Quantum Key Distribution

QTeam Members:

1) Neel Kanth Kundu
2) Nandan Mishra
3) Gottfried Szing
4) Mike Richardson
5) Robert Wamala

Target Audience: Bachelors Students

Table of Contents:

Prerequisites: Linear Algebra, Basics Quantum Information, Qubits, Measurement of Qubits in different bases, Superposition, Pauli-x,y,z Gates, Hadamard Gates.

Warm-up Quiz:

1. What outcome is observed when Qubit |0> is measured in Z-basis (computational basis)?
2. What are the possible outcomes when Qubit |0> is measured in X-basis (Hadamard Basis)?
3. True or False: "*Arbitrary quantum states can be cloned*"?

Motivation (~5 min.):

In this lecture, we will learn about the quantum key distribution (QKD) protocol. Before delving into the QKD protocol, I will explain the motivation for using QKD in cyber security. **All encrypted, secure communications and stores of data are at risk due to the Quantum Computers' capability to execute Shor's algorithm and crack the most secure encryption algorithm such as RSA (Rivest-Shamir-Adelman).** Public key encryption algorithms like RSA are computationally secure encryption algorithms whose security is based on the assumption that current classical computers cannot solve the problem of large prime factorization in a reasonable time. However, with the rapid advancement in practical quantum computing, the RSA encryption algorithm can be broken by running Shor's factoring algorithm on a quantum computer. Hence, quantum computing poses a huge security threat to our daily online transactions including e-banking, e-commerce, email, sensitive personal data (medical records), email, etc. Furthermore, State secrets once thought secure, become vulnerable and the data in motion e.g. TLS/ HTTPS is vulnerable, leading to a full loss of privacy

However, another quantum technology known as QKD can be used to secure our data in the future. QKD can provide unconditional security using the laws of quantum physics. QKD is used to distribute secret keys between two legitimate users, say Alice and Bob in the

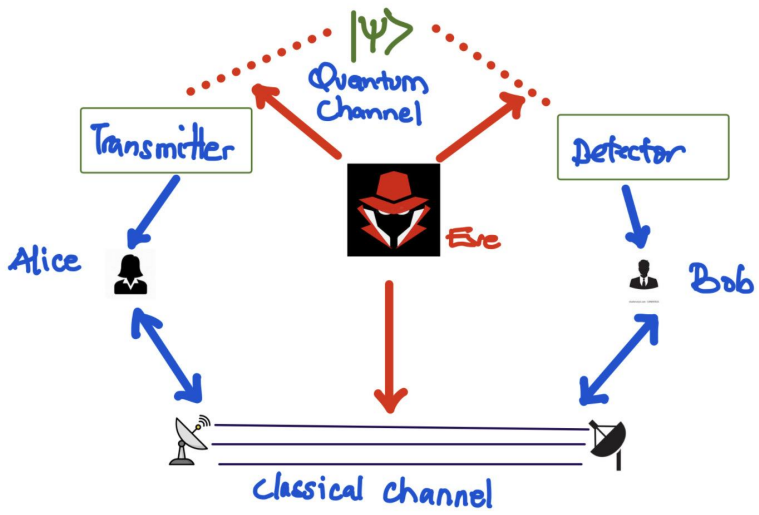
presence of the eavesdropper Eve. **The ultimate aim of QKD is to distribute a random binary key between Alice and Bob which is unknown to Eve. The key string should be identical for Alice and Bob. The key string should be random in the sense that Eve cannot guess this key.**
Interesting Fact: QKD was used at the Berlin Film Festival in 2022 to secure the encryption of media streams.

Theory (~15 min.):

The random key distributed by QKD is used for one-time-pad (OTP) based encryption. The secret data to be transmitted by Alice is converted to 0,1s and then the key string (obtained from QKD protocol) of the same length as that of the data is used for bitwise XOR operation to generate the encrypted data. Now, Bob, who knows the secret key used for encryption, can perform a bitwise XOR operation on the received encrypted data to recover the original message. If the quantum key is truly random and Eve does not know it, Eve cannot decode the data. The classical key distribution algorithm known as Diffie-Hellman is based on the computationally hard problem of the discrete logarithm. The classical key distribution algorithm can also be broken using quantum computers since the discrete logarithm problem can be solved efficiently by a quantum computer. On the other hand, QKD can be used to securely distribute keys. QKD ensures that any eavesdropping during key distribution will be detected by legitimate users. QKD works on the following two principles of quantum physics:

1) No-cloning Theorem
2) Indistinguishability of Non-Orthogonal Quantum States

The first QKD protocol was proposed in 1984 by Bennet and Brassard, popularly known as the BB84 protocol. BB84 QKD encodes the quantum key information in the polarization of the photons (Qubits). The QKD protocol has two phases, in the first phase, quantum communication takes place to share correlated strings. In the next step, classical post-processing takes place to correct for errors and distill quantum secure keys between Alice and Bob. A schematic of the QKD protocol is shown in the figure below. Here, the top link shows a quantum channel used for transmitting quantum states from Alice to Bob and the lower link denotes the classical link used for classical post-processing and error correction.



The main steps of the BB84 QKD protocol:

Step-1: Alice generates a string of random binary keys (0,1,1,0,0,1,....), preferably by using a quantum random number generator (instead of a pseudo-random number generator).

Step-2: Alice randomly chooses a basis for encoding the key. The basis can be either Z-basis (|0>,|1>) with 50% probability or X-basis (|+>, |->) with 50% probability. The quantum states prepared by Alice and transmitted to Bob can be summarized in a table as follows.

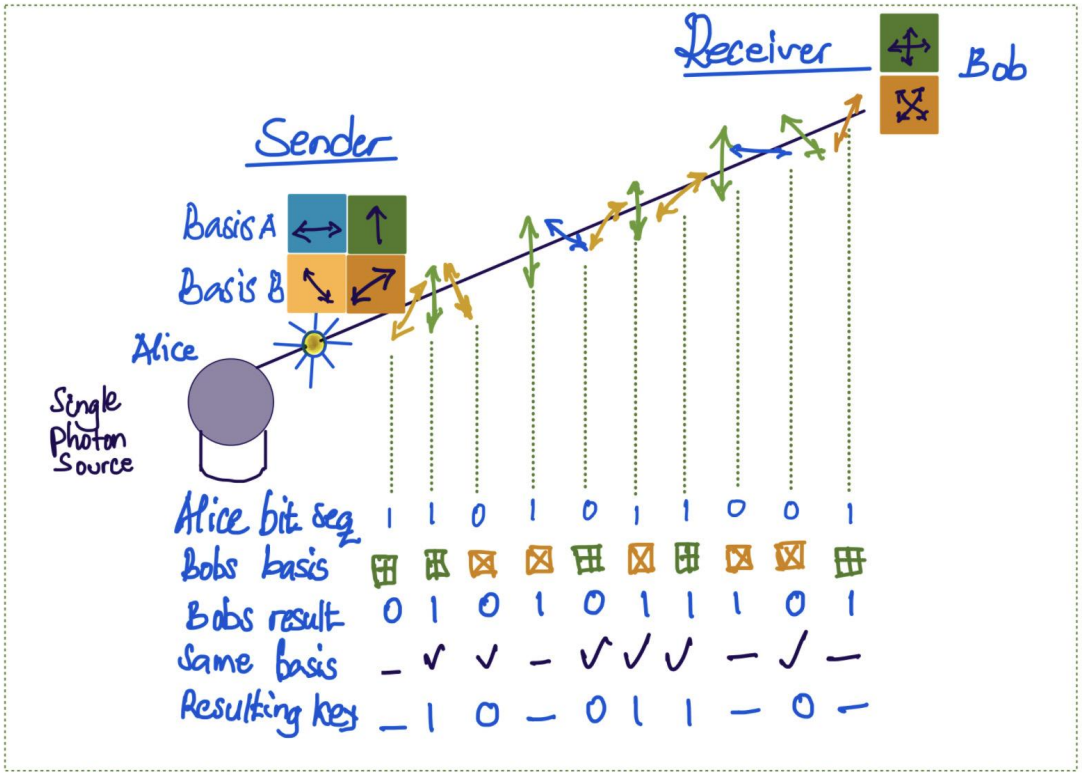
Basis Key Bit →	0	1
Z-Basis	**|0>**	**|1>**
X-Bais	**|+>**	**|->**

Step-3: Bob randomly chooses a measurement basis that is either Z-basis or X-basis with equal probability. Due to this randomness, Bob's measurement basis will match the encoding basis of Alice roughly 50% of the times. If Bob's basis matches Alice's basis, Bob will correctly decode the key bit. If the basis is different then Bob will decode the key correctly only with a 50% probability. This is due to the collapse of the quantum state upon measurement.

Step-4: Now, Alice and Bob declare over the classical channel, the basis used by them for encoding and measuring the quantum states respectively. Alice and Bob retain only those key bits for which the preparation and measurement basis match, and discard the other key bits. This ensures that both Alice and Bob possess identical secret keys (in the absence of an eavesdropper with a noiseless quantum channel).

Step-5: Now, Alice and Bob test the presence of an eavesdropper. This is based on the no-cloning theorem, which states that arbitrary qubits cannot be cloned. Thus, if Eve is listening on the quantum channel, she cannot have a copy of the qubit. Eve can also measure the qubits using a random choice of measurement basis (similar to Bob). This will introduce errors in the shared raw key between Alice and Bob obtained in Step-4. In order to determine the presence of Eve, Alice and Bob compare a subset of the raw keys from Step-4 using the classical channel. If the error rate is above the threshold then the protocol is considered as insecure and the protocol is aborted.

The complete BB84 QKD protocol is summarized in the figure below:



QKD Implementation in Qiskit (~ 15 min.):

- The simulation of the QKD protocol using Qiskit is demonstrated in the Jupyter Notebook: *QKD implementation.ipynb*.
- The students will implement qubit initialization and measurement in different bases to understand the basic quantum physics principles. These concepts are important for understanding the working of the QKD protocol.
- Students will then implement the different QKD Steps explained in the Theory section above in the Jupyter notebook *QKD implementation.ipynb*

QKD Exercise (~15 min.):

Notebook: bonus 1 - transmit_text.ipynb

- Exercise shows how symmetrical encryption is used to secure transmission. The data transmitted is using a simple text message and the keys are generated randomly.

Goal:

- Basic understanding of the algorithm of encryption of data
- Observing the effects of interception of the key on the result
- Learning how to combine the encryption and QKD
- Optionally: replacing the XOR-encryption with a real one

Visualizations (~ 10 min.):

- Students can visualize basic quantum circuits and qubit measurement outcome bar plots in the jupyter notebook *QKD implementation.ipynb*
- Next, students can visualize the encryption of image using OTP based encryption algorithm in the Notebook: bonus 2 - *transmit_image.ipynb*

Learn to transmit more complex data than simple text messages.

Goal:

- Preparing the data for transmission by transforming it into a byte stream
- Encode/decode it with QKD
- Observing the effects of interception of the key on the image

No interception:

With interception:

References:

[1] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” in Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing. Bangalore, India, 1984, pp. 175–179. https://arxiv.org/pdf/2003.06557.pdf
[2] https://qiskit.org/textbook/ch-algorithms/quantum-key-distribution.html