ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Novel Mechanism for Secure Distributed Deduplication Systems with Improved Reliability

Kondapuram Anitha
M.Tech- Computer Science,
Department of CSE,
SRTIST Nalgonda, Telangana.

G.Janardhan
Assistant Professor,
SRTIST Nalgonda, Telangana.

T.Madhu
HOD,
SRTIST Nalgonda, Telangana.

ABSTRACT:

Data deduplication 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, deduplication system improves storage utilization while reducing reliability. Furthermore, the challenge of privacy for sensitive data also arises when they are outsourced by users to cloud. Aiming to address the above security challenges, this paper makes the first attempt to formalize the notion of distributed reliable deduplication system. We propose new distributed deduplication systems with higher reliability in which the data chunks are distributed across multiple cloud servers.The security requirements of data confidentiality and tag consistency are also achieved by introducing a deterministic secret sharing scheme in distributed storage systems, instead of using convergent encryption as in previous deduplication systems. Security analysis demonstrates that our deduplication systems are secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement the proposed systems and demonstrate that the incurred overhead is very limited in realistic environments.

INTRODUCTION:

Distributed computing is a field of computer science that studies distributed systems. A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages.

The components interact with each other in order to achieve a common goal. There are many alternatives for the message passing mechanism, including RPC-like connectors and message queues. Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and independent failure of components. An important goal and challenge of distributed systems is location transparency. Examples of distributed systems vary from SOA-based systems to massively multiplayer online games to peer-to-peer applications. A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs. Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.

The word distributed in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing. While there is no single definition of a distributed system, the following defining properties are commonly used:

- There are several autonomous computational entities, each of which has its own local memory.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

- The entities communicate with each other by message passing.

In this article, the computational entities are called computers or nodes.A distributed system may have a common goal, such as solving a large computational problem.[1] Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users.

Other typical properties of distributed systems include the following:

- The system has to tolerate failures in individual computers.
- The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program.
- Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input.

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

- In parallel computing, all processors may have access to a shared memory to exchange information between processors.
- In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

EXISTING SYSTEM:

- A number of deduplication systems have been proposed based on various deduplication strategies such as client-side or server-side deduplications, file-level or block-level deduplications.
- Bellare 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 deduplication.
- Li addressed the key-management issue in block-level deduplication by distributing these keys across multiple servers after encrypting the files.
- Bellare et al. showed how to protect data confidentiality by transforming the predicatable message into a unpredicatable message.

DISADVANTAGES OF EXISTING SYSTEM:

- 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.
- Most of the previous deduplication systems have only been considered in a single-server setting.
- The traditional deduplication methods cannot be directly extended and applied in distributed and multi-server systems.

PROPOSED SYSTEM:

- 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.

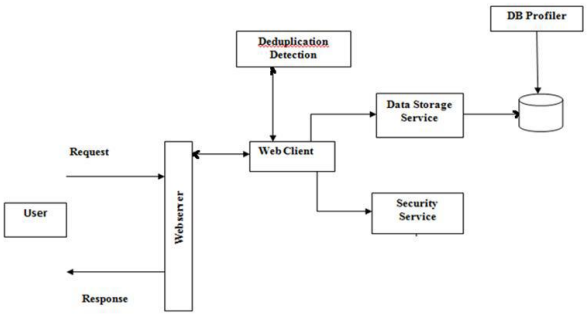
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

- 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.
- Four new secure deduplication systems are proposed to provide efficient deduplication with high reliability for file-level and block-level deduplication, 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.

ADVANTAGES OF PROPOSED SYSTEM:

- Distinguishing feature of our proposal is that data integrity, including tag consistency, can be achieved.
- To our knowledge, no existing work on secure deduplication can properly address the reliability and tag consistency problem in distributed storage systems.
- Our proposed constructions support both file-level and block-level deduplications.
- 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 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.

SYSTEM ARCHITECTURE:



IMPLEMENTATION

MODULES:

- System Model
- Data Deduplication
- File level Deduplication Systems
- Block level Deduplication systems

MODULES DESCRIPTION:

System Model

- In this first module, we develop two entities: User and Secure-Cloud Service Provide.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

- 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. Furthermore, the fault tolerance is required by users in the system to provide higher reliability.

- 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. A deduplication technique, on the other hand, can reduce the storage cost at the server side and save the upload bandwidth at the user side. For fault tolerance and confidentiality of data storage, we consider a quorum of S-CSPs, each being an independent entity. The user data is distributed across multiple S-CSPs.

Data Deduplication:

- Data Deduplication involves finding and removing of duplicate datas without considering its fidelity.
- Here the goal is to store more datas with less bandwidth.
- Files are uploaded to the CSP and only the Dataowners can view and download it.
- The Security requirements is also achieved by Secret Sharing Scheme.
- Secret Sharing Scheme uses two algorithms, share and recover.
- Datas are uploaded both file and block level and the finding duplication is also in the same process.
    - This is made possible by finding duplicate chunks and maintaining a single copy of chunks.

File Level Deduplication Systems:

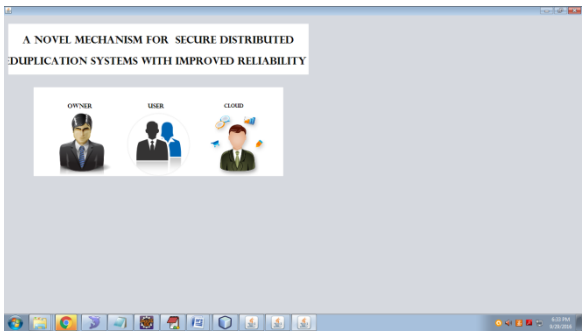
- To support efficient duplicate check, tags for each file will be computed and are sent to S-CSPs.
- To upload a file F , the user interacts with S-CSPs to perform the deduplication.
- More precisely, the user firstly computes and sends the file tag $\phi F = TagGen(F)$ to S-CSPs for the file duplicate check.
- If a duplicate is found the user computesand sendsit to a server via a secure channel.
- Otherwise if no duplicate is found the process continues,i.e secret sharing scheme runs and the user will upload a file to CSP.
- To download a file the user will use the secret shares and download it from the SCSP's .
- This approach provides fault tolerance and allows the user to remain accessible even if any limited subsets of storage servers fail.

Block Level Deduplication Systems:

- In this module we will show to achieve fine grained block-level distributeddeduplication systems.
- In a block-level deduplication system, the user also needs to firstly perform the file-level deduplication before uploading his file.
- If no duplicate is found, the user divides this file into blocks and performs block-level deduplication.
- The System setup is similar to the file level deduplication except the parameter changes.
- To download a block the user gets the secret shares and download the blocks from CSP.

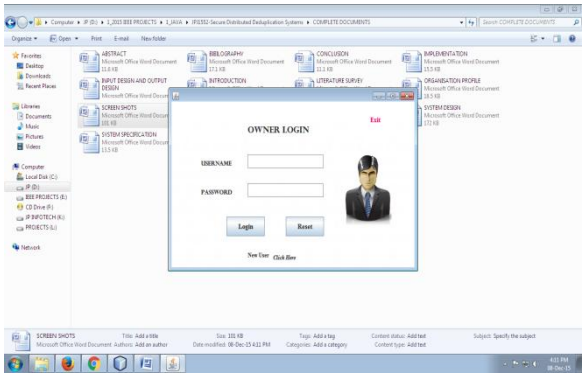
SCREEN SHOTS

Home:

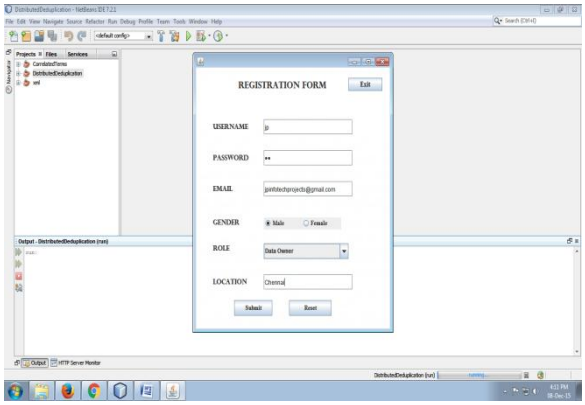


ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

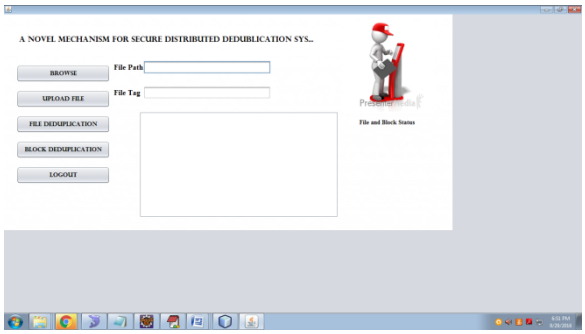
Owner Login:



Registration Form:



Owner Home:



CONCLUSION

We proposed the distributed deduplication systems to improve the reliability of data while achieving the confidentiality of the users' outsourced data without an encryption mechanism. Four constructions were proposed to support file-level and fine-grained block-level data deduplication. The security of tag consistency and integrity were achieved.We implemented our deduplication systems using the Ramp 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] Amazon, "Case Studies," https://aws.amazon.com/solutions/casestudies/#backup

[2]J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digi tal shadows, and biggest growth in the far east," http://www.emc.com/collateral/analyst-reports/idcthe-digital-universe-in-2020.pdf, Dec 2012.

[3]M. O. Rabin, "Fingerprinting by random polynomials," Center for Research in Computing Technology, Harvard University, Tech. Rep. Tech. Report TR-CSE-03-01, 1981.

[4]J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system." in ICDCS, 2002, pp. 617–624.

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

[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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[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]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]C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, "R-admad: High reliability provision for large-scale de-duplication archival storage systems," in Proceedings of the 23rd international conference on Supercomputing, pp. 370–379.

Author's Details:

Ms. K. Anitha

B.Tech college : REC Nalgonda the year 2010-2014

I hereby declare that the project work entitled "**A Novel Mechanism for Secure Distributed Deduplication Systems with Improved Reliability"** submitted to the JNTU Hyderabad, is a record of an original work done by me under the guidance of **G. JANARDHAN** , Department of Computer Science & Engineering, **SWAMI RAMANANDA THIRTHA INSTITUTE OF SCIENCE & TECHNOLOGY,** and this project work is submitted in the partial fulfillment of the Requirements for the award of the degree of Master of Technology in Computer Science& Engineering. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree.

P.Rajendra Prasad
M.Sc(IT),MCA,M.Tech
(Assistant Professor**)**

T. Madhu
(HOD) Associate professor and head of the department in CSE Swami Ramananda Tirtha Institute of Science and technology, Nalgonda,Telangana.