IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Quantum Cryptography Algorithms Assessment: A Comprehensive Study Using IBM's Qiskit Framework

Karan Balkrishna Khandekar
Computer Science Department
Patkar Varde College
Mumbai, India

Nafisa Mohd Saad Ansari
Computer Science Department
Patkar Varde College
Mumbai, India

ABSTRACT

Quantum cryptography is revolutionizing secure communication systems by providing unprecedented security advancements, particularly through Quantum Key Distribution (QKD). The invention of the QKD system has significantly increased the level of security in exchanging private keys. This paper focuses on the study of three widely recognized QKD protocols—BB84, SARG04 and E91. We used IBM's quantum computing framework, Qiskit, to simulate these protocols. We assessed and compared these algorithms by simulating them on our local machine via Qiskit's "qasm_simulator." Our focus was on key generation rates and error rates both in the presence and absence of an eavesdropper. We experimented using varied bit lengths (i.e., number of qubits) to observe how the protocols behave across different scales. Additionally, this study incorporated noise models such as FakeRochester, FakeMelbourne and FakeParis to simulate real-world imperfections such as decoherence and evaluate the simulator's ability to accurately model noise. The results revealed that SARG04 consistently achieved perfect key generation, whereas BB84 demonstrated greater resilience, particularly with longer bit lengths, compared to E91, which performed poorly in noisy environments. The selection of these protocols might be contingent upon particular application needs and their resistance to noise and eavesdropping. Future work could explore enhancements to these protocols or hybrid approaches that combine their strengths to achieve even greater efficiency and security in quantum cryptographic systems.

Keywords—Quantum Key Distribution; Quantum Cryptography; BB84; E91; SARG04

I. INTRODUCTION

The two primary types of cryptosystems used today are symmetric systems, which encrypt and decrypt data using a single secret key, and asymmetric systems, which use a public key that is accessible to all parties and private keys that are only accessible to authorised parties. These traditional encryption algorithms, which depend on the difficult task of factoring large numbers into their prime components to generate keys, have long been the foundation of protecting sensitive data. Since a hacker would need years to even approach the computational capacity needed to factor a large number on a classical computer, this method works very well against the capabilities of classical computing. But the coming development of quantum computing poses an immediate threat. Given their exponentially greater computational capacity, a quantum computer would find a solution within a matter of minutes. Therefore, the advent of quantum computing signals the beginning of the post-quantum era, which calls for a review of encryption techniques to protect against potential weaknesses in data security.

Cryptography is the science and study of securing communications through encoding bits using mathematical theorems and techniques. Unlike traditional encryption methods Quantum cryptography on the other hand leverages the fundamental laws of quantum mechanics and encodes data in quantum states, providing an unique, quantum-physics-based method for developing secure communication channels. Such a quantum system will always be affected significantly by the simple act of measurement or even observation, thus securing the integrity of the data. Quantum Key Distribution is one such common type of quantum cryptography method wherein a secured shared cryptographic key is generated between two parties by communicating through a quantum channel utilising the properties of photons.

In this paper, we have compared three such algorithms namely BB84, SARG04, E91. For the simulation of these algorithms, we have used Qiskit, IBM's Quantum computing framework [2], [9]. We have studied these algorithms extensively mainly in their error production rates and key generation rates with different bit-lengths and later we have introduced an eavesdropper entity into our algorithms to check for the security aspects of these algorithms in the presence of an eavesdropper.

II. EXPERIMENTAL PHASES

A. Varying Bit length

In order to conduct the experiments, different bit lengths were used, which are essentially the number of qubits, a set of two state quantum mechanical systems [3], [4] utilised in the quantum algorithms simulated in Qiskit's QASM simulator. As an illustration, during our research:

Bit Length 4: This translates to using fewer qubits to run the algorithms, which makes computation and simulation simpler.

Bit Length 20: This indicates a higher qubit count, indicating the larger scale behaviour of the algorithms and requiring more processing power.

B. Introducing an Eavesdropper

Simulation Without Eavesdropper: To determine error rates and key generation rates, preliminary assessments of all three algorithms were carried out (for both bit lengths) without the use of an eavesdropper.

Simulation with Eavesdropper: Next, in order to evaluate the algorithms' resilience to possible intrusions, we introduced an eavesdropper into the system and changes in error rates and key generation rates were tracked(for both bit lengths).

C. Fake Noise Models

Fake noise models such as FakeRochester, FakeMelbourne and FakeParis were introduced into the algorithm to check for the noise in the simulators in case of SARG04 and E91 algorithms and the results turned out to be the same as when a noise model was not implemented into the algorithm assuring of the noise simulating capabilities of the simulator.

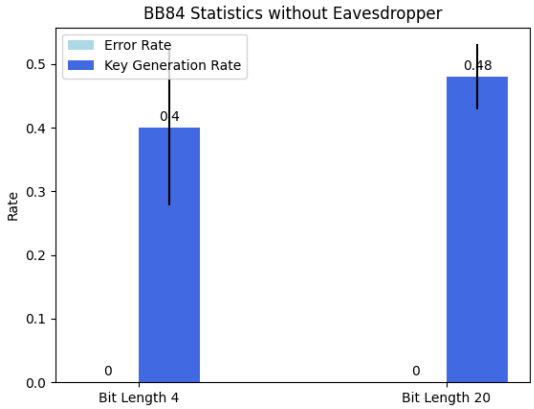
III. SIMULATION OF QUANTUM KEY DISTRIBUTION PROTOCOLS

A. BB84 protocol

The quantum key distribution (QKD) protocol known as BB84 was developed in 1984 by Charles Bennett and Gilles Brassard [1]. Its safety stems from the uncertainty principle, which is a fundamental principle of quantum mechanics. The protocol uses the quantum property that any attempt to obtain information about a quantum system will inevitably cause its state to be disturbed. There are four states of photon polarisation and two measurement bases utilised in the BB84 protocol. The algorithm performs the following processes to demonstrate the BB84 protocol:

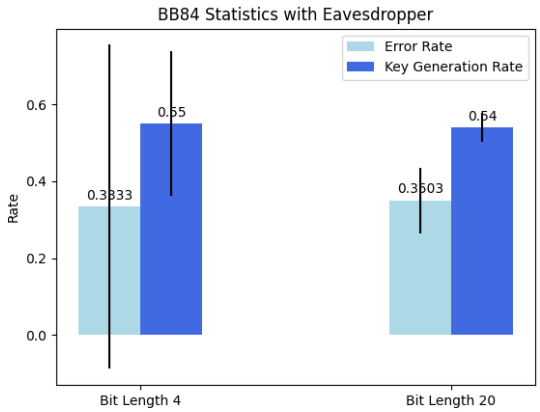
1. Alice creates a random bit-string to encode these into bits by choosing random bases.
2. Alice then encodes her bits into qubits based on the randomly chosen bases. She applies the Pauli-X gate for a bit of '1' and the Hadamard gate for a basis of '1'.
3. On the other side of the quantum channel Bob chooses his own set of random bases to measure these qubits sent by Alice.
4. Bob measures these qubits using his random set of bases and records his measurements.
5. Now, Alice and Bob both compare their chosen bases, if their chosen bases match, they keep the corresponding bits to generate a sifted key, if they don't match they are discarded.
6. Thus, a shared sifted key is generated after the execution of the code on IBM's Quantum Simulator.

The above algorithm was run several times to check the consistency of the results. We calculated the error rates by comparing the sifted keys of both Alice and Bob. The results for the BB84 protocol after simulation on IBM's QASM simulator were recorded and a statistical analysis was done.



- **Figure 1: Error rates and key generation rates for BB84 algorithm simulation with bit-lengths 4 and 20 in absence of an eavesdropper**

In these particular runs, the longer bit length demonstrated slightly more consistent and reliable key generation rates, even though both bit lengths achieved perfect error rates overall. Now, an eavesdropper(Eve) was introduced into the algorithm to check for its security aspects and evaluate how Eve's interference can influence the key generation rates and error rates.



- **Figure 2: Error rates and key generation rates for BB84 algorithm simulation with bit-lengths 4 and 20 in presence of an eavesdropper**

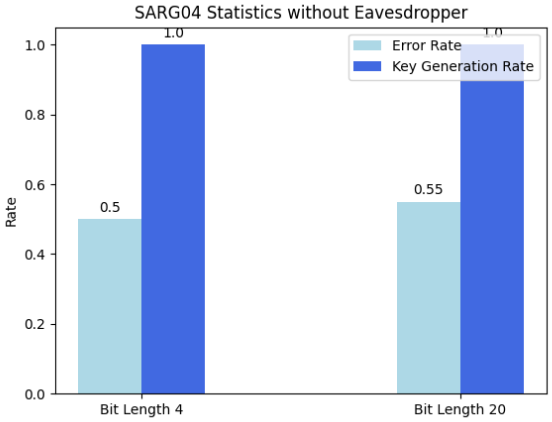
For both the bit lengths, the mean error rates are relatively close. When compared to bit length 4, the standard deviation of the error rates is smaller for bit length 20, suggesting less variability in the error rates. In comparison to bit length 20, the mean key generation rate for bit length 4 is slightly higher. When comparing bit length 4 to bit length 20, the standard deviation of the key generation rate is smaller for bit length 20, indicating more stable and consistent key generation rates throughout the various runs.

B. SARG04 protocol

In 2004, V. Scarani, A. Acin, G. Ribordy, and N. Gisin introduced this protocol [5]. In this protocol principles of quantum entanglement are used. This protocol is similar to BB84 except in the error correction phases. It has been demonstrated that this protocol is safe from a Photon Number Splitting (PNS) attack. It is unlikely that a PNS attack will take place given current technology. The protocol is as followed:

1. A function generates the entangled states required. To entangle the qubits, it employs a sequence of controlled-NOT gates and Hadamard gates [3].
2. An IBM quantum experience is set up connecting to the available provider and choosing a backend simulator.
3. A quantum circuit is initialised with specified bit-length to provide for the entangled state and measurements.
4. Measurements are made on each qubit in the circuit by a function. After that, the circuit is run on the chosen backend, and the outcome is obtained. The measured results are stored.
5. From these outcomes a sifted key is generated.

After the key generation, error rate and key generation rates over several runs for both the bit-lengths are calculated and analysed.

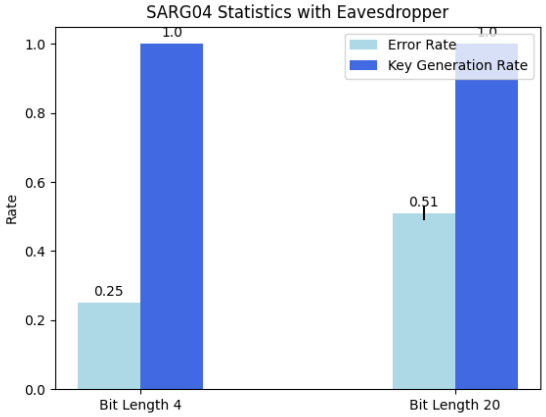


- **Figure 3: Error rates and key generation rates for SARG04 algorithm simulation with bit-lengths 4 and 20 in absence of an eavesdropper**

Both bit lengths showed error rates of 50% and 55% for 4 and 20 bits, respectively, in the absence of an eavesdropper.

The key generation, however, continued to be flawless and consistently achieved a 100% success rate for both bit lengths, indicating successful key extraction, in spite of these error rates.

Now, an eavesdropper is introduced into the algorithm similarly to the previous one. The changes in the key generation rates and error rates are again recorded and analysed.



- **Figure 4: Error rates and key generation rates for SARG04 algorithm simulation with bit-lengths 4 and 20 in presence of an eavesdropper**

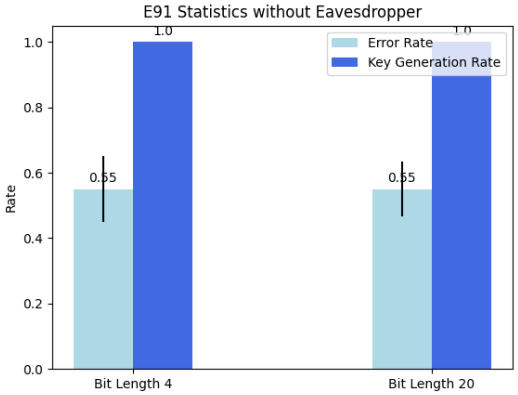
The presence of an eavesdropper resulted in transmission errors for both bit lengths. In contrast to a bit length of 20, the error rate was lower at a bit length of 4. But at 20 bits, the error rate rose, indicating a greater vulnerability to eavesdropping. In spite of errors, both bit lengths consistently generated keys at 100%, indicating that key extraction was successful even when an eavesdropper was present. While the key generation remained consistently successful at 100% for both bit lengths, the presence of an eavesdropper had an impact on the error rates, particularly for the longer bit sequence.

C. E91 protocol

Another quantum key distribution method is the Ekert protocol, which was put out by Artur Ekert in 1991 [7]. Bell's test identities are cleverly applied to entangled qubits, reinforcing the quantum entanglement concept at the very basis of the protocol [8]. E91 leverages Bell's theorem for quantum cryptography. The protocol is as followed:

1. A quantum circuit is initialised in order to produce an E91 entangled state. Hadamard gates (H) and controlled-NOT gates (CX) are used to prepare this condition [3].
2. Based on random selections (Alice's bases), qubits are measured in various bases (Z or X basis).
3. Bob, in the meantime, uses his randomly selected bases to measure the qubits that Alice sent and stores his measurement.
4. Bob and Alice's bases are compared to create the sifted key.

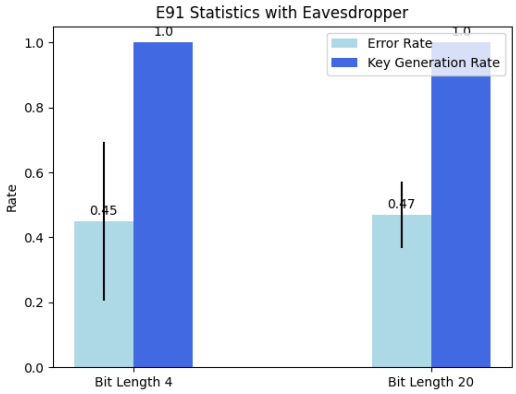
After the sifted key is generated, error rate and key generation rates over several runs for both the bit-lengths are calculated and analysed.



- **Figure 5: Error rates and key generation rates for E91 algorithm simulation with bit-lengths 4 and 20 in absence of an eavesdropper**

The E91 protocol showed a 55% error rate for both 4-bit and 20-bit sequences when there was no eavesdropper present. There was no fluctuation in the key generation rate, which remained constantly ideal at 100%.

Now, an eavesdropper (Eve) was introduced into the algorithm. Eve tried to intercept qubits by performing measurements (like Bob) but using her own random set of bases. The measurements were recorded and the sifted key thus generated was analysed.



- **Figure 6: Error rates and key generation rates for E91 algorithm simulation with bit-lengths 4 and 20 in presence of an eavesdropper**

The E91 protocol showed somewhat greater error rates with an eavesdropper present than it did without one for both 4-bit and 20-bit sequences. Nonetheless, the key generation rate stayed ideal and consistent at 100%.

IV. CONCLUSION

Different error rates and key generation rates were displayed by all three protocols when an eavesdropper was present.

1. When an eavesdropper was present, BB84 and E91 showed greater error rates with lower bit lengths.
2. While greater bit lengths resulted in a slightly higher error rate, SARG04 demonstrated a constant ideal key generation rate with a low error rate.
3. When an eavesdropper was present, E91's error rate was greater than BB84's in both bit length scenarios. In comparison with E91,
4. BB84 showed a decrease in error rates with higher bit lengths and SARG04 continuously produced faultless keys. The selection of these protocols might be contingent upon particular application needs and their ability to withstand eavesdropping.

V. FUTURE WORK

With the availability of real quantum computers through platforms like IBM Cloud, we can now test the scalability and robustness of protocols like SARG04 in practical settings. This access enables us to evaluate the protocol's performance in real-world quantum environments, assessing its faultless key generation under complex conditions and against potential eavesdroppers. Future work can also be on the modifications that would make it more resilient and use IBM's quantum infrastructure for testing and optimization. Techniques for error correction can also be implemented and analyzed on these quantum systems, thus opening up possibilities for more reliable and efficient protocols. Comparative studies with other algorithms in this practical setting could provide valuable insights into their real-world applicability and performance.

VI. REFERENCES

1. G. Bennett, Charles H and Brassard, “Proceedings of the IEEE international conference on computers, systems and signal processing,” IEEE New, 1984

2. “QISKIT,” QISKIT, “qiskit.org,” [Online]. Available: https://qiskit.org/

3. I. L. C. M.A Nielsen, “Quantum computation and Quantum information,” Cambridge Univ. Press, 2000.

4. M. S. Zubairy, “Quantum Mechanics for Beginners: With Applications to Quantum Communication and Quantum Computing,” Oxford Univ. Press, 2020

5. V. Scarani, A. Acin, G. Ribordy, and N. Gisin, "Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations," Physical Review Letters, vol. 92, p. 057901, 2004.

6. Charles H. Bennett, Gilles Brassard, “Quantum cryptography: Public key distribution and coin tossing”, Theoretical Computer Science, Volume 560, Part 1, 2014, Pages 7-11, ISSN 0304-3975

7. A. K. Ekert, “Quantum cryptography based on bell’s theorem,” Physical Review Letters 67, pp. 661–663, Aug 1991.

8. J. S. Bell, “On the problem of hidden variables in quantum mechanics,” Rev. Mod. Phys. 38, pp. 447–452, Jul 1966.

9. IBM, “IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation,” IBM News room, 2016, [Online]. Available: https://www-03.ibm.com/press/us/en/pressrelease/49661.wss