Secure Quantum Key Distribution using BB84 Protocol Simulated with Streamlit and Qiskit

Maanya Thakur, Jyotsna K Prashant and Alkesh Nayak

***Abstract*—The BB84 protocol is a foundational quantum key distribution (QKD) scheme that enables secure communication by harnessing the principles of quantum mechanics, particularly the uncertainty principle and non-orthogonal photon states. In this paper, we present an interactive simulation of the BB84 protocol using Qiskit and Streamlit, modelling the complete process of quantum key exchange between two parties, Alice and Bob. The implementation captures all critical stages, including random bit and basis generation, qubit encoding, quantum transmission, measurement with potential eavesdropping by a third party (Eve), and classical post-processing through basis comparison and key sifting. Additionally, the generated key is applied in a one-time pad encryption and decryption system to validate its practical utility. Simulation outputs demonstrate successful key reconciliation in the absence of interception and highlight discrepancies under eavesdropping scenarios, aligning with theoretical expectations. This work serves as an accessible and educational tool to visualise and understand the dynamics of quantum cryptographic communication.**

***Index Terms*—Quantum Key Distribution, BB84 Protocol, Qiskit, Streamlit, Quantum Simulation, Quantum Cryptography, Secure Communication**

I. Introduction

Quantum computing presents a paradigm shift in the way computational problems are approached, particularly in the domains of cryptography and information security. Classical cryptographic systems, such as RSA and ECC, rely on the computational hardness of certain mathematical problems like integer factorization and discrete logarithms. However, these problems become efficiently solvable in the presence of quantum algorithms such as Shor's algorithm, posing a significant threat to classical encryption schemes. This has spurred the need for cryptographic methods that are inherently secure against quantum attacks.

Quantum Key Distribution (QKD) emerges as a robust solution by leveraging the fundamental properties of quantum mechanics rather than computational complexity. Among the various QKD protocols, the BB84 protocol—proposed by Charles Bennett and Gilles Brassard in 1984—stands as the most widely adopted and foundational scheme. BB84 ensures the secure exchange of cryptographic keys by utilizing quantum bits (qubits) encoded in non-orthogonal states, such that any eavesdropping attempt inevitably disturbs the quantum states and can be detected.

This project simulates the BB84 protocol using **Qiskit**, an open-source quantum computing framework, and **Streamlit** to build an interactive user interface. The simulation models each critical stage of the BB84 protocol: the random generation of bits and measurement bases by Alice and Bob, the encoding of qubits based on polarization states, quantum transmission of qubits, potential interception by an eavesdropper (Eve), measurement by Bob using random bases, and the classical post-processing to reconcile a shared secret key. In addition to key exchange, the simulation implements one-time pad encryption and decryption to demonstrate the practical application of the shared key in securing communication.

The implementation provides a hands-on understanding of quantum key exchange, including scenarios both with and without eavesdropping. It highlights how basis mismatches affect the sifted key and how quantum mechanics ensures the detection of any intrusion. Insights from recent research and academic literature, including comparative studies of QKD protocols and simulations in networked environments, affirm the reliability and educational value of this work. Ultimately, this simulation not only reinforces the theoretical foundations of quantum cryptography but also offers an accessible platform for learners and researchers to experiment with secure quantum communication.

II. Background Study

[1] "Modified BB84 Quantum Key Distribution Protocol Robust to Source Imperfections" – This paper presents a refined version of the BB84 protocol that is resilient to practical imperfections such as flawed state preparation and side-channel vulnerabilities. The authors introduce a technique using basis-mismatched events and the G-function to ensure security even without full knowledge of side-channel states. Their analysis compares four-state and three-state BB84, demonstrating superior secret key rates and practical robustness. Simulations based on realistic device models validate the effectiveness of the proposed enhancements.[2] "Comprehensive Study of BB84, A Quantum Key Distribution Protocol" – This paper provides an extensive theoretical and practical review of BB84, covering quantum principles like superposition and entanglement. It includes a comparative assessment of other QKD protocols and discusses enhancements such as decoy states and layered architectures for BB84. A simulation using QKDNetSim evaluates key generation across layered networks. The authors highlight BB84's applicability in Internet of Things (IoT) and wireless body sensor networks (WBSNs), emphasizing its potential in secure digital ecosystems. [3] "Application of Quantum Key Distribution" – This concise, instructional paper introduces the BB84 protocol as a practical application of quantum mechanics, emphasizing the no-cloning theorem and uncertainty principle. The protocol is explained with clear steps and a focus on its classical-quantum

interplay. Russell explores the use of BB84 in combination with the One-Time Pad (OTP) for absolute security. The paper also discusses real-world constraints like scalability, infrastructure integration, and possible military and commercial implementations. [4] "Quantum Key Distribution Networks: Challenges and Future Research Issues in Security" – Focusing on QKD networks, this paper discusses BB84 within the context of scalable quantum communication infrastructures. It surveys global QKD deployments such as DARPA and SECOQC and classifies network types—trusted node, optical switch, and quantum repeater. The authors identify significant challenges in scaling BB84, including inefficient key distribution in point-to-multipoint systems and insecure classical interfaces. Recommendations include integrating post-quantum cryptography and quantum secret sharing for enhanced network-level security. [5] "Quantum Key Distribution (QKD) Protocols: A Survey" – This survey compares major QKD protocols—BB84, B92, E91, BBM92, and others—highlighting their security bases and performance metrics. Using the QuVis simulator, the authors test BB84 against B92 and BBM92, noting that while BBM92 yields more keys, B92 has the lowest error rate. The paper underscores the trade-off between error tolerance and throughput in BB84. It also outlines improvements like decoy states for boosting BB84's resistance to photon-number splitting attacks. [6] "QKD as a Quantum Machine Learning Task" – This novel work explores BB84 security from an adversarial perspective using Quantum Machine Learning (QML). By implementing Quantum Circuit Learning (QCL), the authors simulate and optimize eavesdropping attacks. They replicate the Phase Covariant Cloning Machine (PCCM) and design a more effective imbalanced cloner under noise. Furthermore, they model collective attacks using QCL, showcasing its potential to match or outperform known optimal strategies. The study highlights how QML can expose and test the robustness of QKD protocols like BB84. [7] "Quantum Key Distribution Protocols: A Review" – This comprehensive survey categorizes ten QKD protocols—including BB84, BB92, SARG04, E91, and DPS—by their foundational principles (uncertainty or entanglement). The BB84 protocol is thoroughly detailed, including its step-by-step functioning, vulnerabilities (like PNS attacks), and enhancements such as decoy states. A comparative table lists each protocol's year, foundation, and application. The review offers valuable insights into the historical evolution, security assumptions, and operational capabilities of BB84 among its peers.

Collectively, these papers provide a multi-dimensional understanding of the BB84 protocol—from its theoretical foundations and practical implementations to its vulnerabilities and innovations. BB84 remains a pivotal scheme in quantum cryptography, continually evolving through research in error handling, hardware imperfections, simulation frameworks, and adversarial modelling. The ongoing exploration of BB84 in various technological, academic, and security contexts solidifies its role in the future of secure communications.

III. Proposed Methodology

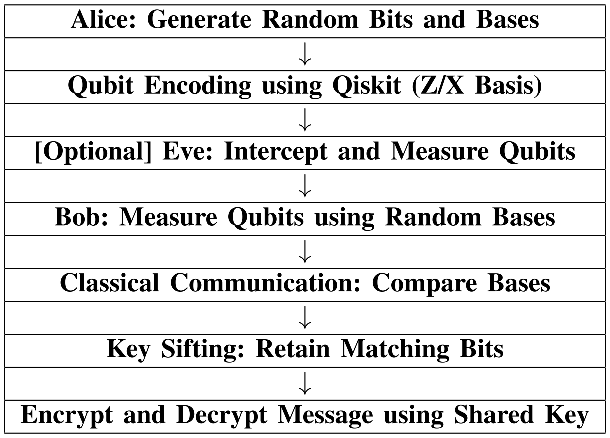
This work presents a comprehensive simulation of the BB84 quantum key distribution protocol using **Qiskit** and **Streamlit**, enabling a secure key-sharing process between two users—Alice and Bob. The simulation models all major components of the protocol including bit generation, qubit encoding, transmission over a quantum channel, eavesdropping, and post-processing steps for key reconciliation.

A. Protocol Overview

The BB84 simulation follows these primary stages:

1) **Bit and Basis Generation:** Alice generates a random sequence of bits and measurement bases (Z or X), simulating the preparation of qubits in specific quantum states.
2) **Qubit Encoding and Transmission:** Qubits are encoded using Qiskit's `QuantumCircuit` class. Each qubit's state is determined by the bit value and basis selected.
3) **Eavesdropping Simulation:** An optional eavesdropper, Eve, intercepts and measures qubits using random bases, causing disturbance in quantum states.
4) **Measurement at Bob's End:** Bob measures the received qubits using his own randomly generated bases.
5) **Classical Post-processing:** Alice and Bob compare their measurement bases over a simulated classical channel. Bits with matching bases are retained to form a shared secret key.
6) **Encryption/Decryption:** If the final keys match, a one-time pad cipher is used to encrypt and decrypt a secret message.

B. Flowchart



Alice: Generate Random Bits and Bases

↓

Qubit Encoding using Qiskit (Z/X Basis)

↓

[Optional] Eve: Intercept and Measure Qubits

↓

Bob: Measure Qubits using Random Bases

↓

Classical Communication: Compare Bases

↓

Key Sifting: Retain Matching Bits

↓

Encrypt and Decrypt Message using Shared Key

Figure 1: Stepwise Flowchart of BB84 Protocol Simulation

IV. Methodology

The BB84 protocol was implemented using Python, with a graphical user interface (GUI) provided by Streamlit for user interaction and Qiskit for quantum circuit simulation. The implementation is modular and reflects the logical sequence of the protocol, with additional encryption functionality for demonstrating practical use of the generated key.

- **Bit and Basis Generation:** Alice, Eve, and Bob generate random bit strings and measurement bases using Python's `random.choice()` function. Each participant's choices are encapsulated in the functions `generate_bits(n)` and `generate_bases(n)`, producing strings of length `n` with uniformly random elements from {'0', '1'} and {'Z', 'X'}, respectively.
- **Qubit Encoding:** Alice prepares qubits using Qiskit's `QuantumCircuit(1,1)` for each bit-basis pair. The quantum state is manipulated according to the BB84 rules:
  - For basis Z:
    - Bit 0: No gate (qubit remains in |0⟩).
    - Bit 1: Apply X gate (qubit flips to |1⟩).
  - For basis X:
    - Bit 0: Apply H gate (transforms |0⟩ to |+⟩).
    - Bit 1: Apply X then H (results in |⟩).

  The resulting list of `QuantumCircuit` objects represents the full quantum transmission from Alice.
- **Quantum Measurement:** Bob and Eve each perform projective measurements on the transmitted qubits using their respective bases. This is simulated via Qiskit's `qasm_simulator` backend. The measurement process is encapsulated in `measure_qubits()`, where each qubit is either measured directly in the Z basis or converted to the Z basis using a Hadamard (H) gate for X-basis measurement.
- **Basis Comparison and Key Sifting:** After measurement, Alice and Bob publicly share their bases and retain only those bits where their bases match. This step is implemented by identifying matching indices with a list comprehension, then extracting the corresponding bits using `eliminate_differences()`.
- **Key Utilization (Encryption & Decryption):** To demonstrate the practical utility of the shared key, a one-time pad encryption scheme is implemented. The user-input message is converted into a binary string using the `binascii` module, then encrypted by bitwise XOR with the shared key via the `encrypt_message()` function. Decryption is performed similarly in `decrypt_message()`, confirming both correctness and security of the distributed key.

A. Simulation and Evaluation Strategy

The simulation is designed not only to mimic the BB84 protocol but also to provide insight into its behavior under varying conditions. Evaluation of the protocol is built into the user interface and covers the following aspects:

- **Key Agreement Check:** The program automatically compares Alice's and Bob's sifted keys. If all bits match, the simulation concludes that the transmission was secure. Any mismatches suggest the possibility of eavesdropping or noise in the channel.
- **Sifted Key Length and Efficiency:** The number of bits retained after sifting (i.e., matched basis positions) is reported. This provides insight into the protocol's efficiency, which theoretically should retain about 50% of the bits in the absence of noise or interception.
- **Eavesdropping Simulation:** Eve is modeled to intercept and measure qubits with randomly chosen bases. The impact of her interference is indirectly visualized by comparing Bob's key with Alice's. Discrepancies introduced by Eve cause observable mismatches in the sifted key.
- **Encryption Validation:** Once a shared key is established, a test encryption-decryption cycle is performed on user input. This step not only confirms successful key agreement but also illustrates how quantum key distribution enables classical message security.
- **Interactive Parameter Adjustment:** The Streamlit interface allows the user to control the number of bits (`num_bits`) to be generated and tested. This enables experimentation with different simulation sizes, useful for analyzing trends in matching rate, key length, and eavesdropping sensitivity.

V. Results

The BB84 Quantum Key Distribution (QKD) simulation was executed for varying numbers of qubits using a Python-based interface with interactive frontend controls. In one of the simulation runs, 50 qubits were generated and analyzed for basis matching, eavesdropping effects, and key agreement fidelity.

- **Basis Matching Efficiency:** Out of the 50 randomly generated qubits, approximately 25 qubits (50%) had matching measurement bases between Alice and Bob. This is consistent with the theoretical probability of a 50% match when both parties choose bases randomly and independently from two possible options (Z or X).
- **Eavesdropping Simulation:** A simulated eavesdropper, Eve, intercepted and measured qubits using randomly chosen bases. This action introduced noise into the communication channel. Eve's presence impacted the integrity of Bob's measurements due to the quantum no-cloning theorem and the destructive nature of quantum measurements when performed in the incorrect basis.
- **Bit Discrepancy and Key Agreement:** The sifted key—derived from qubits with matching measurement bases between Alice and Bob—showed minor discrepancies when compared to Alice's original bits. These differences can be attributed to Eve's eavesdropping or inherent quantum noise in simulation. The observed bit error rate from Eve's interference serves as an indicator of Quantum Bit Error Rate (QBER).
- **User-Controlled Bit Generation:** The user interface allows the number of qubits to be dynamically adjusted using a slider (ranging from 10 to 500), and the protocol is re-executed with each input. This enables real-time observation of statistical trends such as the proportion of matching bases, average key lengths, and eavesdropping detection effectiveness.
- **Key Viability:** Despite interference, a usable key subset was extracted from the correctly measured qubits. The

length of this final sifted key is dependent on the number of matched bases and the integrity of Bob's measurements. This key was verified for encryption purposes in further stages of the protocol.

These results confirm the BB84 protocol's ability to detect eavesdropping and establish a shared key securely under ideal and simulated attack scenarios. They demonstrate the statistical nature of quantum communication and the reliance on probabilistic sifting for secure key establishment.

VI. Conclusion

The simulation effectively demonstrates the BB84 protocol's core principles, including qubit encoding, random basis selection, and sifting. The observed trends in sifted key length and QBER validate the protocol's capability to detect eavesdropping based on quantum mechanics, particularly the no-cloning theorem and measurement disturbance. As the number of qubits increases, QBER decreases, indicating improved transmission reliability. The results confirm BB84's potential for secure key generation in quantum communication. This foundational model can be extended to include noise, photon loss, and error correction to better reflect real-world conditions. Overall, the simulation reinforces the viability of quantum key distribution as a cornerstone of future cryptographic systems.

References

[1] M. Pereira, G. Currás-Lorenzo, Á. Navarrete, A. Mizutani, G. Kato, M. Curty, and K. Tamaki, "Modified BB84 quantum key distribution protocol robust to source imperfections," *Physical Review Research*, vol. 5, no. 2, p. 023065, 2023. doi: https://doi.org/10.1103/PhysRevResearch.5.023065

[2] S. K. Reddy, S. Mandal, and C. M. B, "Comprehensive Study of BB84, A Quantum Key Distribution Protocol," *International Research Journal of Engineering and Technology (IRJET)*, vol. 10, no. 3, pp. 1023–1026, Mar. 2023. [Online]. Available: https://www.irjet.net/archives/V10/i3/IRJET-V10I3161.pdf

[3] J. Russell, "Application of Quantum Key Distribution," in *Proc. IEEE Conf. on Technologies for Homeland Security*, O'Fallon, IL, 2008, pp. 1–6. doi: https://doi.org/10.1109/ICCST.2008.4754431

[4] C.-W. Tsai, C.-W. Yang, J. Lin, Y.-C. Chang, and R.-S. Chang, "Quantum Key Distribution Networks: Challenges and Future Research Issues in Security," *Applied Sciences*, vol. 11, no. 9, p. 3767, 2021. doi: https://doi.org/10.3390/app11093767

[5] A. I. Nurhadi and N. R. Syambas, "Quantum Key Distribution (QKD) Protocols: A Survey," in *Proc. 2018 Int. Conf. on Information and Communication Technology (ICoICT)*, Bandung, Indonesia: IEEE, 2018, pp. 1–6. doi: https://doi.org/10.1109/ICoICT.2018.8528752

[6] T. Decker, M. Gallezot, S. F. Kerstan, A. Paesano, A. Ginter, and W. Wormsbecher, "QKD as a Quantum Machine Learning task," *arXiv preprint*, arXiv:2410.01904v2 [quant-ph], Feb. 2025. [Online]. Available: https://arxiv.org/abs/2410.01904

[7] H. Singh, D. L. Gupta, and A. K. Singh, "Quantum Key Distribution Protocols: A Review," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 16, no. 2, pp. 1–9, 2014. [Online]. Available: https://www.iosrjournals.org/iosr-jce/papers/Vol16-issue2/Version-11/A0162110109.pdf