Journal of Science Engineering Technology and Management Science
Volume 02, Issue 08, August 2025 *ISSN: 3049-0952*
www.jsetms.com *DOI:10.63590/jsetms.2025.v02.i08.452-457*

IMPROVED CLOUD STORAGE AUDITING SCHEME WITH DEDUPLICATION

[1]CH. Deepak Sumanth, [2]G. Manideep, [3]U. Abhinav, [4]B. Vinay Reddy, [5]Mr. Kundan. B
[1234] Students, [5] Assistant Professor
Department Of Computer Science and Design
Teegala Krishna Reddy Engineering College, Meerpet, Balapur, Hyderabad-500097

To Cite this Article
CH. Deepak Sumanth, G. Manideep, U. Abhinav, B. Vinay Reddy, Mr. Kundan. B, "Improved Cloud Storage Auditing Scheme With Deduplication", Journal of Science Engineering Technology and Management Science, Vol. 02, Issue 08, August 2025,pp: 452-457, DOI: http://doi.org/10.63590/jsetms.2025.v02.i08.pp452-457
Submitted: 12-07-2025 *Accepted: 18-08-2025* *Published: 25-08-2025*

ABSTRACT
Data grows at the impressive rate of 50% per year, and 75% of the digital world is a copy Although keeping multiple copies of data is necessary to guarantee their availability and long-term durability, in many situations the amount of data redundancy is immoderate. By keeping a single copy of repeated data, data deduplication is considered as one of the most promising solutions to reduce the storage costs, and improve users experience by saving network bandwidth and reducing backup time. However, this solution must now solve many security issues to be completely satisfying. In this paper we target the attacks from malicious clients that are based on the manipulation of data identifiers and those based on backup time and network traffic observation. We present a deduplication scheme mixing an intra and an inter-user deduplication to build a storage system that is secure against the type of attacks by controlling the correspondence between files and their identifiers and making the inter- user deduplication unnoticeable to clients using deduplication proxies. Our method provides global storage space savings, per-client bandwidth network savings between clients and deduplication proxies, and global network bandwidth savings between deduplication proxies and the storage server. The evaluation of our solution compared to a classic system shows that the overhead introduced by our scheme is mostly due to data encryption which is necessary to ensure confidentiality.

1. INTRODUCTION

The proliferation of digital information has placed tremendous pressure on cloud storage providers (CSPs) to handle large-scale data securely and efficiently. Reports indicate that data is growing at an annual rate of 50%, with approximately 75% of stored digital content being redundant copies. This redundancy results in unnecessary storage costs, increased backup times, and inefficient bandwidth utilization.

Data deduplication has been recognized as an effective solution for reducing storage redundancy. It operates by maintaining a single copy of identical data and replacing duplicates with reference pointers. Deduplication can occur at two levels: intra-user, which eliminates redundancy within a single user's

data, and inter-user, which removes duplication across multiple users. Furthermore, deduplication can be client-side, where data is filtered before upload, or server-side, where redundancy is identified post-upload.

Despite its benefits, deduplication introduces security challenges. Malicious users may exploit client-side deduplication to gain unauthorized access by manipulating identifiers or monitoring network traffic during deduplication operations. This vulnerability necessitates the development of secure deduplication frameworks with integrated auditing to ensure data integrity, ownership verification, and confidentiality.

This paper proposes a two-phase cloud storage auditing scheme with deduplication that addresses the shortcomings of existing solutions. The approach integrates a deduplication proxy (DP) between clients and storage servers, ensuring that deduplication is secure, efficient, and unnoticeable to clients, while also enabling integrity auditing.

2. Literature Review

Several studies have examined deduplication in the context of storage optimization and security:

- **Dutch (2008)** analyzed deduplication ratios and their role in optimizing storage within information lifecycle management systems.
- **Meyer and Bolosky (2011)** demonstrated that whole-file deduplication achieves up to 87% of space savings compared to block-level techniques for backup images.
- **Halevi et al. (2011)** identified security risks in client-side deduplication, where attackers could exploit file hashes to illegally access full files.
- **Harnik, Pinkas, and Shulman-Peleg (2010)** highlighted side-channel vulnerabilities in cloud services using deduplication.
- **Mulazzani et al. (2011)** revealed how services such as Dropbox could be exploited as attack vectors, underscoring the security concerns in cloud storage environments.

These studies confirm that while deduplication is highly effective for reducing storage and bandwidth costs, it requires additional safeguards such as secure auditing mechanisms to prevent malicious exploitation.

3. Problem Statement

Existing cloud storage auditing frameworks incur high costs due to duplicated data and inefficient verification mechanisms. Additionally, inter-user and client-side deduplication schemes are highly vulnerable to malicious exploits, leading to unauthorized data access and information leakage.

The research problem addressed in this paper is the design of a secure, efficient, and auditable deduplication scheme that reduces redundancy, ensures integrity, and mitigates potential security risks.

4. Proposed Methodology

The proposed system introduces a two-phase deduplication scheme with integrated auditing:

1. **Intra-user Deduplication**: Performed at the client side, ensuring that each user eliminates redundancy within their own data.
2. **Deduplication Proxy (DP)**: Positioned between the client and the storage server, the DP validates identifiers, ensures ownership, and performs inter-user deduplication securely.
3. **Inter-user Deduplication**: Conducted at the proxy, enabling bandwidth and storage optimization while concealing deduplication operations from clients.
4. **Auditing Mechanism**: Employs cryptographic techniques such as homomorphic authenticators and multi-functional data tags to verify data integrity without decryption.

This methodology ensures that deduplication is both secure and transparent, mitigating risks such as identifier manipulation and side-channel leakage.

5. Implementation

The prototype was implemented using:

- **Programming & Platform**: Java (Servlets, JSP, JDBC) with Apache Tomcat
- **Cryptography**: SHA-256 for hashing, AES-256 for encryption, RSA-1024 for key management
- **Database**: MySQL / MongoDB for metadata and ownership records
- **Environment**: Virtual machines running Ubuntu with deduplication proxies acting as intermediaries between clients and storage servers

Key features include:

- Secure file upload with convergent encryption
- Proxy-based deduplication and auditing
- User authentication and verification of file ownership
- Integration of auditing protocols for continuous integrity verification

6. Results and Discussion

The evaluation compared the proposed scheme with a conventional deduplication system without encryption. Key observations include:

- **Overhead**: Average delay of ~18ms for a 64MB file due to encryption operations.
- **Bandwidth Savings**: Significant reduction in client-server communication through intra-user and inter-user deduplication.
- **Storage Efficiency**: Elimination of redundant data at both client and proxy levels led to substantial space savings.
- **Security**: Resistance against identifier manipulation and network traffic observation attacks was achieved.

While encryption introduced minor computational overhead, the trade-off is acceptable considering the enhanced security and auditing capabilities.

7. OUTPUT SCREENS

Figure : SecDedup Website Homepage

Figure : User Registration

Figure : User Login

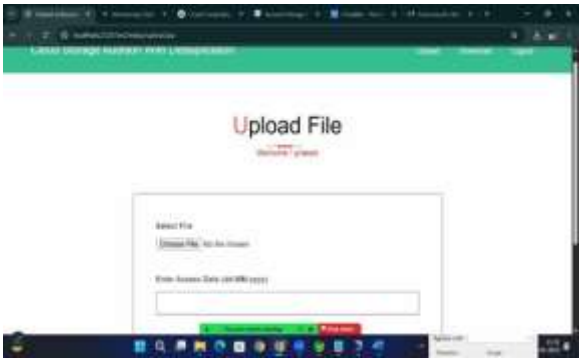

Figure : User Upload File

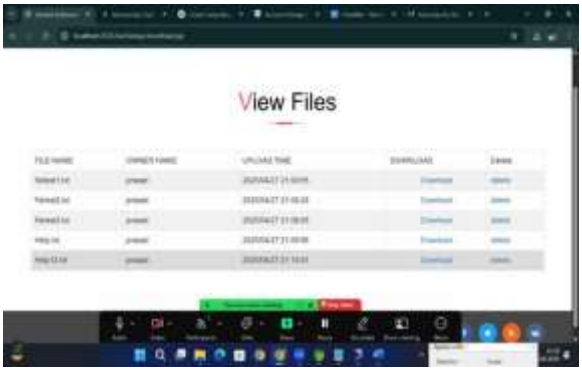

Figure : User View Files (owner)

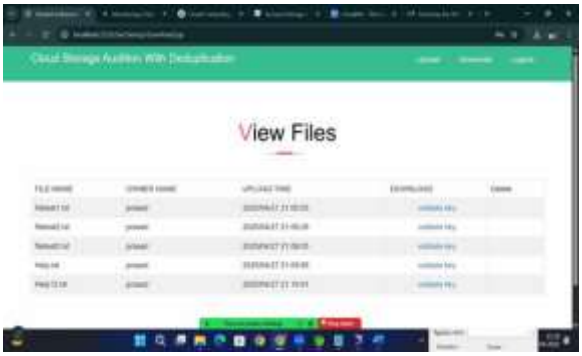

Figure : View Files (user)

Figure : Valid File Key

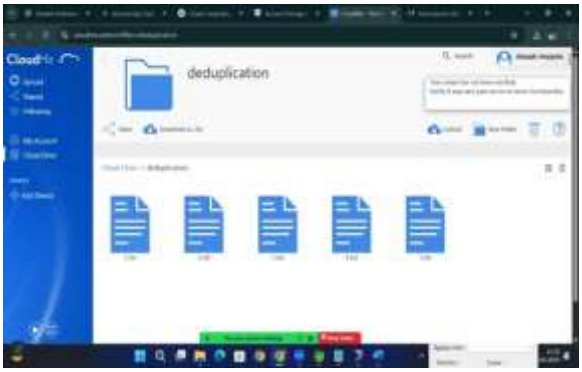

Figure : Connecting to Cloud for Deduplication

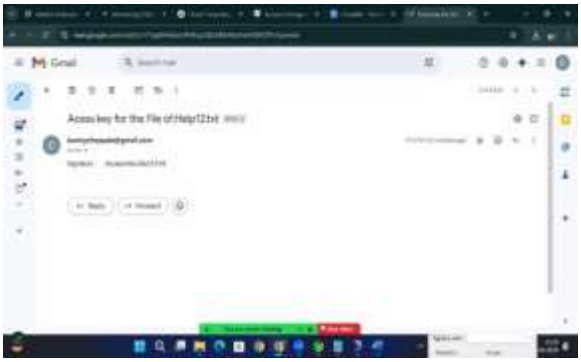

Figure : Login to User Mail

8. CONCLUSION

In this paper, we have presented a two-phase deduplication scheme that (i) ensures that a client actually owns the file he/she wants to store by applying an intra- user deduplication on the client side (ii) ensures that a file corresponds to its claimed identifier through a control by a deduplication proxy located between clients and the storage server and (iii) applies an inter-user deduplication on the deduplication proxy side that makes this inter-user deduplication unnoticeable to clients by adding some delay to put operations so that the length of a file upload is indistinguishable from an upload of its reference. Our method provides protection against attacks from malicious clients, global storage space savings to the CSPs thanks to the inter-user deduplication, per-client bandwidth network savings between clients and the deduplication proxies, and global network bandwidth savings between the deduplication proxies and the storage server. For, future works, we plan to address the confidentiality issues against attacks that can be performed by the CSP. We also plan to extend our solution so that encrypted decryption keys can also be deduplicated

without jeopardizing security properties. We will also consider how to extend the deduplication in our scheme to a block level granularity

References

1. M. Dutch, "Understanding data deduplication ratios," SNIA Data Management Forum, 2008.
2. D. Russel, "Data deduplication will be even bigger in 2010," Gartner, February 2010.
3. D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," in Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST), 2011.
4. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proceedings of the 18th ACM Conference on Computer and Communications security (CCS), 2011.
5. D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage," IEEE Security Privacy, vol. 8, 2010.
6. M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, "Dark clouds on the horizon: using cloud storage as attack vector and online slack space," in Proceedings of the 20th USENIX Conference on Security (SEC), 2011.
7. M. W. Storer, K. Greenan, D. D. Long, and E. L. Miller, "Secure data deduplication," in Proceedings of the 4th ACM International Workshop on Storage Security and Survivability (StorageSS), 2008.
8. M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Server-aided encryption for deduplicated storage," in Proceedings of the 22nd USENIX Conference on Security (SEC), 2013.
9. K. Suzaki, K. Iijima, T. Yagi, and C. Artho, "Memory deduplication as a threat to the guest os," in Proceedings of the 4th ACM European Workshop on System Security (EUROSEC), 2011.
10. M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2013.
11. R. Di Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2012.
12. O. Heen, C. Neumann, L. Montalvo, and S. Defrance, "Improving the resistance to side-channel attacks on cloud storage services," in Proceedings of the 5th International Conference on New Technologies, Mobility and Security (NTMS), 2012.
13. C. Liu, X. Liu, and L. Wan, "Policy-based de-duplication in secure cloud storage," in Trustworthy Computing and Services, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2013, vol. 320, pp. 250–262.
14. W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," in Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC),2012.