NAME - KAUSHAL KISHOR GAGAN

ROLL NO - 180010014

Project Report

CS – 218

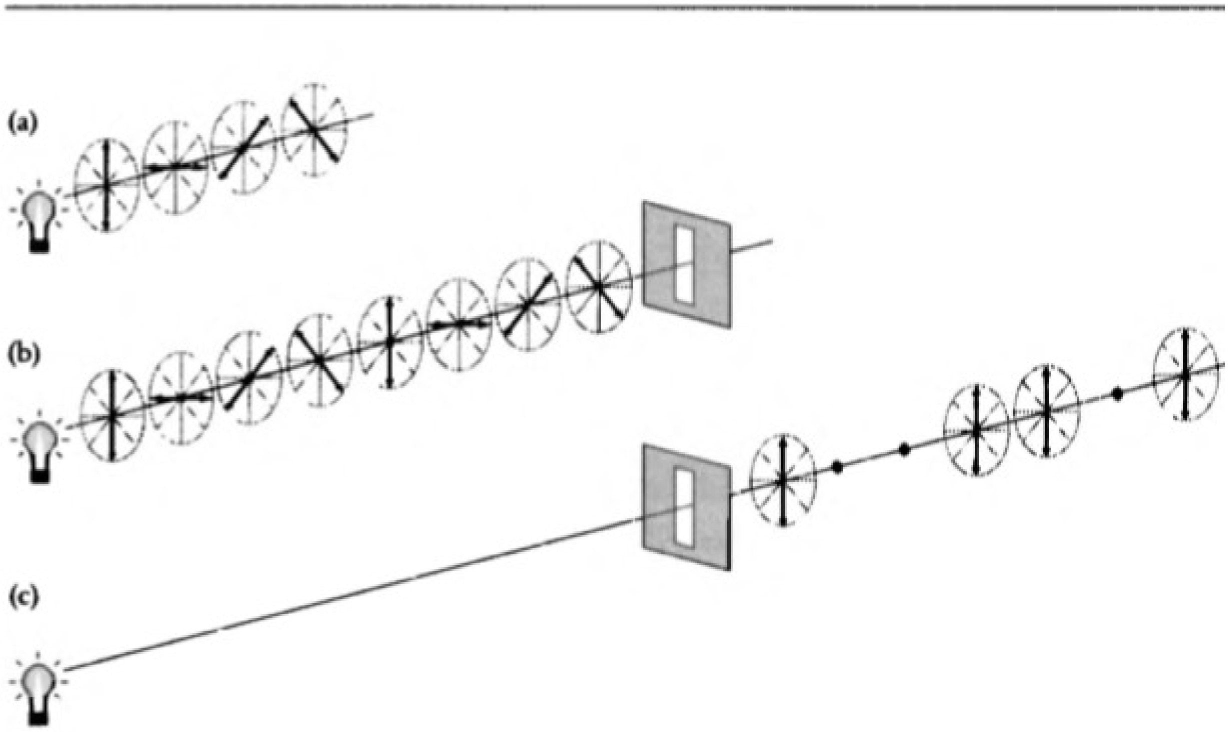
ENCRYPTION AND HASHING

1. INTRODUCTION:

Alice and Bob wants to communicate using a secure line but Eve is the third party who is interested and a spying on them so they have decided to make a secure line using QUANTUM ENCRYPTION based on QUANTUM SUPERPOSITION but they don't have lab equipments and resources for implementing this encryption. But they have PYTHON3, So they have decided to replicate the implementation and trying to gain same level of security using python.

1.1 THEORY AND IMPLEMENTATION:

Work of a Polariser- **In this fig. We can see (a) is a non polarised light In (b) we put a polariser plate in (c) we have polarised light.**



This result is product of the fact that energy of light photons are quantized. So whenever a single linearly polarised photon with phase α passes through polariser at angle θ then wave function of photon can be written as

$$|\psi\rangle = \cos\theta \exp(i\alpha)|x\rangle + \sin\theta \exp(i\alpha)|y\rangle = \psi_x|x\rangle + \psi_y|y\rangle.$$

where basis x and y are

$$|x\rangle \stackrel{\text{def}}{=} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and

$$|y\rangle \stackrel{\text{def}}{=} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

let A = Ψ_x and B = Ψ^y then,

$$|\Psi> = A|x> + B|y>$$

where |A|^2 is probability that wave function will collapse about x-axis similarly |B|^2 is probability that wave function will collapse about y-axis. For simplicity assume α=0 and θ=45° then

probability that it will pass vertical polariser = 1/2

if α=0 and θ=45° then

probability that it will pass vertical polariser = 1

This is the main basis of security in this Encryption method.

Now this is how Alice and Bob will communicate using this Encryption:

1.Alice have two options for encoding 1 and two options for encoding 0. she can use 0°(↕) or 45°(/) polarised photon(angle with respect to vertical) for 1 and for 0 she can use 90°(↔) or -45°(\) polarised photon.

2.After encoding she will send his photons to bob. Bob has 2 types of decoding polariser + type and x type. 0°(↕) and 90°(↔) polarised photons can pass + type polariser with probability 1 but 45°(/) and -45°(\) polarised photons can pass with probability1/2 as we have seen above. Similarly 45°(/) and -45°(\) polarised photons can pass x type polariser with probability 1 but 0°(↕) and 90°(↔) polarised photons can pass with probability1/2 only.

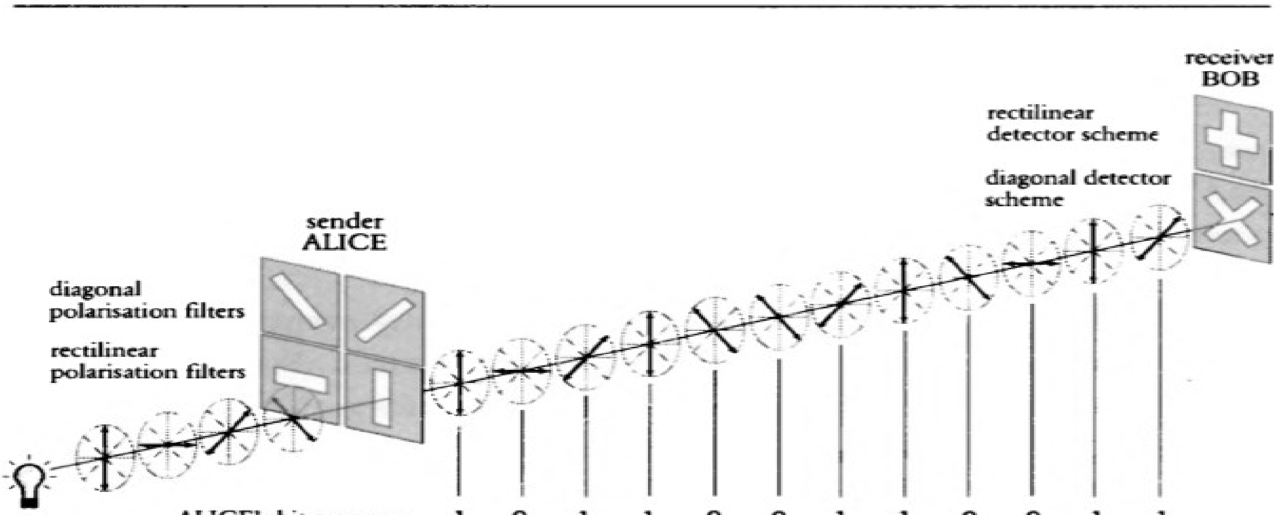
3.Now bob will randomly choose + and x type polariser for measuring state of each photon.

4.After measurement bob will send his sequence of + and x type plates that he chooses randomly through an unsecure line. Where eve can also listen this conversation.

5.Now Alice will tell bob about which choice was correct and which bit bob received corrupt.

6. finally Alice and bob both knew about which bit was corrupted so both will put 0 on corrupted bit and now both will agree upon a key.

This fig. Given below also Demonstrate the communication procedure.

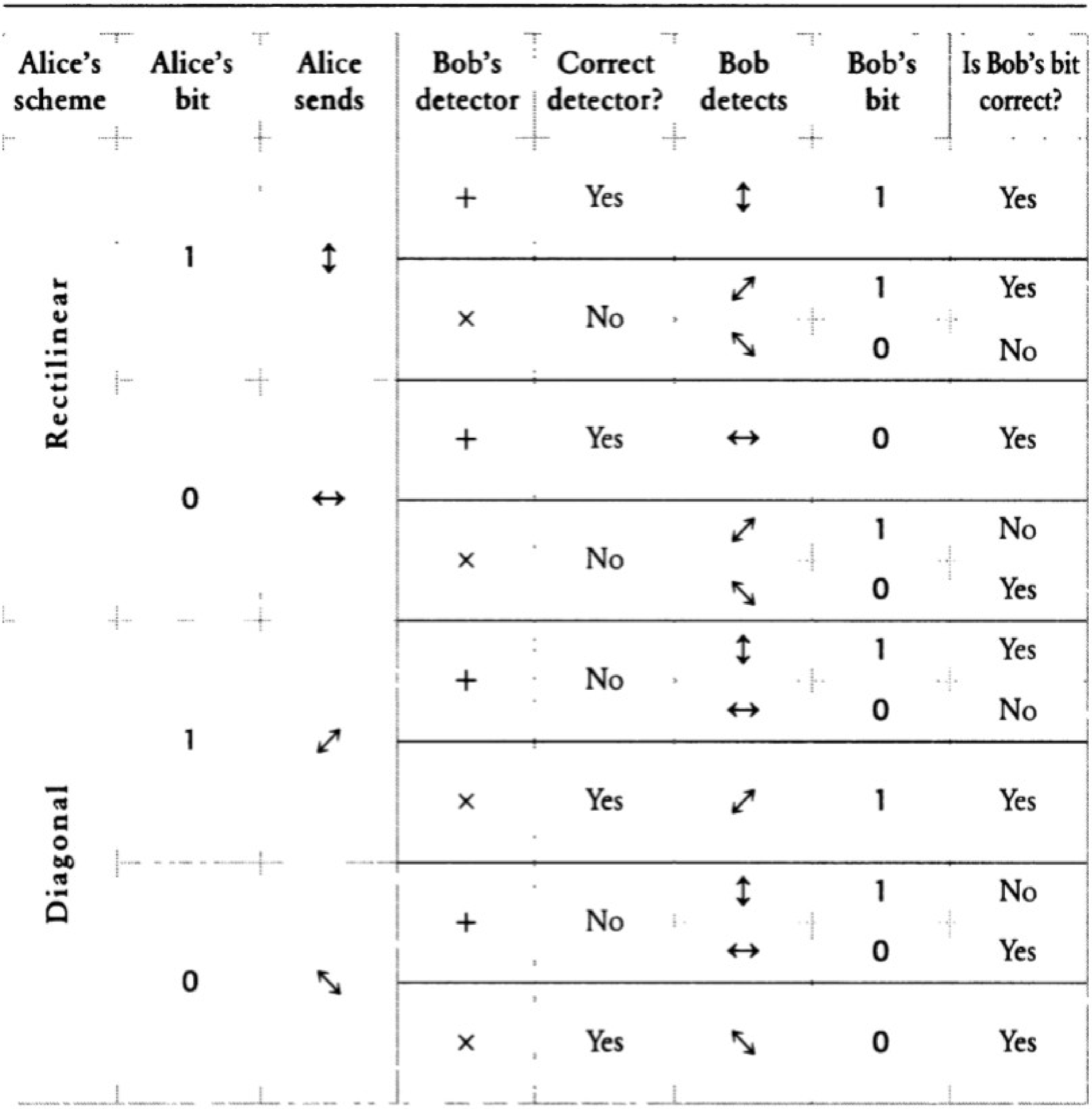

receiver
BOB
rectilinear detector scheme
diagonal detector scheme
sender
ALICE
diagonal polarisation filters
rectilinear polarisation filters

ALICE's bit sequence	1	0	1	1	0	0	1	1	0	0	1	1
BOB's detection scheme	⊞	⊠	⊞	⊞	⊠	⊠	⊞	⊞	⊠	⊞	⊠	⊠
BOB's measurements	1	0	0	1	0	0	1	1	0	0	0	1
Retained bit sequence	1	–	–	1	0	0	–	-1	0	0	–	1

1.2 PROBLEM WITH THIS ENCRYPTION:

This encryption is based upon Quantum Superposition property of Quantum particles. Quantum particles remains in superposition state until environment interacts with them or until we observe them. After observation state of Quantum particles collapses into one of its possible final state. So maintaining states of quantum particles is a problem of this Encryption method.

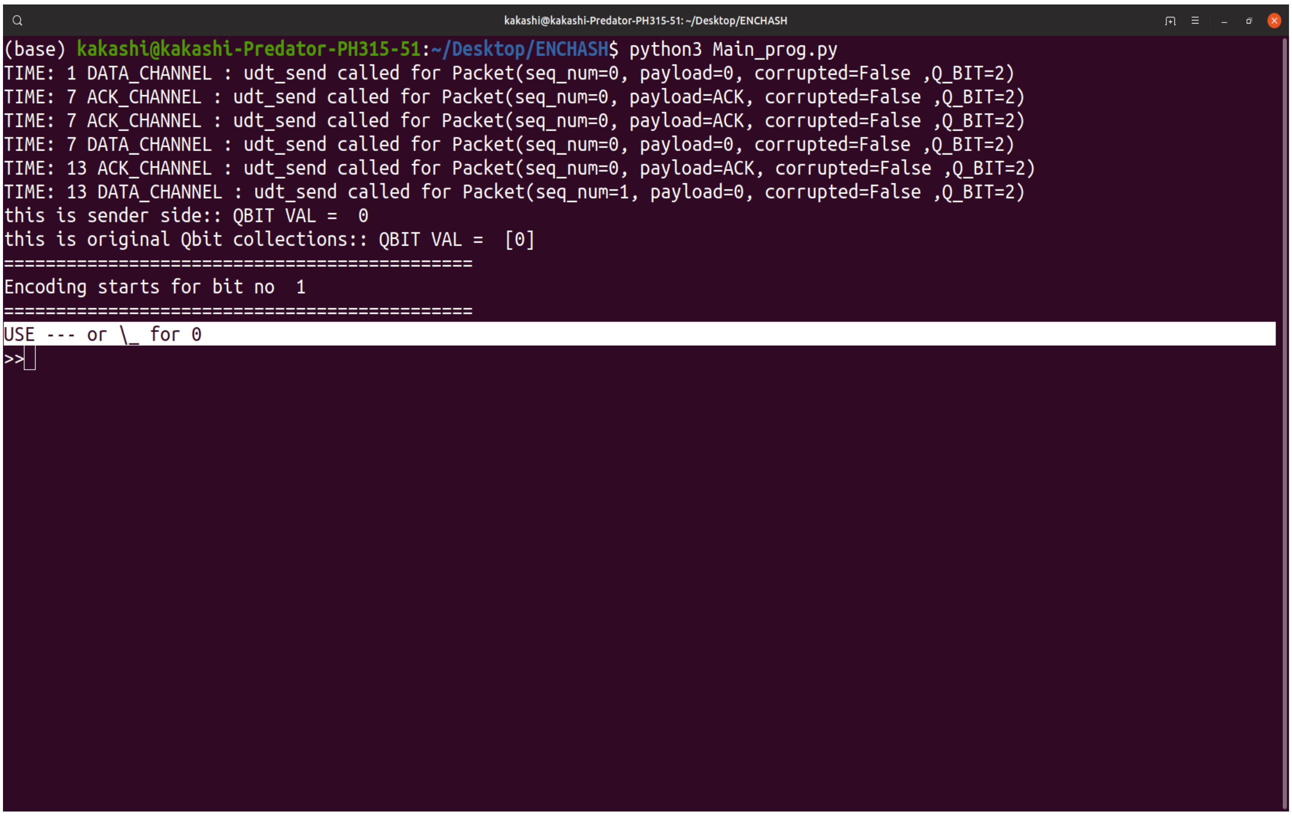
1.3 RESULT TABLE FOR FOUR BIT'S:-



Alice's scheme	Alice's bit	Alice sends	Bob's detector	Correct detector?	Bob detects	Bob's bit	Is Bob's bit correct?
Rectilinear	1	↕	+	Yes	↕	1	Yes
			×	No	⤢ ⤡	1 0	Yes No
	0	↔	+	Yes	↔	0	Yes
			×	No	⤢ ⤡	1 0	No Yes
Diagonal	1	⤢	+	No	↕ ↔	1 0	Yes No
			×	Yes	⤢	1	Yes
	0	⤡	+	No	↕ ↔	1 0	No Yes
			×	Yes	⤡	0	Yes

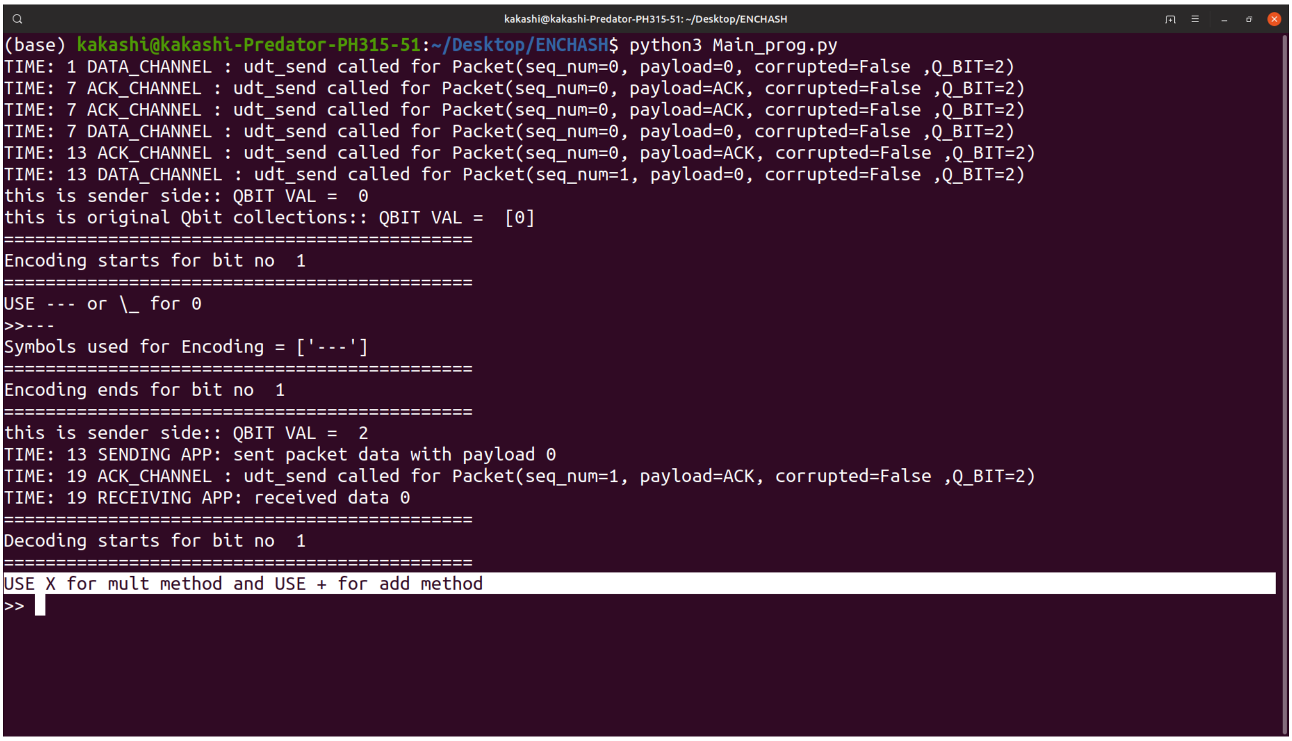
2.PROGRAM RESULTS FOR FOUR BIT’S :-

2.1 ENCODING



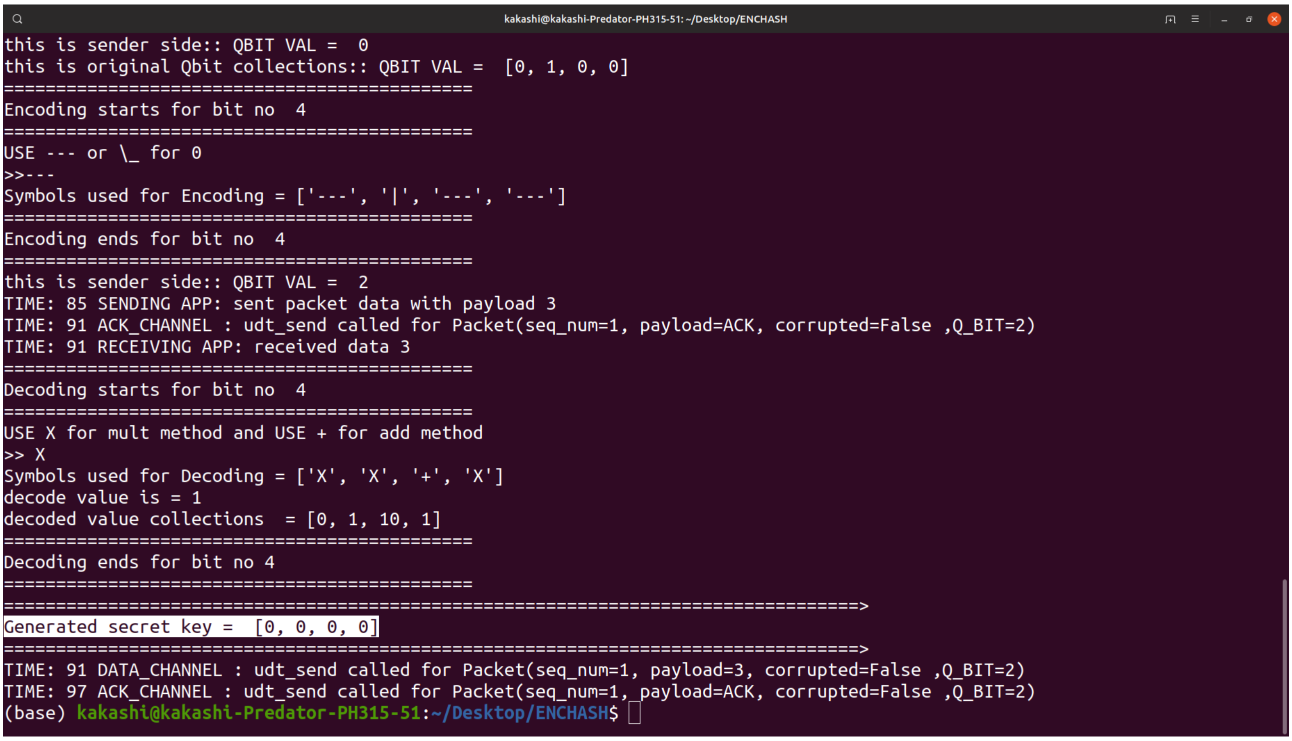
```
(base) kakashi@kakashi-Predator-PH315-51:~/Desktop/ENCHASH$ python3 Main_prog.py
TIME: 1 DATA_CHANNEL : udt_send called for Packet(seq_num=0, payload=0, corrupted=False ,Q_BIT=2)
TIME: 7 ACK_CHANNEL : udt_send called for Packet(seq_num=0, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 7 ACK_CHANNEL : udt_send called for Packet(seq_num=0, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 7 DATA_CHANNEL : udt_send called for Packet(seq_num=0, payload=0, corrupted=False ,Q_BIT=2)
TIME: 13 ACK_CHANNEL : udt_send called for Packet(seq_num=0, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 13 DATA_CHANNEL : udt_send called for Packet(seq_num=1, payload=0, corrupted=False ,Q_BIT=2)
this is sender side:: QBIT VAL =  0
this is original Qbit collections:: QBIT VAL =  [0]
=============================================
Encoding starts for bit no  1
=============================================
USE --- or \_ for 0
>>
```

2.2 DECODING



```
(base) kakashi@kakashi-Predator-PH315-51:~/Desktop/ENCHASH$ python3 Main_prog.py
TIME: 1 DATA_CHANNEL : udt_send called for Packet(seq_num=0, payload=0, corrupted=False ,Q_BIT=2)
TIME: 7 ACK_CHANNEL : udt_send called for Packet(seq_num=0, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 7 ACK_CHANNEL : udt_send called for Packet(seq_num=0, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 7 DATA_CHANNEL : udt_send called for Packet(seq_num=0, payload=0, corrupted=False ,Q_BIT=2)
TIME: 13 ACK_CHANNEL : udt_send called for Packet(seq_num=0, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 13 DATA_CHANNEL : udt_send called for Packet(seq_num=1, payload=0, corrupted=False ,Q_BIT=2)
this is sender side:: QBIT VAL =  0
this is original Qbit collections:: QBIT VAL =  [0]
=============================================
Encoding starts for bit no  1
=============================================
USE --- or \_ for 0
>>---
Symbols used for Encoding = ['---']
=============================================
Encoding ends for bit no  1
=============================================
this is sender side:: QBIT VAL =  2
TIME: 13 SENDING APP: sent packet data with payload 0
TIME: 19 ACK_CHANNEL : udt_send called for Packet(seq_num=1, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 19 RECEIVING APP: received data 0
=============================================
Decoding starts for bit no  1
=============================================
USE X for mult method and USE + for add method
>>
```

2.3 GENERATED KEY



```
this is sender side:: QBIT VAL =  0
this is original Qbit collections:: QBIT VAL =  [0, 1, 0, 0]
=============================================
Encoding starts for bit no  4
=============================================
USE --- or \_ for 0
>>---
Symbols used for Encoding = ['---', '|', '---', '---']
=============================================
Encoding ends for bit no  4
=============================================
this is sender side:: QBIT VAL =  2
TIME: 85 SENDING APP: sent packet data with payload 3
TIME: 91 ACK_CHANNEL : udt_send called for Packet(seq_num=1, payload=ACK, corrupted=False ,Q_BIT=2)
TIME: 91 RECEIVING APP: received data 3
=============================================
Decoding starts for bit no  4
=============================================
USE X for mult method and USE + for add method
>> X
Symbols used for Decoding = ['X', 'X', '+', 'X']
decode value is = 1
decoded value collections  = [0, 1, 10, 1]
=============================================
Decoding ends for bit no 4
=============================================
==================================================================================>
Generated secret key =  [0, 0, 0, 0]
==================================================================================>
TIME: 91 DATA_CHANNEL : udt_send called for Packet(seq_num=1, payload=3, corrupted=False ,Q_BIT=2)
TIME: 97 ACK_CHANNEL : udt_send called for Packet(seq_num=1, payload=ACK, corrupted=False ,Q_BIT=2)
(base) kakashi@kakashi-Predator-PH315-51:~/Desktop/ENCHASH$ 
```

3.CONCLUSION :-

This project is a small step toward digitalization of Qbit so that we can replicate quantum paticles behavior and gain the Quantum computaion power using classical computer or atleast we can use classical computers to aproximate the actual result results of quantum computers.

In this field IBM is now taking small step to achive this goal so IBM’s QSKIT is a good example. A common developer who don’t have quantum computer also can aproximate the results of circuit formed by quantum gates.

4.REFERENCES:-

LINK- https://en.wikipedia.org/wiki/Qiskit

BOOKS – THE CODE BOOK -simon singh, Quantum Computation and Quantum Information -Isaac Chuang, Michael Nielsen