Secure Deduplication on Hybrid Cloud Storage with Key Management

K.Kanimozhi[1], N.Revathi[2]

[1] M.E Student, Dept. of Computer Science Engineering,
Sri Venkateswara College of Engineering, Sriperumbudur, Tamilnadu, India
[2] Assistant Professor, Dept. of Computer Science Engineering,
Sri Venkateswara college of Engineering, Sriperumbudur, Tamilnadu, India

---***---

Abstract - *Cloud storage provides the storage space based on the pay-as-you-go model. Number of users tremendously increases the size of data with more number of duplicate copies of identical data. To reduce the amount of storage space data deduplication technique is used. Data deduplication is the data compression technique for eliminating the duplicate copy of same data. To protect the confidentiality of sensitive data, convergent encryption technique is used and secure proof of ownership(PoW) is used to block unauthorized access of the data. Data are encrypted with a convergent key , which is derived from the content of the data copy to obtain the same ciphertext using the cryptographic hash function. After data encryption, users sends the ciphertext to the cloud. To provide higher level of confidentiality and security, key management technique is used to manage the convergent keys using cryptographic hash function and sends it to the cloud.*

Key Words: **Data Deduplication, Convergent Encryption Technique, Proof of Ownership Protocol, Authorized Duplicate Check**

1.INTRODUCTION

Cloud storage provides cost effective resource usage as a service to users. Every user has large amount of data to store it in a secured storage area. To make data management scalable in cloud storage data deduplication technique is used. Data deduplication is the process is identifying duplicate data stored in the cloud, it will reduce the cost and maximize the storage space to the user. Deduplication can take place in file-level and block-level. Data deduplication brings a lot of benefits. Security and privacy concerns arise as users sensitive data are outsourced to cloud storage. In traditional encryption, data confidentiality and security is incompatible with data deduplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, same data copies of different users will lead to different ciphertexts, it makes deduplication impossible. To provide secure deduplication[6], convergent encryption has been proposed to achieve data confidentiality. It encrypts/decrypts a data copy with a convergent key, which is derived from the data using the cryptographic hash functions to obtain the same ciphertext. To prevent unauthorized access to the data secure proof of ownership protocol[3] is used to prove that user owns the same data and Differential authorization duplicate check is used to allow only a authorized user to check duplicates of data that is for each user set of privileges is issued during the system initialization. This paper will work to provide higher level of confidentiality and security, along with authorized duplication system, key management technique is used manage the convergent keys using the cryptographic hash function.

2. RELATED WORK

Jan Stanek_, Alessandro Sorniottiy, Elli Androulakiy, and Lukas Kencl proposed a secure data deduplication scheme for cloud storage[1], encryption scheme that guarantees semantic security for unpopular data and provides weaker security to unpopular data and better storage and bandwidth benefits for popular data, so that data deduplication can be applied for the (less sensitive) popular data. Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P.C. Lee, and Wenjing Lou presents hybrid cloud approach for secure authorized deduplication[2], to protect the confidentiality of sensitive data with deduplication, the convergent encryption technique is used to encrypt the data before outsourcing and authorized duplicate check is proposed to allow only authorized user to check duplicate of data. Shai Halevi, Danny Harnik, Benny Pinkas, Alexandra Shulman-Peleg, proposed proofs of ownership in remote storage systems[3], proof of ownership is proposed to overcome the attack that user knows the hash signature of file without having full file it convince the storage service that it owns the file and download the entire file. Mihir Bellare Sriram Keelveedhi Thomas Ristenpart proposed dupless: server-aided encryption for deduplicated storage[4], In DupLESS, clients encrypt under message-based keys obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an existing service, have the service perform deduplication on their behalf, and yet achieves strong confidentiality guarantees.

3. PRELIMINARIES

In this section we define some primitives used in secure deduplication.

3.1 Data Deduplication

Data deduplication[3] is the process of identifying duplicate data stored in the cloud, it will reduce the cost and maximize the storage space to the user. Deduplicate with compression technique is one of the major technology involved in the process. Data compression is the major process to redundant with binary level data. Data deduplication process finds duplicate data and avoid it form the storage space. Byte level deduplication process ensures better result in this process. It provides more accuracy and high performance to the users. On other hand deduplication process, where the server stores a single copy of each and every data, many user will access the same file.

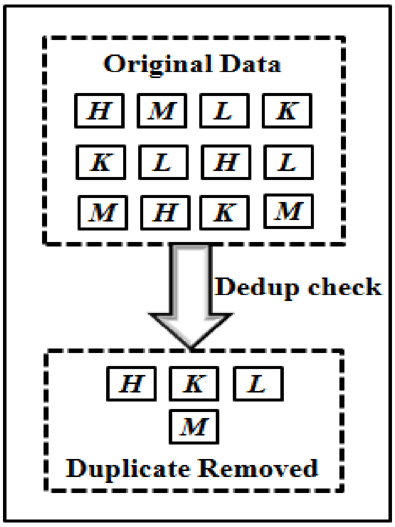

Fig -1: Deduplication

3.2 Traditional Encryption

In Traditional encryption users use their own key(k) to encrypt/decrypt there data, which leads to different ciphertext.

- KeyGen$_{TM}$(x) == k is the key generated from the key generation algorithm using the secret parameter x.
- Enc$_{TM}$(k,F) == C is the ciphertext generated from the data M and key k using the traditional encryption algorithm.
- Dec$_{TM}$(k,C) == F is the original data generated from the ciphertext C and key k using the traditional decryption algorithm.

3.3 Convergent Encryption

Convergent encryption algorithm is also called content hash keying. It encrypts/decrypts the data with convergent key K derived from the content of the data to obtain the same ciphertext and tag is additional hash signature derived to check the duplicate copies of same data which is different from convergent key. It gives secure duplicate check, if user need to upload a file first they need to send the tag derived from the data and check the duplicate. If no duplicate found file is uploaded otherwise no need to upload file. Convergent encryption algorithm defines following four primitives.

- KeyGen$_{CE}$(F) == K is the convergent key which is derived from the data copy using the key generation algorithm.
- Enc$_{CE}$(K,F) == C is the ciphertext derived from traditional encryption algorithm using convergent key K and data F.
- Dec$_{CE}$(K,C) == F is the original data derived from decryption algorithm using convergent key k and data F.
- TagGen(F) == T(F) is the tag used to check the duplicate data copies derived from data F using tag generation algorithm.

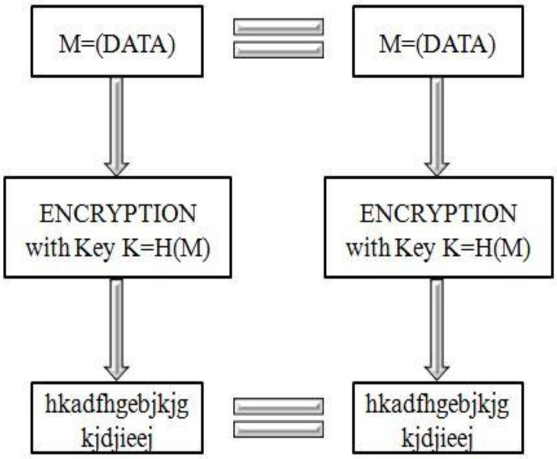

Fig -2: Convergent Encryption

3.4 Proof of Ownership

Proof of ownership(PoW)[3] is used to prove the ownership of the data to the storage-cloud service provider. It is an interactive algorithm. cloud storage provider derives the tag value Φ(F) from the data already stored in cloud and the user need to send the tag value Φ' derived from data are verified Φ'= Φ(F) to proof the ownership of data.

3.5 Authorized Duplicate Check

Authorized users use their individual secret key to query the public cloud to perform the duplicate check directly and tells if duplicate is found or not, with the help file token provided by private cloud storage for certain file and the privileges that user owned.

4. SYSTEM MODEL

Secure deduplication with authorized duplicate check system is achieved using the hybrid cloud storage with convergent key management. This system consisted of four modules those are, tag generation, key generation, encrypting/decrypting using convergent encryption technique and key management.

Table -1: Notations used in paper

Acronym	Description
PoW	Proof of Ownership
P,P'	Privileges set of a users
P'	Privileges set of a file
H,H0,H1	Cryptographic hash function
Φ(F)	Token of a file
K_p	Privilege key

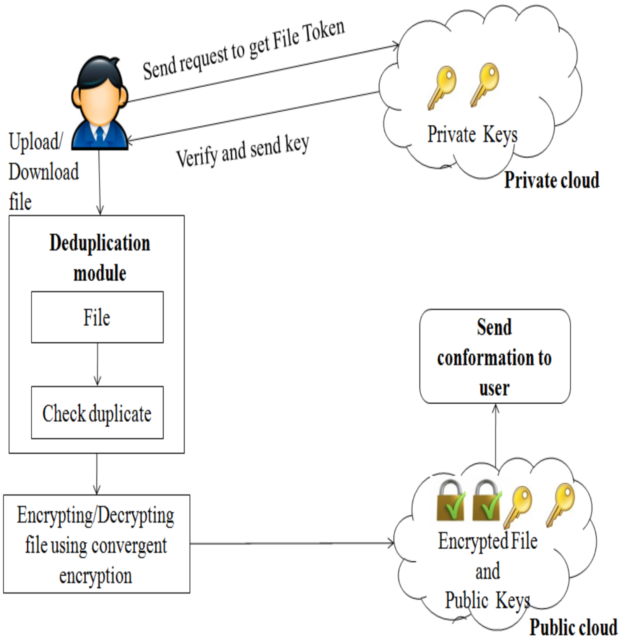
Send request to get File Token
Verify and send key
Private Keys
Private cloud
Upload/ Download file
Deduplication module
File
Check duplicate
Send conformation to user
Encrypting/Decrypting file using convergent encryption
Encrypted File and Public Keys
Public cloud

Fig -3: Architecture of Secure deduplication

4.1 Tag Generation

Hybrid cloud storage provides the advanced deduplication system, to support authorized duplicate check. Each file is associated with some file tokens, which denote the tag with specified privilege p of user. To check duplicate of file, user first computes and send tag TagGen(F,k_p),(where k_p is the secret key bounded with privilege p) as the request to the private cloud storage. The private cloud storage verifies the tag and returns the file token corresponding to the privilege of the user to perform duplicate check of the file in public cloud storage.

4.2 Key Generation

Key generation algorithm is used to generate the convergent key. The convergent key K=KeyGen$_{CE}$(F) are derived from the file using cryptographic hash functions for encrypting and decrypting the file to obtain the same ciphertext to make the deduplication feasible.

4.3 Encrypting/Decrypting using Convergent Encryption Technique

In secure deduplication, confidentiality is achieved using convergent encryption technique. Suppose user with the privilege p' wants to share file F' to the users with set of privilege P'. User first need to compute the tag Φ(F)=TagGen(F',$k_{P'}$) for the file F' with privilege P'. After the verify from the private cloud duplicate check of file is performed. If duplicate is found then proof of ownership is need to be proved. If proof is passed then the pointer for the file is assigned to the users with the privilege P'. If no duplicate is found then file is encrypted C=Enc$_{CE}$=($K_{F'}$,F') using the convergent key $K_{F'}$=KeyGen$_{CE}$(F') and the ciphertext C along with tag Φ(F').

4.4 Key Management

The above solution of authorized duplicate check primarily subject to brute-force attacks launched by public cloud storage. It regain the files falling into known set. Confidentiality and security is possible only for the unpredictable file. To provide higher level of confidentiality, security in duplicate check and to avoid deterministic key generation, the key are managed with the help of private cloud storage with secret key(privilege key) K_p. The manage convergent key $K_{F,p}$= H_0(H(F),K_p)⊕H(F). H_0,H,H_2 are the cryptographic hash functions.

5. IMPLEMENTATION AND RESULT

Secure deduplication with authorized duplicate check system is implemented with four modules as separate java programs. First the system begin with the user login to enter into modules to perform the process. Tag generation program is used to generate the tag to send the request the private cloud storage for computation of file token to check duplicate. Key generation program is to compute the convergent key from the content of data to obtain same ciphertext.

Encrypting/decrypting using convergent encryption program form encrypting and decrypting data store it in public could storage. Key management program is for manage keys in the cloud storage using hash functions. The following functions shows the operation performed in the proposed system.

- Tag(File) == It computes the hash value of the file as tag using SHA-1
- TokenReq(Tag,UserID) == It request the private cloud storage to compute the file token for the specified file tag and userID.
- TokenGen(Tag,UserID) == It generates the token for the specified privileges of user using HMAC-SHA-1.
- ShareTokenGen(Tag,{Priv.}) == It generates the share token with corresponding privilege of the privilege sets using HMAC-SHA-1.
- DupCheckReq(FileToken) == It request the public cloud storage for duplicate check of file using the file token.
- DupCheck(Token) == It check the duplicate file and maps it in the public cloud storage.
- FileEnc(File) == It encrypts the file using the convergent encryption algorithm with AES-256 in cipher block chaining. Where the convergent key for encryption is derived from the file using SHA-1.
- FileUploadReq(FileID,File,Token) == If no duplicate is found.It upload the file to public cloud storage with file token. and update the file token.
- FileStore(FileID,File,Token) == It stores the file on the storage.

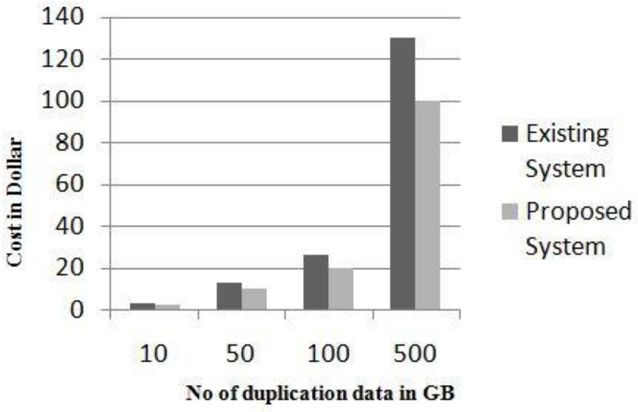

Chart -1: Difference in Deduplication ratio

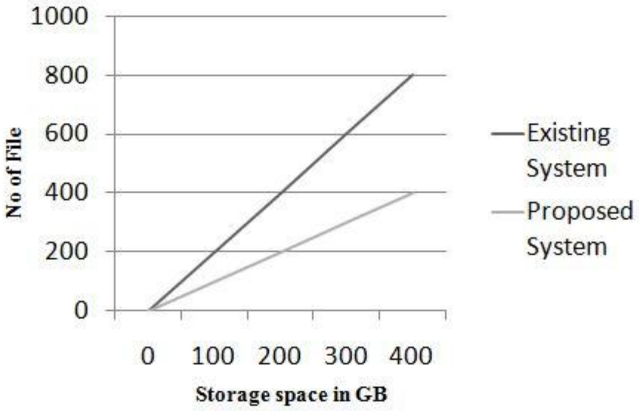

Chart -2: Comparing duplicate File with storage space

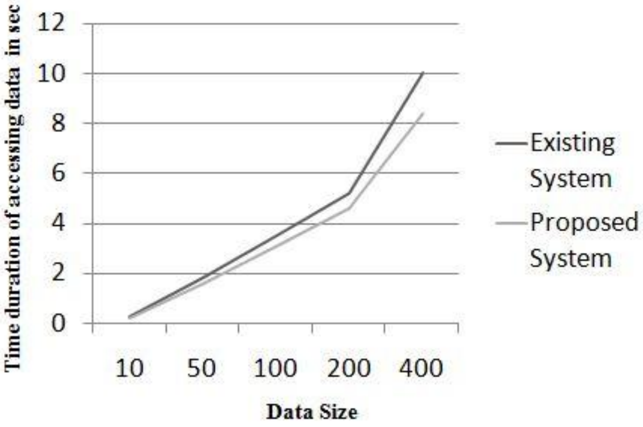

Chart -3: Time breakdown for accessing stored files

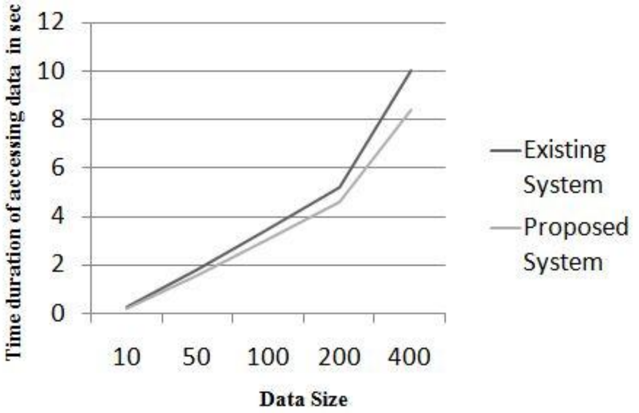

Chart -4: Time breakdown for different number of stored files

6. CONCLUSIONS

In this paper, secure deduplication with authorized duplicate check system is implemented using the hybrid cloud architecture. Duplicate check is allowed only if the user is authorized with the file token. After the duplicate is found, proof of ownership protocol is need to prove ownership of file by the user and the convergent keys are managed and stored in the cloud using the cryptographic hash function with the privilege keys. This system demonstrates that the files which falls into the known cannot determined. This scheme provides secure deduplication with confidentiality and security.

REFERENCES

[1] J.Stanek, A.Sormiotti, E. Androulaki, and L. Kencl, "A secure data dedupliction scheme for cloud storage," *Tech. Rep. IBM Research,* Zurich, ZUR 1308-022, 2013.

[2] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P.C. Lee, and Wenjing Lou, "A Hybrid cloud approach for secure authorized deduplication," *IEEE Transaction on parallel and distributed system,*Vol.26, No. 5, May 2015.

[3] Shai Halevi, Danny Harnik, Benny Pinkas, Alexandra Shulman-Peleg,"Proof of Ownership in Remote Storage Systems," *IBM Research,* April 29, 2011.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart. "Dupless: Server aided encryption for deduplicated storage," *In USENIX Security Symposium,* 2013.

[5] J. Li, Xiaofeng Chen, M. Li, Jingwei Li, Patric. P.C. Lee, and Wenjing Lou. "Secure deduplication with efficient and reliable convergent key management," *In IEEE Transactions on Parallel and Distributed Systems,* Vol.26 No.6 June 2014.

[6] M. W. Storer, K. Greenan, D. D. E. Long, and E.L. Miller, "Secure data deduplication," *in proc. 4th ACM Int. Workshop Storage Security Survivability,* 2008, pp. 1-10.

[7] Jin Li, Xiaofeng Chen, Xinyi Huang, Shaohua Tang, Yang Xiang, Mohammad Mehedi Hassan, Abdulhameed Alelaiwi, "Secure Distributed Deduplication Systems with Improved Reliability," *IEEE Transaction on - computers,* 2015.

[8] Wee Keong Ng, Yonggang Wen, Huafei Zhu,"Private data deduplication protocols in cloud storage." in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ACM, 2012, pp. 441–446.

[9] M. Bellare, S. keelveedhi, and T. T. Ristenpart, "Message-locked encryption and secure deduplication," *in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptographic Techn.,* 2013,pp.291-312.

[10] Bugiel, Sven, et al. "Twin clouds: Secure cloud computing with low latency," *Communications and Multimedia Security. Springer Berlin Heidelberg*, 2011.

[11] Jadapalli Nandhini, Ramireddy Navateja Reddy. "Implementation of hybrid cloud storage approach for secure authorised deduplication," *In International Research Journal of Engineering and Technology*, Vol: 02 Iss: 03 June 2015.

[12] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 40–47, 2010.

[13] R. D. Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication." in *ACM Symposium on Information, Computer and Communications Security*, H. Y. Youm and Y. Won, Eds. ACM, 2012, pp. 81–82.

[14] A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM conference on Computer and communications security*, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 584–597.

[15] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," in *Proc. of USENIX LISA*, 2010.