A Secure and Authorized Data De-duplication and Continuous Data Auditing

Trupti Deore[1], J.V.Shinde[2], Dr.A.H.Sable[3]

[1]Student, Comp Dept, Late G.N.Sapkal C.O.E.,Nashik, India
[2]Asst.Professor,Comp Dept,Late G.N.Sapkal C.O.E.,Nashik, India
[3] Asst.Professor,School of computation,SRTMU,Nanded,India

ABSTRACT

De-duplication eliminates redundant information segments from the data storage and reduces the scale of storage information. This is often significantly helpful in Cloud Storage wherever information is transferred to the storage target over WAN. De-duplication with Cloud Storage not solely reduces the space for storing needs, however conjointly reduces the information that's transferred over the network leading to quicker and economic data protection operations. It raises problems about security, possession and data integrity. Recently, many de-duplication schemes are planned to unravel this downside by permitting every owner to share an equivalent cryptography key for equivalent information. However, most of the schemes suffer from security flaws, since they are doing not take into account the dynamic changes within the possession of outsourced information that occur often in a very sensible cloud storage service. The proposed system includes completely unique server-side de-duplication over encrypted information and server side continuous data auditing. It permits the cloud server to regulate access to outsourced information even once the possession changes dynamically by exploiting irregular oblique encoding and secure possession cluster key distribution. Along with the deduplication check, system also proposes an in-house continuous data auditing for data integrity check. The system performance is tested and it proves that the system is as efficient as the previous system and the extra feature of continuous auditing requires a negligible computation overhead.

Keyword : - *De-duplication, cloud storage, encryption, proof-of-ownership, revocation, data auditing, integrity checking, continuous auditing, Markle Hash Tree.*

---

1. INTRODUCTION

Cloud computing permits access to resources from anyplace and at any time using the internet services. The most advantage of victimization cloud storage from the customers' purpose of read is that customers will scale back their expenditure in getting and maintaining storage infrastructure whereas solely paying for the quantity of storage requested, which may be scaled-up and down upon demand. however it's conjointly terribly true that cloud Storage isn't infinite.

Data de-duplication is that the best thanks to handle these knowledge. Dropbox, Wuala , Mozy, and Google Drive, use a de-duplication technique, wherever the cloud server stores solely one copy of redundant knowledge and provides links to the copy rather than storing alternative actual copies of that knowledge, despite what percentage shoppers raise to store the info. However, typical cryptography makes deduplication not possible for the subsequent reason. De-duplication techniques cash in of knowledge similarity to spot constant data and scale back the space for storing. In distinction, cryptography algorithms randomize the encrypted files so as to create ciphertext indistinguishable from on paper random knowledge.

A de-duplication theme over encrypted information is planned. The planned theme ensures that solely approved access to the shared information is feasible. It is taken into account to be the foremost necessary challenge for economical and secures cloud storage services within the surroundings wherever possession changes dynamically. It is achieved by exploiting cluster key management mechanism in every possession group.

As compared to the previous de-duplication schemes over encrypted information, the planned theme has the subsequent benefits in terms of security, itegrity and potency. First, dynamic possession management guarantees the backward and forward secrecy of de-duplicated data upon any possession amendment. As critical the previous schemes, the information secret writing secret is updated and by selection distributed to valid homeowners upon any possession amendment of the information through a homeless group key distribution mechanism employing a binary tree. The possession and key management for every user are often conducted by the semi-trusted cloud server deployed within the system. Thus, the planned theme delegates the foremost punishing tasks of possession management to the cloud server while not leaky any counsel thereto, instead of to the users. Second, the planned theme ensures security within the setting of prisoner by introducing a re-encryption mechanism that uses an extra cluster key for dynamic possession cluster. Thus, though the secret writing key (that is that the hash price of the file) is discovered within the setting of prisoner, the privacy of the outsourced information remains preserved against outside adversaries, whereas de-duplication over encrypted information remains enabled and information integrity against poison attacks is secure.

A novel approach is proposed in the system which provides data efficient approach for data de-duplication on cloud storage to preserve privacy and also address the issue of ownership management in cloud storage and also proposes new guidelines to address data integrity against inconsistency attack.

The proposed approach also includes the data integrity checks using in-house data auditing technique. A continuous monitoring feature is added at the cloud end to check any change in cloud server data storage. The system raises a notice at any illegal change made in data storage.

2. REVIEW OF LITERATURE

De-duplication techniques will be classified into 2 different approaches:

1. De-duplication over unencrypted data and
2. De-duplication over encrypted information.

In the former approach, most of the prevailing schemes have been planned so as to perform a prisoner method in an economical Associate in Nursing strong manner, since the hash of the file, that is treated as a proof for the whole file, is at risk of being leaked to outside adversaries because of its comparatively tiny size. In the latter approaches, information privacy is the primary security requirement. The security is provided to guard against not solely outside adversaries however additionally within the cloud server. Thus, most of the schemes are planned to produce information encryption, whereas still taking advantage of a deduplication technique, by sanctioning information homeowners to share the encryption keys within the presence of the within and outside adversaries. Since encrypted information is given to a user, information access management will be to boot implemented by selective key distribution once the PoW method. However, not a lot of work has nonetheless been done to deal with dynamic possession management and its connected security downside.

Author [2] incontestable however information de-duplication technique is used as an aspect channel that reveals data to malicious users regarding the contents of files of alternative users. Because the volume of information can increase, so can the demand for on-line storage services, from easy backup services to cloud storage infrastructures. Though de-duplication is merely once applied across multiple users, cross-user de-duplication has serious privacy implications. Some easy mechanisms can modify cross-user de-duplication whereas greatly reducing the danger of information discharge.

Author [3] conjointly introduced an analogous attack state of affairs on cloud storage that uses de-duplication across multiple users. The notion of proofs-of-ownership (PoWs), that lets a consumer expeditiously influence a server that the consumer holds a file, instead of some short data regarding it. They formalize the idea of proof-of-ownership, below rigorous security definitions, and rigorous potency needs of computer memory unit scale storage systems. They then gift solutions supported Merkle trees and specific encodings, and analyze their security. As the volume of knowledge will increase, thus will the demand for on-line storage services, from straightforward backup services to cloud storage infrastructures. though de-duplication is only once applied across multiple users, cross-user de-duplication has serious privacy implications. Some straightforward mechanisms will modify cross-user de-duplication whereas greatly reducing the danger of knowledge discharge.

Author [4] additionally planned a leakage-resilient de-duplication theme to resolve the info integrity problem. This theme additionally permits information owner to cypher data with a random elite key. In an exceedingly proof of possession theme, any owner of identical file F will persuade the cloud storage that he/she owns file F in an exceedingly strong and economical approach, although a definite quantity of arbitrary info concerning file F is leaked.

Shin et al. [5] planned a de-duplication theme over encrypted knowledge that uses predicate coding. This approach permits de-duplication solely of files that belong to identical users that severely reduces the effect of de-duplication. Recently, Li et al. [6] planned, a replacement construction within which users don't have to be compelled to manage any keys on their own however instead firmly distribute the merging key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions laid out in the planned security model.

As a symptom of idea, we have a tendency to implement De-key victimization the Ramp secret sharing theme and demonstrate that De-key incurs restricted overhead in realistic environments and merging key management theme within which users distribute the convergent key shares across multiple servers by exploiting the Ramp secret sharing theme [7]. Li et al. [8] additionally planned a licensed de-duplication theme within which differential privileges of users, additionally because the knowledge, are thought of within the de-duplication procedure in an exceedingly hybrid cloud setting.

Jin et al. [7] projected Associate in Nursing anonymous de-duplication theme over encrypted knowledge that exploits a proxy re-encryption rule and propose a theme to handle the de-duplication of encrypted knowledge with efficiency and firmly with the assistance of guaranteeing the possession of the shared file, encrypting knowledge mistreatment keys at user's can and realizing the anonymous store through the digital certificate. This approach tends to accomplish this aims through proof of possession (POW), proxy re-encryption (PRE) and digital certificate.

Bellare et al. [9] projected a server-aided MLE that is secure against brute-force attack, that was recently extended to interactive MLE [10] to produce privacy for messages that arm each related to and keen about the general public system parameters. However, these schemes don't handle the dynamic possession management problems concerned in secure de-duplication for shared outsourced knowledge and In DupLESS, shoppers encode underneath message-based keys obtained from a key-server via Associate in Nursing oblivious PRF protocol. It permits shoppers to store encrypted knowledge with Associate in Nursing existing service, have the service perform de-duplication on their behalf, and nevertheless achieves sturdy confidentiality guarantees. The system tend to show that cryptography for de-duplicated storage are able to do performance and house savings near that of mistreatment the storage service with plaintext knowledge. Shin et al. [5] projected a de-duplication theme over encrypted knowledge that uses predicate cryptography. This approach permits de-duplication solely of files that belong to identical user, that severely reduces the result of de-duplication.

In CE[11] Convergent encryption is proposed. The convergent key encryption resolves the problem of de-duplication checking over encrypted data. A convergent key is generated from the data hence same key is generated fro the same data and same cipher text is generated from the same plaintext data.

Leakage-resilient de-duplication[12] and Message-locked encryption (MLE)[13] techniques are proposed in a literature. These schemes provide solution against tag consistency attack. In MLE data is encrypted using random key. Then the random key is again encrypted using KEK derived from shared users identity. The data integrity is checked by decrypting the data encryption key with the same KEK.

SecCloud and SecCloud+ is proposed in[14] . It proposes a data auditing and data de-duplication in map reduce cloud environment. It uses a Proof of ownership (PoW) Protocol and prevents the data leakage of side channel information in data de-duplication.

The in-house data auditing is proposed by M. Alles, et al[15]. The continuous monitoring and auditing is introduced in ERP system. This system identified the management of audit alarms and the prevention of alarms floods as it is critical task in CMBPS implementation process. They construct an approach to solve the problem of implementation of hierarchical structure of alarms. Only diverse practical experience provides the facts necessary for identifying trade-offs between effectiveness, efficiency and timeliness of audit procedures and determining how to make CMBPC implementations worthwhile.

S. Lins, et al, about data auditor to verify integrity of data, compliance in data and the dynamic infrastructure of cloud. To address the gap between continuous auditing a conceptualize architecture of CA has been introduced. It supports data auditor to to classify whether or not a high frequency auditing of their CSC criteria is needed. From the proposed approach CA, high level security as well as reliability is achieved in the cloud environment. But the methodologies to efficiently and continuously audit cloud services are remains immature. [16].

De-duplication with ownership management includes data deduplication checking at server end and with secure ownership group key distribution. The proposed scheme uses KEK tree for ownership group key distribution. It uses AES algorithm for encryption. This scheme provides tag inconsistency check for tag integrity preservation. This system do not provide data integrity for users[1].

PROBLEM FORMULATION:

There is need to generate server side data de-duplication system over encrypted data that helps to manage data ownership with multiple users, user revocation and data integrity checking continuously.

3. SYSTEM DESCRIPTION:

Following is the system architecture. It consists of two nodes. User end and cloud end. User is responsible for data uploading and downloading. The de-duplication is performed at the server end. Along with the deduplication system continuous data auditing is performed at the server end.

PRELIMINARIES:

1: AES Encryption:

The Advanced Encryption Standard (AES), also known by its original name Rijndael. It is cryptographic algorithm used for data encryption & decryption. It is the based on principle of substitution-permutation network" and the combination of substitution & permutation. It has fixed block size of 128-bit and key size is of 128, 192 or 256 bits.For instance, if there are 16bytes: {B0, B1, B2,.....B15} these bytes are represented as following matrix:

$$M = \begin{pmatrix} B0 & B4 & B8 & B12 \\ B1 & B5 & B9 & B13 \\ B2 & B6 & B10 & B14 \\ B3 & B4 & B11 & B15 \end{pmatrix}$$

The number of repetitions is depending upon the size of key for AES cipher. Ten cycles of repetition are required for for 128-bit keys. The algorithm follows 4 main streps:
1: Key Expansion: round keys are derived from the cipher key.
2: Initial Round key addition: Using bitwise XOR round key added
3: Processing Round: It contains operation like: sub bytes, shift rows, mix columns and add round key.
4: Final Round: It contains sub bytes, shift rows and add round key.

2: HMAC Algorithm:

It is also a cryptography algorithm termed as a keyed-hash message authentication code. HMAC is a specific type of message authentication code(MAC). HMAC does not perform message encryption. HMAC is represented as:
HMAC(k,m)=H((K' ⊕ opad) || H(K' ⊕ ipad)|$m))
where,
H is cryptographic hash function,
K is secret key,

m is the message to be authenticated,
K' is another secret ke, derived from the original key K,
|| denotes concatenation,
⊕ denotes exclusive or(XOR),
opad is the outer padding(0x5c5c5c...5c5c, one-block-long hexadecimal constant),
and ipad is the inner padding (0x363636....,3636, one-block-long hexadecimal constant).

3: MD5:

MD5 algorithm was developed by Professor Ronald L. Rivest. MD5 algorithm is intended for digital signature application. Takes as input a message of arbitrary length and produces as output a 128 bit fingerprint or message digest of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest. A large file must be compressed in a manner before being encrypted with a private key under a public-key cryptosystem such as PGP.

Algorithmic Steps:

Step1: Suppose a b-bit message as input, and that we need to find its message digest.
Step2: append padded bits: The message is padded so that its length is congruent to 448, modulo 512. Means extended to just 64 bits shy of being of 512 bits long. A single 1 bit is appended to the message, and then 0 bits are appended so that the length in bits equals 448 modulo 512.
Step3: append length: A 64 bit representation of b is appended to the result of the previous step. The resulting message has a length that is an exact multiple of 512 bits.
Step4: Initialize MD Buffer A four-word buffer (A,B,C,D) is used to compute the message digest. Here each of A,B,C,D, is a 32 bit register.
Step5: Process message in 16-word blocks.
Step6: Output. The message digest produced as output is A, B,C, D. That is, output begins with the low-order byte ofA, and end with the high-order byte of D.

4: MHT Construction:

Markle hash tree contains hash values at the leaf nodes. Every non-leaf node is labeled with a value cryptographic hash value. The MD5 hash function is used to generate hash.
In this system user unique identity values are present at the child node. The Hash function is applied to the unique user identifier. The child node contains the hash value of users' identifier. The parent nodes contains hashing of respective child nodes. The common least parent hash value of all child nodes is used as data encryption key for shared data.

4. SYSTEM ARCHITECTURE

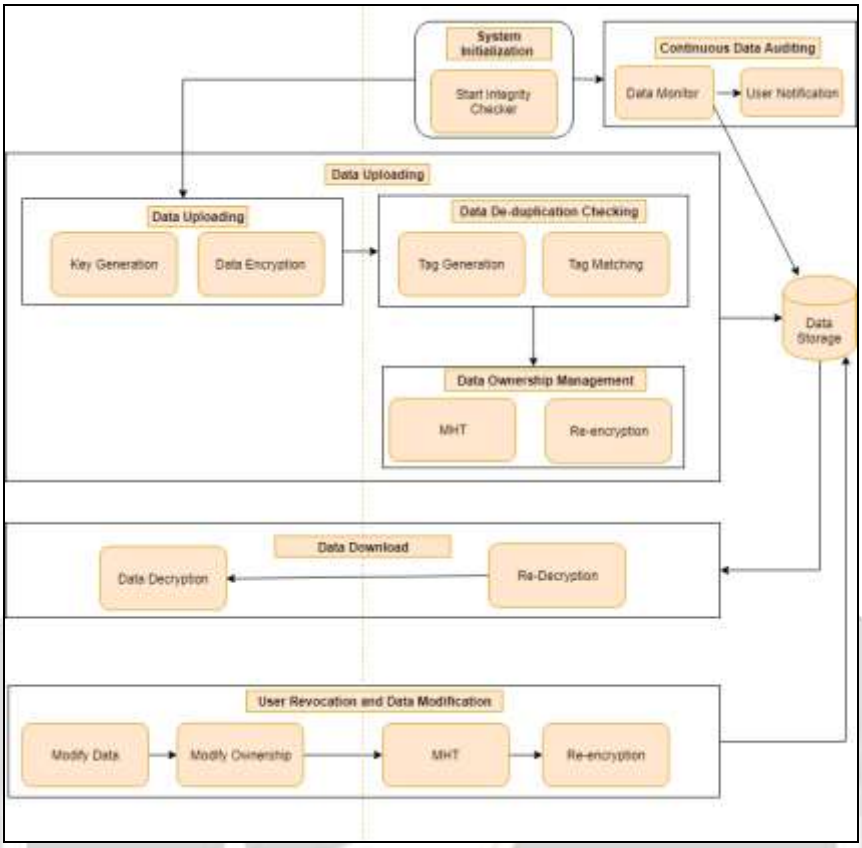

Fig 1: System Architecture

The system has multiple users. User registers on server and then has facility to upload and access the data. After user authentication user has facility to perform following operations:

1: upload and share the data:

In data upload, user data is uploaded on server in encrypted form. User has facility share a data with group of users. For encryption AES algorithm is used. A convergent key is generated from data using HMAC algorithm. The convergent key is used for data encryption. At the server end data de-duplication is checked. Only single copy of data is preserved on the server. For data de-duplication checking data tags are generated. Tags are generated using MD5 algorithm. By matching data tags, data availability is checked on server. If data is already present then only proof of ownership is run by re-encrypting the data with data users' identity. For re-encryption markle Hash tree is generated. Child nodes represent the user identity. The common parent node of all users' of a file generates the key of re-encryption. At the server end data re-encryption key is generated and data is re-encrypted and saved at the server end.

Following diagram 2 represent the Tree structure. For example system has 8 users. If file has u1,u2,u3 and u4 users. Then the common parent node v2 is the key for re-encryption.

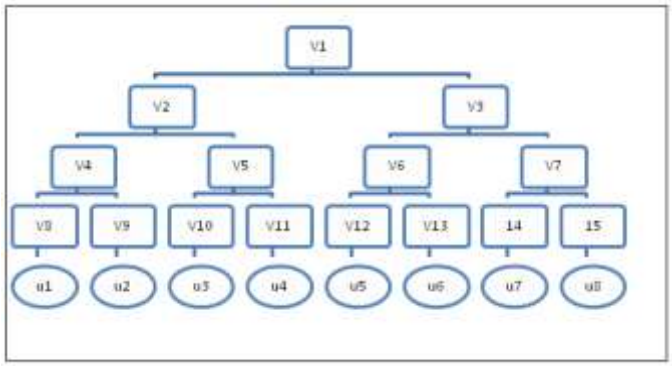

Fig 2: Tree generation of re-Encryption Key

2: Download Data:

For data downloading initially users proof of ownership is checked. The file users list is extracted and decryption key is generated from the Markle Hash Tree. The first decryption is performed at the server end. The data is get downloaded with the convergent key. Using convergent key data is again decrypted and original file is saved at the users end.

3: Revocation of user and Data modification:

After data download, user can modify the file content. File owner has facility to delete the file. The file owner can modify the user access description i.e. owner can share the same file with other users or can remove the user from group. The removal of user is called as user revocation and afterwards user will not be able to access the file.

After data editing the data is again encrypted by convergent key using AES algorithm. At the server end de-duplication is checked. And again re-encryption is performed and data is saved at server end.

Data owner can delete the file. At the server end file references are checked. If more than one group is accessing the same file then only reference of group is deleted and data is re-encrypted with remaining users key. If single reference of group is present then original file is deleted from the server.

In case of user revocation, data is re-encrypted at the server end with file user's identity.

4: Data Auditing:

A continuous data auditing is performed at cloud end. A file watcher is initialize at system start up process. The watcher system monitors the operations performed on file like: ENTRY_CREATE, ENTRY_DELETE, ENTRY_MODIFY. The watcher system checks that the performed change in file is made by authorized user or not. If it is not made by the authorized user then system generates a notification to file owner.

4.1 System Algorithm

1: De-duplication system Algorithm:

Input: File To upload UF,
Sharing Information SI
File name to Download FNM
Output: CF : File at cloud end,
Downloaded File DF
User notification NF
Processing
1. Define data storage path at cloud end
2. Generate MHT for File users
3. if File upload activity
Select file UF for uploading and sharing rights SI

Generate convergent key from UF file data using HMAC algorithm
Encrypt data using AES
Generate Tag T from file
Upload tag T and sharing information SI to cloud
Check for data deduplication using tag
If deduplication found
    Read Data from cloud
    Find re-encryption key K from MHT
    Re-encrypt data using AES
    Notify User
Else
    Upload Data
    Find re-encryption key from MHT
    Re-encrypt data using MHT
    Save file at cloud end CF
    Notify User
4. If File download activity
Select file FNM to download
Apply re-decryption at cloud end
Download File
Apply decryption
Save file DF
5. If Revoke User
Modify user list SI for file CF
Upload list SI to cloud
Find re-encryption key K from MHT
Re-encrypt data using AES

2: Continuous Data Auditing:
Input: File Storage Path
Output: Notification NF
Processing:

1. Initialize Monitoring operation List: ENTRY_CREATE, ENTRY_DELETE, ENTRY_MODIFY
2. Initialize file watcher thread at cloud end
3. While file watcher not stop
4. E: Read event
5. F: get file context
6. If E is in (ENTRY_CREATE, ENTRY_DELETE, ENTRY_MODIFY)
7. Check for dataset entry
8. If valid entry not found
9. Notify owner of a file F
10. While end

5. IMPLEMENTATION

The system is implemented in java using jdk 1.8. The server system is developed using apache tomcat. mysql 5.3 is used to save database The client and server are hosted on windows system having core i5 processor and 4 gb ram.

Datasets:

Synthetic Dataset: Hybrid dataset is generated. It contains various files with different file formats like: text files, pdf files, image files ,video files and archive files. These files are collected from different sources. The files size varies from 100 kb to 5 mb.

Performance Measure:

1. Time : The execution time is evaluated for various operations. Like file data encryption, tag generation, deduplication checking, re-encryption, and data auditing.
2. Efficiency Comparison: The complexity of system is compared with existing system and the comparative study of feature specific system complexity is elaborated.
3. Data Integrity: As a part of contribution, continuous data auditing is proposed. The time required for data auditing is compared with number of files present at the cloud end and the operation performed.

5.1 Result and Analysis:

1. Time Comparison:

a) Time analysis for Encryption, decryption and deduplication checking

Following table shows the time analysis of the system for various file sizes. The file size varies from 1 MB to 5 mb. The data encryption time, decryption time, tag generation time and deduplication check time is captured.

Table 1: Time analysis for Encryption, decryption and deduplication checking

File Size	Data encryption (in Milliseconds)	Data Decryption(in Milliseconds)	Tag generation(in Milliseconds)	Deduplication check(in Milliseconds)
1mb	15	12	8	20
2mb	28	20	17	22
3mb	47	32	29	25
4mb	61	48	41	28
5mb	76	63	55	29

Data Encryption and decryption is performed using AES-128. The time required for decryption is less than the encryption time. Tags are generated after file encryption and uploaded to cloud for data deduplication checking. To preserve data deduplication less time is required than uploading the whole document. Deduplication saves server space as well as bandwidth of network.

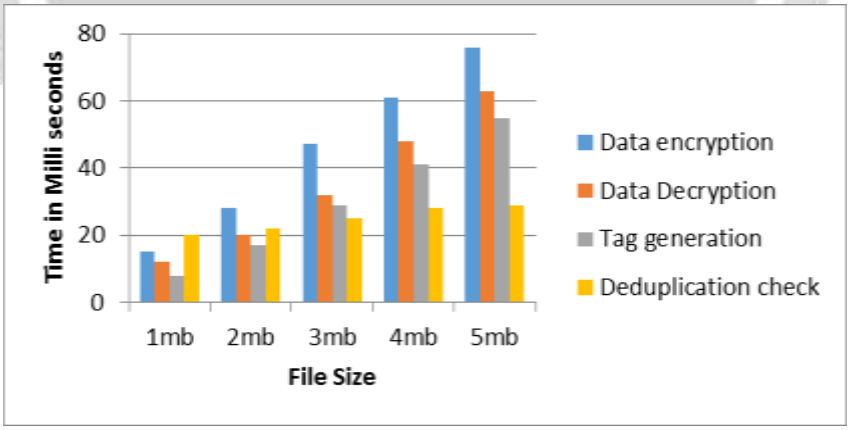

Fig 3: Time analysis for Encryption, decryption and deduplication checking

b) Time analysis for data uploads in deduplication system

In case of data deduplication only tags and sharing information is transferred to the server rather than the whole file data. If file level deduplication found then mapping time of existing file is captured. While mapping to the existing file with new users sharing list then only re-encryption is performed.

Table 2: Time analysis for data uploads in deduplication system

File Size	Upload Without Deduplication time in milliseconds	Upload With Deduplication time in milliseconds
1mb	2203	2406
2mb	2793	2899
3mb	3443	3595
4mb	4630	4865
5mb	5291	5432

Following fig 4. Shows the graphical analysis for data upload without deduplication and data upload with deduplication. If data deduplication found then mapping to the existing file requires less time as compared to the uploading the whole data. Data deduplication saves the upload time, storage space on cloud and network bandwidth.

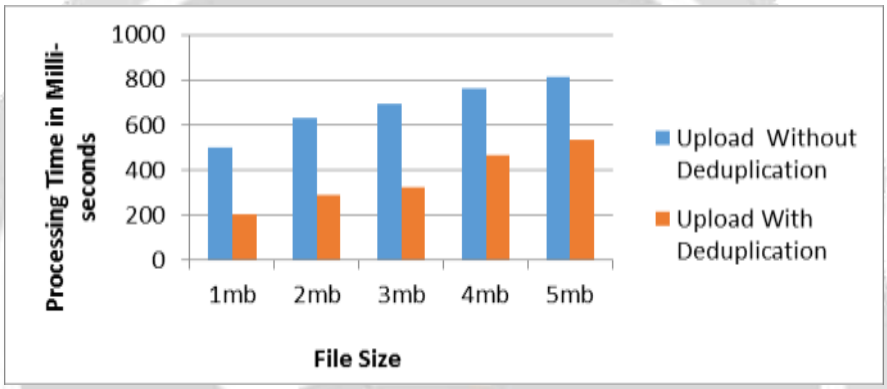

Fig 4: Time analysis for data uploads in deduplication system

c) Time analysis for data uploads in deduplication system

In case of data deduplication only tags and sharing information is transferred to the server rather than the whole file data. If file level deduplication found then mapping time of existing file is captured. While mapping to the existing file with new users sharing list then only re-encryption is performed.

Table 3: Time analysis for Data download and modification

File Size	Data Download time in milliseconds	Data Modify+User revocation time in milliseconds
1mb	469	654
2mb	596	703
3mb	662	763
4mb	707	783
5mb	783	1132

The following graph shows the time analysis for data downloading and data modification. Data modification requires more time than data uploading, because in data modification phase also requires data re-decryption of existing data.

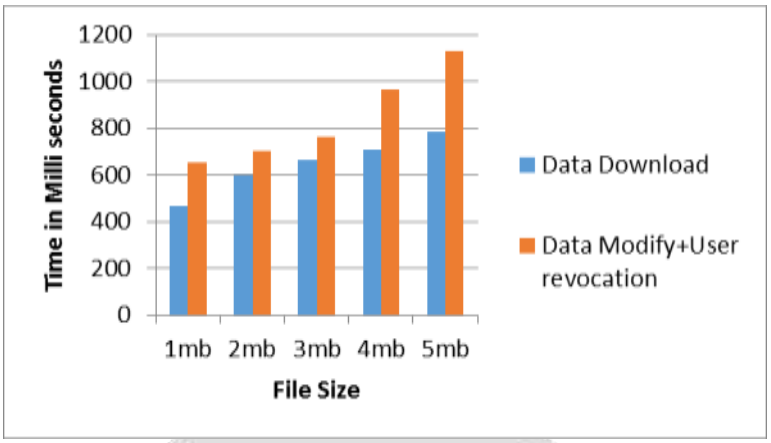

Fig 5: Time analysis for Data download and modification

d) Data Audit Time

The continuous data auditing is present at cloud end. The Time required for data auditing is captured for various file sizes. The data auditing includes monitoring the file data, checking for unauthorized access and user notification generation.

Table 4: Data Audit Time

File Size	Audit Time(in Sec)
1mb	2.167
2mb	2.207
3mb	2.503
4mb	2.794
5mb	3.285

Following graph shows the time analysis for data auditing. As the file size increases the time required for data auditing also increases slightly. The file size and data audit time is not directly proportion.

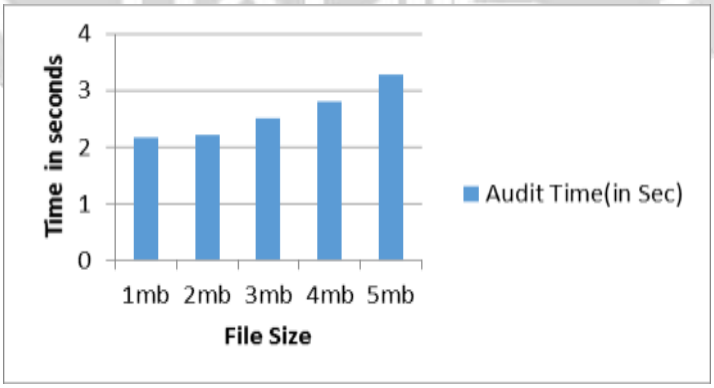

Fig 6: Data Audit Time

e) Time comparison between existing and proposed System

Following table shows the time comparison for data uploading in existing and proposed system. The upload time is equal in existing and proposed system. The proposed system includes the data auditing facility. The audit features present in existing systems (studied in literature) requires high time because auditing requires metadata generation at uploading phase. The continuous audit facility does not require metadata creation at

the time of data upload. The audit service monitors the files present at cloud end and the monitoring thread is completely independent of any user functionality like data upload, modification, download, etc.

Table 5: Time comparison between existing and proposed System

File Size	Data Upload (Existing System) time in milliseconds	Data Upload (Proposed system) time in milliseconds
1mb	502	502
2mb	630	630
3mb	693	693
4mb	762	762
5mb	814	814

Following graph shows the time comparison among existing and proposed system for data uploading. The data uploading is tested for various file sizes. The Existing and proposed system requires equal time for data uploading even with the addition feature of data auditing present in proposed system.

Fig 7: Time comparison between existing and proposed System

2. System Features Comparison:

In the following table, the proposed system is compared with the multiple existing systems in terms of feature present in the system. The proposed system provides a single solution to the user containing the combination of multiple features.

Table 6: System Comparison

Scheme	**Encrypted Deduplication**	**Tag Consistency**	**Ownership management**	**Data Auditing**	**Continuous data Auditing**
CE [11]	YES	-	-	-	-
LR [12]	YES	YES	-	-	-
RCE[13]	YES	YES	YES	-	-
SecCloud+[14]	YES	-	-	YES	-

CA in cloud service contexts[16]	-	-	-	-	YES
Data dedup with DOM[1]	YES	YES	YES	-	-
Proposed system	YES	YES	YES	YES	YES

3. Efficiency comparison
The following table shows the theoretical comparative study among various systems in terms of feature and its complexity. The comparison is shown on the basis of communication overhead, storage overhead and monitoring overhead. Communication overhead includes data uploading, downloading and modification in terms of user revocation changes and data deduplication.

Table 7 : Efficiency comparison

Scheme	Communication Overhead			Storage Overhead		Monitoring Overhead
CE[11]	Upload Message time	Download Message Time	Rekeying Message Size	Key Size	Tag Size	Audit
LR[12]	CC + CT + CID	CC		CK	CT	
RCE[13]	CC + 3CK + CT + Cr + CID	CC(+CPow + CK + CT)+		2CK+CM.CR	CT	
Data dedup with DOM[1]	CC + CK + CT + CID	CC + CK + CT		CK	CT	
Proposed	CC + CK + CT + CID	CC + CK + CT	$(n-m)\log\frac{n}{n-m}CK$	(logn + 1)CK	CT	Log (CM*K)

Where,

CM : Size of a data or file
CC :Size of an encrypted data
CK :Size of a key
CT :Size of a tag
CID : Size of an user identifier
CR : Merkle hash tree node size
CPoW : Proof of ownership details
N: Number of system users
M: Number of owners in an ownership list
K: monitoring operations i.e. insert, update, delete, here k = 3

6. CONCLUSION

Dynamic ownership management is an important and challenging issue in secure deduplication cloud environment. This may happen when the unauthorized users have possessed the data at some time instance and stored the derived

key encryption key K until the moment of request; or, they could receive it from the other colluders. The proposed system is a novel secure data deduplication scheme to enhance a fine-grained ownership management by exploiting the characteristic of the cloud data management system. The proposed scheme features a re-encryption technique that enables dynamic updates upon any ownership changes in the cloud storage. Whenever an ownership change occurs in the ownership group of outsourced data, the data is re-encrypted with an immediately updated ownership group key, which is securely delivered only to the valid owners. Thus, the proposed scheme enhances data privacy and confidentiality in cloud storage against any users who do not have valid ownership of the data, as well as against an honest-but-curious cloud server. Continuous Data auditing is also proposed to ensure data integrity. The scheme allows full advantage to be taken of efficient data deduplication over encrypted data. In terms of the communication cost, the proposed scheme is more efficient than the previous block level deduplication schemes. The proposed scheme achieves more secure and fine-grained ownership management in cloud storage for secure and efficient data deduplication.

7. REFERENCES

[1] Hur, Junbeom, et al. "Secure data deduplication with dynamic ownership management in cloud storage." IEEE Transactions on Knowledge and Data Engineering 28.11 (2016): 3113-3125.

[2] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services, the case of deduplication in cloud storage,"IEEE Security & Privacy, vol. 8, no. 6, pp. 4047, 2010.

[3] S. Halevi, D, Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," Proc. ACM Conference on Computer and Communications Security, pp. 491500, 2011.

[4] J. Xu, E. Chang, and J. Zhou, "Leakage-resilient client-side deduplication of encrypted data in cloud storage," ePrint, IACR, http://eprint.iacr.org/2011/538.

[5] Y. Shin and K. Kim, "Equality predicate encryption for secure data deduplication," Proc. Conference on Information Security and Cryptology (CISC-W), pp. 6470, 2012.

[6] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," IEEE Transactions on Parallel and Distributed Sytems, Vol. 25, No. 6, 2014.

[7] G.R. Blakley, and C. Meadows, "Security of Ramp schemes," Proc. CRYPTO 1985, pp. 242268, 1985.

[8] J. Li, Y. K. Li, X. Chen, P. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 5, pp. 12061216, 2015.

[9] X. Jin, L. Wei, M. Yu, N. Yu and J. Sun, "Anonymous deduplication of encrypted data with proof of ownership in cloud storage," Proc. IEEE Conf. Communications in China (ICCC), pp.224-229, 2013.

[10] M. Bellare, S. Keelveedhi, T. Ristenpart, "DupLESS: Serveraided encryption for deduplicated storage," Proc. USENIX Security Symposium, 2013.

[11] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," Proc. International Conference on Distributed Computing Systems (ICDCS), pp. 617–624, 2002.

[12] J. Xu, E. Chang, and J. Zhou, "Leakage-resilient client-side deduplication of encrypted data in cloud storage," ePrint, IACR, http://eprint.iacr.org/2011/538.

[13] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," Proc. Eurocrypt 2013, LNCS 7881, pp. 296–312, 2013. Cryptology ePrint Archive, Report 2012/631, 2012.

[14]Jingwei Li, Jin Li, Dongqing Xie, and Zhang Cai, ``Secure Auditing and Deduplicating Data in Cloud", IEEE transaction, vol.65, no.8, auguest 2016

[15] K. Singh, P. J. Best, M. Bojilov, and C. Blunt, "Continuous auditing and continuous monitoring in ERP environments", Inf. Syst. J, vol. 28, no. 1, pp. 287-310, 2013

[16] S. Lins ,S. Schneider, A. Sunyaev,"Trust is Good, Control is Better: Creating Secure Clouds by Continuous Auditing ", IEEE Trans. On cloud computing, vol. 27, no. 9, pp. 1717-1726, Jan 2016.