Novel Framework for Public Auditing with Privacy Preserving in Cloud

Laxman Dande[1], Soppari Kavitha[2]

Laxman Dande Studing M-Tech CSE in Holy Mary Institute of Technology, keesara, RangaReddy dist.

Soppari Kavitha working as Head Department of CSE, Professor at Holy Mary Institute of Technology.

Abstract: *Cloud Computing is emerging more into the market as the demand for cloud architecture is increasing day by day. Most of the companies are opting for the cloud architecture and as the cloud being a common resource sharing platform the data should be guarded and must be monitored at regular time intervals. Coming with this paper, it is giving an opportunity for the user to maintain data with more secured manner and also audit functionality is provided to monitor the data present in cloud at any given point of time. Coming with the process of auditing earlier it was like the data from cloud was downloaded and then the auditing was done which is time consuming process and moreover the data might be hacked during this long process. In proposed paper, we have designed a system using which the auditing will be done at the cloud server and not after downloading the data from the cloud.*

Keywords: *Guarded, Auditing, hacked, Cloud Computing.*

INTRODUCTION

Cloud Computing is the technology which makes data available to the end user at any given point of time and can store huge amount of data. Size of the cloud cannot be decided it is very vast. As the size is big, in the same way the cost incurred to maintain the data in the cloud is also considered to be high because of the service made available by the cloud providers. There is a term called TPA (Third Party Auditor) the role of TPA is to authenticate the user logging into the application and provide security to the cloud from intruders. Intruder is a user who tries to access the data from the system without authenticating self or trying to do some crime by taking the data from the cloud. Cloud providers provide the space to the companies to upload their data into cloud and make it available to their clients whenever it is needed for them. The main reason for going to cloud architecture is that whenever the company has got more clients and the data to be served to the clients is more which cannot be withhold by server then comes into picture the external storage which refers to Cloud. Cloud is the database storage which is provided by some other organizations and as many companies depend on Cloud there is an authenticator called TPA who will validate the entries into Cloud and will not allow the users who are non-genuine.

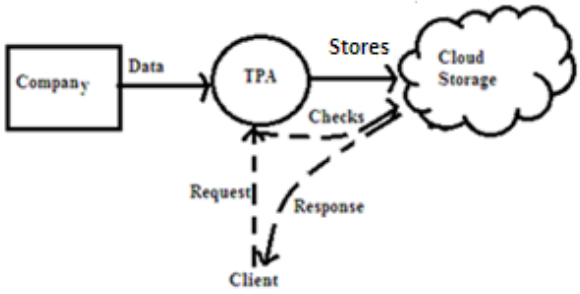


Fig 1: Cloud work flow

As shown above we can see that the company and Client both are connected to Cloud via TPA. The role of TPA is very important in the Cloud Computing processing. The point that arises is whether the data uploaded into cloud is safe or not, for this purpose the setup is coming up with the implementation of encryption algorithms for the data that is made available in the cloud just to make sure that data reaches the correct person but not the intruders.

To tell about the real instances of Cloud computing we use many applications in day to day usage, few of which are discussed below. Consider the Gmail application which is very common these days for mailing purpose and in this application we will go with communication between two people i.e. a person has send a mail with excel sheet attached and the user who received that file does not have any MS-Office in the system to download and read it, but without even downloading it the user is given the flexibility to read the data without downloading and this is possible because of Cloud Computing which is employed with the Gmail application.

Second very famous application which almost many of the users use is Facebook, a social network. This application is used by many people across globe and so the database required for this is very huge and a single server cannot withstand this many users and their data. The other thing that can be noticed here is that a user of facebook can have any id for the login i.e. Yahoo id or Gmail id or Rediff id is accepted as a username,

the point to be understood is that all the servers of various companies are kept in a single room like architecture called Cloud and because of which a genuine user of any of these domains can easily access even Facebook application.

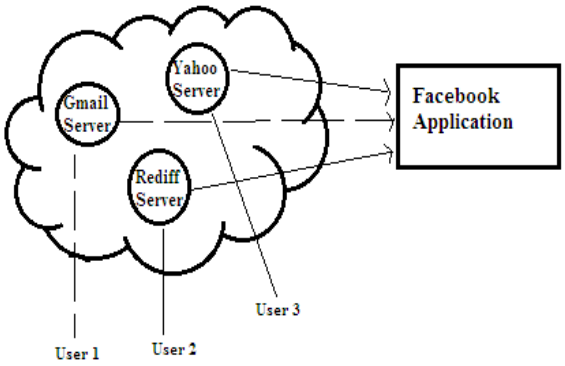


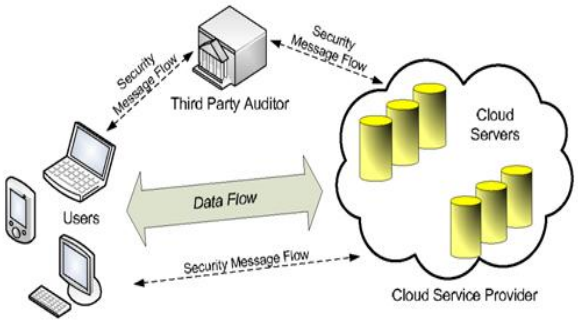
Fig 2: Social community Facebook Architecture

BACKGROUND

All the cloud service providers manage with an enterprise-class infrastructure which offers scalable, secure and reliable environment for all the users, at a much lower marginal price due to the sharing nature of resources. It is very common for users to use cloud storage services to share data with others in a team, as data sharing imposes to be a standard feature in most cloud storage offerings, including applications Drop box and Google Docs.

The data integrity in cloud storage, however, is subjected to skepticism and scrutiny, as data stored in an un-trusted cloud can easily be corrupted or lost, due to various reasons like hardware failures and human errors. To safeguard the integrity of cloud data, it is best assumed to perform public auditing by introducing a third party auditor (TPA), who offers auditing services with more powerful computation and communication abilities than the normal users.

The first provable data possession (PDP) mechanism is to perform public auditing and is designed to check the correctness of data stored in an un-trusted server, without retrieving the entire data from cloud. Moving a step ahead, Wangetal is designed to construct a public auditing mechanism for cloud data, so that during public auditing, the content of private data belonging to a personal user is not disclosed to the third party auditor because of the security reasons. In this paper, to solve the problem related to privacy issue on shared data, we propose a novel privacy-preserving public auditing mechanism. More specifically, we utilize ring signatures to construct homomorphic authenticators, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier. We further extend our mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and can improve the efficiency of verification for multiple auditing tasks. Our proposed theme is compatible with random masking, which has been utilized in WWRL and can preserve data privacy from public verifiers. Moreover, we also leverage index hash tables from a previous public auditing solution to support dynamic data. A high-level comparison among our proposed architecture and existing mechanisms is presented. Security to the complete is maintained not only during the upload time but also during the download time, security to the data is provided by the application.



The architecture of cloud data storage service

Fig 3: Work flow

The above block diagram shows the architecture of the proposed work. We can clearly see the complete work flow of the process like the data flows from the data owner through the TPA to the cloud and on request from user going back to data user. In this complete process also an important task i.e. the audit to make sure the data is present in the cloud or not. To perform the task of auditing in the cloud various steps have been taken into consideration like,

1. KeyGen
2. SigGen
3. GenProof
4. VerifyProof

As mentioned above, these are the 4 steps involved in the auditing process. For each and every file being uploaded to cloud will have a unique key and this key will be shared to the end users at the time of download, the same key must be used for utilizing the data. To generate the key we are depending on the Java API. Once the key is generated, data will be uploaded to cloud and in case any time if the user wants to check whether the file is safe on cloud or not, there is a possibility to verify whether the data available on cloud or not and if it is made available then is it as per the

requirement or not. These kinds of activities will be monitored by the admin dynamically without downloading the data. **Key Generation** can be done using any of the cryptography techniques or by using the JAVA API. Java API is providing us the class named "Random" from the utility package and thus by using this class we can generate the key as per our requirement i.e. 6byte or 8byte or depending on the user requirement. Sample code showing the usage of this class,

```
String
s1="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLM
NOPQRSTUVWXYZ0123456789";

String key1="";
      int j=0;

java.util.Random r=new java.util.Random();

              for(j=0;j<6;j++){

key1=key1+s1.charAt(r.nextInt(s1.length()));

              }
```

From the above snippet, a key will be generated and it can be used while downloading the content from the cloud and it will be shared among all company users.

Signature generation is the next step in the auditing process where a kind of hash value will be generated for the data before uploading it to the cloud and the same value will be stored as a reference for future purpose.

```
String plaintext = "your text here";
MessageDigest m =
MessageDigest.getInstance("MD5");
m.reset();
m.update(plaintext.getBytes());
byte[] digest = m.digest();
BigInteger bigInt = new BigInteger(1,digest);
String hashtext = bigInt.toString(16);
// Now we need to zero pad it if you actually want the
full 32 chars.
while(hashtext.length() < 32 ){
  hashtext = "0"+hashtext;
}
```

[Snippet showing hash algorithm generating hash value]

Whenever, the TPA or admin wants to verify the data integrity, a command will issued which will take the data and a new signature will be generated for the data which is stored at cloud and this signature will be compared with the signature that was generated during the time of uploading the data to the cloud. By comparing these two signatures admin or TPA can conclude or take any decision in regards to the data integrity or availability at the cloud. This step involves internally the other two mechanisms i.e. **generating proof and verifying proof**. As already discussed generating proof is the action which will be done by admin during the time of auditing the data at cloud. Once the signature is generated for the data at the cloud it is then verified using the next step in the process i.e. verifying proof. This step is nothing but comparing both the proofs for a specific file at cloud. In this manner, admin or TPA can implement the technique of auditing the data for multiple files stored at cloud without downloading the data. Our proposed solution makes it possible for the user to dynamically do the audit for multiple files at the cloud without downloading it.

Say suppose the hash value is generated for a specific file and again a new hash value will be generated at cloud for the data it maintains, both these values will be compared and a decision will be taken against it.

Let X1=Q1H1 (Hash Value nothing but a temporary value generated for the file content before uploading) and,
X2=Q1H2 (Hash value for the data that is present at cloud), in this manner two different hash values will be generated and both of these will be compared like,
X1==X2, means the data is safe at cloud and if
X1!=X2, then it means the data which is there at cloud is not the same as it was at the time of upload. Necessary actions need to be taken by the admin or the TPA.

Advantages of our Proposed Work:

As in today's date we can see that many companies are moving towards multi cloud architecture to enhance the business opportunities. When moving towards the multi-cloud architecture or towards the cloud architecture the major threat that comes in picture is security and data availability in all the clouds. This proposed work gives relief to the data owners from this problem. The major advantage is that data owner can anytime check for the data availability in the cloud unlike other real-time applications. This ensures data owner with the data availability in cloud at any given point of time.

CONCLUSIOSN AND FUTURE WORK

Paper addresses to the problem of privacy preserving and also the data integrity at cloud and thereby with the implementation of the solution to this problem we are overcoming with the issue of data corrupt or misleading the data users. Our work shows the auditing implementation process in a better manner than the one which is generally implemented; here the auditing takes place at cloud and not locally. By

implementing the auditing at cloud end will gives better efficiency to the application and thereby increasing the faith of the data owners in storing the data at cloud and makes them trust the cloud services. It is very important concept in the real time servers as the data will be used by many users across globe and it must be made available to the users at any given point of time.

Our future scope of work would be on the security issue in addition to the auditing process. We would propose an advanced cryptographic algorithm using XOR Encryption operations for securing data at cloud to ensure the data stored in the cloud cannot be misused by any unknown users accessing the server/cloud data. And we do all this manual terminology to automated in future.

References

"Byzantine disk paxos: optimal resilience with Byzantine shared memory", Distributed Computing, 2006, pp. 387-408.

C. N. Yang, "New visual secret sharing schemes using probabilistic method," *Pattern Recognit. Lett.*, vol. 25, pp. 481–494, Mar. 2004.

S. J. Lin, S. K. Chen, and J. C. Lin, "Flip visual cryptography (FVC) with perfect security, conditionally-optimal contrast, and no expansion,"*J. Vis. Commun. Image Represent.*, vol. 21, pp. 900–916, Nov.2010.

G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, "Cryptography for general access structures," *Inf. Computat.*, vol. 129, no. 2,pp. 86–106, Sep. 1996.

Reader in Computer Science, Royal Holloway, University of LondonDistributed Systems, Cloud Computing, Fault-Tolerance - I. Abraham, G. Chockler, I. Keidar and D. Malkhi.

G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalableand Efficient Provable Data Possession," in Proc. InternationalConference on Security and Privacy in Communication Networks(SecureComm), 2008.

C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "DynamicProvable Data Possession," in Proc. ACM Conference on Computerand Communications Security (CCS), 2009, pp. 213–222.

Authors Profile

Laxman Dande Studying M-Tech CSE branch in Holy Mary Institute of Technology,Keesara, Ranga Reddy. Affiliated to JNTU, Hyderabad, A.P.,

Soppari Kavitha working as a Professor, Head Department of Computer Science Engineering at Holy Mary Institute of Technology, Keesara, Ranga Reddy. Affiliated to JNTU, Hyderabad, A.P., India. My research interests are Image and Data Processing, Information Security.