Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Secure Healthcare Data Deduplication Scheme With Dependable Multi-Key Management In Cloud Storage

Dr Prabha. B[1], Dr Tiago Zonta[2], Dr Mithileysh Sathiyanarayanan[3]

[1]Assistant Professor, School of Computer Science and Engineering (SCOPE) Vellore Institute of Technology, Chennai, India ORCID: 0000-0002-7059-0435, prabha.b@vit.ac.in
[2]Professor University of West Santa Catarina (UNOESC), Brazil
Email: tiago.zonta@unoesc.edu.br
[3]Scientist MIT Square Services Private Limited, London
Email: mithileysh@mitsquare.com

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

An advancement in computing qualities that are both economical and powerful have been made possible by the Internet's dramatic development in processing and storage methods. A new technology that offers internet application access and data storage, cloud computing is becoming more and more popular. Both opportunities and challenges abound in this system. Two extremely important issues that need to be addressed are data security and the growing amount of similar data duplication in cloud. A cloud computing system, supported via distributed & virtual machine technology, offers the most comprehensive services in the most effective way. When using the cloud technology, customers may access their limitless storage space from any location at any time. Cloud service providers are responsive to customer requests to increase data storage capacity by integrating cloud systems with deduplication techniques, while deduplication techniques erase redundant data that is available in cloud environments, according to the company. Preserving data privacy is another significant issue that needs to be acknowledged for its significance. For reducing duplicate data quantity in storage system, the deduplication technique was developed. This study offers a novel method for eliminating duplicate or repetitive information from cloud servers, that might aid in lowering storage space amount and needed bandwidth. It was illustrated that by using testing outcomes, proposed model eliminates the main drawbacks of the existing methods while also offering increased security for data kept in cloud storage.

Keywords: Cloud computing, data storage in cloud, key management, data deduplication, security.

1. INTRODUCTION

A cloud computing system, supported via distributed & virtual machine technology, offers the most comprehensive services in the most effective way. When using the cloud technology, customers may access their limitless storage space from any location at any time. Cloud service providers are responsive to customer requests to increase data storage capacity by integrating cloud systems with deduplication techniques, while deduplication techniques erase redundant data that is available in cloud environments, according to the company. Preserving data privacy is another significant issue that needs to be acknowledged for its significance [1-3]. A new type of deduplication technology is being utilized to encrypt data before it is grown out in a cloud server environment in order to aid in data deduplication. The great majority of data is currently stored and accessed on the cloud due to the networking and storage environment. Since it is not a data disc, the data disc is unable to detect duplicate data on the disc. A substantial percentage of disc storage space may be used by duplicate data [4,5]. Duplicate data has a negative influence on disc performance, space, speed, and other performance metrics as the amount of data increases [6]. Cloud computing is a system that provides computer services via the internet, allowing users to access a variety of services on demand. Software as a service (SaaS), platform as a service (PAAS), infrastructure as a service (IAAS), and other services are examples of cloud computing services that are geared at users. These days, cloud service companies offer a wide range of services to their customers at extremely low prices, such as massive storage capacity and resource parallel processing [7,8]. These techniques are used to identify redundant data in the cloud data storage, and a groundbreaking approach to data deduplication operates on the incessantly growing volume of digital data stored in data cloud storage. A single, distinct copy is maintained at the end and can be sent to any or all of the authorized clients who have asked for it. To do duplicate checking, the user needs to ask the administrator to examine the selected file for duplicates. Data de-duplication using a secured cloud storage system is created in this research effort, and this is one of the contributions of the research work. Data deduplication and a cryptography-based security strategy are two of the most significant dangers that have been identified in this investigation [9-11]. An innovative technique for key generation is created in this suggested work, which allows for decryption and access to the repeated data chunks. Many secured data deduplication models for the process of encryption were developed recently in safeguarding the cloud users' data privacy. Despite this, duplicate files that aren't necessarily the best choice for the circumstance are usually not handled by traditional deduplication techniques. The experiment shows that the proposed deduplication method can effectively manage the storage burden of more assembled files while also reducing reaction time in a creative way. Our aim is to develop a secure and efficient data deduplication model for cloud storage that ensures both storage optimization and data confidentiality.

Objectives

- To analyze the limitations of existing deduplication techniques in handling encrypted cloud data.
- To design a cryptographic key generation mechanism that supports secure and efficient data deduplication.
- To implement a cloud-based deduplication system capable of identifying and eliminating redundant data while preserving user privacy.
- To evaluate the performance of the proposed system in terms of storage utilization and response time.

Contributions of the Paper

- Proposes a novel deduplication framework that combines encryption and data optimization strategies in cloud environments.
- Introduces an innovative key generation scheme to facilitate secure access to duplicate data chunks.
- Demonstrates the capability of the proposed system to reduce cloud storage burden and enhance response time.
- Provides a comparative analysis of the proposed system against traditional deduplication techniques in terms of efficiency and security.

The following is the structure of the article's remaining section: A summary of earlier pertinent work done for data deduplication objectives is given in Section 2. We covered the Suggested Deduplication System Scheme in Section 3, Section 4, delivers the Results and Discussion and Section 5 offers conclusion of proposed model.

2. RELATED WORKS

There have been various academics that have endorsed the phrase deduplication throughout the last couple of decades, and the period might relatively be new survey area to explore. In order to aid and broaden the data deduplication system, several researchers have proposed different approaches and technologies. It is anticipated that the research flow and approaches used by other researchers at present system would be elaborated in the literature survey section.

In order to generate a hash value for the establishment of a Merkle tree meant for ownership proof, Shai Halevi and colleagues [12] created a notion that is implemented as a hash function SHA256. According to the report, a proof of ownership was proposed, which enables clients to successfully influence servers that use licensed ones. To guarantee data acceptance, Huiying Hou and associates [13] propose a technique for identifying cloud data storage checking in combination with data deduplication management for several data acceptance phases. In order to achieve server-based MLE at the segment level for scaling back compute intake, Yukun Zhou et al. [14] developed the EDedup system, which uses a similarity aware encrypted deduplication methodology. A 'key sharing' strategy is proposed by Liang Wang and colleagues [15] that can be used to prove ownership for safe deduplication, provided that the key's independent facts are supported. A deduplication system for cloud data was created to facilitate certificate re-encryption, including Proof of Ownership Supported Certificate Less Signature (PoW – CLS) and Certificate Less Proxy Re-Encryption (CL-PRE), according to Xiaoyu Zheng et al. [16]. Wen Xia and associates created a parallelized data deduplication method called P-Dedupe [17]. They were able to increase their data deduplication method's speed by about 50% by segmenting the process into four steps. To make the computing block more manageable, it is advised to parallelize the secure hash-based fingerprinting and CDC phases after incorporating chunks and files into these four stages [18]. Haoran Yuan et al. [19] introduced secured and accessible model for deduplication of data termed active client regulation model that modifies clients active cluster at safe side and thus preventing illegal clients from accessing deeper facts that were influenced by effectual clients. Shanshan Li [20] presents a well-structured and safe 'CSED' (Client Side Encrypted Data Deduplication) system which uses PoW. Using the double decryption technique and FHE, they created a hybrid structure scheme for the advanced scheme, which was implemented in the advanced scheme. In the advanced system, data providers are only responsible for encrypting and decrypting algorithms. They demonstrate that the effectiveness of these two multi-key privacy-preserving algorithms [21].

The suggested SDD-RT-BF model is divided into three major phases: the permissible deduction phase, the evidence of ownership phase, and the update of the significant role phase. Initial implementations of the convergent coding approach were employed to prevent data leakage, and the mechanism for resourcefully achieving authorized deduplication was implemented using the re-coding procedure. Specific attention is paid by the management center to the authorized request, which includes the creation of an RT structure for mapping the relationship between roles and keys. Furthermore, BF is used to refresh the data and to verify ownership recovery in a short period of time. The inclusion of RT in conjunction with BF for secure data deduplication demonstrates the importance of this news piece [22].

3. PROPOSED WORK

In this section, proposed deduplication system model employed is described in detail and modules of this proposed work are listed as follows:

3.1 Deduplication and Encryption process

Following are the steps carried in the proposed deduplication system model process.

Deduplication systems have been developed by a large number of researchers that have made significant contributions to the field by employing common encryption methodologies & execution of trusted environments for enabling safer key managements in present model. Depending on previously identified requirements in security positions holdings along with system overhead previously employed, advanced secured deduplication models are investigated, which may incorporate the following features:

a. The input File will be encrypted/encoded with respecting Hashing keys, that is used for the convergent encryption with the use of AES model along with their entity.

b. In addition, the file is separated into four parts, each of which is encrypted using a different hash function key.

c. A variety of hash functions are used to hash the files since a hacker would fail if they were to decipher specified split hashing function thus attempting in accessing remaining splits & joints for attaining information of file.

d. It is not conceivable in our proposed system since the four divisions in the four separate servers use various hash functions & its keys, along with encryption through key likewise differs, thus resulting in enhanced security.

e. Considering the hashing techniques in use, the duplication is analyzed in chunks across all five servers.

Hash key-based Encryption: Throughout the proposed system, AES-based encryption is used. The file is encrypted using the AES algorithm alone, as well as its corresponding hash key, which is used for convergent encryption. The following strategies make up the encryption strategy employed in the suggested system. Known as the hashing method, the hash key is one of the most popular encryption algorithms. A hash function that performs one-way encryption in a single direction is an example. The document's content is used to build a hash key. Every entry has a unique tag and hash key generated for it. A key generation technique may convert copy of information (N) for convergent/united key ®, based on barrier security. A goal of making hash key will be used as distinctive and noticeable hash key to jumble a square of data. One type of encryption that is used in combination with another encryption technique is convergent encryption. In case, two identical records will be mixed with varied keys, then information encoded fails the de-duplication process, creating a variety of encoded information blocks which are not able to be again shared. To get over this restriction, a different strategy known as "Convergent Encryption" (CE) [17] is explained. Official definitions state that a CE strategy typically consists of the following four core tasks:

3.2 Steps of Deduplication system model

3.2.1 User Login

The user login procedure is seen in Fig. 1; initial login communication of user is accomplished through the key-based mechanism usage. This is authentication model which might be employed instead of password in certain situations, such as online banking. Instead of requiring a user's password, an asymmetric cryptography model that employs both public and private keys can be used to confirm the client's uniqueness. After this process is finished successfully, the client will receive a verification code from the code generating database. The client will then use this code to verify that the code they acquired from the code verification database is authentic. This section generates the protected keys needed for the login process using the AES and RSA algorithms. The key that was entered into the database is compared to the created key. Because of industry-standard encryption standards, the original password is completely disguised before it is transferred to the API.

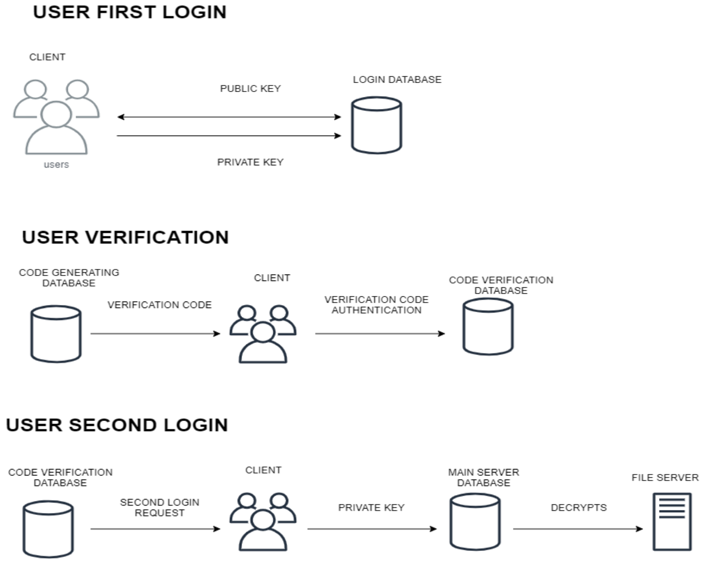

Fig. 1. User login system representation

3.2.2 File Uploading

The process of file uploading is depicted in fig 2. The following are the steps carried in file uploading process.

- After successfully logging in with their login credentials, a registered user is prepared to upload a file to the server. The user will be given access to the file server after finishing the login process (Steps 1 through 9 are detailed in the previous login section). The user will be guided through the file uploading process via the subsequent steps.
- On the basis of the number of tries, step 10 will determine if the login IP and access were successful. Step 11 will determine whether the login IP and access were successful on the basis of the number of attempts.
- Step 12: Checking deduplication will be requested by user.

- Step 13: Server thus responds to the user with the results of the deduplication check.
- Step 14: In case no such duplicate is found after the comparison of server, the file is prepared for upload to the file log server.
- Step 15: After encryption is complete, uploaded file will split as four sections and thus saved.
- Initially, the submitted document is saved in its whole on one server, and a duplicate is split up into four parts and kept on four different servers. Instead of employing the same hash function, the document is now transformed into a hash using four distinct hashing keys for four different servers, and the hash function is used to confirm that the duplicate document exists. They are listed in the following section.
- The encryption methods used by Servers 1 and 3 are MD5, SHA-1, and SHA-256, respectively. Server 4 employs the hashing technique SHA-384.

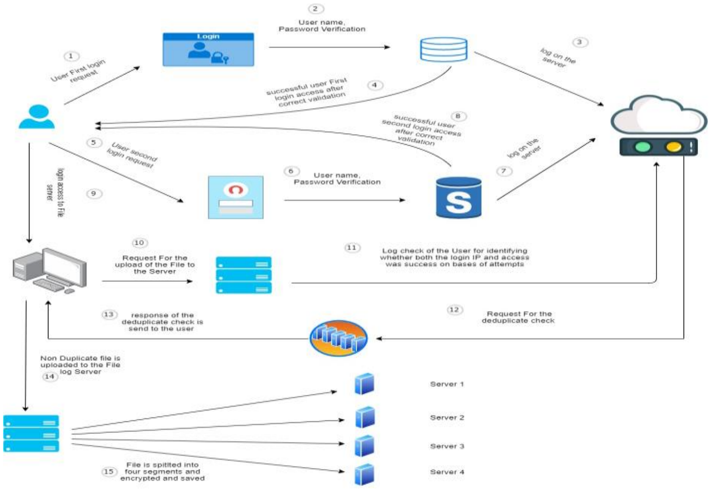
2
User name,
Password Verification
Login
1
User First login request
3
log on the server
4
successful user First login access after correct validation
8
successful user second login access after correct validation
5
User second login request
login access to File server
9
6
User name,
Password Verification
7
log on the server
10
Request For the upload of the File to the Server
11
Log check of the User for identifying whether both the login IP and access was success on bases of attempts
13
response of the deduplicate check is send to the user
12
Request For the deduplicate check
Non Duplicate file is uploaded to the File log Server
14
Server 1
Server 2
Server 3
Server 4
15
File is spitited into four segments and encrypted and saved

Fig. 2. The procedure for uploading files to the cloud storage system

3.2.3 Download Request:

The client downloads the files in this module by following the steps listed below, which are depicted in Fig. 3. After successfully logging in with their login credentials, a registered user is prepared to download the file from the server. The user will be given access to the file server after finishing the login process (Steps 1 through 9 are detailed in the previous login section). The user will be guided through the file download procedure by the subsequent steps.

Step 10: To download the file, the user requests it from the server.

Step 11: The client's identity can be confirmed by the administrator.

Step 12: Based on the number of attempts, the administrator can examine the user log to ascertain in case login IP and access will be successful.

Step 13: Whether the attempt was successful or not is shown by the log server's response.

Step 14: Despite the successful attempt, even if the file has already been downloaded, the client is given access to it with a time constraint.

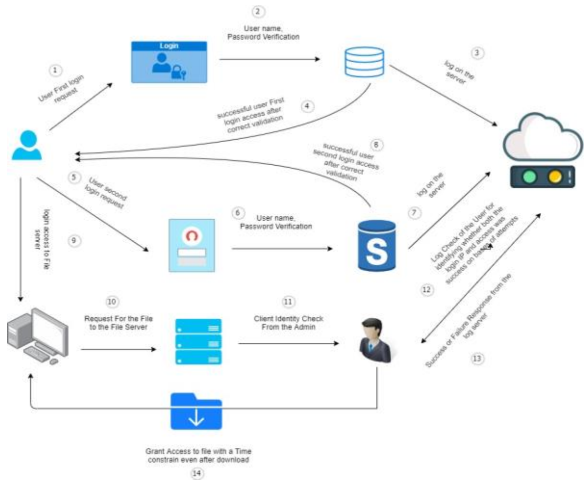
2
User name,
Password Verification
Login
1
User First login request
3
log on the server
successful user First login access after correct validation
4
8
successful user second login access after correct validation
5
User second login request
log on the server
7
login access to File server
9
6
User name,
Password Verification
Log Check of the User for identifying whether both the login IP and access was success on basis of attempts
12
10
Request For the File to the File Server
11
Client Identity Check From the Admin
Success or Failure Response from the log server
13
Grant Access to file with a Time constrain even after download
14

Fig. 3. The procedure for downloading files in a cloud storage system

3.2.4 Request to download own file:

The user must take the actions listed below, which are shown in Fig. 4, for downloading its own file from cloud. Since they must also finish the aforementioned login procedure (Steps 1 through 9).

- Step 10: To download the file, the user requests it from the server.
- Step 11: The client's identity can be confirmed by the administrator.
- Step 12: Based on the number of attempts, the administrator can examine the user log to ascertain in case login IP and access will be successful.
- Step 13: Whether the attempt was successful or not is shown by the log server's response.
- Step 14: Despite the successful attempt, even if the file has already been downloaded, the client is given access to it with a time constraint.
- Step 15: Finds the split files according to the log files
- Step 16: File is merged and access is granted to file with a time constraint even after download
- Step 17: The requested file server is connected to the file path.
- Step 18: At last, the necessary file is downloaded.

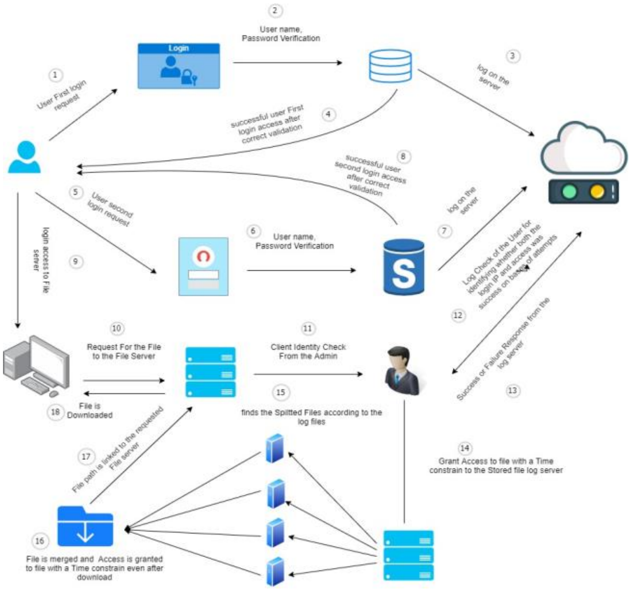
2
User name,
Password Verification
Login
1
User First login request
3
log on the server
successful user First login access after correct validation
4
8
successful user second login access after correct validation
5
User second login request
log on the server
7
login access to File server
9
6
User name,
Password Verification
Log Check of the User for identifying whether both the login IP and access was success on basis of attempts
12
10
Request For the File to the File Server
11
Client Identity Check From the Admin
Success or Failure Response from the log server
13
18
File is Downloaded
15
finds the Splitted Files according to the log files
17
File path is linked to the requested File server
14
Grant Access to file with a Time constrain to the Stored file log server
16
File is merged and Access is granted to file with a Time constrain even after download

Fig. 4. Make a file request to download your own files from the cloud storage system.

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4. EXPERIMENTAL RESULTS

Using a range of key creation techniques, the encoding and decoding procedures are used here for estimating deduplication system proposed. Every step of the investigation was conducted on an Intel Xeon E5530 (2.40 GHz) server that was running Windows 10. As can be seen, information will be taken from sample page of login where several users will be thus requested for entering with their own password. They were employed in the procedure to generate the encrypted key, which was subsequently finished within the allotted time frame.

4.1 User Login

An initial, total login times, phrase in sec are displayed in Table 1.

Table 1 Time spent logging in in sec

S.NO	Ini_Login(SEC)	Phrase _login(sec)	Login Time (sec)
1	1.63	0.00208	1.63208
2	0.182	0.00069	0.18269
3	0.843	0.00096	0.84396
4	0.235	0.0008	0.2358
5	0.173	0.00086	0.17386
6	0.241	0.00066	0.24166
7	0.384	0.00063	0.38463
8	0.25	0.00299	0.25299
9	0.206	0.00123	0.20723
10	0.272	0.00818	0.28018

4.2 Uploading

After file upload it is first saved in its entirety on one server, document copy will be split as four parts and thus kept them on four parts and kept on four different servers throughout the internet. To guard against security threats (MD5, SHA − 1, SHA − 256, and SHA − 384) on each server because of security concerns (Copy 1 to Copy 4), each server uses a different technique to generate the key and encrypt the data. Table 2 is the timing of encryption process, file size, and key generation.

Table 2 Timing of the encryption process, file size, and key generation

Type of file	File's Total size (mbps)	File 1 size (kbs)	Key gen_Time (sec) copy 1	Time of encryption 1 (Nano sec)	File size 2	Gen_time (sec) copy 2	Encryption time 2 (Nano sec)
Image 1	1.04	20,142	0.173	0.247	23,173	0.1639	0.2836
Image 2	61,054	12,333	0.08105	0.1261	14,288	0.106	0.131
Music 1	1.04	20,142	0.173	0.247	23,173	0.163	0.283
Music	1.06	20,142	0.130	0.192	23,173	0.174	0.24465
Video 1	2.45	48,393	0.418	0.474	56,205	0.4006	0.54221

Video 2	2.45	48,393	0.37719	0.597	56,205	0.765	0.6283
Text 1	156	20	0.03063	0.00476	20	0.00025	0.00754
Text 2	156	26	0.00286	0.00171	26	0.0003	0.00199
Size of file 3	**Copy 3 Gen_Time(SEC)**	**Encryption Time3(NANO SEC)**	**Size of file 4**	**Copy 4 Gen_Time(SEC)**	**Encryption Time4 (Nano SEC)**	**Upload time (SEC)**	
61,363 Kbs	0.57458	0.56389	60,826kbs	0.62967	0.55914	1.54989	
61,363 Kbs	0.57875	0.63231	60,826kbs	0.45279	0.61916	1.44375	
15,336 kbs	0.13987	0.13227	15,271 kbs	0.11737	0.15665	0.49533	
15,336 kbs	0.1456	0.16013	15,271 kbs	0.14729	0.1387	0.48412	
26,447kbs	0.24054	0.22853	26,399kbs	0.24387	0.24899	0.82402	
26,347kbs	0.27465	0.30358	26,325kbs	0.36943	0.23765	0.97108	
26kbs	0.00037	0.00163	26kbs	0.00247	0.00183	0.21803	
20kbs	0.00042	0.00334	20kbs	0.01342	0.00382	0.34543	

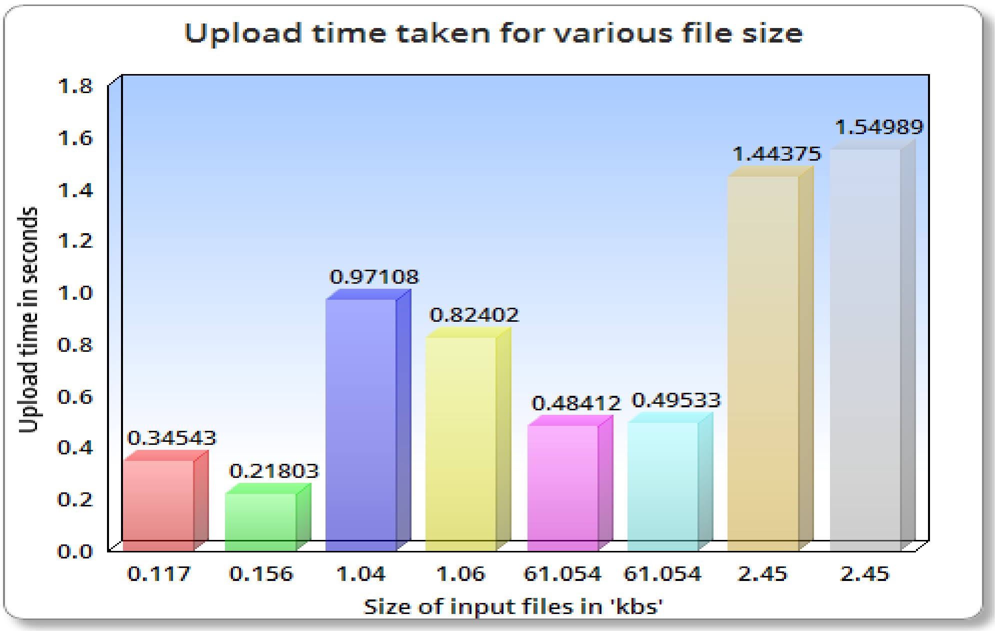

Fig. 5. Time Spent Uploading Different File Sizes

Table 2 shows the size of the files, as well as the times taken by the key creation and encryption processes. Lastly, the amount of time needed for the file upload process is calculated. Fig. 5 displays the upload time for various file sizes.

4.4 Download

To retrieve the files from log server, user will send request to download in cloud service provider, like download request to the cloud service provider, as described in the section above. Using file ID at log server for retrieving every chunk file encrypted from provider of cloud service, then use the appropriate key to decode each encrypted chunk. Lastly, rebuild the original file from the ground up.

Table.3 gives specific details, such as the size of the downloaded file, the total file merge time in seconds, and the decryption time in nanoseconds for each chunk. The time needed to download a file of different sizes is shown in Figure 6.

Table 3 Timing of the decryption procedure

S. NO	**File Type**	**File Size** (kbs)	**Decryption Time 1** (**NANO SEC**	**Decryption Time 2** (**NANO SEC**	**Decryption Time 3** (**NANO SEC**	**Decryption Time 4** (**NANO SEC**	**Total Merg (Seconds)**
1	Image 1	61,054Kbs	0.16417	0.1614	0.16541	0.20014	0.69112
2	Image 2	61,054Kbs	0.14245	0.15678	0.16267	0.17245	0.63435
3	Text file 1	117kbs	0.01013	0.0031	0.00581	0.00347	0.02251
4	Text file 2	156Kbs	0.00078	0.00256	0.00062	0.00045	0.00441
5	Video 1	2.45mbs	0.53731	0.62084	0.79836	0.64468	2.60119
6	Video 2	2.45mbs	0.54866	0.62814	0.76312	0.645	2.58492
7	Music 1	1.04mbs	0.24135	0.24324	0.32559	0.29622	1.1064
8	Music 2	1.06mbs	0.32896	0.25487	0.27488	0.30439	1.1631

It is evident that the encoding and decoding processing times are significantly greater when the test data are displayed in tables and graphs. In fact, the encryption time for a data chunk in the current system result is significantly longer than the encoding time for 117 kbs (0.34543) and 2.45 mbs (1.54989) during file upload. When a file is downloaded, the decoding time is shorter than when a data block is encrypted. The decryption part of the system may require less time to develop if the encoding and decryption modules are pipelined. Testing the proposed system with various file sizes (e.g., 156Kbs, 1.04mbs, 1.06mbs, 61,054Kbs, 2.45mbs, & 1.04mbs) produced a similar set of results. Accompanying the growth in file size has been a large increase in the computing cost of creating keys, executing symmetric encryption, and decrypting.

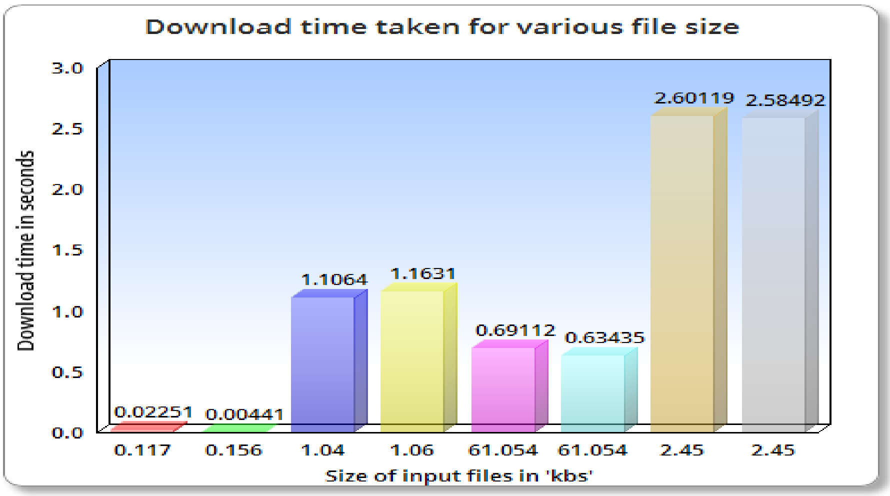

Fig. 6. Time Spent Downloading Different File Sizes

5.CONCLUSIONS AND FUTURE WORKS

Cloud storage systems are increasingly being challenged by the exponential growth of data, necessitating innovative solutions for efficient storage management and data privacy preservation. This study proposed a secure data deduplication framework that integrates cryptographic techniques to ensure both deduplication efficiency and data confidentiality. By introducing a novel key generation mechanism, the proposed system enables secure access to deduplicated data, effectively addressing the shortcomings of conventional methods. Experimental results demonstrate that the system significantly reduces storage consumption and improves response time when managing large volumes of data. Future work will focus on extending this framework to support dynamic data updates and scalability in multi-user environments to further enhance its applicability and robustness in real-world cloud infrastructures..

Acknowledgement: No funding is available for this research work.

Conflicts of Interest: The authors declare that they are having no conflict of interest for reporting regarding present work.

Data Availability Statement: Data which supports this finding of present model are available on request from corresponding author.

REFERENCES

[1] Waghmare, V., & Kapse, S. (2016). Authorized deduplication: an approach for secure cloud environment. Procedia Computer Science, 78, 815-823.

[2] Aldar, B. D., & Devmane, V. (2015). A survey on secure deduplication of data in cloud storage. Int. J. Innov. Eng. Technol, 6.

[3] Vijayakumar, D., Srinivasagan, K. G., & Vivekrabinson, K. (2025). Enhancing cloud storage security through blockchain-enabled data deduplication and auditing with a fair payment. *Peer-to-Peer Networking and Applications*, *18*(3), 147.

[4] Lakade, S. A., Gurav, Y. R., & Lohia, H. Encrypted Cloud Data Deduplication Technique for Secure Data Storage Optimization Over Cloud. *JOURNAL OF TECHNICAL EDUCATION*, 28.

[5] Ruba, S., & Kalpana, A. M. (2025). Advanced chunk-based data deduplication framework for secure data storage in cloud using hybrid heuristic assisted optimal key-based encryption. *Wireless Networks*, 1-23.

[6] Preethi, S. T., Mahalakshmi, P., & Hariharasudhan, S. (2024, December). Enhancing Data Security and Efficiency: A Hybrid Cloud Approach to Secure Authorized Deduplication. In *2024 International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-6). IEEE.

[7] Jain, K., Singh, P., & Li, X. (2024, August). Privacy-Preserving Disease Prediction with Secure Data Deduplication on Untrusted Cloud Servers. In *2024 IEEE 7th International Conference on Multimedia*

Information Processing and Retrieval (MIPR) (pp. 660-666). IEEE.

[8] Goel, A., Prabha, C., Malik, M., & Sharma, P. (2024). Security Concerns and Data Breaches for Data Deduplication Techniques in Cloud Storage: A Brief Meta-Analysis. *International Journal of Safety & Security Engineering*, *14*(2).

[9] Akbar, M., Ahmad, I., Mirza, M., Ali, M., & Barmavatu, P. (2024). Enhanced authentication for de-duplication of big data on cloud storage system using machine learning approach. *Cluster Computing*, *27*(3), 3683-3702.

[10] Altowaijri, S. M. (2024). Deduplication-Aware Healthcare Data Distribution in IoMT. *Mathematics*, *12*(16), 2482.

[11] Sathyabama, P., Shalini, M., Bharathi, R. D., Sathi, G., Thilagam, K., & Maheswari, S. U. (2024, June). Ensuring Confidentiality: A Secure Deduplication Model to Prevent Healthcare Records over Cloud Computing Environment with Advanced Crypto Logics. In *2023 4th International Conference on Intelligent Technologies (CONIT)* (pp. 1-7). IEEE.

[12] Xia, W., Feng, D., Jiang, H., Zhang, Y., Chang, V., & Zou, X. (2019). Accelerating content-defined-chunking based data deduplication by exploiting parallelism. *Future Generation Computer Systems*, *98*, 406-418.

[13] Singh, P., Agarwal, N., & Raman, B. (2018). Secure data deduplication using secret sharing schemes over cloud. *Future Generation Computer Systems*, *88*, 156-167.

[14] Yuan, H., Chen, X., Jiang, T., Zhang, X., Yan, Z., & Xiang, Y. (2018). DedupDUM: Secure and scalable data deduplication with dynamic user management. *Information Sciences*, *456*, 159-173.

[15] Li, S., Xu, C., & Zhang, Y. (2019). CSED: Client-side encrypted deduplication scheme based on proofs of ownership for cloud storage. *Journal of Information Security and Applications*, *46*, 250-258.

[16] Halevi, S., Harnik, D., Pinkas, B., & Shulman-Peleg, A. (2011, October). Proofs of ownership in remote storage systems. In *Proceedings of the 18th ACM conference on Computer and communications security* (pp. 491-500).

[17] Hou, H., Yu, J., & Hao, R. (2019). Cloud storage auditing with deduplication supporting different security levels according to data popularity. *Journal of Network and Computer Applications*, *134*, 26-39.

[18] Zhou, Y., Feng, D., Hua, Y., Xia, W., Fu, M., Huang, F., & Zhang, Y. (2018). A similarity-aware encrypted deduplication scheme with flexible access control in the cloud. *Future Generation Computer Systems*, *84*, 177-189.

[19] Wang, L., Wang, B., Song, W., & Zhang, Z. (2019). A key-sharing based secure deduplication scheme in cloud storage. *Information Sciences*, *504*, 48-60.

[20] Zheng, X., Zhou, Y., Ye, Y., & Li, F. (2020). A cloud data deduplication scheme based on certificateless proxy re-encryption. *Journal of Systems Architecture*, *102*, 101666.

[21] Cui, H., Deng, R. H., Li, Y., & Wu, G. (2017). Attribute-based storage supporting secure deduplication of encrypted data in cloud. *IEEE transactions on big data*, *5*(3), 330-342.

[22] Li, J., Chen, X., Li, M., Li, J., Lee, P. P., & Lou, W. (2013). Secure deduplication with efficient and reliable convergent key management. *IEEE transactions on parallel and distributed systems*, *25*(6), 1615-1625.