Secure Blockchain-Based File Sharing

Mr.Rajasekaran T[1], Ajaykumar M[2], Arjun S[3], and Harish Kumar S T[4]
[1,2,3,4] *Member, Department of Computer Science and Engineering, SRM Valliammai Engineering College, Chennai, India*

***Abstract*—Decentralized secure file sharing is a challenging task, especially ensuring access control and anonymity preservation. In this paper, a blockchain-based file-sharing system is presented that makes use of the InterPlanetary File System (IPFS) for off-chain storage and AES-256 encryption to provide stronger security. The system architecture incorporates smart contracts in Solidity that provide robust access control with immutability and transparency characteristics.**

Conventional file-sharing systems using central storage possess certain weaknesses regarding breaches, misuse, and the existence of single points of failure. The approach that we suggest overcomes these weaknesses by applying end-to-end encryption before the data is uploaded to IPFS, thereby securing the information in case it finds its way to unauthorized users. Moreover, file permission management can be achieved using the application of smart contracts to ensure that the file data finds its way to authorized users in a way that facilitates safe decryption and download by both parties. Apart from enhancing the privacy, security, and integrity of the data, the decentralized nature of the system eliminates the need for third-party storage providers, thereby offering a secure and resilient solution for file sharing.

***Index Terms*—Access control, AES-256 encryption, Blockchain, Data privacy, Decentralized file sharing, End-to-end encryption, IPFS, secure storage, smart contracts, solidity,**

I. INTRODUCTION

Within today's digital climate, the safe exchange of files has become a crucial aspect of managing data, especially as the prevalence of cyberattacks and privacy violations increases. Traditional file-sharing methods are mainly dependent on centralized cloud storage and consequently expose wide security risks towards data breaches, unauthorized access, and server downtime. In addition, these central schemes tend to push users toward a dependence on the services of external providers, whose actions might include privacy violations, data manipulation, or even criminal monitoring. For these and other concerns that have raised issues with using file-sharing facilities online, an escalation of calls has been occurring in favor of decentralized, tamper-proof, and privacy-facilitating facilities for file exchange.

To prevent such problems, we recommend a file-sharing system based on blockchain technology combined with AES-256 encryption and thus providing integrity of the data, which currently resides in the distributed Interplanetary File System (IPFS). The files would be encrypted in the form of AES-256 to discourage any unauthorized individual from viewing the data, even if he by chance manages to get the stored data. As opposed to cloud models, our system does not utilize central servers and instead depends on IPFS for decentralized and efficient storage of files. The access control system, in the form of smart contracts, provides permission control and facilitates decryption and access to the shared files only by authorized staff, thus providing an immutable record of all access transactions.

One of the significant advantages of our approach is the combination of blockchain's decentralization and immutability with cryptographic encryption to provide an immutable and trustless file-sharing environment. By eliminating middlemen and providing automated access permissions through smart contracts, our framework enhances data privacy, integrity, and transparency. Additionally, the decentralized nature of IPFS ensures high availability and censorship resistance, making the system highly resistant to data loss and outside attacks.

II. RELATED WORK

The field of secure file sharing has seen significant advancements with the integration of blockchain technology. This section explores the existing literature under key themes, focusing on decentralized storage, encryption techniques, and smart contract-based access control.

A. Blockchain-Based File Sharing Systems

1. "A Blockchain-Based File-Sharing System with IPFS for Distributed Storage and Smart Contract-Based Access Control" by Ahmed et al. (2021). This study demonstrates enhanced security by eliminating central storage threats and automating access permissions through Ethereum smart contracts.
2. "Decentralized Storage and Access Control in Blockchain-Based File Sharing" by Kumar et al. (2022). This research highlights the advantages of content addressing via IPFS to prevent single points of

failure and ensure data integrity using cryptographic hashing.

B. Encryption Techniques in Secure File Sharing

1. "AES-256 Encryption for Secure File Storage in Blockchain Systems" by Wang et al. (2020). This paper investigates the application of AES-256 encryption to guarantee confidentiality by performing client-side encryption before storing files in IPFS.
2. "Homomorphic Encryption for Secure Blockchain-Based File Sharing" by Huang et al. (2023). This study explores the potential of homomorphic encryption in enabling computations on encrypted data without decryption, preserving data privacy.

C. Zero-Knowledge Proofs and Authentication Mechanisms

1. "Privacy-Preserving Authentication in Blockchain File Sharing Using Zero-Knowledge Proofs" by Chen et al. (2021). This research highlights the ability of ZKP to facilitate authentication without revealing identity, ensuring confidential access control.
2. "Multi-Signature Authentication for Enhanced Security in Decentralized File Sharing" by Zhang et al. (2024). This study demonstrates how multi-signature authentication strengthens security by reducing the risks associated with single-key authentication systems.

D. Advanced Security Frameworks and Scalability Solutions

1. "Hybrid Blockchain Framework with Digital Signatures and Multi-Factor Authentication" by Patel et al. (2023). This research proposes a multi-layered authentication mechanism to enhance access control within file-sharing systems.
2. "Performance Optimization in Blockchain-Based File Sharing: The Role of Layer 2 Solutions" by Gupta et al. (2022). This study addresses transaction overhead issues and suggests sidechains as a viable solution to improve scalability and operational efficiency.

These studies form the foundation for combining blockchain, encryption, and decentralized file storage to build secure, scalable, and privacy-preserving file-sharing frameworks. Our proposed system integrates AES-256 encryption, IPFS storage, and Ethereum smart contracts to ensure robust security and efficient data sharing.

III.PROPOSED MODEL

The system proposed guarantees secure and decentralized file sharing by combining blockchain technology, AES-256 encryption, and the InterPlanetary File System (IPFS). The approach follows a structured form that ensures data confidentiality, availability, and integrity and, additionally, supports transparent and immutable access control. The process starts with a user choosing a file to share, which is encrypted using the AES-256 encryption standard to ensure confidentiality. The encrypted file is then stored in IPFS, a decentralized file storage system that distributes data across multiple peer-to-peer nodes, thereby avoiding possible single points of failure. Successful storage of the file causes IPFS to generate a unique cryptographic hash known as a Content Identifier (CID), which is an immutable pointer to the file. This CID is stored on the blockchain via a smart contract, which acts as an access control mechanism.

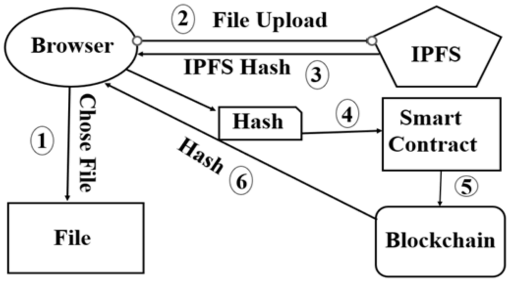


Figure 1: Data flow model

This Ethereum blockchain-deployed smart contract keeps track of possession records and access rights, enabling file owners to dynamically grant or revoke access by adjusting the contract parameters. Upon receiving a request to access a shared file from a user, they fetch the CID from the blockchain and use it to download the respective encrypted file from IPFS. Decrypting the file necessitates the recipient to have the right AES-256 decryption key, which is securely shared through an off-chain process. The system also includes a file expiration feature, where the owner can specify a fixed time of validity for file access. After the expiration time, the smart contract automatically revokes access, ensuring shared files are not accessed for more than their intended lifespan. In addition, an immutable audit log stores all file access transactions on the blockchain, preserving significant metadata such as timestamps, file IDs, and Ethereum wallet addresses of users. This provides accountability and improves security by preventing unauthorized tampering. By taking advantage of blockchain's immutability, IPFS's decentralization, and AES-256 encryption, the proposed system offers a trustless and scalable file-sharing solution, removing dependency on third-party intermediaries while ensuring robust data protection and privacy.

IV. SYSTEM ARCHITECTURE

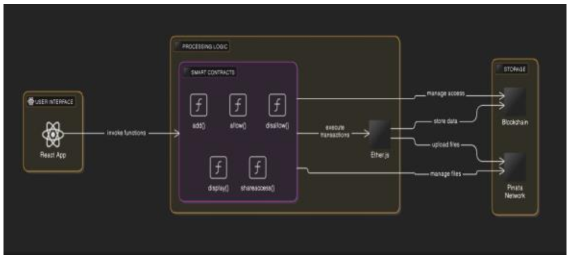


Figure 2: System Architecture

Employing blockchain and IPFS for safe sharing of files and AES-256 encryption ensures data integrity, privacy, and data decentralization. Conventional file-sharing activities involve the use of centralized servers, making the system vulnerable to threats such as data breaches, unauthorized access, and single point of failure vulnerability. Employing blockchain for immutable access control, IPFS for data storage decentralization, and AES-256 encryption for data safeguarding, the system provides a secure and effective file-sharing atmosphere. The system also employs a feature of file expiration and end-to-end access logs to provide absolute control and transparency of data shared.

A. User Input

The process starts when a user chooses a file to upload. The file can be related to any number of formats, i.e., documents, images, or videos, and is to be processed in a secure way. The system checks whether the file is ready to be encrypted and stored in a decentralized manner, thus ensuring it is safe from unwanted individuals. Visual representations, like snapshots of the window of file selection, can conveniently depict this starting point.

B. AES-256 Encryption

Before it's uploaded, the file is first encrypted with AES-256, which transforms it into an unreadable ciphertext. This robust symmetric encryption method allows only the owners of the proper decryption key to view the original content of the file. Because the encryption is client-side, the confidential information in the file is protected from storage providers and network administrators.

C. IPFS Storage and Hash Generation

Once the file is encrypted, it is uploaded onto the IPFS (Interplanetary File System). IPFS fragments the file into pieces and saves them in a decentralized network of nodes. Every file gets a cryptographic hash (CID) that is unique, serving as its identity. This CID is not modified, which implies the content of the file is verifiable without any modifications. IPFS is different from traditional cloud storage, where the files are within a centralized database, as it provides data availability, durability, and censorship resistance.

D. Blockchain-Based Access Control

To ensure safe access to files, the uploaded file's content identifier (CID) is saved in a smart contract on the Ethereum blockchain. The smart contract manages a mapping of access-granted users and file owners, thus making final and irreversible access controls. When sharing a file, one updates the smart contract to allow access to an Ethereum address. The system ensures that only permitted users can obtain the CID and access the encrypted file.

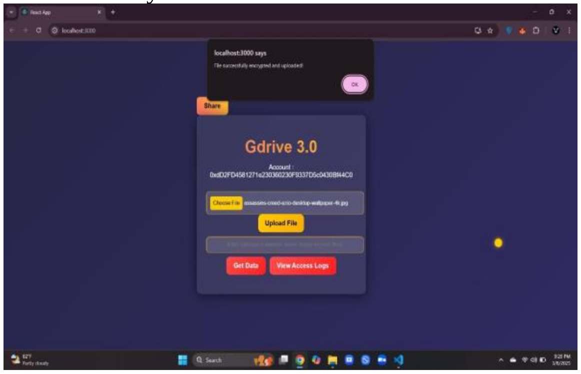
E. File Decrypt and Retrieve

When a file-access-authorized individual initiates a request for file access, he or she initiates a connection with the smart contract, retrieves the Content Identifier (CID) of the targeted file, and goes on to download the encrypted file from the Interplanetary File System (IPFS). The recipient will use a securely distributed private key to decrypt the ciphertext to its original state. In addition, the system has a file expiration feature where the owner can define a particular time period during which the file is made available. Upon the expiration of the defined time period, the system automatically denies access to the file. Every access attempt to the file is painstakingly logged, recording information such as the timestamp, file ID, and Ethereum wallet address of the user. Such logs enable end-to-end accountability and traceability. In addition, these features could be enhanced further with the addition of a user interface that graphically displays a countdown timer for expiration and a table of access history.

F. Final outcome

The result is a distinctly stable, decentralized, and transparent record-sharing device that uses AES-236 encryption contracts, IPFS, and smart storage based on Ethereum to obtain input to control. This combination ensures that documents remain personal, unchanging, and useful for authorized customers while eliminating the risks related to centralized garage solutions. The machine offers stronger safety through implementation functions such as document validity and logging mechanisms, further strengthening privacy and responsibility.

When a user selects a document for upload, the system applies AES-256 encryption, changing the registration to inappropriate ciphertext before storage. As encryption occurs in the aspect of the consumer, even with storage or blockchain, we cannot enter the exclusive content material of the file. After encryption, the machine loads the file to IPFS, which generates a completely exclusive cryptographic hash (ICD) that serves as its eternal identifier. Unlike conventional cloud garage responses, IPFS ensures excessive availability, decentralization, and integrity of content material, interrupting unauthorized modifications or exclusions. The encrypted registration ICD is then stored in an intelligent contract implemented in Ethereum Blockchain, which maintains a mapping of reporters and authorized users. When a file owner wants to share a record, it provides the entry for a particular Ethereum to deal with, making sure that simpler authorized customers can recover ICD and get admission to the encrypted report. Because blockchain transactions are immutable, admission to permissions cannot be changed or forged, improving acceptance as true and safety.



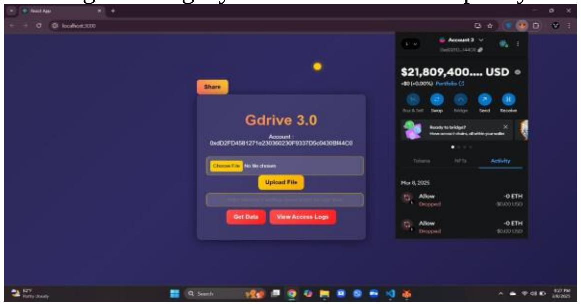
localhost:3000 says

File successfully encrypted and uploaded!

Account :
0xdD2FD4581271e230360230F9337D5c0430Bf44C0

Figure 3: Document transmitted to the network

An authorized user accessing a file uses a Blockchain CID to first access it. He then goes ahead and downloads the encrypted IPFS file using the CID. The file remains encrypted and cannot be accessed without the relevant decryptive key. The recipient of the key must then download it from the file owner via any form of communication outside the blockchain. Upon downloading the key, the AES-256 decryptographic algorithm downloads the file to its original state, restricted only to authorized parties. This implies that even if an intruder manages to gain access to a dictation file stored on the IPFS, he cannot decrypt the contents of the file without the key, hence ensuring confidentiality for such information. The system also has an additional range of protection and control for filing. The owner, in splitting a file, can set an outlet time, after which file access is automatically deleted. The system has a provision for administrators or users to then delete access after the time of the specified executions, preventing extended access to unauthorized sensitive files. This facility specifically comes in handy for temporary sharing of confidential information since it ensures that the information is not accessible beyond the given deadline. With the blockchain, it ensures that this mechanism cannot be altered by the administrator or by any unauthact, hence ensuring the integrity of the access control policy.



Account :
0xdD2FD4581271e230360230F9337D5c0430Bf44C0

$21,809,400.... USD

Mar 8, 2025

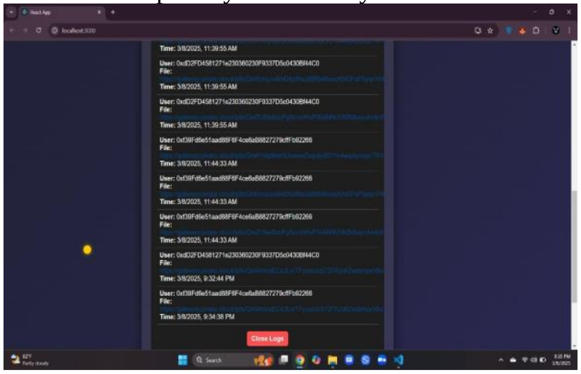
Allow Dropped -0 ETH

Allow Dropped -0 ETH

Figure 4: MetaMask

The integrated file access logging feature enables both resilience and transparency. All actions on file access are comprehensively logged, such as the Ethereum wallet address of the user, the time of access, the CID of the accessed file, and an indicator of whether the file was accessed successfully or not. Since file owners are able to identify users who attempted to access a file, even in cases of unauthorized access, accountability is facilitated. Since all access logs are stored in a permanent state, this ensures users that no logs have been modified, which is of utmost concern in cases involving strict security compliance.

The integration of these fundamental properties successfully mitigates the security risks of conventional centralized file-sharing techniques. In contrast, operations such as encryption, decentralized storage, immutable access controls, expiration limits, and access tracking guarantee that files are stored securely, are only accessible to authorized parties, and have a high degree of tamper resistance or unauthorized alteration. This system offers a trustless, censorship-resistant, and verifiable framework for secure digital file sharing, utilizing a new blockchain-based methodology. In doing so, it establishes a new benchmark in privacy and security.



```
Time: 3/8/2025, 11:39:55 AM
User: 0xdD2FD4581271e230360230F9337D5c0430Bf44C0
File:
Time: 3/8/2025, 11:39:55 AM
User: 0xdD2FD4581271e230360230F9337D5c0430Bf44C0
File:
Time: 3/8/2025, 11:39:55 AM
User: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
File:
Time: 3/8/2025, 11:44:33 AM
User: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
File:
Time: 3/8/2025, 11:44:33 AM
User: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
File:
Time: 3/8/2025, 11:44:33 AM
User: 0xdD2FD4581271e230360230F9337D5c0430Bf44C0
File:
Time: 3/8/2025, 9:32:44 PM
User: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
File:
Time: 3/8/2025, 9:34:38 PM
```

Figure 5: Access log

V. CONCLUSION AND FUTURE WORK

This project is all about creating a secure and decentralized way of file sharing, using blockchain technology, IPFS for distributed storage, and AES-256 encryption to keep it all private, intact, and accessible. In contrast to the typical file-sharing mechanisms that rely on centralized servers, this mechanism does away with the weaknesses of single points of failure, unauthorized access, and tampering with the data. AES 256 encryption truly makes files extra safe and secure when they're just sitting there on a public IPFS network. And then, to make it even more awesome, not anyone can just go accessing those files when using Ethereum smart contracts that are entirely controlled and public transactions and data. The system also has features such as file expiry and monitoring through file access logs, which give file owners the ability to set expiry time and monitor who's accessing their files. The decentralized nature of IPFS ensures efficient and censorship-resistant storage of files; hence, files remain always accessible without relying on a single server. With files being encrypted before upload, even storage guys or nodes that are part of a blockchain just can't decrypt the content. This way nothing gets leaked and everything remains private and really secure.

Although this project is a solid basis for secure file sharing, there is tremendous scope for development and innovation in the future. One such area of importance is the integration of Zero-Knowledge Proof (ZKP), which enables users to prove their permission to access without disclosing confidential data. This protocol would further increase privacy and security by granting access to authorized parties alone without compromising the confidentiality of their identity and related metadata. Further optimization of the encryption and decryption process could further improve performance, making the system more scalable for global adoption. Presently, the decryption key needs to be conveyed through an off-chain secure communication channel, which involves some degree of manual intervention. Through the integration of decentralized key management protocols, users would be able to obtain their keys securely without depending on external communication. Besides this, integrating multi-user permission control with varying permission levels—view, edit, or full control—would make the system more adaptable for enterprise-level file sharing.

Decentralized identity management, or DID, is a method through which the authentication of users is enabled in a trustless system using decentralized identity management methods. The process would eliminate the use of external authentication systems and make the authentication process smooth and secure. Additionally, the goals of upcoming projects can include minimizing the cost of storage and enhancing retrieval efficiency in the IPFS network by using dynamic pinning methods for maximizing the optimization of files with frequent access without compromising decentralization. Implementation of this project along with cross-chain compatibility would incorporate an enhancement that is achieved with the ability of communication across the whole blockchain system, enhancing responsiveness for different ecosystems and applications. The results of this research prove that decentralized file sharing can be efficient and secure and serve as an alternative to classical centralized cloud storage systems. Nevertheless, the system can be optimized further by incorporating more features such as automatic key management, decentralized identity verification, and better retrieval mechanisms, which would render it scalable and more user-friendly. Blockchain technology integration with decentralized storage has come a long way. This system, as with other systems, is a start towards the realization of more secure, transparent, and trustless file-sharing technology, which attempts to give users more privacy and control of data.

REFERENCES

[1] Benet, J. (2014). IPFS: Content Addressed, Versioned, P2P File System. https://ipfs.tech/ This paper introduces the InterPlanetary File System (IPFS), a peer-to-peer hypermedia protocol designed to make the web faster, safer, and more open through decentralized storage and content addressing.

[2] Steichen, M., Norvill, R., Shbair, W., & State, R. (2018). Blockchain-Based, Decentralized Access Control for IPFS. In Proceedings of the IEEE International Conference on Blockchain (pp. 1499-1506). This study presents a decentralized access control mechanism for IPFS using blockchain technology, enhancing secure file sharing by leveraging smart contracts for permission management.

[3] Yao, J., Chang, Y. P., Tsai, Y. L., & Huang, J. L. (2019). Cloud-Based File Sharing with Blockchain-Based Access Control for Lightweight Encryption. IEEE Transactions on Cloud Computing, 9(1), 89-98. This article explores a cloud-based file-sharing system that integrates blockchain for access control and employs lightweight encryption methods to ensure data security and efficiency.

[4] Huang, W., Yeh, L., & Huang, J. (2019). A monitorable peer-to-peer file-sharing mechanism. In Proceedings of the 20th Asia-Pacific Network Operations and Management Symposium (pp. 1-4). This paper proposes a peer-to-peer file-sharing mechanism with monitoring capabilities, enhancing security and reliability in decentralized networks.

[5] Androulaki, E., et al. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the 13th EuroSys Conference (pp. 1–15). This research discusses Hyperledger Fabric, a modular blockchain framework that supports permissioned networks, relevant for implementing secure and scalable file-sharing systems.

[6] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf This foundational paper introduces Bitcoin and the underlying blockchain technology, providing insights into decentralized systems and cryptographic security mechanisms.