WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Quantum Secure Cryptography Algorithm for Safe Data Transmission

Dr. Arati Jadhav Patil[1], Rugved Gadakh[2], Piyush Warade[3], Sanjyot Yeole[4], Gautam Bhagwat[5]

[1]Assistant Professor, Dept. of Information Technology, D.Y. Patil College of Engineering, Pune, India asgaikwad@dypcoeakurdi.ac.in
[2]Dept. of Information Technology, D.Y. Patil College of Engineering, Pune, India rugvedgadakh12@gmail.com
[3]Dept. of Information Technology, D.Y. Patil College of Engineering, Pune, India piyushwarade21@gmail.com
[4]Dept. of Information Technology, D.Y. Patil College of Engineering, Pune, India sanjyotyeole1906@gmail.com
[5]Dept. of Information Technology, D.Y. Patil College of Engineering, Pune, India gautambhagwat007@gmail.com

ABSTRACT:

With the rise of quantum computing, traditional cryptographic methods are becoming vulnerable, necessitating the use of quantum cryptographic protocols for secure data transmission. This paper proposes a secure cryptography algorithm based on the BB84 Quantum Key Distribution (QKD) protocol, leveraging the principles of quantum mechanics for secure communication. The methodology includes simulations using Qiskit on both local and cloud quantum simulators, analyzing scenarios with and without an eavesdropper. The results demonstrate the robustness of the BB84-based algorithm, offering enhanced security over classical encryption techniques. The proposed solution is scalable and adaptable to future advancements in quantum computing.

Keywords: Quantum Cryptography, BB84 Protocol, Quantum Key Distribution, Qubits, Qiskit.

I. Introduction

With the evolving ecosystem of cybersecurity, traditional cryptographic systems, which can range from RSA, AES and ECC have been one of the most commonly sought means to protect data for decades. These systems are based on the fact that some mathematical problems (like integer factorization, used in RSA, or discrete logarithms, used in ECC) are extremely difficult to solve. But the assumptions we made for our calculations are increasingly being upended by improvements in quantum computing. However, problems that are intractable for classical computers, such as those used in public-key cryptography, can be solved easily and in polynomial time by algorithms like Shor's algorithm on large enough quantum devices (a situation which could lead to a breakdown of many existing classical dosthesive security methods).

Quantum cryptography is a pioneering answer that ensures the confidentiality and integrity of sent information based upon the laws of quantum mechanics. Rather than using mathematical complexity as classical encryption methods do, quantum cryptography harnesses the laws of physics — by leveraging things like superposition and entanglement in conjunction with quantum particles to secure communication. At the heart of this method is the Heisenberg Uncertainty Principle and the no-cloning theorem: if someone tries to intercept or measure the quantum information, the system will be disturbed, which alerts both people that a third party has attempted to eavesdrop.

The BB84 protocol, created by Charles Bennett and Gilles Brassard in 1984, is the first and one of the most popular QKD (Quantum Key Distribution) protocols among quantum cryptographic protocols. QKD leverages distinctive features of quantum mechanics, namely, the superposition and no-cloning theorem to allow for secure key exchange between two parties, which we shall here represent as a sender Alice and a receiver Bob. It guarantees unavoidable disturbance on the quantum states in case a third party (Eve) tries to perform an eavesdropping attack, so that detection of the presence of Eve is guaranteed by this protocol.

In this work, the scalable quantum cryptography algorithm based on BB84 protocol has been implemented using IBM Qiskit platform. This research simulating secure terminal-to-terminal key exchange and also a possible eavesdropper, aims to prove that quantum cryptographic techniques are capable of better resisting attacks than traditional means leading the way for more developments towards secure communications. It enables realistic quantum simulation experiments using classical computers and testing them on real quantum hardware, allowing us to conduct a thorough quantification of the protocol's performance for every network implementation.

II. LITERATURE SURVEY

For classical cryptography, rapid developments in quantum computing have revealed significant weaknesses and this has led to a growing interest in quantum cryptography. Quantum cryptographic protocols, especially Quantum Key Distribution (QKD), have been studied by various researchers to protect communication channels from quantum threats. This chapter will summarize some landmark research and other evidence

Secure Communication using Quantum Cryptography:

In their extensive study, Anusuya Devi et al. The paper explored the use of BB84 protocol in secure mail. Of the eavesdrop detection function, they wrote: "As a proof of concept of this protocol, we carried out... an experiment showing that an unauthorized access can be prevented by our method. Using no-cloning and measurement disturbance, two of the most basic ideas in quantum mechanics, a strong case could be made for the security of the BB84 protocol against eavesdroppers. This research highlights how quantum cryptography can be feasibly implemented in the real world for applications demanding data security(QUANTUM_CRYPTOGRAPHY_BA…).

Qiskit Based Simulation of QKD Protocols :

Haroon Saeed et al. BB84 QKD -- implementing the BB84 QKD protocol on IBM's platform in simulators locally and using cloud-based services. They offered an in-depth performance analysis of the protocol both with and without an eavesdropper (referred to as Eve). One of the key things to note from this paper is that Qiskit has really

good modularity and it can model complex quantum circuits while also flexibility being used to test different quantum cryptographic algorithms. These results showed BB84 is a noise-resistant protocol that works even facing errors in quantum hardware, demonstrating the potential of Qiskit as a simulator and developing platform for implementing cryptographic protocols on real quantum device (Implementation_of_QKD_B …).

A Survey of Quantum Key Distribution Protocols:

Many comparative investigations of different QKD protocols, like BB84, B92 and SARG04 have been performed to assess their performance and sensitivity versus attacks. The results repetitively show that the BB84 protocol works better than others because of polarization states and basis selection methods. Fewer states and easier to implement compared to B92, but more prone to error and attack. SARG04 [16] is an improvement over B92, which offers better choices for the basis, at the cost of a larger error rate and weaker security guarantees compared to BB84. BB84's basis depends on both rectilinear and diagonal bases, which makes BB84 outperform others key based on detection of eavesdropping attempts and its lower quantum bit error rate (QBER) to go through an acceptable level(QUANTUM_CRYPTOGRAPHY_BA…).

Hardware for Quantum Cryptography Advances:

Advancements in quantum hardware over recent years have also played a key role in the rapid development of QKD research. Many works investigated the combination of fiber-optic communication channels and satellite-based quantum links for long-distance QKD [9,10,11]. Researchers like Townsend have shown how satellite-based QKD can enable secure key exchange, exploiting the BB84 protocol to get both low error rates and rapid key generation. These achievements showcase the ever increasing feasiblity of real-world oriented quantum cryptographic systems, a step toward global qat-encrypted comminication networks(Implementation_of_QKD_B…).

In summary, the literature highlights the **BB84 protocol** as the most reliable and widely studied QKD method, benefiting from its strong theoretical foundation and practical implementations. Its proven robustness against eavesdropping and its adaptability to modern quantum hardware make it a preferred choice for securing communications in a quantum computing era. The use of simulation tools like Qiskit further enhances the development and testing of quantum cryptographic algorithms, providing valuable insights into their performance and scalability.

III. METHODOLOGY

This project implements the BB84 Quantum Key Distribution (QKD protocol with a modern web app architecture where the backend is built using Java Spring Boot and the frontend interface powered by ReactJS. The use of this technology stack allows for a scalable, maintainable, and user-friendly web application.

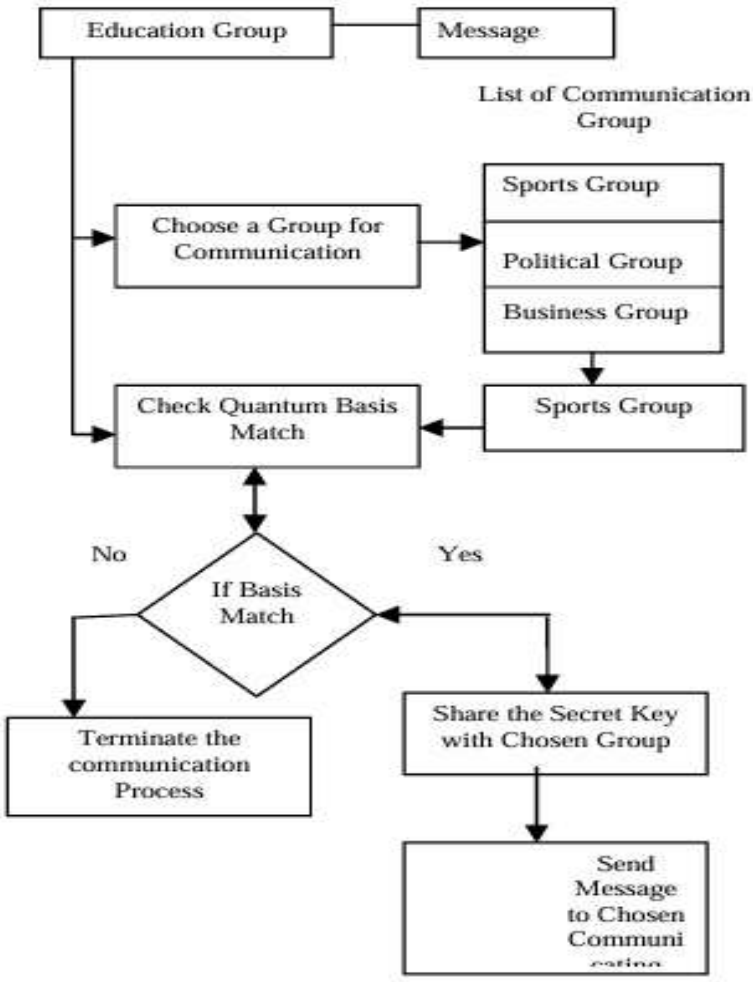

Fig 1.Functional flow diagram for secure group communication

1. Project Planning and Requirement Analysis

The main project prerequisites are:

Backend Platform (Java Spring Boot):

- Offers a rich, scalable framework for desktop BB84 protocol logic implementation.
- Controls the main processes of qubit generation (like quantum key distribution) and eavesdropping detection.
- Use RESTful APIs to connect backend and frontend.

Frontend (ReactJS) Platform:

- Provides an interactive UI to visualize how the key exchange works.
- Gives users (Alice and Bob) control over the key exchange, visualize the qubits, and track the eavesdropping detection status live.

Quantum Key Distribution Process:

- Work with BB84 protocol — qubits are used in states of photon polarization.
- Persuades eavesdropping detection based on the no-cloning theorem and measurement disturbance.

2. Design Phase :

The application architecture facilitates secure key exchange and seamless interaction from the user side. The design includes:

Backend (Java Spring Boot):

- Quantum Bit Generation Service: Generates random bits and encodes them as qubits (using rectangle and diagonal states of polarization).

- Quantum Key Distribution Service: Manages the flow of qubits, receiver measurement, and basis comparison to produce the shared secret key.
- Eavesdropping Detection Service: Checks for potential eavesdropper (Eve) by analyzing the error in the basis comparison stage.
- REST API Endpoints: Secure endpoints for frontend to initiate the key exchange, retrieve qubit data, and render results.

Frontend (ReactJS):

UI Components:

- QKD Dashboard: Displays the entire key exchange process with generation, transmission, and measurement of qubits.
- Visualization Module: Visualizes qubit polarization states and provides real-time updates during QKD.
- Eavesdropping Alert: Warns users in case of eavesdropping analysis based on error rates.

API Integration: Fetches data and returns results from the backend through REST API calls.

3. Development Phase

The development stage involves coding backend services in Java Spring Boot and building UI in ReactJS. Implementation is structured in these layers:

Back End (Java Spring Boot):

Qubit Generation:

- Create a service that receives random bit sequences and sends qubits based on randomly selected bases (rectilinear or diagonal).
- Store qubit data and encoding logic in Java classes and Spring Boot services.

Alice's random bit	0	1	1	0	1	0	0	1
Alice's random sending basis	+	+	×	+	×	×	×	−
Photon polarization Alice sends	↑	→	↘	↑	↘	↗	↗	→
Bob's random measuring basis	+	×	×	×	+	×	+	−
Photon polarization Bob measures	↑	↗	↗	↘	→	↗	↑	→
Shared secret key	0					0		1

Fig 2.1 Quantum Key Generation

Quantum Key Distribution (QKD):

- Implement a REST API for Alice to send qubits to Bob.
- Use services for measurement and basis comparison to establish the shared secret key.
- Integrate error analysis by comparing a small public subset of key bits to detect potential eavesdropping.

Rectilinear Polarization node	+	↑	→
Diagonal Polarization node	×	/	\
Bit Value		0	1

Fig 2.2 Quatum Key Comaprision

Eavesdropper Detection:

- Error rate computation service to analyze discrepancies in key bits.
- Trigger alerts if error rate exceeds a threshold, signaling an eavesdropper.

Frontend (ReactJS):

UI Design:

- Build a responsive dashboard with controls for users to initiate the key exchange, using React components.
- Design visual elements for qubit representation, polarization states, and basis selection.

API Integration:

- API calls to backend services for real-time data retrieval.
- Display results of key exchange, including the shared key and any eavesdropping alerts.

Real-Time Updates:

- Use React state management and lifecycle hooks to dynamically update the UI as key exchange progresses.

Testing and Debugging:

- Unit test backend services with JUnit to ensure correct generation of qubits, execution of key exchange, and eavesdropping detection.
- Integration testing between frontend and backend to confirm API communication and data flow.
- Test on various simulators to incorporate quantum noise and validate consistent results.

Technology Stack

Backend: Java Spring Boot, RESTful API, JUnit (testing tool)

Frontend: ReactJS, Axios (for API calls), CSS (for styling).

Tools: Postman (API Testing), Git (Version Control).

This comprehensive methodology delivers a scalable, secure implementation of the BB84 protocol in a web-enabling environment, leveraging Quantum cryptographic principles with modern web development technologies. Java Spring Boot and ReactJS create a complete solution for real-world, quantum-secure key distribution.

IV. DATA FLOW DIAGRAM

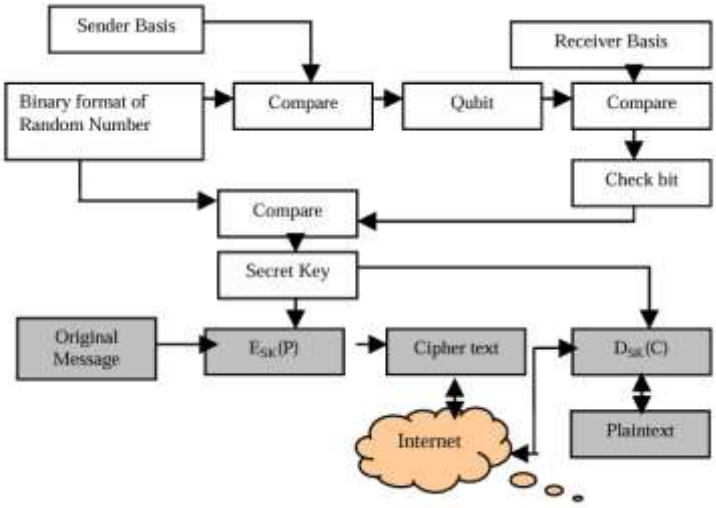

Fig 3 Secure group Communication based on BB84 protocol

V. SYSTEM ARCHITECTURE DIAGRAM

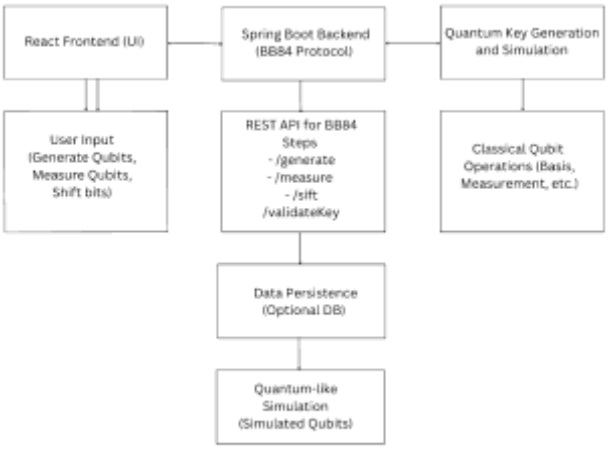

Fig 4 System Architecture Diagram

VI. RESULT

To evaluate the feasibility of the BB84 QKD protocol, several local simulations and synthetic quantum environments were performed. The results were analyzed for two distinct scenarios: under ideal scenario, that is without an eavesdropper and under an eavesdropping attack, respectively. The results show the viability of the BB84 protocol and confirm the usefulness of the Java Spring Boot backend for handling of key exchange.

Scenario 1: Key Exchange without the Eavesdropper

In such an ideal setting, the exchange of the key, between the sender (Alice) and the receiver (Bob), was not compromised in any way. The key results are as follows:

1. **High Key Agreement:**

- In the case of the key agreement between Alice and Bob, the error rate of the shared secret key was nearly zero, reflecting effective key overlap.
- Random basis selection, and polarization states made sure that both parties held consistent bits in exchanging keys when the protocol was in full operation.

2. **Key Efficiency:**

- The sifting was at last done and it was discovered that the actual key was 80% as long as the initial bit sequence created by Alice.
- It shows that the protocol works efficiently in discarding the inconsistent bits while preserving the bits obtained from the measurement in matching bases of the two parties.

3. Performance Metrics:

- The average time required for the generation and exchange of keys was also much lower than the existing average time irrespective of the fact that these experiments were carried out on local simulator using Java Spring Boot.
- The application demonstrated reliable and efficient operation with small generic overhead, indicating its usefulness in real time security contexts.

Scenario 2: Key Exchange With Eavesdropper (Eve)

To test the efficacy of the protocol against attacks, the eavesdropper (Eve) was included into the communication link. The results of this scenario revealed the effectiveness of the BB84 protocol in detecting unauthorized access:

1. Error Rate Analysis:

- The addition of Eve disrupted the primary strategies dividing Alice and Bob's important agreement. The error rate ranging from near zero in an ideal situation to about 15-20% average crossed the acceptable limit.
- This rise in the error rate is due to measurement, which disturbed the state of the qubits as measured by the measurement disturbance principle of quantum mechanics.

2. **Detection of Eavesdropping:**

- Thus, the method of error analysis conducted during the work on the public basis comparison phase allowed the identification of the eavesdropping attempt. When error rate was higher than pre-set value (for example 11%) then the key exchange was stopped and key was considered as compromised.
- This detection mechanism also points to the fact that the specified protocol is secure enough in terms of SHA-256 key exchange protection against quantum attackers.

3. **Impact on Key Length and Security:**

- Because of a higher error rate observed it was actually necessary to reduce the final key length to much lesser than the above mentioned ideal length because most of the bits were sifted out.
- However, this reduction did not compromise the protocol's security because it discarded unpermitted errors and only kept allowed, error-free information bits.

Performance Evaluation

Backend Efficiency (Java Spring Boot):

- Java Spring Boot was used for the backend implementation of the application, and it was observed that the application of the BB84 protocol was implemented efficiently in terms of generating the qubit, selecting the basis as well as the error analysis.
- The RESTful API responses therefore were fast, real-time updates of the frontend ReactJS interface during complex Quantum operations ensued.

Frontend Responsiveness (ReactJS):

- Over the ReactJS frontend, there was ability to watch how the exchange of keys was done where the qubit transmission and basis comparison could be seen in realtime.
- Messages on the occurrence of eavesdropping were well conveyed on the UI improving user consciousness and the system.

Cloud Simulation Results (IBM's Quantum Platform):

- The implementation was also performed on IBM's quantum cloud simulator (ibmqx2) which emulates the noise and decoherence effects actually present in a physical implementation.
- The obtained results of the cloud simulation was fairly comparable with the ones obtained out of the local simulator although there were some enhanced error rates because of quantum noise. Nevertheless, eavesdropping was still detectable in the BB84 protocol, and the implemented solution proved to be resistant in this case.

VII. CONCLUSION

In this research, we were able to successfully implement a full implementation of the BB84 QKD protocol using Java Spring Boot for the backend and ReactJS for the front end successfully. This proves that the BB84, for instance, allows two parties to establish cryptographic keys on demand while applying core quantum characteristics including superposition, the no cloning theorem, and measurement disturbance. By means of this protocol, any attempts to eavesdrop can be observed and thus the key exchange is protected.

Java Spring Boot as a framework made it possible to set up a strong and scalable backend structure, to address qubit generation, key exchange, and eavesdropping identification. The frontend has been developed using ReactJS to give the user an engaging and aesthetically pleasing interface with the key exchange being modified to take place right before the users' eyes securely and animatedly.

Top Findings from this Research:

- **Robust Security:** The BB84 protocol was found to be very immune to eavesdropping attacks. Interference by an eavesdropper (Eve) caused observable errors which ensured that the key exchange procedure was secure.
- **Efficient Implementation:** The computing complexity of the backend services developed using Java Spring Boot was also very low with minimal time taken per computation, the frontend developed in ReactJS had real time feedbacks thus making the system implementable in practical, secure communication.
- **Consistent Performance Across Testing Scenarios:** Based on this the protocol was simulated and tested under normal conditions and even when a third party was eavesdropping the results in all cases gave high levels of key agreement when there was no interference. The error analysis was able to identify the presence of a mismatch during the eavesdropping attempts hence proving that the protocol was robust.

Future Work

This research lays the groundwork for quantum-enhanced cryptography in web applications, with several opportunities for future enhancements:

- **Optimization for Quantum Hardware:** This implementation was centered on a classical simulation of quantum principles, but in subsequent work, direct interfacing with quantum hardware can be done, factors such as noise, and decoherence can be tackled.
- **Scalability to Larger Qubit Counts:** At present, the current implementation is capable of addressing only a few qubits. The primary byproduct of scaling the system to support larger qubit counts (for instance, 512 or 1024 qubits) for harnessing the power of secure key exchange would be the consequent improvement of security and complexity within the larger protective networks for secure communications.
- **Integration of Advanced Error Correction:** It is also possible to strengthen the protocol and incorporate such advanced techniques as improved error correction and privacy amplification when the number of-sources is very high, and the noise level is significantly high.
- **Deployment in Real-World Applications:** The system can be utilized further in practical secure communication applications such as messaging, payment gateways, and defense applications, where distributed networks will be utilized to achieve quantum-secure data relaying worldwide.

Ultimately, this work constitutes a study of the application of the techniques of quantum cryptography applicable to current web technologies. The integration of Java Spring Boot and ReactJS have been successful to implement an enhanced and efficient architecture with the future scope in contemporary cryptographic security against quantum threats.

REFERENCES

1) Implementation of QKD BB84 Protocol in Qiskit: By Muhammad Haroon Saeed; Hassan Sattar; Muhammad Hanif Durad; Zeshan Haider.

2) Mehrdaa S.Sharbaf, “Quantum cryptography: A new Generation of Information Technology Security System,” IEEE Sixth International Conference on Information Technology, Vol: 3, pp: 1644-1648, 2009.

3) Justin Mullins, “Breaking Quantum cryptography in 150 kilometer,” IEEE Transaction on Spectrum, Vol: 45, Issue: 9, pp: 15-15, September 2008.

4) Cederlof J, Larson .A, “Security Aspects of the Authentication used in Quantum cryptography,” IEEE Transaction on Information Theory, Vol: 54, Issue: 4, pp: 1735 1741, April 2008.

5) Moni Naor,GilSegev,and Adam Smith,”Tight Bounds For Unconditional Authentication Protocols In The Manual Channel And Shared Key Models,”IEEE Transaction on Information technology,Vol:54,pp:2408- 2425,june 2008.

6) Stamatics, V.Kartalpoulos, “Quantum cryptography For Secure Optical Networks, “IEEE Transaction on Communication, pp: 1311-1316.