RESOLVING DATA DE-DUPLICATION IN CLOUD COMPUTING

Subash Chandar(subashchandar@gmail.com)

R.Suriyakumar(suriya1620174@gmail.com)

J.wilfer raja(wilferraja@gmail.com)

E.Pavithran(monsterpavir3@gmail.com)

Department Of Computer Science And Engineering
Jeppiaar Engineering College , Chennai , India

---***---

Abstract - Cloud storage allows cloud users to overcome space constraints and extend their storage without upgrading their computers as one of the essential cloud computing services. Data is often outsourced in an encrypted form to ensure the security and privacy of cloud users. On the other hand, encrypted data could waste a lot of cloud storage space and make data sharing among approved users more difficult. Encrypted data storage and management of deduplication continue to be a problem. Traditional deduplication systems are often based on particular application situations in which data owners or cloud servers have complete control over deduplication. They are unable to meet the diverse demands of data owners based on the degree of data sensitivity. This paper suggests a multiple Cloud Service Providers (CSPs) model in which the data owner uploads the file, and the hash MD5 algorithm is used to search for data replication during cloud CSP storage. With different security criteria, it can achieve data deduplication and access control. We've also suggested a method known as Provable Ownership of the File (POF). As a consequence, data storage management is more safe, reliable, and efficient.

***Key Words*:** Cloud Storage, Cloud Service Providers (CSPs), hash MD5 algorithm, data deduplication, Provable Ownership of the File(POF).

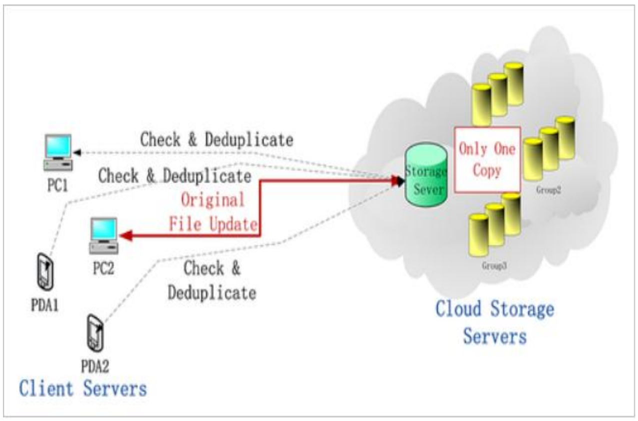
1.INTRODUCTION *(Size 11, Times New roman)*

Cloud computing provides centralized data storage as well as remote access to information services and resources. It provides a modern way of delivering Information Technology (I.T.) services by rearranging different tools and makes them available to users for their needs. Due to a range of properties such as scalability, elasticity, fault-tolerance, and pay-per-use, cloud computing has significantly enriched ubiquitous services and has become a good service network.

One of the most commonly used cloud services is data storage Cloud users have reaped significant benefits from cloud storage because they can store large amounts of data without having to upgrade their computers and access it at any time and from any place. However, Cloud Service Providers (CSPs) also have several concerns with cloud data storage.

Data duplication in the cloud with data access control is still an issue. Duplicated data may be encrypted and processed in the cloud by the same or different CSPs. From a usability standpoint, it is highly anticipated that data deduplication would work well with data access control. The same data is only saved once in the cloud to be accessed by various users depending on the data owners' or data holders' policies. Although cloud storage space is plentiful, duplicated data storage can waste many networking resources, drain a lot of electricity, boost operational costs, and complicate data management.

Current research cannot provide a generic solution that supports both deduplication and access control in a scalable and consistent manner over the cloud. To address the identified problem, we propose a holistic and heterogeneous data storage management scheme.



2.Body of Paper

2.1 Access control on encrypted data

Existing research has suggested encrypting data before submitting it to the cloud to avoid CSP data privacy violations. Only approved entities are allowed to decrypt encrypted data according to access control on encrypted data. It's better to encrypt all data once and only send out relevant keys to approved parties once. On the other hand, key management becomes difficult due to confidence relationships' changeability, which necessitates regular vital updates.

To gain access control on encrypted cloud data, Attribute-Based Encryption (ABE) was suggested. It defines a set of attributes to identify users and encrypts data using the attributes-defined access structure. As a result, encrypted data can only be decrypted by users who have the appropriate qualities to determine the access structure.

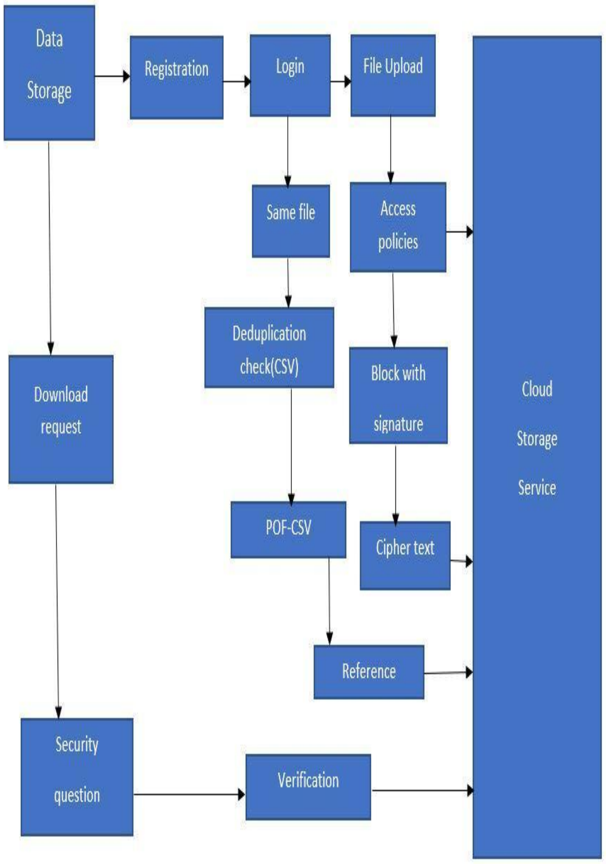
Existing System

A heterogeneous data storage management scheme is currently in use, which flexibly provides both deduplication management and access control through several Cloud Service Providers (CSPs). Security analysis, Comparison, and implementation are used to assess its performance. To achieve deduplication data access control controlled by the data owner, they use Attribute-Based Encryption (ABE). This scheme was created to resolve the issue of access control. There are some disadvantages, and some of them are the efficiency and security analysis are insecure. They cannot address the problem of duplicated data storage in cloud computing and the question of access control.

Proposed system

We propose to store data through multiple Cloud Service Providers (CSPs) and maintain data protection by handling deduplication in this proposed approach. We also add a concept called Provable Ownership of the File in this section (POF). Data deduplication includes data ownership proofing, which is particularly important for encrypted data. They improve the efficiency of realistic implementation while improving user privacy. The random hash code problem is used to check data ownership, ensuring that the data owner owns the original data rather than its hash code. This enhances stability, effectiveness, and productivity in terms of potential use. They allow cloud storage to be shared across multiple CSPs while maintaining data protection in encrypted form. They also define a set of attributes to classify users and encrypt data based on those attributes. This method addresses the issue of access control.

System design



We suggest a deduplication-based scheme for managing heterogeneous data storage. It can be applied in several situations where cloud data deduplication is done by 1) the data owner or 2) a trusted third party. 3) by the cloud's owner or a reliable third party. We use the hash code of data to search for data duplication during cloud storage. The data holder signs the hash code of the data for it to pass CSP's originality verification.

3.Working Principles:

As the deduplication in the cloud is undergone, the modules developed for the system includes

1)Cloud User Authentication 2) File upload and comparison 3) Set access policy for file

4) File Download Request and Handling

Cloud user Authentication

In Cloud Service Provider, the owner has an initial level registration process (CSP). This process requires users to have their personal information. The information is then stored in the server's database. They have to go through the login phase to gain access to the Cloud Service Provider.

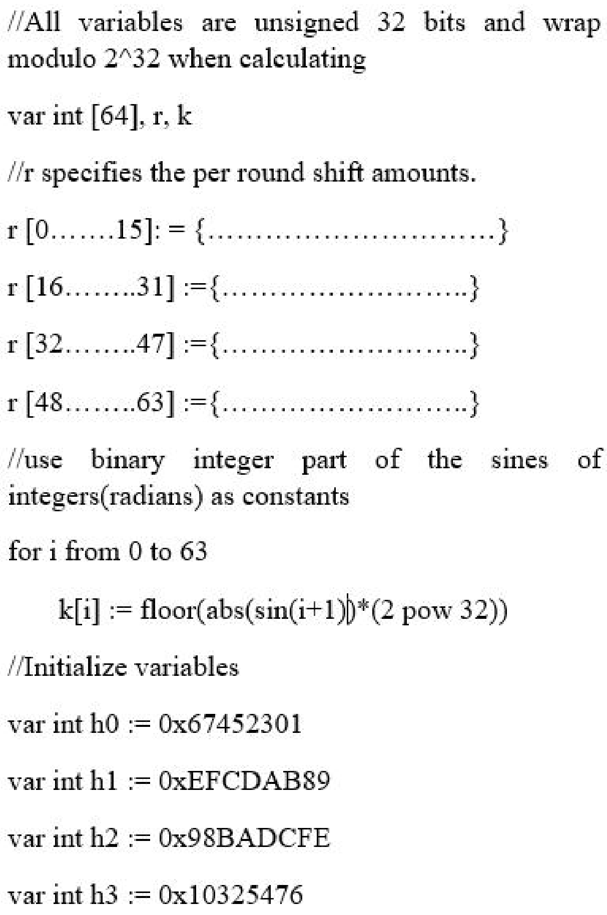
File upload and Comparison

The data owner creates an account in the public cloud and uploads it to cloud storage in this module. The scheme of Provable Ownership of the File (POF) is suggested here. The hash key is generated using the MD5 algorithm while the data owner uploads the file. For each uploaded file, the hash key is different. However, suppose the data owner uploads the same file. In that case, it will not allow the file to upload and instead substitute the reference id by index mapping, and both the data owner and the Cloud Service Provider search the file for physical presence or absence.

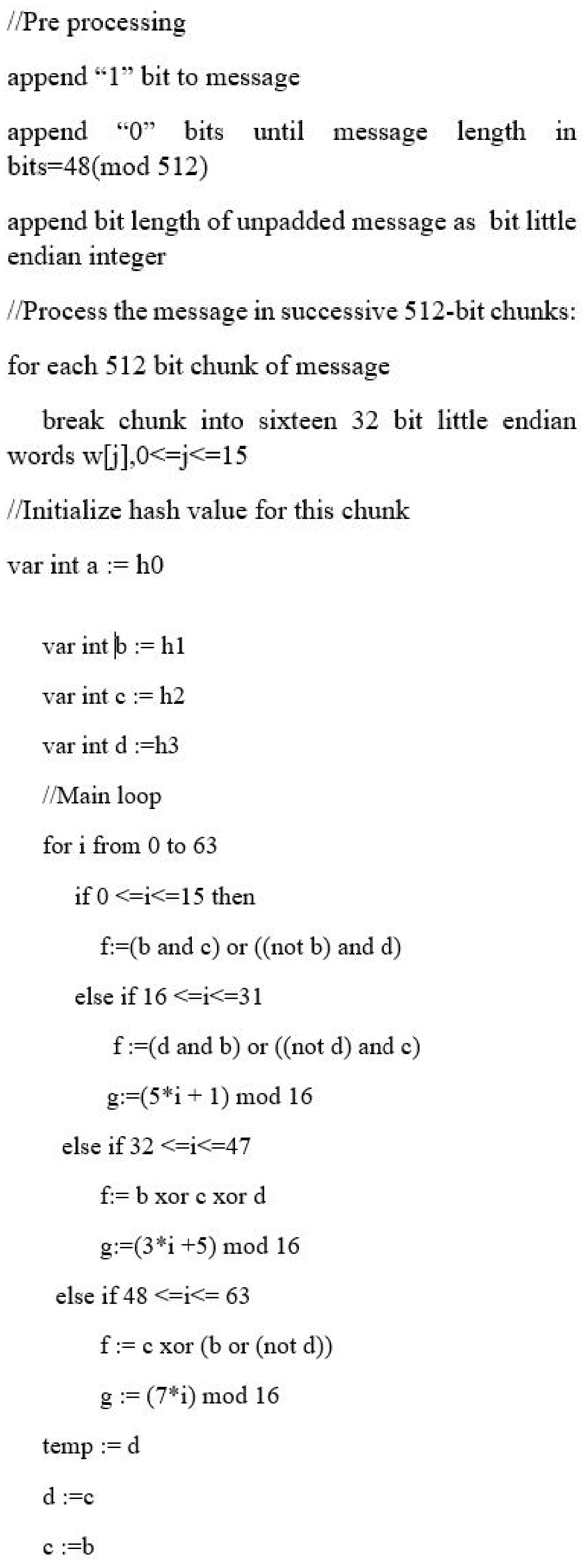
Set Access Policy for File

In this module, the user selects a file and uploads it to a storage system that uses the Hadoop Distributed File System (HDFS). The course will create a signature in a specific file, which will then be divided into several blocks. Each block will produce a keyed signature. The signature uses the MD5 message-digest algorithm, a 128-bit hash value usually represented in text format as a 32-digit hex value, allowing files to be deduplicated. Then, for each block split, create convergent keys to store CSV file information such as file name, file path, blocks, username, password, and block resolutions.

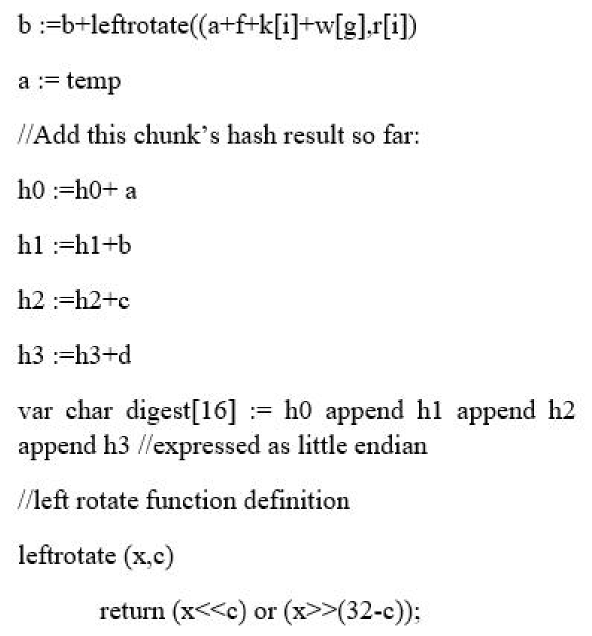
MD5 algorithm:



```
//All variables are unsigned 32 bits and wrap
modulo 2^32 when calculating

var int [64], r, k

//r specifies the per round shift amounts.

r [0…….15]: = {…………………………}

r [16……..31] :={……………………..}

r [32……..47] :={……………………..}

r [48……..63] :={……………………..}

//use binary integer part of the sines of
integers(radians) as constants

for i from 0 to 63

      k[i] := floor(abs(sin(i+1)))*(2 pow 32))

//Initialize variables

var int h0 := 0x67452301

var int h1 := 0xEFCDAB89

var int h2 := 0x98BADCFE

var int h3 := 0x10325476
```



```
//Pre processing

append "1" bit to message

append "0" bits until message length in
bits=48(mod 512)

append bit length of unpadded message as bit little
endian integer

//Process the message in successive 512-bit chunks:

for each 512 bit chunk of message

    break chunk into sixteen 32 bit little endian
words w[j],0<=j<=15

//Initialize hash value for this chunk

var int a := h0

    var int b := h1

    var int c := h2

    var int d :=h3

    //Main loop

    for i from 0 to 63

        if 0 <=i<=15 then

            f:=(b and c) or ((not b) and d)

        else if 16 <=i<=31

             f :=(d and b) or ((not d) and c)

            g:=(5*i + 1) mod 16

      else if 32 <=i<=47

            f:= b xor c xor d

            g:=(3*i +5) mod 16

     else if 48 <=i<= 63

            f := c xor (b or (not d))

            g := (7*i) mod 16

    temp := d

    d :=c

    c :=b
```



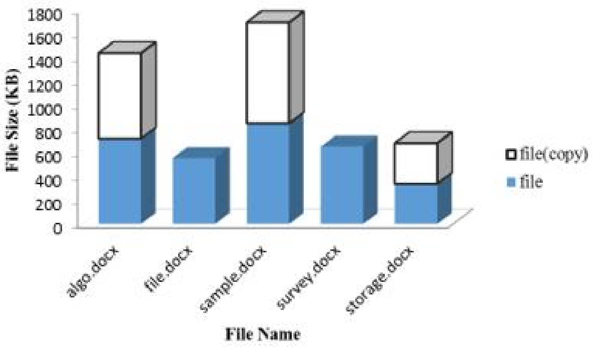
```
b :=b+leftrotate((a+f+k[i]+w[g],r[i])
a := temp
//Add this chunk's hash result so far:
h0 :=h0+ a
h1 :=h1+b
h2 :=h2+c
h3 :=h3+d
var char digest[16] := h0 append h1 append h2
append h3 //expressed as little endian
//left rotate function definition
leftrotate (x,c)
        return (x<<c) or (x>>(32-c));
```

File Download Request and Handling

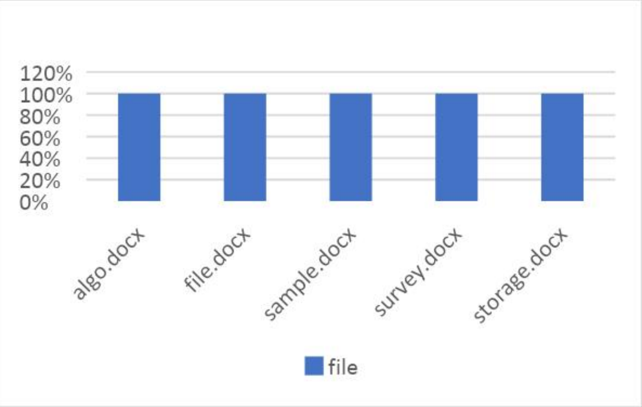
In this module, the data owner will download the file from the cloud service provider. If they do not find the file, they will request to download the file from different Cloud Service Provider and check whether the file is present or not, and then it gives the response to the data owner.

Performance analysis

It takes to upload a file to increase as the file size increases. The experimental results show that the proposed approach effectively reduces the time taken to upload the redundant file since the deduplication search is performed on the client-side. As it generates the unique hash key using MD5, the file duplication is avoided, and the bandwidth time is saved.



Performance analysis before deduplication



Performance analysis after deduplication

4.Conclusion

Existing methods for locating deduplication in cloud storage were effective in achieving data integrity or storage reliability, not both. Furthermore, the current plans do not help with cross-user deduplication testing. Client-side deduplication methods have not protected the user's privacy. The approach suggested has CSP has dedicated storage space for users, and they have been successful in making good use of it. Our system also allows many users to share a single memory space, allowing us to save more space than previous methods. It also protects the user's privacy by verifying their evidence of ownership. Using a random key generation process prevents unauthorized users from accessing other files stored in the cloud, increasing the data files' protection.

The work can be implemented in a real-time environment to improve the reliability and usability of file auditing by enabling users to authenticate their cloud-based file content without downloading the entire file. The obfuscation process, which can be used to enhance the user file's protection, is another way to provide security.

References

[1] R. Chow et al., "Controlling data in the cloud: outsourcing computation without outsourcing control," in Proc. ACM Workshop Cloud Comput. Secure., 2009, pp.85-90.

[2] 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. Secure., 2006, pp. 89-98.

[3] S. Muller, S. Katzenbeisser, and C. Eckert, "Distributed attributebased encryption," in Proc. 11th Annu. lnt. Conf. lnf. Secure. Crypto., 2008, pp. 20-36.

[4] S. C. Yu, C. Wang, K. Ren, and W. J. Lou, "Attribute-based data sharing with attribute revocation," in Proc. ACM Asia Conf. Comput. Commun. Secure., 2010, pp.261-270.

[5] Dropbox," A file storage and sharing service." [Online]. Availabl http://www.dropbox.com/,retrieved March 2017.

[6] Google Drive. [Online]. Available: http://drive.google.com, retri May 2017.

[7] Mozy, "Mozy: A file storage and sharing service." [Online]. Available: http://mozy.com/, retrieved May 2017.

[8] J.R. Douceur, A.Adya, W.J.Bolosky, P.Simon, and M.Teimer," Reclaiming space from duplicate files on a serverless distributed fil system," in a serverless distributed file system," in Proc.22nd lnt. Cc Distributed Comput. Syst., 2002, pp. 617-624.

[9] C. Yang, J. Ren, and J. F. Ma, "Provable ownership of the file i deduplication cloud storage," in Proc. IEEE Global Commun. Conf 2013, pp. 695-700.

[10] C.-l. Fan, S.-Y. Huang, and W.-C. Hsu, "Hybrid data deduplic in a cloud environment," in Proc. Int. Conf. lnf. Secure. Lntell. Con 2012, pp.174-177.

[11] N. Kaaniche, and M. Laurent, "A secure client-side deduplicat scheme in cloud storage environments," in Proc. 6th lnt. Conf. New Technol., Mobility Secure., 2014, pp.1-7.

[12] Z. Yan, M.J. Wang, Y.X. Li, and A. V. Vasilakos," Encrypted management with deduplication in cloud computing," IEEE Cloud Comput. Mag., Vol.3, no.2, pp.28-35, Mar.-Apr.2016.

[13] Z. Yan, X. Y. Li, M. J. Wang, and A.V. Vasilakos, "Flexible d access control based on trust and reputation in cloud computing," II Trans. Cloud Compt., 2015. doi: 10.1109/TCC.2015.2469662.

[14] J. Hur, D. Koo, Y. Shin, and K. Kang, "Secure data deduplicat with a dynamic ownership management in cloud storage," IEEE Tr Knowl. Data Eng., Vol.28, no.11, pp. 3113-3125, Nov.2016.