the attacks in the quantum channel, especially such as the Trojan and the Intercept-Resend.

In practical applications of QKD, the use of machine learning can be a useful solution to ensure the security of the quantum channel. It is necessary, of course, to guarantee the security of the classical channel, which can be a good complementation to this paper. Furthermore, it is also possible to use different models in various places in the channel: there can be a model for Alice and a different model for Bob in order to manipulate the weight for other parameters besides the QBER, such as the entropy of the shared key [6].

In summary, this simulation has successfully demonstrated the possibility of integrating machine learning in QKD, while still introducing other aspects that can be considered and analysed in order to improve the security of quantum channels. It is possible to use other types of models, include the amplitude and phase damping (although it has a considerable memory cost), and analyse other parameters, such as the entropy of the key. The applications in IoT can benefit from the further development of these analyses of QKD protocols to assure the privacy of communication in the era of quantum technology.

9. References

- [1] Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Santa Fe, New Mexico, USA; 1994. 124-134. doi.org/10.1109/SFCS.1994.365700.
- [2] Rivest RL, Shamir A, Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*. 1978;21(2):120-126. doi.org/10.1145/359340.359342
- [3] Bennett CH, Brassard G. Quantum cryptography: public key distribution and coin tossing. *Theoretical Computer Science*. 2014;560:7-11. doi.org/10.1016/j.tcs.2014.05.025. (Reprint of: *Proceedings of the International Conference on Computers, Systems & Signal Processing*. 1984;1:175-179).
- [4] Javadi-Abhari A, Treinish M, Krsulich K, Wood CJ, Lishman J, et al. *Quantum computing with Qiskit*. 2024. doi.org/10.48550/arXiv.2405.08810
- [5] Wootters W, Zurek W. A single quantum cannot be cloned. *Nature*. 1982; 299: 802-803. doi.org/10.1038/299802a0
- [6] Wolf R. *Quantum Key Distribution*. Springer, Cham;2021.229 doi.org/10.1007/978-3-030-73991-1
- [7] Nielsen MA, Chuang, IL. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, Cambridge; 2010.702. doi.org/10.1017/CBO9780511976667
- [8] Wolf R. *Quantum Key Distribution*. Springer, Cham;2021.229 doi.org/10.1007/978-3-030-73991-1
- [9] Github Repository with codes used to generate the Figures 2, 3, 4 and the results analysed: github.com/giovelasco/BB84_Protocol