ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 7, July 2016**

# Secure Data Deduplication in Cloud Storage Using Twin Cloud and Client Side Encryption

Pawar P.R, Prof. Arti Mohanpurkar

Dept. of Computer, Dr.D.Y.Patil School of Engineering and Technology, Savitribai Phule Pune University, India.

Dept. of Computer, Dr. D.Y.Patil School of Engineering and Technology, Savitribai Phule Pune University, India.

**ABSTRACT:** Cloud storage has become popular trend in recent days, because most of the cloud providers not only provide huge storage but data security also. Most surveys clearly shows that duplicated data occupies most of the storage as duplicate files or data covers nearly half of storage on cloud. In order to manage such a huge data, deduplication is introduced. Deduplication performs scheme for storage management, it instead of storing multiple copies of data which are exactly identical, stores only one copy. Previously, in cloud environment only one cloud was used but hybrid cloud approach ensures client with good deal of data security and privacy.

**KEYWORDS:** Cloud Storage, Data deduplication, convergent encryption, Proof of Ownership, Data Confidentiality.

## I. INTRODUCTION

Cloud storage has become so familiar service model thatlets clients and companies to outsource the data to remote cloud service providers at a low cost. As huge amount of growth of contents and resources on line, cloud storage looks forward for better utilization of storage resources. So, it became prime need of avoiding such data redundancy for better storage management [1]. To avoid such data redundancy, few mechanisms were tried but data deduplication became successful and popular in very less time. As a result, Data deduplication [1] plays important role as it saves cost for cloud storage[14], by storing only one physical copy of duplicate data, and provide pointers to clients of that duplicate or redundant copies. In cloud computing there is need to manage data, hence deduplication [1] has attracted more and more attention recently in the backup process. Deduplication not only saves space but also bandwidth which is required for uploading or downloading data. For recent cloud storage services biggest challenge is to handle exponentially increasing data. By the year 2020, 40 trillion gigabytes level is expected to be crossed [4], according to the analysis report of IDC. And deduplication came for the rescue. Data deduplication can be viewed as data compression technique [2]. Deduplication eliminates redundant data as it keeps only one physical copy instead of storing multiple. For that reason, if a user wants store a file, cloud service provider checks weather the file is already saved previously,
if yes, cloud provider will update the owner list of that file by adding user to it. De- duplication can be done in two ways, first is file-level deduplication in which whole file is considered and another is block-level deduplication in which similarities across the files are considered. Where File-level deduplication refers to the whole file whereas block-level deduplication refers to the fixed or variable size data block. With the help of various encryption techniques, deduplication
can be made secure. If we consider traditional encryption, several users encrypt their original file or copy of data with their private keys, but for identical data though their content is exactly same as plain text, ciphertexts generated from different users are not identical so deduplication can not be applied for traditional encryption techniques. Convergent encryption [3] provides a solution to the issue raised by traditional encryption by implementing data confidentiality [13]. Convergent encryption, a cryptosystem that generates same ciphertext, if plain text is same, no matter what encryption keys are used [15]. Convergent encryption encrypts/decrypts a data with a special key called as convergent key. From original data copy hash is computed and key is formed or generated [3]. After convergent key generation and data encryption takes place with those keys, convergent keys are stored by users and files are uploaded on cloud. Now deduplication can be applied on the ciphertexts. At thetime of downloading, the ciphertexts are decrypted with their convergent keys. The original data copy or file of user is first encrypted with the help of a convergent key and this

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

convergent key is again encrypted by a user's private key which resides locally at client. The private key can be used to decrypt the encrypted convergent keys. So, user is required to store only master key and data about those private keys known as metadata. Before going to implementation it is required to understand basic concept and various types of data deduplication. Thus, convergent encryption allows the cloud to perform deduplication on the ciphertexts and the proof of ownership prevents the unauthorized user to access the file.

## II. DATA DEDUPLICATION AND CONVERGENT ENCRYPTION

### A. An overview

Data deduplication has many forms. Different organizations may use multiple deduplication strategies. It is needed to take backup and issues corresponding to backups into account. Data deduplication have following types.

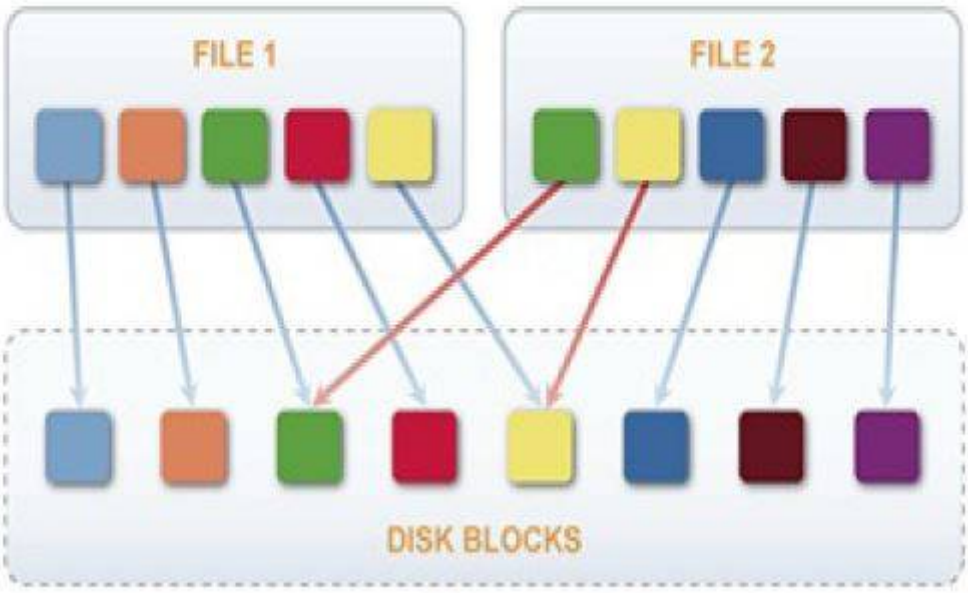


**Fig. 1 Data deduplication**

### B. Data Deduplication Types

1) File-level deduplication: File level deduplication checks certain attributes of given files against the specific index, if the file is unique, that copy is stored as single-instance storage, if file is not unique then only pointer is updated. Single physical copy of file is saved instead of keeping same and several duplicate copies of files which will result in storage redundancy.

2) Block-level deduplication: In this, first file is being divided into parts or segments which are called as blocks or chunks, those chunks of data are checked for redundancies or previously stored information. Chunk has size which can vary. Each chunk is assigned with an identifier, which is generated by hash.. The comparison takes place between that identifier and particular unique Id. If ID is already present, it indicates that data duplicate. For that only a pointer is saved or updated.If ID is found to be new which means block is unique, it isstored and index is updated also.

### C. Traditional Encryption

Data deduplication, which has been proved to be essential for data storage, some security issues arises due to it. Specifically, in case of traditional encryption several users encrypt data using respective private keys [15]. Thus, even though files are exactly identical their ciphertext will be different and deduplication cannot be done as after traditional encryption generated files are not exactly same, it means no scope for deduplication.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 7, July 2016**

**D. Convergent Encryption**

Unfortunately, deduplication and encryption cannot walk hand in hand. Main intention of deduplication is to search duplicate data chunks and store one copy. But the result of encryption of two identical data segments becomes indistinguishable. This means that if encryption is traditional, the cloud service storage provider cannot implement deduplication since two identical data segments will be different after encryption. Also, id data is not encrypted or uploaded as it is, confidentiality cannot be ensured and as cloud is curious, there would be great challenge of insiders from storage providers. Convergent encryption [5] provides data confidentiality while making deduplication possible even though different users are using different keys to encrypt data. A file is encrypted with a convergent key. That convergent key is generated by the hash value of file. When key is generated from file and data is encrypted with that convergent key, users stores the keys locally and upload the ciphertext which is generated to the public cloud. There will be same convergent keys for identical data copies of files which generate same ciphertext, because encryption process is deterministic i.e. it produces the same ciphertext for a given plain text and key, even over separate executions of the encryption algorithm. Proof of Ownership (PoW)[8] protocol is used to make every access authorized and secure. In this protocol, user must provide the proof that user holds the file when a duplicate file is found, on the basis of response/verify manner. After the proof is submitted, users are provided with pointer to access that file and no need to upload that file. Then pointers will be used to download the file from server, file downloaded is always encrypted and only data owner of that file can decrypt it with the help of respective convergent keys.

## III. CLIENT SIDE ENCRYPTION

Client-side encryption provides facility for client to choose encryption key and let client to save that key at client side, cloud provider has nothing to do with client's data. It is highly recommended that any data or file must be encrypted before uploading on cloud. Client-side encryption creates an encryption key that is available to the client only and not for service provider. Due to this it become difficult or impossible for curious service providers to decrypt hosted data. Clientsideencryption ensures high level privacy by creating zero knowledge applications. Client-side encryption is widely recognized. Not because of its robustness but due to data security strategy.It eliminates the possibility of curious cloud provider to watch client's data. It ensures that data and files that are stored in the cloud can only be viewed on the client side of the exchange. This prevents data loss and the unauthorized disclosure of private or personal files, providing increased peace of mind for both personal and business users. Hence, in client side encryption encryption key is generated and stored locally and there is no scope for cloud provider to watch client's data. Client Side Encryption has great significance
Because giants also not providing it.

## IV. SYSTEM DESIGN

**A. Design Goals**

Privacy preserving is the measure issue which is focused in this paper while providing deduplication in cloud storage. So, new system will give emphasis on following things.

1) Differential Authorization: Each user is authorized first and then get token for his/her file and that token is used for checking duplicate checking. Token is given on the basis of privileges. So, no user can generate such token when he failed to submit respective privileges. User must request private cloud in order to request token.

2) Authorized Duplicate Check: Only those users are allowed to upload or download file who have privileges with token received from private cloud authorization.

3) Role Based Access Controls: In role based access controls (RBAC)[11][12] permissions are associated with roles, and users are made members of appropriate roles. This greatly simplifies management of permissions. Roles are closely related to the concept of user groups in access control. However, a role brings together a set of users on one side and a set of permissions on the other, whereas user groups are typically defined as a set of users only.

**B. Modules**

Proposed system has three basicmodules [9].User authorization is done by private server and duplicate check and storage is done on public cloud.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

1) S-CSP: Data storage is done on public cloud which is also known as S-CSP. The S-CSP allows through various storage techniques, data uploading by users. S-CSP avoids redundancies via deduplication which stores only one copy of files. In this way storage cost is reduced. It is required that S-CSP is always active and can manage huge data with computational functionalities.
2) Data users or Client: A user is module which tries tooutsource or upload data to public cloud and download the data. Upload bandwidth is saved because only one physical copy is uploaded. The convergent key provides protection for file and that key is in turn encrypted by user with their private keys.
3) Private Cloud: This module is introduced for users to use cloud storage securely. As public cloud is honest but also it is curious. So is not fully trusted, for that reason private cloud works as an interface for user/client and the storage server. Privilege Key management is done by the private cloud, who provides tokens in response to the file token requests. Hence due to interface provided with the help of the private cloud, it will be easy for user to upload files and submit queries in secure manner.

## V. PROPOSED SYSTEM

### A. Current Systems

While implementing proposed system it is sure that Convergent encryption [3], [9] will take care of Data confidentiality for users in order to provide data deduplication. Convergent key is derived from file itself which is SHA1 OR MD5 and encrypts that file or data with the convergent key. Tag is also derived which is used while token generation. It is clear that if two data copies are exactly same or identical, it means their tags are same or identical. Firstly, user sends a tag to storage server or public cloud. On public cloud it is checked whether given tag is already stored or not. It is to note that, convergent key and the tag are derived independently. Thus data confidentiality is not compromised. After the encryption, the encrypted file along with its tag will be stored on cloud. With the help of the following primitive functions convergent
encryption is defined as shown:

generateKeyCE(F) =) Key This function generates convergent key form file F or copy of data.
_ encryptFileCE(Key,F) =) C This function generate cipher by symmetrically encrypting file F, taking convergent key and data F itself as parameters. _ decryptFileCE(Key,C) =) F This function is used to decrypts ciphertext. Original file F is then recovered. _ generateTag(F) =)T(F) This function generates tagfrom original file or data copy F.

### B. Previous attempt

Before introducing hybrid cloud architecture for construction of differential deduplication, it was attempted in straightforward manner without using hybrid cloud. In this straightforward construction main thing is to assign respective privilege keys to users, who will get file tokens and look for performing the duplicate check. Duplicate check is done using privilege keys. Suppose there are N users in the system and P be the set of privileges as P =f p1,p2,p3,..pn g in universe. Private key kp will be selected for each privilege p in P. PU, set of privilege for user U, he will be assigned fkpi gpi _ PU which is nothing but the set of privilege keys.
1) File Uploading: While doing file uploading and sharing, file token is computed by user and sent to S-CSP as _0F = TagGen(F) for all p _ PF . _ If duplicate file found by the server, the user runs proof of ownership protocol for claiming ownership of given file. Only after providing required information, then and then pointer is given to user, for accessing the file. _ Also, if file is found to be unique, encrypted file is computed by user as CF = encryptFile CE (Key, F) using convergent key kF = generateKeyCE(F). The key and token is uploaded i.e. (CF , _0 f,p) to the cloud.
2) File Retrieving: In order to download a file F, user is needed to send a request for access along with the name of file to the storage server/S-CSP. When request reaches to server, it checks whether user has privilege to access the file. If not,storage server sends abort signal. Abort signal indicates that download process is failed. If it has required privileges the corresponding encrypted file is given to user. Once user get file, he/she applies convergent key to decrypt file and recover file.

### C. Problems

Deduplication scheme, above described has following problems: _ Initially, Private keys fkpi gpi _ PU are issued for each user. File token is computed by user using issued private keys fkpi gpi _ PU and that token is used for duplicate

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 7, July 2016**

check. User generates file tokens during process of uploading and that file tokens will be used to share among different users. Different users have privileges PF . In order to generate file tokens, private keys for privilege PF must be known to user. For selection of privilege keys PF there is restriction that PF can be taken from PU only. _ Another problem is that user must share private keys with other users. Thus we cannot prevent of sharing private keys. Problem arises because all users are given same keys for same privilege. Which can result in colluding keys and generate privilege private keys for a new privilege set P* i.e. this set is unique. For example, if one user is having privilege set PU1 which is colluded with
another set PU2 and there is possibility of generating new set which is union of both sets.
_ The above mentioned approach is vulnerable to brute force attacks. Which canbe effective on the files of which information is known. Attack can recover known files. Hence, files that are predictable for that deduplication system cannot protect the security.

**D. Proposed System Description**

In previous Section, due to limitations like user must share private keys among other users, there is need to propose another deduplication system which is able to accomplish authorized duplicate check in efficient manner. Hybrid cloud architecture is implemented to overcome the above mentioned problems. In this architecture instead ofissuing private key directly to users, private keys are saved and managed by private cloud. Hence, the users not have to share their private keys of privileges among other users as in the previous construction. User sends a request to private cloud with tag, in order to get a file token. After user has got file token from private cloud, he/she can use that token for requesting public cloud or storage server for duplicate check. Users identity is also checked by private cloud server, prior to issue corresponding file token to the user. Before uploading this file on cloud storage, authorized duplicate checking of that file is done by public cloud or storage server. When duplicate checking is done, the user is allowed to either upload the file or run PoW

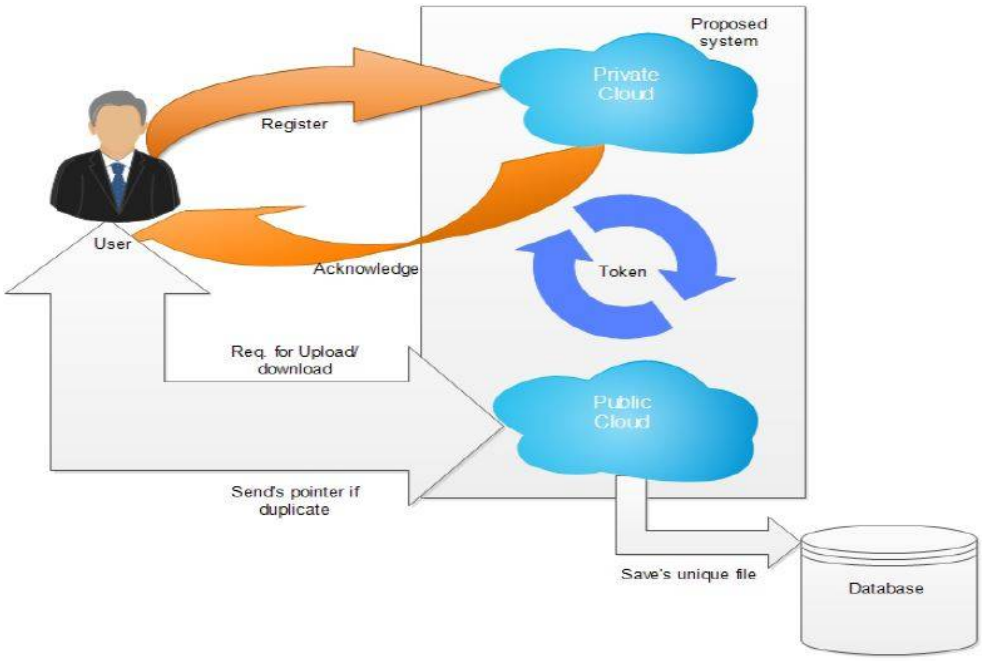


**Fig. 2 System Architecture**

Binary relation can be written for it which can be expressed as R= prv, prv0 ), in that prv and prv0 are two privileges such that prv and prv0 both will matches only when R(prv, prv0 ) = 1.If hierarchical relation is considered, prv matches prv0 if prv is a higher-level privilege. It can be understood by simple example, in organization there are Project lead,

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 7, July 2016**

Director and Engineer according to their privileges, where Director is at the top level according to hierarchy and at bottom there is an Engineer. From above example it can be easily seen that, the Director is having more privileges than others. Proposed deduplication system is divided into:
1) System Setup: Let p be the set of privileges one user can have, it is called as the privilege universe P. A symmetric key is selected i.e. fkpi gpi _ P which is set of keys are sent to the private cloud. Another protocol is also defined for identification of user which is identification protocol _= (Proof, Verify), in which Proof and Verify are algorithms used for users proof and clouds verification. Also, suppose each user U have a secret key skU which can be used in identificationprocess with servers. PU is privileges issued for user U. TheProof of ownership protocol is also initiated. Private server keeps table having information about each users privileges and corresponding public information.
2) File Uploading: If user wishes to upload file on server or share it with other users, he must request to private server in order to perform duplicate check with public cloud. Then, there is identification process for data owner in which owner must proves his/her identity with private key skU. When private cloud server has found the corresponding privileges PU of the user from table, identification is passed. Then tag is generated using method TF =generateTag(F). Private cloud with the help of tag generates token and returns it to client i.e. T'F ,p = generateToken (TF ,kp_ ). After this user sends the file token T'F , p_ to public cloud for duplicate check. _ When it is found that file is duplicate, Proof of Ownership protocol is run on S-CSP to prove whether the user is owner of that file or data. If proof is passed, then and then only user is issued a pointer for accessing file. This Proof includes time stamp and pku. The user sends two things to private cloud, first one is privilege set PF = pj for the file F and second is proof . The tag is generated by private cloud T'F , p_ = generateTag (TF , kp_ ) , after proof is passed, for all p_ satisfying R(p, p_ ) = 1 for each p _ PF PU, user is given set of privileges after which user submits these tokens to the public cloud server to perform duplicate check. _ When file is found to be unique, a proof is returned, which includes a signature on T'F ,p_ , also a time stamp and pkU. To private cloud user sends two things first is privilege set PF = pj for the file F and second is proof. When private cloud receives request form user, it verifies the proof. If proof is passed, private cloud server computes T'F ,p_ = generateTag (T'F ,p_ ), for all p_ satisfying R(p, p_ ) = 1 and p PF. At the last, encrypted file is generated by the user computes the ciphertext CF = encryptFileCE(kF,F). For this encryption, convergent key is generated as kF = generateKeyCE(F) and stores CF , T'F ,p_ with privilege PF on public cloud.
3) File Retrieving: Original file is recovered by user with the help of convergent key after downloading the encrypted file from the public cloud.

## VI. IMPLEMENTATION

### A. System module Implementation

A Client program is for the data users to upload or download files. On the other hand a Private Server module is there which manages both private keys and file token. A public cloud or a Storage Server program is used to model the S-CSP which stores files and duplicate check is done there. Deduplication of files is done on public server. Also, cryptographic operations are implemented like hashing and various encryption techniques with the help of OpenSSL library.
1) Client Implementation: Client provides the following function calls:
_ generateFileTag(File) Method computes SHA-1 of given file and this hash is treated as tag for that file; _ requestForToken(Tag, UserID) This method requests to the Private Server for generating file token for that arguments are passed which are file tag and userid ofuser. _ requestForDupCheck(Token) This method requests the S-CSP or storage server for checking whether the file is already stored on cloud or not. User sends token got from the private cloud to public cloud in order to perform duplicate check. _ requestToShareToken(Tag, Priv.) User requests to private cloud for generating the Share File Token, this method is taking tag and set of privileges as arguments; _ encryptFile(File) 256-bit AES algorithm is used to encrypt the file, where the convergent key is generated which is nothing but SHA-1 hash. _ requestForFileUpload(FileID, File, Token) In order to store the file, this method requests the Storage Server to upload file, if the file is not duplicate i.e. no duplicate isfound and finally file Token is updated.
2) Private Server Implementation: Following methods are used in private cloud or server:
_ generateToken(Tag, UserID) This function loads privilege keys which are associated with that user and token is generated with HMAC-SHA-1; and _ generateShareToken(Tag, Priv.) - Share token required for sharing privileges among different users is generated for sharing privilege set.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

3) Public Server Implementation: Following methods are used in public cloud or server:
_ dulicateCheck(Token) This function finds the file to token map for checking whether the file is duplicate or not; and
_ storeFile(FileID, File, Token) - This function stores the File on storage and updates table where mapping is stored.

**B. Proof of Ownership Protocol**

At the time, when user wants to upload file on S-CSP, if file is found to be duplicate, user have to proof that he/she is indeed owner of that file. User has to run protocol or (PoW) [8]. PoW has two modules which can be implemented on interactive basis, first module is prover and other is verifier. Prover module is user or client and verifier is S-CSP or public cloud where data resides. The verifier computes a short value which is _(F) from a file F. Prover has to submit a proof that
he/she owns that file F, for that prover have to send _ 0 to the verifier such that _ 0 = _(F). Threat model is the base of PoW, where an attacker knows who have or who owns file, he does not know all details of entire file. PoW ensures that user holds entire file rather than some information about that file. It has two phases: first is Proof and second is Verify. In Proof, prover should submit some distinct attributes which only he can posses like credit card number etc. In this case input used for prover is his/her private key. This private key should not be shared with anyone else. The verifier which is public cloud can verify identity from input of public information related to private key. Thus verifier either accepts the request or decline on that basis it is decided that proof is passed or not passed. Some identification protocols are certificate-based, identity-based identification etc. [6], [7]. In storage server decision of whether the file is duplicate or not is taken by looking at hash tables.In that hash tables all registered users and their submitted hashes of files . If there were no hash found in the hash table then normally server demands entire file. On the opposite side, if file already stored on cloud server tells client that no need to upload that file again. In both cases server assumes client as a owner of that file. And that relation continued, means whether client is original party looking for uploading or not does not matter. Hence, some time client can ask to restore file. In responseserver asks hash value. client has not uploaded file he may ask for restoring that file. For example, a if server is telling that no need to send file, it clearly indicates that some one else holds that file, this is also important information leak.

1) A New Attack.: Direct threat is for storage system that offers client side deduplication across multiple users. Also,accepting only hash value and hand over the entire file is not a very good idea. Because by accepting the hash value in place of whole file, the server can allow anyone to get file who submits hash. For eg: Ajay is a employee who routinely publishes hashes of files. Bob takes daily backup carefully with him, to an online file storage service. Attacker Alice is seeking for learning the reports uploaded by Bob. Alice logs in for same storage system, asks for reports, and storage system demands for hash, she submits the hash of Bob which he publish as a time-stamp for his reports. After same time or days, Alice demands to find or restore that file and storage system without looking that whether the Alice is originally owner of that file or report hands over that file of Bob to Alice. It can clearly summarise that from only piece of information attacker can access entire file from storage server. Also, it is common that which has might be used. Due to this life of storage system becomes more difficult. A temporary solution is to attach some identity with that hash and be verified by storage system. Due to this,same hashing mechanism is usedagain and again. However, solution is not permanent, as it isunable to point out the root cause used for performing attack.

2) Potential Attacks: Some attacks which are performed by attacker are summarised below,

3) Usage of a common hash function.: As described earlier, nearly all deduplication systems uses a common hash. And usage is very straightforward. It is known that Even DropBox uses SHA256 hash. In this way at least one thing is clear to attacker that hash used is common. Attacker can access the hash values of files owned by other users with not more efforts.

4) Using the storage service as an unintended CDN.: Now, another type of attack is performed as attacker uploads a copy of a file to the storage server this file is huge in size.Attacker then publishes the hash which corresponding to that file. Then users who wants to access the file can try to upload it to storage server, do formality of submitting hash to the service and gives the proof that he/she is owner of that file. Again Alice the attacker requests to restore the file from the storage server. In this way Alice essentially uses the storage service as a content distribution network (CDN). From this behaviour it is clear that, there should be some ownership roles, rules or terms that can avoid foul of business model. This kind of attack may lead to piracy and copyright related infringing behaviour.

5) Server break-in. : In this scenario an attacker anyhow compromises server machine, getting access to internal information, which could be set of the hash of all files some of files might be recently accessed. After getting all these

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 7, July 2016**

information, the attacker can easily download all files, including confidential or more sensitive files. Though it is found that system has compromised there is hardly to do about it. Only possible solution is to restrict client side deduplication of files which are stolen.

6) Malicious client software.: Attack can be performed on client side also. For example, Alice can install software on client side or on the victim's machine who is client. That software sends Alice the hash of files. Once hash data is acquired Alice can download rest of files entirely. Hash based attacks are proved to be successful no matter how strong are victim's security credentials. These type of attacks cannot detected easily since attacker gives illusion that he/she is legitimate user of that system.

7) Accidental leakage.: There is no need of malicious software, just a short value is sufficient to leak big files. There is possibility of swapping the value into disk. If same file is owned by many clients(e.g., a department-wide document), it is possible that one of them will accidentally leak the short information about that file which is sufficient for attacker to access that file unauthorised way.

8) Effect on Open APIs for Remote Storage.: Given the many avenues, described above, for the attacker to get hold of the hash values of files, it is clear that using only a small hash as a proxy for a large file is a security vulnerability in systems that use client-side deduplication. This vulnerability must be addressed before we can attempt to create open APIs for remote-storage that support client side deduplication, since such APIs have to be usable in environments that are sensitive to some of the attacks above. We thus view a solution to this problem as an enabler for the creation of advanced open storage APIs.

## VII. CONCLUSION

From the Study of Data deduplication and its types can be very helpful for implementing secure and cost efficient storage system. New system is proposed which can handle differential privileges providing security and authorized access. Due to use of Hybrid cloud approach data confidentiality and authorization is separated from storage mechanisms at public cloud. Hybrid cloud approach also helped in problems which arose during single cloud approach like sharing of private privilege keys. Also, Focus is thrown on Proof of Ownership protocol for users to verify their identities. With all above mentioned entities one can implement secure and authorized data deduplication in cloud environment.

## VIII. FUTURE SCOPE

Even though proposed system performs well almost for all required circumstances and if rate of growth of data is also considered then traditional relational database systems (RDBMS) used in proposed system that might unable to scale to process large amounts of data which requires scalable andcost effective parallel processing to handle such a huge amount of data in future.

## ACKNOWLEDGEMENT

This is to acknowledge and thank all the individuals who played defining role in shaping this conference paper. Without their constant support, guidance and assistance this paper would not have been completed alone. I take this opportunity to express my sincere thanks to my guide and Head of Computer Department, Prof. Arti Mohanpurkar for her guidance, support, encouragement and advice. I will forever remain grateful for the constant support and guidance extended by my guide, in making this conference paper work. I wish to express my sincere thanks to P.G. Coordinator Prof. Roshani Raut and the all departmental staff members for their support.

## REFERENCES

[1] P. Anderson and L. Zhang, Fast and Secure Laptop Backups with Encrypted De-Duplication,USENIX LISA, , pp. 1-8 ,UK:University of Edinburgh 2010.

[2] Pasquale Puzio ,Refik Molva , Melek nen and Sergio Loureiro, Secure Deduplication with Encrypted Data for Cloud Storage,SecludIT and EURECOM ,France, March, 2013.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 5, Issue 7, July 2016**

[3] J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer, Reclaiming Space from Duplicate Files in a Serverless Distributed File,Microsoft Corporation ,Microsoft Research, Redmond, WA 98052, July 2002.
[4] David Geer, Reducing the Storage Burden via Data Deduplication, computer.org , December 2008.
[5] J. Xu, E.-C. Chang, and J. Zhou, Weak leakage-resilient client-side deduplication of encrypted data in cloud storage, ASIACCS, pages 195206 ,Singapore,2013.
[6] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identitybased identification and signature schemes, J. Cryptology,22(1):161 ,University of California, San Diego,2009.
[7] M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks,CRYPTO, pages 162177 ,2002.
[8] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg. Proofs of ownership in remote storage systems,ACM Conference on Computer and Communications Security, pages 491500 ,ACM, 2011.
[9] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou, A Hybrid Cloud Approach for Secure Authorized Deduplication ,IEEE Transactions on Parallel and Distributed Systems,Volume:PP,Issue:99 ,Date of Publication :18.April.2014.
[10] C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups, In Proc. of APSYS ,April, 2013.
[11] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Rolebased access control models, IEEE Computer , Feb,1996.
[12] D. Ferraiolo and R. Kuhn.Role-based access controls, In 15th NISTNCSC National Computer Security Conf ,1992.
[13] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage,In USENIX Security SymposiumUniversity of California, San Diego, 2013.
[14] J. Yuan and S. Yu, Secure and constant cost public cloud storage auditing with deduplication., IACR Cryptology ePrint Archive, 2013:149. University of Arkansas at Little Rock,2013.