Novel Implementation of Quantum Key Distribution protocol: BB84 on FPGA

***Abstract*—This paper presents the implementation of Quantum Key Distribution (QKD) Protocol:BB84 on FPGA. Quantum Communication Methodology has been designed using cryptographic protocol along with parallel processing power of FPGAs. The multiple module hierarchy has been used to ease the realization of Algorithm. The main module characteristics and sub module functionalities have been discussed in detail. The first sub-module works on the basis selection and qubits generation at the transmitter end. The next submodule deals with basis selection and comparison of received qubits at the receiver end. In between two modules, a spy module has been introduced that shall help us in analysis of the different possible scenarios in actual quantum transmissions. The third submodule gives us the estimate of deviation in received message at the receiver and decides whether the transmission channel has been hijacked or not. The major advantage of using FPGAs is the high processing speed additionally Verilog helps us in easy development of the algorithm.**

Keywords—Quantum, FPGA, BB84, Cryptographic, Transmission, Verilog.

I. INTRODUCTION

Over the past two decades, there have been significant advancements in communication technologies. With advancing technologies there arises a need to come up with better encryption technique to ensure that there is no leak of data to the spies present in the classical channel of communication.

The problem has been solved to a great extent by the quantum cryptography techniques. The Quantum Key Distribution Protocols generates a safe key that is shared between the transmitter and the receiver ends through the quantum channel [1]. It not only helps in preventing Eavesdropping but also enhances the level of security in the communication channel.

In 1984, Bennett-Brassard came up with BB84 protocol that is the most acceptable protocol for long distance communications till date [2]. In this paper we present the implementation of this BB84 protocol on FPGA using Verilog. With the availability of Altera Cyclone IV FPGA, the hardware implementation of BB84 Protocol was done.

FPGAs are most versatile devices as far as Logic Implementation is concerned. Hence, our project holds some significance in research of Quantum Cryptography.

II. BB84 PROTOCOL

A. The Communication Channels and Qubits

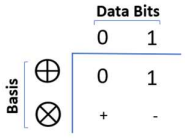
There are two communication channels used in BB84 communication Protocol, namely, Quantum Channel and the Classical Communication Channel. The data transfer is in the form of Qubits.

These qubits are the 4 quantum states. These Qubits are nothing but polarized Photons. The quantum state of the photons depends upon the Basis Chosen and the Bit Steam to be transferred from the transmitter to the receiver [3].

The chosen basis can be:

- The rectangular base $|\rightarrow\rangle$, $|\uparrow\rangle$
- The diagonal base $|\nwarrow\rangle$, $|\nearrow\rangle$ [4].

The input stream gives us the logic level to be chosen. Given table shows how the quantum state is chosen depending upon the basis chosen and input:



⊕ Rectangular Base
⊗ Diagonal Base

B. Data Transmission

The two transmission stations are named as Alice and Bob. Alice. Both quantum channel and the classical channel are used in this process. Alice sends the encrypted data stream through the quantum channel; it is encrypted as per the BB84 protocol.

Alice takes two Bit Streams A and B. A bit stream is the data to be transmitted to Bob. The bit stream B helps in choosing the basis for the polarisation of the photon (Diagonal basis or rectangular basis).

The scheme chosen has been depicted in the table below:

A	0	1	1	0	1
B	1	1	0	0	1
Basis	⊗	⊗	⊕	⊕	⊗
Qubits(tx)	+	-	1	0	-

Bob receives the transmitted qubits and decrypts this data using a randomly chosen basis at his own end.

C. Comparison of the received and transmitted signal over the classical channel and Error Estimation

Once the message is decrypted at Bob's end, both Alice and Bob compare the two message signals. This comparison is done via communication over the classical channel.

There are only two possible cases that Bob's randomly chosen base matches with the Alice's chosen base or it is just opposite. The above chosen scheme has been carried forward at Bob's end below:

Qubits(rx)	+	-	1	0	-
B'	1	1	1	0	0
Basis	⊗	⊗	⊗	⊕	⊕

A’	0	1	0/1	0	0/1

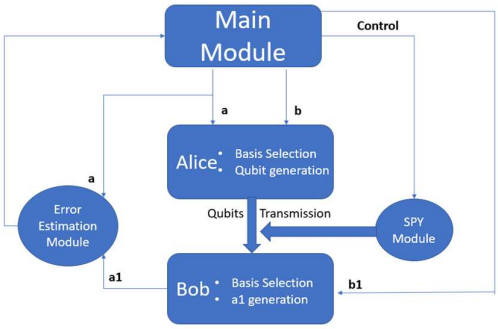
Suppose Bob has chosen the basis as:
B’ = 0 for rectangular base & B’ = 1 for the diagonal base

Once compared, the constant choices of bits are kept and taken as the key for encryption. Alice and Bob check the percentage of the mismatch, i.e., the error in the received signal obtained at Bob’s end.

The mismatch occurs as per the No-Cloning Theorem, the exact state of the qubit can never be sent as copied by the eavesdropper. In case the mismatch is high, it is considered that there has been some eavesdropping in the communication channel and the data is reshared.

III. Algorithm Followed

Parallel processing of FPGAs is the key advantage of using to them to implement this protocol. Every functionality has been assigned different module for processing. The details for the same have been given below:



Flowchart to depict the Algorithm followed

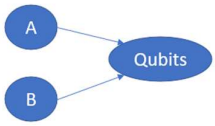
A. Main Module

Main Module is the mother module, all the input/output ports have been attached here. Alice’s Bit Streams (Data to be transmitted), Bob’s Bit Stream (Basis Chosen), Spy (Control Input) and Qubits data are all present in this main module.

B. Alice

Alice module implements the functionality of assigning the polarization state to the photons *(for data to be transmitted bit stream A)* as per the chosen basis *(bit stream B)*.

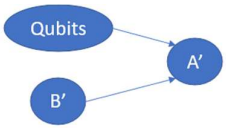
Each Qubit state has been defined as a parameter in the program, once received by the block, a photon polarisation state is assigned and Qubits are generated likewise.



C. Bob

Here, in the Bob Module, the Qubits are received which were transmitted by the Alice block. As per the parameter assigned to the Input stream, a state is now assigned in the state machine wherein a fixed output is pushed to A’. In case of undefined state, ‘Don’t Care’, 1’bX is pushed to the output A’

This way we get the data stream extracted from the received qubits sequence.



Spy control input has been taken in this module itself, when control signal is 1 (presence of a spy), the Qubits are given a state different from the one that was being originally transferred. This has been implemented using an inverter that inverts the state code assigned to a particular qubit.

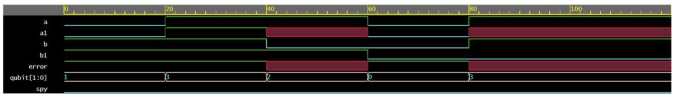
D. Error Estimation

The mismatch in the input bit stream A and output stream A’ are compared using an Exclusive-Nor Gate. Whenever there is a mismatch, it outputs either a zero or 1’bX. The percentage error is calculated and its concluded whether there has been any spy in the channel or not.

A	0	1	1	0	1
A’	0	1	0/1	0	0/1

IV. Simulations

The waveforms obtained are attached below:



V. Results and Discussions

Altera’s Cyclone IV FPGA was used for hardware implementation, at the software part Quartus Prime Lite from Intel was used for this project. The waveforms obtained are in accordance with the stated protocol.

VI. Conclusion

In this paper, we have successfully implemented BB84 protocol on FPGA. The Quantum Cryptographic Protocols are a great asset for secure encryption and safe transmission of data.

Acknowledgment

This work was supported by the Department of Electronics and Communications Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi.

We thank our guide for this project, Dr. Md. Waseem Akram, Assistant Professor at the department, who provided us with all the required resources.

References

[1] Alharith A., Abdullah and Noor R., “Efficient Implementation for Prince Algorithm in FPGA based on BB84 Protocol,” 2021 Journal of Physics: Conference Series, 2021

[2] C. Bennett and G. Brassard: in Proceedings of the IEEE ICCSSP, 1984, p. 175