Enhanced Distributed Accountability for Data Sharing in Cloud

Vaidehi R. Nerkar
PG Student, Dept. of Computer Engineering
Smt. Kashibai Navale College of Engineering
Pune, Maharashtra, India

S. P. Kosbatwar
Asst. Professor, Dept. of Computer Engineering
Smt. Kashibai Navale College of Engineering
Pune, Maharashtra, India

Abstract— **Cloud computing provides scalable services to be easily accessible to access over the Internet whenever needed. An important feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. With new emerging technology user also fears of losing his data. To overcome this problem, we propose a enhanced decentralized information accountability framework which keeps track of the actual usage of the users' data in the cloud. We propose an approach that enables enclosing our logging mechanism together with users' data and policies. We use JAR programmable capabilities to both create a dynamic and traveling object, and ensure authentication and automated logging locally to the JAR files. In this paper we give an experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches**

Keywords- Auditing, Accountability, Cloud computing, Data sharing, Logging.

I. INTRODUCTION

Cloud computing is use of hardware and software services which are delivered as a service over the Internet. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. In this case, cloud providers update and operate application software in cloud and cloud users access the software from cloud clients. The data processed on clouds are outsourced, causing many issues related to accountability .Such issues are becoming a significant barrier to the wide adoption of cloud services. To solve the above problem it is necessary to provide an effective mechanism for users to monitor the usage of their data. Some conventional access control approaches developed for databases and operating systems, are not suitable in this situation, because of following features.First, data is outsourced by the cloud service provider (CSP) in the cloud. Second, entities are allowed to join and leave the cloud in a flexible manner.As a result, data handling in the cloud goes through a complex and dynamic hierarchical service chain which does not exist in conventional environments.

A. Cloud Information Accountability

Cloud Information Accountability framework proposed in this paper, have automated logging and distributed auditing of data access performed by any customer, carried out at any point of time at any cloud service provider.

It has two important components: logger and log harmonizer.

Logger: The logger is the component which is strongly coupled with the user's data. logger is accessed when data are downloaded. Its main task is automatic logging access to data items.

1. Logger:

It may also be configured to ensure that access and usage control policies associated with the data are honored. Logger includes nested java JAR file which contain file to protect and access policies. Logger component contain the log entries or log records which is helpful for monitoring the data usage. When Logging mechanism occurs, any access to the data in the JAR file is recorded and the details are sent to the log record. Each log entry is encrypted before appending to log record.

2. Log harmonizer:

The log harmonizer performs auditing. The log harmonizer generates the master key it is used to hold the decryption key for IBE key pair. If the path between log harmonizer and client is not trusted than descryption is carried out on client side.it supports two strategies : push and pull.

It supports following two mechanisms:

Push mode: This refers to logs being periodically sent to the data owner or stakeholder

Pull mode: This refers to an alternative approach whereby the user (or another authorized party) can retrieve the logs as needed.

B. Distributed Accountability

We take CIA (Cloud information accountability) framework, as a base work for our project and propose solutions called framework for accountability and auditing in cloud to solve the disadvantages of CIA framework.

User identity:

To reduce the overhead in proving access control based on user, we will use Attribute based access control. Based on user attributes, access policy will be defined. User matching to this attributes will be givenaccess to data items. This way user management will become easy and also user revocation is easy.

Access of data items:

For each data item, attribute is defined for accessing data and the user matching to their attributes policy will get only the portion he/she entitled to.

To overcome these problems we propose an approach, called Cloud Information Accountability (CIA) framework based on Information Accountability. The proposed CIA framework provides end-to-end accountability in a distributed manner. This combines access control, usage control and authentication. By means of CIA, data owners can track whether the service level agreements agreed and enforce access and control rules. The designing of the CIA framework presents certain *challenges*, including unique identifying CSPs, ensuring the reliability of the log, adapting to a highly decentralized infrastructure, etc . Data owner will send their data along with access control policies and logging policies that they want to enforce, in JAR files, to cloud service providers. Any access to the data will trigger an automated logging mechanism local to the JARs.

.

Fig 1. Remote connection between android device and local PC

II. RELATED WORK

Cloud computing has come across a range of important privacy and security issues. Such issues are due to that, in the cloud, users' data and applications reside at least for a certain amount of time on the cloud cluster which is owned and maintained by a another party. Main part to concern is that why the information is required by other party,and to whom that data will be forwarded. Some of the work is carried in this area with respect to accountability. The basic idea is that the users personal data is uploaded on the cloud in an encrypted form, and the processing is performed on the encrypted data. The output of the processing is taken by the privacy manager to reveal the correct result. Once data is being enclosed privacy manager will not guarantee protection. In, the authors present a layered architecture for addressing the end-to-end trust management and accountability problem. A representative work of this area is given by. The authors propose the usage of policies attached to the data and present a logic for accountability data in distributed settings. In most commercial and legal transactions, the ability to hold individuals or organizations accountable for transactions is important. Accountability is an important aspect of any computer system. It assures that every action executed in the system can be traced back to some entity. Accountability is even more crucial for assuring the safety and security. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. Cloud computing enables highly scalable services to be easily consumed over the Internet on an as-needed basis, we propose a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. In particular, we propose an object centered approach that enables enclosing our logging mechanism together with users' data and policies. We influence the JAR programmable capabilities to both create a dynamic and travelling object and to ensure that any access to user's data will trigger authentication and automated logging local to the JARs. To strengthen user's control, we also provide distributed auditing mechanism. In Crispo an Ruffo proposed an interesting approach related to accountability in case of delegation. Delegation is complementary to our work, in that we do not aim at controlling the information workflow in the clouds. Similarly, Jagadeesan et al. recently proposed a logic for designing accountability-based distributed systems. In a summary, all these works stay at a mandatory theoretical level and do not include any algorithm for tasks like logging. To the best of our knowledge, the only work proposing a distributed approach to accountability is from Lee and colleagues. The authors have proposed an agent-based system specific to grid computing. Distributed jobs, along with the resource consumption at local machines are tracked by static software agents.

III. SYSTEM MODEL

A. Design

How system will work is elaborated through flow chart diagram shown in figure 2. Diagram is showing a set of activity which is carried out step by step. Each step represents the activity of the system.

Project will be implemented in four steps.

Step 1 : In first phase the user will upload his/her jar file on the cloud.

Step 2 :In second step the customer who wants to access the data,will request to view data.

Step 3 :In the third step authentication is performed to check whether the person who requested the data is a authenticated person.

Step 4: In the forth step access is given to that members who have made the payment,and the person who has not done payment would get to view only the abstract part of the paper.

B. Block diagram

Architecture of the proposed system is shown in following figure 3. At first every user creates a pair of public and private keys using identification based algorithm. By using generated key user creates a logger component which is JAR file.The JAR files contains control rules which specify how the cloud serve should access the content itself. Then that JAR file is send to the cloud service provider which he is subscribed to.Open SSL based algorithm is used to authenticate cloud service provider to JAR. After successful authentication the user (service provider) will be allowed to access the data which is enclosed in JAR file. As for logging, each time there is an access to the data, the JAR will automatically generate a log record. Encrypt it using the public key distributed by the data owner,and store it along with the data. Our proposed framework prevents various attacks such as detecting illegal copies of users' data.

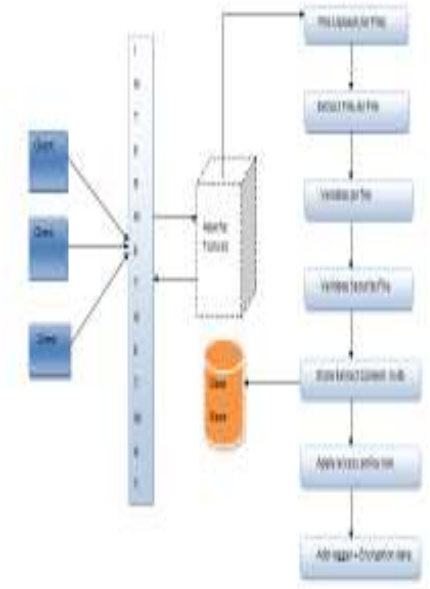

Figure 1. Architecture of system

The system mention in the paper have below drawbacks that we will implement.

1. JAR file don't have any signature , we will can add security and signature policy for the jar file.

2. In our paper data is not only bounded with uploading photos in JAR , It is not compulsory that it should have only photos , we can add videos as well as valuable document also.

3. Encryption and decryption for the jar content to avoid the spoofing attack.

4. We will deploy the system on heterogonous cloud.

5. Security is also provided where data is stored mean you are storing the photos and video into the data base we will store into the encrypt manner.

6. OTTB security will be provided in case any password loss or need to perform any important transaction ,for example user need to purchase and photo , then he has to go under OOTB security.

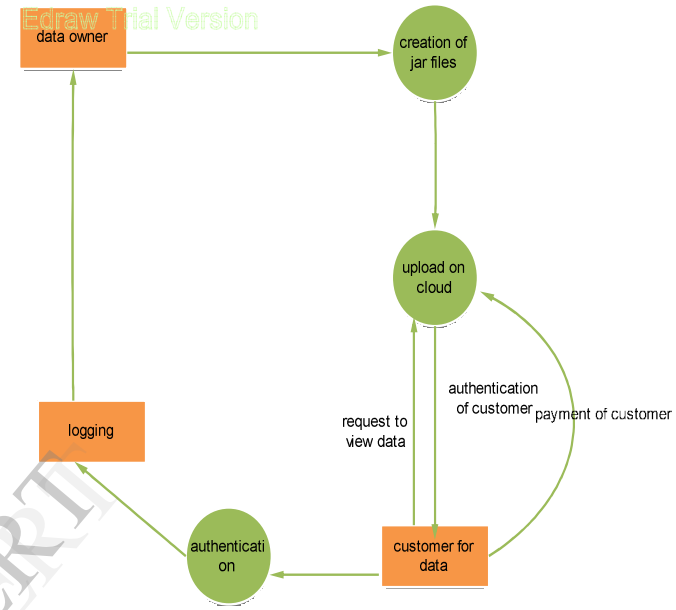

Figure 2. Flow chart of system

IV. PROPOSED SYSTEM

A. Implementation details:

1. Web Portal :It is web base application which will provide the user interface ,hosted on the apache tomcat web server. Through this application user is able to login , upload the data in jar format and define the access rule Viewer is also able to view the uploaded data.This web `application will be hosted on tomcat which was installed on private cloud ubantu.

Figure 4. Deployment diagram

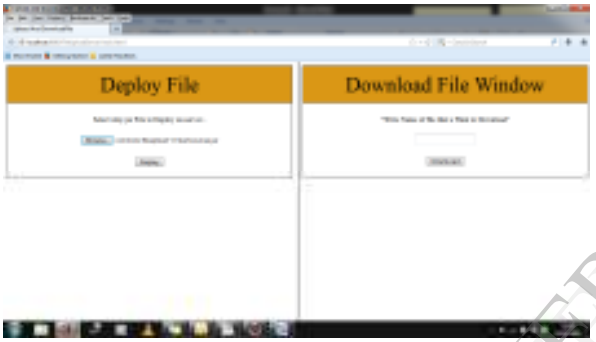

Figure 5. Upload the JAR file

Figure 6.Successfuly upload of JAR file.

Figure 7. download of JAR file.

In fig 4 you have to deploy the JAR file which the user (data owner) have to upload on the cloud. After the successful upload of file you will get a message ”your file is successfully uploaded on server” as shown in figure 6 . After successful uploading of file, the file can be downloaded as shown in the figure 7.
That file can be downloaded from download window.

2. Data Extraction & Storage: This is the second module which performs the job of the extraction of the data and after extraction it will try to store the with db. We are using oracle as back end for the system because it can support multimedia file and able to store into encrypted format, which will add the security to project .After data extraction we will delete the data from the file system as it might happen that overflow and data short of disk space.
3. Access Rule / Policy: While user is sending the jar file , that jar file containing signature file .Signature file contains information about access rules. Below is the successful access rule file format.

```
<access-rule>
        <user><user>
        <FileCollection>
                <File></File>
                <Read></Read>
                <Download></Download>
        </FileCollection>
</access-rule>
```

After reading the access rule , access rule injector will give the store the file into db accordingly. When any user wanted to see any file , it has pass through access rule validation.

4. Logger: Logger is the last module, It will log each and every activity of the user. So the owner of the data will get clear information how is accessing the data and all. This log information is also helpful for him for accouting purpose. We are using apache log4j framework and logs were store into html format. User is able to see the logs on web portal.

ACKNOWLEDGMENT

I am very thankful to my guide for his valuable guidance in this work and thankful to our Head of Department for his valuable support. I am also thankful to IJERT committee for giving me an opportunity to present my paper.

REFERENCES

[1] Smitha Sundareswaran, Anna C. Squicciarini, Member,“IEEE Transactions On Dependable And Secure Computing Vol.9 No.4 Year 2012” and Dan Lin Ensuring distributed accountability for data sharing in the cloud
[2] S. Sundareswaran, A. Squicciarini, D. Lin, and S.Huang,“Promoting Distributed accountability in the cloud,” Proc. IEEE Int’l Conf. Cloud Computing, 2011Pretschner,M.Hilty,F.Schuo¨tz.C.Schaefer, and

T.Walter,"Usage Control Enforcement: Present and Future," IEEE Security & Privacy, vol. 6, no. 4, pp. 44-53, July/Aug. 2008.Cheng-Lin Tsao, Sandeep Kakumanu, and Raghupathy Sivakumar School of Electrical and Computer Engineering Georgia Institute of Technology Atlanta, GA 30332, USA (clt-sao,ksandeep,siva)at ece.gatech.edu, "SmartVNC: an effective remote computing Solution for smartphones" in 2011

[3] "Information accountability," Comm.ACM, vol. 51, no. 6, pp. 82-87, 2008.

[4] "Accountability as a Way Forward for Privacy Protection in the Cloud," Proc. First Int'l Conf. Cloud Computing,09.

[5] Logic for auditing accountability in decentralized systems," Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust , pp. 187-201, 2005..

[6] The Design and evaluation of accountable grid computing system," Proc. 29thIEEE Int'l Conf. Distributed Computing Systems (ICDCS '09),pp. 145-154, 2009.

[7] Usage Control Enforcement: Present and Future," IEEE Security & Privacy, vol. 6, no. 4, pp. 44 53,July/Aug. 2008.

[8] S. Etalle and W.H. Winsborough, "A Posteriori Compliance Control," SACMAT '07: Proc. 12th ACM Symp. Access Control Models and Technologies, pp. 11-20, 2007.Kiran Karra, Virginia Polytechnic Institute and State University, "Wireless distributed computing on the android platform", 2012

[9] X. Feng, Z. Ni, Z. Shao, and Y. Guo, "An Open Framework for Foundational Proof-Carrying Code," Proc. ACM SIGPLAN Int'l Workshop Types in Languages Design and Implementation, pp. 67-78,2007.

[10] J. Park and R. Sandhu, "Towards Usage Control Models: Beyond Traditional Access Control," SACMAT '02: Proc. Seventh ACM Symp. Access Control Models and Technologies, pp. 57-64, 2002.
Tristan Richardson, Quentin Sta_ord-Fraser, Kenneth R.Wood and Andy Hopper, Virtual network computing, Volume 2, Number 1, jan/feb 1998