

Secureshare: A Blockchain and IPFS-Base Decentralized File Sharing System

Jiss Johny, Department of Computer Science & Engineering, Nehru College of Engineering & Research Centre

Prathyusha Sunil, Department of Computer Science & Engineering, Nehru College of Engineering & Research Centre

Prinisha P.J, Department of Computer Science & Engineering, Nehru College of Engineering & Research Centre

Rohan R Menon, Department of Computer Science & Engineering, Nehru College of Engineering & Research Centre

Mrs. Soumya T, Assistant Professor, Department of Computer Science & Engineering, Nehru College of Engineering

& Research Centre

Abstract - In the modern digital era, secure file sharing has become increasingly important due to the rapid growth of sensitive data exchanged over the internet. Traditional cloud storage platforms such as Google Drive and Dropbox rely on centralized architectures, which introduce challenges including data breaches, unauthorized access, single points of failure, and lack of user ownership. To overcome these limitations, this paper presents SecureShare, a blockchain-based decentralized file sharing system that ensures data confidentiality, integrity, and transparency. The proposed system integrates the InterPlanetary File System (IPFS) for decentralized storage and Ethereum smart contracts for secure ownership verification and access control. Before uploading, files are encrypted using the Advanced Encryption Standard (AES) algorithm to ensure privacy and prevent unauthorized access. IPFS generates a unique Content Identifier (CID) for each encrypted file, while the blockchain stores file metadata and ownership information in an immutable manner. Additionally, Pinata pinning services are used to maintain persistent file availability within the IPFS network. SecureShare eliminates dependence on centralized intermediaries and enables users to retain full control over their digital assets. Smart contracts enforce secure and transparent access management, allowing only authorized users to retrieve and decrypt files locally. The proposed system demonstrates how blockchain, decentralized storage, and cryptographic techniques can be combined to create a scalable, secure, and privacy-preserving file sharing platform suitable for applications in healthcare, education, research collaboration, and secure document management.

Key Words: Blockchain, IPFS, Ethereum, AES Encryption, Smart Contracts, Decentralized Storage, Secure File Sharing.

1. INTRODUCTION

In the modern digital era, the exchange and storage of files have become essential components of communication,

business operations, education, healthcare, and research activities. Large volumes of sensitive information such as financial records, academic documents, medical reports, multimedia files, and confidential organizational data are frequently shared through online platforms. Traditional cloud-based storage services such as Google Drive, Dropbox, and Microsoft OneDrive provide convenient file access and collaboration features; however, these systems rely on centralized infrastructures where data is stored and controlled by third-party service providers. This centralized architecture introduces several security and privacy challenges including unauthorized access, data breaches, server failures, service downtime, and loss of user control over stored information.

One of the major concerns in centralized file-sharing systems is the dependency on trusted intermediaries for data management and security. Once files are uploaded to centralized servers, users often lose transparency regarding how their data is stored, accessed, or modified. Furthermore, centralized systems create a single point of failure, where a successful cyberattack or infrastructure failure can compromise the availability and integrity of user data. The absence of verifiable ownership and tamper-proof mechanisms further increases the risk of data manipulation and unauthorized modifications.

To overcome these limitations, decentralized technologies such as blockchain and distributed storage systems have emerged as promising alternatives for secure data management. Blockchain technology provides an immutable and transparent ledger capable of securely recording transactions and ownership details without relying on centralized authorities. Similarly, the InterPlanetary File System (IPFS) offers decentralized and content-addressable storage that improves file availability and eliminates dependency on centralized servers. By integrating blockchain with decentralized storage and cryptographic techniques, it is possible to create a secure and transparent file-sharing environment.

This paper presents SecureShare, a blockchain-based decentralized file-sharing system designed to ensure data

confidentiality, integrity, and ownership verification. The proposed system combines Ethereum smart contracts, IPFS distributed storage, AES encryption, and Pinata pinning services to create a secure file management platform. Before uploading, files are encrypted locally using the Advanced Encryption Standard (AES) algorithm to protect sensitive information from unauthorized access. The encrypted files are then uploaded to IPFS, which generates a unique Content Identifier (CID) representing the file. The CID and ownership metadata are stored on the Ethereum blockchain through smart contracts, ensuring immutable verification and transparent access control.

The SecureShare system enables users to securely upload, manage, and share digital assets while maintaining full ownership and control over their data. Smart contracts are used to manage file access permissions, request handling, audit logs, and ownership verification. By eliminating centralized intermediaries and implementing decentralized security mechanisms, the proposed system enhances transparency, reliability, and privacy in digital file sharing. The platform can be effectively utilized in domains such as healthcare, education, research collaboration, and enterprise document management where secure and tamper-proof data exchange is critical.

2. LITERATURE SURVEY

2.1 Proxy Re-Encryption for Secure Distributed Storage

Ateniese et al. [1] introduced the concept of proxy re-encryption for secure distributed storage systems. The proposed method allows a semi-trusted proxy to transform ciphertext encrypted for one user into ciphertext decryptable by another authorized user without revealing the original plaintext. This mechanism improves confidentiality and enables controlled data sharing in distributed environments. The study demonstrated that proxy re-encryption can be effectively applied in cloud storage systems with minimal computational overhead while maintaining strong security. The concept is highly relevant to decentralized file-sharing platforms such as SecureShare, where secure and authorized sharing of encrypted files is essential.

2.2 Self-Sovereign Identity in Decentralized Systems

Bartolomeu et al. [2] proposed a blockchain-based Self-Sovereign Identity (SSI) framework that enables users to manage their digital identities without relying on centralized authorities. The framework utilizes Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) to provide secure and privacy-preserving authentication. The study highlighted that traditional identity systems are vulnerable to privacy breaches and unauthorized access due to centralized control. By

integrating blockchain technology, the proposed approach improves transparency, user ownership, and identity verification. These concepts are applicable in SecureShare for decentralized authentication and secure access management.

2.3 InterPlanetary File System (IPFS)

Benet [3] introduced the InterPlanetary File System (IPFS), a decentralized peer-to-peer storage protocol designed to improve data availability and eliminate dependency on centralized servers. Unlike traditional web storage systems that use location-based addressing, IPFS uses content-addressable storage through cryptographic hashes called Content Identifiers (CID). This approach enables files to be retrieved from multiple distributed nodes, thereby improving scalability, redundancy, and fault tolerance. IPFS also supports Merkle Directed Acyclic Graphs (Merkle DAGs), which enhance storage efficiency and data deduplication. In SecureShare, IPFS is used as the primary decentralized storage layer for encrypted files.

2.4 Hyperledger Fabric Architecture

Cachin [4] presented the architecture of Hyperledger Fabric, a modular permissioned blockchain framework developed for enterprise applications. The framework supports pluggable consensus protocols, private communication channels, and chaincode-based smart contracts for secure transaction processing. The study emphasized scalability, fault tolerance, and secure data management in blockchain systems. Hyperledger Fabric separates transaction ordering from execution, thereby improving network efficiency and performance. Although SecureShare uses Ethereum instead of Hyperledger Fabric, the modular design principles and secure smart contract architecture discussed in this study influenced the system design of SecureShare.

2.5 Blockchain-Based Secure File Sharing

Cui et al. [5] proposed a blockchain-based secure file-sharing framework aimed at improving scalability, privacy, and trust in distributed environments. The study utilized smart contracts to automate file access control and audit mechanisms while storing only metadata on the blockchain. Large files were maintained in off-chain distributed storage systems to reduce blockchain overhead and improve performance. The proposed architecture reduced dependency on centralized servers and enhanced transparency through blockchain verification. This hybrid storage model directly inspired the SecureShare architecture, where encrypted files are stored in IPFS while metadata and ownership information are maintained on the Ethereum blockchain.

2.6 Decentralized Cloud Storage with IPFS

Doan et al. [6] proposed a decentralized cloud storage framework integrating IPFS and blockchain smart contracts for secure and transparent data management. The system addressed major limitations of traditional cloud storage systems such as single points of failure, lack of transparency, and centralized control. Files uploaded to IPFS were identified using unique Content Identifiers (CID), while blockchain smart contracts managed ownership verification and access permissions. The study also highlighted the importance of file persistence in decentralized storage systems and recommended the use of pinning mechanisms to ensure long-term file availability. SecureShare applies similar concepts by integrating Pinata pinning services with IPFS storage.

2.7 Distributed Off-Chain Storage in Healthcare Systems

Kumar, Marchang, and Tripathi [12] proposed a blockchain and IPFS-based healthcare storage architecture for managing sensitive medical records. The system utilized off-chain storage to overcome the storage limitations and high transaction costs associated with blockchain networks. Encrypted healthcare data was stored in IPFS, while metadata such as file hashes, timestamps, and ownership information were maintained on the blockchain. Smart contracts were used to implement secure access control and transparent data sharing among authorized entities. The study demonstrated improved confidentiality, scalability, and integrity in healthcare data management. This hybrid architecture is closely related to the SecureShare system.

2.8 Blockchain-Based Decentralized Applications

Zheng et al. [16] presented a comprehensive survey on blockchain-based decentralized applications (DApps), discussing their architecture, consensus mechanisms, storage models, and scalability challenges. The study explained how decentralized applications combine blockchain networks with distributed storage systems such as IPFS to achieve transparency, security, and fault tolerance. It also highlighted the limitations of storing large files directly on-chain and recommended hybrid architectures using off-chain storage for better performance. Additionally, the survey discussed future directions including decentralized identity, cross-chain interoperability, and privacy-preserving mechanisms. These concepts strongly align with the design and future scope of the SecureShare platform.

3. PROPOSED SYSTEM

The proposed system, SecureShare, is a decentralized and secure file-sharing platform that integrates blockchain technology with the InterPlanetary File System (IPFS). Unlike traditional cloud storage systems such as Google Drive and

Dropbox, which depend on centralized servers, SecureShare utilizes a distributed architecture to eliminate third-party control and reduce the risks associated with centralized data management. The system combines cryptographic encryption, decentralized storage, and blockchain-based verification to ensure secure, transparent, and tamper-proof file sharing.

In SecureShare, files are encrypted locally using the Advanced Encryption Standard (AES) algorithm before being uploaded to the storage network. The encrypted files are then stored in IPFS, which generates a unique Content Identifier (CID) for each file. This CID acts as a cryptographic reference to the file and is recorded on the Ethereum blockchain through smart contracts along with metadata such as file ownership and timestamps. By integrating blockchain with IPFS, the system ensures data integrity, decentralized access control, and verifiable ownership.

The SecureShare architecture is designed based on the three fundamental principles of information security: confidentiality, integrity, and availability. AES encryption protects file confidentiality, blockchain technology ensures integrity through immutable records, and IPFS with Pinata pinning services guarantees decentralized availability and persistence of files.

3.1 System Overview

SecureShare is implemented as a web-based decentralized application that combines encryption, blockchain, and distributed storage technologies. The system provides a user-friendly interface while securely handling backend operations. Users can connect their MetaMask wallets to authenticate themselves and interact with blockchain smart contracts.

When a user uploads a file, the file is first encrypted locally on the client side using AES encryption. The encrypted file is then uploaded to IPFS, where a unique CID is generated. The CID and file metadata are stored on the Ethereum blockchain using smart contracts. This approach ensures that large files remain off-chain for scalability while ownership and access records remain immutable and transparent on the blockchain. When another user requests access to a file, the smart contract verifies whether the user has the required permission. Once authorization is confirmed, the encrypted file is retrieved from IPFS using the CID and decrypted locally by the authorized user. Pinata pinning services are used to maintain long-term availability of uploaded files in the IPFS network.

3.2 FUNCTIONAL MODULES

3.2.1 User Interface Module

The user interface is developed using React.js and provides an interactive environment for users to manage files and blockchain operations. The module allows users to connect

MetaMask wallets, upload encrypted files, view transaction records, manage file permissions, and retrieve shared files. The interface simplifies complex blockchain interactions and improves usability.

3.2.2 Encryption Module

The encryption module secures files before storage by using the Advanced Encryption Standard (AES) algorithm. Encryption is performed locally on the client side to ensure that plaintext files are never exposed to external servers or storage nodes. Only authorized users possessing the correct decryption key can access the original file content.

3.2.3 IPFS Storage Module

The IPFS storage module manages decentralized storage of encrypted files. Once uploaded, IPFS generates a unique Content Identifier (CID) for each file. Since files are distributed across multiple nodes, the system eliminates single points of failure and improves reliability. Pinata pinning services are integrated to ensure persistent file availability.

3.2.4 Blockchain Smart Contract Module

Smart contracts developed using Solidity are deployed on the Ethereum blockchain to manage file metadata and access control. The smart contract stores the CID, owner address, timestamps, and access permissions. It also supports operations such as granting access, revoking access, and verifying ownership. Blockchain immutability ensures transparency and tamper-proof verification.

3.2.5 File Retrieval and Decryption Module

This module handles secure file retrieval and decryption. When a user requests access to a file, the blockchain verifies authorization through the smart contract. The encrypted file is then retrieved from IPFS using the CID and decrypted locally using AES decryption. This process ensures secure and controlled access to shared files.

3.2.6 Pinning and Persistence Module

The system integrates Pinata to maintain file persistence within the IPFS network. Uploaded files are pinned automatically to prevent removal from distributed nodes. This mechanism ensures long-term storage availability even if certain nodes become inactive or unavailable.

3.3 System Architecture

The SecureShare system follows a three-layer architecture consisting of the application layer, blockchain layer, and storage layer.

Application Layer

The application layer handles user interaction and frontend operations. It is developed using React.js and Node.js and interacts with MetaMask for blockchain authentication and transaction signing.

Blockchain Layer

The blockchain layer uses Ethereum smart contracts to manage file metadata, ownership verification, and access permissions. It provides immutable and transparent records of all file-related operations.

Storage Layer

The storage layer uses IPFS for decentralized file storage and Pinata for persistence. Only encrypted files are stored in IPFS, ensuring that sensitive information remains protected.

The layered architecture improves modularity, scalability, and secure communication between different system components.

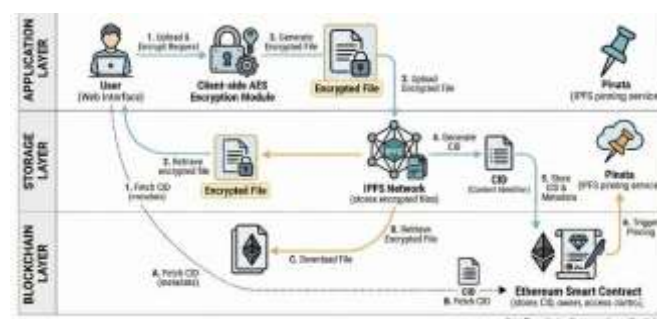


Figure 3.1: Proposed System Architecture of SecureShare

3.4 Workflow of the System

The workflow of SecureShare consists of the following steps:

1. The user selects a file through the web interface.
2. The file is encrypted locally using AES encryption.
3. The encrypted file is uploaded to IPFS.
4. IPFS generates a unique CID for the uploaded file.
5. The CID and metadata are stored on the Ethereum blockchain using smart contracts.
6. Pinata pins the uploaded file to ensure persistent availability.
7. Authorized users can retrieve the encrypted file from IPFS and decrypt it locally.

This workflow ensures secure, decentralized, and transparent file-sharing operations.

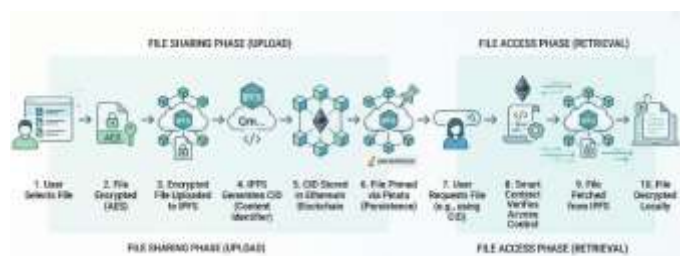


Figure 3.2: Workflow of SecureShare System

3.5 Technology Stack

Layer	Technology
Frontend	React.js, HTML, CSS
Backend	Node.js, Express.js
Blockchain	Ethereum, Solidity
Wallet	MetaMask
Storage	IPFS, Pinata
Encryption	AES
Tools	Remix IDE, Truffle, Ganache
Version Control	Git, GitHub

Table 3.1: Technology Stack of SecureShare

3.6 Security and Privacy

SecureShare ensures security through multiple layers of protection. AES encryption guarantees confidentiality by securing files before upload. Blockchain technology provides immutable ownership verification and secure access control. MetaMask enables secure user authentication without centralized credentials. CID-based verification in IPFS ensures data integrity by detecting unauthorized modifications. Additionally, blockchain transparency provides public auditability of file operations.

3.7 Advantages of the Proposed System

The SecureShare system provides several advantages over traditional centralized storage systems:

- Eliminates dependency on centralized servers
- Ensures secure and encrypted file storage
- Provides tamper-proof ownership verification
- Enables transparent access control through blockchain
- Improves file availability using decentralized storage
- Enhances user privacy and control over data

- Supports scalability through off-chain storage mechanisms

3.8 Summary

The proposed SecureShare system combines blockchain technology, IPFS distributed storage, AES encryption, and smart contracts to create a secure and decentralized file-sharing environment. The system ensures confidentiality, integrity, transparency, and availability while eliminating reliance on centralized storage providers. By integrating decentralized technologies with cryptographic security mechanisms, SecureShare provides a scalable and reliable solution for secure digital file management and sharing.

4. IMPLEMENTATION AND DISCUSSION

4.1 System Functionality

The SecureShare system successfully implements the core functionalities required for decentralized and secure file sharing. Users can securely authenticate themselves using MetaMask, which provides decentralized wallet-based authentication without requiring traditional username-password credentials. Files uploaded by users are encrypted locally using the Advanced Encryption Standard (AES) algorithm before being transmitted to the storage layer, ensuring end-to-end confidentiality.

The encrypted files are stored in the InterPlanetary File System (IPFS), where each file is assigned a unique Content Identifier (CID). The generated CID along with metadata such as owner address and timestamp are recorded on the Ethereum blockchain through smart contracts. The system also supports secure file retrieval by verifying user permissions through blockchain-based access control mechanisms. Authorized users can retrieve encrypted files from IPFS and decrypt them locally, ensuring secure and transparent file management.

4.2 Module wise Implementation

4.2.1 User Management Module

The user management module enables secure authentication and interaction within the system using MetaMask wallets. Each user is uniquely identified by their Ethereum address, and all file-related operations are securely associated with blockchain transactions. The module eliminates the need for centralized authentication systems and enhances user privacy.

4.2.2 Encryption Module

The encryption module uses the AES algorithm to encrypt files locally before upload. Since encryption occurs on the client side, sensitive information is never exposed to external

servers or storage nodes. Only authorized users possessing the correct decryption key can access the original file contents.

4.2.3 IPFS Storage Module

The IPFS module manages decentralized file storage by distributing encrypted files across multiple nodes. Each uploaded file generates a unique CID that acts as a cryptographic reference for retrieval. This decentralized approach improves availability, fault tolerance, and data integrity while eliminating dependency on centralized storage servers.

4.2.4 Blockchain Module

The blockchain module uses Ethereum smart contracts to manage file metadata, ownership verification, and access control. Smart contracts store details such as CID, owner address, and timestamps in an immutable manner. All transactions are publicly verifiable, ensuring transparency and tamper-proof record management.

4.2.5 File Retrieval Module

The retrieval module enables secure and controlled access to shared files. Access permissions are verified through smart contracts before retrieval. Once authorized, the encrypted file is downloaded from IPFS using the CID and decrypted locally using AES decryption.

4.2.6 Pinata Persistence Module

The system integrates Pinata pinning services to ensure long-term availability of uploaded files within the IPFS network. Pinning prevents files from being removed from distributed nodes and improves reliability and accessibility.

4.3 Workflow Demonstration

The workflow of SecureShare begins when a user connects their MetaMask wallet to the application. After authentication, the user selects a file for upload. The selected file is encrypted locally using AES encryption and then uploaded to IPFS. IPFS generates a unique Content Identifier (CID), which is recorded on the Ethereum blockchain along with ownership metadata through smart contracts.

The uploaded file is pinned using Pinata to ensure persistent storage and availability. When another user requests access to the file, the system verifies authorization through blockchain-based access control mechanisms. If access is approved, the encrypted file is retrieved from IPFS using the CID and decrypted locally by the authorized user.

This workflow demonstrates secure coordination between encryption, decentralized storage, blockchain verification, and access management mechanisms.

4.4 Experimental Results

4.4.1 Wallet Connection Interface

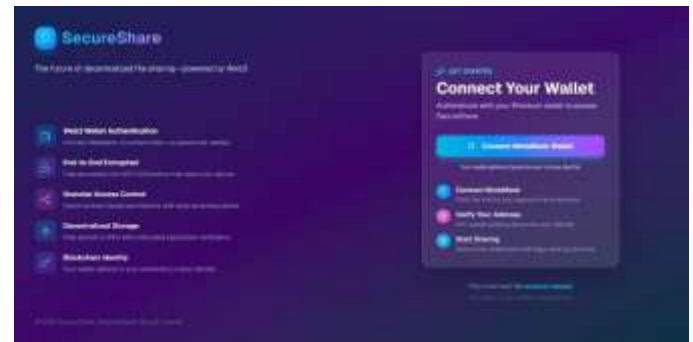


Fig. 4.1: MetaMask Wallet Connection Interface

Shows the decentralized wallet authentication interface used for secure user login.

4.4.2 Dashboard and File Management



Fig. 4.2: Dashboard Showing Uploaded Files

Displays uploaded files, encryption status, ownership details, and file management operations.

4.4.3 File Upload Interface



Fig. 4.3: File Upload and Encryption Interface

Represents the file upload page where files are encrypted locally before being uploaded to IPFS.

4.4.4 Shared Files Interface



Fig. 4.4: Shared Files and Access Status

Shows shared files along with access permissions and sharing status details.

4.4.5 Access Control Interface



Fig. 4.5: Access Control Management Interface

Illustrates the blockchain-based module used for granting and revoking file permissions securely.

4.4.6 Analytics Dashboard



Fig. 4.6: File Access Analytics Dashboard

Displays file usage statistics, access counts, and interaction details within the decentralized system.

4.5 Performance Discussion

The SecureShare system demonstrates efficient performance in terms of security, reliability, and decentralized storage management. IPFS enables fast distributed file retrieval and eliminates dependency on centralized storage servers. Blockchain smart contracts ensure secure ownership verification and transparent access management, although

blockchain transaction confirmation may introduce minor latency depending on network conditions.

AES encryption provides strong confidentiality with minimal computational overhead, making client-side encryption practical for secure file sharing. The integration of Pinata improves file persistence and ensures long-term accessibility of uploaded files within the IPFS network. Overall, the system achieves a balanced combination of security, transparency, and performance.

4.6 Discussion

The SecureShare platform successfully addresses major limitations of traditional centralized file-sharing systems. By integrating blockchain technology with IPFS distributed storage, the system eliminates centralized control and enhances user ownership of data. AES encryption ensures confidentiality, while blockchain immutability guarantees transparency and tamper-proof verification.

Compared to traditional cloud storage platforms, SecureShare provides improved privacy, decentralized access control, and transparent file verification. The use of smart contracts and decentralized storage mechanisms demonstrates the effectiveness of blockchain-based architectures for secure and scalable file-sharing applications.

3. CONCLUSIONS

The SecureShare system demonstrates an effective approach for building a secure, transparent, and decentralized file-sharing platform using blockchain technology and the InterPlanetary File System (IPFS). The proposed system successfully addresses the limitations of traditional centralized storage platforms, including unauthorized access, data breaches, dependency on third-party providers, and single points of failure.

By integrating AES encryption with decentralized storage, SecureShare ensures that files remain confidential and protected throughout the storage and sharing process. The use of IPFS enables distributed file storage with improved availability and reliability, while Ethereum smart contracts provide immutable ownership verification and secure access control. Additionally, Pinata pinning services enhance file persistence and long-term accessibility within the decentralized storage network.

The implementation and testing results confirm that the system operates efficiently across different stages including file upload, encryption, blockchain verification, access management, and secure retrieval. Smart contracts successfully automate access control mechanisms, enabling transparent and tamper-proof file sharing without centralized intermediaries.

Overall, SecureShare demonstrates the practical application of blockchain, cryptographic security, and decentralized storage technologies in modern file-sharing systems. The proposed

architecture provides improved security, transparency, reliability, and user ownership, making it suitable for applications where data confidentiality and integrity are critical.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to the management and administration of Nehru College of Engineering and Research Centre (NCERC) for providing the academic environment, infrastructure, and resources necessary for the successful completion of this research work.

The authors extend their heartfelt appreciation to the Principal and the Department of Computer Science and Engineering for their constant encouragement and institutional support throughout the development of the SecureShare project.

The authors would like to convey special thanks to Mrs. Soumya T, Assistant Professor, Department of Computer Science and Engineering, for her valuable guidance, technical expertise, constructive suggestions, and continuous motivation during every stage of this work. Her support and mentorship played a significant role in the successful completion of the project.

The authors also acknowledge the cooperation and assistance provided by the faculty members, friends, and well-wishers who directly or indirectly contributed to this work. Finally, the authors express their deep gratitude to their family members for their unwavering encouragement, patience, and support throughout the course of the project.

REFERENCES

- Ateniese, G., Fu, K., Green, M., & Hohenberger, S. (2006). "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage." *ACM Transactions on Information and System Security (TISSEC)*, 9(1), 1–30.
- Bartolomeu, P., Vieira, E., & Ferreira, J. (2019). "Self-Sovereign Identity in Decentralized Systems." *IEEE Access*, 7, 150237–150249.
- Benet, J. (2014). "IPFS – Content Addressed, Versioned, P2P File System." *Protocol Labs Whitepaper*.
- Cachin, C. (2016). "Architecture of the Hyperledger Blockchain Fabric." *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL)*.
- Cui, Y., Asghar, M. R., & Russello, G. (2018). "Blockchain-Based Scalable and Trustworthy File Sharing." *IEEE Access*, 6, 15055–15064.
- Doan, M., Tran, N. H., & Niyato, D. (2022). "Toward Decentralized Cloud Storage with IPFS." *IEEE Internet of Things Journal*, 9(4), 3120–3134.
- Dong, C., Russello, G., & Dulay, N. (2011). "Shared and Searchable Encrypted Data for Untrusted Servers." *Data and Applications Security and Privacy XXV*, 127–143.
- Ethereum Foundation. (2019). "Ethereum White Paper: Foundation of Smart Contracts." *Ethereum.org Publication*.
- Filecoin, J. Benet et al. (2017). "Proof of Replication and Incentive-Based Storage." *Protocol Labs Whitepaper*.
- Guidi, B., Michienzi, A., & Ricci, L. (2021). "Data Persistence in Decentralized Social Applications." *IEEE Access*, 9, 41058–41069.
- Jayabalan, S., & Jeyanthi, N. (2022). "A Scalable Blockchain Model Using Off-Chain IPFS Storage." *International Journal of Communication Networks and Information Security (IJCNIS)*, 14(3), 345–355.
- Kumar, R., Marchang, N., & Tripathi, R. (2020). "Distributed Off-Chain Storage in Healthcare Systems Using Blockchain and IPFS." *Journal of Network and Computer Applications*, 168, 102775.
- Moran, T., & Orlov, I. (2019). "Proofs of Space-Time and Rational Proofs of Storage." *Advances in Cryptology – CRYPTO 2019*, 381–406.
- Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System." *Bitcoin.org Whitepaper*.
- Proof of Replication Protocol. (2017). "Technical Specification: Proof of Replication in Decentralized File Systems." *Protocol Labs Documentation*.
- Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2023). "Blockchain-Based Decentralized Applications: A Comprehensive Survey." *IEEE Transactions on Network and Service Management*, 20(1), 44–66.