Implementation of Quantum Key Distribution using Python

Prithvik H C[1], Charan K R[2], Sachin B S[3] and Rakesh K R[4]

[1] Department of Telecommunication Engineering, RV College of Engineering,
Mysore Road, Bengaluru-560059, India
prithvikhc.tc16@rvce.edu.in

[2] Department of Telecommunication Engineering, RV College of Engineering,
Mysore Road, Bengaluru-560059, India
charankr.tc16@rvce.edu.in

[3] Department of Telecommunication Engineering, RV College of Engineering,
Mysore Road, Bengaluru-560059, India
sachinbs.tc16@rvce.edu.in

[4] Department of Telecommunication Engineering, RV College of Engineering,
Mysore Road, Bengaluru-560059, India
rakeshkr@rvce.edu.in

Abstract: As the internet grows, the latest communication systems pose a significant challenge for data security and safety. Modern technologies have been tested and implemented to improve the existing vulnerabilities. Though most of the research is done with respect to improve the conventional methods of encryption systems, one technology that completely complements the existing system by its computational model is the Quantum Key Distribution (QKD) which is practically an unbreakable cryptosystem. This paper shows the simulation of QKD, particularly by using the BB84 Protocol. In quantum cryptography, a quantum channel is used to safely share keys and prevent unauthorized parties or eavesdropper from knowing sensitive information. QKD exploits the principles of quantum physics to share the secret key by encoding the information in different quantum states. The basic unit of information in a Quantum Computing system i.e. Qubits (Quantum Bits) are polarized to particular Quantum states depending on the data that is being transmitted and this provides high level security while sending sensitive information. This paper shows the working, basic principles and implementation of the BB84 protocol using the programming tool Python. The Python programming language provides a flexibility to create a quantum environment and show the generation of qubits, its manipulation and also calculate the Quantum Bit Error Rate (QBER) at the end to measure the efficiency of the system.

Keywords: BB84 Protocol, Quantum Cryptography, Quantum Bit, Quantum Bit Error Rate, Quantum Key Distribution, Quantum Mechanics.

1. INTRODUCTION

In today's world, high speed computational devices can easily breach the authentication/verification mechanism and can intercept the messages being transmitted between the sender and receiver. The ever-lasting vulnerabilities that are rampant in the existing systems needed something that is immune to any type of attacks and enable secure communication over long distances. With the advent of Quantum computing systems and the advancements in quantum cryptography helped addressing the short-comings of these vulnerable systems and ensure secure communication. Though Quantum Cryptography was proposed in the early 80's, it was just a theoretical concept waiting for its physical implementation. With extensive research and development in quantum computers in the last decade, QKD has gained importance since it provides the ultimate security for data and has been tested for its public use. Since past few years QKD has been a prime technology in providing a secure communication system.

The presence of an unauthorized party or eavesdropper cannot be inherently detected in classical cryptographic systems which makes the communication vulnerable to hacking. QKD on the other hand, ensures security and integrity of data and can detect any eavesdropper in the communication infrastructure. This paper includes a brief introduction to quantum

cryptography and the principles involved in governing the sub-atomic particles that the QKD takes advantage of in creating a secure communication system that can be seen in the next section. Later, we look into the software implementation and demonstration of BB84 Protocol using Python and also measure the QBER to analyze the error rate of the proposed system.

2. QUANTUM PRINCIPLES AND THEORIES

A. Heisenberg's Uncertainty Principle

The most fundamental justification of Quantum Physics is the Uncertainty Principle of Heisenberg (HUP), which notes that a quantum entity will know with certainty just one property between pairs of the conjugate properties [12, 2, 3]. Heisenberg explained how any possible calculation of the particle's momentum would disrupt its other conjugate property (direction), in relation to the position and momentum of the particle [1]. Therefore, it is impossible to know both the properties with certainty simultaneously. Quantum cryptography follows this principle of its conjugate property and incorporates the polarization of photons using the different basis. Photons are transmitted through optical fiber links and may be the most feasible quantum systems for communication between two parties that want to share valuable information.

B. Quantum Entanglement

Quantum Entanglement is another quantum phenomenon pertinent to QKD. Quantum pairs that appear to be a single entity may be created even if they are separated by large distances and are called as Einstein-Podolsky-Rosan pair or EPR pairs. For example, the 'spin' property of the quanta can have spin 'up' (+1/2) or spin 'down' (-1/2), such that the total spin of the system is zero. But it is uncertain which of the pair it belongs to until a measurement is made. Separation of the EPR pair by a small distance and the measurement of one quanta of the pair results in the collapse of the other wave function in the other state [6]. These immediate actions contradict Einstein's observation that there no such thing in the universe that can travel faster than light. These entangled particles helps in developing a multiple qubit quantum computing system.

C. Quantum No Cloning Theorem

The copies of an unknown quantum state is prevented based on the Quantum No Cloning principle. It is a protection mechanism as quantum mechanics does not allow generation of similar quanta with same properties [1, 2]. So, it is not possible to take back-up copies of quantum states and to use it in the quantum computing routine. This adds to the fact that the eavesdropper cannot create copies of quantum information sent along the quantum channel [7].

3. BB84 PROTOCOL

BB84 protocol is one of the earliest QKD protocols that was proposed which is named after its inventors C. H. Bennett and G. Brassard [5]. BB84 protocol is one of the QKD protocols that can be easily implemented [3] which requires less hardware compared to other protocols [5]. This protocol uses four non-orthogonal polarization states (0º, 90º, 45º,135º) and they are as follows:

- Rectilinear Basis (0º, 90º)
- Diagonal Basis (45º,135º)

The symbols "+" and "X" are used to representing Rectilinear and Diagonal basis

respectively. These symbols "+" & "X" [10] represents the four photon bases and their corresponding binary values used to encode the "quantum bits" or "qubits" [1,7]. The conventions used in representing corresponding bits and bases are listed in Fig. 1 and Table 1.

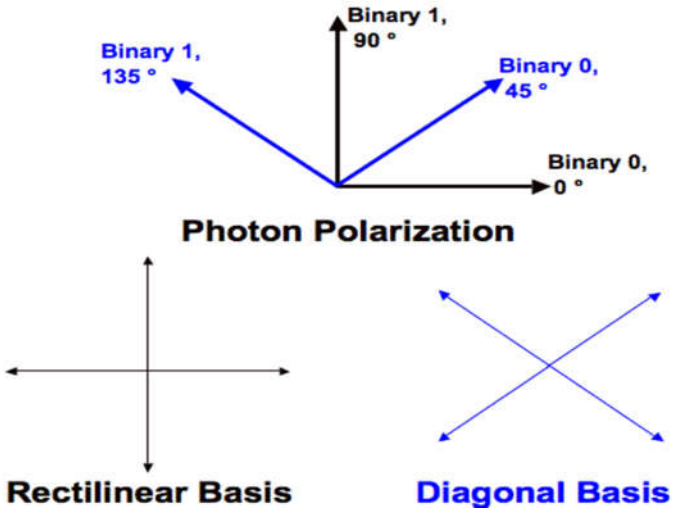

Figure 1. Basis Representation

Table 1: Photon bases and their corresponding values

Polarization State	Photon Base	Symbol	Angle	Qubit
Rectilinear State	+	—	0°	0
		|	90°	1
Diagonal State	X	/	45°	0
		\	135°	1

The transmission of information from the sender 'Alice' to the Receiver 'Bob' using BB84 protocol happens in three phases. They are as follows:

A. Raw Key Exchange

The sender 'Alice' sends the quantum information to the receiver 'Bob'. The receiver measures the quantum states and is the only phase where quantum channel comes into picture. The later stages of the protocol uses the classical channel for verifying the information.

B. Key Sifting

In this phase, Alice and Bob exchange data over the classical channel to decide which measurements to be retained and which to be discarded. The decision-making between the sender and the receiver depends on what protocol and the types of basis that are being used [1].

C. Key Distillation

This is the post key sifting process; this also uses an authenticated public channel which has the ability to repair the losses in information which might be caused due to imperfections in the channel or presence of an eavesdropper. The first two stages in this phase are error correction and privacy enhancement whereas the final step is to authenticate people which hampers man-in-the-middle attacks.

4. METHODOLOGY

QKD using BB84 protocol is implemented using Python packages like QuTip and it makes use of various scientific libraries. The program was developed such that it can mimic the quantum environment and ensure the randomness of the system just as in the real quantum system. The steps for implementing BB84 protocol using Python Programming language is as follows:

Step 1: The polarization of photons is done using the conjugate basis, a rectilinear (0° and 90° polarizations) or a diagonal [1, 3] (45^o and 135^o polarizations) basis is used.

Step 2: The polarization of photons enables embedding of information corresponding to binary values i.e. '0' or '1' depending on the type of basis used (one photon/qubit corresponds to one qubit of information), and before there is any exchange of quantum states, the basis are pre-determined and agreed upon before-hand.

Step 3: One of the two states of polarization basis is chosen at random and each photon/qubit is polarized according to the chosen basis. Every photon is polarized using the randomly chosen basis and a stream of photons are sent from 'Alice' to 'Bob' to make the measurements by the receiver.

Step 4: 'Bob' has to detect the quantum states using one of the two polarization basis, in this case 'Bob' does not know which basis were chosen by 'Alice', he randomly guesses one of the basis for detection and passed through a filter and photon counter to note the results. If chosen correctly, the polarization is recorded correctly or else he might make measurements with randomly chosen basis and all the information is lost for that photon.

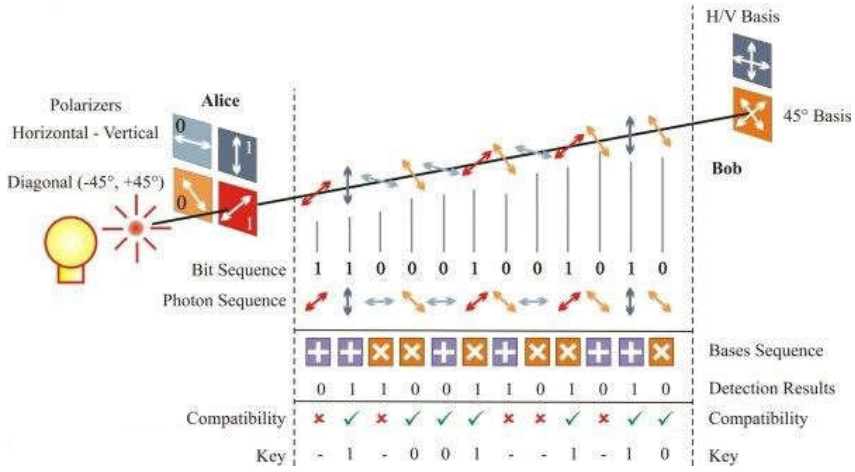

Figure 2. Working demonstration of BB84 protocol with an example [13]

Step 5: Both 'Alice' and 'Bob' broadcast the basis that were used by each other to measure the polarization of photons over a public channel. Since, only the details about the basis are being exchanged publicly, no information regarding the actual key can be obtained by anyone eavesdropping at this stage. After comparing the basis, the photons that were detected using the same basis as used by the sender to polarize is retained and the rest is discarded. Since there are only two possibilities here, there is a 50% probability for a photon to be retained and hence there is roughly 50% reduction in the key size compared to the raw key size. The example of the same is depicted in Fig. 2 whose final key after sifting is 100110.

5. SIMULATION RESULTS

Python programming language was used to implement the BB84 Protocol since it provides the independency of choosing the required modules for the development of the code and simulation of the protocol. Pool, random and some system packages were used as support packages to implement the code. The protocol was implemented in the following way.

Consider an example of a 16-bit key which and used for assigning with basis randomly and it exchanged with the receiver. After decoding and verifying the basis, the 8-bit key is left and it is used as the key.

```
Step 1: Alice prepares a random string of bits and encodes them randomly in either the X or Z bases
Alice's bits and bases:
[1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1]
['Z', 'X', 'Z', 'Z', 'Z', 'Z', 'Z', 'X', 'X', 'Z', 'X', 'Z', 'Z', 'X', 'Z', 'X']
Press enter to proceed to Step 2...

Step 2: Alice sends each qubit to Bob (intercepted and then resent by Eve)
Step 3: Bob randomly measures each qubit in either X or Z bases and records his results
Eve's bits and bases:
[1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1]
['Z', 'Z', 'Z', 'X', 'X', 'Z', 'X', 'X', 'Z', 'X', 'X', 'Z', 'X', 'Z', 'Z', 'X']

Bob's bits and bases:
[0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1]
['X', 'X', 'Z', 'Z', 'Z', 'X', 'Z', 'X', 'Z', 'X', 'X', 'X', 'X', 'Z', 'Z', 'X', 'X']
```

Figure 3. Sending bits using Randomly chosen Basis

As shown in the Fig. 3 ‘Alice’ chooses random basis to polarize each bit of the raw key and sent to the receiver, Bob.

‘Bob’ randomly choses one of the two basis of polarization states to detect the photon. As shown in Fig. 4, they discuss the basis that were used for polarization through authenticated classical channel and announce which ones were detected using the correct base.

```
Step 4: Bob publicly tells Alice what basis he measured each qubit in
Step 5: Alice tells Bob for which qubits he chose the correct basis
Indices of bits/bases in which Alice and Bob have the same basis:
[1, 2, 3, 5, 8, 10, 12, 15]
Press enter to proceed to Step 6...

Step 6: Alice and Bob delete all of their corresponding qubits for which the bases disagree (and Eve tries to)
The Key is 77
Alice's sifted key and bases:
[0, 1, 0, 0, 1, 1, 0, 1]
['X', 'Z', 'Z', 'Z', 'X', 'X', 'Z', 'X']

Bob's sifted key and bases:
[0, 1, 1, 0, 0, 1, 1, 1]
['X', 'Z', 'Z', 'Z', 'X', 'X', 'Z', 'X']

Eve's sifted key and bases:
[1, 1, 0, 0, 0, 1, 1, 1]
['Z', 'Z', 'X', 'Z', 'Z', 'X', 'X', 'X']
```

Figure 4. Received bits after key sifting

Bob who received the information from Alice discards the bits that were detected using improper basis and performs the key shifting as shown in Fig. 4 Here the final key is 77.

```
Length of sifted raw key:  8
Percentage of reduction:  50.0 %
Press enter to proceed to Step 7...

Step 7: Alice and Bob agree on a small subset of the sifted raw key to publicly reveal in order to calculate the quantum bit error rate.
The QBER is: 40.0%
The secure key rate is: 97.09505944546686%
```

Figure 5. Measurement of QBER

The Quantum Bit Error Rate (QBER) [3, 2, 8] which is the measure of the presence of eavesdropper is the ratio of error rate to the key rate is calculated as in Fig. 5 If QBER surpasses certain threshold, the key is discarded and a new communication channel is established because

there might be some sort disturbance in the quantum channel which might be due to the presence of eavesdropper.

6. CONCLUSION

The key generated and exchanged using all the above steps will be used for further encryption processes. The key deduction by data sniffing becomes a more difficult task because in Quantum Key Distribution none of the mathematical algorithms are used to produce a key that is being used for encryption. Even if eavesdropper tries to intercept during key generation process the users can detect the presence of an eavesdropper since the number bits reaching the other end will be less. This is due to inherent property of Qubits, that is they get vanished once intercepted and cannot be regenerated. Since randomness achieved is more in case of QKD, the key is discarded when the deduced efficiency is found to be less and new process is started again, which could not be guessed by eavesdropper. Quantum computing has changed the way the conventional systems function using classical bits. Quantum systems use qubits which are very much different in nature as compared to the classical bits. This paper demonstrated the polarization of qubits using the basis as described the by the BB84 protocol using python programming language and the QBER calculation helped in analysing the efficacy and the robustness of the proposed system. But in real-time scenarios other protocols proposed in [6] can also be used that are more modified and efficient protocols.

REFERENCES

[1] Archana B and Kritika S, “Implementation of BB84 Quantum Key Distribution using OptiSim”. In the Proceedings of IEEE Sponsored 2nd International Conference on Electronics And Communication System, pp.457-460, 2015.

[2] Nicolas Gisin, Gregoire Ribordy, Wolfgang Tittel and Hugo Zbinden, “Quantum Cryptography”. Reviews of Modern Physics, Vol. 74, Issue 01, pp. 145-195. doi: 10.1103/RevModPhys.74.145, 2002

[3] Hui-fang LI, Li-Xin ZHU, Kai Wang, Kai-Bin Wang, “The Improvement of QKD Scheme Based on BB84 Protocol”. In the Proceedings of the International Conference on Information System and Artificial Intelligence, pp. 314-317, 2016.

[4] Petros Wallden, Vedran Dunjko, Adrian Kent and Erika Andersson, “Quantum digital signatures with quantum-key-distribution components”. Physical Review A, Vol. 91, Issue 04, doi: 10.1103/PhyRevA.91.042304, 2015

[5] Charles H. Bennett and Gilles Brassard, “Quantum Cryptography: Public key Distribution and Coin Tossing”. In the proceedings of the IEEE International Conference on Computers, Systems & Signal Processing, Bangalore, India, pp. 175-179, December 1984

[6] Sheila Cobourne, “Quantum Key Distribution Protocols and Application”. Technical Report

[7] Patryk Winiarczyk and Wojciech Zabierowski. “BB84 analysis of operation and practical considerations and implementations of quantum key distribution systems”. In the Proceedings of the 11th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics, pp. 23-26, 2011.

[8] Guoqing Cui, Yuan Lu and Guihua Zeng, “A new scheme for Quantum Key Distribution in Free-space”. In the Proceedings of the 15th Asia-Pacific Conference on Communications, pp. 637-640, 2009.

[9] Sandeep V and Niranjan A, “Implementation of a Modified BB84 Algorithm for Secure Key Exchange in a Normal Network”. International Journal of Engineering Research & Technology, Vol. 02, Issue 14, pp. 48-50, 2014.

[10] Rahul Aggarwal, Heeren Sharma, Deepak Gupta, "Analysis of Various Attacks over BB84 Quantum Key Distribution Protocol". International Journals of Computer Applications, Vol.20, Issue 08, pp. 28-31, 2011.

[11] Mario Berta, Matthias Christand, Roger Colbeck, Joseph M. Renes, Renato Renner. "The uncertainty principle in the presence of quantum memory". Nature Physics, Vol. 06, Issue 09, pp. 659-662. doi:10.1038/nphys1734, 2010.

[12] C. Erven, C. Couteau, R. Laflammeand G. Weihs. "Entangled quantum key distribution over two free-space optical links". Optics Express, Vol. 16, Issue 21, pp. 16840-16853, doi:10.1364/oe.16.016840, 2008.