IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14415

Blockchain-Driven Decentralized Storage Network: A Secure and Scalable Alternative to Traditional Cloud Storage

Pooja Patil[1], Kshitij Nalawade[2], Parth Patil[3], Mayuresh Satam[4], Chaitanya Kardile[5]

Assistant Professor, Department of Computer Engineering, TSSM BSCOER Narhe Technical Campus, Pune, India[1]

Student, Department of Computer Engineering, TSSM BSCOER Narhe Technical Campus, Pune, India[2-5]

Abstract: A Decentralized Storage Network (DSN) that combines blockchain technology with peer-to peer (P2P) communication to enable secure, tamper-proof file storage across distributed nodes. This system ensures data confidentiality and integrity by splitting files into encrypted chunks, redundantly storing them across the network, and recording storage commitments on an immutable blockchain ledger. This System achieves P2P distributed storage system which does not have single point of failure, and tamper-proof file storage with intuitive design and ease of access. The data chunks are duplicated and stored on multiple nodes for fault tolerance and leveraging Proof of storage consensus for Security and Validating Nodes and rewarding them. This network include a lightweight Merkle root-based transaction validation mechanism, UDP hole-punching for NAT traversal, and AES-256 encryption for data confidentiality.

Keywords: Decentralized Storage Network (DSN), Blockchain-based Storage, Peer-to-Peer (P2P) Networking, Tamper-proof File Storage.

I. INTRODUCTION

In today's digital landscape, data has become one of the most valuable assets, yet the systems we rely on to store and manage it remain vulnerable. High-profile data breaches, service outages, and opaque control structures have revealed the limitations of centralized cloud storage, raising serious concerns about security, availability, and user autonomy.

Traditional cloud architectures rely on centralized servers that act as single points of control and failure. This model introduces systemic risks, including susceptibility to cyberattacks, loss of data availability during outages, and limited transparency in how data is managed and accessed. Emerging technologies such as blockchain and peer-to-peer (P2P) networking offer promising alternatives by enabling decentralized systems that distribute both control and responsibility. This research addresses the critical gap in storage and it's security by proposing a decentralized storage network (DSN) that eliminates reliance on centralized intermediaries, data leaks, man in middle attacks, etc. Although previous studies have explored blockchain-based storage and P2P communication independently, few have combined these approaches into a cohesive, permissionless architecture that is both secure and efficient in real-world conditions.

Our work builds upon existing efforts in decentralized systems by integrating a blockchain layer for immutable transaction recording with a P2P protocol for data storage. Notable contributions of our system include:

1. A permissionless blockchain that securely logs storage transactions and data ownership without central authority.
2. A chunked file storage mechanism that ensures cryptographic data integrity while storing multiple copies of chunks across distributed nodes ensuring fault tolerence.
3. NAT traversal using UDP hole-punching, allowing seamless P2P connectivity even across restricted network environments.

This research is significant not only for its technical contributions but also for its potential to reshape how individuals and organizations think about data ownership and availability. By decentralizing storage infrastructure and enhancing transparency and security, this system provides a foundation for secure, tamper-proof data storage.

The remainder of this paper is organized as follows: Section 2 presents related work and contextualizes our approach within current literature. Section 3 details the system design, including architecture and protocol mechanisms. Section 4 discusses implementation strategies and experimental setup. Section 5 evaluates performance based on key metrics. Finally, Section 6 concludes with insights and directions for future work.

II. RELATED WORKS

The integration of blockchain technology into decentralized storage systems has attracted growing attention in recent years, driven by the need for tamper-proof data management and trustless coordination. Nakamoto's seminal work on Bitcoin [4] introduced the decentralized ledger model, laying the foundation for secure, verifiable transactions without a central authority. Building upon this, Park et al. [1] explored the role of blockchain in enhancing cloud security, particularly through transparent auditing and distributed consensus mechanisms. In alignment with the objectives of our DSN architecture, Li et al. [2] proposed Block-Secure, a peer-to-peer (P2P) cloud storage framework that utilizes blockchain to ensure data ownership and integrity, demonstrating the feasibility of integrating blockchain into distributed storage models.

The role of P2P networking in enabling scalable and fault-tolerant distributed storage has also been well established. Parameswaran et al. [5] highlighted the core principles of P2P architectures for decentralized data sharing, which underpin our DSN's use of UDP holepunching for direct node-to-node communication. More recently, Wang et al. [8] presented a blockchain-based public auditing scheme for P2P-shared data, reinforcing the importance of transparency and integrity in decentralized environments.

Security and privacy, particularly through cryptographic techniques, remain central challenges in cloud and decentralized storage. Kaaniche et al. [7] emphasized the importance of AES-256 and SHA-256 for protecting sensitive data, a practice adopted in DSN's encryption workflow. Complementing this, Li et al. [6] proposed an intelligent cryptographic model that balances strong encryption with performance efficiency in distributed systems, addressing trade-offs relevant to our chunked storage and retrieval mechanisms.

Maintaining data integrity and enabling verifiable storage are also key concerns. Cao et al. [3] demonstrated how blockchain can be leveraged to ensure the immutability of electronic health records in eHealth systems, a concept conceptually mirrored in DSN's Merkle root-based transaction validation approach. Furthering data access control, Zhang et al. [10] introduced attribute-based encryption schemes to preserve user privacy while supporting finegrained decryption principles we incorporate in our secure access mechanisms.

In addition to security and trust, performance remains a critical factor in the viability of decentralized storage systems. Azhir et al. [9] provided a systematic study of query optimization strategies in cloud environments, offering valuable insight into efficient metadata indexing and retrieval concepts directly relevant to DSN's chunk management and file reconstruction processes. Collectively, these works inform the design and development of our Decentralized Storage Network, which aims to bridge the gap between secure, transparent storage and practical performance in trustless P2P environments.

III. PROPOSED SYSTEM

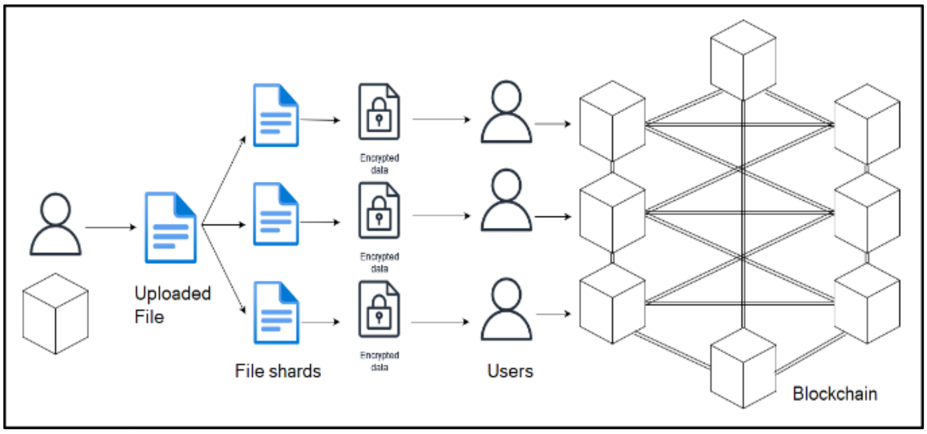

Fig. 1. System Overview

The proposed Blockchain-based Distributed Storage System offers a secure, transparent, and scalable alternative to traditional cloud storage by merging the robust security features of blockchain technology with the efficiency of peer-to-peer (P2P) networking.

DOI: 10.17148/IJARCCE.2025.14415

This decentralized approach mitigates the weaknesses found in centralized storage, such as single points of failure, susceptibility to data breaches, and opaque management by distributing file storage across numerous independent nodes. Each node contributes to the network's overall resilience, offering a system where data integrity, privacy, and availability are paramount. As shown in Figure 1.

At the foundation of our system is a dedicated blockchain layer that maintains an immutable and transparent ledger of all file storage transactions. The design of this blockchain involves two primary modules:

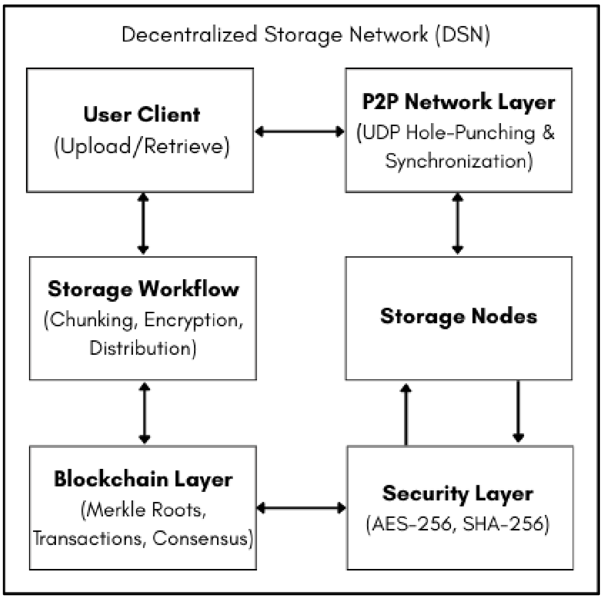
Decentralized Storage Network (DSN)
User Client
(Upload/Retrieve)
P2P Network Layer
(UDP Hole-Punching & Synchronization)
Storage Workflow
(Chunking, Encryption, Distribution)
Storage Nodes
Blockchain Layer
(Merkle Roots, Transactions, Consensus)
Security Layer
(AES-256, SHA-256)

Fig. 2. Block Diagram

A. Blockchain Layer

1) Block Structure

Each block in the blockchain is constructed with a well-defined structure to ensure both security and verifiability:
Index: Indicates the sequential position of the block in the chain.
Timestamp: Records the creation time of the block.
PreviousHash: Stores the SHA-256 hash of the preceding block, ensuring that blocks are cryptographically linked.
Transactions: Contains a list of Storage Commitment Transaction objects, which hold metadata about the encrypted file shards.
MerkleRoot: The root hash of all transactions, computed via pairwise SHA-256 hashing to allow for rapid integrity verification.
BlockHash: A final SHA-256 hash generated from the block's header data (index, timestamp, previous hash, and Merkle root), which secures the block against tampering.

2) Blockchain Management

The blockchain starts with a Genesis Block (Index = 0 with empty transactions). New blocks are added when the number of pending transactions surpasses a specific threshold (e.g., more than two transactions). This new block links to the previous block via its PreviousHash, reinforcing the chain's integrity and preventing unauthorized modifications. Nodes use simple, UDP-based synchronization commands to share the latest blockchain state, achieving consensus by independently re-computing Merkle roots and verifying block hashes.

B. Storage Workflow Layer

This layer is responsible for the secure management of file uploads and distribution. It comprises file splitting, encryption, and metadata recording:

DOI: 10.17148/IJARCCE.2025.14415

1) File Splitting & Encryption:

Chunking: Files are segmented into uniform shards, typically of 256 KB each, to improve distribution efficiency and allow parallel processing. This process ensures that large files are handled robustly and that redundancy is built into the system loss or corruption of one shard does not result in complete data loss.

Encryption: Every shard is encrypted with AES-256 in CBC mode. A master encryption key, generated at user registration and stored securely, ensures that only authorized parties can decrypt these shards.

Hashing: For each encrypted shard, a SHA-256 hash is computed. These hashes are recorded in objects, which are subsequently added to the blockchain for immutable auditability.

2) Distributed Storage:

Chunk Distribution: Once encrypted, shards are sent to reputed nodes filtered by proof of storage consensus. The node selection process ensures load balancing and resilience.

UDP Hole-Punching: To enable direct P2P communication, especially for nodes behind NATs or firewalls, the system employs UDP hole-punching. This method involves nodes sending periodic “keepalive” packets to a public server to establish bidirectional communication channels.

Metadata Recording: Each shard’s associated metadata— its hash, the storage node’s ID, and a timestamp is recorded as part of the storage transaction. This metadata is then integrated into the blockchain, enabling later verification and retrieval.

C. P2P Network Layer

The P2P network layer orchestrates node communication and ensures timely synchronization across the distributed system:

1) Node Communication:

Direct Messaging: The system relies on UDP-based protocols to send and receive data. Large messages (like full blockchain updates or encrypted chunks) are divided into 1200-byte packets and reassembled using a unique message identifier and chunk index.

UDP Commands: Key commands such as SAVESHARD (for transferring shards), ADDTRANSACTION (for broadcasting new transactions), and DOWNLOADBC (for retrieving the latest blockchain) facilitate efficient operation and coordination among nodes.

2) Network Synchronization:

Nodes periodically retrieve and validate the longest available chain, ensuring consistency across the network and mitigating risks associated with network partitions or delayed communications.

D. Security Layer

Security is integrated throughout the system, ensuring that both data confidentiality and integrity are maintained:

1) Cryptographic Protocols:

AES-256 Encryption: Protects file contents from unauthorized access.

SHA-256 Hashing: Validates the integrity of file shards and blocks.

Merkle Trees: Allow for efficient verification of aggregated transaction data and ensure that blocks remain tamper-proof.

2) Authentication & Node Management:

Upon login via a central server, users are authenticated and assigned a unique node address.

Active nodes are discovered and monitored via HTTP-based queries ensuring that only legitimate nodes participate in the decentralized storage system.

E. File Upload and Retrieval Workflow

During file upload, the system performs the following steps:

DOI: 10.17148/IJARCCE.2025.14415

1) File Upload and Sharding:
The user's file is split into encrypted shards. Each shard is individually hashed, and its metadata is recorded. Shards are duplicated and distributed to nodes with good reputation in the network which is generated and adjusted using proof of storage algorithm. Metadata is submitted as storage transactions and added to the blockchain.

2) File Retrieval and Reconstruction:
The system uses the blockchain as a verifiable ledger to locate the data required for file reconstruction. Each uploaded file has an associated metadata file that contains the ordered list of shard hashes; this metadata is stored on-chain during upload.

To retrieve a file, the system:
Traverses the blockchain to identify the metadata file linked to the requested main file.
Extracts the shard hashes listed in the metadata.
Sends retrieval requests to the nodes responsible for storing those shards.
Participating peer nodes respond by transmitting the corresponding encrypted shard data.
The received shards are verified using their hashes, assembled in the correct sequence, as specified by the metadata file, decrypted and merged to reconstruct the original file.

The system's strength lies in its blend of innovative strategies:
Lightweight Blockchain Implementation: Avoids energy-intensive mining by generating blocks based on transaction thresholds.
Efficient NAT Traversal: Utilizes UDP hole-punching to ensure robust P2P communications.
Decentralized Integrity Checks: Implements Merkle tree-based verification to maintain a tamper-proof record without centralized oversight.

Overall, by integrating a blockchain layer, efficient storage workflow, and resilient P2P network protocols, the proposed system addresses the critical need for secure, decentralized file storage. This approach not only enhances data privacy and integrity but also provides a scalable and transparent platform suited for a wide range of applications.

IV. DISCUSSION

Our decentralized storage system combines blockchain and P2P communication to create a secure, transparent alternative to centralized cloud storage. Design minimizes computational overhead while maintaining data integrity through cryptographic hashing and Merkle tree verification.The system's NAT tolerance, achieved via UDP hole-punching, ensures reliable direct communication even for nodes behind firewalls, thereby increasing network robustness, ensuring confidentiality and integrity.However, the linear growth of the blockchain poses scalability challenges as storage and synchronization demands increase over time, potentially leading to longer validation cycles.

Additionally, while the lightweight consensus is efficient, it may not be as robust in adversarial conditions compared to traditional protocols like Proof of Work.Our approach directly addresses gaps in existing research by eliminating centralized vulnerabilities and enhancing transparency through immutable transaction recording.Moreover, our integration of efficient file sharding and secure P2P networking provides a practical solution to overcome limitations seen in prior systems, making the way for further optimization and scalability improvements in decentralized storage technologies.

V. CONCLUSION

In conclusion, the development of a Decentralized Storage Network represents a significant step toward a more secure, transparent, and user-empowered data storage paradigm. This project requires an in-depth understanding of blockchain technology, distributed systems, and smart contracts to successfully address inherent challenges. Our system demonstrates a promising approach that mitigates the vulnerabilities of traditional centralized storage methods.

The proposed system empowers users by granting them greater control and ownership over their data, while simultaneously ensuring data privacy and security through advanced cryptographic techniques. It moves beyond conventional storage frameworks by eliminating single points of failure, providing a tamper-resistant audit trail, and facilitating secure, direct communication between network nodes.

DOI: 10.17148/IJARCCE.2025.14415

A well-planned methodology is critical to this effort. Our work emphasizes the importance of selecting an appropriate technology stack. Conducting a detailed feasibility study allows us to identify potential risks, evaluate benefits, and plan the necessary budget to accommodate complex operations and future scalability challenges, such as managing the linear growth of the blockchain.

Ultimately, the successful realization of a Decentralized Storage Network can significantly impact the future of data storage and security. It creates a decentralized alternative that not only increases the reliability and efficiency of data management but also opens up new opportunities for businesses and individuals. Our research lays the groundwork for future advancements in decentralized storage technology and highlights a path toward building a more secure and efficient digital infrastructure.

REFERENCES

[1]. J.H. Park et al., Blockchain security in cloud computing: use cases, challenges and solutions (2017)
[2]. J. Li et al., Block-Secure: blockchain based scheme for secure P2P cloud storage (2018)
[3]. S. Cao et al., Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain (2019)
[4]. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
[5]. M. Parameswaran et al., P2P Networking: An Information Sharing Alternative (2001)
[6]. Y. Li et al., Intelligent cryptography approach for secure distributed big data storage in cloud computing (2017)
[7]. N. Kaaniche et al., Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms (2017)
[8]. H. Wang et al., Blockchain-based public auditing scheme for shared data (2017)
[9]. E. Azhir et al., Query optimization mechanisms in the cloud environments: a systematic study (2019)
[10]. Yinghui Zhang et al., Ensuring attribute privacy protection and fast decryption for outsourced data security (2017)