International Journal of Engineering &Technology, 7 (3.12) (2018) 437-438

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Encrypted De-duplication over Distributed Cloud Server

R. Aditya Balaji[1*], R. Pragadeeeshwaran[2], G. K. Sandhia[3]

[1,2,3]Department of Computer Science and Engineering, SRM University, Kattankulathur, Tamilnadu, India
**Corresponding Author E-mail: [1]adityabalaji96@gmail.com, [2]rvpragadeesh@gmail.com*
[3]sandhia.g@ktr.srmuniv.ac.in

Abstract

The most common cloud service is Data Storage. In order to reduce the storage space, deduplication is used. Data deduplication is a process of removing redundant copies of same data. If a file which is already present in the cloud, is uploaded by the same user or different user, then it will not be uploaded again. Therefore storage required is decreased but reliability is also reduced. Data are encrypted and stored in cloud to protect the privacy of users and this introduces new challenges. The proposed system uses M3 algorithm for encryption and Chunking technique for deduplication. The results of the evaluation show that the security and reliability are increased in the proposed scheme.

Keywords: *Deduplication, distributed cloud, Chunking, M3 Algorithm*

1. Introduction

Cloud is of three types. The cloud in which the services are available for general use is called public cloud. The cloud in which the services are available only for selected users is called private cloud. The cloud which is a combination of both public and private cloud is called hybrid cloud. There are many deduplication strategies. Client-side de-duplication is performed by the client before uploading to server. Server-side de-duplication is performed by the server after uploaded by the client. File-level de-duplication is performed at the file level. Block-level de-duplication is performed at the block level. The blocks can be variable blocks or fixed blocks.
Bellare et al. [1] proposed message-locked encryption. The message itself is used to produce the key for encryption and decryption. However, the attacker can effectively ensure whether the target possesses a certain file by encrypting associated unencrypted version then merely comparing the output with files possessed by the target. Li et al. [2] showed the key management issue in block-level de-duplication. Data reliability is a major issue in deduplication systems.

2. Literature Survey

Mohammadamin Ajdari et.al…, [3] performed inline deduplication and minimized the high cost of the devices. However, it does not analyse the entire data to reduce the duplication.
Hyunsoo Kwon et.al…, [4] proposed a deduplication technique which eliminates redundant copies and stores just one copy of the contents. It reduces the duplicate data but it analysis small data.
Giacomo Grangia et.al…, [5] proposed a technique where data objects are partitioned into small blocks, called chunks, and hashing techniques are used to compare the chunks and identify . Since hashing technique is used, if the data has no link, it may not reduce the duplicate data.
Yulin Wu et.al…, [6] proposed a system that combines deduplication and dynamic data operations to preserve privacy but it is not secure and efficient.

3. Proposed System

To perpetuate Data Integrity the emphasis on reliability for deduplication systems is done in the proposed system. Distributed systems are being used to improvise the margin of error for deduplication systems.
To realize Confidentiality, integrity and availability, M3 algorithm for encryption process, with the distributed storage systems. For the deduplication part, a hash value by the means of a cryptographic hash is generated through the content which is to be done and sent as fragments and stored at different servers. Hash value only for the first upload is done and distributed at such and such level and the forthcoming users can depend on the previous uploaded values. The incoming data identifiers are compared with the stored data in the storage system.
Recognizing highlight of our proposition is that data integrity and reliability can be accomplished. System constructions apply for both file-level and block-level deduplication. Implementation are done by using the M3 encryption algorithm that enables high Confidentiality, integrity and availability. The process can be executed at either the client or at the Cloud provider. The encrypted data is stored at the Cloud service provider.

4. Architecture

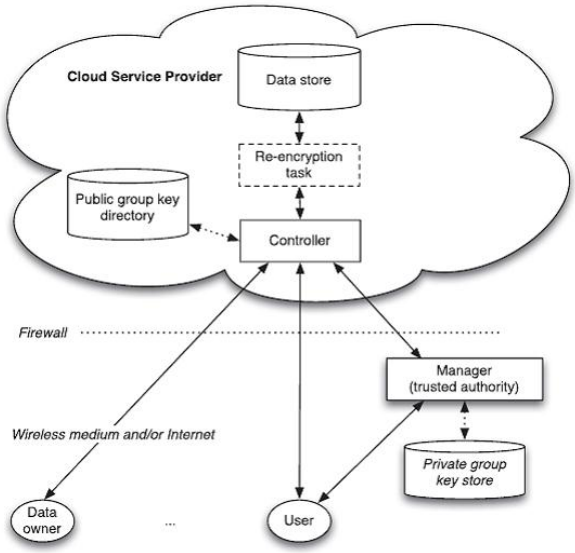

Fig. 1: System Architecture

The system of Architecture would explain about the data flow and existence of an entities etc. The order of data flow are user would upload in cloud database. Administrator would analyse the data and dealt up to the content in database.

5. Algorithms

A. M3 Algorithm

The algorithm is based on self-mutation of keys. To encrypt the key it requires a mapping array. An encrypted key produces the mapping array and the key cannot be encrypted without a mapping array. The clear text key or static hard-coded array (less secure) could produce the mapping array.
The clear text key produces the mapping array. The mapping array encrypts the clear text key. The newly encrypted key produces a new mapping array. The newly created mapping array encrypts the new key. The key self-mutates many times and the mapping array is recreated many times before the last cipher-text is produced.

B. Chunking

Data are broken down into pieces called “chunks”. A unique hash identifier identifies each chunk by comparing the chunks with previously stored chunks and checks for duplication.

6. Conclusion

In cloud, the user stores many copies of the same data. This reduces the storage space for other users. De-duplication is used to solve the problem. We propose another protected distributed deduplication framework in which the data pieces are circulated over numerous cloud servers based on M3 algorithm and Chunking technique. The primary thought is that we can confine the harm of stolen information if we can diminish the estimation of that stolen data to the attacker. For future upgrade, we wanted to execute hash Key Generation, token generation strategy by utilizing diverse methods to guarantee that the information has not changed because of unplanned debasement and confirmation of information should safeguard.

References

[1] Bellare M., Keelveedhi S., Ristenpart T. “Message-Locked Encryption and Secure Deduplication” Springer, 2013.
[2] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee and W. Lou, "Secure Deduplication with Efficient and Reliable Convergent Key Management," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 6, pp. 1615-1625, June 2014.
[3] M. Ajdari, P. Park, D. Kwon, J. Kim and J. Kim, "A Scalable HW-Based Inline Deduplication for SSD Arrays," in IEEE Computer Architecture Letters, vol. 17, no. 1, pp. 47-50, Jan.-June 1 2018.
[4] H. Kwon, C. Hahn, D. Koo and J. Hur, "Scalable and Reliable Key Management for Secure Deduplication in Cloud Storage," 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, CA, 2017, pp. 391-398.
[5] G. Grangia, Q. Xu, A. Bianco and P. Giaccone, "Balancing the Storage in a Deduplication Cluster," 2017 International Conference on Networking, Architecture, and Storage (NAS), Shenzhen, 2017, pp. 1-4.
[6] Y. Wu, Z. L. Jiang, X. Wang, S. M. Yiu and P. Zhang, "Dynamic Data Operations with Deduplication in Privacy-Preserving Public Auditing for Secure Cloud Storage," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, 2017, pp. 562-567.
[7] Z. Yan, W. Ding, X. Yu, H. Zhu and R. H. Deng, "Deduplication on Encrypted Big Data in Cloud," in IEEE Transactions on Big Data, vol. 2, no. 2, pp. 138-150, June 1 2016.
[8] Z. Yan, M. Wang, Y. Li and A. V. Vasilakos, "Encrypted Data Management with Deduplication in Cloud Computing," in IEEE Cloud Computing, vol. 3, no. 2, pp. 28-35, Mar.-Apr. 2016.