ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 5, Issue 12, December 2016

Survey Paper on: File Level and Block Level De-duplication

Jayashree Bhosale, Bhagyashree Kshirsagar,Priyanka Misal,Monica Tiple

Abstract

Data de-duplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. However, there is only one copy for each file stored in cloud even if such a file is owned by a huge number of users. As a result, De-Duplication system improves storage utilization while reducing reliability. Approach is to block-level message locked encryption, can achieve file-level and block-level de-duplication, block key management, and proof of ownership simultaneously using a small set of metadata. We also show that our BL-MLE scheme can be easily extended to support proof of ownership, which makes it for secure cloud storage.

Keywords

De-duplication, message locked encryption, cloud storage, proof of ownership

I. INTRODUCTION

The purpose of the design is to secure de-duplication systems with higher reliability in cloud computing. We introduce the distributed cloud storage servers into de-duplication systems to provide better fault tolerance. To further protect data confidentiality. Data duplication removes the redundancy and replication, but confidentiality and secrecy of data becomes important issue. Data de-duplication stores only one redundant copy of data and provides link to that copy. Two popular available data de-duplication strategies are file wise de-duplication and block level de-duplication. Secure data de-duplication allows cloud storage and service providers to employ data with confidentiality. Cloud environment offers scalable and elastic storage capabilities to users. Data de-duplication is widely used technique to reduce the storage spaces and saving the bandwidth. Data de-duplication stores only one copy of redundant data, and provides link to that copy instead of storing again. It acts as a key component in backup process. De-duplication saves both the disk space and network bandwidth.

Data de-duplication strategies classified as 1) File level de-duplication- Here only a single copy of each file is stored on the server. 2) Block level de-duplication- Each file is divided into blocks and only single copy of each block is stored [1]. Block size can be either fixed or variable. From the architecture perspective de-duplication can be target based or source based.1) Source based de-duplication- Before uploading data user first sends identifier to the server for redundancy checking. 2) Target based de-duplication- Users are unaware of de-duplication, they just upload the files to the storage server. In this paper, we are focusing on de-duplication with security. But, encryption and de-duplication are conflicting technologies. De-duplication detects identical segments and storing them only once, while encryption makes two identical data segments indistinguishable after encryption [8][9]. Convergent encryption technique has been proposed to meet these conflicting requirements. Message locked encryption technique uses content hash keying [2].There are some basics concepts for understanding which are as follows-De-duplication: The elimination of duplicated data, commonly known as De-duplication, is now widely accepted for reducing the amount of storage space used in backup systems and operating system image servers in cloud infrastructures[5][6][7].

II. RELATED WORK

In this paper, we show how to design secure de-duplication systems with higher reliability in cloud computing. We introduce the distributed cloud storage servers into de-duplication systems to provide better fault tolerance. In de-duplication process first file is divided into smaller units and this mechanism is known as chunking. Small units are called as chunks. Whole file, fixed size and variable size chunks are three basic approaches for de-duplication. In whole file strategy file's hash value is computed and used as identifier. Two files having same hash value considered as identical and only one copy is stored. Such form is known as content addressable storage and used in EMC center systems [3]. In second approach files are divided into fixed size blocks before de-duplication. Third one is most suitable approach in which files are break into variable size chunks by manipulating hash value for particular file. Message locked encryption is used to encrypt and decrypt file. User can derive the message key from each original data copy, then using that key encrypt data file. Also user derives tag for data copy to check duplicate data. If tag is same then both files are same. Both message key and tag are independently derives. Message locked encryption [2], [3] also known as content hash keying, is used to produces identical cipher text from identical plaintext files. The simplest implementation of message encryption can be defined as: Alice derives the encryption key from her file F such that K = H(F), where H is a cryptographic hash function. Message Locked encryption scheme can be defined with four primitive functions:

- KeyGen(M) →K is the key generation algorithm that maps a data copy M to a message lock key K;
- Enc(K, M) ->C is the symmetric encryption algorithm that takes both the message lock key K and the data copy M as inputs and then outputs a ciphertext C;
- Dec (K, C) ->M is the decryption algorithm that takes both the ciphertext C and the message lock key K as inputs and then outputs the original data copy M ;
- TagGen(M) -> T (M) is the tag generation algorithm that maps the original data copy M and outputs a tag T (M).

PROOF OF OWNERSHIP

Proof of ownership (POW) of the file for a client to prove to the server that it indeed has the file. We achieve the efficient goal by relying on dynamic spot checking, in which the client only needs to access diminutive but dynamic portions of the pristine file to engender the proof of possession of the pristine file, thus greatly reducing the encumbrance of computation on the client and minimizing the I/O between the client and the server. Concurrently, by utilizing dynamic coefficients and desultorily culled indices of the pristine files, our scheme commixes the arbitrarily sampled portions of the pristine file with the dynamic coefficients to engender the unique proof in every challenge. This technique guarantees the provable- security goal.

Proof of ownership (POW) [4] is a protocol enables users to prove their ownership of data copies to the storage server. POW is implemented as an interactive algorithm run by user and storage server act as prove and verifier. The verifier derives a short value ϕ(M)from a data copy M . To prove the ownership of the data copy M, the approver needs to send ϕ to the verifier such that ϕ= ϕ(M).

BLOCK-LEVEL MESSAGE-LOCKED ENCRYPTION

We now describe the definition of Block-Level Message- Locked Encryption (BL-MLE) for DLSB-de-duplication. It will capture additional functionalities including dual level de-duplication, block keys management, and proof of ownership, compared with a normal file-level MLE.

A. Definition

Syntax: A block-level message-locked encryption scheme is specified by the following algorithm.

• Setup: takes a security parameter λ as input and returns the system parameters P.

• KeyGen: takes the public parameters P and a file message M = M[1]||...||M[n] as input, and returns a master key kmas and block keys {ki }1≤i≤n generated using the following two sub-algorithms respectively,

– M-KeyGen: takes P and M as input, returns the master key kmas;

– B-KeyGen: takes P and M[i] as input, returns the block key ki .

• Enc: takes public parameters P, a block message M[i] and the corresponding block key ki as input, returns the block ciphertext C[i].

• Dec: takes public parameters P, a block ciphertext C[i] and a block key ki as input, returns M[i] or ⊥.

• TagGen: takes public parameters P and a fileM as input, returns the file tag T0 and block tags {Ti } 1≤i≤n generated using the following two sub-algorithms respectively,

– F-TagGen: takes P and M as input, returns the file tag T0;

– B-TagGen: takes P, M and the block index i as input, returns the block tag Ti .

• ConTest: takes a block tag Ti and a block cipher text C[i] as input, returns True or False.

• EqTest: takes as input two block tags T , T _ and the corresponding file tags T0, T _ 0, returns True or False.

• B-KeyRet: takes a master key kmas, a block tag Ti , and a block cipher text C[i] as input, returns a block key ki /⊥.

• PoWPrf: takes a challenge Q and a file M as input, returns a response P.

• PoWVer: takes a challenge Q, the file tag T0, the block tags {Ti }1≤i≤n, and the response P as input, returns True or False.

Remark 1: Unlike a standard MLE scheme, our BL-MLE scheme performs encryption per block. Given a file, we split it into blocks before encryption. The KeyGen algorithm produces the master key and block keys using the sub-algorithm M-KeyGen and B-KeyGen respectively. The former is used to encrypt block keys while the latter are derived from the file blocks and used to encrypt and decrypt block messages. For the tag generation algorithm TagGen, we also define two sub-algorithms, F-TagGen and B-TagGen. The file tag can be used as the file identifier for file-level de-duplication, and the block tag severs multi-purposes, including block identifier, encrypted block key, and PoW tag.

III.LITERATURE SURVEY

With the explosive growth of digital data, de-duplication techniques are widely employed to backup data and minimize network and storage overhead by detecting and eliminating redundancy among data. Instead of keeping multiple data copies with the same content, de-duplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy[11]. De-duplication has received much attention from both academia and industry because it can greatly improves storage utilization and save storage space, especially for the applications with high de-duplication ratio such as archival storage systems[11][12].

For example, a file or volume that's backed up every week creates a significant amount of duplicate data. De-duplication algorithms analyze the data and store only the compressed, unique segments of a file.

IV.PROBLEM STATEMENT

In order to reduce the burden of maintaining big data, more and more enterprises and organizations have chosen to outsource data storage to cloud storage providers. This makes data management a critical challenge for the cloud storage providers. To achieve optimal usage of storage resources, many cloud storage providers perform de-duplication, which exploits data redundancy and avoids storing duplicated data from multiple users.

V.GOALS AND OBJECTIVES

- Improving storage space on cloud.
- Provide redundancy avoidance to the cloud server data.
- Our goal to remove duplicated data on cloud.
- Provide security for large data files.
- Provide Proof of ownership then decoy data upload.
- To protect original data from unauthorized owner.

VI.STATEMENT OF SCOPE

Our new secure de-duplication systems are proposed to provide efficient de-duplication with high reliability for file-level and block-level de-duplication, respectively. The message locked encryption methods, is utilized to protect data confidentiality. Specifically, data are split into fragments and stored at different servers. Our proposed constructions support both file-level and block-level de-duplications.

VII.MATHEMATICAL MODULE

To generate a file token, Let ϕ F;p = TagGen(F, kp) , the token ϕ' F;p we define a binary relation R = f((p, p′)g as follows, Given two privileges p and p ′, we say that p matches p′ if and only if R(p, p′) = 1. The user computes and sends file token ϕ'F;p= TagGen(F, kp) for all p 2 PF if no duplicate is found, the user computes the encrypted file CF = Enc (kF , F) with the message locked key k F = KeyGenCE (F) and uploads(CF, fϕ'F;pg) to the cloud server, a given ciphertext C is drawn from a message space S = {F1, …... , Fn}g of size n, the public cloud server can recover F after at most n off-line encryptions. That is, for each i = {1, ….. , n} it simply encrypts {Fi} to get a cipher text denoted by Ci . If C = Ci , it means that the fundamental file is Fi.

VIII.CONCLUSIONS & FUTURE WORK

We designed a system which achieves confidentiality and enables block-level de-duplication at the same time. Our system is built on top of convergent encryption. We showed that it is worth performing block-level de-duplication instead of file level de-duplication since the gains in terms of storage space are not affected by the overhead of metadata management, which is minimal. Additional layers of encryption are added by the server. Thanks to the features of these components. As the additional encryption is symmetric, the impact on performance is negligible. We also showed that our design, in which no component is completely trusted, prevents any single component from compromising the security of the whole system. Our solution also prevents curious cloud storage providers from inferring the original content of stored data by observing access patterns or accessing metadata. Furthermore, we showed that our solution can be easily implemented with existing and widespread technologies. Finally, our solution is fully compatible with standard storage APIs and transparent for the cloud storage provider, which does not have to be aware of the running de-duplication system. Therefore, any potentially non trusted cloud storage provider such as Amazon, Dropbox and Google Drive, can play the role of storage provider. As part of future work, ClouDedup may be extended with more security features such as proofs of retrievability [7], data integrity checking [9] and search over encrypted data [8].

In this paper we mainly focused on the definition of the two most important operations in cloud storage are storage and retrieval. We plan to define other typical operations such as edit and delete. After implementing a prototype of the system, we aim to provide a full performance analysis. Furthermore, we will work on finding possible optimizations in terms of bandwidth, storage space and computation.

References

[1] Harnik D.Benny Pinkas Alexandra Shulman-Peleg. Cloud Computing: Side Channels in cloud services. IEEE Security and Privacy, 8(6):40–47.

[2] Pasquale Puzio Refik Molva Melek Sergio Onen L. ClouDedup: Secure De-duplication with Encrypted Data for cloud storage. In Proceeding in 2013 IEEE International Conference on Cloud Computing Technology and Science.

[3] S. Annapureddy, M. J. Freedman, and D. Mazières. Shark: Scaling file servers via cooperative caching. In Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI), pages 129–142, 2005.

[4] 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.

[5] Zooko Wilcox-O'Hearn and Brian Warner. Tahoe: the least-authority file system. In Proceedings of the 4th ACM international workshop on Storage security and survivability, pages 21–26. ACM,2008.

[6] Michael O Rabin. Fingerprinting by random polynomials. Center for Research in Computing Techn., Aiken Computation Laboratory, Univ., 1981.

[7] 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.

[8] Is Convergent Encryption really secure? http://bit.ly/Uf63yH.

[9] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. Message locked encryption and secure de-duplication. In Advances in Cryptology–EUROCRYPT 2013, pages 296–312. Springer, 2013.

*[10]*Pointcheval, D., Johansson, T. (eds.): Advances in Cryptology | EUROCRYPT 2012, 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7237. Springer (2012)

*[11]*Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: De_nitional equivalences and constructions without random oracles. In: Wagner [31], pp. 360{378}

[12] Bellare, M., Keelveedhi, S., Ristenpart, T.: Message-locked encryption and secure de-duplication. In: Johansson and Nguyen [21], pp. 296{312}.

Jayashree Bhosale
is currently doing B.E degree at Department of Computer Engineering, Anantrao Pawar College Of Engineering & Research, Savitribai Phule Pune University Pune, Maharashtra, India.

Bhagyashree Kshirsagar
is currently doing B.E degree at Department of Computer Engineering, Anantrao Pawar College Of Engineering & Research, Savitribai Phule Pune UniversityPune, Maharashtra, India.

Priyanka Misal
is currently doing B.E degree at Department of Computer Engineering, Anantrao Pawar College Of Engineering & Research, Savitribai Phule Pune University Pune, Maharashtra, India.

Monica Tiple
is currently doing B.E degree at Department of Computer Engineering, Anantrao Pawar College Of Engineering & Research, Savitribai Phule Pune UniversityPune, Maharashtra, India.