International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Deduplication in Cloud Computing for Improvising Efficiency Towards Potential Practical Usage

J.Johin[1], R.K.Rohith[2], R.S.S.Yukesh Kumar[3] ,G.Paavai Anand[4]

[1234]*Computer Science and Engineering, SRM Institute of Science and Technology, Chennai. India.*

Abstract - *Cloud computing is one of the most used technologies today. The main advantage of cloud computing is that the project developer need not buy the physical devices that will be used for the development of the project. They just need to rent the needed physical devices that are required for the project. The rented devices can easily be scaled up if the requirement of the project is increased. The major problem in cloud computing is the duplication of files that are stored in the cloud server. This will increase the memory usage of the cloud storage, In order to reduce the memory of the cloud storage we need the application that will check the duplicate files in the cloud server before uploading the files in the main memory. If the duplicate of files is available on the cloud server, the application needs to map the original files to the current files that is uploaded so that the user will not get affected by the mapping but on the backend no duplicate files will be created but the files get mapped by the application, In this way the duplication files on the cloud server will be avoided thereby reducing the storage space requirement on the cloud server.*

***Key Words*: Deduplication, Cloud, Hash Code, Security, Efficiency**

1. INTRODUCTION

Cloud storage is one of the most important services of cloud computing. Data ownership proof is an essential process of data deduplication, especially for encrypted data this is used to identify the original user of uploaded file. But this scheme does not provide flexible deduplication control across multiple Cloud Service Providers (CSPs). In this paper, we propose a multiple cloud service provider (CSPs) in which the data owner will upload the file and the hash MD5 algorithm is used to check data duplication during data storage at the cloud. CSPs. It can achieve data deduplication and access control with different security requirements. We have also proposed a scheme called Provable Ownership of the File (POF). This Provable Ownership of the file is used to identify the file uniquely. The result is security, effectiveness and efficiency towards data storage.

2. LITERATURE SURVEY

Robert H. Deng at el (2012) in their work, proposed Several schemes where attribute-based encryption (ABE) are employed for access control of outsourced data in cloud computing. Most of them suffer from inflexibility in implementing the complex access control policies on cloud storage. So to understand scalable, flexible and fine-grained access control of outsourced data in cloud computing, we propose hierarchical attribute-set-based encryption (HASBE).

Mihir Bellare et al (2013) propose an architecture that gives secure deduplicated storage that will resist brute-force attacks that are called DupLESS. In DupLESS, the encryption is message-based keys obtained from a key-server via PRF protocol. It enables the clients to store encrypted data with an existing service in the cloud, have the service perform deduplication on behalf of the client.

Qin Liu, Chiu C at el (2016) added a distribution layer to increase the efficiency of cloud computing, They presented a scheme, termed efficient information retrieval for the ranked query, to further reduce querying costs of the cloud service. Queries are classified into multiple ranks, where a better-ranked query can retrieve a better percentage of matched files.

Information Righteousness and capacity effectiveness are two necessities for distributed storage according to Shubham at el (2017). Verification of Retrievability and Confirmation of data Ownership (PDP) strategies guarantee information respectability for distributed storage. Evidence of Proprietorship (POR) that enhances the secure ownership of the file. We proposed a system that beats the existing POR and PDP plans while giving the additional usefulness of deduplication of files.

3. EXISTING SYSTEM

In existing system the heterogeneous data storage management system offers both deduplication management and access control at the same time across multiple Cloud Service Providers (CSPs). This system evaluates the performance and its security, comparison and implementation. This system uses Attribute Based Encryption(ABE) to realize deduplication data access control managed by data owner. This scheme was to solve the problem of access control. The disadvantages of existing system are security analysis and performance. They cannot overcome the issue of duplicate data storage in cloud computing. They cannot solve the problem of access control.

4. PROPOSED SYSTEM

In this work, we propose storage across multiple CSP's and preserve data security by managing deduplication. We also introduced a scheme called Provable Ownership of the File(POF).It enhances user privacy and improves the performance of practical deployment. The random hash code challenge is applied to verify data ownership, which can guarantee that the data holder really has the original data rather than its hash code. The advantages of the proposed system are it provides security, effectiveness and efficiency towards potential usage. The proposed system saves the cloud storage across multiple CSP's and preserves data security in an encrypted form. The proposed system specifies a set of attributes to identify users and encrypts based on it.

5. SYSTEM ARCHITECTURE

5.1 ARCHITECTURE DIAGRAM

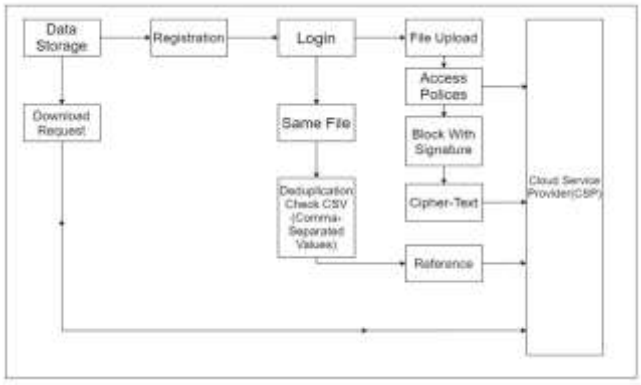

Fig 1: Architecture diagram

As shown in figure 1 the initial registration process needs to be done by the user. After registration the user can login to the system, they have the option to upload the file in the cloud server. The system will separate the file into blocks and generate the cipher-text and save the file and the respective cipher-text in the cloud service provider. When other users login and upload the same file, the system uses the cipher-text to find if the same file has been uploaded in the server and the new file is not uploaded in the cloud service provider. Instead it will refer to the original file. In the user dashboard, the user has the option to download the file that is uploaded by them. The download request will be sent to the cloud service provider to initiate the process.

5.2 USE CASE DIAGRAM

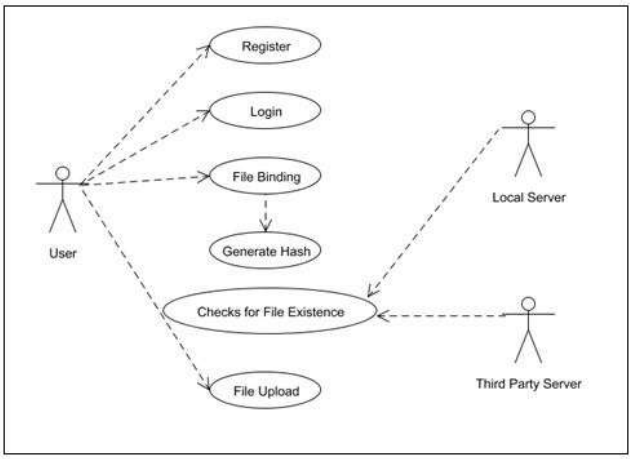

Fig 2: UML use case diagram

As shown in figure 2. The user has to initiate the registration process. After successfully loging in the user can upload the file. The initial process for uploading the file is to bind the file. This bind is useful in case of emergency. Then the cipher-text will be generated. This cipher-text is used to check the existance of the file both in the local server and global server. Every company has a local server for that particular branch and global server for all branches across the nation. If the same file is not uploaded in the cloud server then the new user can upload the file in the cloud server.

5.3 DATA FLOW DIAGRAM LEVEL 0

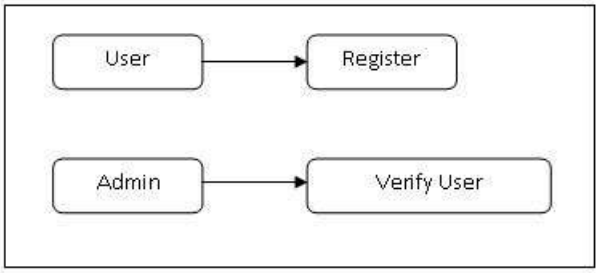

Fig 3: Data flow diagram level 0

As shown in figure 3, at level 0, the new user registers the information in the system and the admin verifies the user and gives access to the user.

5.4 DATA FLOW DIAGRAM LEVEL 1

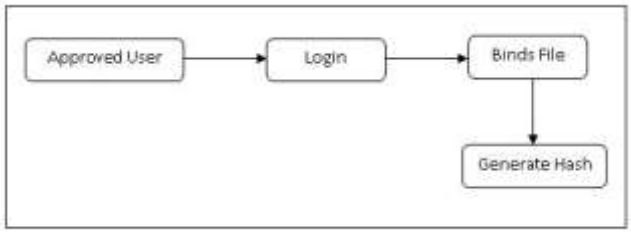

Fig 4: Data flow diagram level 1

As shown in figure 4, at level 1, the approved user can login to the user dashboard. In the user dashboard, there is an option to upload the files. In the process to upload the files to the cloud, it will bind the file and then generate the cipher-text for the content of the file.

5.5 DATA FLOW DIAGRAM LEVEL 2

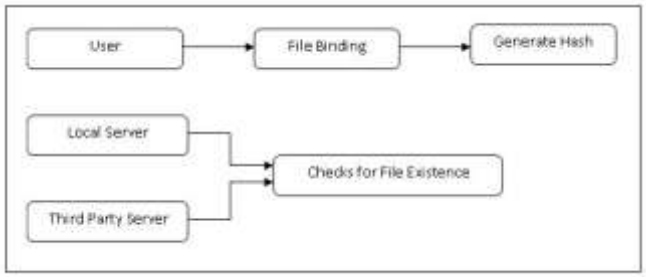

Fig 5: Data flow diagram level 2

As shown in figure 5, at level2, after generating hash code both the local server and global server check for the existence of the file on the cloud.

5.6 DATA FLOW DIAGRAM LEVEL 3

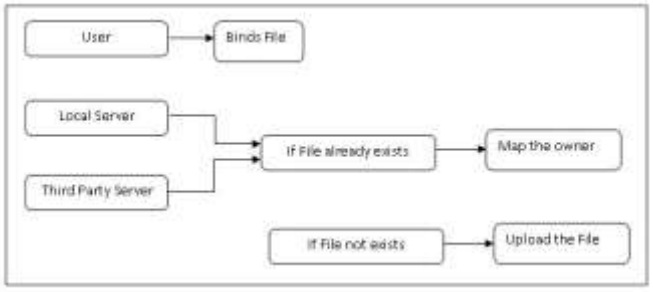

Fig 6: Data flow diagram level 3

As shown in figure 6, at level 3, if the file exists in the cloud, then the mapping of the file takes place. The system will map the current file to the original file. If the file does not exist in the cloud then the file will be uploaded to the cloud.

5.7 CLASS DIAGRAM

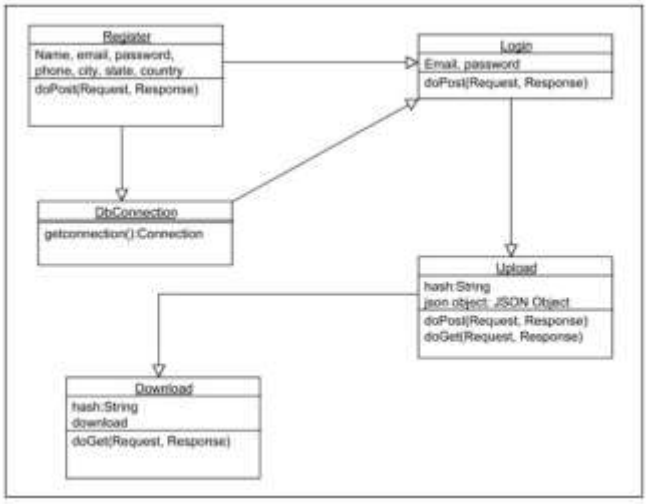

Fig 7: Class diagram

As shown in figure 7, the registration process takes place with the user's basic details. After the registration process, the request will be sent to the database where database admin needs to approve the new user. After the approval from the admin, the new user can login to the application and upload the file to the cloud. And if needed the file can be download by the user.

5.8 SEQUENCE DIAGRAM

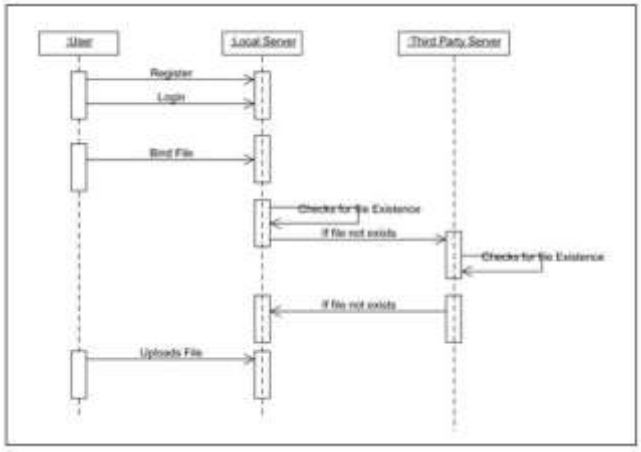

Fig 8: Sequence diagram

As shown in figure 8, after the registration process, the user can use the login id and password to sign in to the application. The user chooses the file that needs to be uploaded to the cloud. The application will bind the file for an emergency. Next, the application will check the availability of the file in the local server. If the file does not exist in the local server then the application checks the availability of the file in the global server. If the file does not

exist in both the servers then the application allows the user to upload the file to the cloud.

5.9 ACTIVITY DIAGRAM

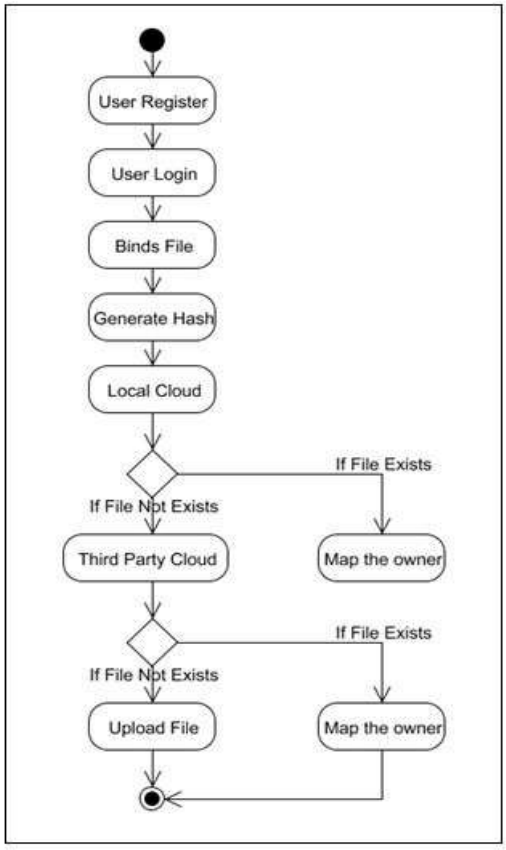

Fig 9: Activity diagram

As shown in figure 9, at first, the new user needs to register their identity on the database then the user can login to the cloud using the login id and password. While uploading the file to the cloud the application will bind the file for emergency. Then the hash code will be generated using the MD5 algorithm. The hash code is used to check the existence of the file in the cloud. If the file exists in the local server then the mapping of file takes place. The mapping of the file can be done by POF (Provable ownership of the file). If the file does not exist on the local server then the application will check the existence of the file on the global server. If the file exists in the global server the mapping of file takes place. If the file does not exist on both the servers then the file will be uploaded to the cloud.

6. MD5 ALGORITHM

The algorithm takes the input message and produces as output a 128-bit 'message digest' of the input. It is believed that it is computationally infeasible to produce two messages that having the same message digest for processing, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is intended to be used for digital signature applications, where a large file must be 'compressed' in a secure manner that is the large file is separated in multiple blocks and again the block of data is divided into multiple chunks of data before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA(Rivest–Shamir–Adleman).

The MD5 hashing algorithm is a one-way cryptographic algorithm that accepts the message of any length as input and returns as output a fixed-length digest value (cypher text) of mostly 16 digits. That output is used for authenticating the original message in the application.

7. RESULTS

Fig 10: MYSQL database

As shown in figure 10, if the same file is to be uploaded in the cloud, the application will detect the duplicate file and map it to the original file. Thus the cloud storage requirement can be reduced and performance is improved.

Fig 11: Hash generation for binded file

8. CONCLUSION

Thus we achieve data de-duplication and access control with different security requirements. Security and efficient of cloud storage are improved.

9. FUTURE ENHANCEMENT

Also, we will conduct a game-theoretical analysis to further prove the rationality and security of the proposed approach. We will improve the system to detect partial duplication of the text file by modifying the proposed scheme.

REFERENCE

[1] G. J. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services" in Proc.17thACM Computer.Communication.Security.,2010,pp.735–737.

[2] Z. Yan, Trust Management in Mobile Environments – Usable and Autonomic Models. Hershey, PA, USA: IGI Global, 2013.

[3] Y. Tang, P. P. Lee, J. C. Lui, and R. Perlman, "Secure overlay cloud storage with access control and assured deletion," IEEE Trans. DependableSecureComput.,vol.9,no.6,pp.903–916,Nov.-Dec.2012.

[4] D. Quick, and K-K R. Choo, "Digital forensic intelligence: Data subsets and open source intelligence (DFINTþOSINT): A timely and cohesive mix," Future Generation Computer. System., 2017 [Inpress]. [Online].Available:http://dx.doi.org/10.1016/j.future.2016.12.032.

[5] D. Quick, and K.-K. R. Choo, "Big forensic data management in heterogeneous distributed systems: quick analysis of multimedia forensic data," Softw.: Practice Experience, 2017 [In press]. [Online]. Available: http://dx.doi.org/10.1002/spe.2429

[6] Q. Duan, "Cloud service performance evaluation: Status, challenges, and opportunities – A survey from the system modelling perspective," Dig. Commun. Netw., Available online 23 December 2016, ISSN 2352-8648. [Online]. Available: http://dx.doi.org/ 10.1016/j.dcan.2016.12.002.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Comput. Commun. Secur., 2006, pp. 89–98.