A Secure Authorized Deduplication using Hybrid Cloud Approach

Rajashree D. I [2], Azhar Sharieff[1], Faizan Ahmed Khan[1], Pravitha.P[1], Sneha Nanaiah.K[1].
[2]Assistant Professor,
Department of Information Science,
AMC Engineering College,
Bangalore.

Abstract: **Today large amount of data is being stored on the cloud. To make data management scalable, deduplication technique is used.Data deduplication is one of important data compression techniques for preventing multiple copies of same data, and has been widely used in cloud storage to reduce the amount of storage space and to save bandwidth. To protect the confidentiality of sensitive data the convergent encryption technique has been used that is, encrypting the data before outsourcing it. In order to support secure and authorized deduplication, differential privileges of the users are considered. For better security this system uses a hybrid cloud architecture, which also supports the authorized duplicate check.**

Keywords—Deduplication, authorized duplicate check, confidentiality, hybrid cloud

I. INTRODUCTION

One critical challenge that the cloud storage service providers are facing today is the management of the ever-increasing volume of data.To make data management flexible in cloud computing, deduplication technique is used. A data compression technique which is used to reduce the number of bytes sent is called Data Deduplication [17]. This technique is used to improve storage utilization. Instead of keeping multiple copies of same data, deduplication keeps only a single copy of the data on the cloud, and a pointer is provided for any other user trying to upload the same file again.

There are two places where the Deduplication can be performed either at file level or the block level file-level. The File level deduplication eliminates multiple copies of same file, whereas the block level deduplication eliminates duplicate blocks of data that occur in different files. The Security and privacy concerns arise as the user's sensitive data is vulnerable to attacks, although it provides deduplication.

To provide security and privacy in deduplication system convergent encryption is used. Convergent key is used to encrypt and decrypt the keys and cipher text is sent to the cloud. The convergent key is obtained by computing the hash value of the content of the data of the file. Secure proof of ownership protocol is used to avoid unauthorized access.

A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys.

A hybrid cloud architecture is used for providing security where in it consists of a private cloud and a public cloud. Private cloud manages all the encryption mechanism and the public cloud stores all the encrypted files.

II. RELATED WORK

Yuan et al. [24] proposed a deduplication system in the cloud storage to reduce the storage size of the tags for integrity check.

Bellare et al. [3] showed how to protect the data confidentiality by transforming the predictable message into unpredictable message. In their system, another third party called key server is introduced to generate the file tag for duplicate check.

Stanek et al. [20] presented a novelencryption scheme that provides security for popular data and unpopular data. The protocol used scales well with large members of files and the users. Storage optimization methods and end to end encryption are the techniques used. File transitions from one mode to the other takes place at the server side if and only if a file becomes popular data.

Li et al. [12] proposed secure deduplication with efficient and reliable convergent key management. This paper suggests that the users need not manage the convergent keys, rather an algorithm is proposed to manage the convergent keys.

Bellare et al. [4] proposed Message-locked encryption and secure deduplication. Message-locked encryption (MLE) is a cryptographic technique, where a key under which encryption and decryption are performed is derived from the message. MLE also allows to achieve secure deduplication

P. Anderson et al. [2] proposed Fast and Secure Laptop Backups with Encrypted Deduplication. Here an algorithm and prototype software where data is encrypted independently without invalidating the deduplication.

Halevi et al. [11] proposed a protocol "proofs of ownership" (PoW) for deduplication systems, the proposed system develops a security scheme which can be used by the users to prove that the file is indeed owned by him/her.

Bugiel et al. [7] proposed Twin clouds: An architecture for secure cloud computing. This architecture allows secure outsourcing of data. The computations are split such that the trusted cloud manages security critical operations and commodity cloud manages the performance critical operations.

III. TECHNIQUES

This section defines the secure primitives used in our deduplication system.

Convergent Encryption: It is a symmetric encryption techniqueencryption, in which the same key is used for the purpose of encryption and decryption, and the key is generated from the data of the file itself. Formally, a convergent encryption scheme can be defined with four primitive functions:

•KeyGen (M) → K is the key generation algorithm that generates the key from the data copy itself.

•Enc (K, M) → Enc is the symmetric encryption algorithm that takes both the convergent key K and the file M as input and then gives the cipher text C as output.

•Dec (K, C) → The Decryption algorithm takes cipher text C and the key K as input and generates the plaintext from the cipher text.

Proof of ownership: The aim of proof of ownership (PoW) [11] enables users to prove their ownership of data copies to the storage server. In this protocol the server behaves as a verifier and the user behaves as a prover to prove the prover has to send the token of the file to the verifier. If the token sent matches with the token of the file that the server has, then it proves that the user is the owner of the file.

IV. SYSTEM ARCHITECTURE

The Fig.1 showsarchitecture diagram of Authorized Deduplication. The entities of the architecture are as follows:

S-CSP: It provides a data storage service in public cloud and also provides data outsourcing service and stores data on behalf of the user.

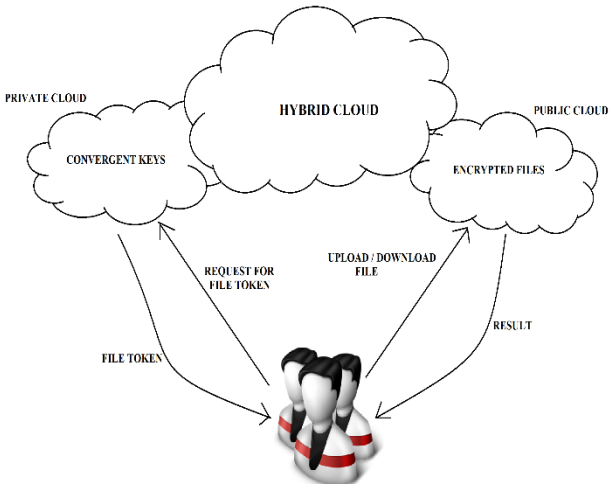

Fig. 1. Architecture of Authorized Deduplication.

Data users: It is an entity that wants to outsource data storage to S-CSP and access the data later.

Private cloud: This entity facilitates user's secure usage of cloud server. It stores the encryption keys and manages the encryption mechanism.

Public cloud: It stores the encryption mechanism.

Admin: This entity helps in various operations like registers a new user, generates keys based on ranks. Admin can also view the cloud details and the transaction details of the user.

Suppose user wants to upload file, the admin creates user and at the time of creation assigns a rank to the user, and based on the rank relevant key for encryption and decryption is provided to the user. The user only have to select the file from storage. The selected file undergoes hashing process (MD5) to generate a hash code .The hash code is obtained from the contents of the file. At the time of upload the hash code undergoes a deduplicate check process, if a duplicate file is found, the user needs to run the PoW protocol with the S-CSP to prove the file ownership. Then the privilege of the user is checked and if the user has the necessary privilege then the user can upload the file to the server. Before uploading the file is encrypted using Convergent Encryption. But if the user does not have the necessary privilege then the file upload fails, but the user will be allowed to download the file.

If a user wants to download a file from the cloud, he should select a file to download, the access right of the user is checked, if the user has the necessary access rights then the file can be downloaded and decrypted using the key, else file download fails, and the user is shown a message stating that he does not have the necessary access rights to download the file.

V. IMPLEMENTATION

In proposed system two levels of deduplication is carried out i.e. in private cloud and then in public cloud and the file is directly uploaded to public cloud. Hybrid cloud architecture is secure and efficient by doing the deduplication in private cloud. Duplication is checked by using MD5 algorithm. If file is already uploaded to private cloud by owner earlier it does not allow storing the file again so unnecessary transmission is prevented. If file is not uploaded by owner he assigns keys to encrypt the file using RNS algorithm and being stored in private cloud. Data owner sets privileges to data users to access files.

In the proposed system three entities are used namely, *Client* program, *private server* program and *storage server* program.

File upload operation are modelled using *Client program*. The private keys and file tokens are managed by modelling *private server* program. Storing of files and Deduplication checks are modelled by *storage server* program.

The following function calls are provided by the client program in order to support token generation, Deduplication checks and file upload.

- Token (File) – The token for the file is generated using MD5 hashing technique
- DupCheck (Token) – The token of the file generated is sent to the public cloud to check for duplicates.
- FileEnc (File) – It encrypts the file using RNS algorithm.
- FileUpload (FileID, File, Token) – If the contents of the file are unique ituploads the encrypted file on the public cloud

The following function calls is provided by the Private Server program for token generation and key generation based on privilege of the user.

- TokenGen (File) – It generates a token for the file using MD-5 hashing technique.

The following function calls are provided by the Storage Server program for data storage and maintaining map between existing files, associated token with Hash

- DupCheck (Token) – It performs Deduplication checks on cloud to check if the same copy of the file is already present.
- StoreFile (FileID, File, Token) - It stores the File on cloud.

VI. EVALUATION

We are evaluating the system by comparing overheads in various steps like token generation and file upload process. We are evaluating overhead by varying factors like 1) File Size 2) Number of Stored Files.

A. File Size

In order to evaluate the effect of file size, we upload 100 files of different size and we are recording the time for file break down. We observe from the Fig, 2 that as the size increases the time for break down also increases.The time taken for tagging, encryption and upload increases linearly with the file size

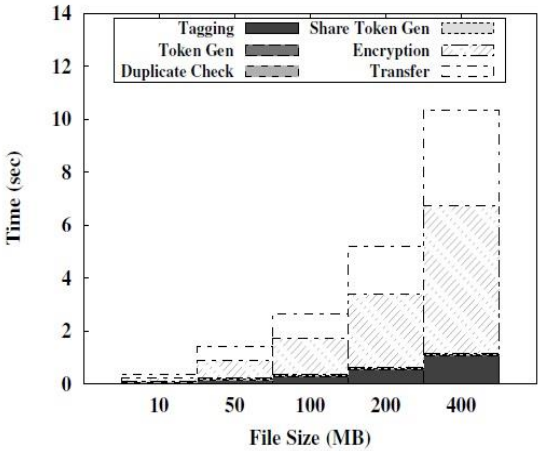

Fig. 2. Time Breakdown for Different File Size.

B. Number of Stored Files

We upload 10000 10 MB unique files to the system for recording the break down time for evaluating the impact on performance by Number of Stored Files. From Fig. every steps remains constant along the line. In case of a collision linear search is used. Even Though we use linear search, the break down time remains stable.

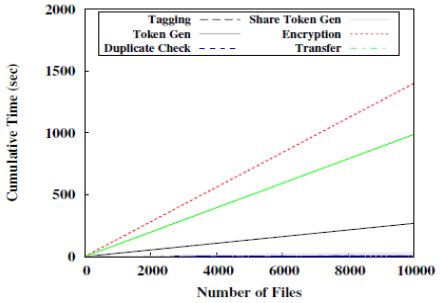

Fig. 3. Time Breakdown for Different Number of Stored Files.

VII. CONCLUSION

In this paper, In order ti provide the data security, the idea of authorized data deduplication was proposed. The concept of defferential privillages is used in deduplication checks. A private cloud is being used for performing all of key generation, encrypltion/ decryption processes. A public cloud is used for storing the encrypted data. We showed that our prortotype incurs minimal overhead.

REFERENCES

[1] OpenSSL Project. http://www.openssl.org/.
[2] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In *Proc. of USENIX LISA*, 2010.
[3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Server aided encryption for deduplicated storage. In *USENIX Security Symposium*, 2013.
[4] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In *EUROCRYPT*, pages 296– 312, 2013.
[5] M. Bellare, C. Namprempre, and G. Neven. Security proofs for Identity-based identification and signature schemes. *J. Cryptology*, 22(1):1–61, 2009.
[6] 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.
[7] 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.
[8] 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.
[9] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conf.*, 1992.
[10] GNULibmicrohttpd. http://www.gnu.org/ software/ libmicrohttpd/.
[11] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg. 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.
[12] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In *IEEETransactions on Parallel and Distributed Systems*, 2013.
[13] libcurl. http://curl.haxx.se/libcurl/.
[14] C. Ng and P. Lee. Revdedup: A reversededupli--cation storage system optimized for reads to latest backups. In *Proc. of APSYS*, Apr 2013.
[15] W. K. Ng, Y. Wen, and H. Zhu. Private data deduplication protocols in cloud storage. In S. Ossowski and P. Lecca, editors, *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 441–446. ACM, 2012.

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

[17] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In *Proc. USENIX FAST*, Jan 2002.

[18] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In *3rd International Workshop on Security in Cloud Computing*, 2011.

[19] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman.Role-based access control models. *IEEE Computer*, 29:38–47, Feb 1996.

[20] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. In *Technical Report*, 2013.

[21] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller. Secure data deduplication. In *Proc. of StorageSS*, 2008.

[22] Z. Wilcox-O'Hearn and B. Warner. Tahoe: the least-authority filesystem. In *Proc. of ACM StorageSS*, 2008.

[23] J. Xu, E.-C. Chang, and J. Zhou. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In *ASIACCS*, pages 195–206, 2013.

[24] J. Yuan and S. Yu. Secure and constant cost public cloud storage auditing with deduplication. *IACR Cryptology ePrint Archive*, 2013:149, 2013.

[25] K. Zhang, X. Zhou, Y. Chen, X.Wang, and Y. Ruan. Sedic: privacy aware data intensive computing on hybrid clouds. In *Proceedings of the 18th ACM conference on Computer and communications security*.