ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 6, July 2017

A Secure Deduplication of data with Encryption in Cloud Storage

Mamatha G[1] & Srivinay [2]

PG Student, Department of Computer science & Engineering, S V I T, Bengaluru, India[1]

Assistant Professor, Department of Computer science & Engineering, S V I T, Bengaluru, India[2]

ABSTRACT: With the continuous and exponential increase of the number of users and the size of their data, data deduplication becomes more necessary for cloud storage providers. By storing a copy of duplicate data, cloud providers reduce their storage and data transfer costs. We propose ClouDedup,a secure and efficient storage service which assures deduplication and data confidentiality. Based on convergent encryption, ClouDedup remains secure thanks to the definition of a component that implements an additional encryption operation and an access control mechanism.we suggest to include a new component in order to implement the key management for each block together with the actual deduplication operation. We show that the overhead introduced by these new components is minimal and does not impact the storage and computational costs.

KEYWORDS: Convergent encryption, Deduplication, Metadata manager.

I. INTRODUCTION

The potentially infinite storage space offered by cloud providers, users tend to use as much space as they can and vendors constantly look for techniques aimed to minimize redundant data and maximize space savings. A technique which has been widely adopted is cross-user deduplication. Deduplication has proved to achieve high space and cost savings and many cloud storage providers are currently adopting it. Deduplication can reduce storage needs by up to90-95% for backup applications and up to 68% in standard file systems [1].

Along with low ownership costs and flexibility, users require the protection of their data and confidentiality guarantees through encryption. Unfortunately, deduplication and encryption are two conflicting technologies. While the aim of deduplication is to detect identical data segments and store them only once, the result of encryption is to make two identical data segments indistinguishable after being encrypted. This means that if data are encrypted by users in a standard way, the cloud storage provider cannot apply deduplication since two identicaldata segments will be different after encryption. On the other hand, if data are not encrypted by users, confidentiality cannot be guaranteed and data are not protected against curious cloud storage providers.

A technique which has been proposed to meet these two conflicting requirements is convergent encryption [2], [3], [4] whereby the encryption key is usually the result of the hash of the data segment. Although convergent encryption seems to be a good candidate to achieve confidentiality and deduplication at the same time, it unfortunately suffers from various well-known weaknesses [5], [6] including dictionary attacks.

In this paper, we cope with the inherent security exposures of convergent encryption and propose ClouDedup, which preserves the combined advantages of deduplication and convergent encryption.Fig.1shows the system architecture. To summarize our contributions:

- ClouDedup assures block-level deduplication and data confidentiality.
- ClouDedup preserves confidentiality and privacy even against potentially malicious cloud storage providers thanks to an additional layer of encryption;
- ClouDedup offers an efficient key management solution through the metadata manager;
- The new architecture defines several different components and a single component cannot compromise the whole system without colluding with other components;
- ClouDedup works transparently with existing cloud storage providers.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 6, July 2017

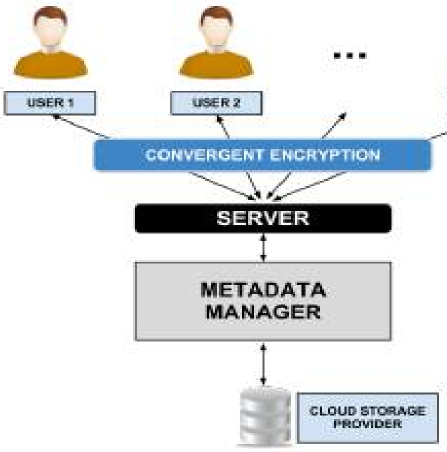

Fig.1: System architecture

II. CLOUDEDUP

The scheme ClouDedup aims at deduplication at the level of blocks of encrypted files while coping with the inherent security exposures of convergent encryption. The scheme consists of two basic components: a server that is in charge of access control and that achieves the main protection against COF and LRI attacks; another component, metadata manager (MM), is in charge of the actual deduplication and key management operations.

A. The Server

A simple solution to prevent the attacks against convergent encryption (CE) consists of encrypting the ciphertexts resulting from CE with another encryption algorithm using the same keying material for all input. This solution is compatible with the deduplication requirement since identical ciphertexts resulting from CE would yield identical outputs even after the additional encryption operation.

B. User

The role of the user is limited to splitting files into blocks, encrypting them with the convergent encryption technique, signing the resulting encrypted blocks and creating the storage request. The user encrypts each key derived from the corresponding block with the previous one and his secret key in order to outsource the keying material as well and thus only store the key derived from the first block and the file identifier. For each file, this key will be used to decrypt and re-build the file when it will be retrieved.

C. Metadata Manager (MM)

MM is the components responsible for storing metadata, which include encrypted keys and block signatures, and handling deduplication. MM maintains a linked list and a small database in order to keep track of file ownerships file composition and avoid the storage of multiple copies of thesame data segments. The tables used for this purpose are file, pointer and signature tables.

The tables used by MM are structured as follows:

- The file table contains the file id, file name, user id and the id of the first data block.
- The pointer table contains the block id and the id of the block stored at the cloud storage provider.
- The signature table contains the block id, the file id and the signature

D. Cloud Storage Provider (SP)

SP is the most simple component of the system. The only role of SP is to physically store data blocks. SP is not aware of the deduplication and ignores any existing relation between two or more blocks. ClouDedup is completely transparent from SP’s perspective, which does collaborate with MM for deduplication. It is possible to make use of well-known cloud storage providers such as Google Drive, Amazon S3 and Dropbox.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Special Issue 6, July 2017

III. PROTOCOL

In this section we describe the two main operations of ClouDedup: storage and retrieval.

A. Storage

During the storage procedure, a user uploads a file to the system. As an example, we describe a scenario in which a user Uj wants to upload the file F1.

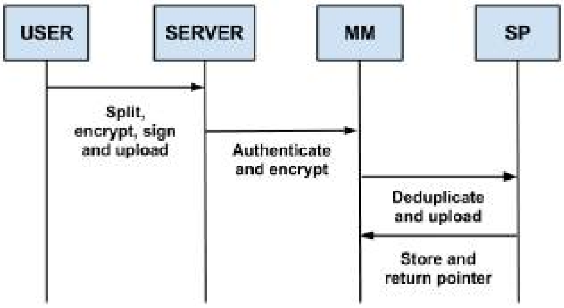

Fig.2: Storage Protocol

USER User Uj splits F1 into several blocks. For each block Bi, Uj generates a key by hashing the block and uses this key to encrypt the block itself.File identifiers are generated by hashing the concatenation of user ID and file name.
SERVER The server receives a request from user Uj and runs SSL in order to authenticate Uj and securely communicate. Each key, signature and blocksare encrypted in the server. The only parts of the request which are not encrypted are user's id, the file name and the file identifier. The server forwards the new encrypted request to MM.
MM MM receives the request from the server and for each block contained in the request, MM checks if that block has already been stored by computing its hash value and comparing it to the ones already stored.
SP SP receives a request to store a block. After storing it, SP returns the pointer to the block.
MM MM receives the pointer from SP and stores it in the pointer table.

A. Retrieval

During the retrieval procedure, a user asks to download a file from the system. As an example, we describe a scenario in which a user Uj wants to download the file F1.

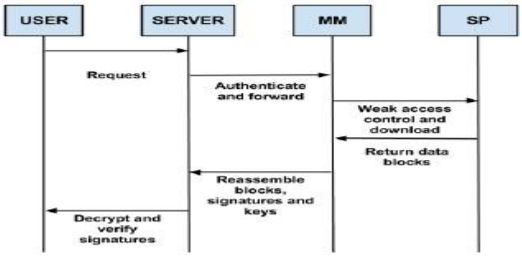
USER
SERVER
MM
SP
Request
Authenticate and forward
Weak access control and download
Return data blocks
Reassemble blocks, signatures and keys
Decrypt and verify signatures

Fig.3: Retrieval Protocol

USER User Uj sends a retrieval request to the server in order to retrieve file F1. The request is composed by the user's id, the file identifier and his certificate.
SERVER The server receives the request, authenticates Uj and if the authentication does not fail, the server forwards the request to MM without performing any encryption.
MM MM receives the request from the server and analyzes it in order to check if Uj is authorized to access File id.If the user is authorized, MM looks up the file identifier in the file table in order to get the pointer to the first block of the file. Then, MM visits the linked list in order to retrieve all the blocks that compose the file. For each of these blocks, MM retrieves the pointer from the pointer table and sends a request to SP.
SP SP returns the content of the encrypted blocks to MM.

MM MM builds a response which contains all the blocks, keys and signatures of file F1. Signatures are retrieved from the signature table. MM sends the response to the server.
SERVER The server decrypts blocks, signatures and keys. If the signature verification does not fail, the server sends a response to Uj.
USER Uj finally decrypt blocks and keys. Uj already knows the key corresponding to the block.

VI. EXPERIMENTAL RESULTS

In Fig.4 we show that the overhead introduced by the MM component is minimal and does not affect space savings of deduplication. These results prove that the overhead for block-level deduplication is affordable even with encryption.

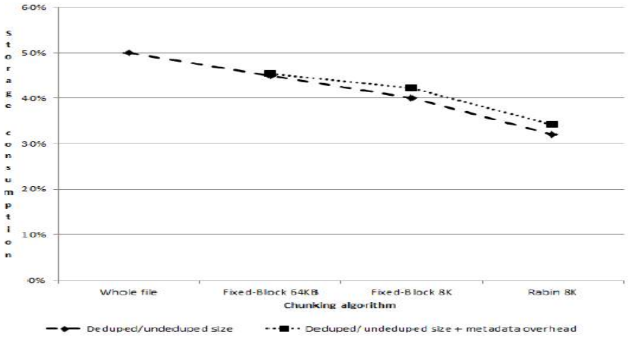

Fig.4: Overhead of metadata management with encryption

V.CONCLUSION

We designed a system which achieves confidentiality and enables block-level deduplication at the same time. Our system is built on top of convergent encryption. We showed that it is worth performing block-level deduplication instead of file level deduplication since the gains in terms of storage space are not affected by the overhead of metadata management, which is minimal.

REFERENCES

[1] Dutch T Meyer and William J Bolosky. A study of practical deduplication. ACMTransactions on Storage (TOS), 7(4):14, 2012.
[2] John R Douceur, Atul Adya, William J Bolosky, P Simon, and Marvin Theimer. ReclaimingSpace from duplicate files in a server less Distributed files system. In Distributed ComputingSystems, 2002 Proceedings. 22nd International Conference on, pages 617–624. IEEE, 2002
[3] John Pettitt.http://cypherpunks.venona.com/date/1996/02/msg02013.html.
[4] The Freenet Project Freenet. https://freenetproject.org/.
[5] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. Message-locked encryption and secure deduplication. In Advances in Cryptology–EUROCRYPT 2013, pages 296–312Springer, 2013.
[6] Perttula. Attacks on convergent encryption. http://bit.ly/yQxyvl.
[7] Zooko Wilcox-O'Hearn and Brian Warner. Tahoe: the least-authority file system. InProceedings of the 4th ACM international workshop on Storage security and survivability,Pages 21–26, ACM, 2008.
[8] Landon P Cox, Christopher D Murray, and Brian D Noble. Pastiche making backup cheap and easy. ACM SIGOPS Operating Systems Review, 36(SI):285–298, 2002.
[9] Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg. Side channels in cloud Services: Deduplication in cloud storage. Security & Privacy, IEEE, 8(6):40–47, 2010.