Implementation of Quantum-Text Steganography using BB84 protocol on an Interactive Module

Shisheer S Kaushik
B.E, Dept. of ECE
BNMIT
Bangalore, India
shisheerkoushik24@gmail.

Keerti Kulkarni
Assistant Professor, Dept. of ECE
BNMIT
Bangalore, India
keertikulkarni@bnmit.in

***Abstract*- In the modern online era, the privacy of individuals and organizations is decreasing day by day. The search of high encryption standards with the use of keys always ends up in encryption techniques like AES, DES but they are suspected to be vulnerable to Quantum Computers. To tackle this respective dispute, this paper proposes a quantum steganography technique administered by BB84 key distribution protocol. This protocol uses plain text encoded into the cover file which dissociates the secret message dispatched between multiple parties. Subsequently, a stego object is created without altering the content of plain text which contain the secret message. These embedded messages are incorporated as phases of the entangled states and ultimately shared among respective parties in prior as a quantum keys, which are later utilized during retrieving the secret message from stego object by the corresponding party.**

***Keywords*- Qiskit, Stego object, steganography, ASCII.**

I. Introduction

The topic of Cryptography is the modus operandi for secure communication between two or more parties. One prominent cryptographic dispute is the conveyance of secret messages. This is possible through certain cryptographic protocols. With the progress of information technology and advancement in extensive research, the concern of data breaches and the matter of information security has become growingly imperative. In order to resolve this dispute, many plausible researches are in progress. One among them is the quantum-Key distribution (QKD), a quantum cryptographic communication protocol which exploits the principles of quantum mechanics to empower the secured and reliable distribution of confidential information. According to the Copenhagen interpretation [1], the wave-function collapses as a consequence of a quantum state being measured, which in turn ends up in changing its state provided the respective quantum state is not an eigenstate. From the changes observed in the quantum state, any attempt of tampering or interceptive action by the third party (eavesdropper) can be detected. This unique characteristic of QKD protocols does play a vital role in developing a secured communication network among two or more connections.

The fact that, several hardware devices are already available commercially, has untangled the vigorous efforts required for deploying QKD protocol to communicate photons across distant network by achieving lower losses and noise. Harnessing the QKD protocol can create a secured universal quantum internet by adopting quantum cryptographic protocols to tackle the vulnerabilities confronted by the classical cryptography. One of the grounds discussed in information security is the certain effective techniques namely cryptography, Information coding, steganography etc. being utilized for a secured exchange of information through the cover media. In accordance with quantum computing domain, this paper demonstrates the flow of quantum information on an interactive module by enabling a quantum steganographic technique synonymous to classical steganography, which uses a classical secret key shared among communicating parties, (i.e. Adam and Bob).

Finally, the effect of noise on the information (secret message) due to the subjective error factors are analysed by testing the proposed protocol on two distinctive backend devices.

II. Related Work

Previous work results of quantum steganography like [2, 3] using quantum error-correcting codes and [4] using modified super dense coding are not adequate as their inference failed to display the technique embedding it within cover file (carrier message). This dispute was resolved by Shaw and Brun's work, which uses quantum error-correcting codes [5] for constructing a quantum steganographic protocol. It displays a meticulous procedure to embed private message (stego-object) to cover file. Three different such quantum stenographic protocols were also proposed by Al [6]. The proposed approach in this paper is based on Martin's approach [7] of using quantum-key distribution protocol (QKD) proposed by Bennett and Brassard, in which he showed how to hide a stenographic channel in the QKD protocol. This paper demonstrates an interactive implementation of BB84 protocol for proving a secured channel for communicating quantum information.

The BB84 protocol

The BB84 communication protocol [8] was the very first proposed QKD scheme, where the states are encoded in $\hat{z}$-basis but eventually measured in the $\hat{x}$-basis (and vice versa) leading to collapse arbitrarily, generating arbitrary measurement of end results, on the other hand the measured states encoded in same basis are accurately communicated in a single bit of information.

When BB84 protocol are photonically implemented, it is transmitted in a sequence of single photons which are polarisation-encoded with arbitrary data. Once the protocol attains its completion, Adam and Bob dispense an arbitrary bit string for using a one-time pad cipher for generating impeccable information security. The BB84 protocol is demonstrated as following steps:

1. Adam selects an arbitrary bit, either 0 or 1.
2. Adam arbitrarily selects a basis, either $\hat{X}$ or $\hat{Z}$.

3. Based on the choices of basis, Adam encodes his bit into the polarization of single photon as shown below.

$$|0\rangle_Z \equiv |H\rangle$$
$$|1\rangle_Z \equiv |V\rangle$$

Or

$$|0\rangle_X \equiv \frac{1}{\sqrt{2}}(|H\rangle + |V\rangle),$$
$$|1\rangle_X \equiv \frac{1}{\sqrt{2}}(|H\rangle - |V\rangle).$$

4. Once the randomly selected basis is encoded, the arbitrarily chosen bit is transmitted to bob.
5. He spontaneously does not attempt (publicly) to announce his selection of chosen bit or basis.
6. Consequently, the received bit is measured in a randomly chosen basis, $\hat{X}$ or $\hat{Z}$ by Bob. This process is repeated until all qubits are received.
7. Adam then (publicly) announces the selected basis, used for encoding every bit sent.
8. The oppositely measured Qubit basis from the one Adam encoded are discarded, as they will be decorated from Adam.
9. The remaining measurement outcomes are guaranteed to generate indistinguishable bits between Adam and Bob.
10. The rest is roughly half as many bits as which were sent, they are random but believed to be identical between Adam and Bob.
11. Some of their bits are sacrificed by openly communicating in order to inspect consistency. This technique can evade the suspected intercept-resend attacks.
12. In an effort to refine the partially compromised key into shorter but more secret one, a more subtle approach such as Privacy amplification could be adopted.

During the stage when there is an intercept-resend attack [9] on the channel caused by an eavesdropper (Eve), amidst communicating between Adam and Bob. Eve will not have the knowledge of corresponding bases since the choice of base which is to be encoded is still not publicized by Adam. It results in random collapsing of measured states onto the inconsistent values of the Adam's encoding. Thus, by publicly communicating among them for comparing and exchanging of keys, such suspected attacks can be evaded. Hence, with the confidence of having shared, secret, random bit string, Adam and bob can sanguinely communicate in a channel with a well-established privacy and security.

IV. Proposed Approach

In the proposed approach the idea is to encode the sender's text (s) using the concept of quantum interference [10] incorporating the BB84 protocol as the basis for exchange of information in a secured channel. The secret message is encoded based on the cover file (a text sentence in this case). The cover file (carrier message) and the secret message is chosen to be in text formats to make it easier for the novice to understand the concept of quantum steganography. Figure 1 depicts the proposed protocol flowchart.

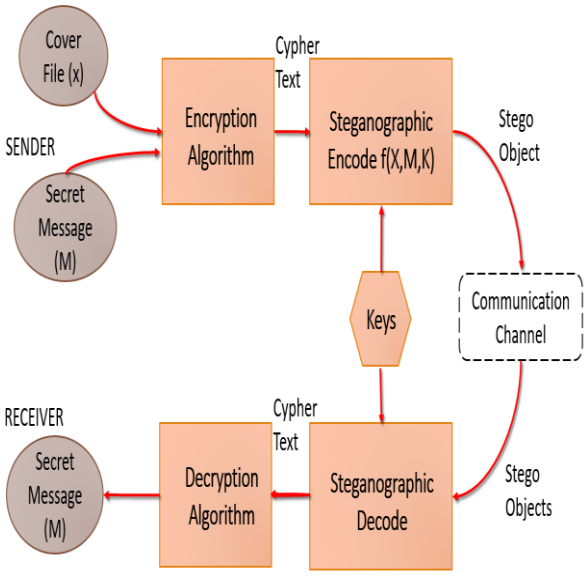

Figure 1: The Flowchart of the proposed protocol

Implementation Details

In order to implement the proposed protocol using a backend code precisely for executing quantum application, an open source software developed by IBM named QISKIT [11] was utilized. This software enables the quantum computing till machine level code of QASM (Quantum Assembly level language) [12]. It uses Qubits and Quantum circuits in place of bits and classical circuits. Synonymous to a bit in classical computer, a Qubit is the basic unit of information in a quantum computer.

The vital distinction among a Classical bit and that of a Qubit, is the binary values that they refer. A classical bit take either zero or one, correspondingly a Qubit, which uses the properties of quantum Mechanics can take the superposition value of zero and one thus resulting in a wider range of possibilities and enhancing the difficulty in deciphering. The supporting backend code for the proposed protocol was implemented using interactive frontend application software, Streamlit [13]. It provides a platform to create circuits based on the number of qubits desired by the user. In the proposed case (ideally) for generating random number, the encoder circuit is based on 7 qubit system and possible up to 10 qubits.

The proposed interactive module follows these respective steps:

1. The desired number of qubits is chosen using the slider. It is suggested to choose 7 as shown in figure 2 for the encoding and decoding scheme to work flawlessly. But eventually any arbitrary number of qubits can be opted, it purely depends upon the choice of processors volume [14] for generating a circuit.

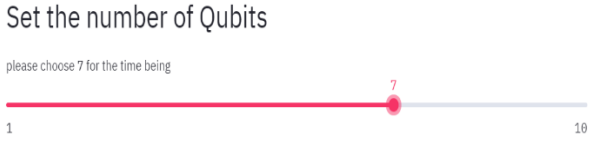

Figure 2: The slider button to set the number of Qubit

2. The Secret message has to be edited and pressed enter to get updated into the encoder. As seen in figure 3, a phrase '***bnmit'*** is chosen as one of the secret message which is to be exchanged.

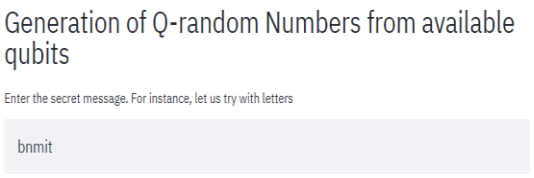

Figure 3: Text box for uploading a secret message

Figure 4 displays the generated circuit with respect to updated secret key into the encoder.

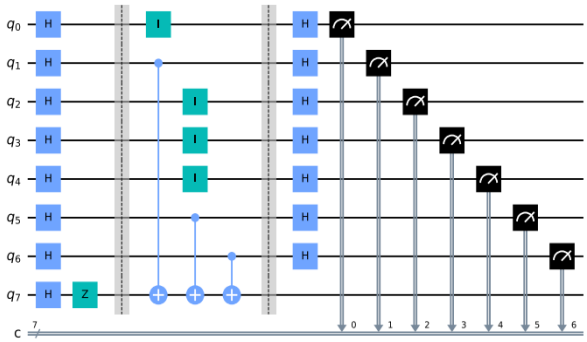

Figure 4: Encoder circuit of a secret key

It is to be noted that during encryption and decryption algorithm the secret key which is in ASCII-Unicode format is converted to binary [15], hence the probabilistic graph in QISKIT shows the results in Binary equivalence for its respective ASCII-Unicode value. Figure 5 displays the probabilistic output graph generated from the updated circuit achieving 4096 counts (i.e. The total number of iterations a circuit is run is specified through the shots argument during executing process) for the text *bnmit*.

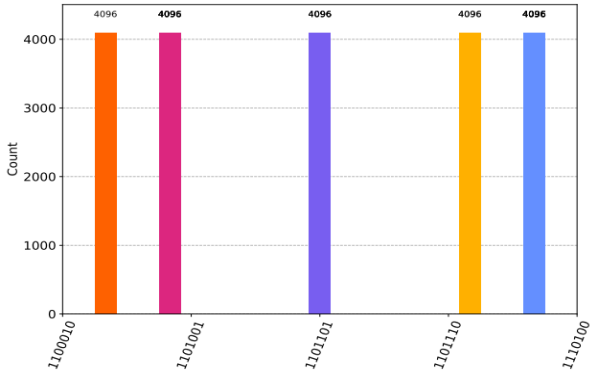

Figure 5: Probabilistic output sent from the source (Adam)

3. A sentence or phrase is updated into the encoder as a cover file (carrier message) which allows the sender to hide the secret message.

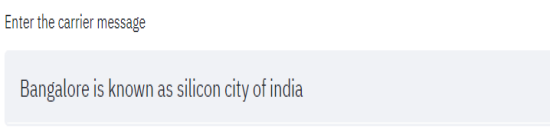

Figure 6: Text box for uploading the cover file (carrier message)

The sentence '***Bangalore is known as silicon city of india***', was chosen in the proposed interface module as shown in figure 6.

4. In order to initialise the encoding process, a 'encrypt' button as displayed in figure 7 has to be clicked in the interactive module. The encrypted text is generated as shown in Figure 7.

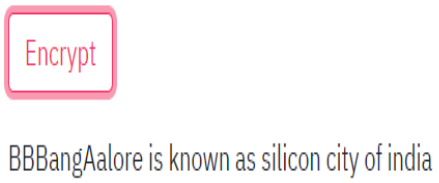

Figure 7: Encoder button and the encoded message (Stego text)

5. Ultimately, once the encrypted message has reached the destination source, the receiver (Bob) gets back the decrypted text by initiating the decoding process by clicking on a 'decode' button as displayed in figure 8.

6. Finally, the decoded text is retrieved by the receiver (Bob) using the decode button as shown in figure 8.

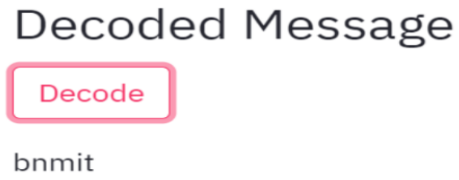

Figure 8: Decoder button and the decoded message

When the code is executed, a GUI window supported by Streamlit API [16] displays the graph showing the probability of the received secret key. It is evident from the interactive module that, the secret message 'x' exchanged between the parties has securely reached its respective destination using the proposed protocol.

IV. Evaluation and Results

Since the proposed protocol is implemented using the Qiskit {SDK}, it helps users to execute all the circuits virtually either in a simulator or in a real quantum processors accessible through cloud platform. Henceforth, in order to obtain more precise, accurate results and also to compare the performance of each backend by analysing its respective error occurrence in counts, the proposed protocol was implemented on two distinctive backend devices by setting the number of simulation shots to be 1000.

In an intension to demonstrate and elucidate the variations observed in the generated circuit's output, while performing the error or noise analysis using 2 distinctive backend devices, a much concise approach was utilized by using a single variable secret key '**x**' from the interactive module, as it produces less noise in the inference.

During the analysis 3 distinctive types of results were observed as follows:

a) On Simulator [qasm-simulator]:

Figure 9 shows the flawless results of retrieving character '*x*' at receivers end with 100 % or achieving full 4096 counts accuracy obtained from the proposed protocol when the 'qasm_simulator' (QASM) was utilized as its backend device.

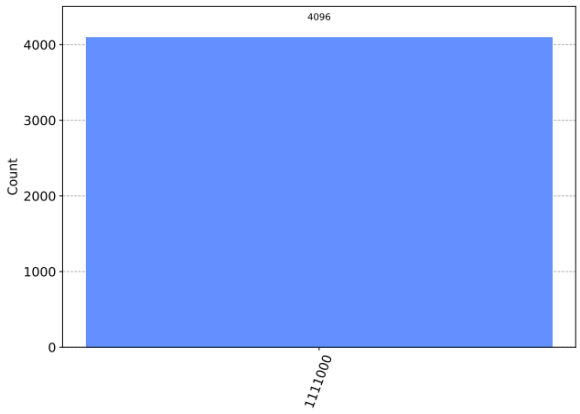

Figure 9: Probabilistic output retrieved by the receiver performed on the 'qasm_simulator'

b) On real Quantum computer [ibmq-melbourne]:

When the proposed protocol was tested using the real quantum processor 'ibmq-melbourne' which supports OpenQASM 3 [17] as its backend device, due to some subjective error factors, the results obtained were unevenly distributed probabilistically with the highest achievable count of 740 (out of 1000) observed at [1111000] or '*x*' (ASCII-Unicode) as shown in figure 10.

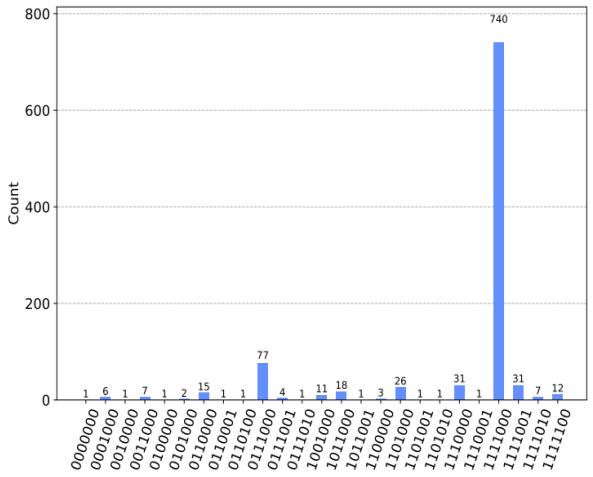

Figure 10: Probabilistic output retrieved by the receiver performed on the 'ibmq_melbourne'

c) On real Quantum computer [ibmq-melbourne] {Hybrid Model}:

When the proposed protocol opted for hybrid method with a simulator at senders end and the real quantum processors (ibm-melbourne) at receivers end, the results were observed to be 792 count (out of 1000) marking as the highest achievable count at [1111000] or '*x*' (ASCII-Unicode) but the percentage of error obtained were comparatively low from the results of real quantum processors (ibmq-melbourne) which supports OpenVAS 3 as shown in figure 11.

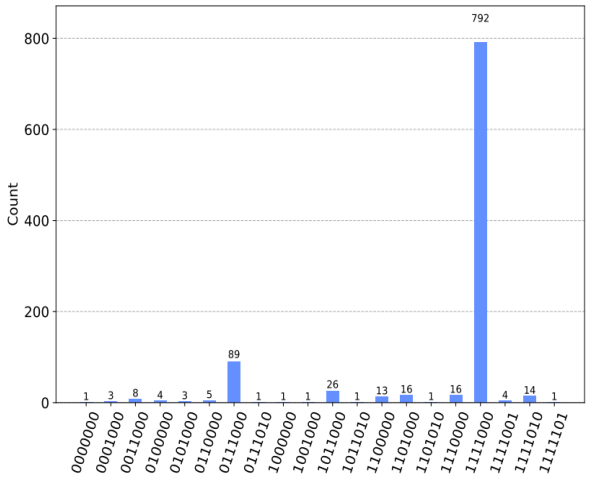

Figure 11: Probabilistic output retrieved by the receiver performed on the 'ibmq_melbourne' {Hybrid}

It is perfectly evident in the graphs that the simulation results performed on real quantum computer which supports OpenQASM 3 is comparatively distinctive from the one performed on the qasm_simulator (QASM). It's because quantum computers are immensely onerous to manufacture and program. Due to effects of quantum decoherence [18] and other environmental factors, the operations performed on the quantum processor are sabotaged by the noise, faults and losses (errors) before any of the program has attained its completion.

V. Conclusion

Quantum steganography, has currently become a significant accomplishment in data security engineering, it plays a vital role in perfectly securing the bank transactions by providing platforms for sharing confidential information between the involved parties. The proposed paper demonstrates the manner in which a shared text (cover file) can be utilized to hide a secret message, using the encoding idea from BB84 protocol.

The notion of utilizing the concept of quantum steganography for transmitting different type of text data over the network by contemplating it as secret message has been the highlight of this work. In addition, the works also describes the subsequent noises and overlapping of expected data due to multiple error factors when the same proposed protocol is tested on 3 distinctive backend devices.

As a future work, with the progressive development in the quantum hardware domain, when the error-free quantum computer with larger volume capacity becomes reality. The proposed protocol can be implemented by utilizing larger data capacity which in turn enables the security of global quantum internet.

VI. Reference

[1] Martin Kober, "Copenhagen Interpretation of Quantum Theory and the Measurement Problem", Quantum Physics, Vol 3, Nov 2009, doi:10.48550.

[2] M. Curty, D. J. Santos, "Quantum Steganography," 2nd Bielefeld Workshop on Quantum Information and Complexity, 12-14 October 2000, pp. 12-14.

[3] J. Gea-Banacloche, "Hiding Messages in Quantum Data," Journal of Mathematical Physics, Vol. 43, No. 9, 2002, pp. 4531-4536. doi:10.1063/1.

[4] S. Natori, “Why Quantum Steganography Can Be Stronger than Classical Steganography,” Quantum Computation and Information, Vol. 102, 2006, pp. 235-240. doi:10.1007/3-540-33133-6_9.

[5] Bilal A. Shaw, Todd A. Brun, “Quantum Steganography”,Quantum Phhysics, Vol 1, 2010, pp 1-22. doi:10.1103/PhysRevA 83.022310.

[6] Al Ghania, Ghania Al Sadi, “Image Steganography Approach”, International journal of computer science and mobile computing, Vol 4, 2015, pp 166 – 169.

[7] K. Martin, “Steganographic Communication with Quantum Information”, Conference: Information Hiding, 9th International Workshop, Vol 4567, 2007, pp 32 – 49. doi:10.1007/978-3-540-77370-2_3.

[8] C. H. Bennett, G. Brassard, “Quantum Cryptography”, Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, 9-12 December 1984, pp. 175-179.

[9] Hiroo Azuma, Masashi Ban, “The intercept/resend attack and the collective attack on the six-state protocol of the quantum key distribution”, Quantum Physics, Vol 1, 30 November, doi:10.48550/arXiv.1912.00196.

[10] Maunz, P. et al. “Quantum interference of photon pairs from two remote trapped atomic ions”. Nat. Phys. 3, 538– 541 (2007).

[11] Cross A W, Bishop L S, Smolin J A and Gambetta J M “2017 Open quantum assembly language”, (arXiv:1707.03429).

[12] Qasm_simulator Documentation : https://qiskit.org/documentation/stubs/qiskit_aer.QasmSimulator.html

[13] Stramlit documentation : https://docs.streamlit.io/

[14] Yuxuan Zhang, Daoheng Niu, “Quantum Volume for Photonic Quantum Processors”, Quantum Physics, August 2022, (arXiv:2208.11724v1),.

[15] ASCII-Binary Character Table : http://sticksandstones.kstrom.com/appen.html

[16] Streamlit API : https://docs.streamlit.io/library/api-reference

[17] OpenQASM 3: Andrew Cross, Ali Javad-Abhari “OpenQASM 3: ABroader and Deeper Quantum Assembly Language” Research Article ACM Transactions on Quantum Computing”, Vol. 3, No. 12, 2022, pp. 1-50. doi:10.1145/3505636.

[18] P.L. Knight, M. B. Plenio, Vlatko Vedral “Decoherence and Quantum Error Correction” , Philosophical Transactions of The Royal Society B Biological Sciences, November 1997, pp 355(1733).doi: 10.1098/rsta.1997.0134.