ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

SECURE FILE STORAGE SYSTEM WITH ENCRYPTION AND ROLE BASED ACCESS

[1]BORRA MARUTHI PRIYA, [2]S K ALISHA

[1]Students, Department of MCA, B V Raju College, Bhimavaram Ap

[2]Associate Professor, Department of MCA, B V Raju College, Bhimavaram Ap

ABSTRACT

With the rapid growth of digital data and cloud-based storage systems, ensuring data security, integrity, and controlled access has become a critical challenge. Traditional file storage systems often lack strong encryption mechanisms and proper access control, making them vulnerable to unauthorized access, data breaches, and tampering. This project proposes a Secure File Storage System with Encryption and Role-Based Access Control, integrating advanced cryptographic techniques and multi-factor authentication to enhance data security and user trust. The proposed system utilizes a hybrid encryption approach by combining the Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA) algorithms. RSA is used for secure key generation, while AES is employed for efficient file encryption, ensuring both security and performance. To further strengthen authentication, the system incorporates Multi-Factor Authentication (MFA) using One-Time Passwords (OTP) sent to the user's email, allowing access only after successful verification. For data integrity verification, the system uses the SHA-256 hashing algorithm, which ensures that any modification in file content can be detected by comparing hash values. The system also implements Role-Based Access Control (RBAC), allowing users to securely share files with specific individuals based on assigned permissions. Only authorized users can view or download shared files, ensuring controlled access. Additionally, all user activities such as login, file upload, sharing, and download are recorded in log files to prevent unauthorized actions and enable auditing. Experimental results demonstrate that the proposed system provides strong security, efficient file management, and reliable access control. Overall, the system offers a robust and scalable solution for secure file storage and sharing in modern digital environments.

Keywords: AES, RSA, Hybrid Encryption, Multi-Factor Authentication (MFA), OTP,

SHA-256, Role-Based Access Control (RBAC), Data Security, File Encryption, Secure Storage

I.INTRODUCTION

With the increasing reliance on digital platforms and cloud-based systems, secure file storage and sharing have become essential requirements for individuals and organizations. Sensitive data such as personal documents, financial records, and confidential business information are frequently stored and transmitted online. However, traditional file storage systems often lack strong security mechanisms, making them vulnerable to unauthorized access, data breaches, and tampering. Issues such as weak authentication, absence of encryption, and improper access control can lead to significant security risks. Therefore, there is a growing need for advanced systems that ensure data confidentiality, integrity, and controlled access.

Cryptographic techniques play a vital role in securing digital data. Symmetric encryption algorithms like AES (Advanced Encryption Standard) provide fast and efficient data encryption, while asymmetric algorithms such as RSA (Rivest–Shamir–Adleman) offer secure key exchange mechanisms. Combining these two approaches results in hybrid encryption, which enhances both security and performance. In addition, hashing algorithms like SHA-256 are used to verify data integrity by generating unique hash values for files. Along with encryption, authentication mechanisms are equally important. Multi-Factor Authentication (MFA), which includes OTP-based verification, adds an extra layer of security by ensuring that only authorized users can access the system.

The proposed Secure File Storage System integrates hybrid encryption, MFA, and Role-Based Access Control (RBAC) to provide a comprehensive security solution. The system allows users to upload, encrypt, and securely store files, as well as share them with specific users based on defined roles and permissions. Each file is protected using AES encryption with RSA-generated keys, and its integrity is verified using SHA-256 hashing. User activities are logged to maintain accountability and prevent unauthorized actions. This system aims to provide a secure, reliable, and scalable solution for file storage and sharing in modern digital environments.

II SURVEY OF RESEARCH

The study by J. Daemen and V. Rijmen (2001) [1] introduced the Advanced Encryption Standard (AES), a symmetric encryption algorithm widely used for securing digital data. The methodology focuses on block cipher encryption with

high efficiency and security. Results demonstrate that AES provides strong protection against brute-force attacks and is suitable for large-scale data encryption. However, secure key management remains a challenge. This research is highly relevant as AES is used in the proposed system for encrypting files.

The work by R. Rivest, A. Shamir, and L. Adleman (1978) [2] introduced the RSA algorithm, an asymmetric cryptographic technique for secure key exchange. The methodology uses public and private keys for encryption and decryption. Results show that RSA ensures secure communication over insecure channels. However, it is computationally expensive for large data encryption. This study supports the use of RSA for key generation in the hybrid encryption approach.

The research by N. Koblitz and A. Menezes (2015) discussed cryptographic hash functions such as SHA-256 [3]. The methodology focuses on generating unique hash values for data integrity verification. Results demonstrate that SHA-256 is highly secure and resistant to collision attacks. However, it cannot recover original data from the hash. This research is relevant for verifying file integrity in the proposed system.

The study by W. Stallings (2017) explored multi-factor authentication (MFA) techniques for secure system access [4]. The methodology combines multiple authentication factors such as passwords and OTPs. Results show that MFA significantly reduces unauthorized access risks. However, it may increase user interaction complexity. This research supports the implementation of OTP-based authentication in the system.

The work by R. Sandhu et al. (1996) introduced Role-Based Access Control (RBAC) for managing user permissions [5]. The methodology assigns roles to users and defines access rights based on those roles. Results indicate that RBAC simplifies access management and improves security. However, role management can become complex in large systems. This study is relevant as the proposed system uses RBAC for file sharing and access control.

The research by B. Schneier (1996) provided insights into applied cryptography and secure system design [6]. The methodology covers encryption techniques, authentication mechanisms, and security protocols. Results highlight the importance of combining multiple security approaches for robust protection. This research supports the integration of encryption, authentication, and access control in the proposed system.

III. WORKING METHODOLOGY

The proposed Secure File Storage System operates through a combination of authentication, encryption, integrity verification, and access control mechanisms. Initially, the system begins with user registration and authentication. New users can sign up by providing required details, including a valid email address. During login, the system implements Multi-Factor Authentication (MFA) by sending a One-Time Password (OTP) to the registered email. The user must enter the correct OTP to complete the login process, ensuring that only authorized users can access the system. This step significantly enhances security by preventing unauthorized access even if login credentials are compromised.

In the next stage, the system handles file upload and encryption using a hybrid cryptographic approach. When a user uploads a file, the system generates encryption keys using the RSA algorithm. These keys are then used by the AES algorithm to encrypt the file before storing it in the system. The encrypted file is saved in a secure storage location, ensuring confidentiality of the data. At the same time, a SHA-256 hash value is generated for the file, which is stored for later verification. The system also allows users to share files with others based on Role-Based Access Control (RBAC), where permissions are assigned to specific users, ensuring controlled access to shared resources.

In the final stage, authorized users can view and download files based on their assigned permissions. When a user requests to download a file, the system verifies access rights and then decrypts the file using the appropriate keys. Before providing the file, the system checks the SHA-256 hash to ensure that the file has not been tampered with. All user activities, including login attempts, file uploads, sharing actions, and downloads, are recorded in log files for auditing and security monitoring. This methodology ensures a secure, transparent, and efficient file storage and sharing system.

IV RESULTS EXPLANATIONS

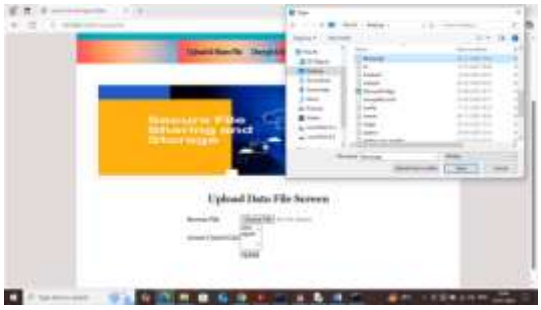
In this project we have added combination of AES along with RSA algorithm to perform hybrid encryption where RSA algorithm is used for KEY generation and then AES employed to encrypt given file using RSA generated keys. To secure application access we have added MFA (multi factor authentication) by allowing application to send OTP to user email and once OTP authenticated then only user will be allowed to upload and download file. To perform file verification we have used SHA256 hash code which can be used to verify file content. If uploaded and verifying

file hash code then only file will be consider as tamper proof. User can upload and share file with other users as ROLE based access and if the user who has permission to access file can view and download file.

To prevent unauthorized access we are logic each activity of all users in a log file.

To implement this project we have designed following modules

1) New User Signup Here: user can sign up with the application
2) User Login: user can login to system, after login user get OTP in mail which has to utilize to complete login process
3) Upload & Share File: user can upload and share file with other desired users and file will be saved at static/files folder in encrypted format
4) Decrypt & Download File: using this module user can view all his uploaded and other user shared files permission list and then can download desired files



In above screen select and upload file and then choose sharing user by holding CTRL key of you want to share with multiple users and then press button to save file and get below page

In above screen select the users and then press button to get below page and in above screen by giving share permission role to Alice

In above screen encrypted file saved at cloud which can see in below screen

In above screen uploaded file can see in encrypted format and all encrypted wil get saved inside 'StorageApp/static/files' folder. Similarly you can upload as many files as you want and now click on 'Decrypt & Download File' to get list of files

In above screen user can see list of uploaded and share files along with SHA256 hash code and now click on 'Download' link to download file and get below page

In above screen file is downloaded in decrypted format. Similarly by following above screens you can upload and download as many files as you want. In below screen you can see all log details saved inside log.csv file which contains all user activities details

V.CONCLUSION

The proposed Secure File Storage System with Encryption and Role-Based Access Control provides a comprehensive and robust solution for protecting sensitive digital data. By integrating hybrid encryption techniques using AES and RSA, the system ensures strong data confidentiality while maintaining efficient performance. The use of Multi-Factor Authentication (MFA) with OTP verification adds an additional layer of security, preventing unauthorized access even in cases of credential compromise. Furthermore, the implementation of SHA-256 hashing guarantees data integrity by enabling detection of any file tampering.

The system's Role-Based Access Control (RBAC) mechanism allows users to securely share files with specific individuals based on defined permissions, ensuring controlled and authorized access. Additionally, maintaining detailed logs of

user activities enhances accountability and enables effective monitoring of system usage. Experimental results demonstrate that the system successfully secures file storage, ensures safe sharing, and provides reliable access control.

In conclusion, the proposed system offers a scalable, secure, and efficient approach to modern file storage and sharing challenges. While the system performs effectively, future enhancements may include integrating cloud-based distributed storage, improving key management techniques, and incorporating advanced authentication methods such as biometrics. Overall, this project highlights the importance of combining encryption, authentication, and access control to build secure digital systems.

REFERENCES

[1] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," *NIST AES Proposal*, 2001.

[2] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[3] N. Koblitz and A. Menezes, "The Random Oracle Model: A Twenty-Year Retrospective," *Des. Codes Cryptogr.*, vol. 77, no. 2–3, pp. 587–610, 2015.

[4] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson, 2017.

[5] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," *IEEE Comput.*, vol. 29, no. 2, pp. 38–47, 1996.

[6] B. Schneier, *Applied Cryptography*, 2nd ed. Wiley, 1996.

[7] M. Bishop, *Computer Security: Art and Science*. Addison-Wesley, 2003.

[8] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," RFC 3174, 2001.

[9] NIST, "Secure Hash Standard (SHS)," FIPS PUB 180-4, 2015.

[10] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[11] C. Paar and J. Pelzl, *Understanding Cryptography*. Springer, 2010.

[12] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[13] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015.

[14] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[15] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2010.