International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 5 – APRIL 2018 (SPECIAL ISSUE).

DUPLICATE FINDER: PROVIDING THE SECURITY FOR STORING THE BIG DATA IN CLOUD ENVIRONMENT

#1 K.Prema, *2 I.Sheeba Angeline, *3 S.Deepika

#1,2 B.E (CSE), Kings Engineering College, Chennai,India
#3 Associate Professor, Kings Engineering College, Chennai, India

***Abstract*Cloudserviceadoptionhasincreasedinrecent yearswiththeadoptionofcloudservice; many of the companies are using this cloud to store and process Big Data. Security measures provided by the service providers might not be enough to secure the data in the cloud. And, we discussthepracticalsolutiononwhichweareworkingat themomenttoprotectthedatainacloud environment by dividing the big data into small data files. These small files can be stored in the cloud without completely compromising the data in cloud effectively leading to securing the Big Data in a cloud environment. .Along with this, we have implemented HMAC algorithm and chunkingtechniquetofindthededuplicationinclouden vironmentforreducingstoragespaceand network bandwidth. To the best of our knowledge, existing approaches, either solely focus on securingthedataincloud.Tosolvesuchproblem,wedev elopanefficientalgorithmwhichreduce the storage space and networkbandwidth.**

Key words: HMAC Algorithm, Chunking technique, Network Bandwidth

I INTRODUCTION

Cloud Computing is a technology which storing massive amount of data. Recent technological advancements in cloud computing, internet of things and social network, have led to a deluge of data from distinctive domains over the past two decades. Cloud data centers are awash in digital data, easily amassing petabytes and even exabytes of information, and the complexity of data management escalates in big data. The goal of cloud computing is tofindingduplicate files for increasing the storage efficiency and providing security.

In, cloud computing providing security, finding duplication in complex format files like video, image, document is one of the major problem . However, all these schemes are oblivious to the content and format of application files, and cannot find the redundancy in files with complex format, like image,videofileHence, their space efficiency can be further improved by exploiting application awareness. This is a codesign of storage and application to optimize deduplication based storage systems when the deduplicated storage layer has extensive knowledge about the file structures and their access characteristics in the application layer.

As shown, the conventional deduplication schemes always improve performance in single-node scenario or distributed scenario without considerations on application awareness. In the latest research works, application aware duplicate detection has been adopted to single-node deduplication to improve deduplication efficiency with low system overhead.

In this paper, we propose HMAC algorithm, to find the duplicate files by generating hash value, to support big data management in cloud storage. Our solution takes aim at large-scale distributed deduplication with thousands of storage nodes in cloud datacenters which would most likely fail in the traditional distributed methods due to some of their shortcomings in terms of global deduplicationratio, single- node through-put, data skew, and communication overhead.

The main idea behind HMAC is to optimize distributed deduplication by exploiting application awareness, data similarity and locality in streams. Our main idea in this work is to see the possibility of implementing a simple Chunking mechanism and deduplication method to store the Big Data files in a cloud environment by splitting them into the small files. Our HMAC algorithm can efficiently increasing the storage efficiency by deduplication method and increased the network efficiency, security and reducing time while uploading and downloading files using chunking technique.

Related Work:

A. Nyre and M. G. Jaatun[1]proposed the probabilistic approach to Information control. In this paper we propose a probabilistic approach to information control based on trust management systems. Our solution provides the user with a view of the amount of information that any given entity probably has received through redistribution, in order to determine the level of aggregation the entity can perform. C. Rong, H. Cheng, and M. G. Jaatun proposed

[2]Securing big data in the cloud by protected mapping over multiple providers. In this paper we present an alternative approach which divides big data among multiple. It protects the mapping of the various data elements to each provider using a trapdoor function. Our initial analysis indicates that this is an efficient and secure approach forsecuring big data. A.Bessani,M. Correia, and B. Quaresma,

[3] proposedDepsky: dependable and secure storage in a cloud-of-clouds. In this paper we present DepSky, a system that improves the availability, integrity, and confidentiality of information stored in the cloud through the encryption, encoding, and replication of the data on diverse clouds that form a cloud-of-clouds.We observed that our protocols improved the perceived availability, and in most cases, the access latency, when compared with cloud providers individually.

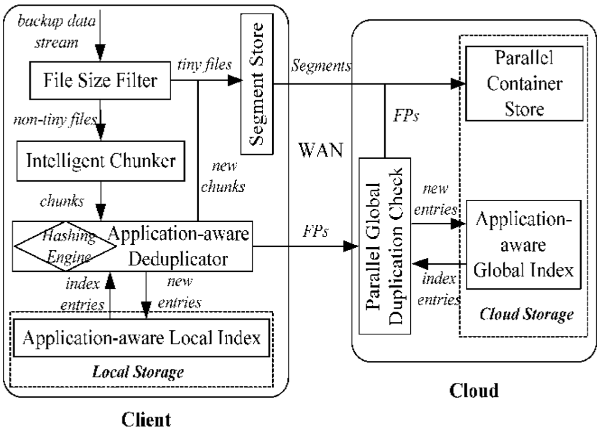
C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou[4] proposed toward secure and dependable storage services in cloud computing.Theproposeddesignallowsusers to audit the cloud storage with very lightweight communication and computation cost. Considering the cloud data aredynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks. M. G. Jaatun, G. Zhao,A.
V. Vasilakos, A. Nyre, S. Alapnes, and Y. Tang[5] proposed the design of a redundant array of independent net-storages for improvedconfidentialityincloudcomputing. As long as each segment is small enough, an individual segment discloses no meaningful information to others, and hence RAIN is able to ensure the confidentiality of data stored in the clouds. We describe the inter- cloud communication protocol, and present a formal model, security analysis, and simulationresults.

Issues in cloud:

In cloud computing redundancy of data is one of the main issue. And also it leads to reduce the storage space. Data deduplication is one of the hottest technologies in storage right now because it enables companies to save a lot of money on storage costs to store the data and on the bandwidth costs to move the data when replicating it offsite for DR. This is great news for cloud providers ,because if you store less, you need less hardware.If you can deduplicate what you store,you can better utilize your existing storage space,which can save money by using what you have more efficiently . In existing approaches, thededplication method cannot be applied in complex format files. So it does not provide the optimal solution. It may lead to the decrease the storage efficiency. Therefore it increase the network bandwidth. The amount of data storage increases quickly in open environment. So, storage efficiency is one of the main challenge in cloud environment. HMAC algorithm is used for finding the duplicate files in complex format files. It increase the storage efficiency and network efficiency in cloud.

System architecture:

The architecture aims to provide the increased storage efficiency and network efficiency. In our proposed system Hash based message authentication codealgorithm and chunking technique is used. In this HMAC algorithm generates the unique hash valueforeachfiles store in the cloud



.

Fig1:-Architecture

In chunking technique it divide the big data files into small files .Join will allow us to combining all small data files to form original big data file.HMAC algorithm solve the storage efficiency problem effectively. The client and director play a vital role, if client can upload their file the hash engine generate the unique hash value by using this hash value it identify the duplicate files. If the file redundancy occurs it cannot be store in the cloud. While downloading the file director sends one time password to client mail. By using file key and one time password client can download the file .It provides the increased storage and network efficiency by sing HMAC algorithm and chunking technique, and it improve the security level by using DES algorithm.

Module description:

User Module: In this module a user has to upload its files in a cloud server, he/she should register first. Then only he/she can be able to do it. For that he needs to fill the details in the registration form. These details are maintained in a database. In this module, any of the above mentioned person have to login, they should login by giving their name and password.

FILE UPLOADING/DOWNLOADING PROTOCOL:

Upload: In this module user upload his file. The uploaded file is encrypted format. In this encryption process we are implementing BEM (Bit Exchanging Method). The uploaded file is not stored into the cloud server. The Director audits user file then only user files is uploaded to the cloud server.

Download: In this module user download the files in decrypted format. The downloaded file is encrypted format the user enter the correct key then only it is decrypted. Decryption process also we are using Bit Exchanging Method algorithm only.

Secure Auditing Protocol: In this module, Director have to login, they should login by giving their username and password. Secure Duplication protocol is sending the status for all files duplication status. Director is audit the uploaded all file status. Director approves only non-duplicate files then only it is stored in cloud.TPA audit user file it is duplicate means Director not provide the uploading permission to that file. Uploaded file is non-duplication means Director give the activation for that file.Then only that file is stored into the cloud server.Director audits the file storage also.File Storage auditing and Deduplication auditing are clearly shown in an graphical format.

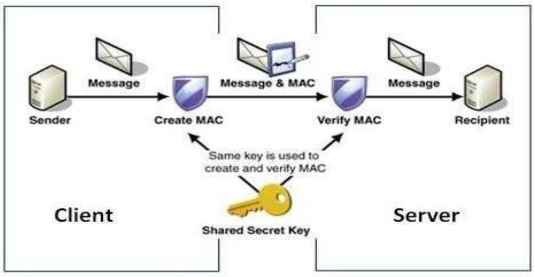


Fig 2:- HMAC algorithm

Block-level deduplication System: We consider block level deduplication in that file is divided into block and check deduplication for block. For encryption we are going to use Bit Exchanging Algorithm.Block-level and byte-level data deduplication methods deliver the benefit of optimizing storage capacity. When, where and how the processes work should be reviewed for your data backup environment and its specific requirements before selecting one approach over another. Data deduplication can generally operate at the file, block or byte level thus defining minimal data fragment that is checked by the system for redundancy. Hash algorithm generates a unique identifier hash number for each analyzed chunk of data. It is then stored in an index and used forfiguring out duplicates the duplicated fragments have the same hash numbers.

Bit Exchanging Method: Encryption taken on the secret message files using simple bit shifting and XOR operation. The bit exchange method is introduced for encrypting any file.Read one by one byte from the secret data and convert each byte to 8 bits. Then apply one bit right shift operation. Divide the 8 bits into to blocks and then perform XOR operation with 4 bits on the left and 4 bits on the right side.The same thing repeated for all bytes in the file.

Director Module: Director module is used to audit the file. Director can login with username and password. After login, Director can view all files uploaded by the user. And then checks the duplicate file. Finally auditor, activate the file.

Algorithm:

Hash based message authentication code algorithm:In cryptography, an HMAC (abbreviated as either keyed-hash message authentication code or hash-based message authentication code) is a specific type of message authentication code (MAC) involving a cryptographic hash function and a secret cryptographic key. It may be used to simultaneously verify both the data integrity and the authentication of a message, as with any MAC. Any cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC- X, where X is the hash function used (e.g. HMAC-MD5 or HMAC-SHA1). The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and the size and quality of the key.

HMAC generation uses two passes of hash computation. The secret key is first used to derive two keys – inner and outer. The first pass of the algorithm produces an internal hash derived from the message and the inner key. The second pass produces the final HMAC code derived from the inner hash result and the outer key. Thus the algorithm provides better immunity against length extension attacks.

An iterative hash function breaks up a message into blocks of a fixed size and iterates over them with a compression function. For example, MD5 and SHA-1 operate on 512-bit blocks. The size of the output of HMAC is the same as that of the underlying hash function (e.g., 128 or 160 bits in the case of MD5 or SHA-1, respectively), although it can be truncated if desired.

Algorithm

```
functionhmac (key,message)
 if (length(key) >blocksize) then
     //keys longer than blocksize are shortened key=hash
 (key)
 end if
 if (length (key) <blocksize) then
     //keys shorter than blocksizeare zero- padded
 key=key //zeros (blocksize-length (key)) end if
      //where blocksize is that of the underlying hash
 function
 o_key_pad=[0x5c * blocksize] ⊕ key i_key_pad=[0x36
 * blocksize] ⊕ key
   //where // is concatenation
 returnhash(o_key_pad            //hash(i_key_pad
 //message)) end
 function
```

Chunking Technique:

Data de-duplication is an emerging technology that introduces reduction of storage utilization and an efficient way of handling data replication in the backup environment. In cloud data storage, the de- duplication technology plays a major role in the virtual machine framework, data sharing network, and structured and unstructured data handling by social media and, also, disaster recovery. In the deduplication technology, data are broken down into multiple pieces called “chunks” and every chunk is identified with a unique hash identifier. These identifiers are used to compare the chunks with previously stored chunksandverifiedforduplication.Sincethe chunking algorithm is the first step involved in getting efficientdata de-duplication ratio andthroughput,itisveryimportantinthede- duplication scenario. In this paper, we discuss different chunking models and algorithms with a comparison of their performances.

Algorithm

```
fill (queue, pointInStream)
pointsProcessed = 0
 root = allocRoot()
 buildRecurse(queue, root)
 free(root)
 return
 defshouldRefine(node,  queue):  a
 = contains(node, back(queue)) b =
 size(queue) >leafMax
 c = isSubdivisible(node)
 return a and b and c
 defbuildRecurse(queue, node): if
 isEmpty(queue) then
 return
 if !contains(node, front(queue))then
 return
 ifshouldRefine(node, queue) then for
 octant = 0...7 do node.child[octant] =
 allocNode(node, octant)
 buildRecurse(queue, node.child[octant]) end
 finalizeInner(node)
```

```
node.idx = tell(nodeOutStream)
write(nodeOutStream, node) else
while contains(node, next(pointInStream)) do
push(queue, next(pointInStream))
advance(pointInStream)
end
finalizeLeaf(queue, node, pointsProcessed) node.idx =
tell(nodeOutStream
write(nodeOutStream, node) fill(queue, pointInStream)
end
```

DES algorithm:

The Data Encryption Standard is an outdated symmetric key method of data encryption.DES woks by using the same key to encrypt and decrypt a message,so both the sender and the receiver must know and use the same private key. Once the go to symmetric key algorithm for the encryption of electronic data. To accomplish encryption, most secret key algorithms use two main techniques known as substitution and permutation. Substitution is simply a mapping of one value to another whereas permutation is a reordering of the bit positions for each of the inputs. These techniques are used a number of times in iterations called rounds. Generally, the more rounds there are, the more secure the algorithm. A non-linearity is also introduced into the encryption so that decryption will be computationally infeasible without the secret key. This is achieved with the use of S-boxes which are basically non-linear substitution tables where either the output is smaller than the input.

Algorithm

```
Cipher (plainBlock[64], RoundKeys[16, 48],
cipherBlock[64])
permute (64, 64, plainBlock, inBlock,
InitialPermutationTable)
split (64, 32, inBlock, leftBlock, rightBlock)
for (round = 1 to 16)
mixer (leftBlock, rightBlock, RoundKeys[round])
if (round!=16) swapper (leftBlock, rightBlock)

combine (32, 64, leftBlock, rightBlock, outBlock)
permute (64, 64, outBlock, cipherBlock,
FinalPermutationTable)
mixer (leftBlock[48], rightBlock[48], RoundKey[48])
copy (32, rightBlock, T1) function (T1, RoundKey, T2)
exclusiveOr (32, leftBlock, T2, T3) copy (32, T3,
rightBlock)
swapper (leftBlock[32], rigthBlock[32]) copy (32,
leftBlock, T)
copy (32, rightBlock, leftBlock) copy (32, T,
rightBlock)
function (inBlock[32], RoundKey[48], outBlock[32])
permute (32, 48, inBlock, T1,
ExpansionPermutationTable) exclusiveOr (48, T1,
RoundKey, T2) substitute (T2, T3, SubstituteTables)
permute (32, 32, T3, outBlock,
StraightPermutationTable)
substitute (inBlock[32], outBlock[48],
SubstitutionTables[8, 4, 16])
for (i = 1 to 8)

row ¨ 2 \ inBlock[i \ 6 + 1] + inBlock [i \ 6 + 6]
col ¨ 8 \ inBlock[i \ 6 + 2] + 4 \ inBlock[i \ 6
+ 3] +
2 \ inBlock[i \ 6 + 4] + inBlock[i \ 6 + 5] value =
SubstitutionTables [i][row][col] outBlock[[i \ 4 + 1] ¨
value / 8; value ¨ value mod 8

outBlock[[i \ 4 + 2] ¨ value / 4; value ¨ value mod 4
outBlock[[i \ 4 + 3] ¨ value / 2; value ¨ value mod 2
outBlock[[i \ 4 + 4] ¨ value
```

CONCLUSION

Nowadays Cloud computing is the trending and emerging technology. The one of the main issue in cloud computing is security related issue and storage efficiency. Cloud computing stores the data and resources in open environment. The amount of data storage increases quickly in open environment. So, storage efficiency and providing security is one of the main challenge in cloud environment. HMAC algorithm and chunking technique is used to find the duplicate files by generating hash value, to support big data management in cloud storage. It also improve the storage and network efficiency and providing high security . A number of techniques have been proposed by researchers for deduplication. However there are many gaps to be filled by making these techniques more effective. More work is required in the area of cloud computing to make it acceptable by the cloud service consumers. This paper presents increased storage efficiency and network efficiency in cloud environment using deduplication method.

REFERENCES

[1]A. Nyre and M. G. Jaatun, “A probabilistic approach to information control,” Internet Technology Journal, vol. 11, no. 3, pp. 407–416, 2010.

[2]C. Rong, H. Cheng, and M. G. Jaatun, “Securing big data in the cloud by protected mapping over multiple providers,” in Digital Media Industry & Academic Forum (DMIAF). IEEE, 2016, pp. 166–171.

[3]A. Bessani, M. Correia, B. Quaresma, F. Andr´e, and P. Sousa, “Depsky: dependable and secure storage in a cloud-of-clouds,” ACM Transactions on Storage (TOS), vol. 9, no. 4, p. 12, 2013.

[4]C. Wang, Q. Wang, K. Ren, N. Cao, and

W. Lou, “Toward secure and dependable storage services in cloud computing,” IEEE transactions on Services Computing, vol. 5, no. 2, pp. 220–232, 2012.

[5]J. Singh, B. Kumar, and A. Khatri, “Improving stored data security in cloud using rc5 algorithm,” in Engineering (NUiCONE), 2012 Nirma University International Conference on. IEEE, 2012.

[6]H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, “Racs: a case for cloud storage diversity,” in Proceedings of the 1st ACM symposium on Cloud computing. ACM, 2010, pp. 229–240.

[7]J. Surbiryala, “A framework for improving security in cloud computing,” in 2nd IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA 2017). IEEE, 2017.

[8]M. G. Jaatun, , “The design of a redundant array of independent net-storages for improved confidentiality in cloud computing,” Journal of Cloud Computing: Advances, Systems and Applications, vol. 1, no. 1, p. 13, 2012.