Secure File Level Deduplication for Cloud Data Storage using Content Hash Keying

T. Sairam[1] Ra. K. Saravanan Guru[2] S. Vijay Kumar[3]
[1]M. Tech Student [2]Assistant Professor [3]Research scholar
[1,2,3]Department of Computer Science & Engineering
[1,2,3]SCSE, VIT University, Vellore

Abstract— Data Deduplication is the keynote technique used in the cloud storage. It is used to avoid the repetition of data and to save the wastage of bandwidth and storage space across the cloud environment. At the time of performing Deduplication, Security is the major concern. Hence we are implementing the CHSK (Content Hash Keying Technique) to provide the confidentiality of data at deduplication. This is the best method to provide security for the successful deduplication. This is known as Hybrid Architecture as this is summation of the both private and public cloud. Private cloud is used to provide the privilege keys and public cloud is used to generate the file token to the user. This is absolutely different from the existing traditional systems. Hence we are providing OWD (Owner of Data) protocol for Authorized Deduplication.

Key words: Regenerative Braking System, Conventional Braking System, Electric D.C. Generator

I. Introduction

A. *Deduplication Means:*

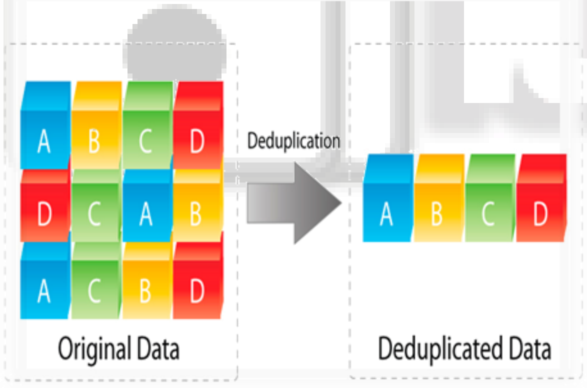

Fig. 1: Occurrence of Deduplication

Deduplication is a Technique to avoid redundancy of data stored in the cloud. Deduplication is used in many organizations to reduce the wastage of storage. This deduplication occurred in different models and ways.

B. *Types of Deduplication:*

1) *Approaches to Deduplication*

Data deduplication methods are segmented according to the requirement. In this way, we have two main data deduplication strategies: (1) File-level deduplication: In File level deduplication, which only a single copy of each file is stored. Depends on Hash value, It detects two or more files are there in the cloud. (2) Block-level Deduplication, in this model it divides into blocks, and keeps a single copy of it. Fixed chunk and variable chunk are the two types in it. In this paper, we are discussing about the file level deduplication. According to the architecture of the deduplication, there are two basic approaches. In the target-based approach deduplication is worked by the target data-storage device, while the client is unaware of any deduplication that might occur. This technology strengthens storage utilization, but Bandwidth is the one of the issue in it. On the other way, source based deduplication acts on the data at the client before it is transferred. Especially, the client software communicates with the backup server to check for the presence of files or blocks. Duplicates are replaced by pointers and the actual duplicate data is never sent over the network. Both storage and bandwidth utilization are the Most important concern in this paper(2).

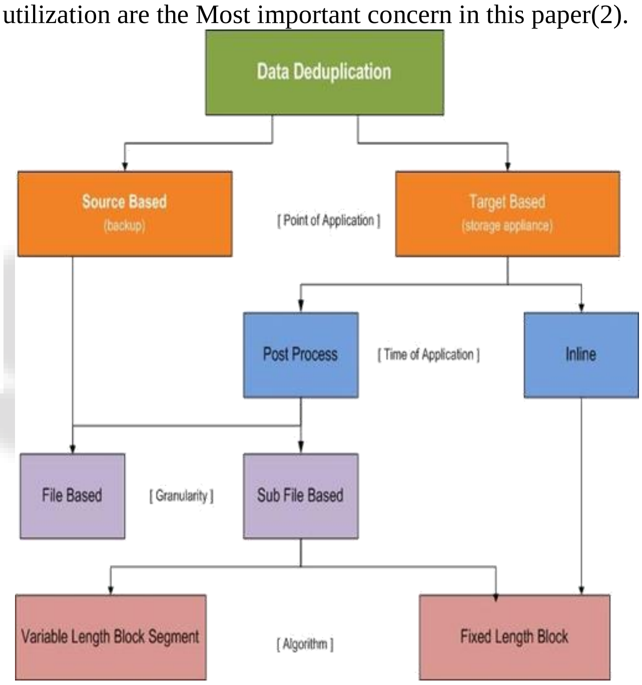

Fig. 2: Types of Deduplication

2) *Source versus target deduplication:*

It is the deduplication occurred at the time of creation. Deduplication occurred at a storage location, is called Target Deduplication. Source deduplication ensures that the data on the data source is deduplicated. It takes place directly within the file system. Here It is searching for the new files, if any and create hashes for that and compare the new files with the existing files. Whenever same hash is found then the file is terminated and the new file points to the old file. Unlike hard links however, duplicated files are fully considered as a separate entities and if one of the duplicated files is later modified, a copy of that file or changed block is created by using a system called Copy-on-write. 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 is known as a backup store such as a data repository or a virtual tape library. The most common forms of data deduplication implementations works by comparing chunks of data to detect duplicates. For that to happen, each chunk of data is assigned identification, calculated by the software,

Typically using cryptographic hash functions. In many implementations, the assumption is made that if the identification is identical, the data is identical, pigeonhole principle is one of the cause for cannot be true in all cases. Due to the other implementations do not assume that two blocks of data with the same identifier are identical, but actually verify that data with the same identification is identical. If the software either assumes that a given identification already exists in the deduplication namespace or actually verifies the identity of the two blocks of data, it's depending on the implementation, and then it will replace that duplicate chunk with a link. Once the data has been deduplicated, upon read back of the file, wherever a link is found, the system simply replaces that link with the referenced data chunk. The deduplication process is intended to be transparent to end users and applications.

3) Post-process deduplication:

In post process deduplication, deduplication performs before storing the data itself. Here there is no waste of time for hash calculations. Unnecessarily store duplicate data for a short time is the drawback of this post process deduplication, at the time of storage capacity is full, it causes a problem.

4) In-line deduplication:

In this process, hash calculations are created on target device only, whenever the data enters. 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 Advantage of in-line over post-process deduplication is that it requires less storage as data is not duplicated. On the negative side, it is frequently checked 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 are similar performance to their post-process deduplication counterparts. Post-process and in-line deduplication methods are most debated concepts in the deduplication.

Privacy Risks in Cloud Storage and Deduplication:

The entrusting data in the cloud is very high risk, since the data owner is usually free releasing control over your data. Yet, in case of reality, the number of users and applications are more than willing to hand over their data storage to a cloud provider. They have trust in the security of access control mechanism issues and integrity of the cloud provider it uses. Setting these issues aside, we found an additional threat the privacy implications of a cross-user deduplication. We demonstrate how deduplication in cloud storage services can be used as a side channel which reveals information about the contents of files of other users. In a different aspect scenario, deduplication used as a converting channel by which malicious software can communicate with its command and its control center, regardless of firewall settings at the attacked machine. We propose a simple mechanism by analyze these threats and then using the cross-user deduplication while heavily eradicating the risk of data confidentiality and leakage. More specifically, the proposed method is a mechanism stating rules by which deduplication is sometimes artificially turned off. We analyze the guarantees of this simple practice. This gives clients a guarantee that adding their data to the cloud has a very limited effect on what an adversary may learn about this data. Thus, it is possible thing to ensure clients of the privacy of their data. Therefore, traditional security mechanisms may not suffice due to unbearable computation or communication overhead. For example, to verify the confidentiality and integrity of data that is remotely stored, it is impractical to hash the entire data set. To this end, advanced protocols and strategies are expected[3].

II. RESEARCH OBJECTIVES

In our system we implement a project that includes the public cloud and the private cloud and also the hybrid cloud which is a combination of the both public cloud and private cloud. In general by if we used the public cloud we can't provide the security to our private data and hence our private data will be loss

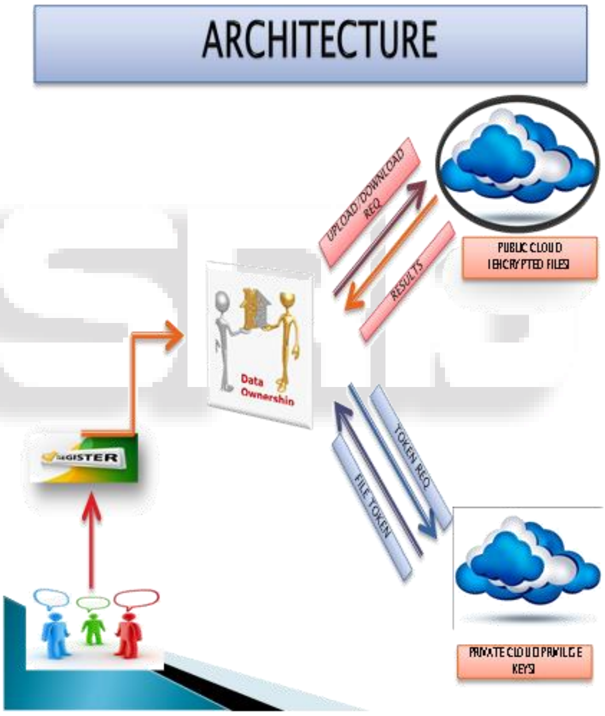

Fig. 3: Architecture of Hybrid Cloud

So that we have to provide the security to our data for that we make a use of private cloud also. When we use private clouds the greater security can be provided. In this system we also provide the data deduplication. This is used to avoid the duplicate copies of data. User can upload and download the files from public cloud but private cloud provides the security for that data. That means only the authorized person can upload and download the files from the public cloud. For that user generates the key and stored that key onto the private cloud, at the time of downloading user request to the private cloud for key and then access that Particular file. First if the user want to upload the files on the public cloud then user first encrypt that file with the content Hash key and then sends it to the public cloud at the same time user also generates the key for that file and sends that key to the private cloud for the purpose of security.

From private cloud its sends the privilege key with the combination of Content Hash Key, it generates FILETOKEN for that file. File token is used to upload the file in public cloud. This is used to avoid the duplicate copies of files which are entered in the public cloud. Hence it also minimizes the bandwidth. That means we require the less storage space for storing the files on the public cloud. In the public cloud any person that means the unauthorized person can also access or store the data so we can conclude that in the public cloud the security is not provided. In general for providing more security user can use the private cloud instead of using the public cloud. User generates the key at the time of uploading file and stores it to the private cloud. When user wants to downloads the file that he/she upload,. He/she sends the request to the public cloud. Public cloud provides the list of files that are uploads the many user of the public cloud because there is no security is provided in the public cloud. When user selects one of the file from the list of files then private cloud sends a message like enter the key! User has to enter the key that he generated for that file. When user enter the key the private cloud checks the key for that file and if the key is correct that means user is valid then private cloud give access to that user to download that file successfully. Then user downloads the file from the public cloud and decrypt that file by using the same content Hash Key which is used at the time of encrypt that file. In this way user can make a use of the architecture.

III. Working of Entities:

ROLE OF ENTITIES IN SYSTEM:

(1) SPCS
(2) Data User
(3) Private cloud
(4) Public cloud

A. *SPCS (service provider of cloud services):*

The purpose of this entity to work as a data storage service in public cloud. On the half of the user SPCS store the data. The SPCS eliminate the duplicate data using deduplication and keep the unique data as it is. SPCS entity is used to reduce the storage cost. SPCS handle abundant storage capacity and computational power. When user send respective token for accessing his file from public cloud SPCS matches this token with internally if it matched then an then only he send the file or cipher text with token, otherwise he send abort signal to user. After receiving file user use content Hash key to decrypt the file.

B. *Data User:*

User is an entity that wants to access the data or files from SPCS .User generate the key and store that key in private cloud. In storage system supporting deduplication, the user only upload unique data but do not upload any duplicate data, which may be owned by the same user or different users. Each file is protected by content Hash key and can access by only authorized person. In our system user must need to register in private cloud for storing token with respective file which are store on public cloud. When he wants to access that file he access respective token from private cloud and then access his files from public cloud. Token consist of file content F and content Hash key CF.

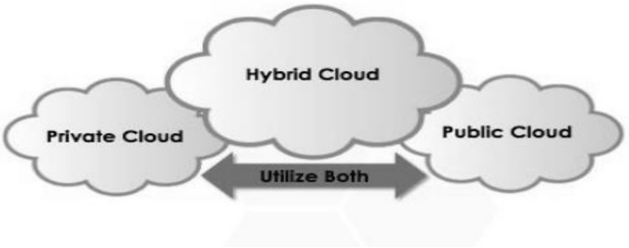

Fig. 4: Hybrid cloud

C. *Private Cloud:*

In general for providing more security user can use the private cloud instead of public cloud. User stores the generated key in private cloud. At the time of downloading system, it asks the key to download the file. User can not store the secrete key internally. For providing proper protection to key, we use private cloud. Private cloud only store the content Hash key with respective file. When user wants to access the key he first checks authority of user then provide key.

D. *Public Cloud:*

Public cloud entity is used for the storage purpose. User uploads the files in public cloud. Public cloud is similar as SPCS. When the user wants to download the files from public cloud, it will be asking the key which is generated or stored in private cloud. When the users key is match with files key at that time user can download the file, without key user can not access the file. Only authorized user can access the file. In public cloud all files are stored in encrypted format. If any chance unauthorized person hack our file, but it is not easy to decrypt the file and read the content.

IV. 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 data copy, such that the tag will be used to detect duplicates(4)

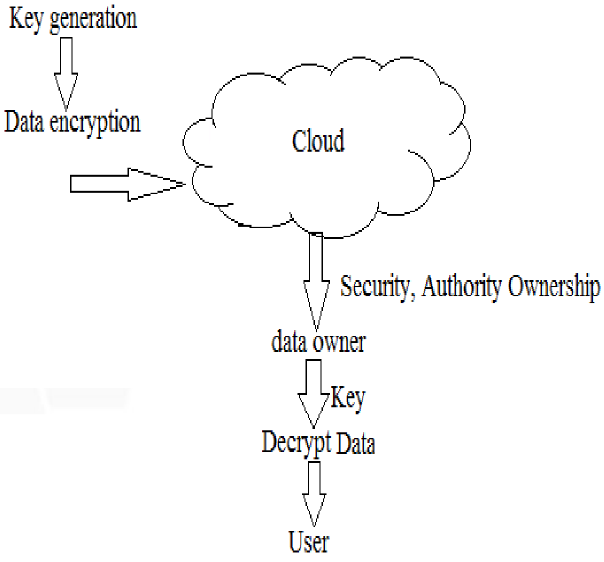

Fig. 5: Content Hash key Technique (CHSK)

A. Content Hash key Technique (CHSK):

It is the technique is used to encrypt and decrypt the data. Content Hash Key Technique is used to convert the original data into individual cipher text. Generation of cipher text is depends on the data in the file. When two same file having same data, at that time it will generates same cipher text for both the files. Here we represented as C=H (F), H is the Hash Function(5).

Here we are four primitive functions in this Technique

GK C(f) :Generation of key is occurred in this step. It maps an original data I into a Content Hash Key C

E (K, I) : This is symmetric Encryption Algorithm takes input as both Content Hash Key C and Data I as a Inputs and the convert into output as a Cipher text O

D (K, C): This is the Decryption Algorithm considers both Cipher Text O and Content Hash Key C as a inputs and produced output as a original data I

Tag Gen (G): T (G) is the tag generation algorithm that maps the original data copy I and outputs a tag T (G).

Notations Used In This Paper:

SPCS- Service Provider of Cloud Storage

OWD- Owner of Data

C_f - Compatible Encryption Key for File F

P_r - Privilege of user

P_s - Specified Privilege of File F

T_p^f – Token of a file with privilege P.

Process Occurred In This Application:

Before uploading file to cloud	While downloading file from cloud
Generate Hash key H_k	Get the Privilege key
Taking the privilege from Private cloud	Get the Hash key
By combining the Hash Key and Privilege of a user ,We are Generating The FILE TOKEN and Then file Uploaded	Decryption occurred and File Downloaded

Table 1: Difference Between Uploading and Downloading

- Tag (File) -It computes SHA-1 hash of the File as File Tag;
- Req. and gen. of Token (Tag, UserID) - It requests the Private Server for File Token generation with the File Tag and User ID;
- AuthorizedDupCheckReq (Token) - It requests the Storage Server for Duplicate Check of the File by sending the file token received from private server;
- TokenReqshare (Tag, {Priv.}) - It requests the Private Server to generate the Share File Token with the File Tag and Target Sharing Privilege Set;
- File Encrypt (File) - It encrypts the File with Convergent Encryption using 256-bit AES algorithm in cipher block chaining mode, where the Content Hash key is from SHA-256 Hashing of the file;
- FileUploadReq (FileID, File, Token) – It Uploads the File Data to the Storage Server if the File is Unique and updates the File Token stored.

Our implementation of the **Private Server** includes Corres-ponding request handlers for the token generation and maintains a key storage with Hash Map.

- Token Gen (Tag, UserID) - It loads the associated Privilege keys of the user and generate the token With HMAC-SHA-1 algorithm; and

V. OWNER OF DATA (OWD)

This is the one of the protocol to verify the user genuineness as it verifies data copy belongs to the user or not. It is implemented by using an Interactive algorithm. This is the protocol verification occurred between user and verifier. Here user is known as data user or prover and verifier is known as storage server. Here storage server derives a short value of $\phi(I)$ from a data copy I. To prove the ownership of data, prover sends ϕ' to the verifier for the verification. Through this owner of data, we are providing enhanced security to the user data. For example, if any user wants to attach the file as a

reference to the existing original file, definitely user need to prove the owner of data protocol by providing the privilege key and content hash key. The main advantage of this system is security, for example if any unauthorized person knows the content hash key, there is no use at all. Because content Hash key also encrypted, without knowing the privilege key form the private server, it is not a possible thing to misuse of data(8).

VI. PERFORMANCE ANALYSIS

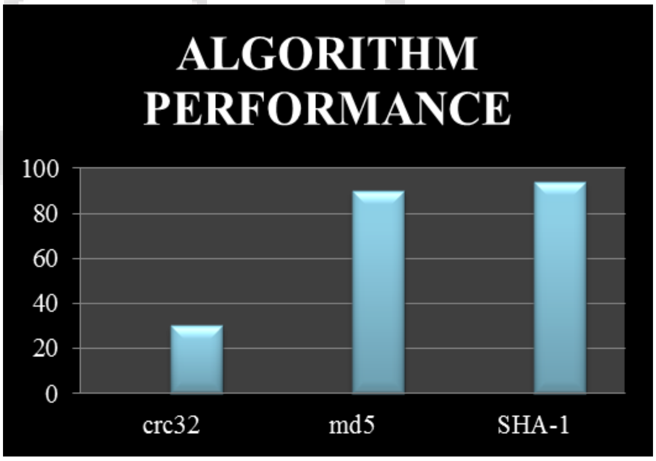

Fig. 6: Performance Graph

The performance evaluated based on the comparison of Md5 and sha1 and crc32 algorithm.sha-1 is the most accurate algorithm, when we compare with remaining two.

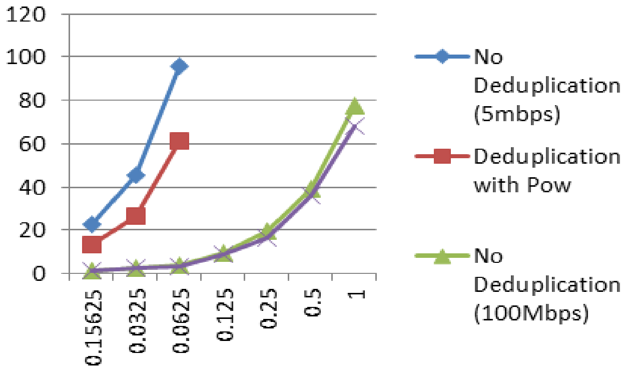

Fig. 7: Performance Graph

Benefits of deduplication with OWD Comparisons between systems that perform deduplication with OWD to those that always transmit the data over the 5Mbps and 100Mbps networks respectively .We consider a relatively conservative workload in which only 50% of the data is deduplicated and we pay the overhead of the reduction phase of the OWD scheme for remaining 50% of the files.

VII. CONCLUSION

By using this model, performing deduplication is much secured. There is no doubt about the security of data. Content Hash Keying Technique and owner of data provides enhanced security to this model. Authorized duplicate is performed by using this system.

VIII. FUTURE SCOPE

In the future for this paper go for sha-2 algorithms for encryption in cloud environment and use more number of attributes for more security purpose then the cloud environment can get accurate results and enhanced options for the usage.

REFERENCE

[1] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013.

[2] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.

[3] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.

[4] M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.

[5] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[6] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In ICDCS, pages 617–624, 2002.

[7] D. Ferraiolo and R. Kuhn. Role-based access controls. In 15th NIST-NCSC National Computer Security Conf., 1992.

[8] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Pele g. Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.