Data Storage and Retrieval with Deduplication in Secured Cloud Storage

S. Muthurajkumar

Abstract*: The Cloud Storage can be depicted as a service model where raw or processed data is stored, handled, and backed-up remotely while accessible to multiple users simultaneously over a network. Few of the ideal features of cloud storage is reliability, easy deployment, disaster recovery, security for data, accessibility and on top of that lesser overall storage costs which removes the hindrance of purchasing and maintaining the technologies for cloud storage. In this modern technology world, massive amount of data are produced in day to day life. So, it has become necessary to handle those big data on demand which is a challenging task for current data storage systems. The process of eliminating redundant copies of data thereby reducing the storage overhead is termed as Data Deduplication (DD). One of the ultimate aim of this research is to achieve ideal deduplication on secured data of client side. On the other hand as the client's data are encrypted with different keys, the cross user deduplication is merely impossible as having a single key encryption among multiple user's leads to an in secure system resulting in fragile to client's expectations. The proposed research adapts Message Locked Encryption (MLE) technique that looks for redundant files in cloud before uploading the client's file which eventually reduces the storage. Since the redundant files are swept, the network bandwidth is considerably reduced with respect to the redundant contents uploaded several times.

***Index Terms*: Cloud Computing, Deduplication, Data Storage, Encryption, Security.**

I. INTRODUCTION

A growing computer paradigm for data and services to function in large and scalable data centres in the cloud and accessed from devices that are connected over the internet. Cloud computing comes into picture when the cloud is used for utilities.

The cloud computing offers services over the internet. Internet is used to provide services of storing data, computation performing and other tasks. For an organization a lot of its processes can be automated over the cloud for improvising the reliability. The day to day exponential growth of digital content has paved way for a high demand in new storage and high traffic in network. On the counter part, there is also a growing essential for a price in effect storage system and high bandwidth for cloud data transfer over a network. As the cloud storage is remotely accessible it provides a cost efficient technology and architecture. When an efficient architecture is deigned, it facilitates the support for transmission, ability to store data in an environment of multiple tenants while leading to computation of data intensively for a business model. Client Deduplication is applied in order to save high consumption of resources in network bandwidth as well as the capability in storage. An user's key is used to encrypt the client's file just before it's stored in cloud. However, the Storage Service (SS) encounters different cipher texts whatever be the data which makes deduplication almost impossible. In addition to this, the common encryption nodes are relatively randomized. This potential problem made us to choose a deterministic encryption technique. The proposed approach checks for the duplicate file in cloud server before actually uploading the file to cloud server without affecting the confidentiality of the file.A Cloud storage can be viewed as a computing model where remote servers are employed to store data. These remote servers can be used to retrieve data with the internet access. With the evolution of cloud storage, the organizations and enterprises have stepped up to outsource their confidential data to many third party service providers called as Cloud Service Providers (CSP). Though there several benefits of cloud storage and computing, few of the notable ones include availability of data, scalability in service, cost efficient resources and disaster recovery. These notable features attract many users to use cloud storage for their personal data. The Cloud Storage Providers usually manage the data stored. Since cloud moves data into a centralized system, it paves for resource sharing and enables access from anywhere across the globe with a network connection. The cloud users usually need not worry about the hinderance in maintaining the cloud environment or software systems as it is taken care by the cloud systems.

II. LITERATURE SURVEY

Aparna et.al [1] have addressed secure data deduplication of authorized users of cloud are allowed to data to read/write. The cloud data confidentiality from public cloud or attackers it be able to encrypt only delicate and treats will be given by data owner. Arokiam et. al [2] have proposed few techniques in cryptography which were applied to solve few problems in the existing system. The access for a user to their data is provided by issuing the data decryption keys for their personal data stored in storage servers. However, the secret keys are maintained by the data owners. Obfuscation of user data happens before it is stored in storage servers. Though Obfuscation is not enough, integration of these techniques increases the security of data being stored while encryption contributing maximum security.

Ateniese et. al [3] have proposed a model using remote checking of data for Provable Data Protection (PDP). This enables the client to verify the

Revised Manuscript Received October 05, 2019

S. Muthurajkumar, Department of Computer Technology, Madras Institute of Technology Campus, Anna University, Chennai, India - 600 044, India,

original data's presence in an untrusted server without actually retrieving it. A sampling of random blocks in sets from server has proved to cut I/O operations drastically by generating Probabilistic Proofs of Possession where the client maintains a constant amount of metadata in order to verify the proof. The transmission of data by the challenge or response protocol minimizes communication over network. This proves that the PDP model supports larger data sets deployed in distributed storage systems and are hence light weighted. This leads to a robust incorporation of mitigating arbitrary amounts of corruption in data.

Baojiang Cui et. al [4] have proposed an approach with key aggregate searchable encryption algorithm. In this approach, the owner of the data would be required to distribute only a single key even though the user has shared large number of documents. This approach was instantiated by a concrete scheme called KASE. In this scheme, the owner of the data needs to surrender a single trapdoor in order to query into their documents. This novel approach has addressed few problems like security in communication, complexity of cloud systems and storage capability.

Bellare et. al [5] have proposed a symmetric encryption scheme called as Message Locked Encryption (MLE) where the message itself is used to determine the encryption and decryption key. Though the authors have no accurate indication on the purpose of the methods, they are identified as an application for secure deduplication.

Bo Mao et. al [6] concentrated to improve the I/O performance of primary storage systems that are deployed in the cloud. Hence they proposed the Performance Oriented I/O Deduplication (POD). The traditional Deduplication schemes for primary storage systems like iDedupe, offline Dedupe are mainly based on capacity. But, the POD is designed to reduce the write and read traffic in network, to improve efficiency in traffic and to maintain storage capacity. One of the integral part of Parity based RAID systems (of Hard Disk Drive) is Online RAID reconstruction. Though computing the hash values consumes extra power, experimental results show that these online RAID reconstruction improves reliability of the system by reducing the reconstruction time.

Chao Yang et. al [8] proved that the server based on actual possession of the whole data present in the storage system with a cryptographically secure technology is more efficient than the one with partial data. In this technique, the client is required to access any small chunk of original data with dynamical coefficient chosen. The File's Provable Ownership could be generated by this method with the ability to maintain high probability in detection of the malfunctioning of any client.

Chen et. al [9] achieved security in deduplication by convergent key management technique. In this technique, the cloud user owns a master key on order the encrypt the convergent key which is then stored in the cloud. This independent master key management technique considerably increases the number of keys generated as the number of cloud users increase. Hence, such a key management technique forces the users to protect their personal master key. In this paper, the author has proposed a novel approach (Dekev) where the users are no longer pushed to store their master keys with themselves rather they are stored across multiple servers. Experimental results shown by the author has proved that this method has improved the security of the system as the Dekev system is implemented using the Ramp Secret Sharing technique.

Deepika et.al [10] have proposed a new way of uploading a file by encrypting sensitive data in order to overcome the attacks in the existing systems like predicting files, creating secured channel and the content distribution attack. It results in avoiding deduplication and hence none of the attacks will be possible. The files can be encrypted by the private key of the user. This key should not be disclosed and should be kept personal by each and every user. However this leads to the problem of dictionary attack. The private key of any user is vulnerable to offline dictionary attacks. Also if two users by accident use same private key then deduplication will still occur. Another problem related with this option is regarding the generation of the personal key and the bookkeeping task for example that if a user forgets his personal key. In the "convergent encryption" method even when the users use different personal keys it is capable of generating identical encrypted files from original files. This method keeps the essence of deduplication while providing a good level of security.

Halevi et. al [12] have confirmed that the deduplication technique is a promising technology where the cloud storage stores strictly only one copy of data. The authors proposed that cloud storage is becoming exponentially popular among users and enterprises where deduplication has considerably reduced the network bandwidth since the data that the users request to upload are compared at the client side itself with those data already present. Though client side deduplication has improved network performance, it also paved way for the hackers and attackers to access arbitrary sized files with the help of small hash signature from the client side itself leading to access for the file for the attackers in the server. This enables the attackers to access entire files of the client from the server.

Hong Liu et. al [14] have taken the privacy issues in cloud storage systems and proposed a Shared Authority based Privacy preserving Authentication (SAPA) protocol where a new mechanism in which anonymous data access request is matched with security and privacy of data storage systems. In addition to this, an attribute based access control is also embedded to the storage mechanism to remember the user that they can access their own data only. When these anonymous requests are processed, proxy re-encryption is applied as the data is shared among multiple users in the cloud. This gives a picture that this mechanism attracts enterprises for multi-client collaborative cloud applications.

Jia Yu et. al [17] proposed a noval approach where a binary tree structure was designed in which the pre-order traversal of the tree was used to determine the secret keys for the client. Inorder to ensure forward security and preserve the property of block-less verifiability, the authors have constructed an authenticator. The proposed algorithm is highly secure and efficient as depicted by the performance analysis and security proof.

Jianghong Wei et. al [18] had proposed an algorithm in order

to build a secure and cost effective data sharing engine that is based on Revocable Storage – Identity Based Encryption (RD-IBE). This algorithm introduces user revocation and updating of cipher text simultaneously which in turn provides forward and backward security to the cipher text. Under the Decisional 1-DBHE assumption, the proposed model has been proved to be adaptive secure in the standard model.

Jingwei et. al [19] have designed a secure system called SecCloud in order to achieve data integrity and Deduplication of data in the cloud. This system helps the clients of the cloud to generate tags associated with the data before putting the data in the network to upload into the cloud. This method also ensures data integrity of secure information stored in the cloud. The authors have also proposed an additional system called SeeCould+ in which the clients of the cloud encrypt their data before putting them in traffic to upload into the cloud.

Joseph et. al [20] focused to develop a system to achieve revocability and confidentiality of data simultaneously. They have introduced a secure mechanism for cloud storage using a two factor data security protection technique. The first authentication involves the data sender to know the identity of the receiver in order to encrypt the data to be shared. On the other hand, the second authentication involves the receiver to use both his/her secret key as well as the sender's secret key to decrypt the data and gain access. In addition to this, the receiver needs to have a security device to complete authentication. This enabled the cloud server to update the corresponding cipher text automatically without knowledge of the data owner.

Junbeom Hur et. al [21] proposed a new approach for data deduplication on the server side. Server side deduplications makes the cloud server eligible to control and access the outsourced data by exploiting the random convergent encryption technique and distribution of ownership group key, in spite of owner changes that occurs dynamically. In this scheme, the deduplication of files occurs in the server which ensures data integrity against any attack. If the deduplication of files encounters a redundant file in the server, the copies are removed immediately. The copies are identified using a hash function. Thus, even if there is no ownership of data, the proposed method ensures confidentiality and privacy of data in the cloud storage. The proposed method is ideal for communication as tag consistency is guaranteed. Also, there are no compromises in fine grained management and security in the cloud ensuring efficient data deduplication.

Kai he et. al [22] have introduced a common framework to achieve data deduplication and data integrity simultaneously by proposing a auditing scheme for secure cloud storage systems. In this method, when a data is uploaded, the server checks the ownership for new owners and the auditor checks for data integrity for deduplication. The author uses proxy re-encryption in order to deduplicate the cloud data to prove that this cloud storage technique is more secure and efficient.

Kan Yang et. al [23] have proposed a revocable data access control technique mainly focused on multi authority storage systems in cloud. I these types of systems, multiple authorities co-exist and each of the authority would have the ability to promote attributes independently. The proposed method is very expressive and efficient that this revocable data access control scheme can achieve forward and backward security using multi authority CP-ABE technique. The author have an analysed the proposed algorithms and confirmed it to be more secure and efficient.

Kan Yang et. al [24] focused on preserving the privacy for auditing protocol by designing a framework for the secure cloud storage systems. The auditing framework was extended to assist dynamic data operations in the cloud server. This type of model is secure and efficient in the random oracle model as the model assists batch auditing in terms of multiple owners and clouds which makes the system cost efficient. Hence, this model reduces compilation cost as the model can support batching without any trusted organizer. The author has combined cryptographic method with bilinear paring by employing the bi-linear property rather than employing masking techniques.

Keelveedhi et. al [25] had proposed that the deduplication in cloud storage systems were performed by secure cloud storage systems like Mozy, Dropbox, etc. in order to save space by maintaining only one copy of the file uploaded in the cloud systems. If the clients frequently encrypt the files being uploaded, then the cost of these operations would be very huge. This tension is resolved by the Message Locked Encryption (MLE), having convergent encryption as the most remarkable manifest. However, these technique may be fooled even by brute force attacks on the files that belongs to a specific group. Hence, the authors have proposed a DupLESS system in order to secure the files from the brute force attacks in the cloud system. In this system, a PRF protocol is being used to encrypt the files being uploaded based on the message based keys obtained from key server. The system proves to be highly secure as the deduplication of the files are performed by the current service achieving confidentiality. The works shows that the DupLESS technique has ensured security and efficient of cloud storage by employing storage service aided by plain text.

Many related works are presented in the literature survey on cloud data storage with security methods [7, 11, 13, 15, 16, 26, 28, 29]. Lakshmi Prithha et. al [27] proposed Data deduplication and Ramp

Secure Secret Key (RSS Key) for a secure access and efficient storage of data. One of the main motive of data deduplication is to eliminate redundant data available in the cloud while the RSS Key enhances the security in the cloud. The author has proposed ALG based data deduplication technique in order to save redundant space thereby improving the security of the system. The authors have embedded AES encryption to the data being stored in the cloud on top of the RSS Key that is generated dynamically. Hence, the AES and RSS techniques pay a unique way for security in the cloud storage systems.

Poornashree et. al [30] proposed a method where the client outsources the data to a cloud server operating remotely which stores and preserves the personal data. This methods is called as Provable Data Possession (PDP). Since, it's

highly costly to maintain these cloud storage servers, the clients are offered to rent the storage system with the help of cloud service providers in order to store their data remotely in a cloud server with a subscription. This forces the clients to make sure that the cloud storage where their data is being stored is secure and possess the original data uploaded by them and ensure that the cloud service providers store all copies of data as per directions issued. However, few important issues of cloud storage systems like security of data, dynamics of data, integrity protection have remained to be tedious tasks. The authors have surveyed many PDP techniques in the cloud storage systems and have compared them in terms of security, efficiency and integrity.

Wenjing et. al [31] employed public auditing supported by privacy preservation to secure the cloud storage for the clients. Upon further research, the author has enabled TPA in order to perform audit for multiple users at the same time efficiently. The authors have performed the performance analysis of the proposed algorithm that reveals that this algorithm is highly efficient and secure.

Zheng Yan et. al [32] proposed a new algorithm based on the challenge in ownership and proxy re-encryption in order to deduplicate the encrypted data that is being stored in the secure cloud storage. The proposed algorithm ensure integrity of deduplication and access control. This algorithm facilitates flexibility in the support for updating of data in addition to the deduplication in spite of data holders being offline. Hence, the result analysis shows that this type of deduplication algorithm promises superior efficiency for practical deployment. Hence, it's suitable for deduplication for big data applications in cloud storage.

III. SYSTEM ARCHITECTURE

The architecture of the system proposed in this paper is depicts in Fig. 1 which consists of eight components namely Client, File Uploading, Proof of Ownership, Cloud service Provider, Cloud DB, File Downloading and User. The main goal of the project is to design a secure cloud data system that has benefits for both the client data (cloud data storage user) and the server data (cloud storage provider). For the benefit of user, an improved cloud data security mechanism for the data stored on the cloud is provided by encrypting the data before even uploading the file to the storage server. On the other side, capacity of the cloud storage is optimally utilised by avoiding redundant storage of the same file on the server. Encrypting the same file by using two different encryption keys produces two different encrypted files which in turn affects de-duplication concept of the cloud service provider. So an efficient mechanism which generates the hash value from the content of the text file and then uses this hash value to encrypt the data is proposed. In order to avoid leakage of data attacks, hash value with unique file identifier id is found which stored in the database. So, when the client uploads the file to the cloud server, the unique id of the file is compared with those stored in the cloud server. To achieve deduplication, the server creates a link to the file by avoiding redundant data storage. The proposed work also effectively utilises the bandwidth of the client because on redundant uploading of file, only the hash value is passed to the server and checked. The client can also provide access to the data stored in cloud to the set of users with the corresponding access permission. When the users requests to download the client's file, the cloud service provider checks the authentication of the user and give access to the files based on the user's access permission.

Client (Cloud User Admin): A Client, informally known as a cloud user admin requests the cloud service providers to store, preserve and share their confidential data with multiple clients. In general a cloud user could be an individual or an enterprise whoever stores their data in cloud.

Cloud Service Provider (CSP): A Cloud Service Provider maintains a cloud storage system with significant amount of resources in order to make the storage distributed among various cloud servers across different regions. The Cloud Service Providers provides service to the Cloud users (clients) in order to store their data in cloud servers for a subscription amount.

Cloud Users: The Cloud Users belong to the group possessed by the Cloud User admin (Client). The Cloud Users are provided access to the contents of the cloud with access rights restrictions (for instance read access, write access, modify access etc).

Each group of users is characterized by a set of users with a Cloud User Admin being the primary user for the group. The Cloud User Admin has privileged rights to access the cloud servers while the other users in the group have access rights limited for each of them. The Cloud Service Provides facilitates user interfaces for the client to manage their data being stored in the server. The Cloud User Admin can modify the access rights for each and every user via the user interface provided. The Cloud Service providers maintain data servers to map the data accurately to the clients with group identifiers to uniquely identify the user groups.

If you are using *Word*, use either the Microsoft Equation Editor or the *MathType* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* MathType Equation). "Float over text" should *not* be selected.

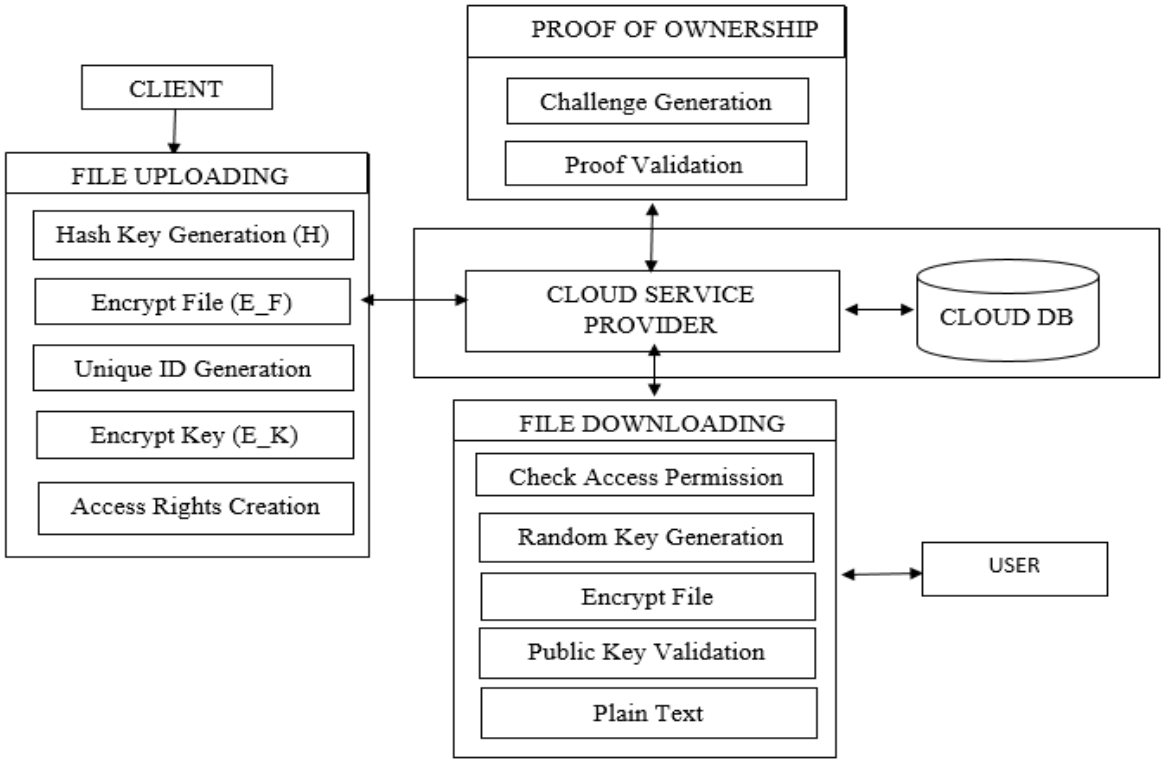

Fig. 1 System Architecture

I. SECURED HASHING ALGORITHM

STEP 1: The input message (M) should always be a product of 512 bits.

Let M be the message of length l, append 1 followed by k zero bits, where l+1+k=448mod512. Append 64 bit equal to the length of message (M).

STEP 2: The padded message is parsed into N 512-bit blocks, $M^{(1)}$,…, $M^{(N)}$. Since the 512bits of the input block may be expressed as sixteen 32-bit words, the first 32 bits of message block i are denoted. $M_0^{(i)}$, the next 32 bits are $M_1^{(i)}$ and so on up to $M_{15}^{(i)}$.

STEP 3: the initial hash value, H(0), shall consist of the following eight 32-bit words, in hex:

H0(0) = 6a09e667
H1(0) = bb67ae85
H2(0) = 3c6ef372
H3(0) = a54ff53a
H4(0) = 510e527f
H5(0) = 9b05688c
H6(0) = 1f83d9ab
H7(0) = 5be0cd19.

These words were obtained by taking the first thirty-two bits of the fractional parts of the square roots of the first eight prime numbers.

For i=1 to N :

1. Prepare the message schedule,W(t).
2. Initialise the eight working variables a,b,c,d,e,f,g,h with (i-1)th hash value like a=H0(i-1) b=H1(i-1) till h=H7(i-1).
3. For t=0 to 63

{

T1= h+ sum(e) + ch(e,f,g)+k(t)+w(t)
T2= sum(a)+maj(a,b,c)
h=g
g=f
f=e
e=d+T1
d=c
c=b
b=a
a=T1+T2

}

4. Compute the ith intermediate hash value H(i)

H0(i) = a + H0(i-1)
H1(i) = b + H1(i-1)
H2(i) = c + H2(i-1)
H3(i) = d + H3(i-1)
H4(i) = e + H4(i-1)
H5(i) = f + H5(i-1)
H6(i) = g + H6(i-1)
H7(i) = h + H7(i-1)

After processing M blocks the resultant hash value H is concatenation of H1,H2 up to H7.

H=H1||H2||H3||H4||H5||H6||H7.

ADVANCED ENCRYPTION STANDARD (AES)

In an Advanced Encryption Standard (AES) technique, the input, plain text is converted to the output, cipher text by performing a finite number of repetitions for transformation as specified by the key size used in it. The key size, aka number of repetitions are structured as follows:

- For keys of size 128 bits, 10 cycles of repetitions is performed.
- For keys of size 192 bits, 12 cycles of

repetitions is performed.
- For keys of size 256 bits, 14 cycles of repetitions is performed.

An AES Encryption algorithm can be characterized by the following steps:

1. Key Expansion
2. Initial Round
3. Intermediate Rounds
4. Final Rounds (without MixColumns)

Each of the stage in AES Encryption algorithm is described as follows:

STEP 1: ***Key Expansion***

In Key Expansion stage, the cipher keys are parsed using Rijndael's Key schedule technique in order to derive the round keys. Each round in an AES would require a separate 128-bit Round key block with an addition of one.

STEP 2: ***Initial Round*** (Add Round Key)

In the initial round, bitwise XOR operation is carried out for combining each byte of the current state with a block of the Round Key derived by the previous step.

STEP 3: ***Intermediate Rounds***

Each of the intermediate rounds are characterized by the following steps.

i. **SubBytes** – This is a non-linear step where each byte of the current state is replaced by another as specified by the lookup table.
ii. **ShiftRows** – This step is mainly a transposition step as the current state is merely altered by transposing last three rows cyclically for a finite number of steps.
iii. **MixColumns** – In this stage, four bytes in each column of the state are combined. Hence, it's a mixing stage.
iv. **AddRoundKey** – (The Round Key is added in this stage)

STEP 4 : ***Final Round*** (without MixColumns)

i. SubBytes
ii. ShiftRows
iii. AddRoundKey

MERKLE HASH TREE

The steps involved in the algorithm are following

STEP 1: The client data file is split into chunks of same size.

STEP 2: Generate hash value for each chunk of file which are the leaf nodes.

STEP 3: Concatenate each pair of hash values and hash the resulting data which acts as the parent node.

STEP 4: Repeat this process using the newly generated hash values across the entire file.

STEP 5: Finally, repeat this entire process until there is a single hash value for the entire file.

II. PERFORMANCE ANALYSIS

The time taken for uploading a file is growing with different sizes of file. Since, deduplication check is performed on the client side the experimental results show that proposed method is more effective in reducing the time taken to upload the redundant file. The proposed method also saves the bandwidth and waiting time of the customers while uploading the redundant file.

Table 6.1 File Based Deduplication

File Name	File size in KB
algo.docx	717
file.docx	553
sample.docx	846
survey.docx	650
storage.docx	339

From Table 6.1, it can be observed that the proposed algorithm used for file based Deduplication in various files.

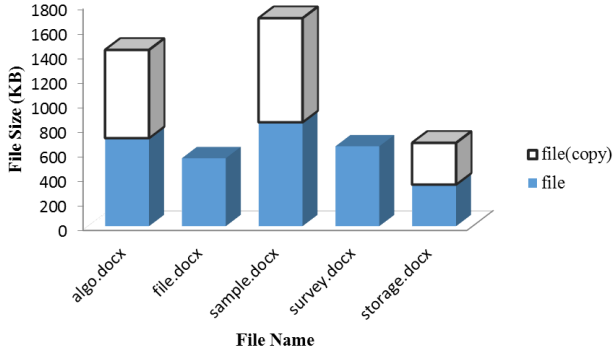

Fig 6.1 Performance Analysis before Deduplication

Figure 6.1 shows the performance analysis of the data Deduplication in various files. The proposed secured

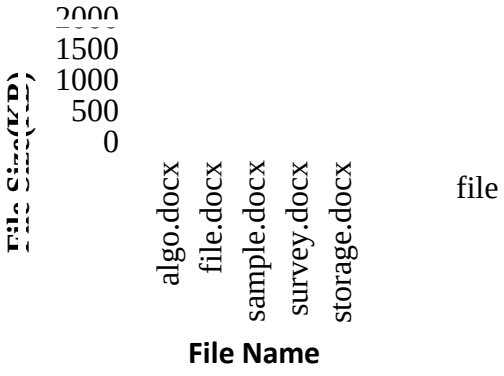

Fig 6.2 Performance Analysis After Deduplication

III. CONCLUSIONS

Previously existing methods for finding deduplication in cloud storage have been successful in achieving either data integrity or storage efficiency but not both. Moreover the existing methods do not provide support for cross user deduplication checking. The client side deduplication methods have failed to protect the privacy of the user.

The proposed method have

been successful in effectively utilising the storage space allocated to the users by CSP. Our method also supports many users to share a common memory space which helps to save more space than the existing methods. It also prevents the confidentiality of user by checking their proof of ownership. It does not allow unauthorised users to access other files stored in cloud by random key generation method thus providing more security to the data files.

The work can be incorporated in real time environment for improving the efficiency and flexibility in auditing the files allowing the user to authenticate their file content stored in the cloud server without downloading the entire file. Another way of providing security is the obfuscation method which can be implemented in order to improve the security of the user file. In order to improve the confidentiality of the system, obfuscation technique can be integrated to the system. This ensures the cloud system to be even more secure by preserving the data from outsiders as well as insiders.

REFERENCES

1. Aparna, B., Kumar, K. S. M. V., "Privacy preserving and authorized data deduplication in public cloud framework," International Journal Of Advanced Research in Computer Science and Software Engineering. 5(10), 2015, pp. 412-434.
2. Arokiam, L., Manikandan, S., "Efficient cloud storage confidentiality to ensure data security," International Conference on Computer Communication and Informatics (ICCCI). 4(3), 2014, pp. 1126-1142.
3. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Khan, O., Kissner, L., Peterson, Z. and Song, D., "Remote data checking using provable data possession," ACM Transaction Information System Security. 14(1), 2015, pp. 12:1–12:34.
4. Baojiang, Cui, Zheli, Liu, Lingyu, Wang., "Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage," IEEE Transaction on Computers. 6(1), 2015, pp. 1-13.
5. Bellare, M., Keelveedhi, S., Ristenpart, T., "Message-locked encryption and secure deduplication." Advances in cryptography-EUROCRYPT 2013 LNCS 7881 Springer, 5(6), 2013, pp. 296–312.
6. Bo, Mao, Hong, Jiang, Suzhen, Wu Lei, Tian., "Leveraging data deduplication to improve the performance of primary storage systems in the cloud," IEEE Transactions on Computers. 65(6), 2016, pp. 278-292.
7. Cao, Ning., "Privacy-preserving multi-keyword ranked search over encrypted cloud data," IEEE Transactions on Parallel and Distributed Systems. 25(1), 2014, pp. 222-233.
8. Chao, Yang, Jian, Ren, Jianfeng, Ma., "Provable ownership of file in de-duplication cloud storage," IEEE Communication and Information System Security Systems. 19(7), 2015, pp. 1-6.
9. Chen, X., Li, J., Li, M., Lee, P., Lou, W., "Secure deduplication with efficient and reliable convergent key management," IEEE Transaction on Parallel Distribution System. 25(6), 2014, pp. 1615–1625.
10. Deepika, Singh, Preetika, Singh., "New challenges for security against deduplication in cloud computing," International Journal Of Advance Research in Computer Science and Management Studies. 2(5), 2014, pp. 653-667.
11. Demel, Abisha, K. S., Revathy, Rajesh, S., "Multithreaded variable chunking in source based deduplication in cloud backup services using support vector machine," International Journal for Research in Applied Science & Engineering Technology (IJRASET). 4(4), 2016, pp. 13-14.
12. Halevi, S, Harnik, D, Pinkas, B, Shulman-Peleg, A., "Proof of ownership in remote storage systems," Proceedings of the 18th ACM conference on computer and communications security, 9(3), 2015, pp. 491– 500.
13. Hermine, Hovhannisyan, Kejie, Lu, Rongwei, Yang, Wen, Qi, Jianping, Wang, Mi Wen., "A novel deduplication-based covert channel in cloud storage service," IEEE Global Communications Conference (GLOBECOM). 47(1), 2016, pp. 1-6.
14. Hong, Liu, Huanshen, Ning, Qingxu, Xiong, Luarence, T., Yang., "Shared authority basedprivacy-preserving authentication protocol in cloud computing," IEEE Transactions on Parallel and Distributed Systems, 26(1), 2015, pp. 768-781.
15. Hussain, Mohammed, Mohamed, Basel, Al-Mourad., "Effective Third Party Auditing in Cloud Computing," 28th International Conference on Advanced Information Networking and Applications Workshops, 10(4), 2014, pp. 91-95.
16. Jia Yu, Kui Ren, Cong, Wang, Vijay, Varadharajan., "Enabling Cloud Storage Auditing With Key-Exposure Resistance," IEEE Transactions on Information Forensics and Security, 10(6), 2015, pp. 1167-1179.
17. Jia, Yu, Kui, Ren, Cong, Wang., "Enabling cloud storage auditing with verifiable outsourcing of key updates," IEEE Transactions on Information Forensics and Security, 11(6), 2015, pp. 690-708.
18. Jianghong, Wei, Wenfen, Liu, Xuexian, Hu., "Secure data sharing in cloud computing using revocable-storage identity-based encryption," Ieee Transactions on Cloud Computing, 14(8), 2015, pp. 1345-1352.
19. Jingwei, Li, Jin, Li, DongqingXie, Zhang, Cai., "Secure auditing and deduplicating data in cloud" IEEE Transactions on Computers. 65(8), 2016, pp. 2386-2396.
20. Joseph, K., Liu, Kaitai, Liang, Willy, Susilo, Jianghua, Liu, Yang, Xiang., "Two-factor data security protection mechanism for cloud storage system," IEEE Transactions on Computers. 65(6), 2016, pp. 256-270.
21. Junbeom, Hur, Dongyoung, Koo, Youngjoo, Shin, Kyungtae, Kang., "Secure data deduplication with dynamic ownership management in cloud storage," IEEE Transactions on Knowledge and Data Engineering, 11(99), 2016, pp. 571-586.
22. Kai, He,Chuanhe, Huang, Hao, Zhou, Jiaoli, Shi, Xiaomao, Wang, Feng, Dan., "Public auditing for encrypted data with client-side deduplication in cloud storage," Wuhan University Journal of Natural Science. 20(4), 2015, pp. 291–298.
23. Kan, Yang, Xiaohua, Jia., "Efficient and revocable data access control for multi-authority cloud storage," IEEE Transactions in Parallel and Distributed Systems. 25(7), 2014, pp. 1735-1745.
24. Kan, Yang, Xiaohua, Jia., "An efficient and secure dynamic auditing protocol for data storage in cloud computing, IEEE Transactions in Parallel and Distributed Systems. 24(9), 2013, pp. 1717-1726.
25. Keelveedhi, S., Bellare, M., Ristenpart, T., "Dupless: server aided encryption for deduplicated storage, Proceedings of the 22nd Usenix Conference on Security, 9(2), 2015, pp. 179–194.
26. Kumar, Naresh, Rahul, Rawat, Jain, S., C., "Bucket based data deduplication technique for big data storage system," 5th IEEE International Conference on Reliability Infocom Technologies and Optimization (ICRITO), 3(2), 2016, pp. 267-271.
27. Lakshmi, Pritha, Velmurugan, Godfrey, Winster, Vijayaraj, "Deduplication based storage and retrieval of data from cloud environment," IEEE International Conference on Innovation Information in Computing Technologies (ICIICT). 14(5), 2015, pp. 1-6.
28. Liu, Joseph., "Two-factor data security protection mechanism for cloud storage system," IEEE Transactions on Computers, 65(6), 2016, pp. 1992-2004.
29. Pritha, Lakshmi, N., "Deduplication based storage and retrieval of data from cloud environment," IEEE International Conference on Innovation Information in Computing Technologies, 19(7), 2016, pp. 1-6.
30. Poornashree, B., R., Srividhya, S., "A survey on provable data possession in cloud computing systems, International Journal Of Engineering Research & Technology, 5(7), 2016, pp. 271-292.
31. Wenjing, Lou. Kui, Ren. Qian, Wang. Sherman, S.M. Chow, Cong, Wang., "Privacy-preserving public auditing for secure cloud storage," IEEE Transactions on Computer, 62(2), 2013, pp. 378-395.
32. Zheng, Yan. Wenxiu, Ding. Xixun, Yu. Haiqi, Zhu. Robert, H. Deng., "De-duplication on encrypted big data in cloud," IEEE Transactions on Big Data, 2(2), 2016, pp. 138-150.

AUTHORS PROFILE

S. Muthurajkumar received M.E. degree in Computer Science and Engineering from Anna University, Chennai where he has completed Ph.D. and he is working as an Assistant Professor in the Department of Computer Technology, MIT Campus, Anna University, Chennai. He has published more than 8 articles in journals and conferences. His area of interest is Cloud Networks security, Cloud Computing and Data Mining.