ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com
International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 2, Special Issue 12, April 2016.

DYNAMIC CROSS CLOUD MULTICOPY DATA POSSESSION USING MC-ARS (MULTI COPY ACTIVE REPLICATION SUPPORT TECHNIQUE

A.Karkuzhali[#1] , K.Tamilselvan[#2]

Department of Computer Science and Engineering

A.V.C College of Engineering

ABSTRACT

Cloud computing involves sharing of resources without having to maintain local servers or any physical devices. Outsourcing data to servers avoid storing data locally and organizations are opting for outsourcing data to remote cloud service providers (CSPs). Customers can rent the CSPs storage infrastructure to store and retrieve almost unlimited amount of data by paying fees metered in gigabyte/month. For an increased level of scalability, availability, and durability, some customers may want their data to be replicated on multiple servers across multiple data centers. The more copies the CSP is asked to store, the more fees the customers are charged. Therefore, customers need to have a strong guarantee that the CSP is storing all data copies that are agreed upon in the service contract, and all these copies are consistent with the most recent modifications issued by the customers. This propose technique MC-ARS (MULTI COPY ACTIVE REPLICATION SUPPORT TECHNIQUE) scheme that has the following features: 1) it provides evidence to the customers that the CSP is not cheating by storing fewer copies; 2) it supports outsourcing of dynamic data, i.e., It supports block-level operations, such as block modification, insertion, deletion, and append; 3) it allows authorized users to seamlessly access the file copies stored by the CSP; 4) Identification and Reconstruction of corrupted file blocks.

KEYWORDS CSP, Data center, Data Relication,Cloud computing

INTRODUCTION

Cloud computing is a model for enabling ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services.

Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand.

OUTSOURCING data to a remote cloud service provider (CSP) allows organizations to store more data on the CSP than on private computer systems. Such outsourcing of data storage enables organizations to concentrate on innovations and relieves the burden of constant server updates and other computing issues. Once the data has been outsourced to a remote CSP which may not be trustworthy, the data owners lose the direct control over their sensitive data. This lack of control raises new formidable and challenging tasks related to data confidentiality and integrity protection in cloud computing. The confidentiality issue can be handled by encrypting sensitive data before outsourcing to remote servers. As such, it is a crucial demand of customers to have a strong evidence that the cloud servers still possess their data and it is not being tampered with or partially deleted over time.PDP is a technique for validating data integrity over remote servers. In a typical PDP model, the data owner generates some metadata/information for a data file to be used later for verification purposes through a challenge-response protocol with the remote/cloud server. One of the core design principles of outsourcing data is to provide dynamic behaviour of data for various applications. This means that the remotely stored data can be not only accessed by the authorized users, but also updated and scaled (through block level operations) by the data owner.

In the proposed design the data owner is allowed to do block level modifications of data. For verifying the data integrity map entry table is maintained with contains physical and logical order, data and number of times the data gets updated. Initial value of count will be one. The data owner requests CSP for modifying data. Whenever the data gets updated its count in map entry table also gets updated. If the count gets incremented due to some modification by an intruder then we can determine that the data integrity has been affected. The affected data block can be replaced with the original data and get replicated in all the servers.

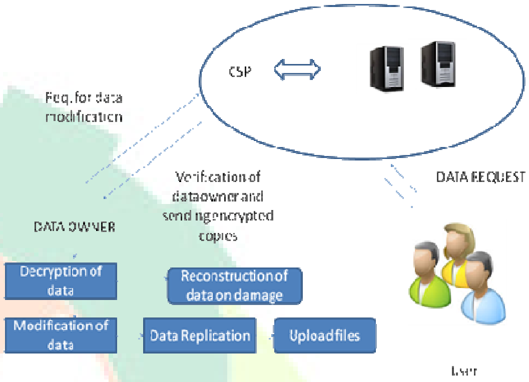
EXISTING SYSTEM

The existing system allows the data owner to archive the file and gives access to block level modifications. The interaction between the authorized users and the CSP is considered in their scheme, where the authorized users can seamlessly access a data copy received from the CSP using a single secret key shared with the data owner. it does not supports the feature of identifying the indices of corrupted copies.

PROPOSED DESIGN

The proposed system involves identification and reconstruction of corrupted files indices. The corrupted data copy even after complete damage can be reconstructed using duplicate copies on other servers. It supports outsourcing multicopy of dynamic data to cloud server and also allows the data owner to do block level modifications. This module involves generation of symmetric key for encrypting the files to get uploaded in the server. It uses DESede algorithm available in javax.crypto package. The user will upload the data to share and he/she is asked to give the number of file copies to replicate and the key for encrypting the file. The data owner can share the key with the users whom he/she wants to share the file. After the completion of this phase the file will be replicated and get uploaded in different servers. For each data block tags will be provided. The logical order will be different from physical ordering. Two Hash Maps (data structure in Java) is used for storing the data blocks one will store the original order and the another will have the logical order. The data owner will have the original order of data and in the server the data block will be in different order unknown to the cloud service provider even. The data owner can request the CSP for modification of file block. After the verification of data owner, he/she will be allowed to modify the data. The changes that have been done in the current data block will automatically get updated in all copies that have been uploaded in different servers. After the updation of data block, the Map-entry table also need to get updated. This map entry table is used for maintaining integrity and for checking the consistency of data block. For all the file copies only single table is created. The block initially created the entry will be one when it is updated, the value will be incremented by one. The corrupted file copies can be identified using the entries present in the map-entry table. If the entry in map table for a data block updated without data owner interference, then it can said that the file block has been corrupted. Once the data owner finds that the file has been corrupted without his interference he can able to replace the data block with the correct ones and the user will avail the correct data thereby reconstructing the data.

ARCHITECTURE DIAGRAM


CSP
Req. for data modification
Verification of dataowner and send ngencrypted copies
DATA REQUEST
DATA OWNER
Decryption of data
Reconstruction of data on damage
Modification of data
Data Replication
Uploadfiles
User

ALGORITHM

UPLOADING OF FILE BY DATA OWNER

Reordering data

Let HM be the HashMap with String as value and line number as key used for storing original order of data

Let var i=0;

For each line l,in file f

If l is empty,then l=”:;”

Hm.put(i,l)

Increment i

Generate Random number with range between 0 and line count and store in an array say ar

Check for duplication of random numbers which are generated above

```
Let hm1 be the Hash Map with String as value and line number as key used for storing logical order of data

Let var cnt =0;

Set s= hm.entrySet();

Iterator it=s.iterator();

While(it.hasNext())

{

 Get the current object

String k=(String)Get the key of current object

Store it in hm1 -->(k,ar[cnt]);

Increment cnt;

}

Update both the physical order and logical order in database for that particular username and file
```

SECRET KEY GENERATION

```
Let key be user given key

SecretKey key = new SecretKeySpec(ky, "DESede");

Create cipher instance cipher<--Cipher.getInstance("DESede");

Set encrypt mode for cipher<--(Cipher.ENCRYPT_MODE, ky);

Byte buf[]←encrypted bytes

Encode data with base64 encoding

Return encoded bytes

Upload encrypted data to all the servers with user given replication number and store secret key in database
```

MODIFICATION OF DATA BLOCK

```
Request for data modification

Get secret key for the particular user
```

DECRYPTION

```
SecretKey key1 = new SecretKeySpec(raw, "DESede");

 Create cipher instance cipher<--Cipher.getInstance("DESede");

Set decrypt mode for cipher<--(Cipher.DECRYPT_MODE, ky);

Cipher cipher1 = Cipher.getInstance("DESede");


cipher1.init(Cipher.DECRYPT_MODE, key1);

byte[] res<--Base64.decodeBase(encryptedbytes)

                  byte[]
op=cipher1.doFinal(message);

                write to log file
```

REORDERING OF DATA

```
 Read the file

Let var lc(linecount) be 0

For(each line l in file f)

{
```

```
Int val=Get value for key lc

Read the line corresponding to val(For ex if val=3,read the 3rd line) and store it in a file

Increment lc

}
```

CONCLUSION AND FUTURE ENHANCEMENT

As said above outsourcing of data become a trend for avoiding storing files locally. There came up lot of problems while having multiple copies of data. Maintaining data Integrity is a primary problem while having copies of same data across multiple servers. The Proposed scheme is meant for checking the data integrity and finding corrupted file blocks. It also allows checking data integrity without accessing it and it is a possession free one. And moreover the corrupted file blocks have been reconstructed with the duplicate data kept on other servers. This scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract. Moreover, the scheme supports outsourcing of dynamic data, i.e., it supports block-level operations such as block modification, insertion, deletion, and append. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP. The system can be further extended to add intimating the user and the data owner whenever the data has been accessed by the intruders.

REFERENCES

[1] Provable Data Possession at Untrusted Stores - Giuseppe Ateniese, Randal Burns, Reza Curtmola, in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY,USA, 2007, pp. 598–609.

[2]Demonstrating data possession and uncheatable data transfer -Decio Luiz Gazzoni Filho, Paulo S ergio Licciardi Messeder Barreto, IACR (International Association for Cryptologic Research) ePrint Archive,Tech. Rep. 2006/150, 2006.

[3] Auditing to Keep Online Storage Services Honest-Mehul A. Shah, Mary Baker, Jeffrey C. Mogul, Ram Swaminathan ,in Proc. 11th USENIX Workshop Hot Topics Oper. Syst. (HOTOS), Berkeley, CA, USA, 2007, pp. 1–6.

[4]Authentication and Integrity in Outsourced Databases-Einar Mykletun, Maithili Narasimha And Gene Tsudik, ,ACM Trans. Storage, vol. 2, no. 2,pp. 107–138, 2006.

[5]Scalable and Efficient Provable Data Possession -Giuseppe Ateniese , Roberto Di Pietro , Luigi V. Mancini ,

and Gene Tsudik, ,in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm), New York, NY, USA, 2008,Art. ID 9.

[6] Ensuring Data Storage Security in Cloud Computing-Cong Wang, Qian Wang, and Kui Ren,Wenjing Lou, IACR Cryptology ePrint Archive, Tech.Rep. 2009/081.

[7] Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing Qian Wang , Cong Wang , Jin Li , Kui Ren , and Wenjing Lou, in Proc. 14th Eur. Symp. Res. Comput. Secur. (ESORICS), Berlin, Germany, 2009, pp. 355–370.

[8] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving auditand extraction of digital contents," IACR Cryptology ePrint Archive,Tech. Rep. 2008/186, 2008.

[9]Robust Remote Data Checking-Reza Curtmola,Osama Khan ,Randal Burns, in Proc. 4th ACM Int. Workshop Storage Secur. Survivability, 2008,pp. 63–68.

[10]M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," IACR Cryptology ePrint Archive,Tech. Rep. 2008/186, 2008.