Quantum Key Distribution (QKD)

By Shivam Naik
225891250

1. Project Overview:
Quantum Key Distribution (QKD) is a secure communication method that leverages the principles of quantum mechanics to share encryption keys, making it practically impossible for eavesdroppers to intercept information without being detected. The goal of this project is to develop a simulation of QKD using Qiskit, IBM's open-source quantum computing framework. The project will cover the implementation of the BB84 protocol, one of the foundational QKD protocols, and will include a simulation of eavesdropping to demonstrate the unique security aspects of quantum cryptography.

2. Motivation:
In classical cryptography, encryption techniques are susceptible to advances in computational power, especially with the potential advent of quantum computing. QKD offers a promising alternative by providing theoretically unbreakable encryption, secured by the laws of quantum physics. Understanding QKD is crucial for anyone interested in quantum cybersecurity, cryptography, or the future of secure communication. This project serves as a practical introduction to quantum cryptography and its real-world implications for data security.

3. Project Objectives:

- Implement a simulation of the BB84 QKD protocol using Qiskit.
- Illustrate the impact of quantum properties such as superposition and entanglement on secure communication.
- Introduce a basic eavesdropping attack to demonstrate the resilience of quantum encryption.
- Analyse the outcomes of QKD under eavesdropping to show the practical advantages of quantum cryptography over classical methods.

4. Project Scope:

- **BB84 Protocol Simulation**: The project will implement the BB84 protocol using Qiskit. The protocol will be simulated between two communicating parties, traditionally referred to as Alice and Bob, for generating and securely sharing a random encryption key.
- **Eavesdropping Simulation**: A third party (Eve) will attempt to intercept the quantum communication. The project will demonstrate how quantum properties reveal the presence of an eavesdropper.
- **Data Analysis**: Analyse the security implications by comparing the key generated under normal conditions with the key generated under eavesdropping, showcasing the drop in fidelity.

5. Learning Outcomes:

- **Quantum Concepts**: Understanding quantum states, superposition, entanglement, and measurement.
- **Quantum Protocols**: Learning about the BB84 protocol, the basic principles of quantum cryptography, and the concept of secure key distribution.
- **Qiskit Implementation**: Gaining hands-on experience with Qiskit for creating quantum circuits and implementing cryptographic protocols.
- **Cybersecurity Application**: Understanding how quantum computing can revolutionize data security and how QKD can serve as a solution against threats posed by classical and quantum computational advancements.

6. Prerequisites:

- **Mathematics**: Basic knowledge of linear algebra, especially matrices and vectors, to understand quantum states.
- **Quantum Mechanics**: Familiarity with the concepts of qubits, superposition, and entanglement.
- **Python Programming**: Knowledge of Python, as Qiskit is based on Python.
- **Classical Cryptography**: Understanding of classical encryption methods and symmetric key cryptography would be beneficial for context.

7. Technologies and Tools:

- **Qiskit**: IBM's open-source quantum computing framework for creating and running quantum circuits.
- **Python**: The programming language used for interacting with Qiskit.
- **Jupyter Notebook**: To run Qiskit code and visualize quantum circuits and results.
- **IBM Quantum Experience**: For executing the QKD protocols on real quantum hardware or quantum simulators available via IBM Cloud.

8. Project Roadmap:

- **Phase 1: Background Study (1-2 Weeks)**
- Study quantum cryptography fundamentals and the BB84 protocol.
- Review basic Qiskit tutorials and quantum computing principles.
- **Phase 2: Qiskit Installation and Setup (1 Week)**
- Install Qiskit and set up the required environment, including Jupyter Notebook.
- Run introductory Qiskit programs to become familiar with quantum circuits.
- **Phase 3: Implementing BB84 Protocol (2-3 Weeks)**
- Create quantum circuits for Alice and Bob to prepare and measure qubits.
- Implement classical communication between Alice and Bob to compare bases and extract a shared key.
- **Phase 4: Eavesdropping Simulation (1-2 Weeks)**

- Add an eavesdropper (Eve) to intercept qubits and analyze the changes.
- Demonstrate how the quantum state collapses upon measurement, leading to detectable errors in the key.
- **Phase 5: Analysis and Reporting (1-2 Weeks)**
- Analyse the generated keys under different scenarios (with and without eavesdropping).
- Visualize the key accuracy and fidelity under attack.
- Prepare a final report and presentation summarizing findings.

9. Challenges and Considerations:

- **Quantum Noise**: Dealing with noise in the quantum channel, which may impact key generation accuracy.
- **Complexity**: Understanding how measurement impacts quantum states, which can be conceptually challenging.
- **Eavesdropping**: Simulating a realistic eavesdropping scenario that accurately demonstrates quantum advantages.

10. Expected Outcomes:

- A working simulation of the BB84 protocol using Qiskit.
- Visualization of key distribution and detection of eavesdropping.
- Practical understanding of the fundamental security benefits offered by quantum cryptography.

11. Real-Life Applications:

- This project is directly applicable to secure communication, particularly in scenarios that require high levels of security, such as government or financial institutions.
- QKD is an emerging technology that may soon be used in secure satellite communications and fibre optic networks, making this project highly relevant for real-world cybersecurity advancements.

12. Resources:

- **Qiskit Documentation and Tutorials**: To get started with Qiskit and learn about quantum programming.
- **IBM Quantum Experience**: To experiment with real quantum computers.
- **Books and Online Courses**:
- “Quantum Computing for Computer Scientists” (Book)
- “Quantum Computing for Everyone” (Coursera or EdX)

This project will not only strengthen your resume by showcasing knowledge of a cutting-edge technology but also provide an opportunity to gain hands-on experience in quantum cryptography, a field at the frontier of cybersecurity and computing.

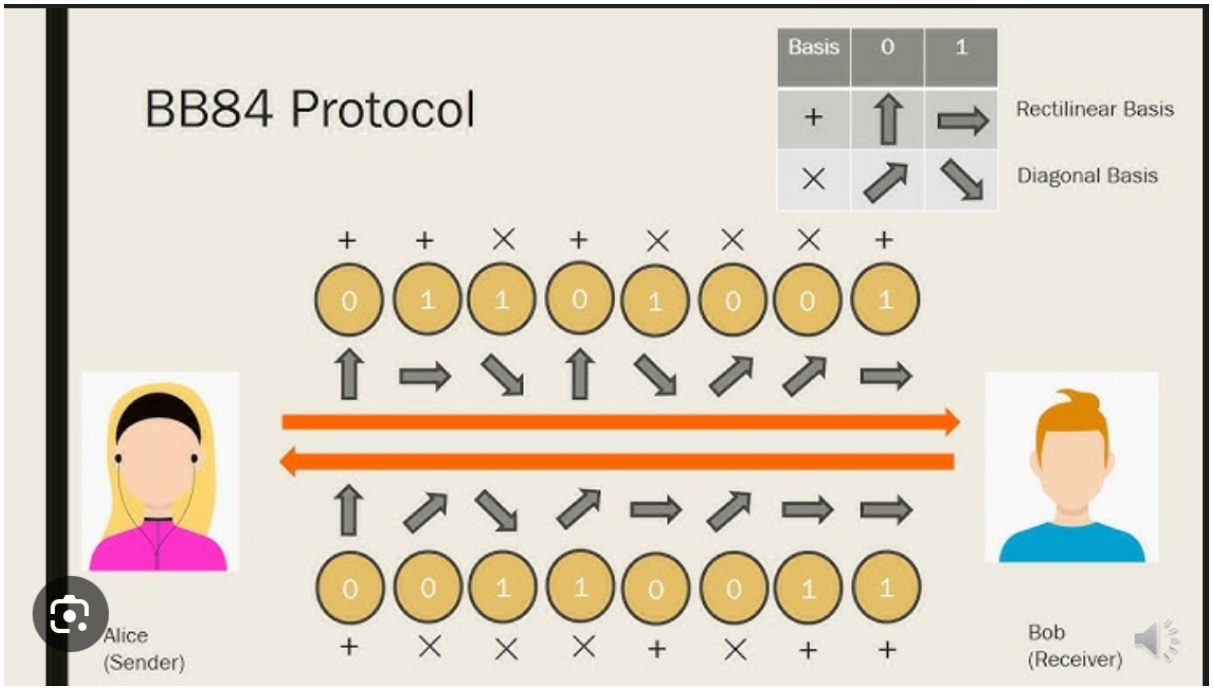
Here's a streamlined summary of all the key steps we implemented in my Quantum Key Distribution (QKD) and secure communication process, without the error-handling details:

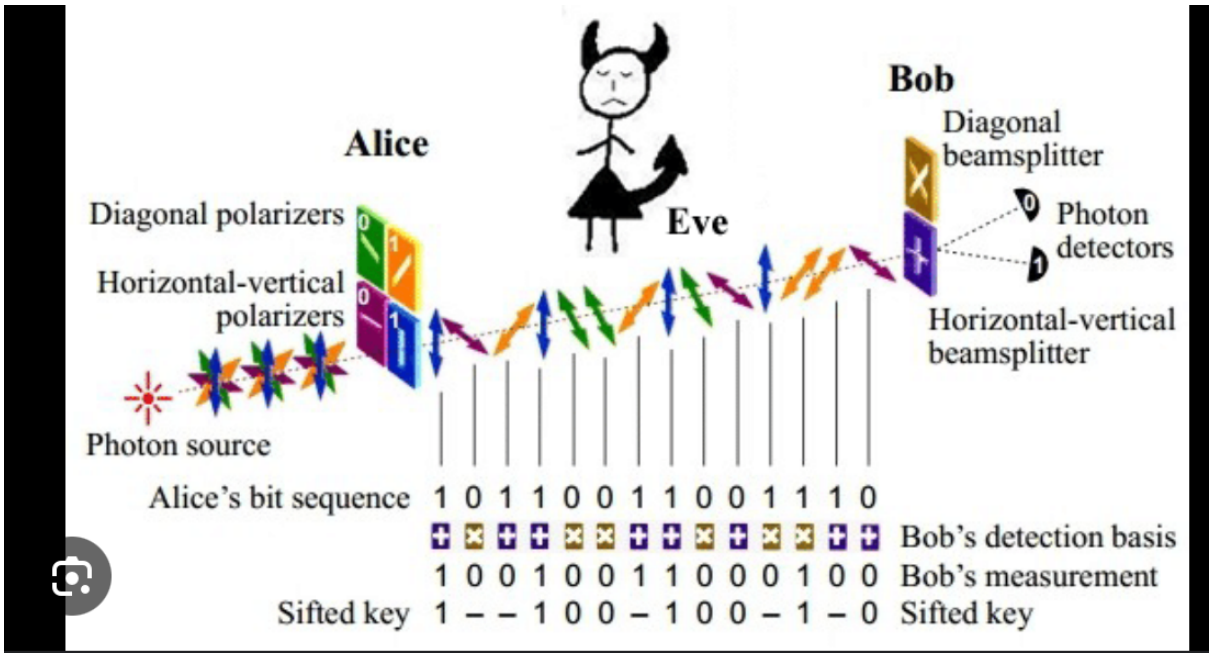
Quantum Key Distribution and Secure Communication Steps

1. **Quantum Circuit Setup for Key Distribution**:
   - Created a quantum circuit with three qubits (Alice, Bob, and Eve).
   - **Alice and Bob Basis Selection**: Both Alice and Bob randomly chose a basis (computational or Hadamard).
   - **Qubit Preparation**: Alice prepared her qubit in the chosen basis and sent it towards Bob.
2. **Simulating Eavesdropping**:
   - **Eve Intercepts**: Eve intercepted Alice's qubit, potentially measuring it in a basis that could differ from Alice's. This simulated a real-world eavesdropping attempt.
   - Eve then passed the qubit on to Bob, whose basis choice remained independent.
3. **Measurement and Aggregation**:
   - Measured Alice's, Bob's, and Eve's qubits after they passed through each participant's respective basis.
   - Aggregated and stored measurement results across multiple runs to analyse potential interference or mismatches.
4. **Key Sifting**:
   - Alice and Bob communicated over a classical channel to compare their chosen bases and discarded results where their bases did not match.
   - The matching results from their bases provided bits for the preliminary shared key.
5. **Error Detection**:
   - Alice and Bob checked the consistency of their shared bits. Any significant discrepancies indicated possible eavesdropping, prompting them to discard or repeat the process. If consistent, they proceeded to use the key.
6. **Encryption and Secure Communication**:
   - **AES Encryption Setup**: Using the validated shared key, we set up AES encryption in EAX mode.
   - **Encrypting the Message**: Alice encrypted a plaintext message using the shared key, generating a ciphertext along with a nonce and tag (for verifying integrity).
   - **Decryption**: Bob used the ciphertext, nonce, and tag to decrypt the message. A successful decryption confirmed that the key was correct and the message had not been tampered with.
7. **Completion of Secure Message Exchange**:
   - The successful decryption verified both the integrity and confidentiality of the transmitted message, completing the QKD protocol and secure communication process.

This QKD process demonstrates how quantum mechanics can securely distribute encryption keys, enabling encrypted message exchange with a high level of security against eavesdropping.

Understanding BB84 Protocol





What is BB84?

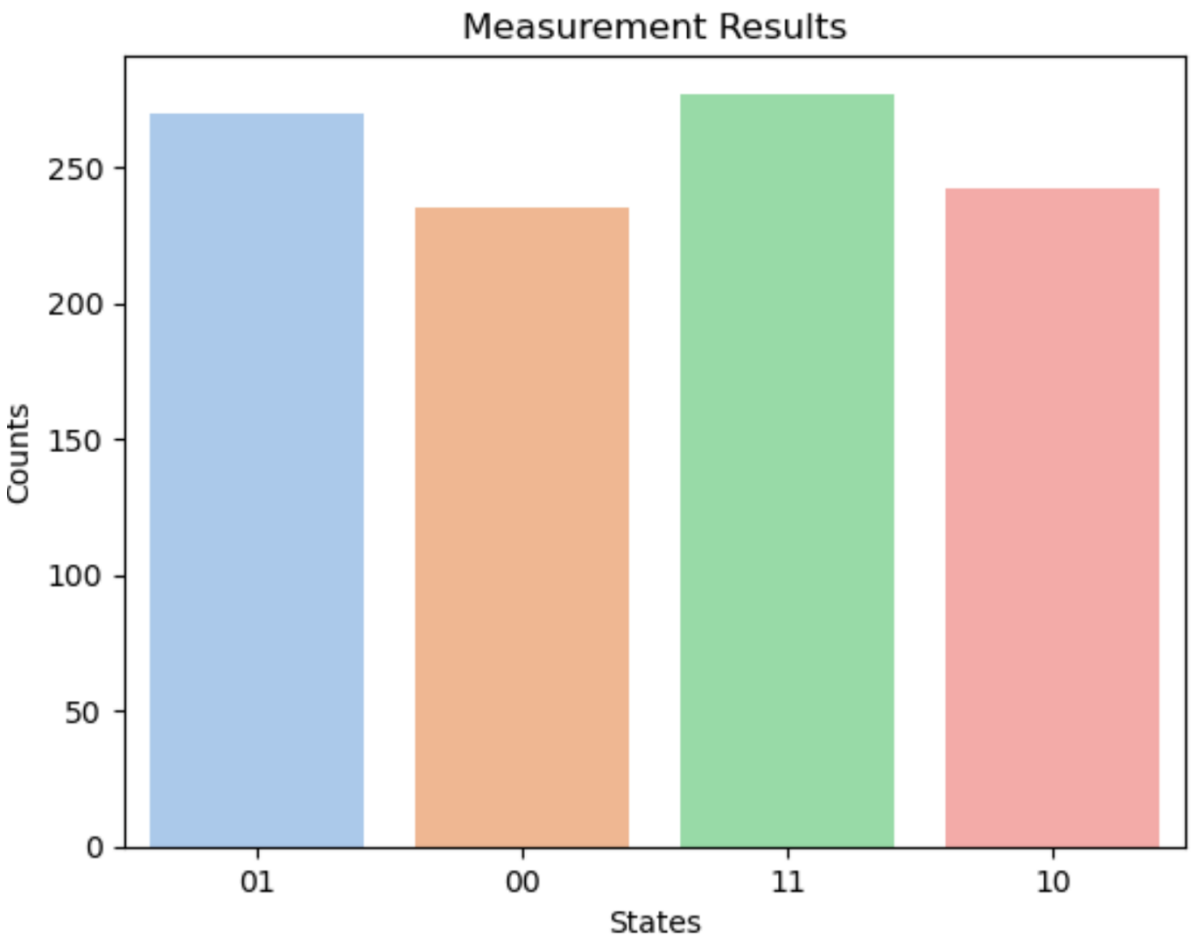
Imagine two people, Alice and Bob, who want to share a secret code that they can use to communicate securely. The BB84 protocol allows them to do this by using the principles of quantum physics to create a key that no one else can intercept without being detected.

How Does It Work?

1. Preparation of Qubits:
   - Alice starts by preparing a series of "quantum bits" (qubits) that can exist in multiple states. She uses two different sets of bases (think of these as different "languages" for sending messages):
   - The standard basis (0 and 1)
   - The diagonal basis (45 degrees and 135 degrees)
   - She randomly chooses one of these bases for each qubit she sends to Bob.
2. Sending Qubits:
   - Alice sends these qubits to Bob one at a time. When Bob receives a qubit, he also randomly chooses a basis to measure the qubit.
3. Measurement:
   - After measuring a qubit, Bob gets a result of either 0 or 1, depending on the qubit's state and the basis he chose. If Bob and Alice used the same basis for a qubit, they can agree on the bit that was transmitted.
4. Key Sifting:
   - After sending and measuring several qubits, Alice and Bob communicate over a public channel to compare their basis choices. They discard any results where they used different bases (since those results are unreliable).
   - The bits they both measured using the same basis form their shared secret key.
5. Error Checking:
   - To ensure their key hasn't been intercepted, they can check a portion of their shared key. If they detect discrepancies, this could indicate that an eavesdropper (let's call her Eve) tried to intercept the qubits, and they should discard that key.
6. Secure Communication:
   - Once they are confident in the integrity of their shared key, Alice and Bob can use it for secure communication, typically by encrypting their messages with the key.

BB84 Protocol Simulation

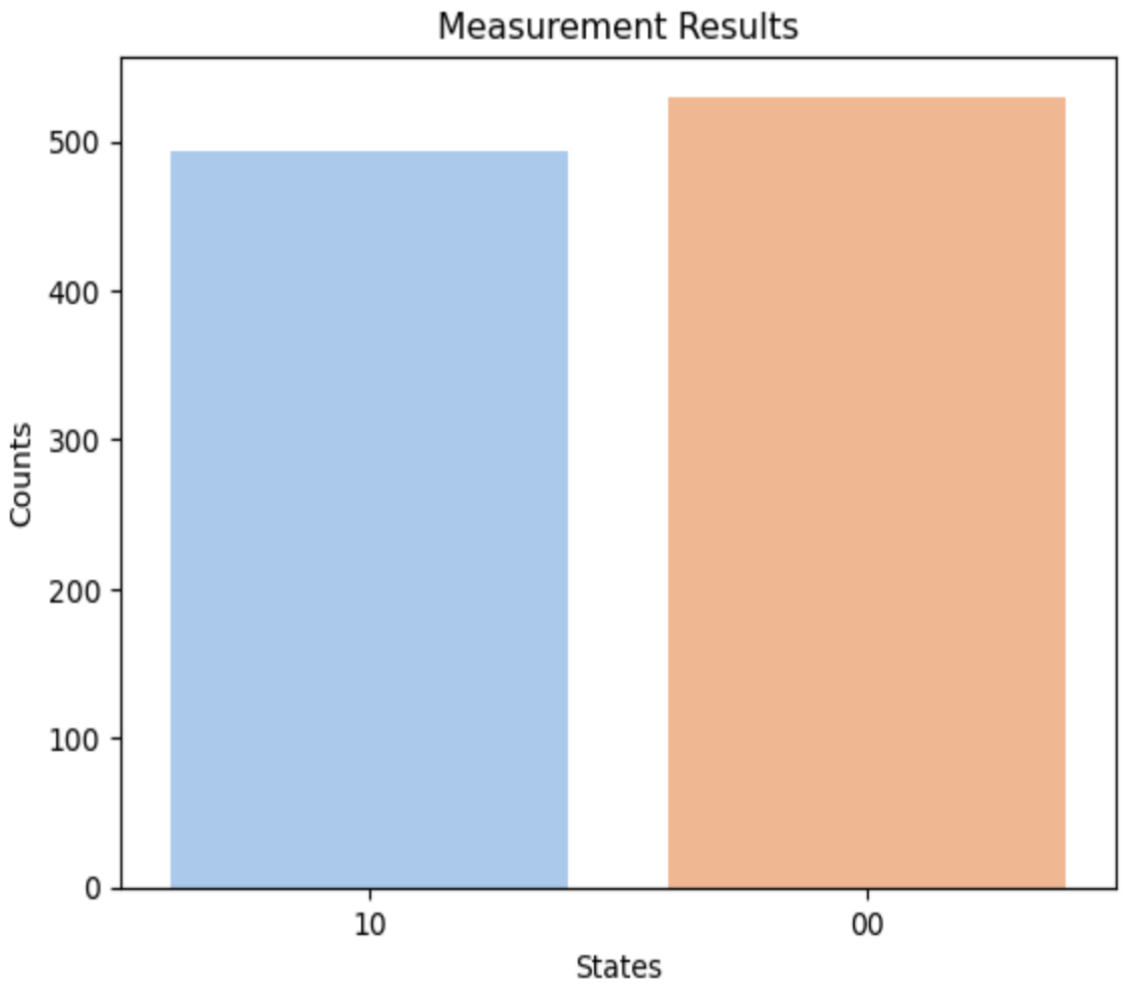
BB84 Protocol Simulation: The project will implement the BB84 protocol using Qiskit. The protocol will be simulated between two communicating parties, traditionally referred to as Alice and Bob, for generating and securely sharing a random encryption key.



Implementing BB84 protocol

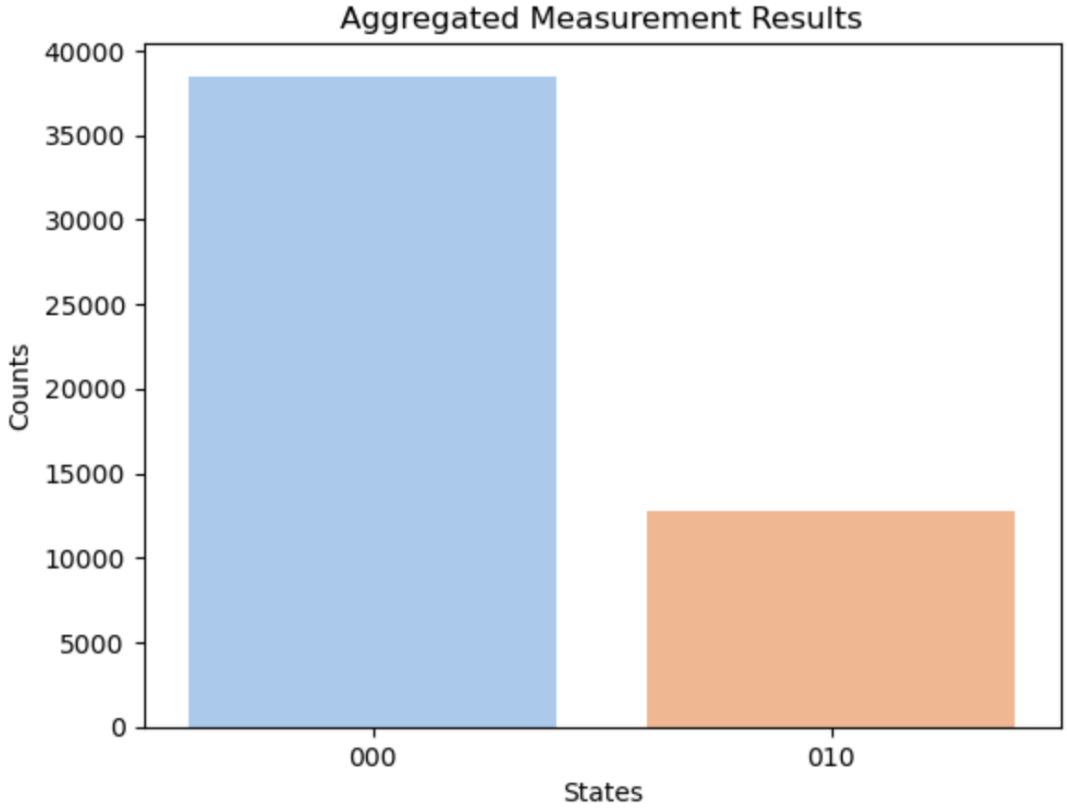
```
Measurement results: {'10': 494, '00': 530}
Alice's basis: Computational
Bob's basis: Hadamard
```

```
/var/folders/zp/_4t4c89s6xgf0z8b9dhl5_dc0000gn/T/ipykernel_7668/2175108261.p

Passing `palette` without assigning `hue` is deprecated and will be removed
e` for the same effect.

  sns.barplot(x=list(labels), y=list(values), palette='pastel')
```



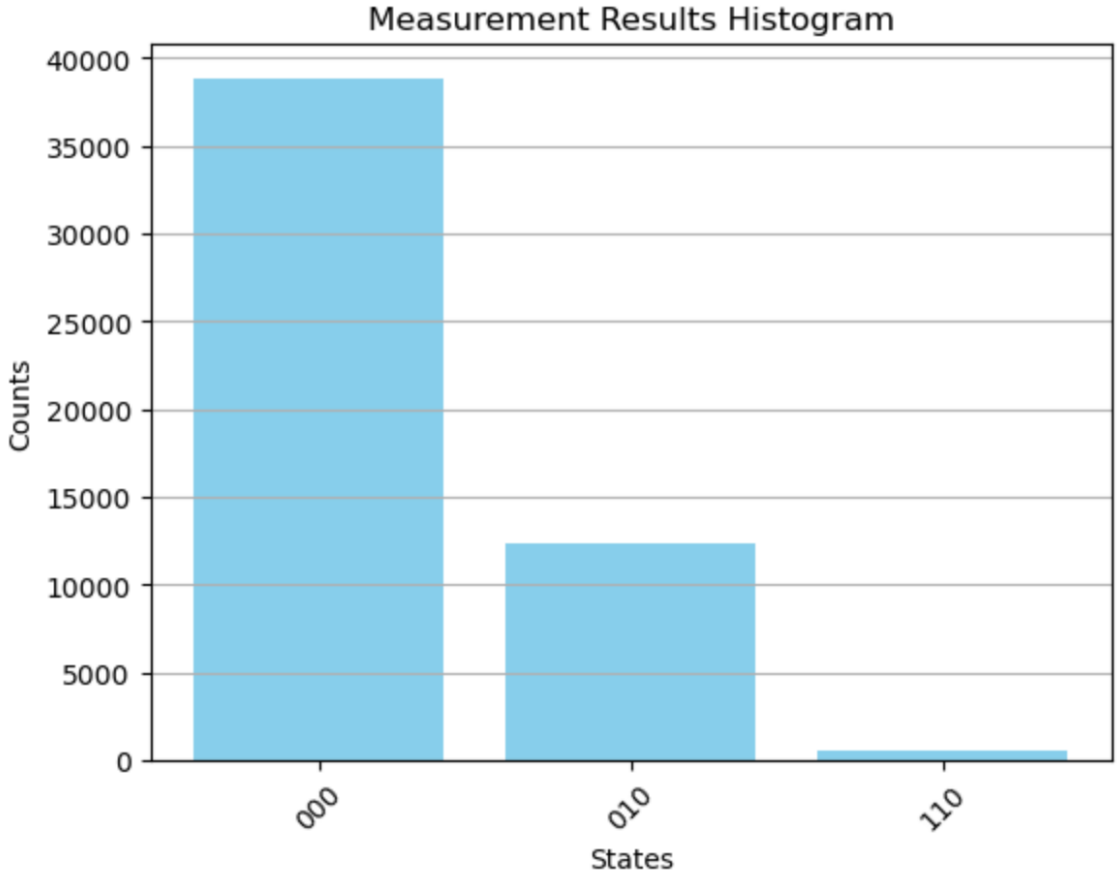
Eavesdropping Simulation

```
Aggregated Measurement results: {'000': 38481, '010': 12719}
Example of runs completed: 50
/var/folders/zp/_4t4c89s6xgf0z8b9dhl5_dc0000gn/T/ipykernel_7668/553216816.py:13: Fu

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
e` for the same effect.

  sns.barplot(data=df, x='States', y='Counts', palette='pastel')
```



Results

```
Total measurements: 51700
Error count: 500
Error rate: 0.009671179883945842
```




```
Ciphertext: b'(\x86\xab\xe3WA\xd6 O\xd8\xe7(\x01\xc2\xd6\xeb\x7fa;f\xb9\xe7\xa7'
Nonce: b'\xa3tBa\x86&\xe17r\x80\x98\xa1-\x0e}9'
Tag: b'\xdd\x1e\xc1\xbd\xc1t\xaeY\x89\x8e\x0b\x95\x03\xde\x95\x81'
Decrypted Message: This is the secret key.
```

Sample Match Ratio: 100.00%
Key consistency verified.

Quantum Key Distribution and Secure Communication Steps

1. **Quantum Circuit Setup for Key Distribution:**
   - **Quantum Circuit Creation:**
   - A quantum circuit is constructed with three qubits: one for Alice (the sender), one for Bob (the receiver), and one for Eve (the eavesdropper).
   - This setup allows for the simulation of secure communication while considering potential interception by an adversary.
   - **Basis Selection by Alice and Bob:**
   - Alice and Bob randomly select a basis for their qubits, choosing either the computational basis (standard measurement) or the Hadamard basis (superposition measurement). This random selection is crucial for ensuring the security of the key exchange.
   - **Qubit Preparation:**
   - Alice prepares her qubit in the chosen basis and sends it to Bob. The state of the qubit is not fixed, reflecting the principles of quantum mechanics.
2. **Simulating Eavesdropping:**
   - **Eve's Interception:**
   - Eve intercepts the qubit sent from Alice. In a real-world scenario, this interception could compromise the security of the key.
   - Eve measures the qubit in a basis that may differ from Alice's, potentially altering its state, which simulates an eavesdropping attempt.
   - **Passing to Bob:**
   - After measurement, Eve sends the qubit to Bob, who independently selects his basis for measurement, ensuring that the attack remains undetected if Eve's choice aligns with Bob's.
3. **Measurement and Aggregation:**
   - **Measurement of Qubits:**
   - All three participants (Alice, Bob, and Eve) measure their qubits after passing through their chosen bases.
   - The measurement results reflect the states of the qubits at the moment of measurement.
   - **Result Aggregation:**
   - The results are aggregated over multiple runs to identify any interference or discrepancies caused by Eve's actions. This data collection is essential for analyzing the effectiveness and security of the key distribution process.
4. **Key Sifting:**
   - **Communication Between Alice and Bob:**
   - Alice and Bob communicate over a classical channel (a traditional communication method) to compare the bases they used for their measurements.

- **Discarding Mismatched Results:**
- They discard any results where their basis selections do not match. The remaining bits from their measurements form the preliminary shared key. This step ensures that only compatible measurements are used in key generation, enhancing security.

5. **Error Detection:**

- **Consistency Check:**
- Alice and Bob check the consistency of their shared bits by comparing a portion of their results. If there are significant discrepancies, this indicates potential eavesdropping, prompting them to discard the key and possibly restart the process.
- **Proceeding with Key Use:**
- If the results are consistent, Alice and Bob agree to use the shared key for encryption, ensuring that they maintain a secure communication channel.

6. **Encryption and Secure Communication:**

- **AES Encryption Setup:**
- With the validated shared key, Alice sets up AES (Advanced Encryption Standard) encryption in EAX mode, which provides both confidentiality and integrity for the transmitted message.
- **Encrypting the Message:**
- Alice encrypts a plaintext message using the shared key, resulting in a ciphertext along with a nonce (a unique number used once) and a tag (used for verifying the integrity of the message).
- **Decryption by Bob:**
- Bob uses the ciphertext, nonce, and tag to decrypt the message. Successful decryption confirms that the key is correct and that the message has not been tampered with during transmission.

7. **Completion of Secure Message Exchange:**

- **Verification of Integrity and Confidentiality:**
- The successful decryption process verifies both the integrity and confidentiality of the transmitted message, completing the QKD protocol.
- **Final Outcome:**
- The project demonstrates how quantum mechanics can be leveraged to create a secure method of communication, highlighting the effectiveness of quantum key distribution in protecting against eavesdropping.

This summary captures the essential steps of the Quantum Key Distribution process, making it understandable for readers unfamiliar with quantum computing.

Executive Summary

This project explores the implementation of Quantum Key Distribution (QKD) and secure communication through a series of systematic steps, leveraging quantum computing principles. The goal is to establish a secure method of exchanging cryptographic keys between two parties, Alice and Bob, while ensuring the integrity of their communication against potential eavesdropping by an intruder, Eve.

1. **Quantum Circuit Setup for Key Distribution:**

A quantum circuit was created with three qubits representing Alice, Bob, and Eve. Alice and Bob independently chose measurement bases for their qubits, preparing them for quantum entanglement.

2. **Simulating Eavesdropping:**

The project simulated an eavesdropping scenario where Eve intercepts Alice's qubit. This step illustrated how an eavesdropper could affect the integrity of the quantum key.

3. **Measurement and Aggregation:**

After the qubits passed through their respective measurement bases, the results were aggregated. This step allowed for the analysis of potential discrepancies in the measurement outcomes caused by Eve's interference.

4. **Key Sifting:**

Alice and Bob communicated over a classical channel to compare their bases and discard mismatched results, thereby sifting through the data to create a shared secret key.

5. **Error Detection:**

The integrity of the shared key was checked for discrepancies. If significant differences were detected, Alice and Bob discarded the key and repeated the process, ensuring only a secure key was retained.

6. **Encryption and Secure Communication:**

Using the validated shared key, the project implemented AES encryption in EAX mode. Alice encrypted a message and sent it to Bob, who successfully decrypted it, verifying both the correctness of the key and the integrity of the message.

7. **Completion of Secure Message Exchange:**

The project concluded with the successful decryption of the message, demonstrating the effectiveness of QKD in achieving secure communication.

This project not only highlights the potential of quantum technologies in enhancing security protocols but also provides a foundational understanding of how quantum mechanics can be applied to cryptography. The methods and results obtained serve as a stepping stone for future research and practical applications in secure communications.