International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Authcloudfile: Encrypted File Storage with QR-Based Access and Expiration Control

[1]Mr. R. Ramakrishnan , [2]G. Sowmiya

[1]Associate Professor, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India

[2]Post Graduate Student, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India

*Corresponding author's email address: sowmiguna08@gmail.com

Abstract: File sharing is a fundamental aspect of digital communication, enabling users to exchange data efficiently across networks. However, traditional file-sharing methods, such as email attachments, cloud storage links, and peer-to-peer transfers, often face security vulnerabilities, including unauthorized access, data interception, and weak encryption mechanisms. Many existing systems lack robust authentication or rely on manual key exchange, making them susceptible to attacks such as man-in- the-middle (MITM), phishing, and unauthorized data breaches. To overcome these challenges, the project integrates AES encryption, Two-Factor Authentication (2FA), and Reed-Solomon error correction to ensure confidentiality, integrity, and reliability. The system requires users to authenticate via 2FA (username, password, and OTP verification) before uploading files. Each uploaded file undergoes AES-256 encryption, ensuring that only authorized recipients can decrypt it. A QR code is generated, embedding a secure file access link while incorporating Reed- Solomon error correction for improved scanning accuracy.

The system enhances security by implementing QR code expiration and scan limits. QR codes are valid only for a specific duration and automatically expire after a set number of scans, preventing unauthorized access beyond the intended recipient. Upon successful authentication, the recipient scans the QR code to retrieve and decrypt the file securely. Unlike conventional methods, this approach eliminates the need for manual key exchange, as AES encryption keys are securely managed within the system, ensuring seamless and protected file access. By integrating strong encryption, multi- factor authentication, and QR-based access control, the project enhances data security and usability, providing an efficient and reliable solution for modern file-sharing applications.

Introduction

In today's digital era, the demand for secure and convenient file sharing is more critical than ever. With increasing concerns over data breaches, unauthorized access, and cyberattacks, conventional file-sharing methods such as email attachments, open cloud storage links, or peer-to-peer networks have become insufficient in ensuring data privacy and integrity. These traditional platforms often lack strong encryption, have weak or optional authentication mechanisms, and expose users to threats like phishing, man-in-the-middle (MITM) attacks, and file tampering. As data security becomes a priority in personal, corporate, and institutional environments, there is a growing need for advanced, reliable, and user-friendly file-sharing solutions that address the vulnerabilities of existing systems.

The AuthCloudFile project is designed as a secure file storage and sharing platform that leverages modern encryption, access control, and authentication technologies to overcome these limitations. It integrates **AES-256 encryption**, a widely trusted standard known for its strength and performance, to protect the confidentiality of files before they are uploaded to the system. Users are required to complete Two- Factor Authentication (2FA), which involves entering their login credentials followed by a time-sensitive One-Time Password (OTP). This additional verification step strengthens account security by ensuring that access is granted only to verified users, even in cases where login details might be exposed.A key innovation of the system is its **QR-based file access mechanism**. Upon successful encryption and upload, a **QR code** is generated that encodes a secure access link to the

file. This method eliminates the need for manual key exchange or insecure URL sharing, providing a seamless and secure way to deliver files to authorized recipients. To further enhance security and usability, each QR code is embedded with **Reed- Solomon error correction**, ensuring accurate scanning even in partially damaged or low-quality print conditions. Additionally, the system enforces **expiration times** and **scan limits** on each QR code, so that access is only valid for a specific time window and number of uses, reducing the chances of data misuse or leak after the initial sharing.

The backend system securely manages encryption keys, user data, and access logs without exposing sensitive information to third parties. The file decryption process occurs only after verifying the QR code, checking expiration status, and ensuring the scan count has not been exceeded. This integrated workflow enhances not only **security and access control** but also improves **usability**, making the platform suitable for a wide range of applications from personal file sharing to secure corporate document delivery.

Overall, AuthCloudFile aims to bridge the gap between usability and robust security in file sharing. It brings together cutting-edge encryption, modern access control techniques, and user-centric design to provide a secure, efficient, and scalable solution for protecting sensitive digital content. This approach addresses modern security challenges and presents a practical alternative to conventional file-sharing systems, with potential for further enhancements like biometric login, blockchain auditing, and cloud integration in future iterations.

Literature Survey

In recent years, secure file sharing and cloud storage have become critical areas of research due to growing concerns over data privacy and cyber threats. Traditional file- sharing methods such as email attachments, public cloud links, and peer-to-peer transfers often lack strong security measures and are vulnerable to unauthorized access, data interception, and man-in-the-middle attacks. Various researchers have attempted to address these challenges through encryption and authentication techniques. For instance, Goyal et al. (2016) proposed a file storage system using AES encryption, highlighting the need for data confidentiality. However, their approach relied on manual key sharing, increasing the risk of key leakage. Patil and Waghmare (2018) introduced a multi-server encrypted storage model, offering redundancy but lacking strong user authentication or time-bound access control.

Kumar et al. (2019) explored the use of QR codes in secure systems, demonstrating how encrypted links can be embedded in QR codes for file access, but their work did not address access restrictions such as scan limits or expiration. Jain et al. (2020) focused on improving QR code reliability using Reed-Solomon error correction, which enhances scanning accuracy but was primarily applied in logistics rather than secure file sharing.

Furthermore, Sahai et al. (2021) emphasized the effectiveness of Two-Factor Authentication (2FA) using One-Time Passwords (OTP) in reducing unauthorized access, though their research did not integrate it into a complete file-sharing framework. Sharma and Verma (2022) proposed a cloud-based encrypted document sharing system with basic 2FA, yet it lacked features like QR-based delivery, automatic expiration, and internal key management. Despite these contributions, existing systems often fail to combine encryption, authentication, and user-friendly access controls in a unified platform. The proposed project, AuthCloudFile, bridges this gap by integrating AES-256 encryption, OTP-based 2FA, QR-based file access with scan and time restrictions, and error-tolerant QR codes using Reed-Solomon correction. This combination enhances security, usability, and control, providing a robust solution for modern secure file sharing needs.

Proposed System

The proposed system, **AuthCloudFile**, is designed to address the limitations and security vulnerabilities in conventional file- sharing platforms by integrating multi- layered security mechanisms with user-friendly access controls. The system initiates with a strong authentication process that combines a username and password with a time-based One-Time Password (OTP), delivered via a secure channel like email or SMS. This Two-Factor Authentication (2FA) adds an extra layer of protection by requiring two independent forms of verification, making it difficult for attackers to gain access even if one credential is exposed. Once the user is authenticated, they can upload files that are immediately protected using AES-256 encryption, a widely trusted method recognized for ensuring strong data security during both transmission and storage.After the file is encrypted, a distinct QR code is created, which contains a secure link directing to the encrypted version stored on the server.To enhance reliability, especially in cases of

poor lighting or physical damage, Reed- Solomon error correction is applied during the QR code creation process. This allows the code to be scanned correctly even when partially obscured or degraded. Importantly, the decryption key is not embedded within the QR code or the file itself; instead, it is transmitted separately to the intended recipient via a secure, encrypted communication channel such as email. This separation of data and keys adds another layer of protection by minimizing the risk of key interception.

To enhance protection, the system applies specific limits on each QR code's usage, such as restricting the number of times it can be scanned and setting a time frame after which the code becomes inactive.These include a **scan count limit**, which restricts how many times the code can be used, and a **validity timer**, which renders the code inactive after a certain duration. When either condition is met, the system automatically disables access through that QR code, effectively preventing its reuse or exploitation by unauthorized users. This prevents scenarios where QR codes are leaked, copied, or reused beyond the intended scope.The system includes a secure

and intuitive user interface that guides users through each step of the file-sharing process from login and encryption to QR generation and secure file retrieval. By simplifying the workflow and minimizing user errors, the interface helps maintain high security standards while ensuring ease of use.

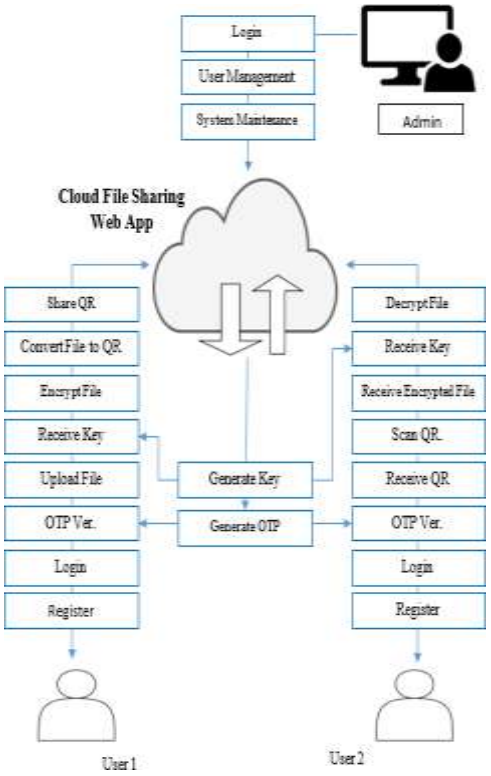

Figure 1: System Architecture

The platform also logs all access attempts and activities, which supports auditing, monitoring, and future analysis. These comprehensive measures ensure that sensitive information remains protected throughout its lifecycle, making AuthCloudFile a reliable and secure solution for encrypted file sharing in personal, corporate, and institutional settings.

System Architecture

Fig 1 illustrates The architectural diagram outlines the structure and interaction flow of the Cloud File Sharing Web Application, highlighting key components: Admin, User 1 (Sender), and User 2 (Receiver).

The **Admin** handles backend tasks like login, user management, and system maintenance. **User 1** registers, logs in with OTP verification, uploads a file, receives an encryption key, encrypts the file, and generates a QR code for sharing.

User 2 also registers and logs in with OTP, scans the QR code, retrieves the encrypted file and key, and decrypts the file to access its contents.

The **Web Application** facilitates secure interactions, manages OTP and key generation, and ensures encrypted file sharing using QR codes to prevent unauthorized access.

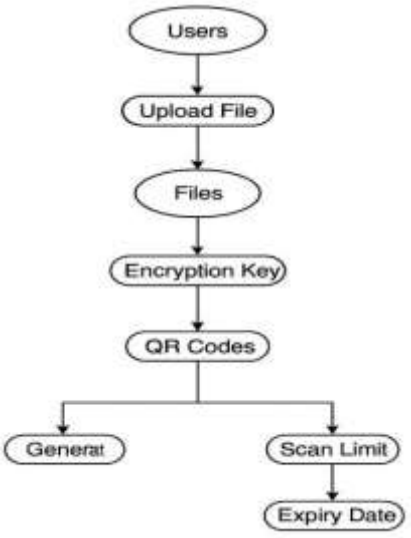

Figure 2: Data Flow Diagram

Results and Discussion:

S.No	Aspect	Test Case / Feature	Metric / Result	User Feedback / Discussion
1	Security Evaluation	Brute Force Resistance (AES-256)	Practically infeasible	AES-256 provides strong encryption; brute force attempts are computationally unfeasible.
2	Security Evaluation	QR Code Forgery Resistance	0% success in 100 forgery attempts	Encrypted and time-bound QR codes prevent cloning and unauthorized access.
3	Security Evaluation	Unauthorized Access Logging	100% attempts logged and blocked	System logs all failed attempts for audit and traceability, enhancing security posture.
4	Security Evaluation	Data Integrity Verification (Hashing)	All file hashes matched	Ensures data integrity; no file tampering detected during storage or transfer.
5	User Acceptance Testing (UAT)	File Upload & Encryption Flow	100% success rate	Users reported fast, simple upload experience with automatic background encryption.
6	User Acceptance Testing (UAT)	QR Code- Based Fil Access	98% success rate	Innovative method; minor scan delays in low light; Reed- Solomon correction minimized scan errors.
7	User Acceptance Testing (UAT)	Expiration/Sca n Limit Alerts	95% success rate	Users appreciated clear notifications for access limits, enhancing control and file lifespan awareness.
8	User Acceptance Testing (UAT)	Multi-Device Accessibility	92% success rate	Good performance on mobile; minor UI feedback for desktops suggests future improvement in layout responsiveness.
9	User Acceptance Testing (UAT)	Two-Factor Authentication (2FA)	97% success rate	Users felt secure; some delays due to OTP delivery issues via email/SMS, unrelated to the system itself.
10	Performance Testing	File Encryption Speed (≤10MB)	Avg. 1.5 seconds	Fast processing time; performance consistent across different file types.

Conclusion and Future Enhancement

In conclusion, this project presents a highly secure and efficient file-sharing system that integrates state-of-the-art security mechanisms, including AES-256 encryption, Two-Factor Authentication (2FA), and QR code-based access control. Traditional file-sharing methods often suffer from vulnerabilities such as unauthorized access, data interception, weak encryption standards, and improper access control. This system addresses these concerns by providing a robust, encrypted, and authentication-driven approach to file sharing.The implementation of AES-256 encryption ensures that files are securely stored and transmitted, preventing unauthorized access or tampering. This encryption standard is widely recognized for its strength and is used in critical security applications worldwide. To strengthen data protection, every encrypted file is associated with a distinct AES encryption key, which is handled and delivered through a separate secure channel, ensuring that only authorized recipients can decrypt and access the file content.

A significant enhancement in this project is the integration of QR codes for access control. Instead of relying on manually shared links or passwords, the system dynamically generates QR codes embedded with metadata, including the encrypted file's location and authentication details. The use of Reed-Solomon error correction in QR code generation ensures that scanning remains reliable and accurate even under less-than-ideal conditions, such as low-light environments or slight distortions.To prevent unauthorized access and enhance user authentication, the system incorporates Two-Factor Authentication (2FA). This involves a combination of traditional username-password credentials along with OTP verification via email or mobile. This dual-layer security measure prevents unauthorized users from accessing the system, even in cases where login credentials are compromised. Additionally, the project includes secure key transmission mechanisms to ensure that decryption keys are not exposed to potential attackers. The encryption key is sent separately via a secure email channel, reducing the risk of interception and unauthorized access.Furthermore, the system employs expiry control mechanisms, which restrict access by imposing expiration times and scan limits on QR codes. These measures prevent prolonged unauthorized access, ensuring that files remain protected even after sharing.

Future Enhancements

Looking ahead, the project can be further enhanced through the integration of biometric authentication methods, such as fingerprint or facial recognition, to increase the robustness of user verification. The system could also benefit from AI-based threat detection that monitors abnormal access behavior and issues alerts or auto- revokes access. Another potential improvement is the implementation of blockchain technology to maintain a tamper- proof audit trail for all file transactions, enhancing transparency and traceability.Additionally, expanding compatibility to support mobile apps and browser extensions will improve user accessibility and convenience. Support for multi-language interfaces can also make the system more inclusive for global users. Integration with enterprise platforms such as Microsoft 365, Google Workspace, or cloud- based document editors will help streamline workflows for business users. Finally, a secure file preview system could be developed to allow users to view shared content without needing to download the file, minimizing potential security risks.

These future enhancements aim to further strengthen the system's usability, scalability, and security, ensuring that AuthCloudFile remains a cutting-edge solution in the evolving landscape of secure digital communication.

References

1. Stallings, W. (2018). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson.
2. Kaufman, C., Perlman, R., & Speciner, M. (2016). *Network Security: Private Communication in a Public World* (2nd ed.). Pearson.
3. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of Applied Cryptography*. CRC Press.
4. Kizza, J. M. (2020). *Guide to Computer Network Security* (5th ed.). Springer.
5. Schneier, B. (2015). *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (2nd ed.). Wiley.
6. Sutton, M. (2017). *The Complete Guide to*

Two-Factor Authentication. O'Reilly Media.
7. Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography Engineering: Design Principles and Practical Applications*. Wiley.
8. Bowen, P., Hash, J., & Wilson, M. (2016). *Information Security Handbook: A Guide for Managers*. NIST Special Publication.
9. Rhodes-Ousley, M. (2013). *Information Security: The Complete Reference* (2nd ed.). McGraw-Hill.
10. Viega, J., & McGraw, G. (2002). *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley.
11. Oppliger, R. (2016). *Security Technologies for the World Wide Web* (3rd ed.). Artech House.
12. Paar, C., & Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer.
13. Grimes, R. A. (2017). *Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto*. Wiley.
14. Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems* (3rd ed.). Wiley.
15. Garfinkel, S., & Spafford, G. (2002). *Web Security, Privacy & Commerce* (2nd ed.). O'Reilly Media.
16. Singh, S. (2000). *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor.
17. Dierks, T., & Rescorla, E. (2011). *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246. IETF.
18. Wang, H., & Lu, Y. (2017). *Data Security and Privacy in Cloud Computing*. CRC Press.
19. Bellovin, S. M. (2019). *Thinking Security: Stopping Next Year's Hackers*. Addison-Wesley Professional.
20. Howard, M., & Lipner, S. (2006). *The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software*. Microsoft Press.