ROBUST AND DEDICATED HYBRID CLOUD APPROACH FOR SECURE AUTHORIZED DEDUPLICATION

Aishwarya Shekhar

Assistant Programmer, Velocis Systems Pvt Ltd, Noida

***Abstract:** Data deduplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. In this process, duplicate data is expunged, leaving only one copy means single instance of the data to be accumulated. Though, indexing of each and every data is still maintained. Data deduplication is an approach for minimizing the part of storage space an organization required to retain its data. In most of the company, the storage systems carry identical copies of numerous pieces of data. Deduplication terminates these additional copies by saving just one copy of the data and exchanging the other copies with pointers that assist back to the primary copy. To ignore this duplication of the data and to preserve the confidentiality in the cloud here we are applying the concept of hybrid nature of cloud. A hybrid cloud is a fusion of minimally one public and private cloud. As a proof of concept, we implement a java code which provides security as well as removes all types of duplicated data from the cloud.*

Keywords: Confidentiality, Deduplication, Data Compression, Hybridity of cloud.

I. INTRODUCTION

Cloud computing provides seemingly unlimited "virtualized" resources to users as services across the whole Internet, while hiding platform and implementation details [1]. Today's cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified privileges, which define the access rights of the stored data [2]. One critical challenge of cloud storage services is the management of the ever-increasing volume of data. To make data management scalable in cloud computing, deduplication has been a well-known technique and has attracted more and more attention recently [3]. Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage [4]. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can take place at either the file level or the block level [5]. For file level deduplication, it eliminates duplicate copies of the same file. Deduplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Although data

deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different cipher texts, making deduplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making deduplication feasible. It encrypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same ciphertext [6]. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file [7]. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform deduplication on the cipher texts and the proof of ownership prevents the unauthorized user to the file [8]. However, previous deduplication systems cannot support differential authorization duplicate check, which is important in many applications [9]. In such an authorized deduplication system, each user is issued a set of privileges during system initialization. Each file uploaded to the cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting his duplicate check request for some file, the user needs to take this file and his own privileges as inputs[10]. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege stored in cloud. In order to save cost and efficiently management, the data will be moved to the storage server provider (SCSP) in the public cloud with specified privileges and the deduplication technique will be applied to store only one copy of the same file[11]. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with specified privileges to realize the access control [12]. Traditional deduplication systems based on convergent encryption, although providing confidentiality to some extent; do not support the duplicate check with differential privileges [13].

In other words, no differential privileges have been considered in the deduplication based on convergent encryption technique. It seems to be contradicted if we want to realize both deduplication and differential authorization duplicate check at the same time [14].

II. A SECURE DATA DEDUPLICATION SCHEME FOR CLOUD STORAGE

As more corporate and private users outsource their data to cloud storage providers, recent data breach incidents make end-to end encryption an increasingly prominent requirement [15]. Unfortunately, semantically secure encryption schemes render various cost-effective storage optimization techniques, such as data deduplication, ineffective[16]. We present a novel idea that differentiates data according to their popularity. Based on this idea, we design an encryption scheme that guarantees semantic security for unpopular data and provides weaker security and better storage and bandwidth benefits for popular data [17]. This way, data deduplication can be effective for popular data, whilst semantically secure encryption protects unpopular content. We show that our scheme is secure under the Symmetric External Decisional Differ-Hellman Assumption in the random oracle model [18].

III. PSEUDO CODE IN JAVA FOR DEDUPLICATION

Runs if Source File Doesn't Exist Which We Want to Copy in Cloud.

```
if(!sf.exists()){
System.out.println("Given file or directory does not exist!!");}
else{
File df=new File("C://Users/Aishwarya Shekhar/Downloads/"+fn);
if(df.exists()){
check(sf, df);        }
else{
System.out.println("Copying the file/folder '"+sf.getName()+"'...");
newCopy(sf, df);
System.out.println("Copying done!");
```

Runs if the File We Want to Copy in the Cloud is a Directory.

```
if(fs.isDirectory()){
arg=c.readLine("Folder '"+ fd.getName() +"' already exists; Press 'M' to merge with old one, 'S' to save with new name or 'C' to cancel operation : ");
if(arg.equalsIgnoreCase("M"))
{
System.out.println("Merging old folder '"+ fd.getName() +"' with new one...");
reCopy(fs, fd);
System.out.println("Merging done!");}
if(arg.equalsIgnoreCase("S")){
System.out.println("Saving folder '"+ fd.getName() +"' with new name...");
System.out.println("Saving done!");}
if(arg.equalsIgnoreCase("C")){
System.out.println("Cancelling the operation...")}}
else{
```
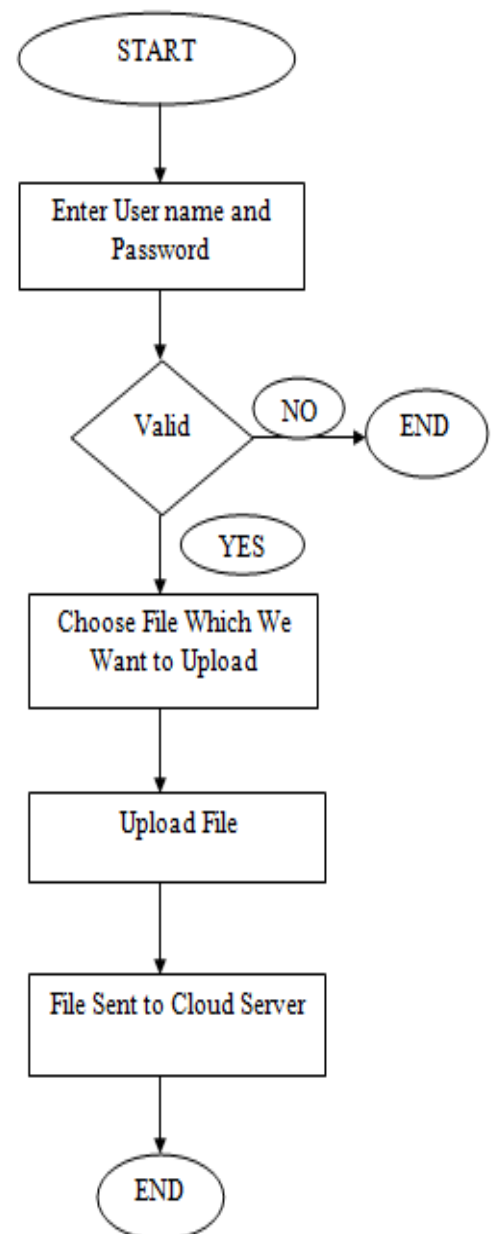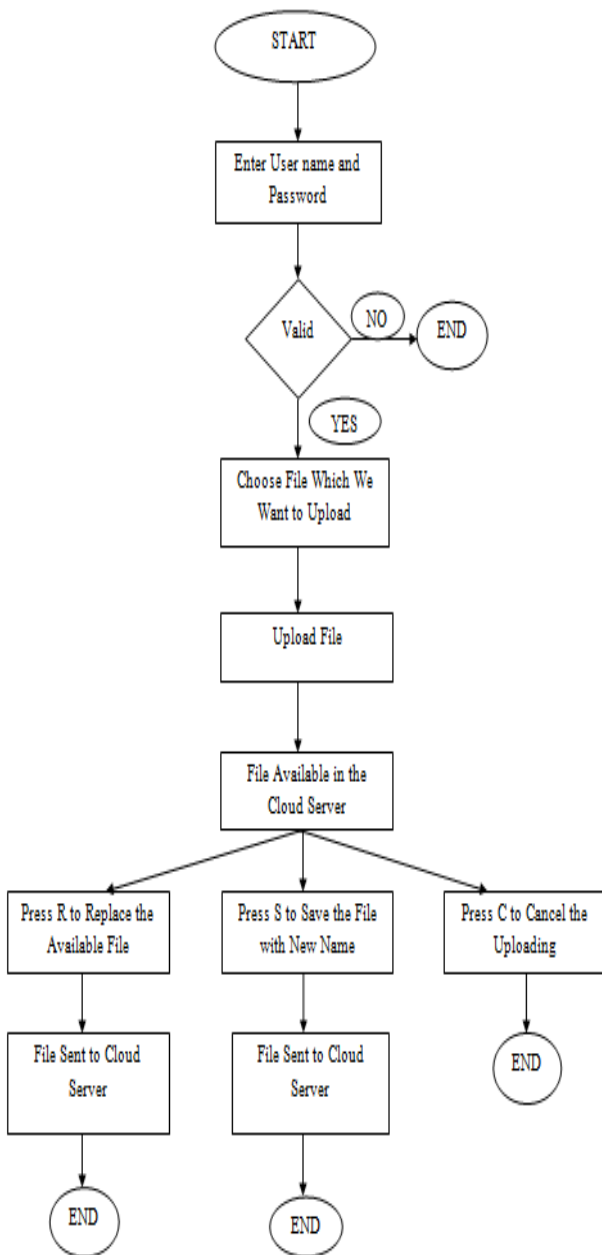
```
arg=c.readLine("File '"+ fd.getName() +"' already exists; Press 'R' to replace with new file, 'S' to save with new name or 'C' to cancel operation : ");
if(arg.equalsIgnoreCase("R")){
System.out.println("Replacing old file '"+ fd.getName() +"' with new one...");
reCopy(fs, fd);
System.out.println("Replacing done!");}
if(arg.equalsIgnoreCase("S")){
System.out.println("Saving file '"+ fd.getName() +"' with new name...");
newCopy(fs, nameChange(fd));
System.out.println("Saving done!");}
if(arg.equalsIgnoreCase("C")){
System.out.println("Cancelling the operation...");
```

IV. FLOW CHART

source-stream -> destination-stream //copying file; source to destination
return message copying done
end function
function recopy(file-name, path)
arg <- M/R, S, C //waiting for input, M/R- merge/replace, S-New name, C-cancel
If arg = M then
copy(file-name, path)
else
If arg = S then
copy(file-name, rename(path)) //calling function to rename the destination file-name
else
exit(1)
end
end
end function
function rename(path)
path = path + some-value //gradually incrementing value appended to file-name
return path
end function

VI. WORKING OF THE CODE

This is a code developed in java so as to remove the duplicated files, folders, text images, videos and everything which can be uploaded or saved in a hybrid cloud with the same name and extension. When we implement the program of deduplication to copy the data it will ask for entering the file names with extension means we have to write the file name with its extension which we want to copy in the cloud. Then after execution it will be copied in the cloud. In the cloud the name of the file has been saved with its extension. Again next time if we want to copy the same data in the cloud the starting procedure will be same but after coming to the procedure that is enter the file name with extension and after its execution. A message will come on the screen that file already exists. Press 'R' to replace with new file, 'S' to save with new name or 'C' to cancel the operation. We have to decide what we want to do with the file. It is followed by three steps which is described below.
A. If we press 'R' then the first file is replaced by the present file which we want to save again.
B. If we will press 'S' the same data with new name will be saved in the cloud.
C. If we will press 'C' then the operation will be cancelled and the process is over.
This java code provides a very efficient way by the help of which duplicated data from the cloud can be removed so as we can use that space in saving other data. By this way deduplication of data happens in the cloud.ensuring that the underlying data is in the right place at the right time as well as dealing with enforcement of governance and security policies regardless of where active instances are operating.

VII. CONCLUSION & FUTURE SCOPE

In this paper, data deduplication was very much possible by using the java code. This java code efficiently removes the

V. ALGORITHM

Algorithm FileCopier
Input: String file-name (existing file/directory, being used as a source), destination path
Output: String copy result (for provided destination)
 if file not exists then
return incorrect file-name
else
if path file exists then
recopy(file-name, path)
else
copy(file-name, path)
end
end
function copy(file-name, path)

duplicated data from the cloud by checking the name of the file with extension and the size of the file. This code mainly works for jpg files, mp3 files, mp4 files, txt files etc. If a folder consists of many folders then in that case also it checks each and every folder for checking the duplicated files. This code mainly works on merging or replacing the data, saving the data with new name or cancellation of the operation. In future work we want to develop a code which checks the name with extension as well as basic contents of data in a file. On the basis of that deduplication will be possible.

REFERENCES

[1] A Very Robust Dedicated and Verified Technique by Applying the Hybridity of Cloud for Deduplication of Data, Aishwarya Shekhar, Devesh Kumar Srivastava Networking and Communication Engineering, Vol 7, No 3 (2015).

[2] New Privacy And Security Issues in a Crowd Based cloud Computing, Aishwarya Shekhar, Devesh Kumar Srivastava, ACME, Vol-3, Issue 1, January 2015.

[3] Jerry Gao, Xiaoying Bai, and Wei-Tek Tsai "Testing as a Service (TaaS) on Clouds" IEEE Seventh International Symposium on Service-Oriented System Engineering 2013.

[4] Thesis by Ivailo P. Sokolov, 2009. Cloud Computing: Overview, Concepts and Business Deployment Scenarios. Bachelor's 2009 Vienna University of Economics and Business..

[5] Pankaj Goyal, 2009. Policy-based Event-driven Services-oriented Architecture for Cloud Services Operation & Management. Prceedings of IEEE International Conference on Cloud Computing(Cloud'09),pp135-138.

[6] L.M. Riungu, O. Taipale, and K. Smolander, " Research issues for software testing the cloud" IEEE Second International Conference on Cloud Computing Technology and Science, Dec. 2010.

[7] L. Riungu-Kalliosaari, O. Taipale, K. Smolander, "Testing in the Cloud:Exploring the Practice, "Special issue on Software Engineering for Cloud Computing, IEEE Software, March/April 2012..

[8] N. Bonvin, T.G. Papaioannou, and K. Aberer, ''A Self-Organized Fault-Tolerant and Scalable Replication Scheme for Cloud Storage,'' in Proc. 1st ACM Symp. Cloud Comput., ser. SoCC'10, New York, NY, USA, 2010, pp. 205-216.

[9] Massoud Pedram, "Energy-Efficient Datacenters," IEEE Transactions on Computer –Aided Design of Integrated Circuits and Systems, Vol. 31, N0. 10, pp. 1465-1484, October 2012.

[10] Peter Mell and Timothy Grance. The NIST definition of cloud computing (draft) recommendations of the national institute of standards and technology. Nist Special Publication, 145(6):7, 2011.

[11] Marios D. Dikaiakos, George Pallis, Dimitrios Katsaros, Pankaj Mehra, and Athena Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific," 1089-7801/09 IEEE Computer Society, 2009.

[12] 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 SYSTEM VOL:PP NO:99 YEAR 2014.

[13] K C. Ng and P. Lee. Revdedup, "A reverse deduplication storage system optimized for reads to latest backups", In Proc. of APSYS, Apr 2013.

[14] W. K. Ng, Y. Wen, and H. Zhu, "Private data Deduplication protocols in cloud storage". In S. Ossowski and P. Lecca, editors, Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.

[15] Y. M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller. "Secure data Deduplication", In Proc. of StorageSS, 2008.

[16] S. Sivathanu, L. Liu, M. Yiduo, and X. Pu, ''Storage Management in Virtualized Cloud Environment,'' in Proc. 3rd IEEE Int'l Conf Cloud, 2010.

[17] M. Creeger, ''Cloud Computing: An Overview,'' ACM Queue vol. 7, no. 5, pp. 1-5, June 2009.

[18] Monitoring Aspects of Cloud over the Big Data Analytics Using the Hadoop for Managing Short Files, Prerna Kumari, Himanshu Sharma, Aishwarya Shekhar Data Mining and Knowledge Engineering, Vol 7, No 3 (2015).