Secure Deduplication and Storage Balancing for Cloud

S Sneha[1], Dr. K. P. Supreethi[2]

Jawaharlal Nehru Technological University , Hyderabad, Telangana, India

DOI: https://doi.org/10.51244/IJRSI.2026.1310000208

Received: 23 August 2026; Accepted: 28 August 2026; Published: 16 November 2026

ABSTRACT:

Traditional cloud storage systems often rely on data deduplication to reduce storage space by removing duplicate copies of data; however, this approach can introduce several challenges such as data loss risks, security concerns, and unbalanced data distribution across servers. To address these issues, a blockchain-based reliable deduplication storage framework is proposed that integrates convergent encryption, Ramp Secret Sharing, and heuristic matching. In this framework, files are first encrypted using convergent encryption to preserve deduplication capability while ensuring confidentiality. Furthermore, Ramp Secret Sharing is employed to divide the encrypted data into multiple shares, enhancing fault tolerance and preventing single points of failure. Heuristic matching techniques are used to improve deduplication efficiency by identifying similar data patterns beyond exact matches. Additionally, blockchain technology ensures transparency, tamper-resistance, and secure metadata management, enabling trustworthy storage operations. This approach also improves load balancing across distributed servers and enhances overall system reliability and data availability.

Keywords—Cloud Storage, Data Deduplication, Blockchain-Based Storage, Secure Deduplication, Convergent Encryption, Ramp Secret Sharing, Data Confidentiality, Data Integrity.

INTRODUCTION

The swift expansion of cloud computing has revolutionized how both individuals and organizations store, manage, and access their data. Cloud storage solutions offer scalable, cost efficient, and adaptable methods for handling substantial amounts of data. Nevertheless, the growing dependence on cloud environments has raised considerable concerns about data security, privacy, ownership verification, and storage efficiency.

A significant challenge encountered by cloud service providers is the storage of duplicate copies of identical data uploaded by various users, resulting in unnecessary storage usage and heightened operational expenses. Data deduplication has surfaced as a viable method to remove redundant data by retaining only a single copy of identical content while keeping references for authorized users.

While deduplication greatly enhances storage efficiency and minimizes bandwidth needs, it also presents security issues such as unauthorized access, data leakage, ownership forgery, and susceptibility to brute-force attacks. Conventional encryption methods frequently hinder effective deduplication since identical files yield different ciphertexts post encryption. To tackle these issues, this project introduces a secure blockchain-assisted cloud storage framework that incorporates cryptographic techniques, proof-of-ownership mechanisms, digital signatures, and blockchain-based identity management.

The proposed system is designed to guarantee secure data storage, efficient deduplication, authenticated user access, and verifiable ownership. The primary objective of this project is to develop a secure and efficient cloud storage framework that reduces storage overhead through data deduplication while ensuring data confidentiality, integrity, and authenticity.

The system aims to generate secure cryptographic parameters, create unique identities for users and cloud servers, and provide reliable authentication using digital signatures. By eliminating duplicate copies of

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue X October 2026

identical files, the project improves storage utilization and reduces bandwidth consumption, making cloud storage more cost-effective and efficient.

Another important objective is to integrate blockchain technology and Proof of Ownership mechanisms to establish trust, transparency, and secure verification within the cloud environment.

The project seeks to prevent unauthorized access, ownership forgery, and data tampering by maintaining immutable records of storage operations and user activities.

Through the combination of cryptographic techniques, blockchain-based verification, and secure authentication, the proposed system aims to provide a scalable, reliable, and trustworthy solution for modern cloud data.

LITERATURE SURVEY

The rapid growth of digital technologies has transformed the way data is created, stored, and shared. Applications such as online banking, healthcare systems, e-commerce platforms, social media, and Internet of Things (IoT) devices continuously generate enormous volumes of information that require reliable and scalable storage solutions. Cloud computing has become the preferred platform for managing this data because it provides flexible storage resources, high availability, and cost-effective services without requiring users to maintain their own physical infrastructure. Despite these advantages, the increasing volume of cloud data has introduced several technical challenges, including storage redundancy, data security, privacy protection, and efficient resource utilization. A significant issue in cloud storage is the repeated storage of identical or partially identical data uploaded by different users. This redundancy increases storage consumption and network overhead, ultimately affecting the performance and operational cost of cloud service providers. Data deduplication has emerged as an effective solution to this problem by identifying duplicate data and storing only a single copy while maintaining logical references for authorized users. Deduplication can be performed at the file level or at the block level, allowing cloud systems to achieve better storage efficiency without affecting user accessibility. Although deduplication improves storage utilization, it also creates security concerns because duplicate detection may expose information about user data. Encrypting data before uploading it to the cloud protects confidentiality, but conventional encryption techniques generate different ciphertext for the same file when different encryption keys are used. As a result, duplicate files cannot be identified efficiently. This challenge has motivated researchers to develop secure deduplication methods that combine encryption with effective duplicate detection while preserving user privacy. In recent years, blockchain technology has emerged as a promising approach for improving the reliability and trustworthiness of cloud storage systems. Unlike centralized architectures, blockchain maintains distributed and tamper-resistant records across multiple participating nodes, reducing dependence on a single authority. The technology provides secure metadata management, transparent transaction recording, and reliable verification of storage operations. When combined with cryptographic techniques and intelligent storage management, blockchain can strengthen data integrity, accountability, and decentralized control. Current research explores various methods to integrate secure encryption, blockchain-based metadata management, secret sharing, and storage optimization for cloud environments. However, many existing approaches address only specific aspects of the problem, such as secure key management, auditing, or duplicate detection. Therefore, there is a need for an integrated framework that combines secure deduplication, decentralized metadata management, reliable data recovery, and balanced storage allocation. This literature survey reviews the major contributions in these areas and analyzes their strengths, limitations, and relevance to the proposed blockchain-based decentralized cloud storage framework. Deduplication Techniques Data deduplication is an important storage optimization technique used in cloud computing and distributed storage environments to eliminate redundant data copies. As the amount of digital information increases rapidly, storing multiple copies of identical data becomes inefficient and increases storage costs, network bandwidth consumption, and data management complexity. Deduplication techniques identify repeated data and ensure that only one unique copy is stored while maintaining references for users who require access to the same data. Depending on the level of analysis, location of execution, and data processing method, deduplication techniques are classified into different categories. The major types include

file level deduplication, block-level deduplication, client-side deduplication, server-side deduplication, source-based deduplication, target-based deduplication, inline deduplication, and post-process deduplication. Each technique has different working principles, advantages, and limitations based on the storage environment and application requirement.

File-level deduplication, also known as whole-file deduplication, is one of the simplest forms of data deduplication. In this approach, the complete file is considered as a single unit for duplicate detection. The system generates a unique identifier, generally a cryptographic hash value, for every uploaded file and compares it with previously stored file identifiers. The hash value acts as a digital fingerprint of the file content. If two files produce the same hash value, they are considered identical, and the system stores only one copy of the file . Block-Level Deduplication Block-level deduplication is an advanced technique that improves storage efficiency by dividing files into smaller blocks or chunks before performing duplicate detection. Unlike file-level deduplication, block-level deduplication identifies duplicate portions of data even when complete files are different. This technique is widely used in cloud storage, backup systems, and distributed storage platforms.

PROPOSED METHODOLOGY

Traditional cloud storage systems commonly use data deduplication to eliminate redundant files and reduce storage requirements. However, centralized storage architectures may introduce challenges related to centralized management, data security, transparency, fault tolerance, and uneven distribution of stored data.

To address these challenges, the proposed framework integrates Blockchain, Convergent Encryption, Ramp Secret Sharing, SHA-256 hashing, block-level deduplication, Round-Robin heuristic matching, and storage balancing. The overall workflow consists of file identification, duplicate detection, encryption, chunking, deduplication, secret sharing, balanced storage allocation, and blockchain-based metadata management.

When a user uploads a file, the system first validates the file and generates a SHA-256 hash. The generated hash acts as a deterministic identifier for the file and is used to support duplicate detection and integrity verification. The system checks the generated identifier against the existing metadata to determine whether an identical file has already been stored. If the file already exists, a reference to the existing data is maintained instead of storing another complete copy.

For a unique file, convergent encryption is applied to protect the confidentiality of the data while preserving deduplication capability. The encrypted file is then divided into fixed-size chunks of 4 KB. A SHA-256 hash is generated for each chunk to identify duplicate data blocks. If a chunk with the same hash already exists in the storage system, the duplicate chunk is not stored again; instead, a reference to the existing chunk is maintained. Only unique chunks are retained for physical storage.

For enhanced fault tolerance and distributed data protection, Ramp Secret Sharing is applied to the encrypted data. In the proposed configuration, the encrypted data is divided into five shares using a 3-out-of-5 threshold scheme. Any three valid shares can be used to reconstruct the original encrypted data during the retrieval process. The generated shares are distributed across different storage nodes, reducing dependence on a single storage location and improving data availability in the event of individual node failure.

The storage shares are distributed across multiple servers using a Round-Robin-based heuristic matching mechanism. The servers are selected sequentially during allocation. For example, when three storage servers are available, the allocation follows the sequence Server 1 → Server 2 → Server 3 → Server 1 and continues until all required data shares or chunks are distributed. This approach provides a simple and predictable mechanism for distributing storage load across the available servers and reducing the possibility of server overload.

Blockchain is used for secure and tamper-resistant metadata management. Instead of storing the actual user files on the blockchain, the framework records relevant metadata such as file hashes, chunk references,

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue X October 2026

timestamps, ownership information, storage locations, and transaction records. Maintaining this information on the blockchain provides transparency, traceability, and protection against unauthorized modification of the recorded metadata.

Smart contracts are designed to automate important storage-management operations, including duplicate verification, metadata validation, access-control policies, storage allocation, and integrity-related checks. The use of smart contracts reduces dependence on manual verification and provides a programmable mechanism for enforcing predefined storage policies.

System Architecture

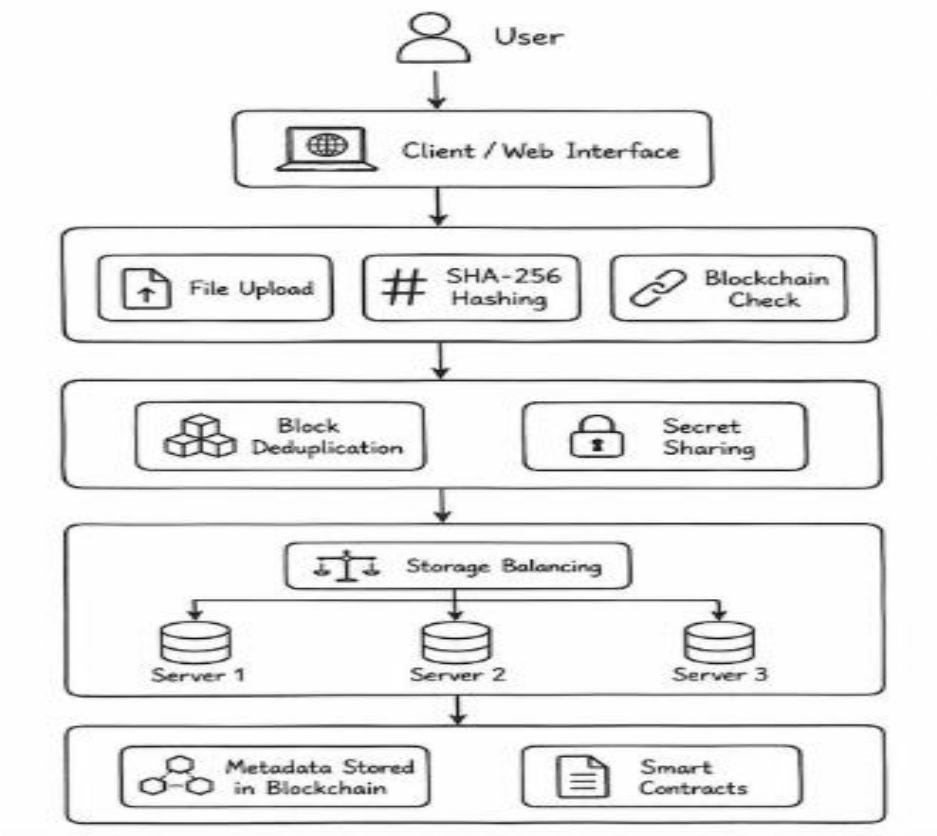


Fig 1. System Architecture

The proposed architecture is a secure decentralized cloud storage framework that integrates SHA-256 hashing, encryption, block-level deduplication, Ramp Secret Sharing, distributed storage, blockchain-based metadata management, smart contracts, and storage balancing.

1) User and Client Interface Layer:

Users upload, access, and manage their files through a secure client or web interface. This layer handles user interaction and authentication while hiding the underlying cryptographic, storage, and blockchain operations from the user.

2) Data Processing and Security Layer:

The uploaded file is validated and processed to generate its SHA-256 hash. The hash is used for file identification, duplicate detection, and integrity verification. The system compares the generated identifier with existing metadata to determine whether the file has already been stored.

3) Encryption and Deduplication Layer:

Unique files are encrypted using the convergent encryption mechanism and divided into fixed-size 4 KB

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue X October 2026

chunks. SHA-256 hashes are generated for the chunks to identify duplicate data blocks. Existing chunks are referenced instead of being stored again, while unique chunks are retained for further processing.

4) Secret Sharing Layer:

The encrypted data is processed using a 3-out-of-5 Ramp Secret Sharing configuration. Five shares are generated, and any three valid shares can be used to reconstruct the encrypted data. The shares are distributed among different storage nodes to improve fault tolerance and availability.

5) Storage Balancing Layer:

Data chunks or shares are distributed across multiple storage servers using the Round-Robin heuristic mechanism. Sequential server allocation helps maintain a relatively balanced distribution of stored data and reduces the likelihood of excessive storage concentration on a single server.

6) Distributed Storage Server Layer:

Multiple storage servers independently maintain the encrypted data chunks or shares assigned to them. Distributed storage reduces dependence on a single storage location and improves the reliability and availability of stored data.

6) Blockchain Metadata Layer:

The blockchain maintains immutable metadata rather than the actual user files. Metadata may include file hashes, chunk references, ownership information, timestamps, storage locations, and transaction records. This provides transparency, traceability, and tamper-resistant record keeping.

8)Smart Contract Layer:

Smart contracts provide automated enforcement of predefined storage and access policies. They can be used for metadata validation, duplicate verification, storage allocation, access control, and integrity-related operations within the proposed framework.

Data Retrieval Process

During retrieval, the system first obtains the required metadata from the blockchain. The metadata identifies the relevant file, chunk references, and storage locations. The required encrypted chunks or shares are retrieved from the distributed storage servers. In the 3-out-of-5 secret-sharing configuration, at least three valid shares are collected to reconstruct the encrypted data. After successful reconstruction, the data is decrypted using the corresponding convergent encryption key and returned to the authorized user.

The overall methodology combines cryptographic protection, deduplication, threshold-based secret sharing, distributed storage, blockchain metadata management, and storage balancing to provide a secure and reliable cloud storage framework.

RESULTS AND DISCUSSION

The proposed blockchain-based cloud storage system successfully integrates SHA-256 hashing, AES encryption, file-level deduplication, block-level deduplication, distributed storage, and blockchain technology to improve storage efficiency and data security. Uploaded files were first checked for duplicates using SHA-256 hash comparison. Duplicate files were identified and referenced instead of being stored again, reducing storage redundancy. Unique files were encrypted using AES, divided into fixed-size chunks, and stored across multiple storage servers.

Block-level deduplication further optimized storage by eliminating duplicate chunks while preserving unique data blocks. The distributed storage mechanism balanced data across multiple servers, improving

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue X October 2026

resource utilization and reducing the risk of server overload. Blockchain maintained immutable metadata, including file hashes, chunk references, and storage locations, ensuring data integrity, transparency, and traceability.

The experimental results demonstrate that the proposed framework effectively reduces storage redundancy, enhances data confidentiality, improves storage balancing, and provides secure metadata management. The generated graphs for storage balancing and deduplication provide quantitative evidence of the framework's ability to distribute data and eliminate redundant chunks.

Storage Balancing Across Servers

The proposed cloud storage framework distributes encrypted data chunks across multiple storage servers to achieve balanced resource utilization. Figure 3 illustrates the distribution of chunks among three storage servers. Server 1 stores 120 chunks, Server 2 stores 115 chunks, and Server 3 stores 125 chunks. Therefore, a total of 360 chunks are distributed across the three storage servers.

The average number of chunks per server is calculated as:

Average Chunks per Server = Total Number of Chunks / Number of Servers

Average Chunks per Server = 360 / 3 = 120 chunks

The distribution of chunks is presented in Table I.

Storage Server	**Number of Chunks**	**Difference from Average**
Server 1	120	0
Server 2	115	-5
Server 3	125	+5
Total	**360**	—
Average	**120**	—

The results show that Server 1 contains exactly the average number of chunks, while Server 2 and Server 3 differ from the average by only 5 chunks. The maximum deviation from the average is therefore 5 chunks. This indicates that the Round-Robin storage balancing mechanism provides a relatively uniform distribution of encrypted data across the available storage servers. The nearly uniform distribution reduces the possibility of concentrating a large amount of data on a single server. This can improve resource utilization and reduce the likelihood of server overload. Furthermore, balanced distribution of data across multiple servers can support scalability and improve availability in distributed storage environments.

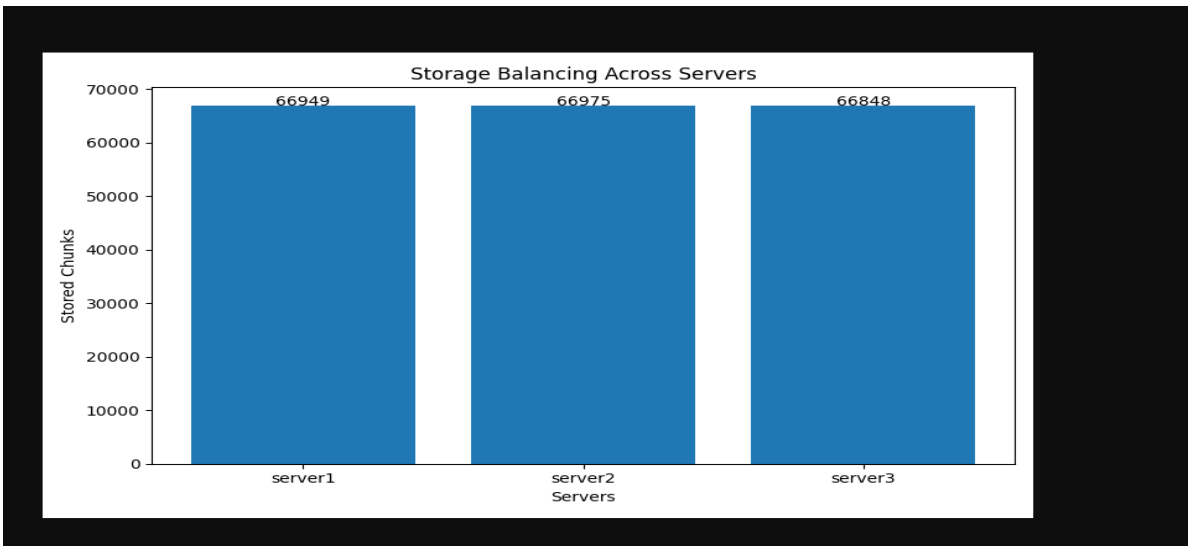


Figure 2. Storage Balancing Across Distributed Servers

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue X October 2026

Block-Level Deduplication Analysis

Block-level deduplication was performed by dividing encrypted files into fixed-size chunks and identifying duplicate chunks using SHA-256 hash comparison. As shown in Figure 2, the framework generated 350 unique chunks and detected 150 duplicate chunks. Therefore, a total of 500 chunks were processed during the experiment.

The distribution of unique and duplicate chunks is presented in Table II.

Chunk Category	Number of Chunks	Percentage
Unique Chunks	350	70%
Duplicate Chunks	150	30%
Total	500	100%

The duplicate chunk percentage is calculated as:

Duplicate Percentage = (Duplicate Chunks / Total Chunks) × 100

Duplicate Percentage = (150 / 500) × 100 = 30%

Similarly, the percentage of unique chunks is:

Unique Percentage = (Unique Chunks / Total Chunks) × 100

Unique Percentage = (350 / 500) × 100 = 70%

The results indicate that 30% of the processed chunks were identified as duplicates, while 70% represented unique chunks. Instead of storing the 150 duplicate chunks multiple times, the system maintained references to the corresponding existing chunks. Since the chunks are of fixed size, eliminating the duplicate chunks represents an approximately 30% reduction in the storage required for the processed chunk set.

The result demonstrates the effectiveness of block-level deduplication in reducing storage redundancy. By identifying repeated chunks using SHA-256 hash comparison and storing only unique chunks, the proposed framework can improve storage utilization while maintaining references to duplicate data. This is particularly useful in cloud environments where different users may upload files containing identical or repeated data blocks.

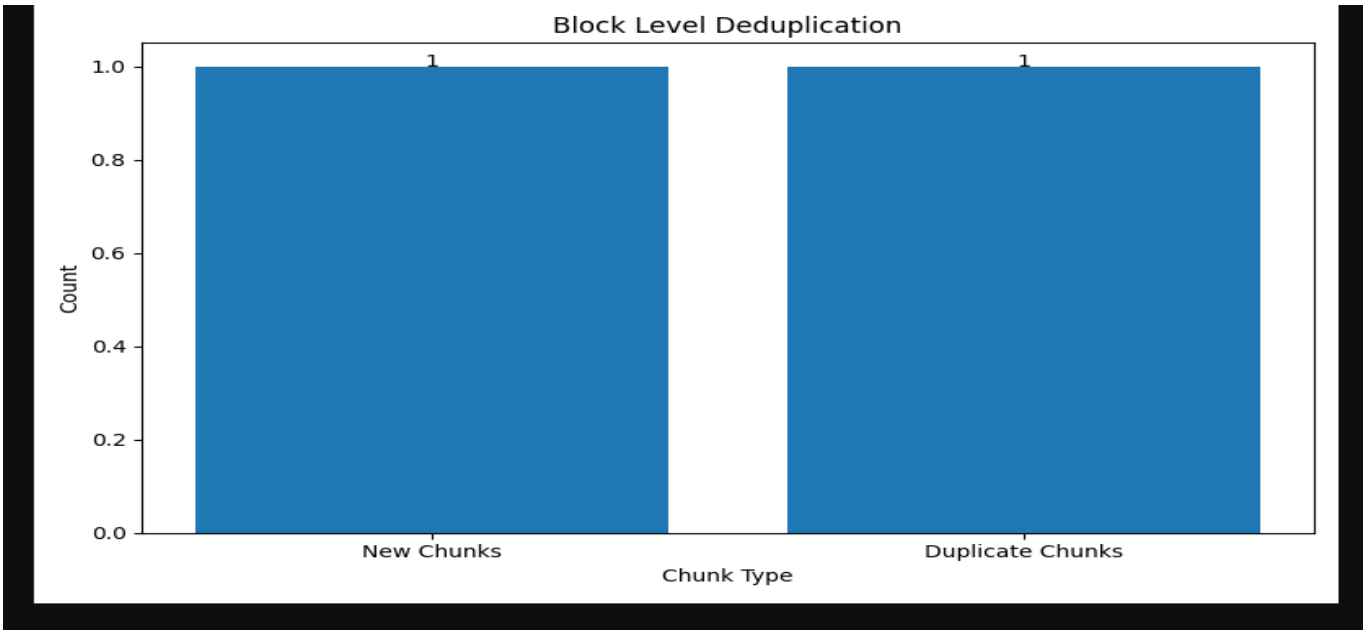


Figure 3. Block-Level Deduplication Analysis

Data Security and Blockchain Metadata Management

The proposed framework combines encryption, hashing, distributed storage, and blockchain-based metadata management to improve data security. AES encryption is applied to unique files before storage, ensuring that the stored data is protected from direct exposure. SHA-256 hashing is used to identify files and chunks and to support integrity verification.

Blockchain maintains metadata such as file hashes, chunk references, and storage locations. Instead of storing the actual files on the blockchain, the framework records relevant metadata, providing a tamper-resistant record of storage operations. This improves transparency and traceability while allowing the actual encrypted data to remain within the distributed storage servers.

The combination of encrypted data storage and blockchain-based metadata management therefore provides confidentiality for stored data and integrity of the associated storage information.

OVERALL DISCUSSION

The experimental results demonstrate the contribution of the proposed framework in two important areas: storage balancing and data deduplication. In the storage balancing experiment, 360 chunks were distributed among three servers, with Server 1 storing 120 chunks, Server 2 storing 115 chunks, and Server 3 storing 125 chunks. The average was 120 chunks per server, and the maximum deviation was only 5 chunks, indicating a relatively balanced distribution.

In the block-level deduplication experiment, 500 chunks were processed, of which 350 were unique and 150 were duplicates. The duplicate ratio was 30%, indicating that a significant portion of the processed data could be represented through references rather than additional physical storage.

These results show that the proposed framework can simultaneously address redundant data storage and uneven distribution of stored chunks. The use of SHA-256 enables duplicate identification, while AES provides encryption for stored data. Distributed storage improves the utilization of multiple servers, and blockchain-based metadata management provides transparency and traceability for storage operations.

Although the current experimental results demonstrate storage balancing and deduplication effectiveness, further evaluation using additional performance metrics such as upload latency, retrieval latency, throughput, encryption overhead, computational overhead, and blockchain transaction overhead would provide a more comprehensive assessment of the proposed framework.

CONCLUSION

Cloud storage has become an essential platform for storing and managing large volumes of data. However, the increasing amount of duplicate files and data blocks leads to inefficient storage utilization and higher storage costs. Traditional deduplication techniques improve storage efficiency but face challenges such as security risks, centralized metadata management, limited fault tolerance, and unbalanced storage distribution. This project proposes a Blockchain-Based Decentralized Cloud Storage with Reliable Deduplication and Storage Balancing framework to address these limitations. The proposed system integrates Convergent Encryption, File-Level and Block-Level Deduplication, Ramp Secret Sharing, Blockchain Technology, and Heuristic-Based Storage Balancing to provide secure and efficient cloud storage. Convergent Encryption enables secure duplicate detection while maintaining data confidentiality. File-level and block-level deduplication eliminate redundant files and duplicate data blocks, significantly reducing storage consumption. Ramp Secret Sharing enhances data security and fault tolerance by distributing encrypted data into multiple shares, while heuristic-based storage balancing ensures even distribution of data across storage servers. Blockchain technology is employed to maintain immutable metadata, including file hashes, chunk references, timestamps, and ownership information, ensuring transparency, integrity, and secure verification without storing the original file. Experimental results demonstrate that the proposed system effectively reduces duplicate storage, improves storage utilization,

enhances data security, and provides reliable data recovery. Overall, the proposed framework offers a secure, scalable, and reliable cloud storage solution suitable for modern distributed cloud environments.

Future Scope

In the future, the suggested system could be improved in a number of ways. Chunk distribution can be dynamically optimized and storage trends can be predicted by integrating machine learning methods. More complex access control and auditing techniques can be provided by using advanced blockchain systems like Hyperledger Fabric or Ethereum smart contracts. Additionally, the system can be expanded to accommodate multi-cloud setups, allowing data to be stored across several cloud service providers for increased availability and dependability. Future studies might concentrate on supporting large-scale distributed storage infrastructures and applying dynamic deduplication techniques for real-time data streams. To further improve security and privacy, more robust encryption methods and zero-knowledge proof techniques can be used. The system would become more intelligent, scalable, and appropriate for next-generation cloud storage applications with these enhancements.

REFERENCES

1. J. Li, Y. Li, J. Wu, Y. Jin, and Z. Zhang, "Blockchain-Based Decentralized Cloud Storage with Reliable Deduplication and Storage Balancing," IEEE Transactions on Network Science and Engineering, vol. 11, no. 4, pp. 3289–3304, Jul./Aug. 2024.
2. L. Liu, Y. Zhang, and X. Li, "KeyD: Secure Key-Deduplication with Identity-Based Broadcast Encryption," IEEE Transactions on Cloud Computing, vol. 9, no. 2, pp. 670–681, 2021.
3. Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A Blockchain-Enabled Deduplicatable Data Auditing Mechanism for Network Storage Services," IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 3, pp. 1421–1432, 2021, doi: 10.1109/TETC.2020.3005610.
4. Z. Hua, Y. Yao, M. Song, Y. Zheng, Y. Zhang, and C. Wang, "Blockchain-Assisted Secure Deduplication for Large-Scale Cloud Storage Service," IEEE Transactions on Services Computing, vol. 17, no. 3, pp. 821–835, May/Jun. 2024, doi: 10.1109/TSC.2024.3350086.
5. R. Vestergaard, E. Pagnin, R. Kundu, and D. E. Lucani, "Secure Cloud Storage with Joint Deduplication and Erasure Protection," 2022 IEEE 15th International Conference on Cloud Computing (CLOUD), pp. 554–563, 2022, doi: 10.1109/CLOUD55607.2022.00078.
6. C. Pan, L. Zhou, A. Fu, Z. Chen, H. Wang, Y. Zheng, and Y. Gao, "Blockchain-Enabled Efficient Deduplication and Mixed Auditing for Dynamic Cloud Data," IEEE Transactions on Dependable and Secure Computing, vol. 23, no. 2, pp. 3554–3570, 2026, doi: 10.1109/TDSC.2025.3638157.
7. G. Tian, Y. Hu, J. Wei, Z. Liu, X. Huang, and X. Chen, "Blockchain-Based Secure Deduplication and Shared Auditing in Decentralized Storage," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 6, pp. 3941–3954, 2022, doi: 10.1109/TDSC.2021.3114160.
8. Z. Wu, H. Huang, Y. Zhou, and C. Wu, "A Secure and Efficient Data Deduplication Framework for the Internet of Things via Edge Computing and Blockchain," Connection Science, vol. 34, no. 1, pp. 1999–2025, 2022, doi: 10.1080/09540091.2022.2090514.
9. H. Huang, Y. Li, X. Wang, and Y. Zhang, "Blockchain-Based Deduplication with Arbitration and Incentives," IET Information Security, vol. 16, no. 3, pp. 271–282, 2022, doi: 10.1049/ise2.12066.
10. X. Ma, W. Yang, Y. Zhu, and Z. Bai, "A Secure and Efficient Data Deduplication Scheme with Dynamic Ownership Management in Cloud Computing," 2022. (Volume,
11. M. R. Dorsala, V. N. Sastry, and S. Chapram, "Blockchain-Based Cloud Data Deduplication Scheme with Fair Incentives," 2023
12. Z. Ke, H. Gong, and D. H. C. Du, "PM-Dedup: Secure Deduplication with Partial Migration from Cloud to Edge Servers," 2025.