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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

A Survey on De-duplication of Distributed Cloud Storage by Improving Confidentiality, Integrity and Reliability

Prerna Lahane, Prof. Sarika Bodake

ME Student, Dept. of Computer Engineering, P.V.P.I.T Bavdhan, Pune, Maharashtra, India

Assistant Professor, Dept. of Computer Engineering, P.V.P.I.T Bavdhan, Pune, Maharashtra, India

ABSTRACT: Data de-duplication is a technique for eliminating or removing duplicate copies of data, and also widely used in cloud computing environment to minimize storage space and network bandwidth. In this environment only one copy for each file stored in cloud which is owned by a huge number of users. Data de-duplication system improves storage utilization and reduces reliability. Also, the challenge of privacy for sensitive data also arises when they are outsourced by users to cloud. Along with addressing security challenges, we attempt to formalize the secure distributed reliable de-duplication system. Here we propose new distributed de-duplication system with more reliability in which the data chunks are distributed across multiple cloud servers. The security requirements of data confidentiality, integrity and reliability achieved by introducing a Shamir secret sharing scheme in distributed storage systems, instead of using convergent encryption as in previous de-duplication systems.

KEYWORDS: De-duplication, distributed storage system, reliability, Shamir secret sharing

I. INTRODUCTION

Data de-duplication is a technique for eliminating duplicate copies of data, and widely used in cloud storage to reduce storage space and network bandwidth. However, there is only one copy for each file stored in cloud, such a file is owned by a huge number of users. As a result, de-duplication system improves storage utilization while reducing reliability. To secure the data on the cloud is the main challenge in the cloud computing environment. Here we propose new secure distributed de-duplication systems with higher reliability in which the data chunks are distributed across multiple cloud servers. The security requirements of data confidentiality is achieved by introducing deterministic secret sharing scheme in distributed storage systems.

A number of de-duplication systems have been proposed based on various de-duplication strategies such as client-side or server-side de-duplications, file-level or block-level
De-duplications.
There are two types of de-duplication in terms of the size:

1. File level De-duplication: Which discovers redundancies between different files and removes these redundancies to minimize storage demands.
2. Block level De-duplication: Which discovers redundancies between chunks or blocks of data in the file and removes theses redundancy to reduce storage space.

Though de-duplication technique can save the storage space for the cloud storage service providers, it reduces the reliability of the system. Data reliability is actually a very critical issue in a de-duplication storage system because there is only one copy for each file stored in the server shared by all the owners. If such a shared file/chunk was lost, a disproportionately large amount of data becomes inaccessible because of the unavailability of all the files that share this file/chunk. If the value of a chunk were measured in terms of the amount of file data that would be lost in case of losing a single chunk, then the amount of user data lost when a chunk in the storage system is corrupted grows with the number of the commonality of the chunk. Thus, how to guarantee high data reliability in de-duplication system is a critical problem. Most of the previous de-duplication systems have only been considered in a single-server setting. However, as lots of de-duplication systems and cloud storage systems are intended by users and applications for higher

DOI: 10.15680/IJIRCCE.2015. 0311176

reliability, especially in archival storage systems where data are critical and should be preserved over long time periods. This requires that the de-duplication storage systems provide reliability comparable to other high-available systems. Furthermore, the challenge for data privacy also arises as more and more sensitive data are being outsourced by users to cloud. Encryption mechanisms have usually been utilized to protect the confidentiality before outsourcing data into cloud. Most commercial storage service provider are reluctant to apply encryption over the data because it makes de-duplication impossible. The reason is that the traditional encryption mechanisms, including public key encryption and symmetric key encryption, require different users to encrypt their data with their own keys. As a result, identical data copies of different users will lead to different cipher texts. To solve the problems of confidentiality and de-duplication, the notion of convergent encryption [4] has been proposed and widely adopted to enforce data confidentiality while realizing de-duplication. However, these systems achieved confidentiality of outsourced data at the cost of decreased error resilience. Therefore, how to protect both confidentiality and reliability while achieving de-duplication in a cloud storage system is still a challenge.

Data reliability is actually a very critical issue in a data de-duplication storage system because there is only one copy for each file stored in the server which is shared by all the owners. If such a shared file/chunk was lost, a disproportionately large amount of data becomes inaccessible because of the unavailability of the file that share this file/chunk. Now to assure high data reliability in de-duplication system is a critical issue. Here we are implementing secure distributed de-duplication system with improved reliability without use of an encryption mechanism.

Our project performs following contributions:

- The secret splitting technique, instead of traditional encryption methods, to protect data confidentiality.
- Confidentiality, reliability and integrity can be achieved in our proposed system.
- We implement our de-duplication systems using the Shamir secret sharing scheme that enables high reliability and confidentiality levels.

II. RELATED WORK

A number of de-duplication systems have been proposed based on various de-duplication strategies such as client-side or server-side de-duplications, file-level or block-level de-duplications. Another categorization criteria is the location at which de-duplication is performed: if data are de-duplicated at the client, then it is called source-based de-duplication, otherwise target-based. In source based de-duplication the client first hashes each data segment he wishes to upload and sends those results to the storage provider to check whether such data are already stored: those only “un de-duplicated” data segments will be actually uploaded by the user.

Li[2] addressed in block-level de-duplication by distributing these keys across multiple servers after encrypting the files.addressed the key-management issue in block-level de-duplication by distributing these keys across multiple servers after encrypting the files. Even though the mechanisms of the server cope with the security weaknesses of CE, the requirement for de-duplication at block-level further raises an issue with respect to key management. As an inherent feature of CE, the fact that encryption keys are derived from the data itself does not eliminate the need for the user to memorize the value of the key for each encrypted data segment. Unlike file-level de-duplication, in case of block-level de-duplication, the requirement to memorize and retrieve CE keys for each block in a secure way, calls for a fully-fledged key management solution. We thus suggest to include a new component, the metadata manager (MM), in the new de-duplication system in order to implement the key management for each block together with the actual deduplication operation. Bellare et al. showed how to protect data confidentiality by transforming the predictable message into an unpredictable message.

Stanek et al[3] explained encryption scheme that provides differential security for popular data and unpopular data. Popular data are not particularly sensitive hence the traditional conventional encryption is performed. Convergent encryption enables duplicate encrypted files to be recognized as identical, but there remains the problem of performing this identification enables duplicate encrypted files to be recognized as identical, but there remains the problem of performing this identification across a large number of machines in a robust and decentralized manner. Convergent encryption purposely discloses information. Some Other research has considered unintentional leaks through side channels such as computational timing, measured power consumption, or response to injected faults.

DOI: 10.15680/IJIRCCE.2015. 0311176

Bellare et al[5] showed how to protect the data confidentiality by transforming the predictable message into unpredictable message to enhance the security of de-duplication and protect the data confidentiality. Here, another third party called key server is introduced to generate the file tag for duplicate check. This system also formalized primitive as message locked encryption, and explored its application in space-efficient secure outsourced storage.

MihirBellare[5] et al. formalized this primitive as message-locked encryption, and explored its application in space efficient secure outsourced storage. There are also several implementations of convergent implementations of different convergent encryption variants for secure de-duplication. The basic idea of convergent encryption (CE) is to derive the encryption key from the hash of the plain text. The simplest implementation of convergent encryption can be defined as follows: Alice derives the encryption key from her message M such that K = H(M), where H is a cryptographic hash function; she can encrypt the message with this key, hence: C = E(K,M) = E(H(M),M)), where E is a block cipher. By applying this technique, two users with two identical plaintexts will obtain two identical cipher texts since the encryption key is the same; hence. the cloud storage provider will be able to perform de-duplication on such cipher texts.

Halevi et al[8] proposed to solve the problem of using a small hash value as a proxy for the entire file, we want to design a solution where a client shows to the server that it indeed has the file. The aim of this paper is to confirm that the file does not have a small representation that when leaked to an attacker allows the attacker to obtain
The file from the server. Ideally, we would like the smallest representation of the file to be as long as the amount of entropy in the file itself. The proof of ownership (PoW) for de-duplication systems so that a client can efficiently prove
To the cloud storage server that he/she owns a file without uploading the file itself.

A. *Disadvantages of Existing system:*

1. Data reliability is actually a very critical issue in a de-duplication storage system because there is only one copy for each file stored in the server shared by all the owners.

2. Most of the previous de-duplication systems have only been considered in a single-server setting.

3. The traditional de-duplication methods cannot be directly extended and applied in distributed and multi-server systems.

III. PROPOSED ALGORITHM

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. There is only one copy for each file stored in cloud, such a file is owned by a huge number of users. As a result, system improves storage utilization while reducing reliability. We propose new distributed systems with higher reliability in which the data chunks are distributed across multiple cloud servers.
Here we will 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. To further protect data confidentiality, the secret sharing technique is utilized, which is also compatible with the distributed storage systems. In more details, a file is first split and encoded into fragments by using the technique of secret sharing, instead of encryption mechanisms. These shares will be distributed across multiple independent storage servers. Furthermore, to support de-duplication, a short cryptographic hash value of the content will also be computed and sent to each storage server as the fingerprint of the fragment stored at each server. Only the data owner who first uploads the data is required to compute and distribute such secret shares, while all following users who own the same data copy do not need to compute and store these shares any more. To recover data copies, users must access a minimum number of storage servers through authentication and obtain the secret shares to reconstruct the data. In other words, the secret shares of data will only be accessible by the authorized users who own the corresponding data copy. Four 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 secret splitting technique, instead of traditional encryption methods, is utilized to protect data confidentiality. Specifically, data are split into fragments by using secure secret sharing schemes and stored at different servers.The distributed de-duplication systems’ proposed aim is to reliably store data in the cloud while

achieving confidentiality and integrity. Its main goal is to enable de-duplication and distributed storage of the data across multiple storage servers. Instead of encrypting the data to keep the confidentiality of the data, our new constructions utilize the secret splitting technique to split data into shards. These shards will then be distributed across multiple storage servers.

A. *Threat Model and Security Goals*

Two types of attackers are considered in our threat model: (i) An outside attacker, who may obtain some knowledge of the data copy of interest via public channels. An outside attacker plays the role of a user that interacts with the S-CSP;
(ii) An inside attacker, who may have some knowledge of partial data information such as the cipher text. An insider attacker is assumed to be honest-but-curious and will follow our protocol, which could refer to the S-CSPs in our system. Their goal is to extract useful information from user data. The following security requirements, including confidentiality, integrity, and reliability are considered in our security model.

1. *Confidentiality*. Here, we allow collusion among

the SCSPs. However, we require that the number of colluded S-CSPs is not more than a predefined threshold. To this end, we aim to achieve data confidentiality against collusion attacks. We require that the data distributed and stored among the S-CSPs remains secure when they are unpredictable (i.e., have high min-entropy), even if the adversary controls a predefined number of S-CSPs. The goal of the adversary is to retrieve and recover the files that do not belong to them. This requirement has recently been formalized in [6] and called the privacy against chosen distribution attack. This also implies that the data is secure against the adversary who does not own the data.

2. *Integrity*. Two kinds of integrity, including tag

Consistency and message authentication, are involved in the security model. Tag consistency check is run by the cloud storage server during the file uploading phase, which is used to prevent the duplicate/cipher text replacement attack. If any adversary uploads a maliciously-generated cipher text such that its tag is the same with another honestly-generated cipher text, the cloud storage server can detect this dishonest behavior. Thus, the users do not need to worry about that their data are replaced and unable to be decrypted. Message authentication check is run by the users, which is used to detect if the downloaded and decrypted data are complete and uncorrupted or not. This security requirement is introduced to prevent the insider attack from the cloud storage service providers.

3. *Reliability*. The security requirement of reliability

In de-duplication means that the storage system can provide
Fault tolerance by using the means of redundancy. In more details, in our system, it can be tolerated even if a certain number of nodes fail. The system is required to detect and repair corrupted data and provide correct output for the users.

B. *Building Blocks*

Secret Sharing Scheme. There are two algorithms in a Secret sharing scheme, which are Share and Recover. The secret is divided and shared by using Share. With enough shares, the secret can be extracted and recovered with the algorithm of Recover. In our implementation, we will use the Ramp secret sharing scheme (RSSS) [7], [8] to secretly split a secret into shards. Specifically, the (n, k, r)-RSSS (where $n > k > r \geq 0$) generates n shares from a secret so that (i) the secret can be recovered any k or more shares, and (ii) no information about the secret can be deduced from any r or less shares. Two algorithms, Share and Recover, are defined in the(n, k, r)-RSSS.• Share divides a secret S into $(k - r)$ pieces of equal size, generates r random pieces of the same size, and encodes the k pieces using a non-systematic k-of-n erasure code into n shares of the same size;• Recover takes any k out of n shares as inputs and then outputs the original secret S.It is known that when $r = 0$, the $(n, k, 0)$-RSSS becomes the (n, k) Rabin's Information Dispersal Algorithm (IDA) [9]. When $r = k-1$, the $(n, k, k-1)$-RSSS becomes the (n,k) Shamir's Secret Sharing Scheme (SSSS).

C. *Advantages of Proposed System:-*

1. Distinguishing feature of our proposal is that data integrity, including tag consistency, can be achieved.
2. To our knowledge, no existing work on secure de-duplication can properly address the reliability and tag consistency problem in distributed storage systems.
3. Our proposed constructions support both file-level and block-level de-duplications.
4. Security analysis demonstrates that the proposed de-duplication systems are secure in terms of the definitions specified in the proposed security model. In more details, confidentiality, reliability and integrity can be

achieved in our proposed system. Two kinds of collusion attacks are considered in our solutions. These are the collusion attack on the data and the collusion attack against servers. In particular, the data remains secure even if the adversary controls a limited number of storage servers.

We implement our de-duplication systems using the Ramp secret sharing scheme that enables high reliability and confidentiality levels. Our evaluation results demonstrate that the new proposed constructions are efficient and the redundancies are optimized and comparable with the other storage system supporting the same level of reliability.

IV. CONCLUSION

We proposed the distributed de-duplication systems to improve the reliability of data while achieving the confidentiality and integrity of the users outsourced data without any encryption mechanism. We have proposed hybrid de-duplication which is combination of file and block level data de-duplication. We implemented our de-duplication systems using the Shamir secret sharing scheme and demonstrated that it incurs small encoding/decoding overhead compared to the network transmission overhead in regular upload/download operations.

REFERENCES

[1] Jin Li, Xiaofeng Chen, Xinyi Huang, Shaohua Tang and Yang Xiang Senior Member,IEEE and Mohammad Mehedi Hassan Member, IEEE and Abdulhameed Alelaiwi Member, IEEE , "Secure Distributed Deduplication Systems with Improved Reliability." in IEEE Transactions on Computers Volume: PP Year: 2015.

[2] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou, "Secure deduplication with efficient and reliable convergent key management" in IEEE Transactions on Parallel and
Distributed Systems, 2014, pp. vol.25(6), pp. 16151625.

[3] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage." in Technical Report, 2013.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Serveraided encryption for deduplicated storage." in USENIX Security Symposium, 2013.

[5] Mihir Bellare, Sriram Keelveedhi and Ristenpart, "Message-locked encryption and secure deduplication." in EUROCRYPT, 2013, pp. 296312.

[6] Shamir, Adi, "How to share a secret." in Communications of the ACM 22 (11): 612613

[7] Xie Tao, Fanbao Liu, and Dengguo Feng , "Fast Collision Attack on MD5." in
2013.

[8] 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 Communica-tions Security, pages 491500. ACM, 2011.

BIOGRAPHY

Prerna Lahane student of ME Computer Engineering second year from the college TSSM's Padmabhushan Vasantdada Patil Institute of Technology, Bavdhan, Pune.

Prof. Sarika Bodake is a faculty in the Computer Engineering from the college TSSM's Padmabhushan Vasantdada Patil Institute of Technology, Bavdhan, Pune, India.