Simulation Report: The BB84 Quantum Key Distribution Protocol

Simulation Analysis (Based on Qiskit Implementation)

November 2, 2025

Abstract

This report details the simulation of the Bennett-Brassard 1984 (BB84) quantum key distribution (QKD) protocol using Qiskit. The objective is to demonstrate the protocol's mechanics, including quantum state encoding, basis reconciliation, and, most critically, eavesdropper detection. The simulation is executed in two scenarios: a secure channel without an eavesdropper and an insecure channel with an active eavesdropper ("Eve") performing an intercept-resend attack. The results clearly show that in the secure case, a key is established with a 0% Quantum Bit Error Rate (QBER), while Eve's presence introduces a high QBER ($\approx 25\%$), which is successfully detected, leading to the key being discarded.

Contents

1 Introduction

Secure communication relies on the ability of two parties, ”Alice” and ”Bob,” to share a secret key without a third party, ”Eve,” intercepting it. Classical cryptography is vulnerable to computational advances (e.g., a quantum computer breaking RSA).

Quantum Key Distribution (QKD) offers a solution based on the laws of quantum physics. The BB84 protocol, proposed by Charles Bennett and Gilles Brassard in 1984, is the foundational QKD protocol. Its security is guaranteed by the no-cloning theorem (an unknown quantum state cannot be perfectly copied) and the fact that measurement disturbs a quantum system.

This report documents a Qiskit-based simulation of the BB84 protocol. The primary goal is to simulate the key exchange process and validate its core security promise: the ability to detect the presence of an eavesdropper.

2 Methodology & Approach

The simulation models the three main phases of the BB84 protocol: the quantum channel, the public channel (sifting), and security verification.

2.1 The BB84 Protocol (Approach)

2.1.1 Phase 1: Quantum Channel (Encoding & Measuring)

Alice generates two random classical strings: one for her bits (e.g., $0110\dots$) and one for her bases (e.g., $ZXZX\dots$). She encodes each bit onto a single qubit using the corresponding basis.

- **Z-basis (’+’):** $0 \rightarrow |0\rangle$, $1 \rightarrow |1\rangle$
- **X-basis (’x’):** $0 \rightarrow |+\rangle$, $1 \rightarrow |-\rangle$

She sends these qubits to Bob over the quantum channel. Bob, unaware of Alice’s basis choices, generates his own random string of bases and measures each incoming qubit.

2.1.2 Phase 2: Public Channel (Key Sifting)

Alice and Bob move to a classical, authenticated public channel.

1. Bob announces the bases he used for measurement.
2. Alice announces which of his bases were correct.
3. Both parties discard all bits where their basis choices did not match.

The remaining, correlated string of bits is known as the ”sifted key.” In an ideal, secure channel, Alice’s and Bob’s sifted keys will be identical.

2.1.3 Phase 3: Eavesdropper Detection (QBER)

To check for an eavesdropper, Alice and Bob publicly compare a random subset of their sifted key bits. They calculate the mismatch percentage, known as the **Quantum Bit Error Rate (QBER)**.

If the QBER is 0 (or below a small threshold for channel noise), they can be confident the key is secure. If the QBER is significantly high, they conclude an eavesdropper was present and **discard the entire key**.

2.2 Simulation Design Choices

The simulation implemented this protocol with several key design choices.

2.2.1 Simulation of Eve: Intercept-Resend

An active eavesdropper (”Eve”) was modeled using a simple ”intercept-resend” attack.

- **Strategy:** Eve intercepts every qubit from Alice.
- **Action:** For each qubit, Eve guesses a random basis (Z or X) and measures it.
- **Result:** She then sends a new qubit to Bob, encoded with her measured bit in her chosen basis.

This attack is fundamental to BB84's security. When Eve's basis does not match Alice's, her measurement is random, and she forwards a corrupted state to Bob. This introduces errors in Bob's measurements, even when his basis matches Alice's.

Theoretically, if Alice and Bob's bases match, but Eve's does not, Eve has a 50% chance of measuring the wrong bit. When Bob measures the (wrong) state Eve sent, he has a 50% chance of getting an error. This results in an expected QBER of $\mathbf{0.5} \times \mathbf{0.5} = \mathbf{25}\%$ in the sifted key.

2.2.2 Simulation Optimization: Batch Execution

A key design choice for performance was to use batch execution. Instead of creating, transpiling, and running a new circuit for each of the 1000+ qubits (which is very slow), the simulation:

1. Loops through all 1000+ protocol steps.
2. *Builds* the final measurement circuit for each step.
3. Appends all these circuits to a single list.
4. Transpiles and runs the entire list as a single batch job on the `AerSimulator`.

This significantly reduces the overhead of simulation and is a much more efficient design.

2.2.3 Security Parameter

The simulation defines a `QBER_THRESHOLD = 0.10` (10%). Any QBER above this value is considered insecure, and the key is discarded.

3 Results & Analysis

The simulation was executed twice: once without Eve and once with Eve active. Both simulations attempted to generate a 1000-bit raw key. The console output from the simulation script provides the definitive results.

3.1 Scenario 1: No Eavesdropping (Secure Channel)

In this scenario, `eve_active=False`. The only source of difference between Alice and Bob's keys is the basis mismatch, which is removed during sifting.

- **Sifted Key Length:** ≈ 500 bits (as expected, ≈ 50% of bases match).
- **Number of Errors:** 0

- **QBER:** 0.0000
- **Conclusion:** The QBER (0%) is well below the 10% threshold. The key is considered **SECURE**.

3.2 Scenario 2: With Eavesdropping (Insecure Channel)

In this scenario, `eve_active=True`. Eve intercepts and resends every qubit.

- **Sifted Key Length:** ≈ 500 bits.
- **Number of Errors:** ≈ 125
- **QBER:** ≈ 0.25 (25%)
- **Conclusion:** The QBER ($\approx$ 25%) is significantly higher than the 10% threshold. The key is considered **INSECURE** and is discarded.

3.3 Summary of Results

The outputs from the two simulation runs are summarized below.

Table 1: Comparison of Secure vs. Insecure BB84 Simulation (N=1000)

Metric	**Scenario 1: No Eve**	**Scenario 2: With Eve**
Sifted Key Length	~500	~500
Errors Detected	0	~125
QBER	**0.00**	$\approx$ **0.25**
Security Threshold	0.10	0.10
Outcome	**SECURE**	**INSECURE (Key Discarded)**

4 Conclusion

The Qiskit simulation successfully and efficiently modeled the BB84 protocol. The results perfectly align with the theoretical promises of quantum key distribution.

In the secure scenario, a sifted key was established with a 0% QBER, resulting in a shared secret. In the insecure scenario, the eavesdropper's intercept-resend attack introduced a catastrophic QBER of approximately 25%, far exceeding the 10% security threshold. This detection forced Alice and Bob to discard the compromised key, preventing any information leak.

This simulation confirms that the security of BB84 does not come from preventing eavesdropping, but from **guaranteeing its detection**.