Secure File Storage System with Encryption & Role-Based Access

Dr.G. Jagan Naik[1], Jyothirmayee,G. Poojitha[2], G. Susmitha[3], K. Pallavi[4]
[1]*Associate Professor,Computer Science and Enigneering (Data Science), Cmr Institute of Technology,Hyderabad,Telangana, India*
[2,3,4]*Student, Computer Science and Enigneering of Data Science, Cmr Institute of Technology,Hyderabad,Telangana, India*

***Abstract*—As digital data continues to grow at an exponential rate, the need to store confidential files in a secure manner with controlled access has become a necessity for organizations and individuals. Many file storage systems use basic authentication mechanisms to prevent unauthorized access, but they do not protect the data in the files from unauthorized use if the storage system is compromised. This paper describes a secure file storage system that employs AES-based symmetric encryption of files before they are stored on the server, controlled file-level permissions for access to files by upload, download, and sharing, user authentication mechanisms to limit access to the system by unauthorized users, and activity logging to monitor file-related activities such as uploads and downloads. The system is developed using a web-based architecture that is simple, easy to use, and reliable. The results indicate that the system can securely store and manage files while remaining simple to use and applicable in environments that require reliable protection of sensitive digital data. The system also provides scope for future enhancements to further improve security and functionality.**

***Index Terms*—Secure File Storage, Encryption, Access Control, Data Security, AES Encryption, User Authentication, Data Confidentiality, File Integrity, Activity Logging.**

I. INTRODUCTION

In today's digital age, the widespread use of online platforms and data-driven applications has resulted in an exponential growth in the production and storage of digital files, which are often stored in digital storage systems that manage sensitive information such as personal records, academic documents, and confidential organizational data. As data storage increasingly moves towards centralized and network-based systems, protecting the confidentiality, integrity, and controlled access of the data has become an important requirement. Secure file storage systems are needed to protect the digital assets from unauthorized access, accidental leakage, and malicious misuse while allowing legitimate users to access required information efficiently. Over the years, various file storage and sharing systems have been developed to address these needs, but existing solutions generally rely on encryption to protect stored data and basic authentication mechanisms to restrict system access.

Some systems included audit logs to record user activities and allow administrators to track file access and detect suspicious behavior, but logging features were not always seamlessly integrated with encryption and access control components, which made monitoring and accountability less effective. In summary, existing research indicated the need for secure file storage solutions that integrate encryption, access control, and activity monitoring in a cohesive, comprehensive system, and although previous approaches addressed these aspects individually, there was still a gap in providing a simple, efficient, and integrated system that ensures encrypted storage, controlled file-level access, and accountability, which motivated the development of a secure file storage system that combines these features in a practical and user-friendly manner. In addition, the increasing adoption of web-based applications has introduced new challenges related to secure file handling and access management. A secure file storage system must therefore ensure that confidentiality, access control, and accountability are enforced consistently throughout all stages of file usage.

While encryption was usually applied to files before they were stored on the server, access control was often limited or inconsistently enforced, making it difficult to balance strong security with ease of use.

This work focuses on the design and implementation of a secure file storage system that integrates encryption with controlled file-level access, ensuring confidentiality of data even if storage resources are compromised, managing access at the application level by distinguishing file owners and authorized users, without relying on complex policy-based access frameworks, along with user authentication to ensure that only legitimate users can interact with the system and activity logging of file upload and download operations. The main contribution is to provide a practical, secure, and user-friendly file storage solution that combines encryption, controlled access, and monitoring in a unified platform, addresses key limitations of previous work, is suitable for real-world usage, and enhances data security, improves accountability, and supports reliable file management.

II. RELATED WORK

Several researchers have explored secure file storage systems with the primary goal of protecting sensitive digital data from unauthorized access and misuse. As the volume of digital information has grown rapidly, especially in organizational and academic environments, ensuring the confidentiality and safety of stored files has become a major research focus. Early work in this area concentrated mainly on cryptographic techniques to secure data during storage and transmission. Among these techniques, symmetric encryption algorithms, particularly the Advanced Encryption Standard (AES), were widely adopted due to their efficiency, speed, and suitability for handling large files, many studies demonstrated that encrypting files before storing them on a server significantly reduced the impact of data breaches, even if the storage infrastructure was compromised.

In addition to encryption, authentication mechanisms were commonly studied as a means of restricting system access to legitimate users. Most existing systems relied on traditional username–password-based authentication models While these mechanisms provided a basic layer of security, several researchers observed that authentication alone was insufficient for protecting sensitive files. In many implementations, once a user successfully logged into the system, broad access to stored files was permitted without adequate restrictions. This approach increased the risk of data misuse, particularly in multi-user environments where users had varying access requirements. As a result, researchers emphasized the need for stronger access management beyond simple authentication.

Access control techniques were introduced in several studies to regulate how users interacted with stored resources These approaches aimed to define permissions that limited file operations such as viewing, downloading, or sharing. Some systems implemented permission-based sharing models, where file owners explicitly granted access rights to selected users. Although these mechanisms improved control compared to unrestricted access models, many implementations lacked fine-grained file-level control . In several cases, access rules were applied broadly, making it difficult to manage permissions efficiently for individual files. Furthermore, complex access control frameworks often increased system overhead and configuration complexity, negatively affecting system usability and performance.

Another important aspect highlighted in earlier research was the role of monitoring and accountability in secure file storage systems. Audit logging mechanisms were proposed to record user activities, including file uploads, downloads, and access attempts . These logs enabled administrators to track file usage patterns and detect suspicious behaviour. However, researchers noted that logging mechanisms were frequently implemented as standalone components, without seamless integration with encryption and access control processes. This separation reduced the effectiveness of monitoring and limited the ability to correlate user actions with access decisions and security events.

Usability was also identified as a critical concern in secure storage research. Several studies reported that overly complex security mechanisms discouraged user adoption and increased the likelihood of configuration errors. Systems that required extensive manual policy management or complex role definitions often introduced usability challenges, especially for non-technical users. As a result, researchers recommended security designs that strike a balance between strong protection and ease of use ,. However, such simplified systems sometimes compromised on detailed access

control or comprehensive activity monitoring, leaving security gaps unaddressed.

Overall, existing research indicates that while encryption, authentication, access control, and logging have each been studied extensively, they are often treated as independent components rather than as part of an integrated solution. Many systems successfully addressed individual security requirements but failed to provide a cohesive framework that combines encrypted storage, controlled file-level access, and effective monitoring in a simple and user-friendly manner. This gap in the literature highlights the need for secure file storage systems that integrate these essential security features into a unified platform, motivating the development of solutions that offer both strong protection and practical usability

III. METHODOLOGY

The proposed Secure File Storage System is designed to protect sensitive digital files through encryption, controlled access, and secure user authentication. The system is implemented using a web-based architecture that allows authorized users to upload, store, and retrieve files securely. The overall workflow integrates authentication, encryption, access control, and monitoring mechanisms to ensure data confidentiality and system accountability.

The process begins with user authentication. Any individual attempting to access the system must first provide valid login credentials. The system verifies these credentials and allows access only to authenticated users. This ensures that unauthorized individuals cannot access the system or perform file operations.

After successful authentication, users can upload files to the system. Before storing the file on the server, the system encrypts the file using the Advanced Encryption Standard (AES). This encryption converts the original file into an unreadable format, ensuring that even if the storage server is compromised, the file contents remain protected.

The system also implements controlled file-level access. The user who uploads a file becomes the file owner and has the authority to grant or revoke access permissions for other users. Only users who receive explicit permission from the file owner can access or download the file. When an authorized user requests a file, the system verifies the user's access permissions. If the permission is valid, the encrypted file is retrieved from storage and decrypted before being delivered to the user. This ensures that only legitimate users can view the original file contents.

Additionally, the system maintains activity logs that record important system operations such as file uploads, downloads, and access attempts. These logs help administrators monitor system usage and detect suspicious activities.

A. System Architecture

The architecture of the Secure File Storage System consists of several interconnected components including the user interface, authentication module, encryption module, access control module, and secure storage database. Users interact with the system through a web interface where they can upload and manage files. The authentication module verifies user credentials, while the encryption module secures files using AES before storage. The access control module manages file permissions, and the database stores encrypted files along with metadata.

Fig. 3.1. System Architecture

B. Flowchart

The workflow of the secure file storage system is illustrated in the flowchart. The process begins with user login and authentication. After successful authentication, users can upload files which are encrypted before storage. When another user requests access to a file, the system checks access permissions. If the user is authorized, the encrypted file is decrypted and delivered to the user.

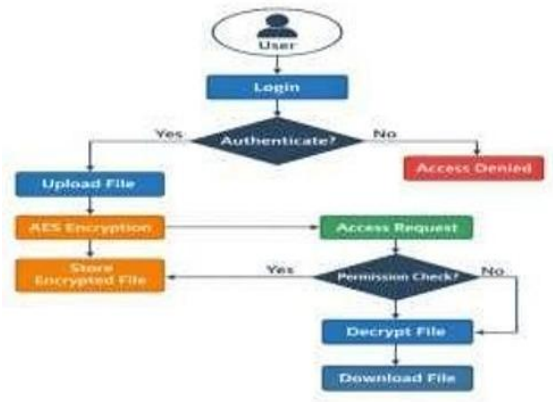

Fig. X. Flowchart of Secure File Storage System

Fig. 3.2. System Flowchart

C. Algorithm: Secure File Storage and Retrieval

Input: File from authenticated user

Output: Secure file storage and controlled access

1. User logs into the system using valid credentials
2. System verifies user authentication
3. User uploads a file
4. System encrypts the file using AES encryption
5. Encrypted file is stored in the server database
6. File owner assigns access permissions to selected users
7. When a user requests a file, system checks permission
8. If access is authorized, retrieve encrypted file
9. Decrypt the file using AES decryption
10. Deliver the file to the authorized user
11. Record activity in system logs

D. Implementation Environment

The system was implemented using Python 3.7.2 with the Django web framework for backend development. The MySQL database was used to store encrypted files and metadata. AES encryption was used to secure file contents, while OTP-based authentication was used to verify user identity. The web interface enables users to upload, share, and retrieve files securely.

IV. RESULT ANALYSIS

The Secure File Storage System with Encryption and Role-Based Access was tested for functionality, security, and reliability in a local environment using Python 3.7.2, Django framework, and MySQL database. The required dependencies were installed from the requirements file, the database was configured using the provided SQL script, and the application server was started to ensure proper system setup and readiness.

Fig. 4.1. Home Screen

The above figure shows Django server execution and system startup after the server was successfully initialized, the local host URL was accessed through a web browser to the system home page, which included navigation options to register a new user or log in an existing user, which would be the entry point for all system operations, verifying the web application worked as expected.

Fig. 4.2. Signup Screen

Fig. 4.3. Login Screen

The above figures show the home page of a secure file storage system new users could register by providing their username, password, contact number, email address, and address; the system would send a One-Time Password (OTP) to their registered email to confirm account registration.

Fig. 4.4. OTP sent to email

The above figure shows OTP received via email after the OTP was entered and validated, the user registration process was completed, and during user login, the OTP was generated and sent to the registered email address; the user was allowed to the system dashboard only after successful OTP validation, thus implementing multi-factor authentication for secure login.

Fig. 4.5. OTP Validation Screen

The above figure shows the OTP validation screen. Once authenticated, users can upload files through the upload and share interface, where the owner of the file would choose one or more users to share by assigning access permissions, and encrypt uploaded files to protect the confidentiality of the data before storing

Fig.4.6. File upload Screen

The above figure shows the file upload and sharing interface. Files were encrypted and stored on the server in an unreadable format, and file metadata such as file owner, upload date, and SHA-256 hash value were maintained to ensure data integrity.

Fig. 4.7. Decryption file Screen

The above figure shows authorized users can view shared files and their corresponding hash values, and the system verified access permissions and decrypted the file before restoring the file to a readable format, which verified that the correct decryption was applied and the file was retrieved securely.

Fig.4.8. Download file Screen

The above figure shows the screen for downloading the decrypted file. Consistency in multiple user sessions was also noted, with multiple users able to be successfully registered and authenticated without conflicts in file access or permissions, with file sharing operations as expected between users, and access restrictions enforced. Hash values that were present alongside files allowed for verification of file integrity after download, ensuring that files were not changed during storage or transmission. The result analysis also showed that the system accurately logs all major user activities such as signup, login, file upload, and file download, allowing for traceability and accountability.

Parameter	Traditional File Storage	Proposed Secure File Storage System
Data Confidentiality	Limited / Vulnerable	Strong (Encrypted Storage)
Encryption Mechanism	Often Absent / Basic	AES-Based Encryption
Access Control	Basic Authentication	Role-Based Access Control
User Authentication	Single-Factor	Multi-Factor (OTP-Based)
File Integrity	Not Explicitly Verified	SHA-256 Hash Verification
Unauthorized Access Protection	Moderate	High
Controlled File Sharing	Limited	Permission-Driven
Audit / Activity Logging	Minimal / Absent	Comprehensive Logging
Data Breach Impact	High Risk	Reduced (Encrypted Files)
Scalability	Moderate	Scalable Architecture
Threat Resistance	Reactive	Proactive Security Controls
Performance Overhead	Low	Moderate (Due to Encryption)
Reliability	Variable	Stable & Secure
Accountability	Limited	Enhanced Traceability
Compliance Readiness	Weak	Improved Security Alignment
System Security Model	Fragmented	Integrated Security Framework

Table 4.1. Comparative Analysis of Traditional and Secure File Storage Systems

V. DISCUSSION

Secure file storage solutions continue to face challenges in achieving an effective balance between data confidentiality, controlled access, and user accountability within a single system. Although encryption and authentication mechanisms are widely adopted, their integration with user-level access control and activity tracking remains limited in many practical implementations. This limitation leads to the research question of whether a unified framework can be designed to ensure secure file storage while maintaining usability and operational efficiency.

VI. CONCLUSION

The study addressed the research gap in traditional file storage solutions by demonstrating an approach to enhance file security through the integration of encryption and controlled file-level access while maintaining usability. The system implemented encrypted file storage, authenticated user access, permission-based file sharing, and activity logging. Experimental results confirmed that files were securely stored in encrypted form and were accessible only to authorized users. OTP-based verification improved the reliability of user authentication. Overall, the system successfully achieved secure storage, controlled access, and accountability, providing a practical and reliable solution for protecting sensitive digital data.

VII. FUTURE WORK

Future work includes enhancements to the authentication mechanisms to further strengthen user identity verification and overall access security, extension to more flexible and fine-grained access control policies to improve file sharing management among users in collaborative environments, and performance optimization techniques to evaluate system behaviour under larger user loads and increased file volumes, which will significantly improve scalability and applicability of the system in real-world organizational environments.

REFERENCES

[1] *OWASP Foundation*, *OWASP Top Ten Web Application Security Risks*, 2024.

[2] *National Institute of Standards and Technology*, *Digital Identity Guidelines*, NIST Special Publication 800-63, Gaithersburg, MD, USA.

[3] *National Institute of Standards and Technology*, *Advanced Encryption Standard (AES)*, FIPS PUB 197 (Updated), Gaithersburg, MD, USA.

[4] *National Institute of Standards and Technology*, *Recommendation for Key Management*, NIST Special Publication 800-57, Gaithersburg, MD, USA.

[5] G. Jagan Naik, B. Dhanalaxmi, Y. Raju, and C. R. S. Rao, "Automated breast cancer segmentation and classification in mammogram images using a deep learning approach."

[6] M. Dworkin, *Recommendation for Block Cipher Modes of Operation*, NIST Special Publication 800-38A, National Institute of Standards and Technology, Gaithersburg, MD, USA.

[7] D. Boneh and V. Shoup, *A Graduate Course in Applied Cryptography*. Stanford, CA, USA: Stanford University, 2020.

[8] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Hoboken, NJ, USA: Wiley, 2020.

[9] M. Whitman and H. Mattord, *Principles of Information Security*, 7th ed. Boston, MA, USA: Cengage Learning, 2021.

[10] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 9th ed. Harlow, U.K.: Pearson, 2023.

[11] G. Jagan Naik, "Effective distributor-based decision-making approach using ETL, data warehousing based on smart business intelligent technology," *International Journal of Environmental Sciences*, ISSN: 2229-7359.

[12] *National Institute of Standards and Technology*, *Computer Security Resource Center (CSRC)*, National Institute of Standards and Technology, Gaithersburg, MD, USA.

[13] R. Sandhu *et al*., "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[14] *National Institute of Standards and Technology*, *Role-Based Access Control (RBAC) Project Overview*, National Institute of Standards and Technology.

[15] *National Institute of Standards and Technology*, *Guide to Attribute-Based Access Control (ABAC)*, NIST Special Publication 800-162, Jan. 2014.

[16] *Cloud Security Alliance*, *Security Guidance for Critical Areas of Cloud Computing*, 2021.

[17] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. New York, NY, USA: Wiley.

[18] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proc. 19th ACM Symposium on Operating Systems Principles (SOSP)*, 2003, pp. 29–43.

[19] G. Jagan Naik, "Explainable AI and blockchain for cyber resilient online retail: A framework for enhanced security and trust," *International Journal of Environmental Sciences*, ISSN: 2229-7359.

[20] *European Union Agency for Cybersecurity*, *Threat Landscape Report for Cybersecurity*, 2023.