International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume2, Issue 4 (April '2016)

SURVEY ON SECURE DISTRIBUTED DEDUPLICATION SYSTEMSWITH IMPROVED RELIABILITY

Mohmmed Jabir Hussain,
B.E. Student,
Department of Information Science,
New Horizon College of Engineering,
Bangalore, Karnataka, India.

J Megha Shree,
B.E. Student,
Department of Information Science,
New Horizon College of Engineering,
Bangalore, Karnataka, India.

Ms. Swathi B,
Assistant Professor,
Department of Information Science,
New Horizon College of Engineering,
Bangalore, Karnataka, India.

Divya Jayaprada L,
B.E. Student,
Department of Information Science,
New Horizon College of Engineering,
Bangalore, Karnataka, India.

Chaitra R,
B.E. Student,
Department of Information Science,
New Horizon College of Engineering,
Bangalore, Karnataka, India.

Abstract -Data Deduplication has been used which is a technique to eliminate the duplicate copies of data. This technique has been used in cloud storage so the storage space decreases and uploads the bandwidth. If there are large number of users who own the same file, only a single copy of the file is stored in the cloud storage. This is how storage utilization is improved which but reduces reliability. Privacy of sensitive data is also at stake. To overcome the security issue, this paper tries to propose a distributed and reliable deduplication system which will have higher reliability in which the data chunks will not only be stored in a single server but will be distributed across multiple cloud servers. Instead of using convergent encryption, a deterministic secret sharing scheme in distributed storage systems is introduced which handles the security requirements of data confidentiality and also the tag consistency. Based on security analysis, our deduplication systems are secure. The incurred overhead is very less in real environments.

Keywords: Deduplication, Deterministic Secret Sharing Scheme, Encryption, Privacy, Reliability.

I. INTRODUCTION

With the unpredictable advancement of digital data, deduplication techniques are widely employed to backup data and reduce network and capacity overhead by identifying and uprooting excess among information. Instead of keeping several data copies with the similar content, deduplication evacuates repetitive information by keeping only a single physical copy and referring other redundant data to that replica. Deduplication has received much interest from both academia and industry because it can greatly progress storage usage and save storage space, especially for the applications with high deduplication ratio such as archival storage systems. A large number of deduplication systems have been proposed based on various deduplication strategies like client-side or server-side deduplications, file-level or block-level deduplications.

There are two types of deduplication in requisites of the size that is:

(i) *file-level deduplication*, which identifies redundancies between different files and eliminates these redundancies to decrease capacity demands, and
(ii) *block level deduplication*, which identifies and eliminates redundancies between data blocks. The file can. be separated into minor rigid-size or variable-size

blocks. Using rigid size blocks simplifies the calculations of block boundaries, while using variable-size blocks results in better deduplication efficiency.

II. LITERATURE SURVEY

A. *Secure Deduplication With Efficient And ReliableConvergent Key Management*

[1] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou "Secure Deduplication with Efficient and Reliable Convergent Key Management" IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 6, June 2014.

In this paper, data deduplication is a method for deleting duplicate copies of data, and has been broadly used in cloud storage to decrease storage space and upload bandwidth. Promising as it is, an arising dispute is to achieve secure deduplication in cloud storage. Even though convergent encryption has been widely adopted for protected deduplication, a significant problem of making convergent encryption realistic is to efficiently and reliably deal with a huge number of convergent keys. This paper makes the first effort to formally deal with the problem of achieving efficient and reliable key management in protected deduplication. We first establish a baseline approach in which each user holds a sovereign master key for encrypting the merged keys and outsourcing them to the cloud. However, such a baseline key management proposal generates a vast number of keys with the rising number of users and requires users to dedicatedly guard the master keys. To this end, we suggest Dekey, another development in which clients don't have to control any keys on their own but as an alternative securely issue the convergent key shares across several servers. Security analysis displays that Dekey is protected in terms of the definitions specified in the projected security model. As an evidence of concept, we execute Dekey using the Ramp secret sharing scheme and display that Dekey incurs limited overhead in pragmatic environments.

B. *Optimizing Cauchy Reed-Solomon Codes for Fault-Tolerant Network Storage Applications*

[2] James S. Plank, Lihao Xu." Optimizing Cauchy Reed-Solomon Codes for Fault Tolerant Network Storage Applications" The 5th IEEE International Symposium on Network Computing and Applications (IEEE NCA06), Cambridge, MA, July, 2006.

In this paper,all modes of storage applications, running from disk array systems to distributed and wide-area systems, have started to struggle with the reality of tolerating many simultaneous failures of storage nodes. Unlike the only failure case, which is optimally handled with RAID Level-5 parity, the many failure case is more complicated because optimal broad purpose strategies are not yet identified. Erasure Codingis the ground of research that deals with these strategies, and this field has blossomed in the current years. Regardless of this research, the decades-old Reed-Solomon erasure code remains the only space-optimal (MDS) code for everything except the smallest storage systems.

The best performing implementations of Reed-Solomon coding utilize a variant called Cauchy Reed-Solomoncoding, created in the mid 1990's [4]. In this paper, we present an enhancement to Cauchy Reed-Solomon coding that is in view of optimizing the Cauchy distribution matrix. We feature an algorithm for generating good matrices and then assess the performance of encoding using all implementations Reed-Solomon codes, plus the best MDS codes from the literature.

The development over the unique Cauchy Reed-Solomon codes are as much as 83% in realistic scenarios, and average roughly 10% over all cases that we tested.

C. *Private Data Deduplication Protocols in Cloud Storage*

[3] Wee Keong, Yonggang Wen, Huafei Zhu." Private Data Deduplication Protocols in Cloud Storage" SAC '12 Proceedings of the 27th Annual ACM Symposium on Applied Computing Pages 441-446.

In this paper, a new concept which we name private data deduplication protocol, a deduplication method for private data storage is introduced and formalized. By instinct, a private data deduplication protocol permits a client who holds a confidential data verifies to a server who holds a conceptual string of the data that he/she is the possessor of that data with- out enlightening additional information to the server. Our concept can be examined as a supplement of the state-of-the-art public data deduplication protocols of Halevietal. The security of private data deduplication protocols is characterized in the simulation-based structure in the circumstance of two-party computations. A production of private deduplication protocols based on the ordinary cryptographic assumptions is then obtained and analyzed. We demonstrate that the projected private data deduplication protocol is provably protected assuming that the essential hash function is collision-resilient, the distinct logarithm is rigid and the era- sure coding algorithm can remove up to α-fraction of the bits in the existence of malicious adversaries. To the best of our information this is

the early deduplication protocol for private data storage.

D. *Proofs of Ownership in Remote Storage Systems*

[4] Shai Halevi, Danny Harnik, Benny Pinkas, Benny Pinkas, "Proofs of Ownership in Remote Storage Systems", April 29, 2011.

In this paper,mentioned Cloud storage systems are increasingly accepted and a promising technology to keep their price down is deduplication, namely eliminating pointless copies of repeating data. Moreover, client-side deduplication attempts to recognize deduplication opportunities previously at the client and save the bandwidth in uploading one more copy of an existing file to the server. In this work we classify attacks that exploit client-side deduplication, allowing an attacker to increase access to potentially enormous files of other users based on a very small quantity of side information. For example, an attacker who knows the hash signature of a file can encourage the storage service that it owns that file, hence the server later lets the attacker download the whole to overcome such attacks, we introduce proofs-of-ownership (PoWs), where a client proves to the server that it in fact holds the data of the file and not just some little information about it. We formalize proof-of-ownership, present solutions based on Merkle trees and definite encodings, and analyze their security. We implemented one alternative of the scheme, our execution estimations demonstrate that our protocol incurs only a minute overhead (compared to naive client-side deduplication that is susceptible to the attack).

E. *Provable Data Possession at Untrusted Stores*

[5] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, "Provable Data Possession at UntrustedStores". CCS '07 Proceedings of the 14th ACM conference on Computer and communications security Pages 598-609.

In this paper,a model is established for demonstrable data possession (PDP) that allows a client that has stored data at a server that is not trusted to verify that the server possesses the original data exclusive of retrieving it. The model generates probabilistic proofs of ownership by sampling random sets of blocks from the server, which extensively reduces I/O costs. The client maintains a steady amount of metadata to verify the proof. The challenge/response protocol transmits a small, steady quantity of data, which reduces network communication. Thus, the PDP model for remote data checking supports huge data sets in widely-distributed storage systems. We depict two provably-secure PDP schemes that are more proficient than earlier solutions, even when compared with schemes that attain weaker guarantees. In particular, the slide at the server is low (or even constant), as to be in opposition to linear in the extent of the data. Experiments using our implementation verify the practicality of PDP andreveal that the presentation of PDP is bounded by disk I/O and not by cryptographic computation.

F. *Reclaiming Space from Duplicate Files in A Serverless Distributed File System*

[6] John R. Douceur, AtulAdya, William J. Bolosky, Dan Simon, Marvin Theimer, "Reclaiming Space from Duplicate Files in a Serverless Distributed File System", July 2002 Technical Report MSR-TR-2002-30.

In this paper, the Far site distributed file system provides accessibility by replicating each file onto some desktop computers. Since this replication get through significant storage space, it is significant to reclaim used space where likely. Measurement of over 500 desktop file systems explains that almost half of all devoured space is occupied by duplicate files. We present a method to retrieve space from this incidental duplication to make it accessible for controlled file replication. Our mechanism includes 1) convergent encryption, which enables duplicate files to coalesced into the space of a distinct file, even if the files are encrypted with diverse users' keys, and 2) SALAD, a Self-Arranging, Lossy, Associative Database for aggregating file content and site information in a decentralized, scalable, fault-tolerant method. Large-scale simulation experiments explain that the duplicate-file coalescing system is scalable, highly efficient, and fault-tolerant.

III. EXISTING SYSTEM

With the arrival of cloud storage, data deduplication techniques become more attractive and vital for the management of ever-increasing volumes of data in cloud storage services which incites enterprises and organizations to outsource data storage to third-party cloud providers. Although deduplication technique can be used to save the storage space for the cloud storage service providers, it reduces the system reliability. Data reliability is actually a very vital issue in a deduplication storage system because there is only a single copy for each file stored in the server that is shared by all the owners. If such a shared file/chunk was misplaced, an excessively large amount of data becomes remote 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 quantity of file data that would be misplaced in case of losing a solitary chunk, then the quantity of user data lost when a chunk in the storage system is corrupted increases with the number of the commonality of the chunk. Encryption mechanisms have regularly been utilized to protect the privacy before outsourcing data into cloud. Most business-related storage service providers are unwilling to apply encryption on the data because it makes deduplication impossible. The reason is that the traditional encryption mechanisms such as public key encryption and symmetric key encryption involve different users to encrypt their data with their own keys. To solve the problems of privacy and deduplication, the concept of convergent encryption has been suggested and widely adopted to implement data privacy while realizing deduplication. However, these systems accomplished privacy of outsourced data at the cost of reduced error resilience. Any of the servers can gain shares of the data stored at the other servers with the same small value as proof of ownership.

IV. PROBLEMS WITH EXISTING SYSTEM

(i) No assurance for high data reliability in deduplication system, is a vital problem.
(ii) Preceding deduplication systems have only been using a single-server setting.
(iii) The challenge for data privacy also increases as more and more sensitive data are being uploaded by users to cloud.
(iv) Reduced error resilience.
(v) The traditional deduplication methods cannot be directly completed and applied in distributed and multi-server systems.
(vi) Cannot oppose the collusion attack launched by multiple servers.

V. PROPOSED SYSTEM

The tag consistency, which was first formalized prior to avoid the duplicate/cipher text substitute attack, is considered in our protocol. In more details, it avoids a user from uploading a maliciously-generated cipher text such that its tag is similar with another honestly-generated cipher text. To accomplish this, a deterministic secret sharing technique has been formalized and utilized. To our knowledge, no existing work on secure deduplication can properly deal with the reliability and tag consistency problem in distributed storage systems.

Four novel secure deduplication systems are proposed to present efficient deduplication with high reliability for file-level and block-level deduplication, respectively. The secret splitting technique, as an alternative of traditional encryption methods, is utilized to protect data privacy. Specifically, data are divided into fragments by using secure secret sharing schemes and stored at various servers. Our proposed methods support both file-level and block-level deduplications. We execute our deduplication systems by means of the Ramp secret sharing scheme that enables high reliability and privacy levels.

VI. CONCLUSIONS

This paper proposed the distributed deduplication systems to advance the reliability of data while achieving the privacy of the outsourced data of the userswithout an encryption mechanism. Four methods were proposed to support file-level and fine-grained block-level data deduplication. The protection of tag consistency and integrity were accomplished. We implemented the proposed deduplication systems using the Ramp secret sharing scheme and established that it incurs small encoding/decoding overhead as compared to the network transmission overhead in usual upload/download operations.

VII. REFERENCES

[1] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou "Secure Deduplication with Efficient and Reliable Convergent Key Management" IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 6, June 2014.

[2] James S. Plank, Lihao Xu." Optimizing Cauchy Reed-Solomon Codes for Fault-Tolerant Network Storage Applications" The 5th IEEE International Symposium on Network Computing and Applications (IEEE NCA06), Cambridge, MA, July, 2006.

[3] Wee Keong,Yonggang Wen, Huafei Zhu." Private Data Deduplication Protocols in Cloud Storage" SAC '12 Proceedings of the 27th Annual ACM Symposium on Applied Computing Pages 441-446.

[4] Shai Halevi, Danny Harnik, Benny Pinkas, Benny Pinkas, "Proofs of Ownership in Remote Storage Systems", April 29, 2011.

[5] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, "Provable Data Possession at Untrusted Stores". CCS '07 Proceedings of the 14th ACM conference on Computer and communications security Pages 598-609.

[6] John R. Douceur, AtulAdya, William J. Bolosky, Dan Simon, Marvin Theimer, "Reclaiming Space from Duplicate Files in a Serverless Distributed File System", July 2002 Technical Report MSR-TR-2002-30.

