Cloud data Optimization by encrypted data deduplication storage 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

Various cloud services can be facilitated by cloud computing with cloud storage being the most popular one. Storage and performance are experiencing a lot of problems. Data holders often store sensitive information in an encrypted form to ensure its privacy and security. Owning data stored on the cloud is a major challenge due to the lack of support for access to data in more schemes. One method for lowering the quantity of storage space required by an organization to store its data is data deduplication. With the help of two main goals, this study seeks to strike a compromise between security and storage optimization. The goal suggests a novel title- and keyword-based deduplication method for documents that are encrypted. To help with the early detection of possible duplication, the model starts with title matching, which compares the titles of the documents. Second, the paper using the MD5 technique to determine the file's hash and for encryption we are using AES algorithm **.***On the other hand, although deduplication is necessary to achieve optimal storage, encryption is necessary to ensure data security. Thus, to guarantee safe and effective storage, deduplication and encryption must cooperate.*

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

1. INTRODUCTION

One such storage optimization method that keeps duplicate copies of data out of storage is deduplication. Data saved in the cloud and other big storage places is now encrypted to maintain security [1]. 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 [2][3]. Data has become vital for both individuals and organizations in the modern digital environment. It is unacceptable to save duplicate data contents since the amount of data generated grows exponentially over time [4][5]. Therefore, using storage optimization strategies is a must for tasks involving huge storage regions, such as cloud storage. In this work, several approaches to this problem are examined. A method called data deduplication is used to find and remove duplicate data chunks from a storage device. Numerous companies are utilizing this technology to provide effective data storage, but it has advantages and disadvantages of its own. However, deduplication solutions in software and databases are more significant at the shared storage level [8]. The goal of hybrid clouds is to combine the security and more control and management of private clouds with the benefits of public cloud computing, such as scalability, dependability, fast deployment, and possible cost savings. Following the division of all files into smaller blocks, each block's fingerprints must be generated and used as an identification using MD5 or SHA-1. These identifiers are then used to hunt up individual fingerprints. The results of current research essentially overlook hash collisions because their probability is orders of magnitude lower than that of disk corruption. [10]. In other words, two fingerprints are identical if and only if the two matching pieces of data are also identical. Furthermore, while a message of any length can be entered into either of the cryptographic hash functions, the results are predetermined fingerprint MD5 and SHA-1 have lengths of 128 bits and 160 bits, respectively. Although SHA-1 performs better overall than MD5, its processing overhead is larger and its operating rate is somewhat slower.The disk bottleneck issue, which arises from the need to query an index of all fingerprints currently in existence, significantly limits the deduplication system's performance. Because there is typically relatively little memory available, only a small portion of the index is kept in memory; the remainder is kept on disk. Therefore, memory index optimization is essential to raise index lookup hit rates. The Bloom filter was initially applied by DDFS [11, 12] to a de-duplication system in which the bloom filter is kept in memory and the fingerprint index is kept on disk. Nonetheless, Bloom filter uses more memory than Sparse indexing [13]. The latter restricts the range of search and uses less memory by sampling based on block similarity and a sparse index rather than the whole index.

2. RELATED WORKS

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.

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 level to 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

Above table shows different encryption methods are used for storing the data in cloud these methods have different drawbacks to overcome these drawbacks we need propose new methods which can maintain the security of data also used to avoid the deduplication in storing the data in cloud. Here we are proposing two models which work for optimizing the data while storing over the cloud. To use a data deduplication method for encrypted documents based on titles and the MD5 algorithm is used to create the file's hash, and the AES technique is used for encryption. .These two methods we are descussing here to resolve the problem of storing the data over the cloud.

3. PROPOSED MODEL

3.1 Model 1- To use a data deduplication method for encrypted documents based on titles and keywords

The technique of "keyword extraction" (using RAKE) allows one to find important terms or phrases in a document. Tokenizing the text into words or phrases and pre-processing it to remove common words and special characters are the first two processes in this approach. Based on co-occurrence and word frequency, RAKE then finds possible keywords, rates these candidates, and finally chooses the highest-ranked keywords as the most pertinent and representative phrases for the document. The process of document encryption, which encrypts the complete document to guarantee data security, comes after keyword extraction. Encryption uses encryption algorithms and a secret key to transform plain text into unintelligible ciphertext. This protects confidential data and keeps outsiders from viewing the contents of the document. Lastly, Matching Similar Documents locates related documents in a database by utilizing the title and extracted keywords of the document. The most pertinent documents may be retrieved while maintaining data secrecy through encryption thanks to this technique, which also compares keyword sets for content evaluation and ranks documents based on similarity scores. Title matching is used as an initial filter [6][7].

3.1.1 Title matching

Comparing document titles to find commonalities is a crucial activity in a number of disciplines, such as content management, plagiarism detection, and information retrieval. Owing to the enormous amount of writing that is created every day, there is an increasing need for efficient methods to quickly identify documents that are duplicates. One useful tactic to deal with this problem is to look at document names, which often provide a brief synopsis of the contents. The goal is to offer a methodical procedure that can reliably determine the degree of similarity between two titles, thereby disclosing the possible association of the underlying papers.

3.1.2 Lowercase Conversion: All of the characters in the titles should be changed to lowercase.
3.1.3 Eliminate Punctuation: Take removes all punctuation.
3.1.4 Tokenization: Segment the titles into separate terms.

3.1.5 Extract Keywords: Determine which keywords are most important in each title. Either choosing nouns or manually identifying words that convey the main idea could be used to accomplish this.

3.1.6 Match Keywords: Look for any similarities between the keywords in the two titles.
explain Similarity Establish a cutoff point for what qualifies as "similar." For instance, if at least 50% of the keywords in the titles match, you might consider them comparable.
3.1.7 Ascertain Similarity: The titles are deemed comparable elements if the percentage of matching keywords is higher than your criteria

3.2 Model 2 - The MD5 algorithm is used to create the file's hash, and the AES technique is used for encryption. Recent research on cloud storage de-duplication mostly focuses on cloud storage security. Shen suggests using version control and proxy encryption for safe de-duplication. In other words, for popular data, it provides greater storage and laxer security while ensuring semantic security for content that is less popular.

PROPOSED SYSTEM MODEL

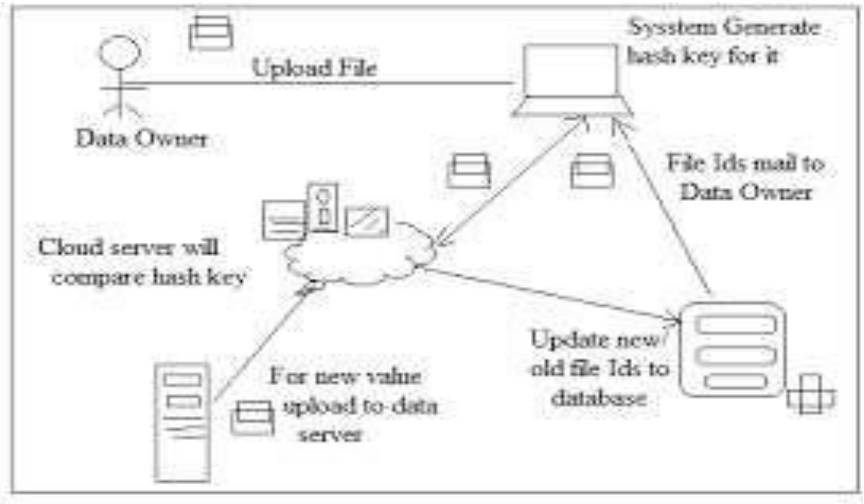
Upload File
Data Owner
Sysstem Generate hash key for it
File Ids mail to Data Owner
Cloud server will compare hash key
Update new/ old file Ids to database
For new value upload to data server

Fig-1 System Model for Uploading hash value data after encryption

Optimization of data can be done with encrypted data de-duplication. Once the hash value matching with old hash value data cannot stored over cloud. Above Fig 1 Shows how hash value is compared. DE-duplication works on the simple principle of storing duplicate data (files or blocks) only once. However, in the event that users fail to encrypt their data, confidentiality cannot be ensured, nor are data safe from prying cloud storage providers [8]. Cloud providers significantly lower the cost of their storage, bandwidth, and data transport by keeping a unique copy of duplicate data. Numerous cloud storage companies are presently implementing de-duplication since it has been shown to result in significant space and cost savings. Users need cheap ownership costs, flexibility, and the assurance of data security and secrecy provided by encryption [9]. By maintaining a single copy of each uploaded file, the de-duplication procedure is carried out on the cloud server to conserve space. When data is uploaded to the cloud, it is shown in encrypted form, offering complete protection. The user must input their credentials to log into their account before they may upload a file to the cloud. Once logged in, users are able to upload multiple files. The document's hash is internally computed by the suggested software/system and compared to the remaining documents stored in the cloud. If the system detects a match with the current document, it prompts the user, saying, "This document is already present; you cannot upload this document." If no match is found, the document is uploaded successfully. We use the MD5 algorithm to create the file's hash, and the AES algorithm is used for encryption. The proposed plan includes the following primary elements [10].

3.2.1Sendin

Before uploading, the file's hash value is calculated and then examined to see whether there are any duplicates with the same hash value already registered on the metadata server. In the event that the file is new, it will have new information added, be encrypted, and be uploaded to the cloud.

3.2.2Revision

If the file is already in existence, its metadata will be modified, and the system might have to make or remove clones of it in accordance.

3.2.3 Eliminate

The de-duplicator counts the number of files that the user wishes to remove and that reference the same hash value. All copies of the file will be removed if the hash is only mentioned once. Only the metadata, however, will be updated if any other files make reference to the hash.

3.2.4 Algorithms

To safeguard the files on the drive, we are employing three different kinds of cryptographic methods. For experimental analysis, the algorithms are employed. They are chosen based on how frequently they appear in the body of current literature. The following is a list of algorithms:

3.2.4.1MD5

Ronald Rivest created MD5 in 1991. With a 128-bit hash value, the MD5 message-digest method is a popular hash function. It serves as a checksum to ensure the accuracy of the data. A single bit changed in the original text causes the hash to alter by about 50%. Being a one-way function, it is computationally impossible to extract the text from the hash. In this project, the hash value of a file is calculated using MD5.

3.2.4.2SHA-1

Any text or image can be converted into a message digest using the secure hash method SHA1. The hash value is used to verify the integrity of the data. It receives an input and outputs a message digest, which is a 160-bit (20-byte) hash value. It

is intended to protect data and is employed to verify data integrity This protects the AES key during encryption against brute force attacks.

3.2.4.3AES

Advanced Encryption Standard or AES. The cipher key is generated using it. The number of rounds for 128-bit keys is 10, for 192-bit keys it is 12, and for 256-bit keys it is 14. When encrypting and decrypting files, the AES key that is produced upon login is used.

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. By removing redundant data from storage systems, data de-duplication can efficiently reduce the amount of data, and encryption allows the sharing of sensitive data over cloud platforms. De-duplication works on the simple principle of storing duplicate data (files or blocks) only once. Data segments following encryption will differ. 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 significantly lower the cost of their storage, bandwidth, and data transport by storing a unique copy of duplicate data. Numerous cloud storage companies are presently implementing de-duplication since it has been shown to result in significant space and cost savings.

4. RESULTS AND DISCUSSIONS

4.1 Result of Model 1

4.1.1 Data Deduplication Technique for Encrypted Documents

Method	Deduplication efficiency (DE)
AppAware	52.1
∑-Dedupe	46.6
AppDedupe	56.6
Proposed Method	58.2

Table 2: Deduplication efficiency

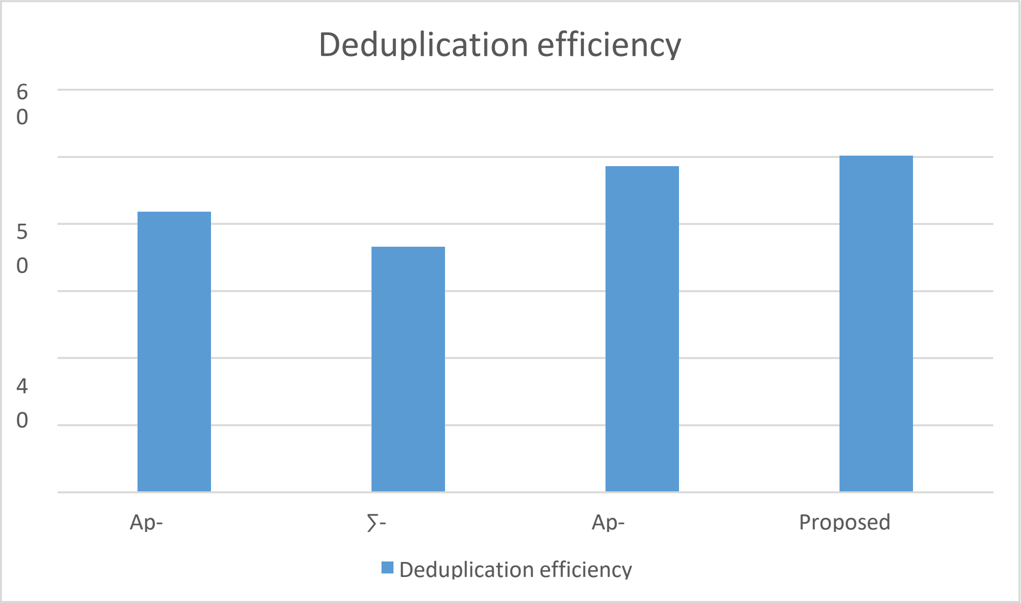

Fig-2: Normalized deduplication ratio

The data shown in Table 2 and Fig-3 shows the deduplication efficiency (DE) values for AppAware, ∑-Dedupe, AppDedupe, and a recommended method, among other approaches. Efficiency in deduplication relates to the effectiveness of a data deduplication method in cutting out unnecessary or duplicate data, maximizing storage space. The numbers represent the amount of deduplication that each method is able to achieve. Among the strategies that were looked at, the recommended strategy stands out since it achieves the highest deduplication effectiveness of 58.2%. This suggests that it can find and eliminate duplicate data more effectively. The metric indicated above is crucial for assessing how well deduplication techniques work. A higher figure denotes a more effective method of reducing data redundancy.

4.2 Result of model 2- The MD5 algorithm is used to create the file's hash, and the AES technique is used for encryption. Java has helped us put our algorithms into practice. We have implemented our cryptographic algorithms and accessed Google Drive by utilizing built-in Java libraries and packages. Google Drive compatible library files are utilized to link the suggested system to Google Drive.

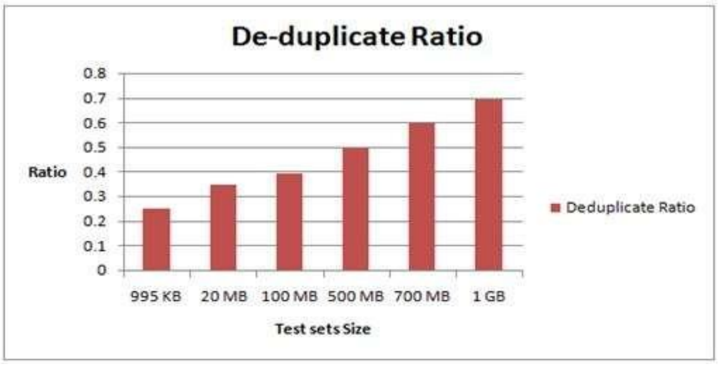

Fig-3 De-duplicate Ratio Graph

Fig-2 Present the suggested de-duplicate ratio for test sets as a graph, with the ratio increasing in accordance with the size of the test sets as determined.

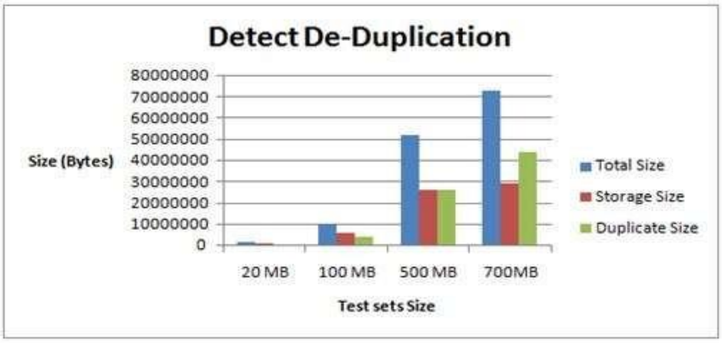

Fig-4 Total Size, Storage Size, Duplicate Size (bytes) Graph

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.

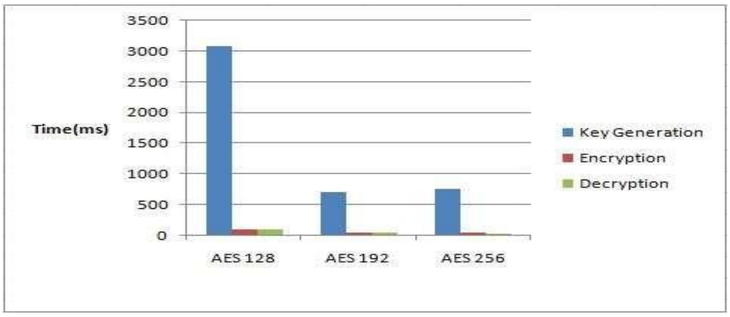

Fig-5 Time Comparison of AES Algorithm

Fig-5 AES Algorithms 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 time than the other two.

5. CONCLUSION AND FUTURE WORK

The suggested methodology emphasizes three essential elements, namely title matching, keyword extraction with the initial goal of deduplicate encrypted texts. The initial step in the procedure involves comparing document titles to identify any possible duplicates. Subsequently, keywords are extracted from the encrypted documents, which is a crucial step in protecting data privacy while understanding context and content to detect duplication more precisely. The technique takes a holistic approach, combining these elements to safely and successfully identify duplicates in encrypted documents. In the cloud, deduplication techniques are typically employed to minimize storage capacity. Maintaining ownership of the same data even after it has been uploaded multiple times is necessary to achieve high levels of security for encrypted data on the cloud. Our proposal is a client-side application for de-duplication called De-duplication software. With a graphical user interface, our suggested application may perform many functions such as file uploading, downloading, and registration. The program allows for the flexible support of encrypted data, giving data holders peace of mind regarding the security of their cloud-stored information. The outcomes of the computer simulation demonstrate the viability.

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.