ISSN2277-2685
IJESR/April. 2026/ Vol-16/Issue-2s1/59-68
Mohammed Sulaiman Ahmed Khan *et.al.*/International Journal of Engineering &Science Research

*Full Length Article*

# Quantum Secret Sharing Protocol Security In Cloud Environment

**Mohammed Sulaiman Ahmed Khan[1], Syed Khaja Faizuddin[2], Mohammed Amanullah[3], Ms. Rahma Bamasdoos[4]**
[1,2,3]B.E Students; Department Of Computer Science Engineering ISL Engineering College Hyderabad India
[4]Assistant Professor; Department Of Computer Science Engineering ISL Engineering College Hyderabad India
Mail Id: sakxlnc@gmail.com,syedkfaizuddin@gmail.com , Mohammedamanullah8841@gmail.com ,rahmabamasdoos@gmail.com

***Accepted 24-04-2026***


***Abstract***
*Quantum Secret Sharing (QSS) is a cryptographic primitive that distributes a secret among multiple parties using quantum information, ensuring only authorized subsets can reconstruct it. Existing QSS approaches rely on conventional network topologies and maximally entangled states, often requiring full participation and lacking efficiency in redundancy, authentication, and privacy. This work proposes a QSS framework on a distributed quantum network using a threshold-based (t, n) scheme, enabling flexible secret reconstruction. A custom weighting system and quantum variant of Dijkstra's algorithm dynamically select optimal players, improving performance. Additionally, CRYSTALS-Kyber post-quantum cryptographic primitives are integrated for secure authentication and identity obfuscation. Security analysis and simulations show that the proposed protocol outperforms traditional QSS models, offering stronger resilience against classical and quantum adversarial threats.*

***Keywords:*** *Quantum Secret Sharing (QSS), Threshold Scheme (t, n), Distributed Quantum Networks, Quantum Cryptography, Entanglement, Quantum Communication, Dijkstra's Algorithm (Quantum Variant), Post-Quantum Cryptography, CRYSTALS-Kyber, Authentication, Privacy Preservation, Secure Multiparty Computation, Quantum Security, Network Optimization.*

## Introduction

Secret sharing is a cryptographic methodology used to distribute a secret among multiple participants such that no individual possesses meaningful information independently. The secret can only be reconstructed when a required number of participants combine their shares, ensuring an "all or nothing" security principle. Two major schemes exist: the *(n, n)* scheme, which requires full participation, and the *(t, n)* threshold scheme, where any subset of *t* participants can reconstruct the secret. Foundational work by G.R. Blakley and Adi Shamir (1979) introduced geometric and polynomial-based secret sharing techniques. However, the emergence of quantum computing, particularly through Shor's algorithm, exposed vulnerabilities in classical cryptographic systems, leading to the development of Quantum Secret Sharing (QSS). QSS protocols utilize quantum principles such as entanglement, particularly Greenberger–Horne–Zeilinger (GHZ) states, to enable secure multi-party communication. Over time, various approaches including dynamic QSS, hierarchical models, d-dimensional GHZ schemes, and single-photon methods have been proposed to enhance security and efficiency. Despite these advancements, challenges such as fault tolerance, scalability, flexible routing, and secure authentication remain unresolved. Distributed network architectures improve scalability by allowing nodes (players) to connect dynamically without relying on centralized systems, thereby eliminating bottlenecks and single points of failure.

## Existing System

While existing distributed systems offer flexibility, they lack adaptability in dynamic environments and exhibit weaknesses in handling evolving network conditions. To overcome these limitations, the proposed system introduces a distributed quantum network framework with threshold-based secret sharing, dynamic routing, and enhanced security compliance with the extended CIA Triad (Confidentiality, Integrity, Availability, Authenticity, Accountability, and Nonrepudiation).

## Project Description

Quantum Secret Sharing (QSS) protocols aim to securely distribute a secret among multiple participants such that only authorized subsets can reconstruct it. Based on foundational work, QSS has evolved into two primary paradigms: entangled-state-based schemes and mutually unbiased bases (MUB)-

based schemes. While entanglement-based protocols provide strong security, MUB-based approaches offer flexibility and eavesdropping detection through verification states. Despite these advancements, challenges remain in scalability, fault tolerance, privacy, and authentication in dynamic network environments.

**The Proposed System**

Introduces a distributed QSS framework consisting of four main modules: **Player, Dealer, Central Authority, and Network Topology**. The Player registers using an email and password, which are used to generate a unique network topology and weights. Authentication is performed using SHA-256 hashing, ensuring secure login. Player data is encrypted using CRYSTALS-Kyber post-quantum cryptography, generating public and private keys to ensure confidentiality, integrity, and resistance to quantum attacks. The Dealer module enables secure access to encrypted data. Dealer credentials are protected using SHA-256 hashing, and upon successful authentication, the dealer can request data access, download public keys, and retrieve encrypted data after authorization. The Central Authority acts as a control unit responsible for authentication, dealer authorization, monitoring data access, and sending notifications for every data download. This ensures accountability, transparency, and secure data management.

The Network Topology module represents the system as a weighted graph, where nodes correspond to participants and edges represent communication links. A quantum-enhanced Dijkstra's algorithm is used to dynamically select the optimal subset of players and compute the shortest path based on network weights such as distance, delay, or cost. This improves flexibility, performance, and routing efficiency in distributed environments. The system defines clear input-output mappings. For Players, inputs include email, password, and login credentials, while outputs include generated topology, SHA-256 authentication, encrypted data, and CRYSTALS-Kyber key pairs. For Dealers, inputs include credentials and data requests, with outputs including authenticated access, public key retrieval, and secure data downloads. For the Central Authority, inputs include login credentials and authorization requests, while outputs include dealer validation, monitoring updates, and notifications. The proposed algorithm is based on Dijkstra's algorithm, a greedy approach used to compute the shortest path in a weighted graph. It initializes distances, selects the minimum distance node, updates neighboring nodes through relaxation, and repeats until all nodes are visited. This algorithm ensures efficient routing and optimal path selection in the network.

**Existing System**

Existing QSS systems based on GHZ or Bell states suffer from limitations such as high sensitivity to noise, poor scalability, high implementation cost, low transmission efficiency, and strict synchronization requirements. These systems are also vulnerable to practical attacks such as Trojan-horse and detector-based attacks. The proposed system overcomes these limitations by integrating post-quantum cryptography, dynamic routing, and centralized control, providing a secure, scalable, and efficient QSS framework.

**LITERATURE SURVEY**

**TITLE**: "Quantum computing with Qiskit
**AUTHOR**: A. Javadi-Abhari et al.
**YEAR**: 2024
**DESCRIPTION**:

For many practical applications, Quantum computers offer major speed-ups over traditional machines. The production of quantum chips has recently made tremendous strides, with the quantity of Qubits increasing and fidelity increasing. The smallest unit of information is the binary digit i.e. bit in general computing. In Quantum Computing the term Qubit (Quantum Bit) is used for the same purpose. In order to differentiate between a classical bit and a Qubit, Bra-Ket or Dirac notation is used. So, the Qubits are represented as $|0\rangle$ and $|1\rangle$ and are often read as Ket 0 for zero state and Ket 1 for one state. In this paper mainly programming and application sides of Quantum Computing are focused. For that, Qiskit module and its tools are used in Python programming language. General circuit optimization scheme is given. The Quantum circuits with the concept of Quantum Gates implemented are stronger compared to the optimizing algorithm and with faster execution time. In addition, for prime factorization, access to IBM's Quantum computer is carried out successfully.

**TITLE:** 'Practical security bounds against trojan horse attacks in continuous-variable quantum key distribution,
**AUTHOR:** Y. Pan, L. Zhang, and D. Huang
**YEAR:** 2024
**DESCRIPTION:**

In the quantum version of a Trojan-horse attack, photons are injected into the optical modules of a quantum key distribution system in an attempt to read information direct from the encoding devices. To stop the Trojan photons, the use of passive optical components has been suggested. However, to date, there is no quantitative bound that specifies such components in relation to the security of the system. Here, we turn the Trojan-horse attack into an information leakage problem. This allows us to quantify the system security and relate it to the specification of the optical elements. The analysis is

supported by the experimental characterization, within the operation regime, of reflectivity and transmission of the optical components most relevant to security.

**TITLE:** "Challenges in estimating the information capacity of the Fiber-optic channel

**AUTHOR:** M. Shtaif, C. Antonelli, A. Mecozzi, and X. Chen,

**YEAR:** 2023

**DESCRIPTION:**

Since its early commercial deployment in the late 1980s, optical Fiber has evolved to become the predominant carrier of the globe's communications. Yet, after accommodating the world's exponentially growing appetite for transmitted data for more than three decades, its ability to continue doing so is being challenged by fundamental factors. In this article, we review these factors and examine their consequences in terms of information capacity. In particular, we review the difficulties that are imposed by the nonlinear nature of Fiber-optic transmission on the assessment of the capacity and on the definition of fundamental concepts, such as bandwidth and spectral efficiency. We discuss relevant approximations and regimes of operation in which bounds for the capacity can be effectively assessed while covering a broad range of applications ranging from interpatient communications to links spanning transoceanic distances. We relate to a broad variety of transmission schemes and discuss the potential benefits of spatial multiplexing with multimode and multicore fibres. State-of-the-art transmission experiments are also reviewed and compared with theoretical capacity bounds.

**TITLE:** 'Entanglement swapping theory and beyond

**AUTHOR:** Z. Ji, P. Fan, and H. Zhang,

**YEAR:** 2024

**DESCRIPTION:**

While there exist theories that have states more strongly entangled than quantum theory, in the sense that they show Clauser-Horne-Shimony-Holt (CHSH) values above Tsirelson's bound, all known examples of such theories have a strictly smaller set of measurements. Therefore, in tasks that require both bipartite states and measurements, they do not perform better than quantum mechanics. One of the simplest information processing tasks involving both bipartite states and measurements is that of entanglement swapping. In this paper, we study entanglement swapping in generalized probabilistic theories (GPTs). In particular, we introduce the iterated CHSH game, which measures the power of a GPT to preserve nonclassical correlations, in terms of the largest CHSH value obtainable after $n$ rounds of entanglement swapping. Our main result is the construction of a GPT that achieves a CHSH value of 4 after an arbitrary number of rounds.

## METHODOLOGIES

Additional opportunities to extend fairness into more complex network configurations that are not fixed to standard topologies and are developed over real-world operative conditions have arisen.

**MODULES NAME:**

**The modules are**

1.Player
2.Dealer
3.Central Authority
4.Network Topology

**This project having the following modules:**

**1.Players**

The players have a password and email address for their registration. The network topology and network weight will be generated using the email address and password. After entering their user ID and password, the player will receive an SHA-256 and be able to log in. The topology is visible to the player. The players' data is converted into encrypted cryptography, and the public and private keys for Crystal Kyber are created a keys.

**2. Dealer**

The dealer has a password, email address, and name on file. A SHA-256 has been generated by the password. The dealer uses a user ID and password to log in. The dealer can access encrypted data, download a public key, and submit a request. The dealer has a data download.

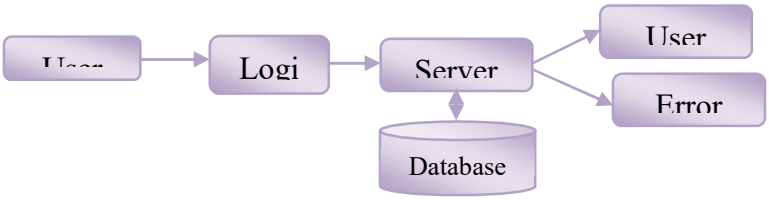
**3. Central Authority**

A user ID and password are required to log in to the central authority. A dealer is authorized by the central authorities. Dealer data from the central authority. A notification of a data download has been sent to the central authority.

## MODULES EXPLANATION AND DIAGRAM

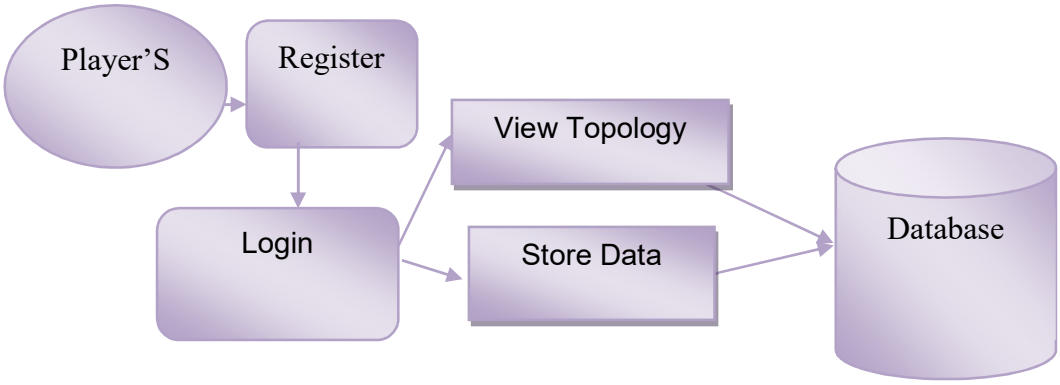
**User Interface Design**

To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password, Email id, City and Country into the server. Database will create the account for the entire user to maintain upload and download rate. Name will be set as user id. Logging in is usually used to enter a specific page. It will search the query and display the query.



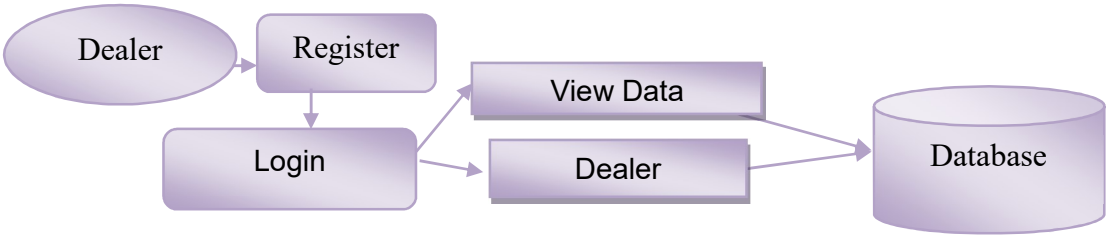

**Player's**

Players register using an email address and password, which generate a unique network topology and network weights. Authentication is performed using SHA-256 hashing, allowing secure login and access to the topology. Player data is encrypted before storage or transmission using CRYSTALS-Kyber, a post-quantum cryptographic algorithm that generates public and private key pairs. This ensures confidentiality, integrity, and resistance to quantum attacks, providing a secure authentication and encryption framework for accessing the system.




**Dealer**

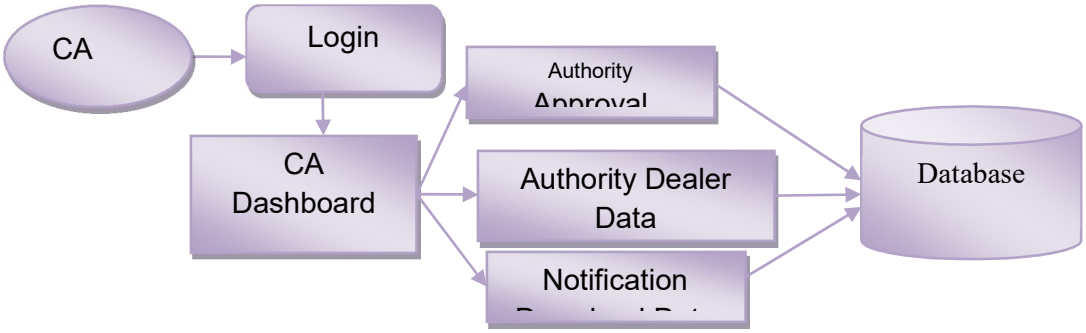
The dealer is a registered user with name, email, and password securely stored using SHA-256 hashing, ensuring credentials are not stored in plain text. During login, the system verifies the SHA-256 hash to authenticate the dealer and grant access to the dashboard. The dealer can access encrypted data, download public keys for secure communication, and submit data access requests. Upon approval, the dealer can securely download data, ensuring confidentiality, integrity, and controlled data handling throughout system interaction.




**Central Authority**

The Central Authority uses user ID and password-based authentication to ensure secure access and acts as the primary control unit for monitoring users, managing permissions, and maintaining system security. It authorizes dealers, ensuring only legitimate users can access or distribute data, thereby maintaining controlled data flow. The Central Authority also monitors all data access activities and receives notifications for every data download, enabling transparency, auditing, and detection of suspicious behaviour. This ensures accountability,

secure data management, and effective system monitoring




.

**Network Topology**

The protocol enhances flexibility and performance by dynamically selecting the optimal subset of players using a quantum version of Dijkstra's algorithm and a custom weighting system. The network is modeled as a weighted graph where nodes represent devices and edges represent links with weights such as distance, cost, or delay. Dijkstra's algorithm initializes the source node with distance zero and others with infinity, then iteratively selects the nearest unvisited node and updates neighbouring distances. This process continues until all nodes are visited, resulting in the shortest path and minimum weight for efficient routing and network optimization.

**Given Input–Expected Output**

For the **Player**, inputs include email address, password, and login credentials, resulting in network topology and weights generation, SHA-256 authentication, login access, topology display, encrypted data, and CRYSTALS-Kyber public/private key pair generation. For the **Dealer**, inputs include name, email, password, login credentials, and data access request, producing SHA-256 authentication, access to encrypted data, public key download, request submission to Central Authority, and secure data download after approval. For the **Central Authority**, inputs include login credentials, dealer authorization request, and data download events, leading to successful login, dealer authorization (approved/rejected), secure data management, notification generation for downloads, and system audit and monitoring updates

**Proposed and Existing Algorithms**

The proposed system utilizes **Dijkstra's algorithm**, a greedy algorithm used to compute the shortest path in a weighted graph from a source node to all other nodes. The network is modeled as vertices (nodes) and edges (links) with associated weights such as distance, delay, or cost. The algorithm initializes the source node with distance zero and all others with infinity, then repeatedly selects the nearest unvisited node and updates distances of its neighbouring nodes through relaxation. This process continues until all nodes are visited, ensuring optimal path selection for efficient routing and network optimization. Due to its simplicity and efficiency, Dijkstra's algorithm is widely applied in communication networks and routing systems.

In contrast, existing Quantum Secret Sharing (QSS) algorithms based on **GHZ or Bell states** provide strong theoretical security but suffer from practical limitations. These include high sensitivity to noise and decoherence, poor scalability due to multi-particle entanglement complexity, and high implementation cost requiring specialized quantum hardware. Additionally, these systems experience low transmission efficiency, vulnerability to attacks such as Trojan-horse and detector-based attacks, and strict synchronization requirements among participants. These challenges limit their real-world applicability.

The proposed approach overcomes these limitations by combining efficient shortest-path routing with improved flexibility and scalability, providing a more practical and robust solution for secure data sharing in distributed environments.

**REQUIREMENTS ENGINEERING**

Recent advancements in Quantum Secret Sharing (QSS) include trusted third-party systems for identity verification and entanglement distribution, as well as security mechanisms for detecting eavesdropping in n-qubit systems. However, further research is required to improve scalability and deployment through decentralized architectures and fault-tolerant mechanisms. The system requires standard hardware including an Intel Core i3 processor or higher, 4 GB RAM, 100 GB storage, and basic peripherals such as keyboard, mouse, and webcam. The software environment includes Java EE (JSP, Servlets) for the front-end, MySQL for the backend, Apache Tomcat as

the web server, and Windows OS with browser support.

The functional requirements are divided into four modules. The Player module supports registration, SHA-256 authentication, network topology generation, and data encryption using CRYSTALS-Kyber public/private keys. The Dealer module enables secure login, access to encrypted data, public key download, request submission, and authorized data retrieval. The Central Authority module manages authentication, dealer authorization, monitoring, and notification of data access activities. The Network Topology module represents the system as a weighted graph and dynamically computes optimal routing paths using a quantum-enhanced Dijkstra's algorithm, improving performance and flexibility.

## SOFTWARE SPECIFICATION (JAVA FEATURES)

Java is a general-purpose, object-oriented programming language developed by James Gosling and released as part of Sun Microsystems. It follows the principle of "write once, run anywhere" by compiling code into bytecode that executes on the Java Virtual Machine (JVM), ensuring platform independence, portability, and minimal implementation dependency. Java's simplicity, security, and scalability make it widely used in applications ranging from web systems to enterprise and distributed environments.

Java provides key object-oriented features such as inheritance, encapsulation, polymorphism, and dynamic binding, enabling modular, reusable, and flexible program design. It supports concurrent programming and multithreading, allowing efficient execution of multiple tasks simultaneously. Java Swing and AWT frameworks provide graphical user interface components, where Swing offers lightweight, platform-independent UI elements for building interactive applications.

The Java Collection Framework simplifies data handling through structures such as Lists, Sets, and Maps. Lists maintain order and allow duplicates, Sets prevent duplicates, and Maps store key-value pairs for efficient data access. These collections enhance performance and flexibility in managing large datasets.

Advanced Java technologies support enterprise and web development. Servlets and Java Server Pages (JSP) enable server-side processing and dynamic content generation, while JDBC facilitates database connectivity and query execution. JavaBeans provide reusable components, and the MVC architecture separates logic, presentation, and control. Additional features such as filters, listeners, and session management improve request handling and user interaction. Apache Tomcat serves as a lightweight web server and servlet container for deploying Java applications.

Java's strong database support through JDBC and ORM frameworks, combined with its portability, multithreading, and security features, makes it highly suitable for building scalable, reliable, and secure database-driven applications in both local and cloud environments.

**UnitTesting:**

Validates individual software units by checking internal logic, decision branches, and input-output correctness. It ensures each component performs according to specifications before integration.

**FunctionalTesting:**

Verifies system functions against requirements by testing valid/invalid inputs, expected outputs, and interaction with system procedures.

**SystemTesting:**

Evaluates the complete integrated system to ensure it meets requirements and produces predictable results based on defined workflows.

**PerformanceTesting:**

Measures system efficiency by ensuring fast response time, minimal processing delay, and timely output generation under different conditions.

**IntegrationTesting:**

Tests interaction between multiple components to detect interface defects and ensure smooth communication between modules.

**AcceptanceTesting:**

Ensures the system meets user requirements through end-user validation. It includes data synchronization checks such as acknowledgements, route operations, and node status updates.

**TestPlan:**

Defines a structured testing strategy by dividing the system into units, identifying defects early, and ensuring reliable and error-free system performance.

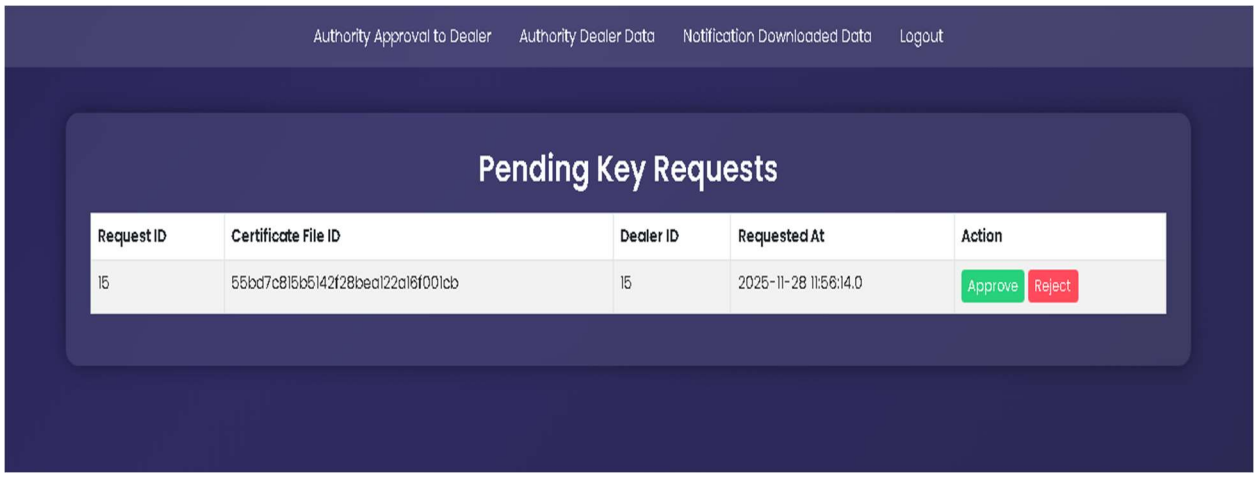
To address these opportunities, our work introduces a novel QSS protocol designed for distributed quantum networks. By leveraging a quantum Dijkstra algorithm and a flexible weighting system, it enables dynamic participant selection and adaptive routing, ensuring fault tolerance and scalability.

```java
package Algorithms;

import java.security.KeyPair;

public class PublicKeyGenerator {
    public static String generateRsaPublicKeyBase64() {
        try {
            KeyPairGenerator keyGen = KeyPairGenerator.getInstance("RSA");
            keyGen.initialize(2048);

            KeyPair pair = keyGen.generateKeyPair();
            PublicKey publicKey = pair.getPublic();

            return Base64.getEncoder()
                         .encodeToString(publicKey.getEncoded());

        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

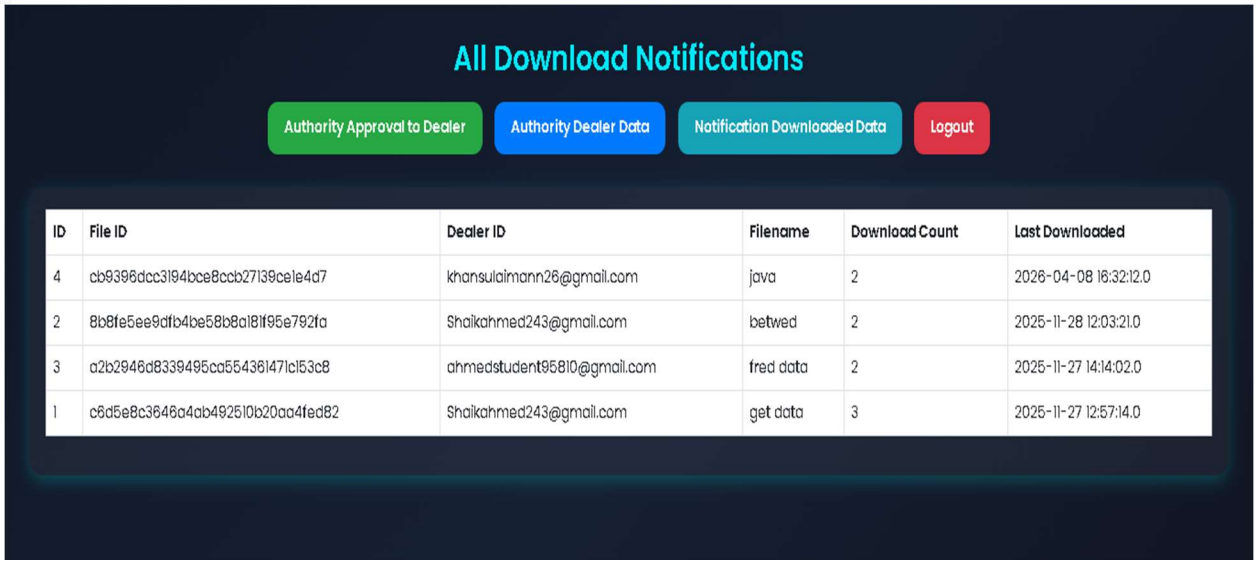
***ScreenShot 1 Algorithm***

**Results:**


***ScreenShot 2 HomePage***




***ScreenShot: 3 Request Panel***




| ID | File ID | Dealer ID | Filename | Download Count | Last Downloaded |
|---|---|---|---|---|---|
| 4 | cb9396dcc3194bce8ccb27139ce1e4d7 | khansulaimann26@gmail.com | java | 2 | 2026-04-08 16:32:12.0 |
| 2 | 8b8fe5ee9dfb4be58b8a181f95e792fa | Shaikahmed243@gmail.com | betwed | 2 | 2025-11-28 12:03:21.0 |
| 3 | a2b2946d8339495ca55436147lcl53c8 | ahmedstudent95810@gmail.com | fred data | 2 | 2025-11-27 14:14:02.0 |
| 1 | c8d5e8c3646a4ab492510b20aa4fed82 | Shaikahmed243@gmail.com | get data | 3 | 2025-11-27 12:57:14.0 |

***ScreenShot: 4 Notification Page***

**Application and Future Enhancement (Summarized):**

The proposed QSS framework is applicable in secure cloud storage, distributed networks, and quantum communication systems requiring confidentiality, authentication, and efficient routing. Future work focuses on strengthening post-quantum cryptography, particularly CRYSTALS-Kyber based on the LWE problem, while exploring alternative secure methods against potential quantum breakthroughs. It also includes developing fully quantum-native algorithms, improving scalability and fault tolerance, and promoting wider adoption of quantum cryptographic solutions for long-term data security.

**Conclusion**

This work presents a novel entanglement-based QSS protocol that enhances flexibility, security, and resilience in quantum networks. It prevents collusion by hiding network structure and uses a quantum Dijkstra algorithm for efficient selection of *t* participants. Integration of CRYSTALS-Kyber strengthens authentication and protects against identity threats, while the *(t, n)* scheme ensures fairness and data integrity under the extended CIA Triad. Simulation results confirm practical viability, despite current hardware limitations, and demonstrate strong potential for real-world deployment. Overall, the protocol provides a secure, scalable, and future-ready approach for quantum communication systems.

**Reference**

[1] G. R. Blakley, “Safeguarding cryptographic keys,” in Proc. Int. Workshop Manage. Requirements Knowl., 1979, pp. 313–318, doi: 10.1109/AFIPS.1979.98.

[2] A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979, doi: 10.1145/359168.359176.

[3] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” SIAM J. Comput., vol. 26, no. 5, pp. 1484–1509, Oct. 1997, doi: 10.1137/S0097539795293172.

[4] C. Lu, F. Miao, J. Hou,W. Huang, and Y. Xiong, “A verifiable framework of entanglement-free quantum secret sharing with information-theoretical security,” Quantum Inf. Process., vol. 19, no. 1, 2019, Art. no. 24, doi: 10.1007/s11128-019-2509-x.

[5] B. C. Hiesmayr, “Free versus bound entanglement, a NP-hard problem tackled by machine learning,” Sci. Rep., vol. 11, no. 1, 2021, Art. no. 19739, doi: 10.1038/s41598-021-98523-6.

[6] R. van Houte, J. Mulderij, T. Attema, I. Chiscop, and F. Phillipson, “Mathematical formulation of quantum circuit design problems in networks of quantum computers,” Quantum Inf. Process., vol. 19, no. 5, 2020, Art. no. 141, doi: 10.1007/s11128-020-02630-8.

[7] F. Liu, S.-J. Qin, and Q.-Y. Wen, “A quantum secret-sharing protocol with fairness,” Phys. Scripta, vol. 89, no. 7, 2014, Art. no. 075104, doi: 10.1088/0031-8949/89/7/075104.

[8] M. Hillery, V. Bužek, and A. Berthiaume, “Quantum secret sharing,” Phys. Rev. A, vol. 59, no. 3, pp. 1829–1834, 1999, doi: 10.1103/PhysRevA.59.1829.

[9] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond Bell's theorem," 2007, arXiv:0712.0921, doi: 10.48550/arXiv.0712.0921.
[10] H.-Y. Jia, Q.-Y.Wen, F. Gao, S.-J. Qin, and F.-Z. Guo, "Dynamic quantum secret sharing," Phys. Lett. A, vol. 376, nos. 10/11, pp. 1035–1041, 2012, doi: 10.1007/s11128-012-0380-0.
[11] C.-H. Liao, C.-W. Yang, and T. Hwang, "Dynamic quantum secret sharing protocol based on GHZ state," Quantum Inf. Process., vol. 13, no. 8, pp. 1907–1916, 2014, doi: 10.1007/s11128-014-0779-x.
[12] J. L. Hsu, S. K. Chong, T. Hwang, and C. W. Tsai, "Dynamic quantum secret sharing," Quantum Inf. Process., vol. 12, no. 1, pp. 331–344, 2013, doi: 10.1007/s11128-012-0380-0.
[13] S. Mishra, C. Shukla, A. Pathak, R. Srikanth, and A. Venugopalan, "An integrated hierarchical dynamic quantum secret sharing protocol," Int. J. Theor. Phys., vol. 54, no. 9, pp. 3143–3154, 2015, doi: 10.1007/s10773-015-2552-z.
[14] H. Qin and Y. Dai, "Dynamic quantum secret sharing by using ddimensional GHZ state," Quantum Inf. Process., vol. 16, no. 3, 2017, Art. no. 64, doi: 10.1007/s11128-017-1525-y.
[15] Z. You, Y. Wang, Z. Dou, J. Li, X. Chen, and L. Li, "Dynamic quantum secret sharing between multiparty and multiparty based on single photons," Phys. A, Statist. Mech. Appl., vol. 624, 2023, Art. no. 128893, doi: 10.1016/j.physa.2023.128893.
[16] P. Priyanka, V. Siwach, and P. Bijaranian, "Quantum secret sharing with (m, n) threshold: QFT and identity authentication," Quantum Inf. Process., vol. 23, 2024, Art. no. 348, doi: 10.1007/s11128-024-04532-5.
[17] N. Zhou, Z. Chen, Y. Liu, and L. Gong, "Multi-party semi-quantum private comparison protocol of size relation with d-level GHz states," Adv. Quantum Technol., 2024, doi: 10.1002/qute.202400530.
[18] L. Gong, M. Li, H. Cao, and B. Wang, "Novel semi-quantum private comparison protocol with Bell states," Laser Phys. Lett., vol. 21, no. 5, 2024, Art. no. 055209, doi: 10.1088/1612-202X/ad3a54.
[19] W. Hu, R.-G. Zhou, X. Li, P. Fan, and C. Tan, "A novel dynamic quantum secret sharing in high-dimensional quantum system," Quantum Inf. Process., vol. 20, no. 5, 2021, Art. no. 159, doi: 10.1007/s11128-021-03103-2.
[20] Y. Song, Z. Li, and Y. Li, "A dynamic multiparty quantum direct secret sharing based on generalized GHZ states," Quantum Inf. Process., vol. 17, no. 9, 2018, Art. no. 244, doi: 10.1007/s11128-018-1970-2.
[21] C.-W. Yang and C.-W. Tsai, "Improved dynamic multiparty quantum direct secret sharing protocol based on generalized GHZ states to prevent collusion attack," Modern Phys. Lett. A, vol. 35, no. 8, 2020, Art. no. 2050040, doi: 10.1142/S0217732320500406.
[22] Y. Tian, J. Wang, G. Bian, J. Chang, and J. Li, "Dynamic multiparty to multi-party quantum secret sharing based on Bell states," Adv. Quantum Technol., vol. 7, no. 7, 2024, Art. no. 2400116, doi: 10.1002/qute.202400116.
[23] Y. Kang, Y. Guo, H. Zhong, G. Chen, and X. Jing, "Continuous variable quantum secret sharing with fairness," Appl. Sci., vol. 10, no. 1, 2020, Art. no. 189, doi: 10.3390/app10010189.
[24] X. Li, K. Zhang, L. Zhang, and X. Zhao, "A new quantum multiparty simultaneous identity authentication protocol with the classical third-party," Entropy, vol. 24, no. 4, 2022, Art. no. 483, doi: 10.3390/e24040483.
[25] S. Schauer, M. Huber, and B. C. Hiesmayr, "Experimentally feasible security check for n-qubit quantum secret sharing," Phys. Rev. A, vol. 82, no. 6, 2010, Art. no. 062311, doi: 10.1103/PhysRevA.82.062311.
[26] G. Gao, C.-C. Wei, and D. Wang, "Cryptanalysis and improvement of dynamic quantum secret sharing protocol based on two-particle transform of Bell states," Quantum Inf. Process., vol. 18, no. 6, 2019, Art. no. 186, doi: 10.1007/s11128-019-2301-y.
[27] R. Golden and I. Cho, "On symmetric polynomials," 2015, arXiv:1511.08870, doi: 10.48550/arXiv.1511.08870.
[28] K. Thas, "The geometry of generalized Pauli operators of N-qudit Hilbert space, and an application to MUBs," Europhys. Lett., vol. 86, no. 6, Jul. 2009, Art. no. 60005, doi: 10.1209/0295-5075/86/60005.
[29] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," Theor. Comput. Sci., vol. 560, pp. 7–11, 2014, doi: 10.1016/j.tcs.2014.05.025.
[30] F. Li, T. Chen, and S. Zhu, "A (t,n) threshold quantum secret sharing scheme with fairness," Int. J. Theor. Phys., vol. 62, no. 6, 2023, Art. no. 119, doi: 10.1007/s10773-023-05383-z.
[31] W. Liu, Q. Wu, J. Shen, J. Zhao, M. Zidan, and L. Tong, "An optimized quantum minimum searching algorithm with sure-success probability and its experiment simulation with Cirq," J. Ambient Intell. Humanized Comput., vol. 12, no. 11, pp. 10425–10434, Jan. 2021, doi: 10.1007/s12652-020-02840-z.
[32] G. L. Long, "Grover algorithm with zero theoretical failure rate," Phys. Rev. A, vol. 64, no. 2, 2001, Art. no. 022307, doi: 10.1103/Phys-RevA.64.022307.
[33] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numer. Math., vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/BF01386390.

[34] Z. Ji, P. Fan, and H. Zhang, "Entanglement swapping theory and beyond," 2022, arXiv:2009.02555, doi: 10.48550/arXiv.2009.02555.
[35] X. Gitiaux, I. Morris, M. Emelianenko, and M. Tian, "SWAP test for an arbitrary number of quantum states," Quantum Inf. Process., vol. 21, 2021, Art. no. 344, doi: 10.1007/s11128-022-03643-1.
[36] M. Shtaif, C. Antonelli, A. Mecozzi, and X. Chen, "Challenges in estimating the information capacity of the fiber-optic channel," Proc. IEEE, vol. 110, no. 11, pp. 1655–1678, Nov. 2022, doi: 10.1109/JPROC.2022.3197188.
[37] J. Bos et al., "CRYSTALS—Kyber: A CCA-secure module-lattice-based KEM," in Proc. IEEE Eur. Symp. Secur. Privacy, 2018, pp. 353–367, doi: 10.1109/EuroSP.2018.00032.
[38] A. Javadi-Abhari et al., "Quantum computing with Qiskit," 2024, arXiv:2405.08810, doi: 10.48550/arXiv.2405.08810