Optimization and Secure Data Storage over cloud using Encrypted cloud data Deduplication technique.

Mr. Suhas A. Lakade, Dr. Harsh Lohiya, Mr. Yuvraj R. Gurav,

(Department of Computer Science and Engineering, School Of Engineering, Sri Satya Sai University of Technology and Medical Science,Sehore (MP) India.)

Email- suhaslakade@gmail.com,lohiya27harsh@gmail.com , yuvi1333@gmail.com

Abstract

Because cloud storage systems may offer consumers convenient and affordable network storage, they are becoming more and more popular. But as data grows exponentially, cloud storage systems are under increasing pressure to store more and more data, particularly because a lot of redundant data takes up a lot of storage space. Data holders frequently store their data in encrypted form to protect data security and privacy. The majority of schemes don't allow for data access, which makes it difficult to claim ownership of data kept in the cloud. In this research, we offer a scalable and secure data deduplication technique with dynamic user management that prevents unwanted cloud users from accessing sensitive data held by legitimate users and securely updates users in dynamic groups Data saved in the cloud and other big storage places is now encrypted to maintain security. One issue with this is that deduplication techniques cannot be used to encrypted data. As a result, it seems difficult to execute deduplication securely over encrypted data in the cloud. In this work, several approaches to this problem are examined. The document uses the AES algorithm for encryption and the MD5 technique to find the file's hash.On the other side, data security requires encryption even while deduplication is required for optimal storage. Consequently, encryption and deduplication need to work together to ensure safe and efficient storage. Data deduplication is one way to reduce the amount of storage space needed by an organization to store its data. This work aims to achieve a balance between storage optimization and security through two key objectives. The objective proposes a novel title- and keyword-based deduplication technique for encrypted documents. The model begins with title matching, which contrasts the document titles in order to aid in the early identification of potential duplication.

KEYWORDS: *Cloud Storage, De-Duplication, DeDup, Optimization*

1 Introduction

The process of data deduplication reduces the need for large storage capacity by getting rid of redundant data. When data is being written to the storage system, deduplication can be performed inline. Alternatively, it can be performed in the background to remove duplicates after the data has been stored to disk [1]. The process of removing redundant data from a dataset is known as deduplication. A deduplication evaluation tool finds redundant data and removes it as part of a secure data deduplication procedure, leaving just one instance of the data to be stored.Object storage, file storage, and block storage are the three primary categories of cloud storage. Each has a unique set of benefits and applications. Cloud storage can be separated into four categories Public cloud storage, Private cloud storage, Hybrid cloud storage, Community cloud storage [2][3]. Absence of redundancy: Email backups are a prime illustration of deduplication. Since emails may all use the same footer picture, only one backup copy of this image is required, and each email can point to the backup file[4][5]. Using distinct fields for email, non-email, and calendar invitation data, deduplication compares file content and metadata to find duplicates. This functionality only conceals files according to the selected deduplication view; it doesn't actually destroy any files. There are several ways to resolve or eliminate duplicates from a data set, which is known as deduplication. Key-based

deduplication, for instance, compares and eliminates duplicates using a distinct key or identifier, such as a primary key, surrogate key, hash key, or composite key[6][7]. Regrettably, encryption and deduplication are two incompatible technologies[8]. When two identical data segments are encrypted, they become indistinguishable, While the goal of deduplication is to identify identical data segments and store them just once, the outcome of encryption is to obfuscate the difference between two identical data segments[9][10].This implies that if users encrypt data using a common method, the cloud storage provider is unable to use deduplication because encryption will cause two identical data segments to differ. However, in the event that users fail to encrypt their data, confidentiality cannot be ensured, nor are data safe from prying cloud storage providers. On the other hand, shared storage is where deduplication solutions in software and databases are most important. Hybrid clouds aim to bring together the advantages of public cloud computing—such as scalability, dependability, fast deployment, and potential cost savings—with the security and increased control and management of private clouds. All files must be divided into smaller blocks, and then each block's fingerprints must be created using MD5 or SHA-1 and used as an identity. Subsequently, these markers are employed to locate specific fingerprints. Because of their far lower probability than disk corruption, hash collisions are essentially ignored in the results of current study. Stated differently, two fingerprints are similar if and only if the two corresponding pieces of information are likewise similar. Additionally, even if any length of message can be entered into either cryptographic hash algorithm, the outcomes are preset fingerprints. The lengths of MD5 and SHA-1 are 128 bits and 160 bits, respectively. SHA-1 outperforms MD5 overall, but at the expense of a higher processing overhead and a slightly slower operating rate. The disk bottleneck problem severely restricts the deduplication system's speed since it results from the requirement to query an index of every fingerprint that is currently in existence. Only part of the index is stored in memory; the rest is stored on disk because there is usually not much memory available. To increase index lookup hit rates, memory index optimization is therefore crucial. Initially, DDFS [11, 12] applied the bloom filter to a de-duplication system, where the fingerprint index is stored on disk and the bloom filter is maintained in memory. However, compared to Sparse indexing, Bloom filter consumes more RAM [13]. By using a sparse index and block similarity sampling instead of the entire index, the latter reduces the search space and memory use.

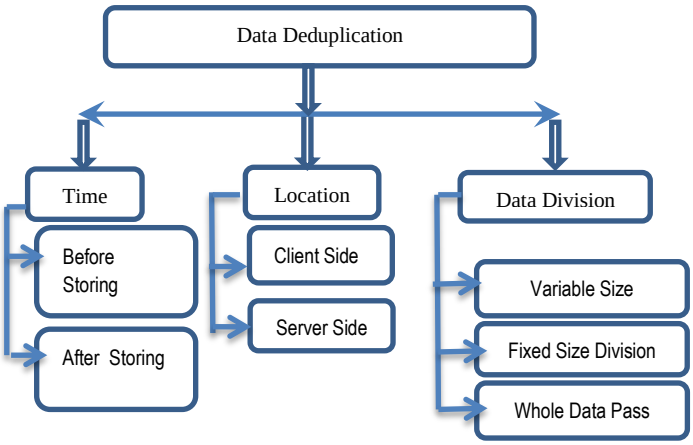

Fig -1: Strategies of Data deduplication

There are multiple methods for identifying and removing duplicate data. All ultimately lead to the same conclusion: decrease the size to conserve storage. The data deduplication strategies are displayed in Figure 1. To maximize storage, researchers begin looking into deduplication methods [5]. The data can be deduplicated in multiple ways. Previous studies reveal that over 90% of the data kept in cloud backups is replicated. This study focuses on the method, which disregards uploading redundant data. The data is checked using a hash method; if it matches, the upload is ignored; if not, it is counted as unique, and the upload is continued to store the data.

Data Division:-Using this method, the data is split into a series of bytes, and the redundancy is tested using the divided blocks. Deduplication is achieved by storing just the distinct block. Various data division strategies are available to eliminate redundant data.

Location:- In the cloud, a kind of network. One place where the data could be kept is on the client side, and another would be on the server side. The location determines where the deduplication operation is carried out.

Time:-One of the most crucial factors in the processing and computing fields is time. Reduce the number of duplicate files, increase performance, and reduce processing time. Depending on the time, there are two different kinds of

deduplication techniques. The first occurs prior to the data being stored in storage, and the second occurs following data storage.

Methods	Scheme for Encryption	Utilizing a deduplication strategy
Message-locked encryption and safe deduplication techniques	Encryption locking messages	File Level
Block-Level Message-Locked Encryption (BL-MLE): A Secure Method for Large File Deduplication	Block Level Encryption with Message Locking	Dual level: File level and Block level
HEDup: Homomorphic Encryption and Secure Deduplication	Homomorphic encryption	File level
DupLESS: Encryption for Deduplicated Storage Assisted by a Server	Enhanced encryption at the message levelto bolster defense against brute force attacks	File level
ClouDedup: Protect Deduplication for Cloud Storage Using Encrypted Data	Convergent encryption with additional measures for access control	File level
Safe Duplication Using Dependable and Effective Convergent Key Management	convergent encryption	Block level
An architecture for safe cloud computing called "twin clouds"	Convergent encryption	File level
A hybrid cloud strategy for authorized, secure deduplication	Convergent encryption	File level
Secure Data Deduplication	Convergent encryption	File level
A safe method of data deduplication for cloud storage	symmetric encryption on popularly categorized data	File level

Table 1 contrast of deduplication methods applied to Encrypted data

The table above illustrates the many encryption techniques used to store data in the cloud. Each of these techniques has a disadvantage, thus in order to get around them, new techniques that may preserve data security and prevent deduplication while storing data in the cloud must be proposed. Here, we offer two methods that optimize data while it is being stored in the cloud.

Critical Security Control Services	AWS (2020)	Azure (2020)	Cloud (2020)	Oracle (2020)	IBM (2020)	Alibaba (2020)
Firewall & ACLs	✓	✓	✓	✓	✓	✓
IPS/IDS	3rd Party	✓	3rd Party	3rd Party	3rd Party	✓
Web Application Firewall (WAF)	✓	✓	✓	✓	✓	✓
SIEM & Log Analytics	✓	✓	✓	✓	✓	✓
Antimalware	3rd Party	✓	3rd Party	3rd Party	3rd Party	✓
Data Loss Prevention (DLP)	✓	✓	✓	3rd Party	3rd Party	✓
File Integrity Monitoring (FIM)	3rd Party	✓	3rd Party	3rd Party	3rd Party	3rd Party
Key Management	✓	✓	✓	✓	✓	✓
Encryption At Rest	✓	✓	✓	✓	✓	✓
DDoS Protection	✓	✓	✓	✓	✓	✓
Email Protection	3rd Party	✓	✓	3rd Party	3rd Party	3rd Party
SSL Decryption & Reverse Proxy	✓	✓	✓	3rd Party	✓	✓
Endpoint Protection	3rd Party	✓	3rd Party	3rd Party	3rd Party	✓
Certificate Management	✓	✓	3rd Party	3rd Party	✓	✓
Container Security	✓	✓	✓	✓	✓	✓
Identity and Access Management	✓	✓	✓	✓	✓	✓
Privileged Access Management (PAM)	3rd Party	✓	3rd Party	3rd Party	3rd Party	3rd Party
Multi-Factor Authentication	✓	✓	✓	✓	✓	✓
Centralized Logging & Auditing	✓	✓	✓	✓	✓	✓
Load Balancer	✓	✓	✓	✓	✓	✓
LAN	✓	✓	✓	✓	✓	✓
WAN	✓	✓	✓	✓	✓	✓
VPN	✓	✓	✓	✓	✓	✓
Governance Risk and Compliance Monitoring	✓	✓	✓	3rd Party	3rd Party	✓
Backup and Recovery	✓	✓	✓	✓	✓	✓
Vulnerability Assessment	✓	✓	✓	✓	✓	✓
Patch Management	✓	✓	3rd Party	✓	3rd Party	3rd Party
Change Management	✓	✓	3rd Party	3rd Party	3rd Party	✓

Table 2 Critical Security Control Services Provided by Major Cloud Services Providers

2.Related work

Kwon H, et.al [15] This Data security is maintained while redundant data copies are removed by the secure deduplication technology. Using a key generated from the content of the file, Convergent Encryption (CE) is used to encrypt and decrypt data at the file level. [15].

Akhila K et.al [16] to save storage space, users delegate the ciphertext (CT) to the Cloud Server (CS) while keeping the encryption key. Updating the CT in the central cloud and user-level public keys without disclosing the private keys ensures consistent privacy.[16].

Bellare et.al [17] Suggest an encryption system in which the message's own key is used for both encryption and decryption.The encryption algorithm uses the key K to create the message's cipher text C after the MLE key generation process translates the message M to it. After that, ciphertext C is mapped to tag T, which the server uses to check for duplicates. Because the keys used in the MLE scheme are fixed and shorter in length, there is less storage cost.[17].

Puzio et.al in [18] Offer ClouDedup, a safe and effective storage solution that combines block level key management with convergent key encryption to ensure both data confidentiality and block level deduplication[7, 2].The ClouDedup architecture incorporates access control and user authentication measures in order to thwart known attacks on convergent encryption. As a result, user-performed convergent encryption is topped with server encryption. Every data segment has a signature associated with it, which must be confirmed in order to retrieve the data. The architecture now includes a metadata manager (MM) to handle block level key management.MM employs a signature database to store meta data about signatures for meta data management, a file table to store meta data about files, and a pointer table to manage storage.[18]

Zhou et.al [19[18]'s primary goals are to combat the issue of large key space overhead and fend off brute force attacks. This approach makes use of Multi Level Key Management (MLK) and User Aware Convergent Encryption (UACE) for that purpose.Here, UACE is used to do both single user block level and cross-user file level deduplication. Level keys for files While chunk level keys are generated with user assistance, convergent encryption keys are generated through server assistance. When chunk keys are encrypted with file level keys, key space is not increased despite an increase in the number of sharing users.Additionally, in order to completely eliminate the possibility of a single point of failure, this system makes use of numerous key servers. These servers are connected to each other via Shamir's secret sharing scheme [20], and each share-level key is derived from a file level key.

3. PROPOSED SYSTEM 1

3.1 SYSTEM 1:-The file's hash is generated by the MD5 algorithm, and encryption is accomplished using the AES method. The majority of recent studies on cloud storage de-duplication concentrate on cloud storage security. For safe de-duplication, Shen advises utilizing proxy encryption and version control. Put differently, it ensures semantic security for less popular content while offering more storage and laxer security for popular data.

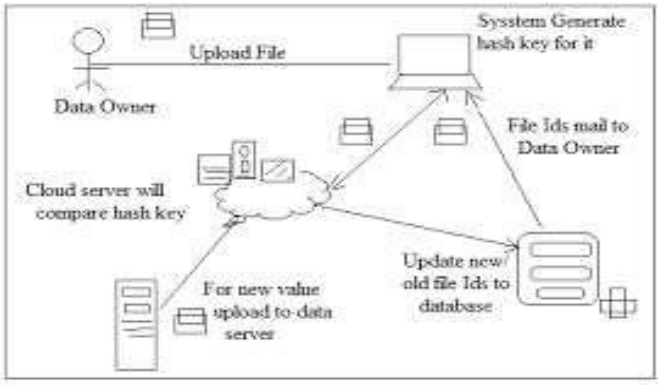

Fig-2 System Model for Uploading hash value data after encryption

Encrypted data de-duplication can be used for data optimization. Data cannot be saved on the cloud after the hash value matches the previous hash value. The comparison of hash values is shown in Fig. 1 above. The basic idea behind DE-duplication is to store duplicate data (files or blocks) just once. Nevertheless, customers cannot guarantee confidentiality or protect their data from snooping cloud storage providers if they neglect to encrypt their data [8]. Cloud providers maintain a unique copy of duplicate data, which dramatically reduces the cost of their bandwidth, storage, and data transmission. De-duplication is currently being used by a lot of cloud storage businesses because it has been demonstrated to produce considerable space and cost

savings. Consumers require low ownership costs, adaptability, and the confidentiality and security of data that encryption ensures [9]. The de-duplication process is done on the cloud server by keeping a single duplicate of every file that is uploaded in order to save space. Complete protection is provided by the encrypted format that data appears in when it is uploaded to the cloud. Before uploading a file to the cloud, the user needs to enter their login details. Users are able to upload numerous files after logging in. The recommended software/system calculates the document's hash internally and compares it to the other documents that are kept on cloud storage. The user is prompted with the message "This document is already present; you cannot upload this document" if the system finds a match with the current document. "The document has been successfully uploaded if there is no match. The file's hash is generated using the MD5 technique, and encryption is accomplished using the AES algorithm. The following are the main components of the suggested scheme [10].

3.1.1 Sendin

The hash value of the file is computed prior to uploading, and it is then checked to determine whether any duplicates have already been registered on the metadata server with the same hash value. If the file is fresh, it will be uploaded to the cloud, have new data added, and be encrypted.

3.1.2 Revision

The file will have its metadata updated if it already exists, and the system may need to create or delete clones of it accordingly.

3.1.3 Eliminate

The de-duplicator keeps track of how many files the user wants to delete that share the same hash value. If the hash is only given once, then all copies of the file will be deleted. If any other files reference the hash, only the metadata will be updated.

3.1.4 Algorithms

We are using three distinct encryption techniques to protect the files on the device. The algorithms are used for experimental analysis. The selection of these is based on how frequently they are found in the corpus of contemporary literature. The list of algorithms is as follows.

3.1.4.1 SHA-1

SHA1, the secure hash algorithm, can be used to transform any text or image into a message digest. The data's integrity is confirmed using the hash value. It takes in an input and outputs a message digest, which is a hash value of 160 bits (20 bytes). It is used to confirm data integrity and is meant to safeguard data. By doing this, brute force attacks against the AES key during encryption are prevented.

3.1.4.2 MD5

In 1991, Ronald Rivest developed MD5. Popular hash function is the MD5 message-digest method, which has a hash value of 128 bits. It acts as a checksum to guarantee that the data is accurate. The hash changes by approximately 50% for every byte that is altered from the original text. It is computationally impossible to retrieve the text from the hash because it is a one-way function. In this project, MD5 is used to determine a file's hash value.

3.1.4.3 AES

AES stands for Advanced Encryption Standard. It is used to produce the cipher key. There are ten rounds for 128-bit keys, twelve for 192-bit keys, and fourteen for 256-bit keys. The AES key generated upon login is utilized for both encryption and decryption of files.Although cloud storage systems are become more and more popular, they may not always provide consumers with convenient or inexpensive network storage. However, because redundant data takes up a lot of storage space, cloud storage systems are under increasing pressure to store more and more data as data rises exponentially. Data de-duplication can effectively reduce the amount of data by deleting redundant data from storage systems, and encryption enables the transfer of sensitive data over cloud platforms. The basic idea behind de-duplication is to store duplicate data (files or blocks) just once. Segments of data after encryption will be unique. However, if customers do not encrypt their data, there is no way to ensure secrecy and no defense against nosy cloud storage providers. Cloud providers store a unique copy of duplicate data, which dramatically reduces the cost of their bandwidth, storage, and data transmission. De-duplication is currently being used by a lot of cloud storage businesses because it has been demonstrated to produce considerable space and cost savings.

3.2 PROPOSED SYSTEM 2

3.2 SYSTEM 2:-Utilizing a data deduplication technique based on titles and keywords for encrypted documents.RAKE's "keyword extraction" technique makes it possible to locate crucial terms or phrases within a document. The first two steps in this approach are tokenizing the text into words or phrases and pre-processing it to eliminate special characters and common terms. RAKE then identifies potential keywords based on co-occurrence and word frequency, ranks these candidates, and selects the top-ranked keywords as the most relevant and representative phrases for the document. Keyword extraction is followed by document encryption, which encrypts the entire document to ensure data protection. Using encryption, plain text can be changed into unintelligible ciphertext by using encryption algorithms and a secret key. This preserves private information and prevents unauthorized parties from accessing the document's contents. Finally, Matching Similar Documents uses the document's title and extracted keywords to find related documents in a database. This technique compares keyword sets for content evaluation and ranks documents based on similarity scores, allowing the most relevant documents to be retrieved while retaining data privacy through encryption. As a first filter, title matching is employed [6][7].

3.2.1 Title matching

Finding similarities in document names is an essential task in several fields, including information retrieval, content management, and plagiarism detection. Due to the massive volume of writing that is produced daily, effective ways to rapidly detect papers that are duplicates are becoming more and more necessary. A helpful strategy to address this issue is to examine the document names, which frequently offer a concise summary of the contents. The intention is to provide a systematic process that can consistently ascertain the level of resemblance between two titles, thereby revealing the potential relationship of the underlying papers.

3.2.2 Lowercase Conversion: It is necessary to alter every character in the titles to lowercase.

3.2.3 Eliminate Punctuation: Removes all punctuation with Take.

3.2.4 Tokenization: Divide the titles into distinct categories.

3.2.5 Extract Keywords: Ascertain the most crucial keywords for every title. To do this, one might either use nouns or manually identify words that express the main notion.

3.2.6 Match Keywords: See if the keywords in the two titles have any overlap. Illustrate the similarity Decide on a threshold for what is "similar." For example, you may classify them as comparable if the titles contain at least 50% of the same keywords.

3.2.7 Ascertain Similarity If the titles have a higher percentage of matching terms than your requirements, they are considered comparable elements.

4 RESULTS AND DISCUSSIONS

4.1 Result of System 1

The file's hash is generated by the MD5 algorithm, and encryption is accomplished using the AES method. Java has aided us in implementing our algorithms. We have used the built-in Java tools and packages to implement our cryptographic methods and access Google Drive. The recommended system is connected to Google Drive via library files that are compatible with Google Drive.

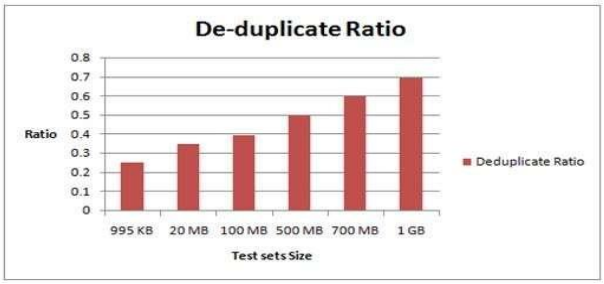

Fig-3 De-duplicate Ratio Graph

Fig-3 Display the recommended de-duplicate ratio for test sets as a graph in Figure 2; the ratio rises as the test set sizes are ascertained.

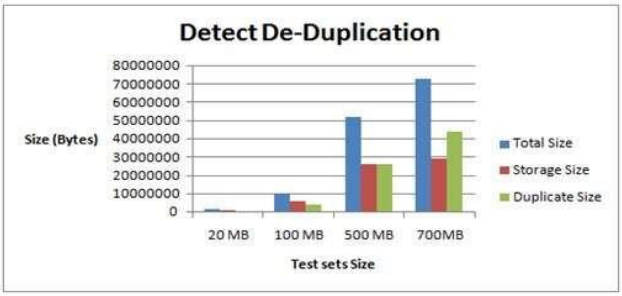

Fig-4 Total Size,Storage Size Duplicate Size (Bytes) Graph

Fig-4 Present the file size findings for each of the three test sets (total, storage, and duplicate sizes). The total size in bytes exceeds both the duplicate size and storage size. Compare the times of AES 128, AES 192, and AES 256 for files up to 1 MB, demonstrating that AES 256 uses less energy and requires less time than the other algorithms.Show the file size results (total, storage, and duplicate sizes) for each of the three test sets in Figure 4. The duplicate size and storage size are both less than the total size in bytes. AES 256 uses less energy and takes less time than the other algorithms when you compare the times of AES 128, AES 192, and AES 256 for files up to 1 MB.

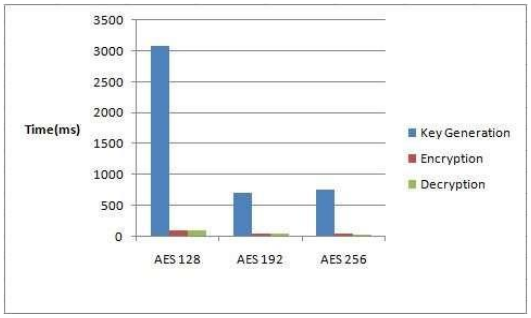

Fig-5 Time Comparison of AES Algorithm

For files up to 1 MB, AES Algorithms compare the timings of AES 128, AES 192, and AES 256, showing that AES 256 consumes less time and energy than the other two.

4.2 Result of System 2

4.1.1 Data Deduplication Technique for Encrypted Documents

Method	Deduplication efficiency (DE)
AppAware	51.1
∑-Dedupe	45.6
AppDedupe	53.6
Proposed Method	55.2

Table 3 Deduplication efficiency

The deduplication efficiency (DE) values for AppAware, ∑-Dedupe, AppDedupe, and a suggested method, among other ways, are displayed in Table 3 and Fig. 6. The efficiency of a data deduplication technique is measured by how well it eliminates redundant or unneeded data while optimizing storage capacity. The amounts show how much deduplication can be accomplished with each method. The suggested technique sticks out from the others since it delivers the maximum deduplication efficacy of 55.2%. This implies that it has a higher ability to locate and remove redundant data. The aforementioned statistic is essential for evaluating the efficacy of deduplication algorithms. A higher number indicates a less ineffective way to cut down on redundant data.

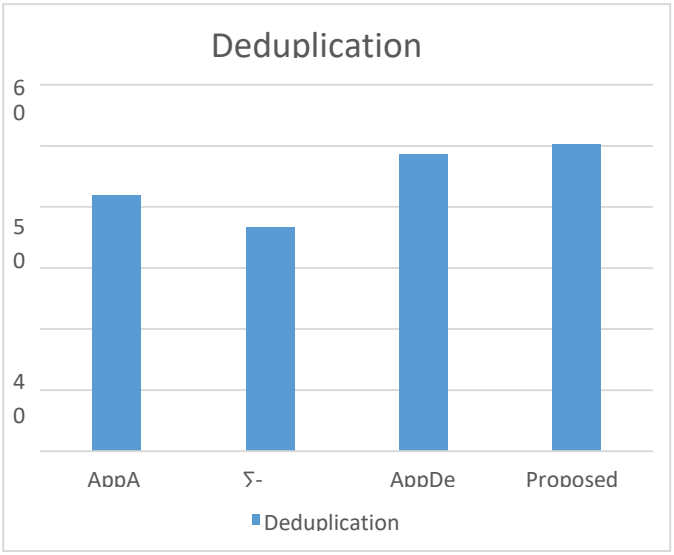

Fig-6 Normalized deduplication ratio

5. CONCLUSION AND FUTURE WORK

By integrating these components in a comprehensive manner, the method safely and effectively finds duplication in encrypted documents. Usually, deduplication techniques are used in the cloud to reduce storage capacity. To attain high levels of security for encrypted data on the cloud, ownership of the same material must be preserved even after it has been uploaded several times. We propose a client-side de-duplication program, which we name De-duplication software. Our proposed program can do a lot of things with a graphical user interface, like uploading, downloading, and registering files. The software gives users peace of mind about the security of the data they save on the cloud by enabling customizable support for encrypted data. The computer simulation results show that it is feasible. The proposed methodology focuses on three major components: title matching, keyword extraction, and the original objective of deduplicate encrypted texts. To find any potential duplication, the first stage in the process is to compare the titles of the documents. The next stage involves extracting

keywords from the encrypted documents, which is essential for maintaining data privacy and comprehending context and content to more accurately identify duplicates.

6. ACKNOWLEDGEMENT

I am grateful to all the faculty & Staff members of Computer science & Engineering department for their kind cooperation and help I acknowledge my sincere thanks and deep sense of gratitude to my guide Dr. Harsh Lohiya (Department of Computer science & Engineering, School of Engineering, Sri Satya Sai University of technology and medical science, Sehore MP) for his valuable help and guidance. I am thankful to him for encouraging to me in completing this research paper entitled "Cloud data Optimization by encrypted data deduplication storage technique".

References

[1] Parast, Fatemeh Khoda, Chandni Sindhav, Seema Nikam, Hadiseh Izadi Yekta, Kenneth B. Kent, and Saqib Hakak. "Cloud computing security: A survey of service-based models." Computers & Security 114 (2022): 102580.

[2] Subramanian N, Jeyaraj A (2018) Recent security challenges in cloud computing. Compute Electr Eng 71:28–42. https://doi.org/10.1016/j. compeleceng.2018.06.006

[3] R.Zhou, M. Liu, and T. Li, "Characterizing the efficiency of data deduplication for big data storage management," *Proc. - 2013 IEEE Int. Symp. Workload Charact. IISWC 2013*, no. April, pp. 98–108, 2013, doi: 10.1109/IISWC.2013.6704674.

[4] N. Sharma, A. V. Krishna Prasad, and V. Kakulapati, "Data deduplication techniques for big data storage systems," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 10, pp. 1145–1150, 2019, doi: 10.35940/ijitee.J9129.0881019.

[5] Jiang S, Jiang T, Wang L (2017) Secure and efficient cloud data Deduplication with ownership management. IEEE Trans Serv Comput 12: 532–543. https://doi.org/10.1109/TSC.2017.2771280

[6] Wang L, Wang B, Song W, Zhang Z (2019) A key-sharing based secure deduplication scheme in cloud storage. Inf Sci (Ny) 504:48–60. https://doi.org/10.1016/j.ins.2019.07.058

[7] Akhila K, Ganesh A, Sunitha C (2016) A study on Deduplication techniques over encrypted data. Procedia Comput Sci 87:38–43. https://doi.org/10.1016/j.procs.2016.05.123

[8] Koo D, Hur J (2018) Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing. Futur Gener Comput Syst 78:739–752. https://doi.org/10.1016/j.future.2017.01.024

[9] Li S, Xu C, Zhang Y (2019) CSED: client-side encrypted deduplication scheme based on proofs of ownership for cloud storage. J Inf Secur Appl 46:250–258. https://doi.org/10.1016/j.jisa.2019.03.015

[9] Zuhair S. Al-sagar, Mohammad S. Saleh and Aws Zuhair Sameen,"Optimizing the Cloud Storage by Data De-duplication", International Research Journal of Engineering and Technology , e-ISSN: 2395 - 0056 ,Volume 02, Issue 09 ,pp. 2524-2527,2015.

[10] Renuka C. Deshpande and S. S. Ponde," De-duplication Using SHA-1 and IBE with Modified AES", International Journal of Science and Research, Volume 6,Issue 2,pp. 1886-1889,2016.

[11] W. Xia *et al.*, "A Comprehensive Study of the Past, Present, and Future of Data Deduplication," *Proc. IEEE*, vol. 104, no. 9, pp. 1681–1710, 2016, doi: 10.1109/JPROC.2016.2571298.

[12] E. Manogar and S. Abirami, "A study on data deduplication techniques for optimized storage," *6th Int. Conf. Adv. Comput. ICoAC 2014*, pp. 161–166, 2015, doi: 10.1109/ICoAC.2014.7229702.

[13] Manjesh. K.N and R K Karunavathi,"Secured High throughput implementation of AES Algorithm", International Journal of Advanced Research in Computer Science and

Software Engineering, Volume 3, Issue 5,pp. 1193-1198,2013.

[14] N. Kumar, S. Antwal, G. Samarthyam, and S. C. Jain, "Genetic optimized data deduplication for distributed big data storage systems," *4th IEEE Int. Conf. Signal Process. Comput. Control. ISPCC 2017*, vol. 2017-Janua, pp. 7–15, 2017, doi: 10.1109/ISPCC.2017.8269581.

[15] Kwon H, Hahn C, Kim D, Hur J (2017) Secure deduplication for multimedia data with user revocation in cloud storage. Multimed Tools Appl 76:5889–5903. https://doi.org/10.1007/s11042-015-2595-4

[16] Akhila K, Ganesh A, Sunitha C (2016) A study on Deduplication techniques over encrypted data. Procedia Comput Sci 87:38–43.

[17] Bellare, Mihir, Sriram Keelveedhi, and Thomas Ristenpart. "Message-locked encryption and secure deduplication." Advances in Cryptology–EUROCRYPT 2013. Springer Berlin Heidelberg, 2013. 296-312.

[18] Puzio, Pasquale, Refik Molva, Melek Önen, and Sergio Loureiro."ClouDedup:Secure Deduplication with Encrypted Data for Cloud Storage." .In Cloud Computing Technology and Science(CloudCom),2013 IEEE 5th International Conference on (Volume:1)p.363 – 370

[19] Zhou, Yukun, Dan Feng, Wen Xia, Min Fu, Fangting Huang, Yucheng Zhang, and Chunguang Li. "SecDep: A User-Aware Efficient Fine- Grained Secure Deduplication Scheme with Multi-Level Key Management." Mass Storage Systems and Technologies (MSST), 2015 31st Symposium on, pp. 1-14.

[20] A. Shamir,"How to share a secret," Commun. ACM, vol. 22, no. 11,pp. 612–613, 1979.1.N. N. Pachpor and P. S. Prasad, "Securing the Data Deduplication to Improve the Performance of Systems in the Cloud Infrastructure," in Performance Management of Integrated Systems and its Applications in Software Engineering, Springer Singapore, 2020, pp. 43–58.

[21] W. Bin Kim and I. Y. Lee, "Overview of Data Deduplication Technology in a Cloud Storage Environment," in Lecture Notes in Electrical Engineering, 2020, vol. 536 LNEE, pp. 465–470, doi: 10.1007/978-981-13-9341-9_80.

[22] V. S. R. and D. K. Singh, "Secure Deduplication Techniques: A Study," Int. J. Comput. Appl., vol. 137, no. 8, pp. 41–43, 2016, doi: 10.5120/ijca2016908874.

[23] J. Malhotra and J. Bakal, "A survey and comparative study of data deduplication techniques," 2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015, vol. 00, no. c, pp. 0–4, 2015, doi: 10.1109/PERVASIVE.2015.7087116.