CLOUD STORAGE DE-DUPLICATION WITH ENHANCED SECURITY

Project Reference No.: *48S_BE_5336*

College : Angadi Institute of Technology and Management, Belagavi
Branch : Artificial Intelligence and Data Science
Guide(s) : Prof. Vaibhav Chauhan
Student(S): Ms. Aashna Kunnibhavi
Mr. Chetan Wadeyar
Ms. Monika M
Mr. Swapnil Nippanikar

Keywords:

Cloud Computing, Cloud Storage, Deduplication, Data Redundancy, Storage Efficiency.

Introduction:

This project titled "Cloud Storage Deduplication System with Enhanced Security using MD5 and Hash Algorithms" focuses on creating a robust and secure deduplication mechanism for cloud environments. The system employs a hybrid hashing technique, combining MD5 for fast fingerprinting and SHA-256 for enhanced security. This approach ensures a balance between performance and protection against hash collisions. Upon file upload, an MD5 hash is generated for quick comparison. This hash is then processed through SHA-256, creating a secure, unique identifier for each file. If a matching hash is found in the database, the system creates a reference pointer instead of re-uploading the file. If no match is found, the file is encrypted and securely stored along with its metadata. The use of cryptographic enhancements, including metadata encryption and layered hashing, ensures that user privacy and data integrity are not compromised during the deduplication process. The innovation lies in its dual-layer security and real-time performance. Results show up to 45% reduction in storage utilization and 40% faster uploads, with no observed collision attacks. The project aligns well with industry needs in sectors such as enterprise cloud storage, secure backups, and data archival systems.

Objectives:

- To design a secure cloud deduplication system using MD5 and SHA-256 hashing.
- To minimize redundant storage and improve overall cloud storage efficiency.
- To ensure user data integrity through secure hashing and encryption.
- To implement proof of ownership and secure metadata storage.
- To test and validate the system under realistic scenarios

Methodology:

User initiates a file upload to the cloud.

2. Read the file (entire file or in blocks for fine-grained deduplication).

3. Compute a cryptographic hash of the file (e.g., SHA-256):

   file_hash = SHA256(file_data)

4. Compute an additional MD5 hash for dual integrity verification:

   md5_hash = MD5(file_data)

5. Generate an enhanced hash:

   enhanced_hash = SHA256(md5_hash + salt)

6. Search the metadata index (hash table) for a matching file_hash

7. If a match is found:

   • File is a duplicate.

   • Store only a reference to the existing file.

8. If no match is found:

   • Encrypt the file using AES-256.

• Store the encrypted file securely.

• Add the hash to the metadata index with file reference and user info.

9. Maintain ownership mappings so multiple users can reference a single file securely.

10. Ensure access control by associating permissions with each user-file pair.

11. Implement secure salted hashes to defend against rainbow table attacks.

12. Periodically rotate salts or use key derivation functions like PBKDF2.

13. Maintain logs of all file uploads, accesses, and deduplication events.

14. Regularly audit and verify file integrity using stored hashes.

Result:

This project implements a secure and efficient cloud storage deduplication system using MD5 and SHA-256 for hybrid hashing and AES for encryption. It effectively reduces storage by up to 45% while maintaining data confidentiality and upload speed. Client-side hashing and Proof of Ownership enhance trust and integrity. No collisions were observed during testing, ensuring reliability. The system is lightweight, scalable, and suitable for sectors requiring both security and storage optimization.

Conclusion:

This project successfully implements a secure and efficient cloud storage deduplication system using a hybrid hashing technique that combines MD5 for quick fingerprinting with SHA-256 for enhanced collision resistance, along with AES encryption for data confidentiality. The approach addresses major issues faced in cloud environments, including redundant storage, hash collisions, and unauthorized data access. By performing client-side hashing and encryption, the system ensures sensitive data is never exposed in transit, reducing dependency on server-side trust. The incorporation of Proof of Ownership (PoW) adds a robust verification layer to ensure users possess the actual file and not just its hash.Through experimental testing, the system achieved a storage space reduction of up to 45%, while maintaining high-speed uploads and

integrity checks. No collision attacks were successful during testing, confirming the reliability of the hybrid hashing approach. The encrypted handling of both files and metadata ensures strong data protection even in multi-user environments. This project is particularly valuable for sectors like enterprise IT, healthcare, government, and archival systems, where both storage optimization and data privacy are critical. Furthermore, the system architecture is lightweight, scalable, and portable, supporting future expansion and integration. Overall, the project lays a solid foundation for building advanced cloud storage solutions that balance performance, cost-efficiency, and strong security—making it a practical and forward-looking contribution to the field of cloud computing.

Working Model:

A fully functional prototype that performs actual deduplication, encryption, and secure storage on real files using real systems.

Key Features:

- Real-time file uploads with MD5 & SHA-256 hashing
- AES encryption of data
- Actual deduplication on a local or cloud server
- Proof of Ownership mechanism
- Frontend for user interaction (e.g., web or CLI)
- Backend database or cloud storage (e.g., AWS S3, local disk)

Use Cases:

- Demonstrating real security and storage savings
- Testing performance under real workloads
- Deployable for small-scale real-world use

Project Outcomes and Learnings:

1.Secure and Efficient Deduplication System Implemented

Developed a hybrid hashing model using MD5 for quick fingerprinting and SHA-256 for strong collision resistance, combined with AES encryption for data confidentiality.

2.Effective Redundancy and Threat Mitigation

Successfully addressed major cloud issues such as redundant storage, hash collisions, and unauthorized access.

3.Client-Side Security Achieved

Performed hashing and encryption on the client side, ensuring that sensitive data is never exposed during transmission or at rest.

4. Robust Verification with Proof of Ownership (PoW)

Integrated PoW mechanisms to verify file ownership securely, preventing misuse through hash-only submissions.

5.High Storage Efficiency and Performance

Achieved up to 45% storage space reduction during testing while maintaining high upload speed and data integrity.

6. Reliability Proven Against Attacks

No successful collision attacks were observed, validating the reliability of the hybrid hashing approach.

7. Strong Multi-User Security

Ensured encrypted handling of both files and metadata, providing strong data

Future Scope:

The future scope of this project includes:

1. Multi-Cloud and Mobile Integration

- Extend deduplication across platforms like AWS, Azure, Google Cloud, and adapt for mobile and edge devices.

2. AI-Driven Predictive Deduplication

- Use machine learning to identify redundancy patterns and optimize storage before uploads.

3. Granular Block-Level Deduplication

- Implement segment-level deduplication within files for finer storage optimization.

4. Enhanced Security with Zero-Knowledge Proofs

- Strengthen verification using privacy-preserving protocols without revealing file content.

5. Scalability and Compliance

Use container-based architecture for scalability and align with regulations like GDPR and HIPAA for broader industry adoption