IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 1, January 2021

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

Secure Data Deduplication Using Efficient Cryptographic Method

Mrs.S.Ruba, Dr.A.M.Kalpana

Assistant Professor, Department Computer Science and Engineering, Government College of Engineering, Salem(D.t), India

Professor/Department of Computer Science and Engineering, Government College of Engineering, Salem(D.t), India

ABSTRACT: Cloud computing is a growing technology, which offers storage facilities in an online environment. Deduplication is the process of eliminating replicated data content from storage facilities like cloud datastore, online databases, cloud datastore, local file systems, etc. Deduplication in cloud technology is a very important issue to be addressed. It is a data pre-processing process to eliminate redundant data that requires unnecessary storage spaces and computing power. In this paper, a novel Q-learning approach is proposed to achieve a high deduplication ratio by removing duplicate data in online storage systems. The experimental results with different datasets demonstrate that the proposed system achieves a 13% higher deduplication ratio than a state-of-the-art deduplication algorithm.

KEYWORDS: Cloud computing, Cloud Storage, Chunking Algorithms, Data Deduplication, Q-Learning, Security.

I. INTRODUCTION

The cloud service provider is responsive to enlarging the data storage area and combining the data deduplication method into the cloud system, while data deduplication eliminates redundant data that is present in a cloud environment. Data privacy conservation is also a vital topic to be deliberated in deduplication. To support deduplication, a new kind of deduplication method is utilized for encrypting the data before outsourcing to a cloud environment [1]. The redundant data plays a vital role in increasing the cost of data storage [1]. Security of the cloud environment can be ensured by an authorized deduplication system which could perform the encryption and decryption along with its key for the given data using the specific algorithm and token-based access could maintain the security of the deduplication scheme [1]. Today's cloud service providers offer client different type of services such large storage space and parallel computing of resources at very cheap price. In the cloud system, data deduplication is a novel technique that works on quickly growing number of digital data in cloud data storage, these techniques used for identifying redundant data. The outcome of unique single copy is kept and it can be served to all or any of the authorized clients.

For duplicate checking, user requests for duplicate check of selected file to administrator. A token which are sends towards administrator is in encrypted format, this encryption done with client's password as encryption key. At administrator side, administrator decrypted this token by using client's password as decryption key. Once decryption is completed, the token administrator checks this token in the database. At time of file uploading, file is get encrypted with token as encrypted keys and then uploaded this encrypted file on cloud. So, the files are stored in encrypted format in public cloud.The general system of data deduplication is illustrated in Figure 1.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

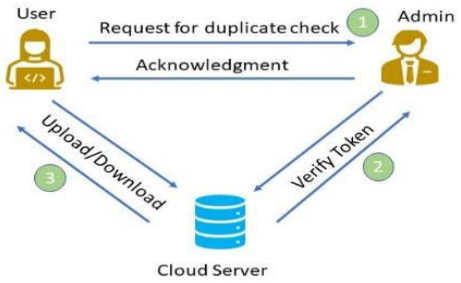

Figure 1 General architecture of Data Deduplication

Data deduplications have significant role to reduce the storage cost in cloud storage management system. Nowadays, many secure deduplication encryption systems have been implemented to secure the privacy of clients' information. Available systems cannot maintain a cost-effective cloud service to achieve the data system with a best duplication structure [2]. At this point, it will create a main challenge for the cloud users to be acquiring the deduplication scheme. This inattentive behaviour has even led to serious scandals. To attain the authorized deduplication, satisfy dynamic privilege updating and revoking and avoid data leakage, the proposed system implements a novel secure role encryption and decryption method with secured data deduplication in cloud environment. The experiment proves that the proposed system can efficiently moderate the response time and succeed the storage load of files more composed.

The rest of this paper is systematized as follows: In section 2, review of the previous related work done for data deduplication. In section 3, discussed Data Deduplication Design Criteria, Section 4 define proposed scheme of deduplication system, Section 5 described Implementation details, and section 6 illustrates result & discussion and various evaluation parameters.

II. RELATED WORKS

Halevi et al. [3] used an idea which is Hashing function SHA256 implemented for creating hash value for building Merkle tree for Proof of Ownership. The author proposed the Proof of ownership that deals a client proficiently influence a server which consume licensed one. During this system they kept just some kind of information about file instead of complete file. This is often done by using Merkle trees with certain encodings to examine their security level.

Hou et al. [4] find the approach to recognize cloud storage auditing associate with deduplication management for various security levels affording to data popularity. To obtain the optimal solution, this author elucidates the one key issue to ensure that the third-party auditor who still can audit the integrity of cloud data after data popularity modifications. At that point, they suggested the main cloud storage auditing system with deduplication supporting various safety levels. This methodology implements convergent threshold cryptosystem to offer semantic safety for detested data also cipher text deduplication for common data. Clients are don't want to create new authenticators by himself as well as be online while the data popularity modifications.

Zhou et al. [5] developed EDedup system, which utilize a similarity aware encrypted deduplication method to accomplish the server based MLE at segment level for scale back computation ingestion. To prevent privacy leakage, the EDedup system integrates source based related segment recognition and target based duplicate chunk checking. This system also creates arbitrary message derived file keys to manage the metadata also understands access control with revocation through accepting proxy-based CP-ABE (Ciphertext Policy Attribute-Based Encryption) to encrypt file keys.

Zheng et al. [6] designed a deduplication system on cloud data to support the certificate less proxy re-encryption. It contains Certificate Less Proxy Re-Encryption (CL-PRE) and proof of ownership supported certificate less signature (PoWCLS). They use certificate less cryptography to unravel the matter of key escrow and avoid things where a key generation centre (KGC) impersonates a user to decrypt the cipher text. The certificate less cryptography method was used by this author to undo the substance of key guarantee and avoid things where a KGC (Key Generation Centre) imitates a client to decrypt the cipher text.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

Xia et al., [7] utilized 'P-Dedupe' with parallelized data deduplication method, which speed up deduplication method through dividing the deduplication method into 4 phases such as chunking, fingerprinting, indexing, and writing. Pipelining these 4 phases with chunks and files then parallelizing 'CDC' (Content Defined Chunking) also secure hash based fingerprinting phases to more ease the calculation block. This system will first split the information stream into numerous segments that is called 'Map', in this point every segment running 'CDC' in parallel with a liberated thread then it rechunk as well as combine the boundaries of those segments that is called as 'Reduce' to validate the chunking efficiency of parallelized 'CDC'.

Yuan et al. [8] proposed an accessible with safe system for data deduplication through active client controlling, which modifies active cluster clients during a safe approach and confines the illegal clients of the profound facts influenced by using effective clients. Shanshan Li [10] propose a safe also well-organized 'CSED' (Client-Side Encrypted Data Deduplication) system sustained PoW and incorporate it into 'CSED' to repel illicit data circulation attacks. A fanatical key server is presented in producing 'MLE' (Message Locked Encryption) keys to repel 'brute force attacks' in CSED. Furthermore, a hierarchical storage planning is utilized to improve the 'I/O' effectiveness on the cloud server. They acquire the key server to aid the users in making the 'MLE' keys also adopt the rate limiting approach to break brute force attacks.

N. Kumar and S. Jain [9] in this paper, we suggest Differential Evolution (DE) rely on technique which is optimized Two Thresholds Two Divisors (TTTD-P) Content Defined Chunking (CDC) to decrease the number of computing process utilizing single dynamic optimal parameter divisor D with optimal threshold value exploiting multi-process nature of TTTD. The TTTD technique provides an additional backup divisor D′ that has a higher probability of discovering cut points to reduce chunk size variance; nevertheless, introducing a second divisor reduces chunking throughput. To this purpose, Asymmetric Extremum (AE) considerably enhances chunking throughput through overcoming Rabin and TTTD boundary shift problems by using local extreme values in a variable-sized asymmetric window, while maintaining a near-identical deduplication ratio (DR). On Hadoop Distributed File System, we suggest DE-rely on TTTD-P optimized chunking to maximize chunking throughput while increasing DR; The use of a scalable bucket indexing strategy minimizes the time it takes to find and declare duplicated hash values (HV). chunks about 16 times greater than Rabin CDC, 5 times greater than AE CDC, and 1.6 times greater than FAST CDC (HDFS).

S. Rubaet al., [10] in this research, suggested a dynamic data deduplication (DD) strategy for cloud storage, in arrange to strike a balance among changing storage efficacy and criteria for fault tolerance, as well as to increase cloud storage performance. We adjust the number of copies of files in real time to match the changing degree of QoS. Show results of the experiments reveal that our suggested scheme works effectively and can deal with scalability issues. We also intend to keep track of how users' file demands change. We also intend to assess the system's availability and performance.

Y. Fan et al., [11] in this research article, every cloud user is given a set of privileges in our system, and deduplication can only be done if and only if the cloud users have the appropriate privileges. Furthermore, our system improves the capacity of like cryptosystems to resist selected plaintext and selection ciphertext attacks by augmenting convergent encryption with users' privileges and relying on TEE to provide secure key management. Our system is secure sufficient to facilitate data deduplication (DD) as well as protecting the privacy of sensitive data, according to a security analysis. Moreover, we create a prototype of our system and analyse its performance. Experiments reveal that our system overhead is practical in real-world scenarios.

M. Oh et al., [12] in this paper, suggest a novel deduplication technique that is extremely compatible and scalable with the exhausted storage currently in use. Our deduplication technique, in particular, uses a double hashing algorithm (HA) that takes advantage of hashes employed through the implicit scale-out storage, which overcomes the limitations of present fingerprint hashing (FH). Furthermore, our approach combines file system and deduplication meta-information into a single object, and it manages the deduplication ratio online through initial aware of post-processing-related scheme demands. On open-source scaleout storage, we implemented the proposed deduplication approach. When executing a variety of standard storage workloads, the experimental findings illustrate that our solution could save greater than 90% of total storage space while providing the same or similar performance as traditional scale-out storage.

Yoosuf,et al., [13] in this article, suggested method uses an access control mechanism on data belonging to privileged authorities to safeguard the data deduplication (DD) process. The data that will be outsourced will be encrypted files. The secure authorities are given access control mechanisms to do data deduplication (DD) on the data that was outsourced. Encryption techniques are used in the Access Control Mechanism. It employs convergent randomized encryption and a reliable distribution of owning party keys to allow the cloud service provider to manage outsourced

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

data access even when control shifts on a regular basis. This prohibits data from being leakage not only to individuals who have had their rights to the data revoked, however also to a degree to a trustworthy but dubious cloud storage server. In addition, the suggested technique safeguards data integrity against attacks relies on label discrepancies. As a precaution, the suggested technique has been changed to improve security. The suggested scheme is virtually as effective as the current ones, with only a minor raise in computational expenses, according to the efficiency study.

R. Kirubakaran et al., [14] in this article, present a cloud-based technique for achieving deduplication of a huge amount of data available. The approach includes data deduplication before uploading to cloud storage as well as data reverse deduplication when obtaining the required data. Any of the algorithms indicated in the literature survey can be used to implement the suggested technique. The model is more effective and accurate than existing deduplication systems because of the type of algorithm utilized.

Kumar et al., [15] in this paper, proposed data deduplication technique, the main goal of this technique is to delete reiterate data from the cloud. It can also aid in the reduction of bandwidth and storage space usage. This suggested technique is to delete reiterate data; however, each user has their own unique token and has been allocated various privileges based on the duplication check. The hybrid cloud architecture is used to achieve cloud deduplication. The proposed technique is more secure and uses fewer cloud resources. It was also demonstrated that, when compared to the standard Deduplication technique, the proposed system had a low overhead in duplicate removal. On this work, both content level and file level deduplication of file data is examined in the cloud.

Zheng, Xiaoyuet al.,[16] in this research article, suggest data duplication in clouds, which is managed using the deduplication technique. Although some deduplication techniques are used to prevent data redundancy, they are inefficient. The major goal of this research is to gain enough knowledge and a decent concept of deduplication techniques through reviewing existent ways, and this work may aid future research in establishing effective cloud storage management (CSM) solutions for researchers and practitioners.

Zhang, Yuan, et al.,[17] in this article, focus on non-centre cloud storage data deduplication and present a new two-side data deduplication (DD) mechanism. Duplicate data is first recognized and stopped from being uploaded to the cloud data centre on the client side, and then new duplicate data is detection and erased on the cloud server side. The Chord algorithm (CA) is optimized using a compressed finger table and search optimization of the finger table, and it is then utilized to build a logical network that makes searching for fingerprints in storage nodes considerably faster. Show the results illustrate that the enhanced Chord algorithm outperforms the original in terms of the number of fingerprint values and the cost of searching time, and the suggested two-side data deduplication (DD) technique outperforms the traditional data deduplication (DD) mechanism in terms of deduplication rate.

X. Xu and Q. Tu [18] in this work suggest a deduplication scheme architecture for cloud storage environments (CSE), as well as the procedure for averting duplication on the client side at the file and chunk levels. DelayDedupe, a delayed target deduplication strategy rely on chunk-level deduplication and chunk access frequency, is suggested to decrease response time in storage nodes (Snodes). When used in conjunction with replica arrangement, this technique evaluates whether fresh multiplied chunks for data update are hot and, if they aren't, eliminates the hot duplicated chucks. The findings of the experiment show that the DelayDedupe method may successfully minimize response time while also balancing the storage demand on Snodes.

III. PROPOSED METHODOLOGY

In existing work researchers have given good contribution in making deduplication system by utilizing standard encryption methodology and trusted execution environment to provide secure key management.According to the recognized necessities in positions of security possessions plus system overhead in the earlier work, we examine advanced secure deduplication solutions that can be include the below lines:

1. The input File is encrypted with its respecting Hash Key used for Convergent Encryption and been encrypted using AES algorithm as a whole.

2. The File is also split into four and encrypted using their own hash function keys.

3. The files are hashed using different hash Function because if the hacker wants as found the hashing

function of a particular split and try opens the remaining splits and joins to get the information of the

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

file.

4. In this Proposed system, it is not possible as four splits in the four different server uses different hash function and keys and encryption by the keys also vary so the security is being very much advanced secure which is explained in the given figure (2).
5. The Duplication is checked as chunks in all 5 servers with respect of its hashing techniques.

a) Login

User-first login communication is done with key based process. It is a type of authentication that may be utilized as a substitute to password authentication. As a replacement for requiring a user's password, it is probable to authorize the client's uniqueness by asymmetric cryptography method, with public and privatekeys. After successful completion of this process client can able to get verification code from code database then client verify that code authentication from code verification database. Then, it decrypts and store the data into file server. When it comes to this process, it is always exchange of data between client and server or passing of data through the network. And such passing of data can be dangerous if the data to be exchanged is not encrypted. So, here we used the robust method to secure the data. In this section, AES and RSA algorithms are utilized to generate the secured keys for login process.

When a user attempts to log in to the system for the first time, a common password is required. This password is logged as an input attempt along with the IP address as soon as the password is verified. If the user immediately enters the passphrase, a passphrase is required. The machine will check the IP address only if it matches the previous one, and only if you do this authentication, it will check the password entered by the user. Users grant access to the cloud path. The validation was carried out when the user inputs the password, which is encrypted and the key was produced using both encryption standards. With the key saved in the database the generated key is checked. Before the key is sent to the API the original password is disguised fully by encryption standards.

b) File Uploading

A registered user enters login id and password, after successful attempt user is ready to upload file. After completion of login process (As we mentioned in the above login section: Step 1 to Step 9) user get the login access of file server, the below steps involved in file uploading process,

Step 1: User sends request to server to upload the file.

Step 2: Check the user log for identifying whether both the login IP and access was success on bases of attempts

Step 3: Request for the deduplication check

Step 4: Response of the deduplication check is sent to the user from the server.

Step 5: If no duplicate found after comparison in server, then that file is ready to upload into the file log server

Step 6: The uploaded file is split into four segments and the files are stored after encryption is completed

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

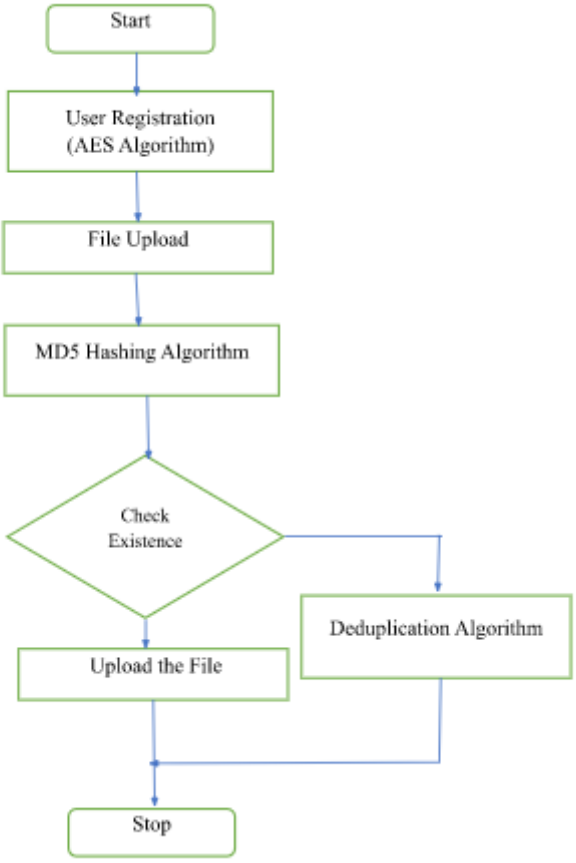

Figure 2: General flowchart for Data Deduplication

The document which is been uploaded is first stored as a whole in the one server and the copy of the document is been split into four parts and is been stored in the four different servers. Now the document is converted into hash not using the same hash function four different hashing key for four different servers and the duplication is checked according to the hash function. They are listed below,

- Server 1 -MD5
- Server 2 - SHA-1
- Server 3 - SHA-256
- Server 4 - SHA-384

c) Request for file Download

In this module, client downloads the files by applying below steps which are illustrated in Figure 5. A registered user enters login id and password, after successful attempt user is ready to download the file. After completion of loginuser get the login access of file server, the below steps involved in file downloading process,

Step 1: User sends request to server to download the file

Step 2: Admin can check the identity of the client

Step 3: Check the user log for identifying whether both the login IP and access was success on bases attempts

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

Step 4: Response is received from the log server whether the attempt is success or failure

Step 5: if that attempt is success, then the client gets grant access to file with a time constraint even after

download

d) Request for own file download:

To download their own file from cloud server, user need to follow the below steps. Since they also need to complete the above-mentioned loginprocess.

Step 1: User sends request to server to download the file

Step 2: Admin can check the identity of the client

Step 3: Check the user log for identifying whether both the login IP and access was success on basesattempts

Step 4: Response is received from the log server whether the attempt is success or failure

Step 5: If that attempt is success, then the client gets grant access to file with a time constraint to the stored file log server

Step 6: Finds the split files according to the log files

Step 7: File is merged and access is granted to file with a time constraint even after download

Step 8: File path is linked to the requested file server

Step 9: Finally, the required file is downloaded

IV. RESULTS

In this section, we evaluate the proposed deduplication system with encoding and decoding process using various key generation algorithms. All our experiments were performed on an Intel Xeon E5530 (2.40 GHz) server with windows-10 OS. The data is collected from an example login page, where many people are requested to login using their own password. Those passwords were implemented to produce the encrypted key in the algorithm thereafter with time indicated to finish the full procedure.

1.1. Initial Password- Dataset

Table 1 illustrates the sample data (Initial password) with the AES and RSA parameters. We also calculate the Dec_Time (NANO SEC) and Total Execution time (SEC) for key generation process.

Table 1. Initial password generation factors

S.NO	UP	UP_KL	UP_KS	AES Key _Len	AES Key SIZE	RSA Key length	RSA Key Size	RSA KEY Generating Time (NANO SEC)	Dec_Time (NANO SEC)	Total Execution time (SEC)
1	Aenkdfn@	9	56	44	113	15	96	0.198	0.062	1.236
2	a5dfE5dg@	11	75	40	120	16	116	0.025	0.024	0.862
3	gg545hGH56	12	96	37	146	17	198	0.0269	0.365	0.123
4	DDFFffgg@@3	13	82	35	189	18	182	0.459	0.548	0.357
5	2D2g3D5h#$1	15	43	40	166	14	177	0.898	0.699	0.951

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

1.2. Phrase Password

Table 2 illustrates the sample data (Phrase password) with the AES and RSA parameters. We also calculate the Dec_Time(NANO SEC) and Total Execution time(SEC) for key generation process.

Table 2. Phrase password generation factors

S.NO	UP	UP_KL	UP_KS	Base64 Key_Len	Base64 Key_ SIZE	RSA Key length	RSA Key Size	RSA KEY Generating Time (NANO SEC)	Dec_Time (NANO SEC)	Total Execution time (SEC)
1	Hi buddy i am gone hack you	21	96	40	120	46	128	0.014	0.062	0.00208
2	your are always My hero	23	82	40	120	44	128	0.022	0.032	0.00234
3	this is my game and u cannot play	25	76	40	120	46	128	0.025	0.072	0.00066
4	Kill the robber	27	92	40	120	44	128	0.026	0.064	0.00456
5	My first money	31	88	40	120	46	128	0.029	0.055	0.00123

1.3. Login

Table 3 shows the initial, Phrase, and total login time in seconds.

Table 3. Login time in seconds

S.NO	Ini_Login (SEC)	Phrase _login (SEC)	Login Time (SEC)
1	1.23	0.00322	1.55026
2	0.234	0.00075	0.14585
3	0.625	0.00094	0.24586
4	0.842	0.00096	0.32547
5	0.776	0.00085	0.25896

1.4. File Upload

The file that has been uploaded is first stored as a whole in the one server and the copy of the document has been split into four parts and is been stored in the four different servers. Due to security concerns, each server utilizing various algorithms (MD5, SHA-1, SHA-256 and SHA-384) to generate the key and encrypt the files (Copy 1 to Copy 4). Table

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

4. demonstrates the size of the files also key generation and encryption process timings. Finally, we calculate the file upload timing process. Figure 7 shows the uploading time for various file size.

Table 4. File size, key generation and encryption process timing

File Type	Total size of file	Size of split 1	Copy 1 key Gen_Time (SEC)	Encryption Time1 (Nano Sec)	Size of split 2	Copy 2 Gen_time (SEC)	Encryption Time2 (NANO Sec)
Text file 1	155kbs	20kbs	0.03063	0.00476	20kbs	0.00025	0.00754
Text file 2	240Kbs	26kbs	0.08963	0.01490	26kbs	0.01478	0.00589
Music 1	5.2 mbs	20,142kbs	0.12586	0.15963	20,142kbs	0.16391	0.28456
Music 2	4.5mbs	20,142kbs	0.12473	0.24586	20,142kbs	0.17852	0.23652
Image 1	53,054Kbs	19,101kbs	0.01478	0.24589	19,101kbs	0.15882	0.12457
Image2	62,004 kbs	16,333kbs	0.01258	0.24583	16,333kbs	0.24580	0.23568
Video 1	4.25 mbs	59,232kbs	0.58965	0.54892	59,232kbs	0.45892	0.52486
Video2	6.59 mbs	47,124kbs	0.54782	0.64896	47,124kbs	0.78524	0.62485
Size of file 3	Copy 3 Gen_Time (SEC)	Encryption Time3 (NANO SEC)	Size of file 4	Copy 4 Gen_Time (SEC)	Encryption Time4 (Nan0 SEC)	Upload time (SEC)	
20kbs	0.00042	0.00334	20kbs	0.01342	0.00382	0.34543	
26kbs	0.00024	0.00148	26kbs	0.02477	0.01244	0.32510	
20,142kbs	0.25860	0.36201	20,142kbs	0.32560	0.32541	0.5622	
20,142kbs	0.21301	0.41025	20,142kbs	0.24581	0.36566	0.4211	
19,101kbs	0.14201	0.15620	19,101kbs	0.11254	0.12450	0.4582	
16,333kbs	0.12302	0.16235	16,333kbs	0.12489	0.14120	1.9802	
59,232kbs	0.64890	0.68740	59,232kbs	0.69852	0.78520	1.5980	
47,124kbs	0.75410	0.70101	47,124kbs	0.78750	0.74444	1.5555	

1.5. File Download

As we discussed in previous section, user sends a download call to cloud service supplier to acquire the file from log server. Utilize the file id in log server to acquire every encrypted chunk from the cloud storage supplier and decrypt every chunk by using the appropriate key. Finally, reconstruct the original file. Table 5 illustrates the decryption time in nano-seconds for each chunk, total file merge time in seconds and size of the downloaded file. Figure 8 shows the downloading time for various file size.

Table 5. Decryption process timing

S.NO	File Type	Decryption Time 1 (NANO SEC)	Decryption Time 2 (NANO SEC)	Decryption Time 3 (NANO SEC)	Decryption Time 4 (NANO SEC)	Total Merge Time (SEC)	File Size (Kbs)
1.	Text file 1	0.01523	0.0025	0.0056	0.0097	0.02458	155kbs
2.	Text file 2	0.01245	0.0069	0.0044	0.0086	0.04975	240Kbs
3.	Music 1	0.34589	0.2450	0.3250	0.3458	1.7068	5.220mbs
4.	Music 2	0.39548	0.2013	0.3980	0.3450	1.5592	4.500mbs
5.	Image 1	0.24586	0.1240	0.1904	0.2451	0.57324	53,054Kbs
6.	Image2	0.20140	0.1901	0.1402	0.1547	0.68921	62,004 kbs

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

7.	Video 1	0.56980	0.7890	0.7364	0.6988	3.70119	4.250 mbs
8.	Video2	0.54012	0.7923	0.7124	0.6452	3.20014	6.529 mbs

Examining the experimental results in graphs and tables, it shows that the encoding/decoding processing time. Exactly, at the file upload, the encoding time of 117kbs is 0.34543 and 2.45mbs is 1.54989, and is less than that of encrypting a data chunk in existing system result. At the file download, the decoding time is smaller than the time of encrypting a data chunk. The system can pipeline both the encoding as well as decryption modules, creating the decryption portion taking less time. This proposed system has also verified the cases with other file sizes (like 156Kbs, 1.04mbs, 1.06mbs, 61,054Kbs, 61,054Kbs and 2.45mbs) and made similar observations. With the size of file raising, the computational price of creating keys, symmetric encryption and decryption has expressively raised.

The proposed algorithm is implemented using NetBeans IDE in Java. The hardware configuration used is Intel(R) Core(TM) i5-processor with 8GB RAM. In this application, a user selects a file for upload and hash value of the file is generated. The text area shows the list of text files with their hash values already present in the cloud environment. As the user selects to upload, the hash value of the selected file is compared to the hash value of the files present on the cloud environment. If no match found, then the upload of file will be successful as shown in Fig. 2. If the match is found, the file upload will be unsuccessful as shown in Fig. 3.

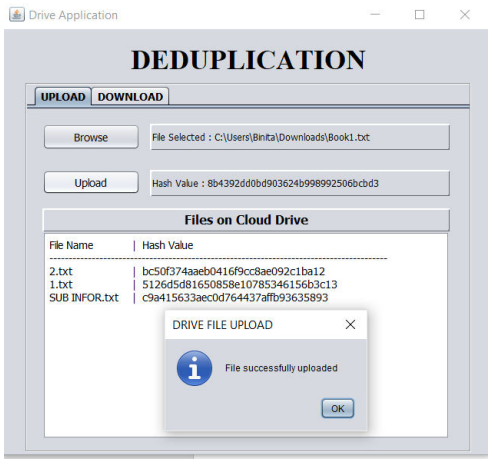

Figure 3 Implementation of Proposed Deduplication Strategy for Successful File Upload

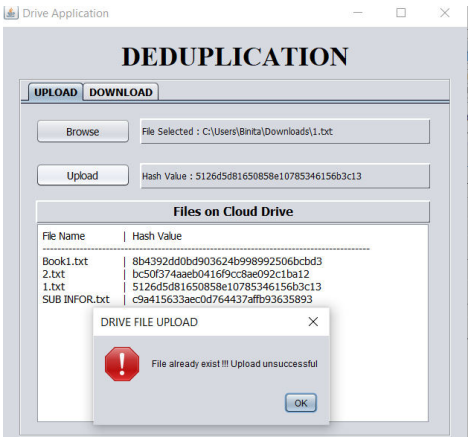

Figure 4 Implementation of Proposed

Deduplication Strategy for Unsuccessful File Upload

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

1.6. Evaluation Parameters

In proposed deduplication system, we assess the overhead by changing various elements such data/file size, amount of stored file and deduplication rate etc. factors which are mentions are given below.

- **File Size:** This is factor effects on time required on processing of file on authorized deduplication system. The time required on encryption, upload increase with respect to increase in file size, but the other operation such as token generation and duplicates check time remain constant throughout.
- **Deduplication Ratio:** The deduplication ratio is described as percent of storage space that has save by using this proposed deduplication system.

Table 5. Evaluation Metric

Algorithm	Number of Chunks	Deduplication	Time in Seconds
LIPA	623961	0.67921	4526
MD-5	154289	0.84523	2314
Bloom Filter	644933	2.02586	4962
SHA-1	960061	2.01285	5601
Proposed Algorithm	895233	2.59870	8001

V.CONCLUSION

Cloud storage services deals on request virtualized storage resources also clients only pay for the area they essentially spent. With the growing demand and data storage in the cloud, data deduplication is one of the systems utilized to advance storage proficiency. Data deduplication is a focused data compressionmethod for removing duplicate copies of data in storage. This paper proposes secure deduplication with the aid of multiple key generation mechanisms. This data deduplication method contributes a lot of advantages, along with security as well as privacy concerns are also become solve. As the client's profound data are safe from both insiders also outsider of the attacks. The stored file content in cloud storage should not be exposed to anybody excluding the client who owns the data.

In proposed deduplication systems, only a valid client/user who owns the original data (communication is done with key based process) should be confirmed as an authentic owner cloud storage.The quantity of ciphertexts generally increases fast in cloud storage. So, we must set aside enough types of ciphertext for future extension. If not, we require the public key to extend. We cannot compromise both on security and duplication of data across storage locations in this information-dense environment. A plan has to be developed to improve storage optimization and to provide deduplication technology to data storage servers where the data provided is encrypted without negotiating encryptions.

While data deduplication has been the subject of several studies in the last decade, some challenges remain, especially in the Big Data Era. In this article, we have reviewed the most recent big data deduplication frameworks suggested in the literature. We also proposed a novel end-to-end big data deduplication framework based on a Semi-supervised clustering approach. The experiments have shown that the framework outperforms the existing big data deduplication approaches with an F-score of 98,21%. The suggested framework is also extended with an online continual learning phase that continuously improves the deduplication model performance and increases the model accuracy by 1.16%. In future work, we aim to enhance our framework by reducing the error rate when used on a very-poor quality dataset. Also, we aim to extend our framework to address more quality dimensions.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com |

|| Volume 10, Issue 1, January 2021 ||

DOI:10.15680/IJIRSET.2021.1001098

REFERENCES

[1]Ruba, Mrs S., and A. M. Kalpana. "Deep Learning Distributed a Cloud Storage Systems for Secure Data Uploading with Chunk-Based Deduplication." Solid State Technology 63.5 (2020): 8606-8620.
[2] K. Bilal, E. Baccour, A. Erbad, A. Mohamed, and M. Guizani, "Collaborative joint caching and transcoding in mobile edge networks," Journal of Network and Computer Applications, 2019.
[3]Halevi et al. "Secure Data Deduplication System with Efficient and Reliable Multi-Key Management in Cloud Storage." Journal of Internet Technology 23.4 (2020): 811-825.
[4]Hou, Huiying, Jia Yu, and Rong Hao. "Cloud storage auditing with deduplication supporting different security levels according to data popularity." Journal of Network and Computer Applications 134 (2019): 26-39.
[5]Zhou, Yukun, et al. "A similarity-aware encrypted deduplication scheme with flexible access control in the cloud." Future Generation Computer Systems 84 (2018): 177-189.
[6]Zheng, Xiaoyu, et al. "A cloud data deduplication scheme based on certificateless proxy re-encryption." Journal of Systems Architecture 102 (2020): 101666.
[7]Xia, Wen, et al. "Accelerating content-defined-chunking based data deduplication by exploiting parallelism." Future Generation Computer Systems 98 (2019): 406-418.
[8]Yuan, Haoran, et al. "Secure and scalable data deduplication with dynamic user management." Information Sciences 456 (2018): 159-173.
[9]Kumar, Naresh, Shobha Antwal, and S. C. Jain. "Differential Evolution based bucket indexed data deduplication for big data storage." Journal of Intelligent & Fuzzy Systems 34.1 (2018): 491-505.
[10]S. Ruba and A. M. Kalpana "Survey on Content based Chunking in Data Deduplication on Cloud Storage Environment" CiiT International Journal of Data Mining and Knowledge Engineering, Vol 10, No 10, October - November 2018.
[11]Fan, Yongkai, et al. "A secure privacy preserving deduplication scheme for cloud computing." Future Generation Computer Systems 101 (2019): 127-135.
[12]Oh, Myoungwon, et al. "Design of global data deduplication for a scale-out distributed storage system." 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018.
[13]Yoosuf, Mohamed Sirajudeen, and R. Anitha. "Low latency fog-centric deduplication approach to reduce IoT healthcare data redundancy." Wireless Personal Communications 126.1 (2022): 421-443.
[14]Kirubakaran, R., C. Mano Prathibhan, and C. Karthika. "A cloud based model for deduplication of large data." 2015 IEEE international conference on engineering and technology (ICETECH). IEEE, 2015.
[15]Kumar, PriyanMalarvizhi, et al. "A study on data de-duplication schemes in cloud storage." International Journal of Grid and Utility Computing 11.4 (2020): 509-516.
[16]Zheng, Xiaoyu, et al. "A cloud data deduplication scheme based on certificateless proxy re-encryption." Journal of Systems Architecture 102 (2020): 101666.
[17]Zhang, Yuan, et al. "Secure encrypted data deduplication for cloud storage against compromised key servers." 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019.
[18]Xu, Xiaolong, Nan Hu, and Qun Tu. "Two-side data deduplication mechanism for non-center cloud storage systems." 2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB). IEEE, 2016.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details