International Journal of Research

International Journal of Research

Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

Secure Distributed Deduplication Systems with Improved Reliability

[1]Mrs. M.SHRUTHI, [2] Mr. K.ASHOK KUMAR

[1] Pursuing M.Tech(CSE)from Jagruti Institute of Engineering and Technology

[2] Assistant Professor, Department of Computer Science and Engineering,

Jagruti Institute of Engineering and Technology, Telangana State, India.

Abstract: *De-duplication is one of the most recent advances in the present business sector since it has capacity to diminish costs. Information deduplication method is one of the essential information pressure systems for wiping out excess duplicates. Distributed information duplication framework is utilized as a part of distributed storage to diminish memory space and transfer transmission capacity stand out duplicate for every record put away in cloud regardless of the possibility that that document can be utilized by number of clients. Fundamental reason for this paper is to makes the primary endeavor to formalize circulated dependable deduplication framework. In this framework information pieces are appropriated over different servers. In disseminated stockpiling frameworks, rather than united encryption as utilized as a part of past deduplication frameworks security necessities of information label consistency and classification are accomplished by utilizing a deterministic mystery sharing plan. Security analysis exhibit that our plan is secure regarding the definitions indicated in the proposed security model.*

Keywords: Secret Sharing, Deduplication, Distributed Storage System, Reliability

I. INTRODUCTION

Cloud computing is model of the distribution of the information services in which the resources are the retrieved from the web through some of the interfaces and applications, instead forming direct connections to the server. The fast expansion in information sources has mandatory for the users to make use of some of the storage systems for storing their secret data. Cloud storage systems provide the management of the ever increasing quantity of data by keeping in mind factors like reduce occupation storage space and the network bandwidth. To make the scalable and consistent management of the data in the cloud computing, deduplication technique plays an important role. Data deduplication also helps to improve the results in efficiency term and searches are quicker. Data deduplication may happen as file level deduplication or as block level data deduplication. Instead of maintaining numerous duplicate copies of file or the data with alike content, deduplication senses and remove the redundant data by keeping original physical copy. Data deduplication is a technique of eliminate duplicate copies of data, and it is used in cloud storage to reduce storage space and bandwidth. An arising challenge is to perform secure deduplication in cloud storage even if convergent encryption is extensively adopted for secure deduplication; a critical issue is that making of convergent encryption practical to manage a huge number of convergent keys efficiently and reliably. As a result, deduplication system improves storage utilization while reducing reliability. The challenge of privacy for sensitive data also occurs when they are outsourced by users to cloud. Aiming to address the above security challenges, this makes the first attempt to celebrate the notion of distributed reliable deduplication system.

2. OUR CONTRIBUTION

In this paper, we show how to design secure deduplication systems with higher reliability in cloud computing. We introduce the distributed cloud storage servers into deduplication 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 deduplication, 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.

Another distinguishing feature of our proposal is that data integrity, including tag consistency, can be achieved.

The traditional deduplication methods cannot be directly extended and applied in distributed and multi-server systems. To explain further, if the same short value is stored at a different cloud storage server to support a duplicate check by using a traditional deduplication method, it cannot resist the collusion attack launched by multiple servers. In other words, any of the servers can obtain shares of the data stored at the other servers with the same short value as proof of ownership. Furthermore, the tag consistency, which was first formalized by [5] to prevent the duplicate/ciphertext replacement attack, is considered in our protocol. In more details, it prevents a user from uploading a maliciously-generated ciphertext such that its tag is the same with another honestly-generated ciphertext. To achieve this, a deter-monistic secret sharing method has been formalized and utilized. To our knowledge, no existing work on secure deduplication can properly address the reliability and tag consistency problem in distributed storage systems.

This paper makes the following contributions.

- Four new secure deduplication systems are pro-posed to provide efficient deduplication with high reliability for file-level and block-level deduplication, respectively. The secret splitting technique, in-stead 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. Our proposed constructions support both file-level and block-level deduplication.
- Security analysis demonstrates that the proposed deduplication 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 deduplication systems using the Ramp secret sharing scheme that enables high re-liability 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.

System architecture

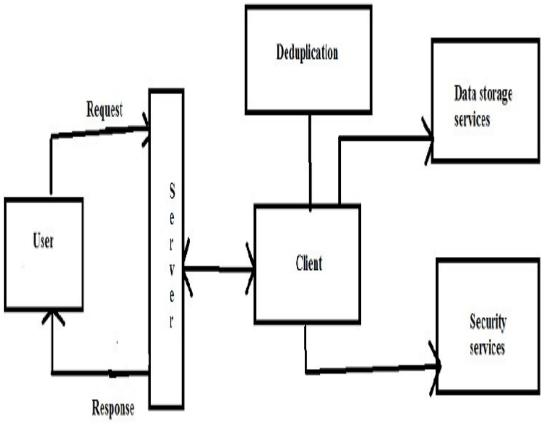

Fig 1.System Architecture

3. LITERATURE REVIEW

Literature survey is the process of presenting the summary of the journal articles, conference papers and study resources. So in this section I have studied the related topics summarized it below. In 2002 John R Douceur[1] gives mechanism to address the problems of identifying and coalescing identical files in the Farasite. Farasite-gives advantages of high availability by distributing multiple encrypted replicas of each file among a multiple desktop computers. Due to replication consumes significant storage space it is necessary to reclaim used space as possible. This paper gives the solution for control replication. John R.Douceur presents two mechanisms:

- Convergent Encryption
- SALAD (Self Arranging Lossy Associative Database)

A) Convergent Encryption:

It enables duplicate files to coalesce into space of single files, even if the files are encrypted with different user's keys. It produces identical cipher text files from identical plaintext files irrespective of encryption keys. Convergent encryption enables identical encrypted files to be recognized as identical but there remains the problem of performing this identification across large no of machines in decentralized manner. This problem solved by storing location of file & content information in distributed data structure it is nothing but SALAD.

B) SALAD-Self Arranging Lossy associative Database:

It aggregate file content and location information in decentralized, scalable, fault tolerant manner. Collectively these components called as DFC (Duplicate File Coalescing)sub system of Farasite. In 2008 Mark W.Storer[2] developed a solution that provides both data security and space efficiency in single-server storage and distributed storage systems to solve the problem such that deduplication exploits identical content, while encryption tries to make all content appear random ,the same content encrypted with two different keys results in very different cipher text. Deduplication and encryption are opposed to one another. Deduplication takes benefit of data similarity to achieve a reduction in storage space & the goal of cryptography is to make cipher text indistinguishable from theoretically random data. The goal of a secure deduplication systemis to provide data security, against both inside and outside adversaries. Storer developed two models for secure deduplicated storage authenticated and anonymous in both of these authenticated and anonymous model, an inside adversary at the chunk store would not be able to modify data without being detected. Since the chunk's name is based on the content, a user would not be able to request the modified chunk, or at the very least could tell that the chunk they have requested is different from the chunk that was returned to them.

In 2010 P.Anderson [3] presents an algorithm which takes benefits of the data which is common between users to reduce the storage requirements, and increase the speed of backups. This algorithm supports client end per-user encryption which is important for confidential personal data, also supports a unique feature that allows immediate detection of common sub trees, avoiding the necessity to query the backup system for every file. This system has shown that a community of laptop users shares a considerable amount of data in between. This gives the potential to significantly decrease backup times and storage requirements. However, they have shown that manual selection of the relevant data -eg, backing up only home directories is a poor strategy; this become fails to take backup of important files, at the same time as unnecessarily duplicating other files. This exploits a novel algorithm to reduce the number of files which have to be scanned and therefore decreases backup times.

In 2013 M.Bellare [4] Cloud storage service providers like Drop box, Mozy, and others perform deduplication to save space by only storing one copy of each file uploaded. They propose an architecture that provides secure deduplicated storage to resist brute-force attacks, and realize it in a system called DupLESS. In DupLESS, clients encrypt under message-based keys which obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an existing available service, have the service perform deduplication on their behalf, and achieves strong confidentiality guarantees.

In 2014 Jin Li and Yan Kit Li makes [5] the first attempt to address the problem of authorized data deduplication. The system present new deduplication constructions to support authorized duplicate checking. This paper shows that authorized duplicate check method incurs minimal overhead as compared to conversion encryption.

4. PROPOSED SYSTEM

To secure private information the mystery sharing method is utilized which is relating to dispersed stockpiling systems. In this paper the mystery sharing strategy is utilized for insurance of private data. In point of interest a record is separation and encode into areas by utilizing mystery sharing procedure. These areas will be conveyed over numerous independent stockpiling servers. A cryptanalysis hash estimation of the substance will likewise be figured and send to capacity server as the sign of the section put away at every server. Just the information client who first transfer the information is required to compute and circulate such mystery shares and taking after clients own same information duplicate don't have to figure and stores these shares. Recover information duplicates proprietor must get to a base number of capacity server by an approval and acquire the mystery shares to change the data. In diverse way, the approved uses will get to the mystery offers information duplicate. Another recognizable component of our proposition is that information culmination encloses label consistency, can be determined. To clarify further if the same worth is put away in different distributed storage then deduplication check by methods. It can't restrict the impact assault set up by numerous servers. As far as anyone is concerned no related work on secure deduplication can appropriately address, the unwavering quality and label consistency issue. The document level and piece level deduplication is utilized for higher reliability. The mystery part procedure is utilized for ensure information. Our proposed structure supports both customary deduplication methods. Privacy, credibility and honesty can be accomplished in our proposed system. In answer for sort of mystery agreement assaults are considered. These are the assault on the information and the assault against servers. The information is secure when the rival control predetermined number of capacity servers.

At the point when the client needs to transfer and download the record from distributed storage around then first client solicitation to the web server for transferring document. It implies just affirmed client can transfer the document to web server for that reason it utilize the evidence of possession calculation. Client to demonstrate their connection of a proprietor to the thing had of information duplicates to the capacity server. At the point when record is transferred it parts into pieces i.e as a matter of course size of square is 4KB. As per record estimate the square happens. After that deduplication recognition happens.

5. SYSTEM STUDY

5.1. THE DISTRIBUTED DEDUPLICATION SYSTEMS :

The distributed deduplication systems future aim is to reliably store data in the cloud while achieving privacy and consistency. Its main objective is to allow deduplication and distributed storage of the data diagonally multiple storage servers. As an alternative encrypting the data to keep the privacy of the data, new structures put on the top-secret intense technique to split data into shards. These shards will then be distributed transversely in multiple storage servers.

5.2 The File-level Distributed Deduplication System

To maintain efficient duplicate check, tags for each file will be calculated and are directed to S-CSPs. To avoid a conspiracy attack hurled by the S-CSPs, the tags deposited at different storage servers are computationally autonomous and different. the details of the structure as follows.

System setup. In our structure, the number of Storage servers S-CSPs is expected to be i with identities denoted by id1, id2,· · ·, idn, correspondingly. Describe the security parameter as 1 and set a secret sharing scheme SS = (Share, Recover), and a tag generation algorithm TagGen. The file storage system for the storage server is set to be #.

File Upload. To upload a file F , the user relates with S-CSPs to achieve the deduplication. More exactly, the user firstly calculates and sends the file tag ϕF = TagGen(F) to S-CSPs for the file duplicate check.When a duplicate is found, the user calculates and sends ϕF;idj=TagGen′(F, idj)to the j-th server with identity idj via the secure channel for 1 ≤j≤n. The motive for presenting an index j is to avoid the server from receiving the shares of other S-CSPs for the same file or block, which will be described in detail in the security analysis. If XF;idj equals the metadata stored with XF , the user will be provided a pointer for the shard stored at server idj . Else, if no duplicate is found, the user will continue as follows. He runs the secret sharing algorithm SS over F to get {cj} = Share(F), where cj is the j-th shard of F . He also calculates XF;idj = TagGen′(F, idj), which helps as the tag for the j-th S-CSP. As a final point, the user uploads the set of values {ϕF , cj , XF;idj } to the S-CSP with identity idj via a secure channel. The S-CSP stores these values and returns a pointer back to the user for local storage.

File Download. To download a file F , the user first downloads the secret shares {cj} of the file from k out of n storage servers. Exactly, the user sends the pointer of F to k out of n S-CSPs. After meeting enough shares, the user reconstructs file F by using the algorithm of Recover({cj}).This method provides fault tolerance and lets the user to remain available even if any limited subsets of storage servers fail.

5.3 The Block-level Distributed Deduplication System

We demonstrate how to attain the fine-grained block-level distributed deduplication. In a block-level deduplication system, the user also needs to firstly achieve the file-level deduplication before uploading his file. If no duplicate is found, the user splits this file into blocks and does block-level deduplication. The system arrangement is the same as the file-level deduplication system, excluding the block size parameter will be defined in addition. Following, the details of the algorithms of File Upload and File Download are mentioned.

File Upload. To upload a file F , the user first achieves the file-level deduplication by sending ϕF to the storage servers. If a duplicate is found, the user will achieve the file-level deduplication, Else, if no duplicate is found, the user achieves the block-level deduplication as follows.

Initially divides F into a set of fragments {Ai} (where i = 1, 2,· · ·). For each fragment Ai, the user will achieve a block-level duplicate check by computing XBi = TagGen(Ai), where the data handling and duplicate check of block-level deduplication is the same as that of file-level deduplication if the file F is substituted with block Bi.

Upon getting block tags {XBi}, the server with identity idj computes a block signal vector RBi for each i.

i) If RBi =1, the user additionally computes and sends XBi;j=TagGen′(Bi, j)to the S-CSP with identity idj. If it also equals the matching tag stored, S-CSP sends a block pointer of Bi to the user. At that time, the user keeps the block pointer of Bi and does not need to upload Bi.

ii) If RBi =0, the user runs the secret sharing al-gorithm SS over Bi and gets {cij} = Share(Bi), where cij is the j-th secret share of Bi. The user also computes XBi;j for 1 ≤j≤n and uploads the set of values {XF , XF;idj , cij , XBi;j} to the server idj through a secure channel. The S-CSP returns the consistent pointers back to the user.

File Download. To download a file F = {Ai}, the user firstdownloads the secret shares {cij} of all the blocks Ai in F from k out of n S-CSPs. Exactly, the user sends all the pointers for Ai to k out of n servers. Subsequently gathering all the shares, the user recreates all the fragments Ai using the algorithm of Recover({·}) and gets the file F ={Ai}.

5.4 Building Blocks :

Here we discuss about Secret Sharing Scheme. Let us have a look on two algorithms in a secret sharing scheme, which are Share and Recover. The secret is separated and shared by using Share. With enough shares, the secret can be pull out and improved with the algorithm of Recover. Here, the Ramp secret sharing scheme (RSSS) [7], [8] is assumed to secretly split a secret into shards. Definitely, the (i, j, p)-RSSS (where ni> j> p ≥ 0) produces n shares from a secret so that (i) the secret can be improved from any j or more shares, and (ii) No evidence about the secret can be assumed from any p or less shares. Two algorithms, Share and Recover, are defined in the (I,j,p)-RSSS.

Share splits a secret S into (j -p) pieces of equal size, generates p random pieces of the same size, and translates the j pieces using a non-systematic j of-i removal code into i shares of the same size;

Improve takes any j out of i shares as inputs and then outputs the original secret S.

We can say that when p= 0, the (i ,j , 0)-RSSS turn into the (i ,j) Rabin's Information Dispersal Algorithm (IDA) [9]. When p = j− 1, the (I, j, j− 1)-RSSS becomes the (i, j) Shamir's Secret Sharing Scheme (SSSS) [10].

Tag Generation Algorithm. In our structures below, two kinds of tag generation algorithms are defined, that is, TagGen and TagGen'. TagGen is the tag generation algorithm that records the original data copy C and outputs a tag T (C). This tag will be produced by the user and practical to achieve the duplicate check with the server. Alternative tag generation algorithm TagGen' proceeds as input a file C and an index j and outputs a tag. This tag, generated by users, is used for the proof of ownership for C .

Message authentication code. A message authentication code (MAC) is a tiny piece of data used to authenticate a message and to make available integrity and validity assurances on the message. Here the message verification code is applied to attain the reliability of the contract out stored files. It can be simply made with a keyed i.e cryptographic hash function, which takes input as a secret key and an arbitrary-length file that supplies to be authenticated, and outputs a MAC. Individual users with the same key making the MAC can confirm the exactness of the MAC value and notice whether the file has been changed or not.

Advantages of Proposed work:

- Unique feature of the proposal is that data integrity, as well as tag consistency, can be achieved.
- For our knowledge, no current work on safe deduplication can appropriately address the reliability and tag consistency problem in distributed storage systems.
- The proposed constructions maintain both file-level and block-level deduplications.
- Security analysis determines that the proposed deduplication systems are safe in terms of the definitions stated in the proposed security model. If we want to elaborate we can also say that confidentiality, reliability and integrity can be achieved in the proposed system. Two kinds of collusion attacks are measured in our solutions. These are the collusion attack on the data and the collusion attack against servers. In specific, the data remains secure even if the opponent controls a limited number of storage servers.
- The implementation of deduplication systems using the Ramp secret sharing scheme allows high reliability and confidentiality levels. The evaluation results prove that the proposed constructions are efficient and the redundancies are optimized and similar with the other storage system supporting the same level of dependability.

6. CONCLUSION:

The proposed distributed deduplication systems are to increase the consistency of data however attaining the privacy of the user's outsourced data without an encryption appliance. The security of tag consistency and integrity were attained. The implementation of deduplication systems using the Ramp secret sharing scheme here gives the demonstration that it acquires small encoding/decoding overhead compared to the network transmission overhead in regular download /upload operations.

REFERENCES

[1] J. S. Plank and L. Xu, “Optimizing Cauchy Reed-solomon Codes for fault-tolerant network storage applications,” in NCA-06: 5th IEEE International Symposium on Network Computing Applications, Cambridge, MA, July 2006

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at Untrusted stores,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS' 07. New York, NY, USA: ACM, 2007

[3] A. Juels and B. S. Kaliski, Jr., “Pors: proofs of retrievability for large files,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS '07. New York, NY, USA: ACM, 2007

[4] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, “Secure data deduplication,” in Proc. of StorageSS, 2008. Swapnali et al., International Journal of Advanced Research in Computer Science and Software Engineering 5(10), October- 2015, pp. 77-80 © 2015, IJARCSSE All Rights Reserved Page | 80

[5] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, “Proofs of ownership in remote storage

systems." in ACM Conference on Computer and Communications Security, Y. Chen, G. Danezis, and V. Shmatikov, Eds. ACM, 2011, pp. 491–500.

[6] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, "A secure cloud backup system with assured deletion and version control," in 3rd International Workshop on Security in Cloud Computing, 2011

[7] Kavitha Sree et al., International Journal of Computer Engineering In Research Trends Volume 2, Issue 12, December-2015, pp. 908-912

[8] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage." in Proceedings of the 27th Annual ACM Symposium on Applied Computing, S. Ossowski and P. Lecca, Eds. ACM, 2012, pp. 441– 446.

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

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

[11] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," in IEEE Transactions on Parallel and Distributed Systems, 2014, pp. vol. 25(6), pp. 1615– 1625.