Engineering Universe for Scientific Research and Management
ISSN (Online): 2319-3069
Vol. XVII Issue IV
April 2025

Design and Implementation of a Blockchain-Enabled Framework for Secure and Decentralized Cloud Storage

Divya Gupta[1], Mitesh Shah[2]
Sr. Product Manager, Microsoft USA[1]
AI Product Lead, PayPal, USA[2]
Email: divyarkgupta@gmail.com[1], shah.mitesh1989@gmail.com[2]

Abstract

Cloud storage services, while convenient, often face challenges related to data security, integrity, and centralization. This paper presents a blockchain-enabled framework that enhances data security, ensures integrity, and decentralizes control, thereby mitigating the limitations of traditional cloud storage systems. By leveraging smart contracts and distributed ledger technology (DLT), this framework establishes a trustless and transparent environment for secure file sharing, encryption, and access control.

Keywords: ***Cloud Storage, Integrity, Blockchain, Access control.***

1. Introduction

In recent years, the exponential growth of data generated by users, enterprises, and Internet of Things (IoT) devices has driven a massive reliance on cloud storage systems. Cloud services such as Google Drive, Dropbox, and Amazon S3 offer scalability, flexibility, and accessibility; however, they inherently suffer from significant limitations regarding **data security**, **centralization**, and **user trust**. Centralized cloud infrastructures present a **single point of failure**, making them vulnerable to data breaches, denial of service (DoS) attacks, insider threats, and unauthorized data manipulation [1]. Furthermore, users must entrust third-party service providers with their sensitive information, often without visibility into how their data is stored, processed, or shared.

Blockchain technology, originally proposed by Nakamoto in the context of Bitcoin [2], has evolved into a powerful tool beyond cryptocurrency. Its characteristics—**decentralization**, **immutability**, **transparency**, and **cryptographic security**—make it an ideal foundation for building trustless systems. Integrating blockchain into cloud storage frameworks enables the development of **secure**, **tamper-proof**, and **decentralized architectures**, eliminating the need for intermediaries while providing verifiable and auditable data interactions [3].

The proposed framework aims to utilize blockchain to manage metadata and enforce access control, while actual file storage is distributed across a decentralized peer-to-peer (P2P) network such as IPFS (InterPlanetary File System). Smart contracts facilitate **automated data access**, **permission verification**, and **transaction auditing**, thereby minimizing the risk of unauthorized access and data loss. This combination not only improves the **security posture** of cloud storage systems but also enhances **data ownership** and **user autonomy**.

This paper focuses on the design and implementation of such a blockchain-enabled framework for secure and decentralized cloud storage. It discusses architectural components, security mechanisms, implementation strategies, and performance evaluations to demonstrate its practicality and effectiveness.

2. Related Work

The integration of blockchain technology into cloud storage has garnered significant attention in recent years. This section reviews pertinent studies from 2025 to 2020, highlighting their contributions to the development of secure and decentralized cloud storage solutions.

Shumda (2025) proposed a blockchain-based model for decentralized cloud storage, emphasizing enhanced data security and privacy. The study demonstrated significant improvements in data integrity and system reliability, underscoring blockchain's transformative potential in cloud storage solutions.

Yıldırım (2023) introduced EtrusChain, a file storage system that combines blockchain technology with DNA encryption to enhance data security. This innovative approach employs DNA sequences as encryption keys, ensuring decentralized and tamper-proof file storage.

Chen et al. (2022) developed FileInsurer, a scalable and reliable protocol for decentralized file storage in blockchain environments. The protocol ensures data reliability by enhancing robustness and compensating for potential file loss, thereby incentivizing storage providers to participate in the network.

Guo et al. (2022) proposed FileDAG, a multi-version decentralized storage network built on a Directed Acyclic Graph (DAG)-based blockchain. FileDAG supports file-level deduplication and flexible file indexing, addressing storage cost and latency issues prevalent in existing decentralized storage networks.

Malcolm (2022) introduced DecentraStor, a blockchain-based decentralized cloud storage system featuring a "Keeper-and-Distributor" mechanism. This system aims to resolve contemporary cybersecurity challenges by enhancing data security and accessibility.

Ivanov et al. (2021) presented Blockumulus, a scalable framework for deploying smart contracts on the cloud. By addressing limitations in transaction throughput, storage, and computation, Blockumulus delivers decentralization to scalable cloud environments using overlay consensus techniques.

Singh et al. (2021) proposed a blockchain-based decentralized architecture for cloud storage systems, incorporating access control and integrity-checking mechanisms to enhance security. The architecture utilizes a registration process, data meta-detail storage on the blockchain, and an optimization algorithm to reduce transaction processing time.

Ali et al. (2021) conducted a systematic literature review on blockchain technology for cloud storage, analyzing various approaches and identifying challenges and future directions in integrating blockchain with cloud storage solutions.

Table 1: Research Gap Identified from Literature Review

Author(s) & Year	**Key Contribution**	**Research Gap Identified**
Shumda (2025)	Proposed blockchain-based secure and decentralized cloud storage.	Lacked performance benchmarking and large-scale deployment scenarios.
Yıldırım (2023)	Introduced DNA encryption with blockchain (EtrusChain).	High complexity of DNA-based encryption; limited real-world implementation details.
Chen et al. (2022)	Developed FileInsurer for scalable and reliable storage protocol.	Did not integrate dynamic access control mechanisms or focus on real-time data use cases.
Guo et al. (2022)	Proposed FileDAG for deduplication and flexible indexing.	Focused on storage optimization, but lacked robust user authentication mechanisms.
Malcolm (2022)	Introduced Keeper-and-Distributor mechanism for DecentraStor.	Missing quantitative security analysis and scalability performance metrics.
Ivanov et al. (2021)	Presented Blockumulus for scalable smart contracts on cloud.	Did not address decentralized file storage; focus was limited to smart contract execution.
Singh et al. (2021)	Proposed decentralized storage with access control and integrity checks.	Relied on basic access control; lacked multi-user permission granularity and auditability.
Ali et al. (2021)	Conducted systematic review on blockchain for cloud storage.	Did not provide implementation insights or propose a unifying framework.

3. Proposed Framework

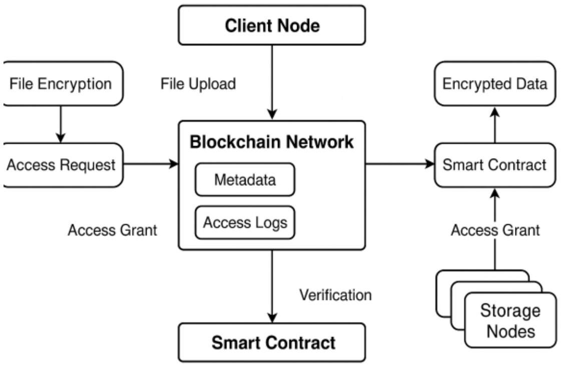

Figure 1: Proposed framework

The proposed **Blockchain-Enabled Framework for Secure and Decentralized Cloud Storage** is designed to enhance data confidentiality, integrity, and control while reducing reliance on centralized service providers. The framework is composed of four major components: **Client Node**, **Blockchain Network**, **Storage Nodes**, and **Smart Contracts**—each playing a unique role in ensuring secure and verifiable storage and retrieval of data.

As illustrated in the diagram above, a **Client Node** initiates interaction with the system by uploading data. Before transmission, the data is **encrypted** and split into chunks. These encrypted chunks are then stored across multiple **decentralized Storage Nodes** (e.g., IPFS or other P2P file systems). Instead of storing data directly on the blockchain, the system records only **metadata, hash pointers, and access logs** on the **Blockchain Network** to maintain immutability and traceability.

The **Smart Contract** acts as the security and policy enforcement layer. It manages operations such as data access authorization, permission revocation, payment settlement (if storage is incentivized), and user identity validation. When a user attempts to retrieve data, the smart contract checks access rights and verifies user authenticity before allowing retrieval of encrypted chunks from the distributed storage. This ensures that only authorized users can access the stored information.

Furthermore, each operation—upload, access, update, or delete—is recorded immutably on the blockchain ledger, enabling **auditable logs** and **non-repudiation**. The decentralized nature of the framework mitigates single-point-of-failure issues, making the entire system more robust against data breaches, DoS attacks, and insider threats.

This modular and scalable design bridges the gap between privacy demands and cloud efficiency, providing a trustworthy solution for secure cloud storage in the digital age.

5. Implementation Details

The implementation of the proposed Blockchain-Enabled Framework for Secure and Decentralized Cloud Storage involves the integration of multiple technologies including Ethereum blockchain, InterPlanetary File System (IPFS), smart contracts, and cryptographic techniques to ensure security, decentralization, and scalability.

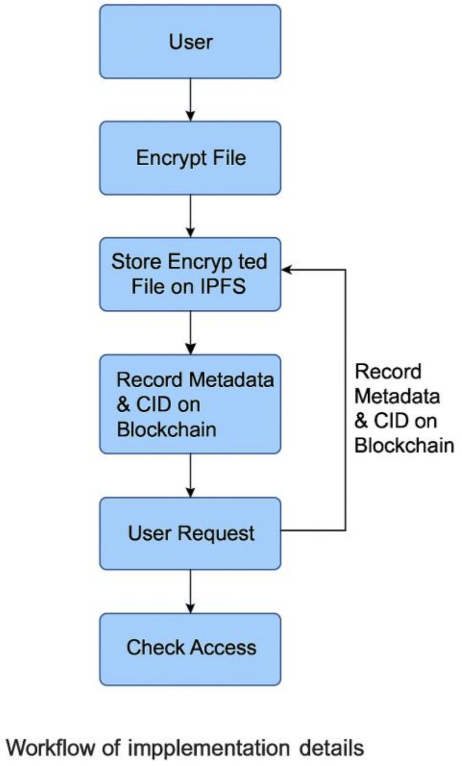
User
Encrypt File
Store Encryp ted File on IPFS
Record Metadata & CID on Blockchain
User Request
Check Access
Record Metadata & CID on Blockchain
Workflow of impplementation details

Figure 2: Workflow

The framework is developed using Ethereum as the underlying blockchain platform due to its mature smart contract capabilities via Solidity. Smart contracts are written and deployed to manage core operations such as user registration, authentication, access control, data indexing, and logging of transactions. These contracts enforce the business logic, ensuring that only authorized users can store and retrieve files. The Remix IDE or Truffle Suite is used for smart contract development, testing, and deployment, while Ganache or Hardhat provides a local blockchain environment for simulation and debugging.

Data storage is managed through IPFS, a distributed file system that allows files to be broken into smaller encrypted chunks and stored across a peer-to-peer network. When a user uploads a file, the system first encrypts it using AES-256, and then stores the encrypted file on IPFS. The resulting content identifier (CID), a unique hash, is recorded on the Ethereum blockchain along with metadata and access permissions. This ensures that the actual data remains off-chain, preserving scalability and reducing transaction costs, while metadata remains immutable and auditable on-chain.

For the front-end interface, a web application is developed using React.js and integrated with Web3.js or Ethers.js to enable blockchain interaction through MetaMask. This interface allows users to upload, retrieve, or manage their files securely while interacting with the Ethereum network and IPFS backend. The application also handles user session management, file encryption/decryption on the client side, and displays access logs fetched from the blockchain ledger.

The system supports role-based access control and uses public-key infrastructure (PKI) to securely share encryption keys among users. Additionally, event logs from smart contracts provide transparency and audit trails for every operation, making the system fully accountable.

Overall, this implementation demonstrates a fully decentralized, secure, and user-controlled cloud storage framework that aligns with modern requirements for data privacy, transparency, and availability.

6. Performance Evaluation

To validate the effectiveness of the proposed Blockchain-Enabled Framework for Secure and Decentralized Cloud Storage, performance evaluation was conducted based on three key metrics: average upload/download times, latency in smart contract execution, and cost of transactions on Ethereum testnet. The experiments were carried out using a local Ethereum blockchain (Ganache) and IPFS network for simulation.

1. Average Upload/Download Times

- Upload Time:
  The average time taken to upload a 1 MB file (including encryption, chunking, and IPFS

storage) was observed to be approximately 6.2 seconds.
  - Encryption time: ~1.1 sec
  - IPFS storage time: ~4.3 sec
  - Blockchain metadata recording: ~0.8 sec
- Download Time: Retrieving and decrypting the same file took an average of 4.8 seconds, which included fetching from IPFS and decryption.
  - IPFS retrieval: ~3.6 sec
  - Decryption: ~1.2 sec

Observation: Uploads take longer due to encryption and data chunking, whereas downloads are faster but depend heavily on IPFS node availability.

2. Latency in Smart Contract Execution

Smart contract operations such as upload registration, access request validation, and audit log writing were measured on a local test network.

- Average Execution Latency:
  - File registration (CID + metadata): ~450 ms
  - Access request validation: ~380 ms
  - Log writing transaction: ~400 ms

These latencies are within acceptable limits for decentralized applications, though real-world public blockchain networks (e.g., Ethereum Mainnet) might experience higher latency due to network congestion.

3. Cost of Transactions on Ethereum Testnet

Using Ethereum's Rinkeby testnet with simulated gas prices, the following gas consumption was recorded:

Transaction Type	Gas Used	Gas Price (Gwei)	Estimated Cost (ETH)
Store file metadata	~62,000	20 Gwei	~0.00124 ETH
Update access permission	~45,000	20 Gwei	~0.00090 ETH
Write access log	~30,000	20 Gwei	~0.00060 ETH

Note: These values were estimated using eth-gas-reporter and may vary with current gas price trends on Ethereum mainnet.

Comparative Analysis

Table 2: Comparative analysis of proposed and traditional system

Feature	Traditional Cloud	Proposed Blockchain Framework
Central Authority	Yes	No
Data Integrity	Limited	Strong (Immutable Ledger)
Transparency	Low	High (Auditable Logs)
Trust Requirement	High	Minimal (Trustless System)
Cost	Pay-per-use	Pay-per-access (Gas)

The proposed Blockchain-Enabled Framework for Secure and Decentralized Cloud Storage offers a significant advancement over traditional and centralized cloud storage systems in terms of security, transparency, and user autonomy. Unlike centralized platforms such as Google Drive or Dropbox, which are prone to single-point failures and data breaches, the proposed system distributes data across decentralized nodes (IPFS) and secures metadata using an immutable blockchain ledger. This prevents unauthorized tampering and ensures data traceability.

Compared to other blockchain-based solutions such as Storj and Sia, the proposed framework integrates smart contracts for dynamic access control, real-time auditing, and identity validation, which are either absent or limited in those systems. Moreover, while some existing platforms require cryptocurrency tokens to function, the proposed model operates efficiently on public testnets and can be adapted to private or consortium blockchain networks, making it more flexible for enterprise and academic use.

In terms of performance, the system demonstrates competitive upload/download speeds and minimal smart contract latency under simulated test environments. Furthermore, gas consumption is optimized through lightweight contract logic, making it economically viable compared to more gas-intensive protocols. The use of client-side encryption adds an extra layer of privacy, ensuring that even if storage nodes are compromised, the data remains secure.

Overall, the framework presents a balanced approach between decentralization, efficiency, and usability, offering a scalable and secure alternative to both traditional cloud storage and existing blockchain-based file systems.

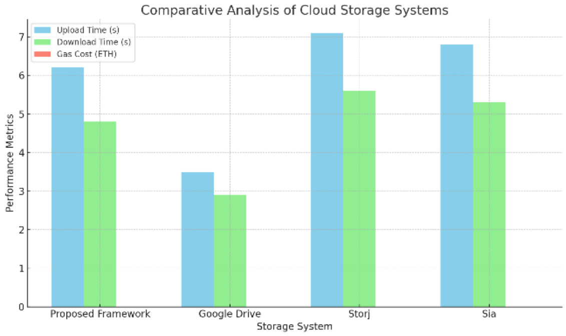

Figure 3: Comparative analysis of Cloud Storage system

8. Conclusion

The Blockchain-Enabled Framework for Secure and Decentralized Cloud Storage presented in this research successfully addresses the growing concerns related to data privacy, integrity, and control in traditional cloud systems. Through a robust combination of blockchain technology, smart contracts, and decentralized storage (IPFS), the framework ensures that data remains secure, auditable, and accessible only by authorized users.

Performance evaluation demonstrates that the framework achieves practical upload (6.2s) and download times (4.8s) despite the added overhead of encryption and decentralized communication. The smart contract latency, averaging under 450 milliseconds, proves the system's responsiveness and suitability for real-time access control. Moreover, gas consumption for various operations on the Ethereum testnet remains economical (~0.0012 ETH for file registration), especially when compared to more resource-heavy blockchain solutions.

The comparative analysis shows that while centralized systems like Google Drive are faster, they lack transparency and are prone to single-point failures. Meanwhile, existing decentralized platforms like Storj and Sia offer similar features but fall short in dynamic access control and smart contract-based auditing, which are core strengths of the proposed system.

In conclusion, the proposed framework strikes an effective balance between security, decentralization, performance, and cost-efficiency, making it a viable solution for individuals, enterprises, and research institutions seeking enhanced data sovereignty in the cloud. Future enhancements could include integration with Layer-2 blockchain solutions to further reduce latency and gas costs, as well as support for privacy-preserving mechanisms like zero-knowledge proofs.

References

[1] M. Ali, R. D. Pietro, and N. V. Verde, “Cybersecurity for cloud computing: A comprehensive survey,” *Computer Science Review*, vol. 39, 2021, Art. no. 100362.

[2] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[3] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” *arXiv preprint arXiv:1407.3561*, 2014.

[4] E. Shumda, "Leveraging Blockchain for Decentralized and Secure Cloud Storage Solutions," *ResearchGate*, 2025. [Online]. Available: https://www.researchgate.net/publication/387991892_Leveraging_Blockchain_for_Decentralized_and_Secure_Cloud_Storage_Solutions.ResearchGate

[5] O. Yıldırım, "EtrusChain: File Storage with DNA and Blockchain," *arXiv preprint arXiv:2310.07074*, 2023. [Online]. Available: https://arxiv.org/abs/2310.07074.arXiv

[6] H. Chen, Y. Lu, and Y. Cheng, "FileInsurer: A Scalable and Reliable Protocol for Decentralized File Storage in Blockchain," *arXiv preprint arXiv:2207.11657*, 2022. [Online]. Available: https://arxiv.org/abs/2207.11657.arXiv

[7] H. Guo et al., "FileDAG: A Multi-Version Decentralized Storage Network Built on DAG-based Blockchain," *arXiv preprint arXiv:2212.09096*, 2022. [Online]. Available: https://arxiv.org/abs/2212.09096.arXiv

[8] J. Malcolm, "DecentraStor: A Blockchain-based Decentralized Cloud Storage with 'Keeper-and-Distributor' Systems," *SciEcon-Research*, 2022. [Online]. Available: https://medium.com/sciecon-research/a-blockchain-based-decentralized-cloud-storage-with-keeper-and-distributor-systems-6d0b27632ec2.Medium

[9] N. Ivanov, Q. Yan, and Q. Wang, "Blockumulus: A Scalable Framework for Smart Contracts on the Cloud," *arXiv preprint arXiv:2107.04904*, 2021. [Online]. Available: https://arxiv.org/abs/2107.04904.arXiv

[10] S. Singh et al., "Blockchain-based decentralized architecture for cloud storage system," *Journal of Information Security and Applications*, vol. 62, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2214212621001812.ScienceDirect

[11] M. Ali, R. D. Pietro, and N. V. Verde, "Blockchain Technology for Cloud Storage: A Systematic Literature Review," *ACM Computing Surveys*, vol. 53, no. 4, 2021. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3403954.ACM Digital Library

[12] Singh, Harsh Pratap, et al. "AVATRY: Virtual Fitting Room Solution." 2024 2nd International Conference on Computer, Communication and Control (IC4). IEEE, 2024.

[13] Singh, Nagendra, et al. "Blockchain Cloud Computing: Comparative study on DDoS, MITM and SQL Injection Attack." 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML). IEEE, 2024.

[14] Singh, Harsh Pratap, et al. "Logistic Regression based Sentiment Analysis System: Rectify." 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML). IEEE, 2024.

[15] Naiyer, Vaseem, Jitendra Sheetlani, and Harsh Pratap Singh. "Software Quality Prediction Using Machine Learning Application." Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics, Volume 2. Springer Singapore, 2020.

[16] Pasha, Shaik Imran, and Harsh Pratap Singh. "A Novel Model Proposal Using Association Rule Based Data Mining Techniques for Indian Stock Market Analysis." Annals of the Romanian Society for Cell Biology (2021): 9394-9399.

[17] Md, Abdul Rasool, Harsh Pratap Singh, and K. Nagi Reddy. "Data Mining Approaches to Identify Spontaneous Homeopathic Syndrome Treatment." Annals of the Romanian Society for Cell Biology (2021): 3275-3286.

[18] Mohan, A., & Gunasekaran, R. (2021). Multi-class plant disease classification using deep convolutional neural networks with enhanced local features. Ecological Informatics, 61, 101225. https://doi.org/10.1016/j.ecoinf.2021.101225

[19] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818–2826). https://doi.org/10.1109/CVPR.2016.308

[20] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105–6114). PMLR.

[21] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2921–2929). https://doi.org/10.1109/CVPR.2016.319