IMPROVEDDE DUPLICATION IN CLOUD USING HDFS STORAGE PROVIDERS

[1]T. Ragunthar, [2]S. Selvakumar
[1]AssistantProfessor,
[2]Professor
[1,2]Department of Computer Science and Engineering
[1]Sri Sai Ram Institute of Technology
[2]G.K.M College of Engineering and Technology

ABSTRACT: Cloud computing offers a new way of service provision by re-arranging various resources over the Internet. The most important and popular cloud service is data storage. In order to preserve the privacy of data holders, data are often stored in cloud in an encrypted form. However, encrypted data introduce new challenges for cloud data deduplication, which becomes crucial for big data storage and processing in cloud. Traditional deduplication schemes cannot work on encrypted data. Existing solutions of encrypted data deduplication suffer from security weakness. They cannot flexibly support data access control and revocation. Therefore, few of them can be readily deployed in practice. In this paper, we propose a scheme to deduplicate encrypted data stored in cloud based on ownership challenge and proxy re-encryption. It integrates cloud data deduplication with access control. We evaluate its performance based on extensive analysis and computer simulations. The results show the superior efficiency and effectiveness of the scheme for potential practical deployment, especially for big data deduplication in cloud storage

Keywords: Comma Separated Values, Convergent Keys, Key Management, MD5, Proof of Ownership, Triple DES.

I. INTRODUCTION

The main aim of this project is to achieve new distributed de- duplication systems with higher reliability in which the data chunks are distributed across HDFS and reliable key management in secure de-duplication using slave nodes.

Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a Parallel and distributed computing environment. It makes Use of the commodity hardware Hadoop is Highly Scalable and Fault Tolerant. Hadoop runs in cluster and eliminates the use of a Super computer. Hadoop is the widely used big data processing engine with a simple master slave setup.

Big Data in most companies are processed by Hadoop by submitting the jobs to Master. The Master distributes the job to its cluster and process map and reduce tasks sequentially. But nowadays the growing data need and the competition between Service Providers leads to the increased submission of jobs to the Master. This Concurrent job submission on Hadoop forces us to do Scheduling on Hadoop Cluster so that the response time will be acceptable for each job.

II. SYSTEM ANALYSIS

EXISTING SYSTEM

Most of the previous deduplication systems have only been considered in a single-server setting. However, as lots of deduplication systems and Storage systems are intended by users and applications for higher reliability, especially in archival storage systems where data are critical and should be preserved over long time periods. This requires that the deduplication storage systems provide reliability comparable to other high-available systems. Specifically, each user must associate an encrypted convergent key with each block of its outsourced encrypted data copies, so as to later restore the data copies. Although different users may share the same data copies, they must have their own set of convergent keys so that no other users can access their files. As a result, the number of convergent keys being introduced linearly scales with the number of blocks being stored and the number of users. Second, the baseline approach is unreliable, as it requires each user dedicatedly protect their own master key. If the master key is accidentally lost, then the user data cannot be recovered; if it is compromised by attackers, then the user data will be leaked. Cost increases to the storage of content as well as for the keys storage. Increase bandwidth with upload time.

Problem Definition:

- Single server system.
- De-duplication is not scalable.
- An Enormous number of keys with the increasing number of users.
- Cost increases to the storage of content as well as for the keys storage.
- Security lacks.

PROPOSED SYSTEM

To enable the deduplication and distributed storage of the data across HDFS. We have shown the concept of deduplication effectively and security is achieved by means of Proof of Ownership of the file. We outsource the convergent keys to slave machines securely. Dekey supports both file-level and block level deduplications. Key Management overhead is avoided and provides fault tolerance guarantees for key management, while preserving the required security properties of secure deduplication. Instead of using normal encryption and decryption we use Triple DES Technique as the plain text is encrypted triple times with the convergent key so that our data will be secured. Scalability increases as dekey achieved efficiently. Cost efficiency is achieved as multiple users of same data is just referred and not newly added. Deleting content of shared file of different user will allow deleting only convergent keys references not content stored in HDFS file storage.

Abbreviations and Acronyms

JDK Java Development Toolkit.

JMF Java Media Framework.

TCP Transmission Control Protocol.

IP Internet Protocol.

HTTP Hyper Text Transfer Protocol

III. FRAMEWORK

1. Mastering File to Cloud Service Provider:

A user is an entity who wants to outsource data storage to the storage cloud service provider (S-CSP) and access the data later. User registers to the HDFS server with necessary information and login cloud page for uploading the file. User chooses the file and uploads to server where the server store the file in rapid storage system and file level de-duplication is checked. We tag the file by using MD5 message-digest algorithm is cryptographic hash function producing a 128-bit hash value typically expressed in text format as 32 digit hex value so that files of same are de-duplicated.

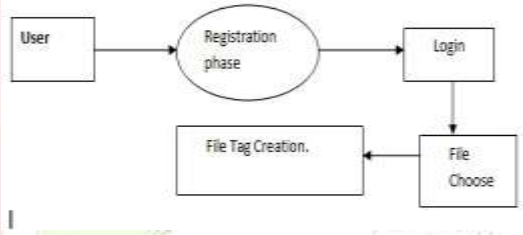


Fig 1Mastering File to Cloud Service Provider

2. Chunking the file chosen:

Chunking the file chosen of fixed size and generating tags for each blocks chunked. After that generate convergent keys for each blocks split to verify block level deduplication. Here we provide filename and password for file authorization in future. Encrypt the blocks by Triple Data Encryption Standard (3DES) algorithm. Here the plain text is encoded triple times with convergent key and so the while decoding the original content it also need the same key to decode again by triple times. Finally the original content is encrypted as cipher text and stored in Storage Cloud Service Provider (S-CSP) file storage system. Blocks are stored in Distributed Cloud Storage Providers.

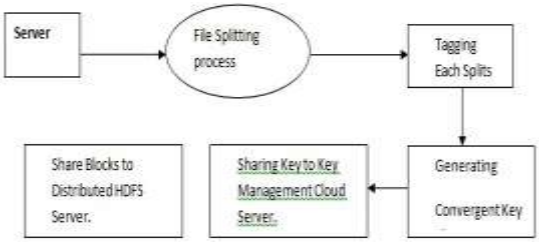


Fig.2 Chunking the file chosen

3 .De-key Based Encryption:

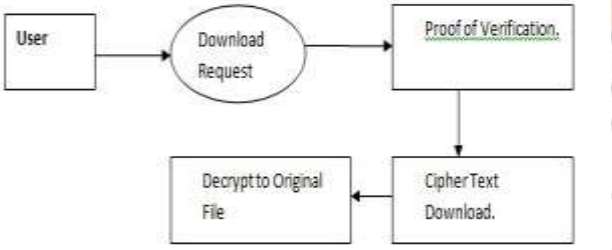
After encryption the convergent keys are securely shared with cloud service provider to Key Management Cloud Service Provider (KMCSP). Key management server checks duplicate copies of convergent keys in KMCSP. Key Management Server maintains Comma Separated Values (CSV) file to check proof of verification and store keys secure. The different users who share the common keys are referred by their own ownership. User request for deletion definitely need to prove proof of ownership to delete own contents.

Fig.3.De-key Based Encryption

4 .Hash value Based Decryption:

The final model where the user request for the downloading their own document which they have been upload and stored in HDFS server. This download request needs proper ownership verification of the document here we create the ownership by unique tag generated by MD5 algorithm and verifies existing tag of user. After verification the original content is decrypted by requesting the Distributed HDFS servers where cloud server request key management server for keys to decrypt and finally the original content is received by the user. The delete request will delete only the reference of the content shared by common users and not the whole content

Fig .4.Hash value Based Decryption



IV. ARCHITECTURE

System Features

The Enhancement of the project is that, to enable duplication in convergent keys and distribute the converged keys across multiple Key Management Cloud Service Provider (KMCSP).and chunks in the various cloud storage providers and downloaded securely by receiving keys from Key Management cloud Management Server and Chunks from the Distributed Server with improved reliability.

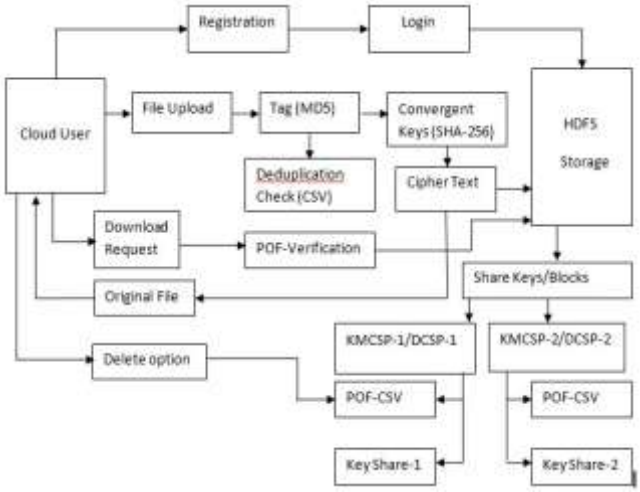


fig.5.Architecture

V. CONCLUSION

In this reserach, the new distributed de-duplication systems with file-level and fine-grained block-level data deduplication, higher reliability in which the data chunks are distributed across HDFS storage, reliable key management in secure de-duplication and the security of tag consistency and integrity were achieved.

REFERENCES

[1] Y.-M. Huo, H.-Y.Wang, L.-A.Hu, andH.-G. Yang, "A cloud storage architecture model for data-intensive applications," in ProcInt.ConfComput.Manage., May 2011, pp. 1–4.

[2] L. B. Costa and M. Ripeanu, "Towards automating the configuration of a distributed storage system," in Proc. 11th IEEE/ACM Int. Conf. Grid Comput., Oct. 2010, pp. 201–208.

[3]C.-Y. Chen, K.-D.Chang, and H.-C. Chao, "Transaction pattern based anomaly detection algorithm for IP multimedia subsystem, IEEE Trans Inform. Forensics Security, vol. 6, no.1, pp. 152–161, Mar. 2011.

[4]G. Urdaneta, G. Pierre, and M. Van Steen,"A survey of DHT security techniques," ACM Comput. Surveys (CSUR), vol. 43, no. 2, pp. 8:1–8:49, Jan. 2011.

[5]T.-Y. Wu, W.-T. Lee, and C. F. Lin,"Cloud storage performance enhancement by real-time feedback control and de-duplication,"in Proc Wireless Telecommun. Symp., Apr.2012, pp. 1–5.

[6]H. He and L. Wang, " P&P: A combined push-pull model for resource monitoring in cloud computing environment," in Proc. IEEE 3rd Int Conf. Cloud Comput., Jul. 2010, pp.260–267

[7]R. Tong and X. Zhu,. "A load balancing strategy based on the combination of static and dynamic," in Proc. 2nd Int. Workshop Database Technol. Appl., Nov. 2010, pp. 1–4

[8]T.-Y. Wu, W.-T.Lee, Y.-S.Lin, Y.-S.Lin, H.-L.Chan, and J.-S. Huang, "Dynamic load balancing mechanism based on cloud storage," in ProcComput. Com. Appl. Conf.,Jan. 2012, pp. 102–106.

[9]Y. Zhang, C. Zhang, Y. Ji, and W. Mi, " Anovel load balancing scheme for DHT-based server farm," in Proc. 3rd IEEE Int. ConfComput. Broadband Netw. Multimedia Technol., Oct. 2010, pp. 980–984.

[10] M. Dutch and L. Freeman, "Understanding data de-duplication in convergent keys and distribute the convergednutplication ratios," http://www.snia.org/, 2009.

[11] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloudservices, the case of deduplication in cloud storage," IEEE Security and Privacy Magazine, vol. special issue of Cloud Security, pp. 40–47, 2010.

[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. Chicago, IL, USA: ACM, OCT 2011, pp. 491–500.

[13] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space," in 20th USENIX conference on Security. San Francisco, CA, USA: USENIX Association Berkeley, AUG 2011, pp. 5–5.

[14] A. Juels and J. P. B. S. Kaliski, "proofs of retrievability for large files," in ACM conference on Computer and communications security.Alexandria, VA, USA: ACM, OCT 2007, p. 584597.

[15] H. Shacham and B. Waters, "Compact proofs of retrievability," in The 14th Annual International Conference on the Theory and Application of Cryptology & Information Security.Melbourne, Australia: Springer - Verlag, DEC 2008, pp. 90–107.

[16] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in ACM conference on Computer and communications security. Alexandria, VA, USA: ACM, OCT2007, p. 598609.