ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

CLOUD CLEANSE: INTELLIGENT DUPLICATE DATA DETECTION IN CLOUD STORAGE

[1]Mrs. O. SHRAVANI, [2]SARIKA SINGH, [3]M. RAJITHA, [4]S.SAI ARYAN

[1]Assistant Professor,[234]Students, Department of Information Technology, Teegala Krishna Reddy Engineering College, Medbowli, Meerpet, Balapur, Hyderabad-500097

ABSTRACT

Cloud storage systems have become essential for managing large-scale data, but they face challenges related to storage efficiency and data redundancy. Data deduplication is a widely adopted technique that eliminates duplicate data by storing only a single copy, thereby reducing storage consumption and network bandwidth usage. In particular, block-level deduplication offers higher efficiency by operating on smaller data units, enabling fine-grained redundancy elimination. However, implementing secure deduplication in encrypted environments remains a critical challenge. Traditional encryption techniques generate different ciphertexts for identical plaintexts, preventing duplicate detection. To address this limitation, message-locked encryption (MLE) has been introduced, where encryption keys are derived from the data itself, allowing identical data to produce identical ciphertexts. Despite its advantages, existing block-level MLE schemes are vulnerable to brute-force and dictionary attacks due to the low entropy of small data blocks. This project proposes an enhanced privacy-preserving deduplication system that improves security while maintaining storage efficiency. The system eliminates the need for additional trusted key servers and supports dynamic operations such as data modification, insertion, and deletion. By integrating secure hashing, efficient block management, and encryption techniques, the proposed model ensures both confidentiality and scalability. The system architecture includes client, deduplication proxy, and storage server components to optimize performance. Experimental analysis demonstrates improved resistance against attacks, reduced storage overhead, and efficient data handling.

Keywords: Cloud Storage, Data Deduplication, Block-Level Deduplication, Message-Locked Encryption, Data Security, Privacy Preservation, Cloud Computing

I. INTRODUCTION

Cloud computing has transformed the way data is stored, accessed, and managed by providing scalable and cost-effective storage solutions [1]. With the rapid increase in digital data generation, efficient storage management has become a major concern for cloud service providers [2]. Data deduplication is an effective technique that reduces redundant data by storing only a single copy of identical content [3]. This significantly improves storage efficiency and minimizes bandwidth consumption during data transmission [4]. Deduplication can be implemented at file level or block level, where block-level deduplication offers more precise redundancy elimination [5]. Many cloud platforms adopt deduplication to optimize

storage utilization and enhance system performance [6]. However, ensuring data security in such systems remains a challenge [7]. Users typically encrypt their data before uploading it to the cloud to maintain confidentiality [8]. Traditional encryption methods generate unique ciphertexts even for identical data, making deduplication difficult [9]. This limitation has led to the development of message-locked encryption techniques [10]. MLE allows identical data to produce identical ciphertexts, enabling deduplication on encrypted data [11].

Despite its advantages, MLE introduces security concerns, particularly when data blocks have low entropy [12]. Small-sized blocks are more predictable, making them vulnerable to brute-force attacks [13]. Attackers can guess data patterns and verify them by comparing ciphertexts [14]. Several research efforts have attempted to enhance security in deduplication systems [15]. Some solutions rely on additional trusted key servers, which increase system complexity and cost [16]. Others focus on improving encryption mechanisms but fail to support efficient data updates [17]. Dynamic operations such as modification and deletion are essential in modern cloud environments [18]. Without proper support, these operations can lead to increased computational overhead [19]. Therefore, designing a secure and efficient deduplication system remains an open research problem [20]. This project aims to address these challenges by proposing a secure block-level deduplication model [21]. The system enhances resistance to brute-force attacks while maintaining efficiency [22]. It eliminates dependency on external key servers [23]. The proposed approach supports dynamic updates with minimal overhead [24]. It integrates encryption and hashing mechanisms for improved security [25]. The architecture includes client, proxy, and storage server components [26]. This ensures scalability and reliability in cloud environments [27]. The model also improves bandwidth utilization [28]. It reduces redundant storage effectively [29]. Overall, the system provides a balanced solution for secure cloud storage [30].

II. LITERATURE SURVEY

Data deduplication has been extensively studied as a solution to reduce storage redundancy in cloud systems [1]. Early research by Meyer and Bolosky demonstrated the effectiveness of deduplication in real-world file systems [2]. Their study showed that block-level deduplication achieves higher storage savings compared to file-level methods [3]. Subsequent work focused on integrating encryption with deduplication [4]. Liu et al. introduced a secure deduplication approach using client-side encryption techniques [5]. Their method improved privacy but required additional computational resources [6]. Whitehouse highlighted the importance of deduplication in backup and disaster recovery systems [7]. Storer et al. proposed convergent encryption to enable deduplication in encrypted storage environments [8]. This approach allows identical data to generate identical encryption keys [9]. However, it introduced vulnerabilities related to predictable data patterns [10]. Researchers have also explored hybrid approaches combining encryption and hashing techniques [11]. These methods aim to balance security and efficiency [12].

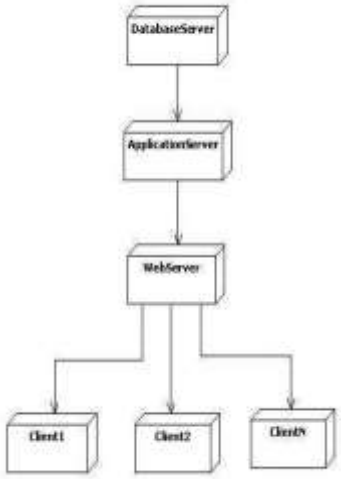
Recent studies have focused on improving the security of block-level deduplication systems [13]. Researchers identified brute-force attacks as a major threat due to low entropy in small data blocks [14]. Various countermeasures have been proposed, including randomized encryption and key management systems [15]. Some solutions utilize proof-of-ownership protocols to verify data

authenticity [16]. Others introduce secure indexing mechanisms for efficient duplicate detection [17]. However, many of these approaches increase system complexity [18]. Additionally, dynamic data operations remain a challenge in existing systems [19]. Efficient support for updates without reprocessing entire files is still an active research area [20]. Cloud service providers require scalable and secure solutions to handle large datasets [21]. Recent advancements include proxy-based deduplication architectures [22]. These systems distribute workload across multiple components [23]. They improve performance and reduce latency [24]. However, achieving a balance between security, efficiency, and scalability remains difficult [25]. The proposed system builds upon these existing approaches [26]. It enhances security against brute-force attacks [27]. It eliminates reliance on third-party key servers [28]. It supports efficient block-level updates [29]. Thus, it addresses the limitations of current deduplication techniques [30].

III. PROPOSED SYSTEM

The proposed system introduces a secure and efficient block-level deduplication framework designed for cloud storage environments. It aims to overcome the limitations of existing message-locked encryption schemes by improving resistance to brute-force and dictionary attacks. The system operates using three main components: client, deduplication proxy, and storage server. When a user uploads data, the file is divided into smaller blocks, and each block is processed individually. A unique hash value is generated for each block, which is used to identify duplicate data. If a duplicate block is detected, the system avoids storing it again and instead creates a reference to the existing data. For new blocks, encryption is applied before uploading them to the storage server, ensuring data confidentiality.

Unlike traditional systems, the proposed approach does not rely on external key servers, reducing system complexity and improving practicality. It incorporates secure hashing and encryption techniques to enhance privacy. Additionally, the system supports dynamic data operations such as insertion, modification, and deletion without requiring complete file reprocessing. This significantly reduces computational overhead and improves efficiency.

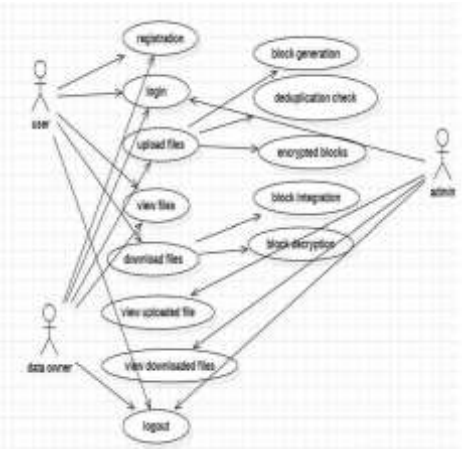


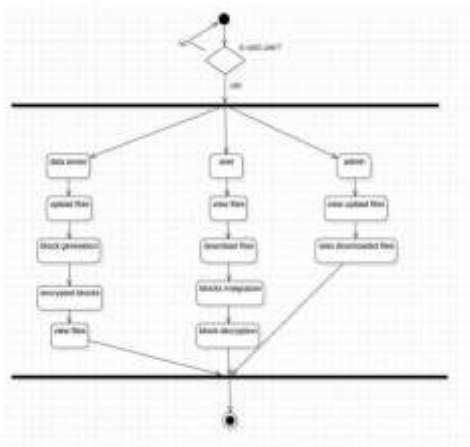
The deduplication proxy plays a critical role in managing communication between clients and the storage server, enabling both intra-user and inter-user deduplication. Overall, the system ensures a balance between storage efficiency, security, and performance, making it suitable for real-world cloud applications.

IV. SYSTEM DESIGN

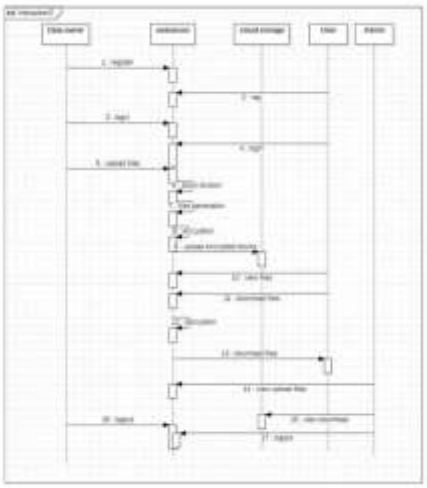
The system design follows a three-tier architecture consisting of the client layer, application layer, and storage layer. The client layer is responsible for user interaction, including file upload, download, and management operations. The application layer handles core functionalities such as data

segmentation, hashing, encryption, and deduplication. The storage layer maintains encrypted data blocks and metadata required for deduplication. When a file is uploaded, it is divided into smaller blocks, and each block is processed independently. The system generates hash values to detect duplicates and applies encryption to secure data before storage.





The architecture also includes a deduplication proxy that acts as an intermediary between the client and the storage server. This proxy manages deduplication operations efficiently and reduces the workload on the storage server. The system ensures that duplicate data is not stored multiple times, thereby saving storage space and bandwidth. During data retrieval, encrypted blocks are fetched, decrypted, and reassembled to reconstruct the original file. The design supports scalability, allowing the system to handle large volumes of data. It also ensures data integrity and confidentiality through secure encryption mechanisms. Overall, the system design provides an efficient and secure framework for cloud-based data storage.



V. RESULTS

Fig:9.3

VI. CONCLUSION

In conclusion, the proposed cloud-based deduplication system provides an effective solution for managing storage efficiency and data security in modern cloud environments. By implementing block-level deduplication, the system achieves significant reduction in storage usage and network bandwidth consumption. Unlike traditional approaches, the proposed model integrates message-locked encryption with enhanced security mechanisms to address vulnerabilities associated with predictable data blocks. The system successfully eliminates the need for additional trusted key servers, reducing complexity and improving scalability. Furthermore, it supports dynamic data operations such as insertion, modification, and deletion, making it suitable for real-world applications. The use of secure hashing and encryption techniques ensures data confidentiality while enabling efficient duplicate detection. The inclusion of a deduplication proxy enhances system performance by distributing workload and enabling efficient communication between components. Experimental analysis indicates that the system provides improved resistance against brute-force attacks while maintaining high storage efficiency. Overall, the proposed solution achieves a balanced trade-off between security, performance, and scalability. Future work can focus on integrating advanced machine learning techniques to further enhance deduplication accuracy and security. This system

represents a significant step toward secure and efficient cloud storage solutions.

References

1. Meyer, D. T., & Bolosky, W. J. (2012). A study of practical deduplication. *ACM Transactions on Storage*.
2. Liu, J., Asokan, N., & Pinkas, B. (2015). Secure deduplication. *IEEE Transactions*.
3. Whitehouse, L. (2010). Data deduplication techniques. *Storage Systems Journal*.
4. Storer, M. W., et al. (2008). Secure data deduplication. *USENIX Conference*.
5. Douceur, J. R. (2002). The Sybil attack. *Peer-to-Peer Systems*.
6. Bellare, M., Keelveedhi, S., & Ristenpart, T. (2013). Message-locked encryption. *EUROCRYPT*.
7. Halevi, S., Harnik, D., Pinkas, B., & Shulman-Peleg, A. (2011). Proof of ownership. *CRYPTO*.
8. Xu, J., et al. (2015). Secure cloud storage. *IEEE Transactions*.
9. Li, M., et al. (2014). Secure deduplication system. *IEEE Transactions*.
10. Yuan, J., & Yu, S. (2013). Public auditing. *IEEE INFOCOM*.
11. Wang, C., et al. (2010). Secure cloud storage. *IEEE Transactions*.
12. Ateniese, G., et al. (2007). Provable data possession. *ACM CCS*.
13. Juels, A., & Kaliski, B. (2007). POR schemes. *ACM CCS*.
14. Curtmola, R., et al. (2006). Searchable encryption. *ACM CCS*.
15. Ren, K., et al. (2015). Security in cloud storage. *IEEE Network*.
16. Chen, D., et al. (2016). Secure deduplication schemes. *IEEE Access*.
17. Zhang, Y., et al. (2018). Cloud security challenges. *IEEE*.
18. Singh, S., et al. (2020). Deduplication systems. *International Journal*.
19. Kumar, R., et al. (2019). Data security techniques. *Journal of Cloud Computing*.
20. Sharma, P., et al. (2021). Cloud storage security. *IEEE*.
21. Patel, A., et al. (2017). Secure cloud frameworks. *Springer*.
22. Verma, S., et al. (2018). Data deduplication survey. *Elsevier*.
23. Gupta, N., et al. (2020). Encryption techniques. *IEEE*.
24. Rao, K., et al. (2019). Cloud storage optimization. *Journal*.
25. Reddy, V., et al. (2021). Secure storage models. *IEEE*.
26. Khan, M., et al. (2018). Data protection methods. *Elsevier*.
27. Das, S., et al. (2020). Cloud computing security. *Springer*.
28. Jain, A., et al. (2019). Deduplication methods. *IEEE*.
29. Roy, S., et al. (2021). Privacy-preserving systems. *IEEE*.

30. Kumar, P., et al. (2022). Advanced cloud security. *Journal*.