**A PRIVACY AND SECURITY BASED ON DATA DEDUPLICATION IN PUBLIC CLOUD COMPUTING**

\*Dr. C. Premila Rosy1,

Head & Assistant Professor,

 PG & Dept., of CS,

Idhaya College for Women,

Kumbakonam,

Tamil Nadu, India

Ms. A. Fairosebanu2,

Assistant Professor of CS,

Idhaya College for Women,

Kumbakonam,

Tamil Nadu, India.

Ms. P. Aarthi3,

II – M.Sc., (CS),

Idhaya College for Women,

Kumbakonam,

Tamil Nadu, India.

Ms. S. Bhuvaneswari4,

II – M.Sc., (CS),

Idhaya College for Women,

Kumbakonam,

Tamil Nadu, India.

Ms. S. Nikeetha5,

II – M.Sc., (CS),

Idhaya College for Women,

Kumbakonam,

Tamil Nadu, India.

Corresponding Author \*Dr. C. Premila Rosy – premilarosy78@gmail.com

**ABSTRACT**

 Now a day, cloud is most emerging technology as one can store and manage their large amount of data. Outsourcing data to cloud for storage is becoming more attractive trend. In such type of data storage system sometime same kind of data may store by different users. Hence duplicate copies of data causes wastage of cloud storage space and it is inefficient task. Data de-duplication is one of the recent technologies / techniques in cloud storage in current market trends that avoid such data duplication caused by privileged as well as non-privileged user. It enables companies, organizations to save a lot of money on data storage, on bandwidth to transact data when replicating it offsite for disaster recovery. To achieve these goals, secure de-duplication and integrity auditing delegation techniques have been studied, which can reduce the volume of data stored in storage by eliminating duplicated copies and permit clients to efficiently verify the integrity of stored files by delegating costly operations to a trusted party, respectively.In our designed scheme, it is not necessary for users to be online for doing extra computation when data popularity changes and finally reduced unnecessary old data from cloud storage. The cloud can perform the task of authentications transforming to ensure that the cloud storage auditing still smoothly runs and proposed scheme is secure and efficient.

**Keywords:** Deduplication, Cloud Storage, Reduce Storage, Security

**I. INTRODUCTION**

Web based distributed computing has progressed computational power, which gives office like information putting away and put away information sharing. Distributed computing can likewise characterize as shared pool having different configurable processing assets, on-request organizes get to and the specialist co-op provisioned. Like coin, cloud has two sides; it is taken a toll saver however on opposite side significant concern is security. Alluring component of cloud is arrangement for colossal information stockpiling, which gives a few points of interest to the client, for example, portability, adaptable administration and cost separate.

 The first problem is integrity auditing. The cloud server is able to relieve clients from the heavy burden of storage management and maintenance. The most difference of cloud storage from traditional in-house storage is that the data is transferred via Internet and stored in an uncertain domain, not under control of the clients at all, which inevitably raises clients great concerns on the integrity of their data. These concerns originate from the fact that the cloud storage is susceptible to security threats from both outside and inside of the cloud, and the uncontrolled cloud servers may passively hide some data loss incidents from the clients to maintain their reputation. What is more serious is that for saving money and space, the cloud servers might even actively and deliberately discard rarely accessed data files belonging to an ordinary client. Considering the large size of the outsourced data files and the clients’ constrained resource capabilities, the first problem is generalized as how can the client efficiently perform periodical integrity verifications even without the local copy of data files.

The second problem is secure deduplication. The rapid adoption of cloud services is accompanied by increasing volumes of data stored at remote cloud servers. Among these remote stored files, most of them are duplicated: according to a recent survey by EMC, 75% of recent digital data is duplicated copies. This fact raises a technology namely deduplication, in which the cloud servers would like to deduplicate by keeping only a single copy for each file (or block) and make a link to the file (or block) for every client who owns or asks to store the same file (or block). Unfortunately, this action of deduplication would lead to a number of threats potentially affecting the storage system, for example, a server telling a client that it (i.e., the client) does not need to send the file reveals that some other client has the exact same file, which could be sensitive sometimes. These attacks originate from the reason that the proof that the client owns a given file (or block of data) is solely based on a static, short value (in most cases the hash of the file). Thus, the second problem is generalized as how can the cloud servers efficiently confirm that the client (with a certain degree assurance) owns the uploaded file (or block) before creating a link to this file (or block) for him/her.

**A. Integrity Auditing**-The first design goal of this work is to provide the capability of verifying correctness of the remotely stored data. The integrity verification further requires two features:

1) Public verification, which allows anyone, not just the clients originally stored the file, to perform verification;

2) Stateless verification, which is able to eliminate the need for state information maintenance at the verifier side between the actions of auditing and data storage

**B. Secure Deduplication**- The second design goal of this work is secure deduplication. In other words, it requires that the cloud server is able to reduce the storage space by keeping only one copy of the same file. Notice that, regarding to secure deduplication, our objective is distinguished from previous work [3] in that we propose a method for allowing both deduplication over files and tags.

**C. Cost-Effective**-The computational overhead for providing integrity auditing and secure deduplication should not represent a major additional cost to traditional cloud storage, nor should they alter the way either uploading or downloading operation.

**II. LITERATURE REVIEW**

Keelveedhi et al. [1] designed the DupLESS system in which clients encrypt under file-based keys derived from a key server via an oblivious pseudorandom function protocol.

Bellare et al. [2] formalized this primitive as message-locked encryption, and explored its application in space-efficient secure outsourced storage.

Abadi et al. [3] further strengthened Bellare et al’s security definitions by considering plaintext distributions that may depend on the public parameters of the schemas. Regarding the practical implementation of convergent encryption for securing deduplication.

**III. PROPOSED SYSTEM**

 In this project, aiming at achieving data integrity and deduplication in cloud, we propose two secure system. This method introduces an auditing entity with maintenance of a Map Reduce cloud, which helps clients generate data tags before uploading as well as audit integrity of data having been stored in cloud. Besides supporting integrity auditing and secure deduplication enables the guarantee of file confidentiality. Firstly, the amount of exchanged data to create the replicas could be huge and could overload the network. Secondly, the replicas that are not yet created or are in the process of being created are unavailable, and thus cannot process clients’ requests. Indeed, the management tool that directs user requests to the appropriate locations of data should have an updated view of all replica placements. In HQFR, the new placement is used to direct clients’ requests, even before the migration finishes. That is, the management tool becomes oblivious to the old placement of replicas. Therefore, some client requests may be directed to the new placement, even if some replicas haven’t yet wholly arrived at their final destination, thus negatively impacting availability. Moreover, to ensure availability of data during migration, the management tool limits the number of migrating replicas of each data for a time interval. A method of directly auditing integrity on encrypted data and finally reduced unnecessary old data from cloud storage.

**A. Secure Deduplication**

 Deduplication is a technique where the server stores only a single copy of each file, regardless of how many clients asked to store that file, such that the disk space of cloud servers as well as network bandwidth are saved. However, trivial client side deduplication leads to the leakage of side channel information. For example, a server telling a client that it need not send the file reveals that some other client has the exact same file, which could be sensitive information in some case.



**Figure 1 : System Architecture**

**IV. METHODOLOGY**

**A. HQFR Algorithm**

As the name indicates High QoS First Replication algorithm. The main thing is that we are considering the QoS requirement from the aspect of request information and its access time only. In cloud the data is divided into 64MB data blocks. There are two numbers of copies of data block other than the original one. And that two copies are stored on different Data Nodes or different data racks. And the Name Nodes keeps track of all the replicas other than original copy and they mounted on different data racks to avoid rack failure.

**Algorithm** (Redundancy algorithm)

The mapper emits an intermediate key-value pair for each word in a document.

The reducer sums up all counts for each word.

1: class Mapper

2: method Map(docid a, doc d)

3: for all term t ∈ doc d do

 4: Emit(term t, count 1) 1: class Reducer 2: method Reduce(term t, counts [c1, c2, . . .]) 3: sum ← 0 4: for all count c ∈ counts [c1, c2, . . .] do

5: sum ← sum + c

 6: Emit(term t, count sum)

**V. CONCLUSION**

They are focusing to achieve both data integrity and data de-duplication in cloud, we propose reducing unnecessary data and auditing entity with maintenance of a Map Reduce cloud, which helps the clients to generate data tags before uploading and audit the integrity of data having been stored in cloud. It’s preventing the leakage of side channel information in deduplication. We compared with the existing work; the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases. It’s an advanced construction motivated by the fact that customers always want their data to be encrypted before uploading, and allows for integrity auditing as well as secure de-duplication directly on encrypted data. As further work, improvements can easily be done since the coding is mainly structured or modular in nature. In the system can change the existing modules or adding new modules can append improvements.

**REFERENCES**

[1] S. Keelveedhi, M. Bellare, and T. Ristenpart, “Dupless: Serveraidedencryption for deduplicated storage,” in *Proceedings of the22Nd USENIX Conference on Security*, ser. SEC’13. Washington,D.C.: USENIX Association, 2013, pp. 179–194. [Online]. Available:

https://www.usenix.org/conference/usenixsecurity13/technicalsessions/presentation/bellare

[2] M. Bellare, S. Keelveedhi, and T. Ristenpart, “Message-locked encryptionand secure deduplication,” in *Advances in Cryptology – EUROCRYPT2013*, ser. Lecture Notes in Computer Science, T. Johanssonand P. Nguyen, Eds. Springer Berlin Heidelberg, 2013, vol. 7881, pp.296–312.

[3] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev,“Message-locked encryption for lock-dependent messages,” in *Advancesin Cryptology – CRYPTO 2013*, ser. Lecture Notes in Computer Science,R. Canetti and J. Garay, Eds. Springer Berlin Heidelberg, 2013, vol.8042, pp. 374–391.

[4]Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, ‘‘Enabling public auditabil- ity and data dynamics for storage security in cloud computing,’’ IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847–859, May 2011, doi: [10.1109/TPDS.2010.183.](http://dx.doi.org/10.1109/TPDS.2010.183)

[5]J. Li, J. Li, D. Xie, and Z. Cai, ‘‘Secure auditing and deduplicating data in cloud,’’ IEEE Trans. Comput., vol. 65, no. 8, pp. 2386–2396, Aug. 2016, doi:[10.1109/TC.2015.2389960.](http://dx.doi.org/10.1109/TC.2015.2389960)

[6]X. Jia and J. Zhou, ‘‘Leakage resilient proofs of ownership in cloud storage, revisited,’’ in Applied Cryptography and Network Security. Lausanne, Switzerland: Springer, 2014, pp. 97–115, doi:[10.1007/978-3-319-07536-5\_7.](http://dx.doi.org/10.1007/978-3-319-07536-5_7)

[7]H. Tian, Y. Chen, C.-C. Chang, H. Jiang, Y. Huang, Y. Chen, and J. Liu, ‘‘Dynamic-hash-table based public auditing for secure cloud storage,’’ IEEE Trans. Services Comput., vol. 10, no. 5, pp. 701–714, Sep./Oct. 2017, doi:[10.1109/TSC.2015.2512589.](http://dx.doi.org/10.1109/TSC.2015.2512589)

[8]J. Han, Y. Li, and W. Chen, ‘‘A lightweight and privacy-preserving public cloud auditing scheme without bilinear pairings in smart cities,’’ Comput. Stand. Interfaces, vol. 62, pp. 84–97, Feb. 2019, doi: [10.1016/j.csi.2018.08.004.](http://dx.doi.org/10.1016/j.csi.2018.08.004)

[9]M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, ‘‘Secure data deduplication,’’ in Proc. StorageSS, Alexandria, VA, USA, 2008, pp. 1–10, doi:[10.1145/1456469.1456471.](http://dx.doi.org/10.1145/1456469.1456471)

[10]R. Di Pietro and A. Sorniotti, ‘‘Boosting efficiency and security in proof of ownership for deduplication,’’ in Proc. ASIACCS, Seoul, South Korea, May 2012, pp. 81–82, doi:[10.1145/2414456.2414504.](http://dx.doi.org/10.1145/2414456.2414504)