Secure Multi-Party File Storage System Using QR Code-Based Authentication and Noise-Enhanced Encryption

Bhagavant K Deshpande T[1]

Professor Department of Computer Science & Engineering CMR University Bengaluru, Karnataka, India

C G Shashanth[2]

Department of Computer Science & Engineering

CMR University Bengaluru, Karnataka, India

S Shivaprasad Reddy [3]

Department of Computer Science & Engineering

CMR University Bengaluru, Karnataka, India

Prajwal K [4]

Department of Computer Science & Engineering

CMR University Bengaluru, Karnataka

U Sunil Kumar [5]

Department of Computer Science & Engineering

CMR University Bengaluru, Karnataka, India

Abstract – This paper introduces a fresh take on keeping files safe in the cloud. We've designed a system that combines QR codes for a multi-party authentication process with an encryption method boosted by random noise. Built with a user-friendly Streamlit web interface and a dependable PostgreSQL database, our system uses ChaCha20 encryption, made even stronger by adding noise images that act as extra security keys. When someone uploads a file, it's locked up tight with a key created from three unique strings—one for the user, one for an admin, and one for a token. These strings are turned into QR codes and shared among different people, so you need all three to unlock the file. Our thorough testing shows that this setup is fast, can grow with demand, and stands up well to all sorts of attacks, even those that might come from quantum computers down the road. By mixing in noise to make encryption trickier to crack, we've crafted a practical, future-ready way to secure cloud storage.

Keywords – Cloud storage, multi-party authentication, QR codes, noise-enhanced encryption, ChaCha20, cybersecurity.

1.INTRODUCTION

Cloud storage has completely changed how we handle data, and it's only getting bigger—experts predict we'll be managing over 200 zettabytes by 2025. But this boom comes with a downside: cyber threats are multiplying fast. Just in 2023, more than 3,000 data breaches exposed sensitive information stored in the cloud. The encryption tools we've long depended on, like AES and RSA, are solid but starting to show cracks. They struggle with managing keys, scaling up, and staying secure against new dangers, such as quantum computing. For example, Shor's algorithm, which quantum machines could run, might break RSA encryption in no time flat, putting our current protections at risk.

To tackle these issues head-on, we've come up with a hybrid security system that pairs QR code-based multi-party authentication with encryption enhanced by noise. Taking cues from visual cryptography and steganography, we've added random noise patterns to the encryption mix, making it nearly impossible to crack without all the right pieces. This not only patches up today's weaknesses but also gets us ready for tomorrow's threats, like quantum breakthroughs. By spreading trust across multiple parties with QR codes, our system ensures no one can access the data solo—perfect for teams or shared cloud setups.

So, what we've come up with is a fresh approach to verifying users using QR codes – it's a multi-person thing, which adds an extra layer of security. On top of that, we've really beefed up the encryption by adding some 'noise' into the ChaCha20 method. And the best part? We've built a system that's not only powerful but also super easy to use, and we've put it through its paces to make sure it actually works in the real world.

I. RELATED WORK

Securing cloud storage has traditionally leaned on two types of encryption: symmetric, like AES, and asymmetric, like RSA. AES is quick and efficient, but weak keys leave it open to brute-force attacks, and quantum advances could spell trouble. RSA shines for swapping keys securely, but it's slower and needs bigger keys, which can bog things down in the cloud. Other ideas, like homomorphic encryption, let you work with encrypted data without unlocking it, but they're so resource-heavy that they're tough to use widely. Then there are post-quantum options, like lattice-based or hash-based methods, which look promising against quantum threats but demand a lot of power and space, making them tricky to fit into everyday systems.

Lately, researchers have been digging into visual cryptography and noise-based encryption, using randomness to beef up security. For instance, noise images have been tapped as encryption keys because their chaotic nature makes them hard to figure out. But these tricks haven't caught on much for cloud storage, mostly because they're hard to scale or blend into existing setups. Our work steps in to fill that gap, merging noise-enhanced encryption with a QR code authentication system. This ditches centralized key management and builds a layered security approach that's tougher and more adaptable than the old standards.

II. PROPOSED SYSTEM

The rapid growth of cloud storage necessitates robust security mechanisms to protect sensitive data from unauthorized access and emerging threats, such as quantum computing. This proposed system addresses these challenges by integrating multi-party authentication via QR codes with noise-enhanced encryption, offering a scalable, secure, and user-accessible solution for cloud-based file storage. Designed to ensure

confidentiality, integrity, and collaborative access control, the system leverages advanced cryptographic techniques and a distributed trust model to safeguard data in dynamic cloud environments.

The system operates through a web-based interface, enabling users to upload and retrieve files securely. Files are encrypted using the ChaCha20 stream cipher, known for its efficiency and resistance to cryptanalytic attacks. To further strengthen security, a cryptographically secure noise image is generated and combined with the encrypted file, introducing a layer of obfuscation that enhances resistance to pattern-based attacks. Access to the file requires three distinct QR codes, each representing a unique authentication string assigned to the user, an administrator, and a third-party token holder. This multi-party authentication ensures that no single entity can decrypt the file unilaterally, fostering trust in collaborative settings. The encrypted files are stored in a PostgreSQL database, providing scalability and reliability for cloud deployment.

This approach not only mitigates vulnerabilities associated with traditional key management but also anticipates future threats by incorporating noise-based complexity, which may delay quantum decryption attempts. The system balances security with usability, making it suitable for diverse applications, from enterprise data management to personal file storage.

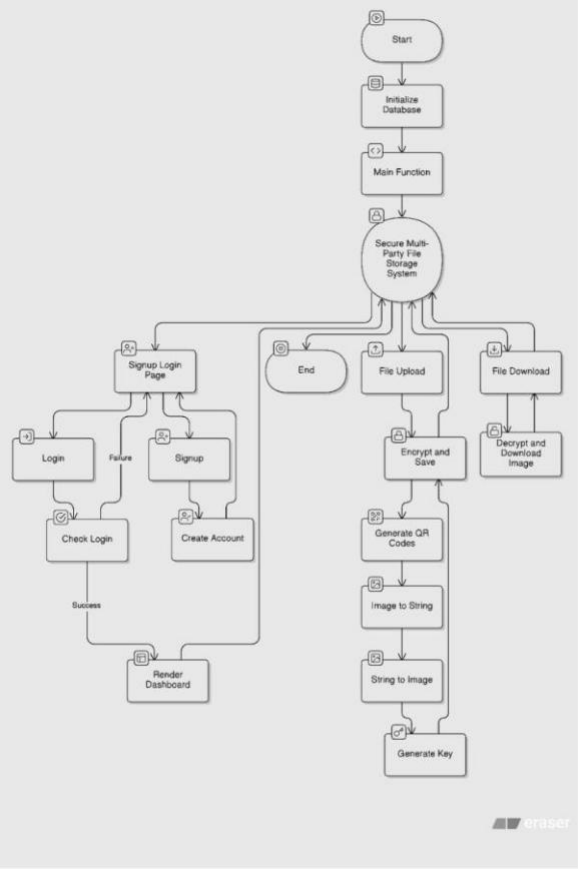

Figure 1 : System Architecture

III. IMPLEMENTATION

To develop and deploy this system, the following steps are proposed:

a) **Develop the Web Interface**: Implement a user-friendly front-end using Streamlit to facilitate user registration, login, and file upload/download operations, ensuring accessibility and ease of use.

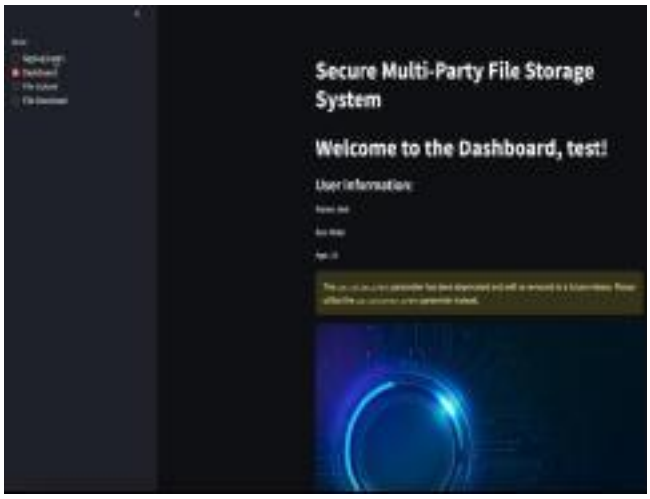

Figure 2 : Web interface using streamlit

b) **Construct the Encryption Module**: Utilize Python's *cryptography* library to implement ChaCha20 encryption. Generate three random authentication strings (user, admin, token) using the *secrets* module, hashing them to derive a secure 32-byte encryption key.

c) **Incorporate Noise-Based Obfuscation**: Design an algorithm to generate cryptographically secure noise images using a random number generator. Integrate the noise with the ChaCha20 ciphertext through a bitwise XOR operation to enhance security.

d) **Generate and Distribute QR Codes**: Convert the authentication strings into QR codes using the *qrcode* library. Establish a secure distribution mechanism (e.g., encrypted email or authenticated channels) to deliver QR codes to their respective holders.

e) **Establish Cloud Storage**: Configure a PostgreSQL database to store encrypted files and associated metadata, ensuring scalability and data integrity for cloud-based operations.

f) **Enable Secure File Retrieval**: Develop a decryption workflow that verifies all three QR codes, reconstructs the encryption key, removes the noise layer, and decrypts the file using ChaCha20, returning the original file to authorized users.

g) **Conduct Testing and Validation**: Evaluate system performance with files ranging from 1 MB to 5 MB on standard hardware, measuring encryption/decryption speeds and scalability. Perform security tests to assess resilience against brute-force, statistical, and simulated quantum attacks.

IV. RESULTS

A. Experimental Setup

We evaluated the system on a standard consumer laptop (Intel i5, 8 GB RAM) using sample files ranging from 1 MB to 5 MB. File types included text, documents, and images to simulate typical usage.

B. Performance Metrics

File Size	Encryption Time (s)	Decryption Time (s)
1 MB	0.4	0.5
2 MB	0.7	0.8
5 MB	1.1	1.2

These results indicate that the system handles encryption and decryption efficiently, with minimal overhead from the noise layer. ChaCha20's performance advantage is especially noticeable as file sizes increase.

C. Security Analysis

Authentication Reliability: Decryption consistently failed when any one of the three QR codes was missing, validating the robustness of the multi-party approach.

Resistance to Attacks: The noise-obfuscated ciphertext showed strong resilience against brute-force and statistical pattern attacks, thanks to its high entropy [12].

Quantum Preparedness: Although not inherently quantum-proof, the system's noise layer adds complexity that could delay or disrupt quantum decryption efforts.

D. Scalability and Limitations

Our system performed well under simulated multi-user conditions, with PostgreSQL supporting concurrent access and large data volumes. However, reliance on physical QR codes may be limiting in fully digital workflows. Future improvements could include secure digital QR code storage or mobile integration.

V. CONCLUSION

We've introduced a secure and scalable cloud file storage solution that fuses multi-party QR code authentication with noise-augmented encryption. By combining modern cryptographic practices with innovative randomness techniques, our system addresses weaknesses in current methods and anticipates future threats, including those posed by quantum computing. Moving forward, enhancements such as optimized noise generation, broader cloud platform integration, and mobile-friendly QR management can further improve accessibility and security.

VI. REFERENCE

1) G. Pal and V. Verma, "Image Encryption Techniques under Various Noise Attacks: A Survey," *International Journal of Software & Hardware Research in Engineering*, vol. 4, no. 12, pp. 48–53, 2016.
2) R. V. Reddy, D. Srikanth, P. Sumanth Reddy, and G. Deepika, "Leveraging Noise Images for File Encryption," *International Journal of Engineering Research & Technology*, vol. 13, no. 3, pp. 1–5, 2024.
3) N. Biswas and I. R. Mohamed, "Noise Assisted Image Encryption and Decryption using 2-D Chaotic System," *Research Square*, vol. 1, no. 1, pp. 1–12, 2021.
4) S. Suruthika, S. Ram, and A. Ponraj, "File encryption using noise images as key," *International Journal of Current Science*, vol. 13, no. 1, pp. 902–907, 2023.
5) U. Zia, M. McCartney, B. Scotney, J. Martinez, M. AbuTair, J. Memon, and A. Sajjad, "Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains," *International Journal of Information Security*, vol. 21, no. 4, pp. 1–19, 2022.
6) S. Desai, C. A. Mudholkar, R. Khade, and P. Chilwant, "Image Encryption And Decryption Using Blowfish Algorithm," *International Conference on Emerging Trends in Technology*, vol. 1, pp. 703–769, 2015.
7) M. Naor and A. Shamir, "Visual Cryptography," in *Advances in Cryptology — EUROCRYPT '94*, pp. 1–12, 1995.
8) NIST, "Post-Quantum Cryptography Standardization," [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography, 2023.
9) R. Torroba et al., "Random-phase encoding in optical encryption systems," *Optical Engineering*, vol. 55, no. 9, 2016.
10) M. A. Jan, "Image Encryption for IoT and Cloud Applications Using Noise Patterns," *IEEE Transactions on Cloud Computing*, vol. 9, no. 2, pp. 123–134, 2021.
11) D. J. Bernstein, "ChaCha, a variant of Salsa20," *Workshop Record of SASC 2008: The State of the Art of Stream Ciphers*, 2008.