© 2015 IJSRSET | Volume 1 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Data De-Duplication Correction and Secure Key Authentication in Cloud

C. Pradeesh Kumar, G. Arunnath , K. Jegadesan , S. Harish
Velammal Institute of Technology
Velammal Knowledge Park, Kalkata High Road, Panchetti, Tamilnadu, India

ABSTRACT

Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to perform secure deduplication in cloud storage. Although convergent encryption has been extensively adopted for secure deduplication, a critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. This paper makes the first attempt to formally address the problem of achieving efficient and reliable key management in secure deduplication. We first introduce a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. To this end, we propose Dekey, a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments.

Keywords: WWW, component, formatting, style, styling, insert, Deduplication, proof of ownership, convergent encryption

I. INTRODUCTION

The main task of this project is to achieve efficient and reliable key management in secure de-duplication. Our aim is to outsource the convergent keys to cloud securely for encryption instead of unique Master Key. Deduplication improves Storage and bandwidth efficiency is incompatible with traditional encryption.

In traditional model encryption requires different users to encrypt their own data with their own master key, thus identical data copies of different users will lead to different cipher texts, making de-duplication impossible. Each such copy can be defined based on different granularities: it may refer to either a whole file (i.e., file level deduplication), or data block (i.e., block-level deduplication).To applying deduplication to user data to save maintenance cost in cloud.

Apart from normal encryption and decryption process we have proposed Master key concept with Dekey concept. For Encryption and Decryption we have used Triple Data Encryption Standard Algorithm where the plain text is Encrypted triple times with the key so that the data is secure and reliable from hackers. We reduced the cost and time in uploading and downloading with storage space.

II. METHODS AND MATERIAL

System Architecture

The system architecture comprises seven modules:

1. Mastering File to Cloud Service Provider.
2. Chunking the file chosen.
3. Dekey Based Encryption.
4. Hash value based Decryption.

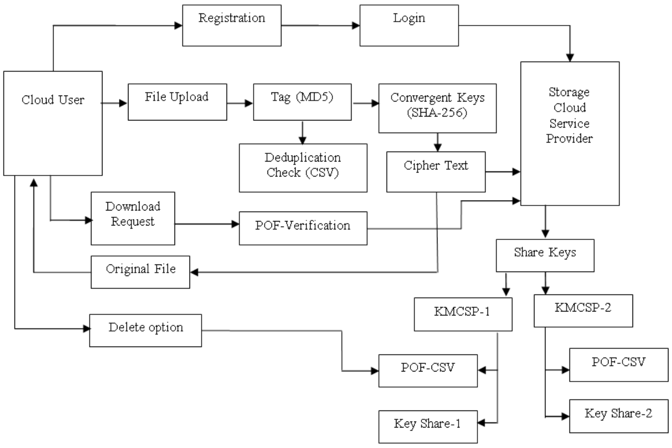

Figure 1 : System Architecture

System Description

The major functional components are briefly described below:

A. **Mastering File to Cloud: Service Provider**
B.

A user is an entity who wants to outsource data storage to the storage cloud service provider (S-CSP) and access the data later. User registers to the cloud server with necessary information and login cloud page for uploading the file. User chooses the file and uploads to server where the server store the file in rapid storage system and file level de-duplication is checked. We tag the file by using MD5 message-digest algorithm is cryptographic hash function producing a 128-bit hash value typically expressed in text format as 32 digit hex value so that files of same are de-duplicated.

B. **Chunking the file chosen**

Chunking the file chosen of fixed size and generating tags for each blocks chunked. After that generate convergent keys for each blocks split to verify block level deduplication. Here we provide filename and password for file authorization in future. Encrypt the blocks by Triple Data Encryption Standard (3DES) algorithm. Here the plain text is encoded triple times with convergent key and so the while decoding the original content it also need the same key to decode again by triple times. Finally the original content is encrypted as cipher text and stored in Storage Cloud Service Provider (S-CSP) file storage system.

C. De-key Based Encryption

After encryption the convergent keys are securely shared with cloud service provider to Key Management Cloud Service Provider (KMCSP). Key management server checks duplicate copies of convergent keys in KMCSP.

Key Management Server maintains Comma Separated Values (CSV) file to check proof of verification and store keys secure. The different users who share the common keys are referred by their own ownership. User request for deletion definitely need to prove proof of ownership to delete own contents.

D. Hash value Based Decryption

The final model where the user request for the downloading their own document which they have been upload and stored in cloud server. This download request needs proper ownership verification of the document here we create the ownership by unique tag generated by MD5 algorithm and verifies existing tag of user. After verification the original content is decrypted by requesting the cloud server where cloud server request key management server for keys to decrypt and finally the original content is received by the user. The delete request will delete only the reference of the content shared by common users and not the whole content.

E. Dekey

De-key is designed efficiently and reliably to maintain convergent keys. Its idea is to enable de-duplication in convergent keys and distribute the convergent keys across multiple Key Management Cloud Service Provider (KMCSP).

III. CONCLUSION AND FUTURE

We have shown the concept of deduplication effectively and security is achieved by means of Proof Of Ownership of the file. De-Duplication improves Storage and bandwidth efficiency is incompatible with traditional encryption. Instead of using normal encryption and decryption we use Triple DES Technique as the plain text is encrypted triple times with the convergent key so that our data will be secured. Cost efficiency is achieved as multiple users of same date is just referred and not newly added. Deleting content of shared file of different user will allow deleting only convergent keys references not content stored in server.

IV. LITERATURE SURVEY

[1] NIST's Policy on Hash Functions, Sept. 2012. [Online]. Available: http://csrc.nist.gov/groups/ST/hash/policy.html.

[2] AmazonCase Studies. [Online]. Available: https://aws.amazon. com/solutions/case-studies/#backup.

[3] P. Anderson and L. Zhang, ''Fast and Secure Laptop Backupswith Encrypted De-Duplication,'' in Proc. USENIX LISA, 2010, pp. 1-8.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, ''Message-Locked Encryption and Secure Deduplication,'' in Proc. IACR Cryptology ePrint Archive, 2012, pp. 296-3122012:631.

[5] G.R. Blakley and C. Meadows, ''Security of Ramp Schemes,'' in Proc. Adv. CRYPTO, vol. 196, Lecture Notes in Computer Science, G.R. Blakley and D. Chaum, Eds., 1985, pp. 242-268.

V. REFERENCES

[6] A.T. Clements, I. Ahmad, M. Vilayannur, and J. Li, ''Decentralized Deduplication in San Cluster File Systems,'' in Proc. USENIX ATC, 2009, p. 8.

[7] 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 Proc. ICDCS, 2002, pp. 617-624.

[8] J. Gantz and D. Reinsel, The Digital Universe in 2020: Big Data, Bigger Digital Shadows, Biggest Growth in the Far East, Dec. 2012. [Online]. Available: http://www.emc.com/collateral/analystreports/ idc-the-digital-universe-in-2020.pdf.

[9] R. Geambasu, T. Kohno, A. Levy, and H.M. Levy, ''Vanish: Increasing Data Privacy with Self-Destructing Data,'' in Proc. USENIX Security Symp., Aug. 2009, pp. 316-299.