International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 02 | Feb 2020 www.irjet.net p-ISSN: 2395-0072

A SECURE ACCESS POLICIES BASED ON DATA DEDUPLICATION SYSTEM

Gayathri.S[1], Ragavi.P[2], Srilekha.R[3]

[1]*Asst Professor, Dept. of Computer Science and Engineering, Jeppiaar SRR Engineering college, Padur.*
[2,3]*Final year student, Dept. of Computer Science and Engineering, Jeppiaar SRR Engineering college, Padur.*

Abstract - *Deduplication techniques are used to a back up data on disk and minimize the storage overhead by detecting and eliminating redundancy among data. It is crucial for eliminating duplicate copies of identical data to save storage space and network bandwidth. It presents an attribute-based storage system with secure data deduplication access policies in a hybrid clouds settings, using public cloud and private cloud. Private cloud is used to detect duplication and a public cloud maintains the storage. Instead of keeping data copies in multiple with the similar content, in this system removes surplus data by keeping only one physical copy and referring other redundant data to that copy. User access policies defines each of such copy, the user will upload the file with access policies. Then similar file with the separate access policies are set the particular file to replace the reference. The user's private key is associated with attribute set and a message is encrypted under an access policy over a set of attributes. The user can decrypt a cipher text with private key if the set of attributes satisfies the access policy associated with this cipher text. Our system has two advantages. The two level of checking file is file level deduplication and signature match checking. It reduced the time and cost in uploading and downloading with storage space in Hadoop system.*

***KeyWords*: Hadoop Software, Ciphertext, Deduplication, Tomcat.**

1. INTRODUCTION

Hadoop software library that permits for the distributed processing of huge data set across cluster of computers using simple programming models. It is designed to proportion from single server to the thousand of machines, each offering local computation and storage. It makes Use of the commodity hardware Hadoop is very Scalable and Fault Tolerant. This provides resource management and scheduling for user applications; and Hadoop Map Reduce, which provides the programming model used to tackle large distributed data processing mapping data and reducing it to a result. Big Data in most companies are processed by Hadoop by submitting the roles to Master. This type of usage is best-suited to highly scalable public cloud services; The Master distributes the job to its cluster and process map and reduces tasks sequentially. But nowadays the growing data and the competition between Service Providers lead to the increased submission of jobs to the Master. This Concurrent job submission on Hadoop forces us to do Schedule on Hadoop Cluster so that the response time will be acceptable for each job.

2. PROPOSED SYSTEM

In this paper, we present an attribute-based storage system, which uses the ciphertext-policy attribute-based encryption (CP-ABE) and to support secure deduplication. To enable the deduplication and distributed storage of the data across HDFS. And then using two way cloud in our storage system is built under a hybrid cloud architecture, where a private cloud manipulates the computation of the user and a public cloud manages the storage. The private cloud is given a trapdoor key related to the corresponding ciphertext, with which it can transfer the ciphertext over one access policy into ciphertexts of the same plaintext under the other access policies without being conscious of the underlying plaintext. After receiving a storage request from the user, the private cloud first checks the validity of the upload item through the attached proof. If the proof is valid, the private cloud runs a tag matching algorithm to ascertain whether an equivalent data underlying the ciphertext has been stored

3. HARDWARE AND SOFTWARE SPECIFICATION HARDWARE REQUIREMENTS:

Hard disk : 500 GB and above.

Processor : i3 and above.

Ram : 4GB and above.

SOFTWARE REQUIREMENTS:

Operating System : Windows 7 and above (64-bit).
Java Version : JDK 1.7
Web Server :Tomcat 6.20
Web Server :Tomcat 7.0.11
Storage : Hadoop 2.7

4. TECHNOLOGY USED

- JAVA
- Cloud Computing
- JAVA Platform

5. APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) could also be an internet container developed at the Apache Software Foundation. Tomcat implements the servlet and thus the JSP specify from Sun Microsystems, providing an environment for Java to run

in cooperation with an online server. It adds tools for configuration and management but also can be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it's also considered a standalone web server.

5.1 Purpose

The main aim of this project is to realize new distributed de-duplication systems we present an attribute-based storage system with secure deduplication during a hybrid cloud setting with higher reliability.

5.2 Project Scope

In this Deduplication techniques are employed to backup data and minimize network of the data and storage of an data overhead by detecting and eliminating redundancy among data from the storage area.so which is crucial for eliminating duplicate copies of identical data so as to save lots of space for storing and network bandwidth. We present an attribute-based storage system with secure access policies data deduplication in a hybrid cloud. Where a personal cloud is liable for duplicate detection and a public cloud manages the storage. Instead of keeping multiple data copy with an equivalent content, during the technique eliminate redundant data by keeping only physical copy and referring other redundant datas thereto copy.

6. Algorithms used

- MD5 algorithm.
- Base 64.
- RSA algorithm.

7. SYSTEM DESIGN

7.1 Uploading a File

In this module, cloud user first register the user details and then login the user credential details. Once user name and password is valid open the user profile screen are going to be displayed. A user is an entity who wants to outsource data storage to the HDFS Storage and access the info later. A user register to the HDFS storage with necessary information and login the page for uploading the file. User chooses the file of data and uploads to Storage in the given space where the HDFS store the file in rapid storage system and file level deduplication is checked.

7.2 Mastering File to HDFS Storage

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 in order that files of same are deduplicated. Chunking the file chosen for the fixed size given of data and generating tags for each blocks chunked. Then generate convergent keys for every blocks split to verify block level deduplication. Provide filename and password for file authorization in future. This Encrypt the blocks by Triple encoding Standard (3DES) algorithm. Here the plain text is encoded triple times with convergent key then while decoding the primary content it also needs an equivalent key to decode again by triple times. Finally the first content is encrypted as cipher text and stored in slave system. Blocks are stored in Distributed HDFS Storage Providers.

7.3 Enabling Deduplication Method

After encrypting the convergent keys are securely shared with slave machines provider to Key Management machines. Key management slave checks duplicate copies of an data to convergent keys in KMCSP. Key Management slave maintains Comma Separated Values file to see proof of verification of data and store keys secure. Various users who share the common keys are referred by their own ownership. User request for deletion definitely got to prove to proof of ownership to delete own contents.

7.4 Hash Value Based Decryption

The final model user request for downloading their document which they need to upload in HDFS storage. This download request needs proper ownership of the data 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 of the data is decrypted by requesting the Distributed HDFS storage where HDFS storage request key management slave for keys to decrypt the given data 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.

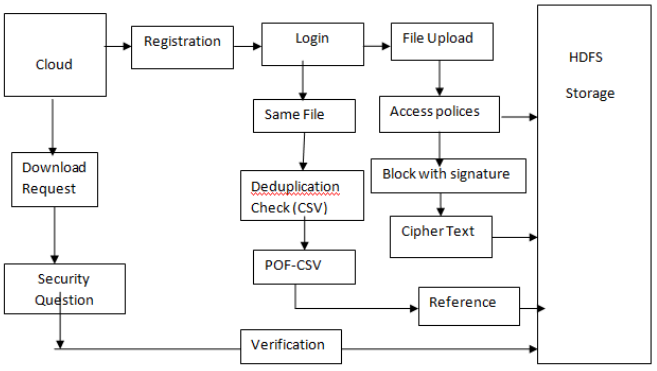

Fig -1: Block diagram

8. CONCLUSIONS

In this project, the new distributed deduplication 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 deduplication and security of tag consistency and

integrity were achieved. The purpose of software requirements specification is to provide a detailed overview of the software project and its purpose and its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements of the given data. It is defined how the client, team and audience see the project and its functionality.

9. REFERENCES

[1] D. Quick, B. Martini, and K. R. Choo, Cloud Storage Forensics. Syngress Publishing / Elsevier, 2014. [Online]. Available:http://www.elsevier.com/books/cloud-storageforensics/ quick/978-0-12-419970-5

[2] K. R. Choo, J. Domingo-Ferrer, and L. Zhang, "Cloud cryptography: Theory, practice and future research directions," Future Generation Comp. Syst., vol. 62, pp. 51–53, 2016.

[3] K. R. Choo, M. Herman, M. Iorga, and B. Martini, "Cloud forensics: State-of-the-art and future directions," Digital Investigation, vol. 18, pp. 77–78, 2016.

[4] Y. Yang, H. Zhu, H. Lu, J.Weng, Y. Zhang, and K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," Pervasive and Mobile Computing, vol. 28, pp. 122–134, 2016.

[5] D. Quick and K. R. Choo, "Google drive: Forensic analysis of data remnants," J. Network and Computer Applications, vol. 40, pp. 179– 193, 2014.

[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, ser. Lecture Notes in Computer Science, vol. 3494. Springer, 2005, pp. 457–473.