https://doi.org/10.48047/AFJBS.6.7.2024.2119-2125

AFJBS African Journal of Biological Sciences
Journal homepage: http://www.afjbs.com
Research Paper Open Access

Data Deduplication using SHA algorithm in Cloud Environment

[1]VIJITHA B ,[2] Dr. B. SRINIVASA RAO

[1]PG Scholar Department of Computer Science and Engineering Teegala Krishna Reddy Engineering College
[1]vijithareddy3@gmail.com
[2]Professor Department of Computer Science and Engineering Teegala Krishna Reddy Engineering College
[2] deanacademics@tkrec.ac.in

Volume 6, Issue7, 2024
Received: 20Apr2024
Accepted:05Jun2024
DOI:10.48047/AFJBS.6.7.2119-2125

Abstract—

The cloud storage revolution is facing new hurdles in today's data - driven world. Our daily lives generate massive amounts of information – text, audio, videos – that fill up storage devices. To ensure data accessibility, cloud storage has become a popular choice. While service-oriented cloud storage offers cost-effective storage, a new challenge arises with the digital world. The proliferation of similar documents, like those in sharing and availability culture, leads to data duplication. This duplication inflates storage needs and backup costs. This paper proposes a system to optimize storage by identifying and filtering similar documents before uploading them to the cloud with the SHA algorithm.

Keywords—deduplication,SHA,Cloud

I. Introduction

In the digital era of common man , Now Data generation is a daily habituated scenario. the transactions are getting recorded for every small activity while using machines for communication. To hold the persistency and providing the availability for the user data, need to store the data using traditional techniques i.e. by taking the backup into individual disks. In the 20th century, cloud computing is the best resource to store the content into third party services as AWS S3, Google, Azure. From 2025 the data generation from a user's around 1T.B data. There are several options for the services to end-user as IAAS, SAAS and PAAS with different cloud architecture as public, private, hybrid and community clouds. In order to achieve the availability, user must maintain the multiple copies of the data in different resource. Even it is cost effective. data store in the cloud storages may contains the similar and near exactly copies or while taking the backup several times of same content. It may lead to produces the more number of copies in the memory. It increases the storage space and as well as Infrastructure cost.

There are two types of data is producing in very frequently as structured and unstructured data. In Structured data, text and file content will be saved into the disks. Unstructured or semi structured data consist multimedia content like images, objects, and videos. Rather than structured information, unstructured context will require more space with an abnormal cost effective while following the ACID properties to the data. Challenges will be increased if the data effected with duplication while taking the backup or providing the availability of the data with different geo-positions. In order to get the optimality into the business landscape for productive organizations, deduplication is emerging technique to enhance the data management by attaching the pre-build nugget frameworks to control the replicas into the cloud environment. Equipe the organizations with intuitive resolution to apply the strategy on controlling the data productivity without replication. It should be happen to improve the storage and backup capacity with the lower cost which leads to optimize the network bandwidth and overlays to produce the better data recovery for the assist viable business improvement relays on continuity approaches. It's an essential part for cloud vendors to provide the best services to end users.

❖ **Benefits of data deduplication:**

- Reduced Storage Costs: By eliminating duplicates, you can store more data on the same amount of physical storage, potentially lowering hardware expenses.
- Improved Backup Efficiency: Deduplication can significantly reduce the size of backups, as redundant data is not backed up repeatedly. This translates to faster backups and less storage needed for backups.
- Network Bandwidth Optimization: When data is deduplicated before transmission, only unique data needs to be transferred across the network, saving bandwidth and potentially improving transfer speeds.

In this paper, the proposed system is used to apply the deduplication on image. Here, the paper constructed with several parts, those are i) introduction to the need of the cloud computing and deduplication. II) literature survey will consist the recent algorithm and approaches of deduplications. III) Methodology, Proposed approach to follow the investigation on image deduplication using MD5, SHA-256 and 512. IV) Results, on several datasets to do the fact finding. V) Conclusions, several challenging prospectives on the bases of results. VI) References, Trusty Research scholar and their contributions.

II. LITARATURE SURVEY

The Traditional innovations or the frameworks had restricted capacity limit, could not deal with the tremendous datasets effectively, and could not store every one of the documents for huge timeframe thusly many organizations were missing to give the functionalities like execution, versatility and adaptability required in the large information. In the multi node or Raid level cluster application or big data, well suitable for handle the data with 4V (variety, volume, velocity and versatile). Deduplication is a process to handle and reduce the redundant data by eliminating same copies. In order to do, there are variety of context is generating in the internet as text, image and video. Content is going to be stored as in granularity, block, byte, text, image or video. File deduplication is the process of identifying and removing duplicate documents from a dataset. There are several techniques that can be used for file deduplication. It can be performed based on location, time and Granularity. Furthermore, Deduplication classified is shown in fig 1.

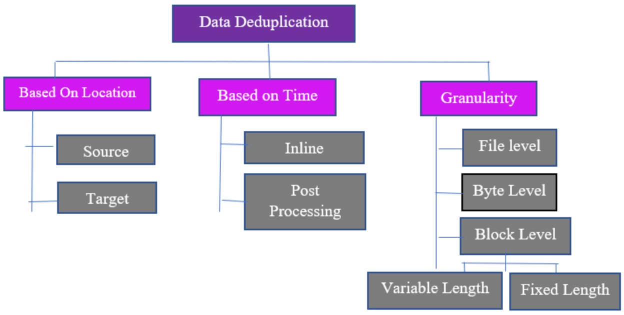


Fig 1.1 Classification of various Data Deduplication Techniques

Based on the location, Data Deduplication can be done at source side, reduction of the content can be happening at client location. After that, Unique context will be forwarded to store in the storage. The main disadvantages are when contrast with target-based deduplication, source is bit slower. Target Based dedup, data is forwarded to storage location, there after reduction of duplication is going to be happen at server side. It is a bit effective to improve the disk utilities. On other hand, another deduplication categorization is based on Time, here inline Processing is reducing the duplication as soon as input is processed. In post Processing deduplication, data is sent to the target side without modifying or trying to check whether forwarded data is duplicate or not. The whole process will be done after the storage. Most likely, it will choose at time of the scheduled backups. Third one is granularity level deduplications. It may proceed in different levels as block, byte or file level. File could be anything, it may be MS-excel or word or text file. The major advantage will be included here is storage space and avoid the network traffic. Explores the practical aspects of data deduplication, a technique used to eliminate duplicate copies of data. The paper focuses on evaluating the effectiveness of deduplication in real-world scenarios and addresses several key questions surrounding its implementation. An overview of deduplication techniques and the challenges they pose in terms of storage overhead, performance impact, and scalability. They then present their empirical analysis of deduplication effectiveness based on data collected from various systems, including storage clusters and data centers.

The deduplications mainly done with the TWO stages i.e **1) Chunking and Fingerprinting** – each data stream will be on the input data stream. Stream content is splits into chucks. Each chunk fingerprint generated using hash functions like MD5, SHA-1. **2) indexing & Writing :** Fingerprint value will be stored in the database to identify the unique copy. Identical chucks will be producing the same fingerprint then it will not be written to database. otherwise, it will be written to database.

In the present cloud environment, serverless application development is thrust are for storing the data either inline deduplication for accepting the business data. sandbox implementations are very high in real-time productions. It is also used to control the deduplication by following dedup agent which controls deduplication at different levels in RdbMS data base. There are serving deep learning methodologies into the market for identifying the text-based deduplications in the perspective of end-to-end applications. From the rdbms, need to pull the information into the expensive buffer management pool. Data independence, views, authorization . The deep learning models are generating to produce the significantly decrease the cost and functional performance leads to produce the real-world workloads as recommend systems, detection of credit card and AI conversions. The main and cost-effective approaches will give a addon as secure and efficiency for every application. Limit crossing approach is edge computing i.e. additional devices at client nodes may produce the computing services, communicational and storage costs. Server-side deduplications are may increase wastage of storage as twice. Instead of server-side deduplication, at every client end deduplication will be applied on specific device may leads to reduce the cost and storage space but it will increase the IO operations to retrieve the data from cryptographic hash functions like md4,md5,sha and whirlpool. The hash function which is used produce the unique key for every document. The key will be indexed in rdbms. It may lead to control the repeated files which consist the same content.

The major challenges which consist to developed applications.

1. Indexing the key and delay will not be tolerant to store the data. 2. Key length for the identification of deduplication.

3. Retrieval of the key for the verification 4. Storing the unique block into the respective cloud with a handshake of API as REST or SOAP which may help us to give the edge computing flexibilities for the client relation management system [CRM].

Here, in this research work, capable to generate application which supports the inline deduplication for files like text, log, doc, excel and system configurations files which are take as backup for several time. With the help of google API and MYSQL collaboration , control the duplicate document storage in the respective cloud environment by cryptographic algorithms. The observations and results entitled to conduct the study on data deduplication Techniques for Optimized Storage.

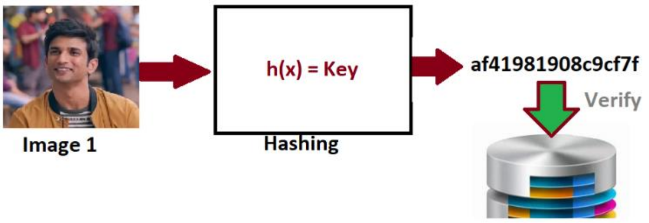
III. METHODOLOGY

In this process of data storage, Initially, data will be CRUD operations are performed by user into cloud. Each iteration brings the new challenges in the storage area. To optimize the approach, in proposed system, to restrict the duplication while uploading the document into the cloud need to follow the steps one by one.

Step 1: Identify the file dedup must be done at which end. Either it is inline or post processing dedup.

Step 2: Find an document to upload into the cloud resources. **Step 3:** Find the Key - Before uploading the document, need to find the key by applying the cryptographic hash functionalities. Each document is consisting its own content. Based upon the context, hash functionality will be applied on document to produce a unique. It must be varied from one to another.

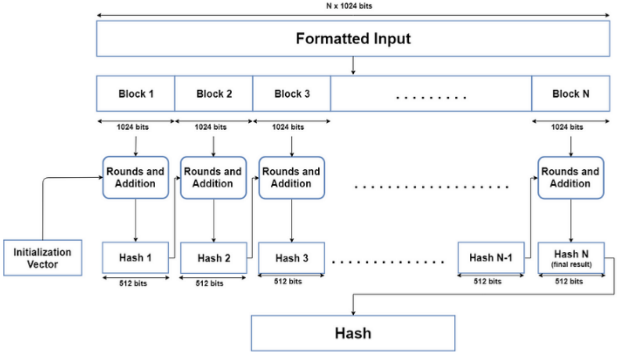
Step 4: This hash key will be generated based upon the block size with the different key length (MD5, SHA256, SHA512) Here, SHA will be computed on block size.



Symmetric and Asymmetric cryptographic algorithms are supports two way functions. Plain text will be converted into encryption and decryption to get the same plain text. Hash functions are one way method to hiding the encrypted format. From the encryption we could not bring the original data. Used for unique identifications. Even though hash functions do not encrypt messages, they are an integral part of cryptography because they play a crucial role in securing and authenticating data, which are key goals in cryptography. MD5 is no more used for hashing due to the limitations. SHA-128 bytes will produce the 32 bits length unique key. SHA-256 bytes- block generates the 64-bit length unique key. SHA512 and SHA1024 will be generated keys with different length.

SHA perform steps to generate the digest. Those are

1. **Input formatting :** the original message, padding bits, size of original message. And this should all have a combined size of a whole multiple of 1024 bits. This is because the formatted message will be processed as blocks of 1024 bits each, so each bock should have 1024 bits to work with. The bits may varys from one to another crypto format(2^n).
2. **Hash buffer initialization :** the default values used for starting off the chain processing of each 1024 bit block are also stored into the hash buffer at the start of processing. The actual value used is of little consequence, but for those interested, the values used are obtained by taking the first 64 bits of the fractional parts of the square roots of the first 8 prime numbers (2,3,5,7,11,13,17,19). These values are called the Initial Vectors (IV).



3. **Message Processing :** Message processing is done upon the formatted input by taking one block of 1024 bits at a time. The actual processing takes place by using two things: The 1024 bit block, and the result from the previous processing. This part of the SHA-512 algorithm consists of several 'Rounds' and an addition operation. This

part of the SHA-512 algorithm consists of several ‘Rounds’ and an addition operation.

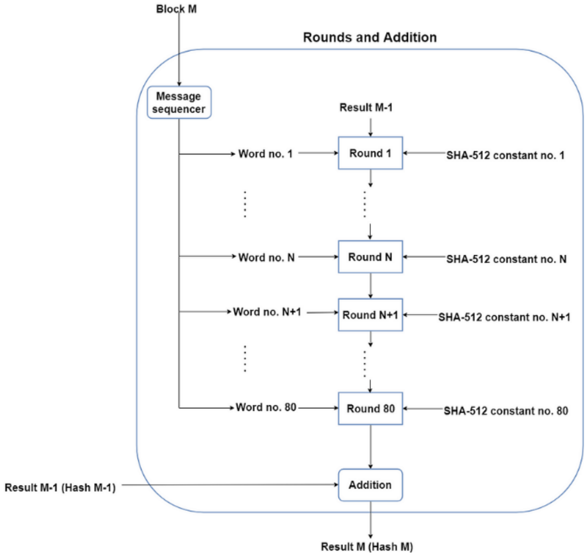


Fig SHA Digest generation cycle

4. **Output:** After every block of 1024 bits goes through the message processing phase, i.e. the last iteration of the phase, we get the final 512 bit Hash value of our original message. So, the intermediate results are all used from each block for processing the next block. And when the final 1024 bit block has finished being processed, we have with us the final result of the SHA-512 algorithm for our original message.

Step 5: Save the Each key in database for further use. Key is generated for the document will be saved into databases like SQL. If it is unique, it will be forwad the process to next step to store the specific document to cloud resource.

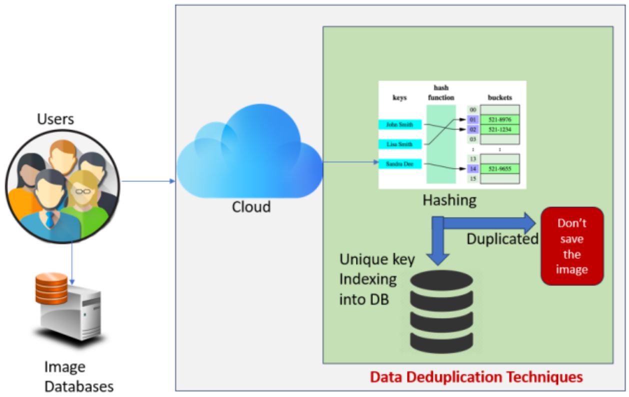


Fig3.1 File Deduplication Technique using algorithm

Here, Google Drive utilized as cloud resource to store the content by enabling the google drive api. It requires google accout (user name and password).

Algorithm for inline dedup ushing SHA with integration of relational database

INPUT ; Document D, The type of cryptography.

OUTPUT; Key Generation [K_G] and storing the block in cloud[s_{cc}].

Define ; L-length of the key, B-BLOCK form I to K iteration.

Create an empty hash table [H_T] (dedup index) to store data blocks.

```
1.  Set a block size $B_S$= N  (N = 4, 8 KB ...).
2.  Initialize the Document $D_A$ from user $U_I$
3.  $H_K$ = none;
4.  If $D_A$ ! = ∅ then
        While $D_I$ ! = $NULL$ receiving new data blocks:
                Read the incoming data block.
                Compute a hash value $H_V$ for the block.
                Repeat and return $H_K$ := $H_V$
5.  Connect = $D_b$ to Check if the hash value exists in the deduplication index:
6.  If $D_b$ not consist the $H_K$ then
            replace the data block with a pointer to the existing block.
            Repeat Step 5;
            Integrate to $G_{api}$ by enable the access to google API.
            Store the $D_A$ in $G_f$ – google storage.
        Else
            add the hash value and data block to the index.
```

OAuth Client Credentials: To authorize your application to access Google Drive resources on behalf of users, need to create OAuth client credentials in the Google Cloud Console. This process involves setting up scopes (permissions) for your application. **Drive UI Integration**: involves integrating with the Google Drive user interface (UI), configure this in the Google Cloud Console. This allows users to interact with application directly within Drive for specific use cases.

Step 6: If any document into excel files or database which further used to perform the insertion or searching the same key representations. To control the replicas, a new image will compute the hash values and before storing the key, it will check whether it is available in the database or not. Here use the databases like oracle, mysql to store the keys for identifyng the unique

IV. COMPARSION BETWEEN CERYPTOGRAPHIC ALGORITHMS

Comparison between MD5, SHA-2 and SHA-3:

Features	**MD5**	**SHA-2**	**SHA-3**
Available Since	1991	2002	2015
Developed By	Ronald Rivest	NIST	Guido Bertoni

SN O	Data set	Number of Instances (Size, no of instance)		MD5	SHA
1	DocFile	50.7MB	55	0.004	0.007
2	CSVFile	6.51GB	32	0.61	0.71
3	ConfigFile	473MB	3003	0.26	0.01
4	WebFiles	293MB	10534	0.92	0.76
5	TextFiles	5MB	3883	0.001	0.002

Block Size	512 bits	1024 bits	1152
Rounds	64	64	24
Collision Level	High	Low	Low
Security Level	Low	High	High
Applications	Data Encryption	TSL, Digital Certificates	Used to replace SHA2
Deprecated	Yes	No	No

Table 4.2. compare crypto-algorithms.

Comparison on running time based upon the file size:

File Size	Programming Language	Average Running Time (milli seconds)		
		MD5	SHA 256	SHA 3
314681 KB	Java	2129	3244	2577
	Python	702	496	868
421010KB	Java	2768	4389	3509
	Python	950	650	1.12
1070135 KB	Java	6361	10587	8216
	Python	5120	1590	2640

Table 4.2. time estimation for key generation for different sized documents with various programming language.

The study is majorly focused on the deduplication will be happening by using the md5, sha-256, sha 3 to produce a unique key as fingerprint or signature for each file to be identified at different locations. The main idea to check the observation, performance checks and CPU utilization will be done in the cloud environment for different sized documents. The study proven that huge collision in md5 compared with any other cryptographic algorithms. Md5 is not considered for dedulication approach any more. The family of cryptographic-algorithm have other options as sha with differnet key length

V. RESULTS

KEY GENERATION ;

Hash	Key generated – 'hello'	Len
MD5	eb61eead90e3b899c6bcbe27ac581660	32
WHIRLPOOL	0A25F55D7308ECA6B9567A7ED3BD1B46327F0F1FFDC804DD8BB5AF40E88D78B88DF0D002A89E2FDBD5876C523F1B67BC44E9F87047598E7548298EA1C81CFD73	128
SHA 256	3733CD977FF8EB18B987357E22CED99F46097F31ECB239E878AE63760E83E4D5	64
SHA 512	33DF2DCC31D35E7BC2568BEBF5D73A1E43A0E624B651BA5EF3157BBFB728446674A231B8B6E97FA1E570C3B1DE6D6C677541B262AC22AFDA5878FA2B591C7F08	128
SHA 3-512	8143d62fff557e3c37e15a7e7e1be8dd031401cd55da9ca74237e70f8c9d0f20bad8a2af7b22986e56ee6a704ee365f79f83fe7fbfe0c359d7caecc8030f6af5	128

Table

The key generation will produce a unique key for every document. The above table consist the key length and unique which is producing for the text –'hello'

Following are performance measures for evaluation of de-duplication [18]:

1. **De-duplication Ratio** [$\boldsymbol{DR_r}$] ; The effectiveness of data deduplication is assessed through its ratio, which quantifies the reduction achieved. This ratio is computed by dividing the total size of input data before deduplication by the total size of output data after deduplication.

   The mathematical expression is-

   $$DR_r = \frac{\text{total input data size before } DD}{total\ O/P\ \ data\ size\ after\ DD}$$

2. ***Throughput*** [$\boldsymbol{DD_T}$] **:** Throughput quantifies the capacity of a system to process units of information within a specific timeframe. In the context of data deduplication, throughput refers to the volume of data deduplicated within a defined period. and mainly measure in bit/sec

   $$DD_T = \frac{\text{total input size of the data}}{DD\ Time}$$

   DD Time = Chunking Time + Hash Generator Time

   $$TR_t = \frac{\text{troughtput}}{data\ size}$$

Transmission rate from the end device to transfer the document into cloud.

S. No.	Dataset item name	Item size - MB	Total size of the database - MB	DR_r	DD_T		
					S	F	K
1	Report.doc	8.83	40.7	21	21	12	13

2	Covid.csv	40.07	3076	1.3	50	9	16
3	Test.conf	29.06	472	6.1	31	34	10
4	0612.html	6.17	293	2.09	25	23	21
5	Caption.txt	3.16	500	0.6	62	53	59

Table V. 1 Observations of data deduplication in cloud.

S. No.	**Dataset item name**	**ddt**	**Transfer time**
1	Report.doc	0.42	0.47 sec
2	Covid.csv	0.81	0.02 sec
3	Test.conf	0.95	0.02 sec
4	0612.html	0.25	0.04 sec
5	Caption.txt	0.05	0.0008 sec

Table V. 2 Observations of data deduplication in cloud

Data deduplication ration used to represent percentage of data deduplication is happening at client side. The user performed the insertion of random file from the dataset to cloud environment. If the insertion or restriction may show the impact on the data size. If it is showing the change, what percentage of duplication may stopped at the client end. The performance metric is used to find the effectiveness. As per the table V.1 from the dataset documents have the highest percentage of duplication identified. There is concern of input file and its size.

Next, performance metric is throughput is used to find the data transmission rate from client to cloud. Here the observation may consist in different stages, Document is storing in the cloud from client. May be data inclusion is successful [S], here key generation[K] , key insertion into database and document loading into cloud will be considered to find the computational time for deduplication. if data inclusion is failure, must for every document key generation has to be done and unsuccessful insertion into Rdbms will be considered for the computation of time [F]. All the experiments are happened on a 8Gb RAM, SSD. The average throughput for the considered operation is 0.495 data per sec and data transformation from the device may consider as 0.11 sec to transfer the document into the cloud through the specific network. There is no much delay on the device.

The computational cost every experiment is consisting the similar variation in the time complexity. The cache may show impact on the results while retrieving the generated key from the database which should be considered to improve the i/o operation using the application with the constraint called deduplication.

VI. CONCLUSION

Deduplication is one of the trust technologies which improves both storage space and the amount of unique data being saved. In this paper, deduplication is classified based on inline data deduplications with the comparative cryptographic approaches with the process and techniques involved in deduplication are discussed. The worthful survey of the various techniques used on deduplication are compared in terms of certain evolution metrics on performance by considering parameters to observe the throughput, transfer and duplication ratio.

The challenges need to have cloud environment which is cost effective to store the content. Computational resource may impact the deduplication process with different kind of size files and byte based deduplications techniques are working absolutely producing the accuracy but the limitations of hashing will not able to identify the small changes which are happening inside of the file. Enhanced future work may keep an eye on edge computing based approach to share the computational resource by clubbing the different deduplication techniques to improve the performance, CPU utilization and client inclusion. Inine and outline meta data maintains will improve the computational reliability.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[3] Priyadharshini.P, Dhamodran.P, Kavitha.M.S “A Survey on De-Duplication in Cloud Computing” in IJCSMC vol.3, Issue 11 (November 2014), ||pp.149-155https://www.ijcsmc.com/docs/papers/November2014/V3I11201435.pdf

[4] https://data-flair.training/blogs/features-of-cloud-computing/

[5] https://docs.microsoft.com/en-us/windows-server/storage/data-deduplication/understand

[6] Rohini Sharma, “Data De-Duplication in Cloud Computing: A Review” in IJEAST vol. 2, Issue 2455-2143(February-March2017),||pp.26-29

[7] ttp://www.ijeast.com/papers/26-29,Tesma204,IJEAST.pdf

[8] Sachit-Ghimire,D.Venkata Subramanian “Chunking Algorithm for Data deduplication” in IJSRD Vol. 2, Issue(05,2014),||ISSN2321-0613 http://www.ijsrd.com/articles/IJSRDV2I5286.pdf

[9] Ider Lkhagvasuren1, Jung Min So1, Jeong Gun Lee1, Chuck Yoo2, Young Woong Ko1, “Byte-index Chunking Algorithm for Data Deduplication System”, in ijsia Vol. 7, No. 5(2013), ||pp.415-424 https://pdfs.semanticscholar.org/0880/325749067aefa23dce9bf63c4e92c6478773.pdf

[10] Tannu1, Karambir2, “Detection of De-Duplication Using SHA-512 and AES-256 in Cloud Storage” in IASIRhttp://iasir.net/AIJRSTEMpapers/AIJRSTEM17-323.pdf

[11] https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm

[12] https://pibytes.wordpress.com/2013/02/09/deduplication-internals-hash-based-part-2

[13] [1] Bhattacherjee, S., Narang, A., & Garg V. K. (2011). High throughput data redundancy removal algorithm with scalable performance. In HiPEAC’11 – Proceedings of the 6th international conference on high performance and embedded architecture and compilation (pp. 87–96).

[14] [2] Deduplication Internals, 2013 Deduplication Internals. (2013). Available from https://pibytes.wordpress.com/2013/02/02/deduplicationinternals.

[15] [3] Chen et al., 2015 Chen R, Mu Y, Yang G, Guo F. BL-MLE:Block-level messagelocked encryption for secure large file deduplication. IEEE Transactions on Information Forensics and Security.201510(12):2643–2652 https://doi.org/10.1109/tifs.2015.2470221.

[16] [4] Oltean and Sengupta, 2013) Oltean, A., & Sengupta, S. (2013). Data deduplication as platform for virtualization and high scale storage, Santa Clara.

[17] [5] Xia et al., 2016 Xia W, et al. A Comprehensive Study of the Past, Present, and Future of Data Deduplication. Proceedings of the IEEE. 2016;104(9):1681–1710.

[18] [6] J. Li, C. Qin, P. P. C. Lee, and X. Zhang, “Information leakage in encrypted deduplication via frequency analysis," in Proc. 47th Annu. IEEE/IFIP Int conf. Dependable Syst. Netw., Jun. 2017, pp. 1_12.

[19] [7] D. Harnik, B. Pinkas, and A. Shulman-Peleg, “Side channels in cloud services: Deduplication in cloud storage,"IEEE Security Privacy, vol. 8, no. 6, pp. 40_47, Nov./Dec. 2010.

[20] [8] R. SHOBANA, K. SHANTHA SHALINI, S. LEELAVATHY and V. SRIDEVI “De-Duplication of Data in Cloud” Int. J. Chem. Sci.: 14(4), 2016

[21] [9] J. Blasco, R. Di Pietro, A. Orfila, and A. Sorniotti, „„A tunable proof of ownership scheme for deduplication using bloom filters,““ in Proc. IEEE Conf. Commun. Netw. Secure. (CNS), Oct. 2014, pp. 481–489.

[22] [10] W. K. Ng, Y. Wen, and H. Zhu, „„Private data deduplication protocols in cloud storage,““ in Proc. 27th Annu. ACM Symp. Appl. Comput., 2012, pp. 441– 446.

[23] [11] J. Xu, E.-C. Chang, and J. Zhou, „„Weak leakage-resilient client-side de-duplication of encrypted data in cloud storage,““ in Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secure, 205, pp. 195–206.

[24] [12] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, “Secure data deduplication," in Proc. 4th ACM Int Workshop Storage Secure. Survivability, 2008, pp. 1_10.

[25] [13] Y. Shin and K. Kim, “Differentially private client-side data deduplication protocol for cloud storage services," Secure. Commun. Netw, vol. 8, no. 12, pp. 2114_2123, 2015.

[26] [14] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, “A hybrid cloud approach for secure authorized deduplication,"

[27] [15] IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206_1216, May 2015.