A Semi-Decentralized File Storage System Using Distributed and Secure Architecture

Diksha V. Mane[1], Sakshi P. Gholap[2], Janhavi U. Kalambe[3], Mansi S. Rupanawar[4]
[1,2,3,4] *Member, Department of Computer Engineering, Dilkap Research Institute of Engineering and Management Studies, Neral, Maharashtra, India*

***Abstract*—With the rapid growth of digital data, traditional centralized storage systems face challenges related to security, data privacy, single point of failure, and scalability. To address these limitations, decentralized file storage systems have emerged as a reliable alternative by distributing data across multiple nodes without relying on a central authority. This project focuses on the design and development of a decentralized file storage system that ensures secure, reliable, and efficient data storage. The system utilizes distributed architecture to store files in an encrypted and fragmented manner, improving data integrity and availability. The proposed solution aims to enhance security, fault tolerance, and transparency while reducing dependency on centralized servers. This paper presents the system architecture, working mechanism, and key features of the developed decentralized file storage system along with its advantages and potential applications.**

***Index Terms*—Decentralized File Storage, Distributed Systems, Data Security, Peer-to-Peer Network, Blockchain Technology, Data Integrity**

I. INTRODUCTION

With the rapid growth of digital data and online applications, secure and reliable data storage has become a critical requirement. Traditional centralized storage systems rely on a single server or authority, which introduces issues such as single point of failure, data breaches, lack of transparency, and dependency on third-party service providers. These limitations raise serious concerns related to data security, privacy, and availability.

Decentralized file storage systems provide an alternative approach by distributing data across multiple nodes in a network rather than storing it in a centralized location. This architecture improves fault tolerance, enhances data availability, and gives users greater control over their data. However, fully decentralized systems often require cryptocurrency-based transactions, which can increase complexity and cost for users.

This project presents the design and development of a semi-decentralized file storage system that combines the benefits of decentralization with cost efficiency. The system uses MetaMask for secure user authentication and IPFS (InterPlanetary File System) as the decentralized storage network. Files are encrypted before uploading to IPFS and decrypted after retrieval to ensure data confidentiality. A semi-decentralized approach is adopted to manage file metadata, eliminating the need for cryptocurrency transactions. This paper discusses the system architecture, working methodology, and advantages of the proposed solution.

II. LITERATURE SURVEY

Table 2.1 Key Literature and Learnings Applied in the Proposed System

Ref. No.	Paper Name	What Was Learned & Applied in the Project
[1]	IPFS – Content Addressed, Versioned, P2P File System	This paper introduced IPFS as a decentralized storage network using content-addressed hashing. The project uses IPFS to store encrypted files in a distributed manner and retrieve them using unique hash values.

Ref. No.	Paper Name	What Was Learned & Applied in the Project
[2]	Bitcoin: A Peer-to-Peer Electronic Cash System	Provided foundational understanding of decentralization and peer-to-peer networks. Helped in understanding trustless systems, although cryptocurrency usage was intentionally avoided in the project.
[3]	Blockchain Technology: Beyond Bitcoin	Explained how blockchain concepts can be applied beyond digital currency. This helped in understanding decentralized data management and system transparency.
[6]	Ethereum: A Secure Decentralized Generalized Transaction Ledger	Provided insights into decentralized identity and wallet-based authentication. MetaMask authentication in the project is conceptually derived from Ethereum-based identity mechanisms.
[8]	Storj: A Peer-to-Peer Cloud Storage Network	Demonstrated practical implementation of decentralized cloud storage. Influenced the idea of distributing encrypted file fragments across multiple nodes.
[9]	Secure Distributed Data Storage in Cloud Computing	Highlighted the importance of encryption and distributed storage for data security. The project applies encryption before uploading files to the decentralized network.
[11]	Compact Proofs of Retrievability	Emphasized data availability and integrity in distributed systems. Helped in understanding how distributed storage ensures data retrievability even when nodes fail.
[14]	An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends	Provided architectural understanding of decentralized systems. Helped design the semi-decentralized architecture used in the project.
[16]	AES Proposal: Rijndael	Provided the basis for symmetric encryption techniques. The project uses encryption mechanisms inspired by AES for secure file storage and retrieval.
[18]	A Survey on Blockchain-Based Data Storage Systems	Summarized challenges such as cost and complexity in fully decentralized systems. This motivated the adoption of a semi-decentralized approach to avoid cryptocurrency payments.

The analysis of the selected literature highlights the evolution of decentralized storage systems and the key technologies that support secure and reliable data management. The reviewed papers provide insights into IPFS-based storage, peer-to-peer architectures, encryption techniques, and decentralized authentication mechanisms. These studies collectively emphasize the importance of data security, fault tolerance, and decentralization while also identifying challenges such as system complexity and cost associated with fully decentralized models. The findings from this literature survey form the foundation for the proposed system design and justify the adoption of a semi-decentralized approach for efficient and cost-effective decentralized file storage.

III. METHODOLOGY

The methodology of the proposed system focuses on secure file storage, decentralized data management, and cost-efficient access control. The system is designed using a semi-decentralized architecture that

integrates MetaMask authentication, encryption techniques, and IPFS-based decentralized storage.

Initially, the user accesses the web application and authenticates using MetaMask. MetaMask provides a secure and decentralized identity mechanism without requiring traditional username and password credentials. After successful authentication, the user uploads a file to the system.

Before storage, the uploaded file is encrypted using symmetric encryption to ensure confidentiality. The encrypted file is then uploaded to the IPFS decentralized storage network. IPFS stores files in a distributed manner and generates a unique content-based hash for each file, which is used to retrieve the file later.

To avoid the requirement of cryptocurrency transactions, a semi-decentralized approach is adopted. File metadata, including IPFS hash values and user access information, is stored on a centralized or semi-centralized server. During file download, the authenticated user retrieves the file hash from the metadata server, fetches the encrypted file from IPFS, and performs decryption to reconstruct the original file.

This methodology ensures secure data storage, efficient file retrieval, and fault tolerance while eliminating the need for token-based payments. The system balances decentralization and practicality, making it suitable for real-world applications.

IV. SYSTEM ARCHITECTURE

The system architecture of the proposed solution consists of a web-based user interface, MetaMask authentication module, encryption and decryption modules, IPFS decentralized storage network, and a semi-decentralized metadata server. Users interact with the system through a web application and authenticate using MetaMask. Encrypted files are stored on IPFS, while metadata is managed separately to avoid cryptocurrency usage. This architecture ensures secure, reliable, and cost-effective decentralized file storage.

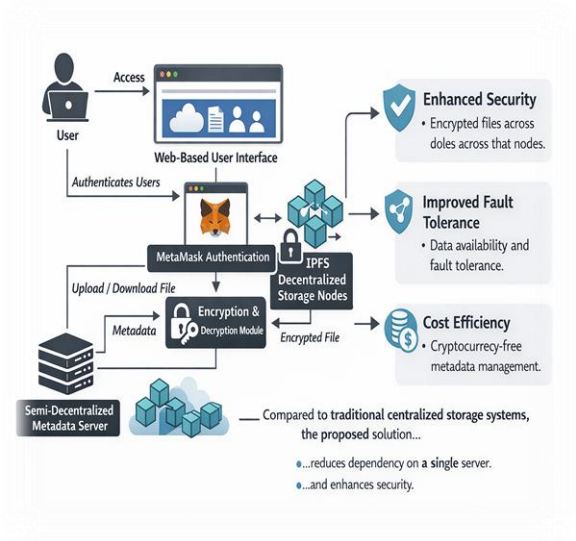

Fig 4.1: System Architecture

V. RESULTS AND DISCUSSION

This section presents the experimental results obtained from the implementation of the proposed **Semi-**Decentralized Encrypted File Storage System. The discussion compares the expected system behaviour with the actual outcomes observed during execution. System screenshots are used as evidence to validate authentication, encryption, decentralized storage, and secure retrieval operations.

A. Expected Outcomes

The system was designed with the following objectives:

- Secure user authentication using MetaMask wallet integration
- Wallet-based cryptographic key generation using digital signatures
- Client-side encryption of files using AES-GCM before upload
- Storage of encrypted files on a decentralized storage network
- Adoption of a semi-decentralized architecture to avoid cryptocurrency payments
- Secure file decryption during download
- User-controlled access without centralized identity providers

The expected system workflow is:
User Authentication → Key Derivation → Client-Side Encryption → Decentralized Upload → Encrypted

Storage → Secure Download → Client-Side Decryption

B. Actual Results Obtained

a) MetaMask Authentication

The system successfully integrates MetaMask for user authentication. Upon clicking the login option, the user wallet is detected and connected, and the wallet address is displayed on the application interface.

Result:

MetaMask authentication was successfully completed and the wallet address was verified.

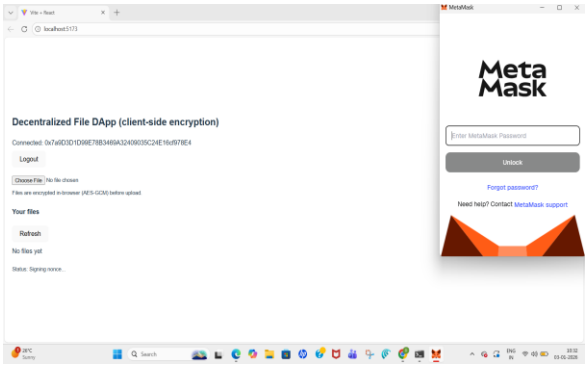

Fig. 5.1 MetaMask wallet connection interface

b) Cryptographic Key Generation

After authentication, the system requests a cryptographic signature from the user wallet. This signature is used to derive a secure encryption key.

Result:

The MetaMask signature request popup appeared and was successfully approved by the user.

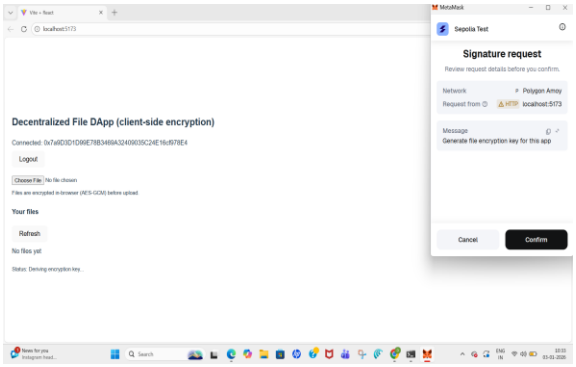

Fig. 5.2 MetaMask signature request for encryption key generation

c) Client-Side File Encryption

Selected files are encrypted locally within the browser using AES-GCM encryption before being uploaded to the storage network.

Result:

Files are encrypted locally within the browser using AES-GCM encryption before being uploaded to the decentralized storage network.

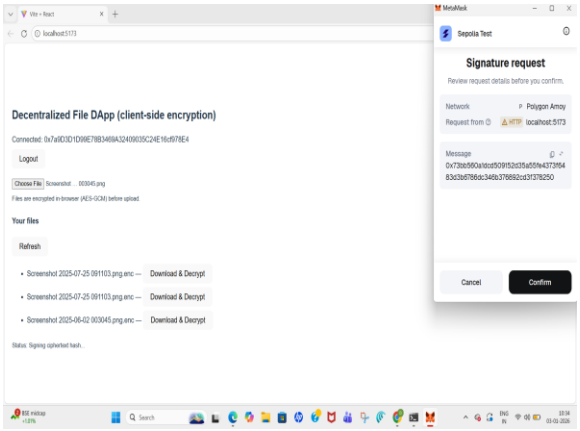

Fig. 5.3 Client-side file encryption before upload

d) Decentralized Storage of Encrypted Files

Encrypted files are uploaded to a decentralized storage system instead of a centralized server.

Result:

Encrypted files were successfully stored and listed in the user dashboard.

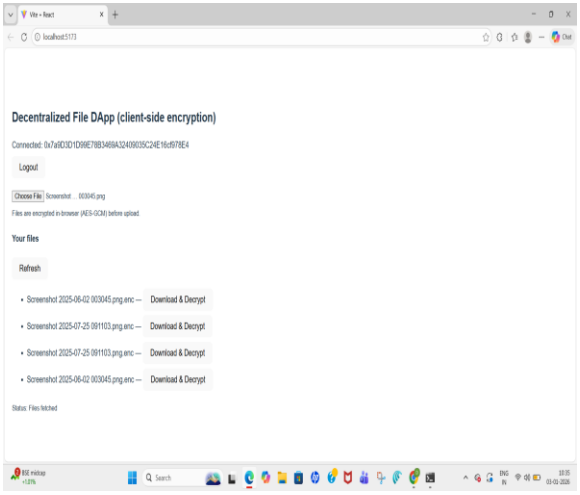

Fig. 5.4 Encrypted files stored in decentralized storage

e) Secure Download and Decryption

Users can download encrypted files and decrypt them locally using the derived encryption key.

Result:

Encrypted files were successfully downloaded and decrypted, restoring the original file format.

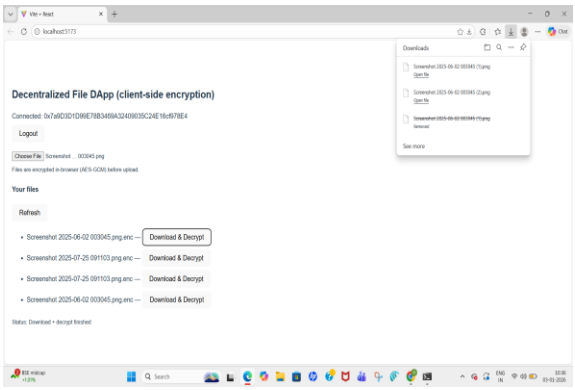

Fig. 5.5 Secure download and decryption of stored files

C. Expected vs Actual Outcome Comparison

Table 5.1 Comparison of Expected Outcomes and Actual Results of the Proposed System

Parameter	Expected Outcome	Actual Outcome
Authentication	MetaMask wallet login	Successfully achieved
Key Generation	Wallet-based encryption key	Successfully derived
Encryption	Client-side AES-GCM	Successfully implemented
Storage	Decentralized encrypted storage	Successfully achieved
Cryptocurrency Usage	No coin or gas fee	Completely avoided
Decryption	Secure local decryption	Successfully achieved
Security Model	Zero-knowledge data storage	Achieved

D. Discussion

The experimental results confirm that the proposed system meets its design objectives. The integration of MetaMask authentication eliminates the need for traditional password-based systems, enhancing security. Client-side encryption ensures that data confidentiality is maintained, as plaintext files are never transmitted or stored externally.

The semi-decentralized architecture successfully avoids cryptocurrency dependencies while retaining the benefits of decentralization. This makes the system suitable for academic and practical applications without financial overhead. The results demonstrate that decentralized file storage combined with cryptographic security and wallet-based authentication provides a reliable and secure alternative to centralized storage solutions.

E. Summary of Results

The developed system successfully demonstrates:

- Secure blockchain-based authentication
- Client-side encrypted file storage
- Decentralized data management
- Secure and reliable file retrieval
- Practical implementation without cryptocurrency costs

These results validate the feasibility and effectiveness of the proposed decentralized file storage system.

VI. CONCLUSION AND FUTURE SCOPE

A. CONCLUSION

This paper presented the design and implementation of a semi-decentralized file storage system that addresses the limitations of traditional centralized storage solutions. By integrating MetaMask for secure user authentication and IPFS for decentralized file storage, the proposed system ensures improved data security, integrity, and availability. Files are encrypted before being uploaded to the decentralized network and decrypted after retrieval, ensuring confidentiality throughout the storage process.

The use of a semi-decentralized architecture allows the system to avoid mandatory cryptocurrency transactions while still benefiting from decentralized storage features. This approach makes the solution more practical, cost-effective, and suitable for real-world applications. The results demonstrate that the proposed system successfully supports secure file upload and download while reducing dependency on a single centralized server. Overall, the system provides a reliable and secure alternative to traditional file storage mechanisms.

B. FUTURE SCOPE

The proposed system can be further enhanced in several ways to improve functionality and scalability. Future work may include the implementation of

advanced access control mechanisms to support role-based permissions and secure file sharing among multiple users. Integration of smart contracts can be explored to automate file management and access policies.

Additionally, system performance can be optimized by improving file retrieval efficiency and reducing latency. The solution can also be extended to support larger file sizes and increased numbers of users. Incorporating audit logs, data versioning, and enhanced security algorithms can further strengthen the system. These enhancements would increase the applicability of the system in enterprise-level and large-scale decentralized storage environments.

VII. ACKNOWLEDGMENT

The authors would like to thank Prof. Rajni Ratnaparkhi, Department of Computer Engineering, Dilkap Research Institute of Engineering and Management Studies, Neral, Maharashtra, India, for valuable guidance and continuous support during the development of this project.

REFERENCES

[1] J. Benet, "IPFS – Content Addressed, Versioned, P2P File System," arXiv preprint arXiv:1407.3561, 2014.
[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," White Paper, 2008.
[3] M. Crosby et al., "Blockchain Technology: Beyond Bitcoin," Applied Innovation Review, vol. 2, 2016.
[4] A. Singla and E. Bertino, "Blockchain-Based Secure Data Storage," IEEE Transactions on Services Computing, 2019.
[5] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, 2016.
[6] G. Wood, "Ethereum: A Secure Decentralized Generalized Transaction Ledger," Ethereum White Paper, 2014.
[7] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," Cisco White Paper, 2011.
[8] S. Wilkinson et al., "Storj: A Peer-to-Peer Cloud Storage Network," White Paper, 2014.
[9] J. Li et al., "Secure Distributed Data Storage in Cloud Computing," IEEE Transactions on Cloud Computing, 2017.
[10] A. Juels and J. Burton, "The Ring of Gyges: Using Smart Contracts for Crime," ACM CCS, 2016.
[11] H. Shacham and B. Waters, "Compact Proofs of Retrievability," ASIACRYPT, 2008.
[12] M. Armbrust et al., "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, 2010.
[13] Y. Deswarte, L. Blain, and J. Fabre, "Intrusion Tolerance in Distributed Computing Systems," IEEE Symposium on Security and Privacy, 1991.
[14] Z. Zheng et al., "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," IEEE BigData Congress, 2017.
[15] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, 1992.
[16] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," NIST, 2001.
[17] A. Gervais et al., "On the Security and Performance of Proof of Work Blockchains," ACM CCS, 2016.
[18] X. Wang et al., "A Survey on Blockchain-Based Data Storage Systems," IEEE Access, 2020.
[19] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication, 2011.
[20] S. Goldwasser and S. Micali, "Probabilistic Encryption," Journal of Computer and System Sciences, 1984.
[21] K. Croman et al., "On Scaling Decentralized Blockchains," International Conference on Financial Cryptography, 2016.
[22] Y. Yuan and F. Wang, "Blockchain and Cryptocurrencies: Model, Techniques, and Applications," IEEE Transactions on Systems, Man, and Cybernetics, 2018.
[23] A. Miller et al., "The Honey Badger of BFT Protocols," ACM CCS, 2016.
[24] S. Ziegler et al., "Security and Privacy in Decentralized Storage Systems," IEEE Security & Privacy Magazine, 2019.
[25] L. Lamport et al., "The Byzantine Generals Problem," ACM Transactions on Programming Languages and Systems, 1982.
[26] M. Swan, "Blockchain: Blueprint for a New Economy," O'Reilly Media, 2015.

[27] D. Boneh and M. Franklin, “Identity-Based Encryption from the Weil Pairing,” SIAM Journal of Computing, 2003.
[28] A. Baliga, “Understanding Blockchain Consensus Models,” Persistent Systems White Paper, 2017.
[29] P. Kahn et al., “Decentralized Identity and Access Management,” IEEE Internet Computing, 2020.
[30] E. Androulaki et al., “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” EuroSys Conference, 2018.