DEDPULICATION IN CLOUD WITH SECURE ACCESS CONTROL

DR.CH.V.PH*ANI KRISHNA*
Professor
Department of CSE
Teegala Krishna Reddy Engineering College,Hyderabad

W.VAISHNAVI TARA
Department of CSE
vaishnavitara.wadepally@gmail.com
Teegala Krishna Reddy Engineering College, Hyderabad

ADRIAN WINCY
Department of CSE
adrianwincy@gmail.com
Teegala Krishna Reddy Engineering College, Hyderabad

D.AADITYAA REDDY
Department of CSE
adithyareddyvirat18@gmail.com
Teegala Krishna Reddy Engineering College,Hyderabad

Abstract- Distributed computing gives another method for administration by offering different assets over Internet. One of the essential administrations given by cloud benefit is information stockpiling. So as to save the protection of clients, these information are put away in cloud in a scrambled shape. Deduplication ends up critical and a testing errand when the information is put away in scrambled shape, which additionally prompts intricacy in putting away extensive information and handling in cloud. A customary deduplication strategy does not take a shot at scrambled information. Existing arrangements accessible for deduplicating encoded information has different security issues. They doesn't give get to control and repudiation as far as capacity. Thus, the deduplication plans are not for the most part conveyed practically speaking. In this paper, we propose a system to deduplicate encoded information put away in cloud dependent on access control along these lines evading repetitive capacity. It coordinates cloud information deduplication with access control. The aftereffect of our plan demonstrates prevalent proficiency and has potential for functional arrangement on account of enormous information stockpiling.

Index Terms– Deduplication; Encrypted data; Secured Access Control; Cloud computing

I.INTORDUCTION

Cloud computing provides various services by rearranging the resources over the Internet. The important cloud service is data storage.In order to preserve the security of these data, they are often stored in an encrypted form. Encrypted data create new challenges for cloud deduplication which Becomes crucial for big data storage and processing in cloud. A traditional deduplication scheme does not work on encrypted data. Therefore in this project we introduce a scheme to deduplicate encrypted data in could based on ownership to deduplicate multiple copies of same data. We aim to solve the issues in deduplication that are being faced by data holders by providing privacy for accessing the file. The results show superior efficiency and effectiveness of the scheme for practical deployment in cloud. The contributions of this paper can be summarized as follows We propose methods to save cloud storage without revealing the privacy of data holders by providing a scheme to deduplicate and manage encrypted data. The scheme manages data deduplication with data sharing even in the absence of the data holder while preserving their privacy. We combine cloud data deduplication with data access control in a simple way.

II. LITERATURE SURVEY

In previous deduplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized deduplication system, each user is issued a set of privileges during system initialization. The overview of the cloud deduplication is as follow:

POST-PROCESS DEDUPLICATION

With post-process deduplication, new data is first stored on the storage device and then a process at a later time will analyse the data looking for duplication. The benefit is that there is no need to wait for the hash calculations and lookup to be completed before storing the data thereby ensuring that store performance is not degraded. Implementations offering policy-based operation can give users the ability to defer optimization on "active" files, or to process files based on type and location. One potential drawback is that you may unnecessarily store duplicate data for a short time which is an issue if the storage system is near full capacity.

IN-LINE DEDUPLICATION

This is the process where the deduplication hash calculations are created on the target device as the data enters the device in real time. If the device spots a block that it already stored on the system it does not store the new block, just references to the existing block. The benefit of in-line deduplication over post-process deduplication is that it requires less storage as data is not duplicated. On the negative side, it is frequently argued that because hash calculations and lookups takes so long, it can mean that the data ingestion can be slower thereby reducing the backup throughput of the device. However, certain vendors with in-line deduplication have demonstrated equipment with similar performance to their post-process deduplication counterparts. Post-process and in-line deduplication methods are often heavily debated.

SOURCE VERSUS TARGET DEDUPLICATION

Another way to think about data deduplication is by where it occurs. When the deduplication occurs close to where data is created, it is often referred to as "source deduplication." When it occurs near where the data is stored, it is commonly called "target deduplication." Source deduplication ensures that data on the data source is deduplicated. This generally takes place directly within a file system. The file system will periodically scan new files creating hashes and compare them to hashes of existing files.The deduplication process is transparent to the users and backup applications. Backing up a deduplicated file system will often cause duplication to occur resulting in the backups being bigger than the source data. Target deduplication is the process of removing duplicates of data in the secondary store. Generally this will be a backup store such as a data repository or a virtual tape library.

ENCRYPTION OF FILES

Here we are using the common secret key *k* to encrypt as well as decrypt data. This will use to convert the plain text to cipher text and again cipher text to plain text. Here we have used three basic functions,

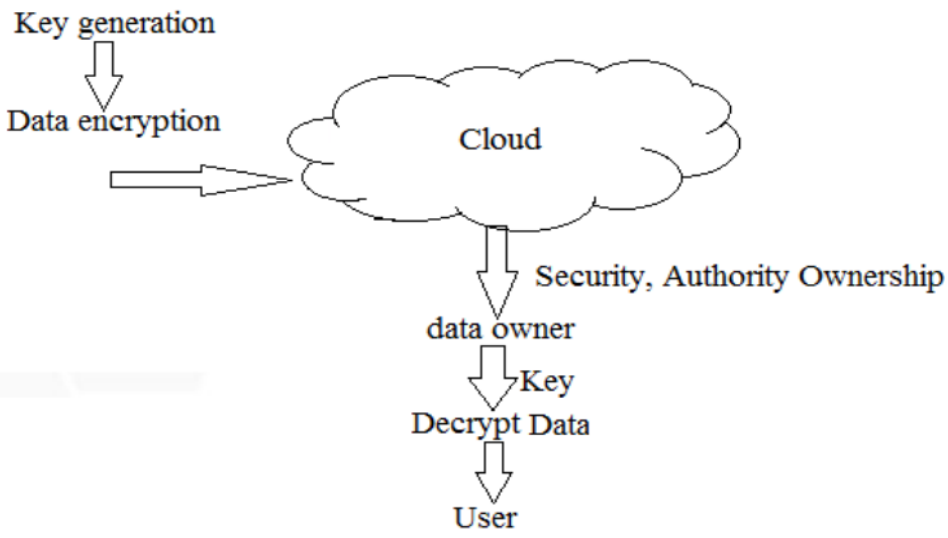
KeyGenSE: *k* is the key generation algorithm that generates κ using security parameter 1.

EncSE (k, *M*): *C* is the symmetric encryption algorithm that takes the secret κ and message *M* and then outputs the ciphertext *C*;

DecSE (k, *C*): *M* is the symmetric decryption algorithm that takes the secret κ and ciphertext *C* and then outputs the original message *M*.

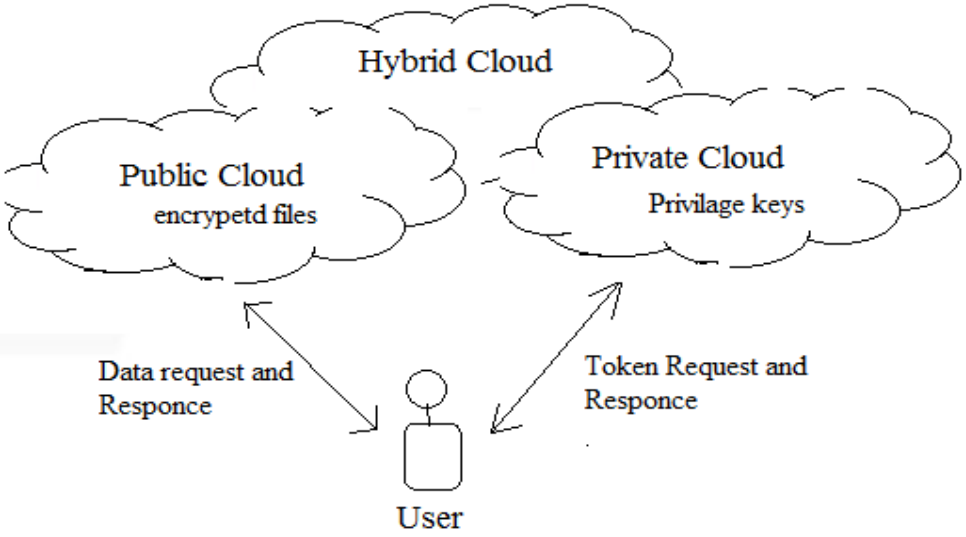
CONFIDENTIAL ENCRYPTION

It provides data confidentiality in deduplication. A user derives a convergent key from each original data copy and encrypts the data copy with the convergent key. In addition, the user also derives a tag for the daa copy, such that the tag will be used to detect duplicates.



PROOF OF DATA

The user have to prove that the data which he want to upload or download is its own data. That means he have to provide the convergent key and verifying data to prove his ownership at server.



III. EXISTING SCHEME

The existing solutions for deduplication that are available cannot flexibly provide access control and revocation at the same time. Most of the solutions cannot ensure reliability, security and privacy for the data. In real time, it is hard to allow a data holder to manage deduplication due to a number of reasons. At first, the data holders may not be always available for managing the data. In case if the perform deduplication manually, it could cause delay in storing the data. Second, deduplication becomes complicated in terms of computations involved in the process. Third, it may violate the privacy of data in the process of analyzing the deduplicated data. Forth, a shared key is provided to the users to access the same data. Whenever a user removes the data from his account, the keys must be reassigned to the remaining users who have access to the data. This reassignment which is a complex task, if not carried out properly, the removed user may have illegal access to that data.

IV. PROPOSED SCHEME

In this paper, we propose an enhanced technique of deduplicating the various kinds of data that can be stored in cloud. Our proposed scheme consists of the following steps

1) Generate hash for given data
2) Check if file exists,
3) Provide access without uploading.
4) Otherwise, Encrypt and store the given data with hash as key. Store the encrypted hash with individual keys.
5) On deletion, Revoke access by removing the individual key
6) If a file is not used by anyone, then remove the file

Using the SHA-256 algorithm we generate the hash value for the given data. The hash value is encrypted using AES algorithm and the encrypted hash value in stored in the cloud storage.

V. SYSTEM ARCHITECTURE

A. *Deduplication while uploading the file*

Whenever the user uploads the file F into the cloud, the hash value is generated for that file HF = H (F). The hash value is used as a key to encrypt the file. The hash value will be unique for every file (a small change in one bit of the file will result in multiple bit changes in its hash value generated. This is known as avalanche effect). A random key will be generated which is used to encrypt the hash value of the file. Encrypted hash will be stored in the database and generated key will be given to the user. Original file name will be stored in the database and a hash will be generated again for the previously generated hash X = H (HF). If a file named X already exists in the data storage, the file will not be stored. The user will be provided access to the file from above steps and upload count will be incremented by one. Otherwise, file will be renamed as the X value that is generated and stored in the cloud with a new upload count as one.

B. *Retrieval of file*

The key provided to the user during encryption is used to decrypt the encrypted hash that is stored in the database. The decrypted hash will be hashed again and it is used as the file name to search for the particular file in the data storage. Then the file will be renamed to its original name saved in the database.

C.Removal of a file and access revocation

*I*f a data holder deletes the file from data storage, the upload count is decremented by one and the encrypted hash value provided to the user will be removed for that file. The *file will be available as long as at least one user is available to access the file.*

D.Database Schema

There are three tables are required for deduplicationg the data in a secu*re manner*. (user information table, file information table, user file mapping table). Eventhough there is a table, file cannot be identified or decryped since altogether stored in an encrypted form. So that the security and privacy of the users is improved.

1. User Information Schema
2. User File Mapping Schema
3. User File Mapping Schema

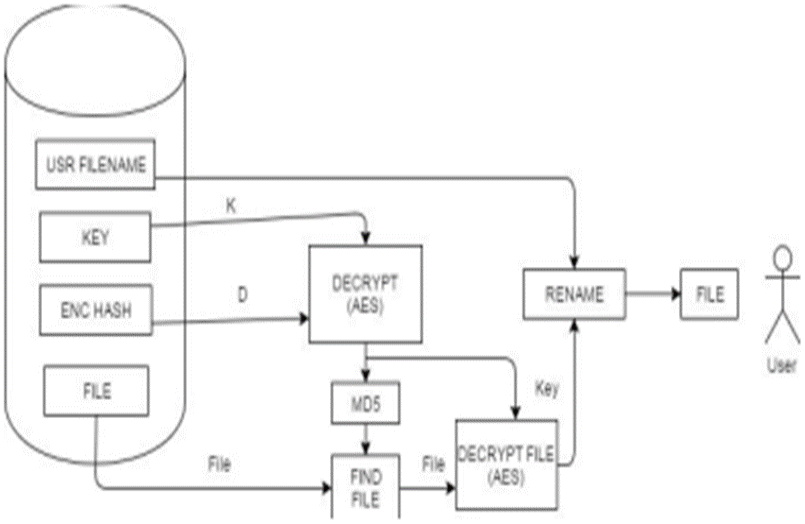


Figure : System Architecture

VI.CONCLUSION

Managed encrypted data with deduplication is important and significant in practice for achieving a successful cloud storage service, especially for big data storage. In this paper, we proposed a scheme to manage the encrypted files in a cloud with deduplication based on ownership. Our scheme can flexibly support data update and sharing with deduplication. Encrypted data can be securely accessed only by authorized data holders can obtain the symmetric keys used for data decryption.

VII.REFERENCES

[1] A secure data deduplication framework for cloud environments, authors: Fatema Rashid, Ali Miri, Isaac Woungang
[2] Attribute-Based Storage Supporting Secure Deduplication of Encrypted Data in Cloud, authors: Hui Cui, Robert H. Deng, Yingjiu Li
[3] Achieving lightweight, time-specific and secure access control in cloud storage, authors: Yanchao Wang,Fenghua Li, Ben Niu
[4] Design and implementation of various file deduplication schemes on storage devices, authors: Yong-Ting Wu , MinChieh Yu , Jenq-Shiou Leu , Eau-Chung Lee,TianSong
[5] T. T. Wu, W. C. Dou, C. H. Hu, and J. J. Chen, "Service mining for trusted service composition in cross-cloud environment," IEEE Systems Syst. J., vol. PP, no. 99, pp. 1–12, 2014, doi:10.1109/ JSYST.2014.2361841.
[6] Liu, C. Yang, X. Y. Zhang, and J. J. Chen, "External integrity verification for outsourced big data in cloud and iot: A big picture," Future Generation Comput. Syst., vol. 49, pp. 58–67, 2015
[7] W. Tsai, C. F. Lai, H. C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," J. Big Data, vol. 2, no. 1, pp. 1–32, 2015, doi:10.1186/s40537-015-0030-3.
[8] L. F. Wei, et al., "Security and privacy for storage and computation in cloud computing," Inf. Sci., vol. 258, pp. 371– 386, 2014, doi:10.1016/j.ins.2013.04.028.
[9] "Deduplication on Encrypted Big Data in Cloud" by Zheng Yan, Senior Member, IEEE, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng, Fellow, IEEE .
[10] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," Inf. Sci., vol. 305, pp. 357–383, 2015, doi:10.1016/j.ins.2015.01.025.