International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 23 Issue 8 –NOVEMBER 2016.

Improve Reliability using Secure Distributed Deduplication System in Cloud

SALMA SHAIK[#1],GUNTAPALLI MINNI[*2] and SAYEED YASIN [*3]

[#] *M.Tech (CSE) Student, Nimra College of Engineering & Technology, A.P., India.*

[*] *Assistant Professor, Dept. of Computer Science & Engineering, Nimra College of Engineering & Technology, A.P., India.*

[*] *Associate professor & Head, Dept. of Computer Science & Engineering, Nimra College of Engineering & Technology, A.P., India.*

***Abstract*— Data duplication is used to improve storage utilization and also be applied to network data transfers to reduce the number of bytes that must be sent. Keeping multiple data copies with the similar content, de-duplication eliminates redundant data by keeping only one physical copy and refer other redundant data to that copy. Data de-duplication occurs file level as well as block level. The duplicate copies of identical file eliminate by file level de-duplication. For the block level duplication which eliminates duplicates blocks of data that occur in non-identical files. Although data deduplication takes a lot of benefits, security, as well as privacy concerns, arise as user's sensitive data are capable to both insider and outsider attacks. In the traditional encryption providing data confidentiality, is contradictory with data de-duplication. To maintain integrity we are providing the Third Party Auditor scheme which makes the audit of the file stored at cloud and notifies the data owner about file status stored at cloud server. This system supports security challenges such as an authorized duplicate check, integrity, data confidentiality and reliability. In this paper new distributed deduplication systems with higher reliability in which the data chunks are distributed across multiple cloud servers is being proposed.**

Index Terms*— *Deduplication, reliability, cloud servers, security requirements, data confidentiality.

I. INTRODUCTION

Main challenge face by cloud is storage service management of duplication. This duplication of data having wastage of storage space to overcome this problem deduplication technique is used, which will check duplicate copies of data; if it is found then it will eliminate these duplicate copies of data to reduce storage space and upload bandwidth. There is only one copy of data will be stored on cloud and that copy will be access by many users. Second main challenge to cloud is security data of user. Security requirement of data confidentiality and tag consistency. This can be achieved by introducing secret sharing in distributed storage system instead of convergent encryption. For authorized user to provide their ownership of data copies to storage system server we used POW that is proof of ownership. This is an interactive algorithm which is run by power and verifier. It is used in content distribution network, where an attacker does not know entire files but has accomplices who have file. Accomplices help attacker to obtain file, subject to constraint that they must sent fewer bits than initial min-entropy of file to attacker. Also for privacy and security purpose we introduced decoy technique. Decoy is the bogus information such as honeypots, honeyfiles or documents that can be generated on demand and serve as information of while detecting on unauthorized access. And also provide poison to ex-filtrated information of thief. This decoy information automatically returns by cloud and deliver in the form of normal information. But owner of file can be identifying by reading that this is bogus information. In this way true data will be remain secure. 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. We are proposed new distributed deduplication system, which has more and more reliability. In that chunks are distributed across multiple cloud servers. Deduplication technique can used for to save the memory space on the memory for the cloud storage service providers; this is reduces the reliability of the system.Security analysis indicate that our deduplication systems are secure in terms of the definitions specified in this security model. As a proof of concept, we implement the proposed systems that indicate the acquired aerial is very limited in actual environments. Deduplication process mostly improves storage utilization & it saves storage space .That's why the deduplication system is useful in industry as well as in academic.It is useful in such application which has high deduplication ratio like as archival storage system.The Most commercial storage to the No of service providers are oppose to apply encryption over the data because it is impossible to make deduplication. The reason of that system is the traditional encryption mechanism. Data reliability is actually a very critical issue in a deduplication 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 deduplication system is a critical problem.

II. PROPOSED SYSTEM

The problem is to determine how to design secure deduplication systems with higher reliability in cloud computing. Hence it is been proposed in the distributed cloud storage servers into deduplication systems to provide better fault tolerance. To protect data confidentiality, the secret sharing technique is utilized, which is also compatible with the distributed storage systems. 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. Data duplication is one of the important data compression techniques to eliminate duplicate copies of repeating data, and has been widely used in cloud storage to reduce amount of storage space and save bandwidth. Two kinds of entities will be involved in this deduplication system, including the user and the storage cloud service provider (S-CSP). Both client-side deduplication and server-side deduplication are supported in this system to save the bandwidth for data uploading and storage space for data storing.

User. The user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth.

S-CSP. The S-CSP is an entity that provides the outsourcing data storage service for the users. In the deduplication system, when users own and store the same content, the S-CSP will only store a single copy of these files and retain only unique data. •Confidentiality: Here, we allow collusion among the SCSPs. However, we require that the number of colluded SCSPs is not more than a predefined threshold. To this end, we aim to achieve data confidentiality against collusion attacks.

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.

Reliability: The security requirement of reliability in deduplication 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.

III. LITERATURE SURVEY

In 2010 p.anderson [1] 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 clientend 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.

Cloud storage systems are becoming increasingly popular. A promising technology that keeps their cost down is deduplication, which stores only a single copy of repeating data [2]. Client-side deduplication attempts to identify deduplication opportunities already at the client and save the bandwidth of uploading copies of existing files to the server. In this work we identify attacks that exploit client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on very small hash signatures of these files. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file; hence the server lets the attacker download the entire file. (In parallel to our work, a subset of these attacks was recently introduced in the wild with respect to the Dropbox file synchronization service.) To overcome such attacks, we introduce the notion of proofs-ofownership (PoWs), which lets a client efficiently prove to a server that that the client holds a file, rather than just some short information about it. We formalize the concept of proof-of-ownership, under rigorous security definitions, and rigorous efficiency requirements of Petabyte scale storage systems. We then present solutions based on Merkle trees and specific encodings, and analyse their security. We implemented one variant of the scheme. Our performance measurements indicate that the scheme incurs only a small overhead compared to naive client-side deduplication

In 2008 Mark W. Storer[3] 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 system is 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 2014 Jin Li and Yan Kit Li makes [4] 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.

IV. REALATED WORK

For this cloud security and reliability, we are implementing four algorithms-

A) SHA 256 : This algorithm is used to hash key generation and to check file deduplication. Also having fixed size hash key value. Algorithm follows steps as-

1) Derive set of round keys from ciphertext
2) Appending padding bits. Original message is padded(extended) to that it's length (in bits) is congruent to 448, modulo 512
3) Appending length 64 bits are appended to the end of padded message to indicate length of original message in bytes
4) Preparing processing function
5) Preparing processing constants
6) Initializing buffers
7) Processing message in 512 bits blocks

B) AES (Advance encryption standard): This algorithm is used for encryption and decryption purposed. AES is a symmetric block cipher, means it uses same key for both encryption and decryption. AES accept block size of 128 bits and a choice of 3 layers 128,192,256. Half of the data is used to modify other half and then halves are swapped. In this case, data block is processed in parallel during each round using substitution and permutation. This nature of AES allows for a fast software implementation of algorithm. AES is not scalable but it is faster for encryption and decryption operation. Also power consumption is low with excellent security. Simulation speed and hardware and software implementation is also faster. Algorithm follows steps as-

1) Derive the set of round keys from cipher key
2) Initialize state array with block data (plaintext)
3) Add initial round key to starting state array
4) Perform 9 rounds of state manipulation
5) Perform 10th round and final round
6) Copy final state array out as encrypted data (ciphertext)

C) Key Generation: SHA 256 algorithm is used in this module. This algorithm creates fixed size of hash key value. In some cases, there is need to appending some padding bits in original length of message or string to extend string in bytes. Also prepared processing function and constants with initializing buffers. This key value for each entry will be stored in database.

D) Encryption of File: AES is used here, it is symmetric hence it uses same key for encryption and decryption. As AES accept 128 bits block size, so that it performs 9 rounds on plaintext data at, at the time of 10th round we will get cipher text data. It means encrypted format of file. Now user will be able to upload the file on public cloud with the help of file token.

E) Token Generation: HMAC- SHA is used for token generation. This token of file is nothing but the File ID + Hash key value. At the time of uploading file, this token is generated. But this token is useful at the time of downloading of file to know who the owner of this file is, this technique is called as proof of ownership. Also for secret sharing of file, this token is needed.

F) Distribution or Splitting of File: Secret sharing schemes is used for splitting and merging. In this phase, file distributed on different server through splitting. At the time of using AES algorithm file is already divided into 3 files -. des, .enc, iv.ene. And further each file from these 3 files divided into number of server. In this way splitting of file is done. And same reverse concept is applied for merging. Due to AES complexity increases for partition of file, but this will also provide faster execution time and high security for confidential data.

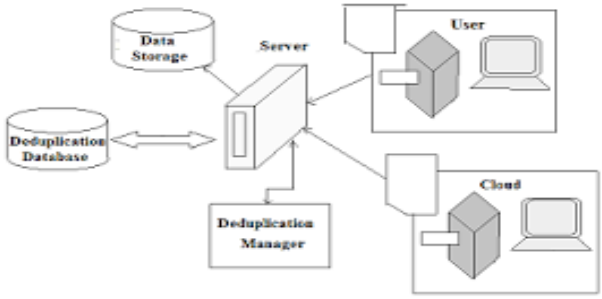

Figure 1: System Architecture

V. CONCLUSION AND FUTURE WORK

The paper, 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. We implement the secure distributed deduplication systems to improve the reliability of data while achieving the secret of the clients outsourced data. Four constructions were proposed to support file-level and fine-grained block-level data deduplication.

.

REFERENCES

[1] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer,"Reclaiming space from duplicate files in a server less distributed file system." in ICDCS, 2002, pp. 617–624.
[2] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication," in Proc. of StorageSS, 2008.
[3] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de- duplication," in Proc. of USENIX LISA, 2010
[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Server aided encryption for deduplicated storage," in USENIX Security Symposium, 2013.
[5] Science, IEEE, 1997.Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication", IEEE Transactions on Parallel and Distributed Systems, Volume: PP, Issue:99, Date of Publication :18.April.2014.

[6] "Message-locked encryption and secure deduplication," in EUROCRYPT, 2013, pp. 296–312.
[7] G. R. Blakley and C. Meadows, "Security of ramp schemes," in Advances in Cryptology: Proceedings of CRYPTO ″84, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds. Springer-Verlag Berlin/Heidelberg, 1985, vol. 196, pp. 242–268.
[8] A. D. Santis and B. Masucci, "Multiple ramp schemes," IEEE Transactions on Information Theory, vol. 45, no. 5, pp. 1720–1728, Jul. 1999.
[9] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," Journal of the ACM, vol. 36, no. 2, pp. 335– 348, Apr. 1989.
[10] A. Shamir, "How to share a secret," Commun.ACM, vol. 22, no. 11, pp. 612–613, 1979.
[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.
[12] 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.
[13] J. S. Plank, S. Simmerman, and C. D. Schuman, "Jerasure: A library in C/C++ facilitating erasure coding for storage applications - Version 1.2," University of Tennessee, Tech. Rep. CS-08-627, August 2008.
[14] [14] 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.
[15] M. Li, C. Qin, P. P. C. Lee, and J. Li, "Convergent dispersal: Toward storage-efficient security in a cloud-of-clouds," in The 6th USENIX Workshop on Hot Topics in Storage and File Systems, 2014.

SALMA SHAIK is a student of NIMRA College of Engineering and Technology, IBRAHIMPATNAM,VIJAYAWADA. She is presently pursuing her M.Tech degree from JNTU,Kakinada.

G.MINNI is presently working as Assistant professor in CSE department in Nimra college of Engineering and Technology, Jupudi, Nimra Nagar,VIJAYAWADA. She has obtained M.Tech degree from JNTU, Kakinada. She is pursuing Ph.D., in A.N.U, GUNTUR. She has published several research papers in various national and international Journals. She has more than Ten years of experience in teaching field, her area of interests are networks & Web Designing.

SAYEED YASIN received his M.TECH in Computer Science & Engg from JNTU Hyderabad. He is pursuing Ph.D., in Rayalaseema University, Kurnool. He is currently working as an Associate Professor & Head in Nimra College of Science & Technology the Department of Computers Science and Engineering & Technology, Jupudi, Ibrahimpatnam,Vijayawada-521456. He has more than Eight years of experience in teaching field. His area of interests are wireless networks & programming, & Mobile Computing.