Data Sharing Accountability in Cloud Computing

Shubhangi P. keni
M.E. Student, Dept. of Computer Engineering
Shah and Anchor Kutchhi Engineering College,
Chembur, Mumbai-400088, India

Shahzia Sayyad
Professor, Dept. of Computer Engineering,
Shah and Anchor Kutchhi Engineering College,
Chembur, Mumbai-400088, India

Abstract - Cloud computing is the emerging technology which makes it possible for you to access your data from anywhere at any time. Where in traditional method computers setup requires you to be in the same location as your data storage service. A distinct feature of the cloud services is that user's data are remotely process in unknown machines that user do not own or operate. Data handling goes through complex and dynamic hierarchical chain through various abstraction layers. While enjoying this new emerging technology, user's fear of losing control of their data particularly (health & financial data, etc.).Hence cloud computing can become a significant barrier to the wide acceptance of cloud services. Accountability traces each aspect of any data sharing on data usage and access control in cloud where it accounts each activity in the system. Here we review the Cloud Information Accountability (CIA) framework proposed by many authors which provides automated logging for ensuring accountability and distributed auditability performed by any entity, carried out at any point of time at any cloud service provider. To address this problem, we propose a methodology to keep track of actual usage of the user's data in the cloud computing environment. To strengthen user's control we also provide auditing mechanisms. Apart from that we are going to enhance the integrity of the log files.

Keywords: accountability, auditability, cloud computing, data sharing, security, integrity

I. INTRODUCTION

Cloud computing is the utilization of hardware & software resources like (RAM, CPU cycle, storage, network bandwidth, virtual machine, web server & database etc.), which are used for computation delivered as a service over a network or Internet with minimum cost effect and IT services. While enjoying the convenience & uniqueness of cloud computing such as on demand self-service, rapid elasticity, pay as you need, broad network access& multi-tenant nature of working environment etc., it is very much difficult to keep control on data and process which are carried out remotely on someone's physical or virtual machine's location that user don't know. The data processed on clouds are often subcontracted, leading to a number of issues related to accountability including the handling of personally identifiable information. People find the loss of control over data or information which leads to lack of trust in cloud. There are four trust components such as privacy, security, accountability and auditability in cloud computing [1].

Preventive controls to solve privacy & security problem has been focused or researched by using many encryptions and hashing techniques. But detective controls to solve accountability & auditability problem is still little focused or researched. Accountability is the verifying or checking the authorization polices (e.g. user's roll based read, write, copy, & execute permission over the data or process). The end user is allowed to access the data as per their access privileges, which they specifies while registering to access the data in the cloud and authentication is provided and data usage will be verified. Auditing plays an important role for monitoring the irregularities in cloud environment. It can be carried out periodically or randomly or as per the requirement of the stakeholders or cloud Server Provider. Accountability & auditability both plays an important role of keeping an eye on ongoing process or activity in real time carried out in cloud computing in order to empower the trust relationship between the data owner, end user and cloud service provider.

II. RELATED WORK

This section related works addressing data sharing accountability and auditability in the cloud computing environment. Many author approaches the framework to address automated logging and distributed accountability known as Cloud Information Accountability Framework (CIA) to solve the problem [3, 4, 5, 6, 7, 8, and 9].

The overall CIA framework, combining data, users, logger and harmonizer is sketched in Fig.2.1 firstly data owner will upload data, he will first generate public and private keys using IBE Weil pairing scheme described by author [3, 4, 5, 6, and 7]. The cloud information accountability framework can be explained by given step number in the figure. Data owner data set access policies (Access Control List rules). At the beginning cloud service provider creates public key and private key based on identity based encryption scheme. Using the generated key, the user will create logger component which is Java archives files, to store the data items. The Java archive file includes a set of simple access control rules specifying whether and how the cloud server and possibly other user are authorized to access the content itself. Data owner sends the Java archives file to cloud service provider where certificate authority certified the cloud service provider and the user. Authentication of cloud service provider done using secure socket layer certificate is used. End user authentication is done by using Secure Assertion markup language based certificate is used. There are two major component of CIA, one is logger and second one is the log harmonizer. The logger is strongly coupled with user's data. Whenever, there is access to data, the java archives will automatically create log records, encrypt it using public key of data owner and store it Its main tasks include automatically logging access to data item that it contains, and encrypting the log records using the public key of the content owner, and periodically send to log harmonizer. The logger is responsible for generating the error correction information for each log record and send same to the log harmonizer. The second one is the log harmonizer is responsible for auditing. The log harmonizer generates master key. The encrypted log files can be decrypted and their integrity can be verified. The log files can be accessed by data owner for auditing purpose. The decryption carried out on the client end if path between the log harmonizer and the client is trusted. The harmonizer sends the key to the client in a secure key exchange. It support two auditing strategies such as push and pull mode. Under push mode log files is pushed back to data owner periodically, wherein in the pull mode is on demand approach, log files is obtained by the data owner as often as requested. The log harmonizer is responsible of merging log record as well as handling the log file corruption. The logger and log harmonizer are implemented as lightweight and portable JAR files. The JAR file implantation provides automatic logging function.

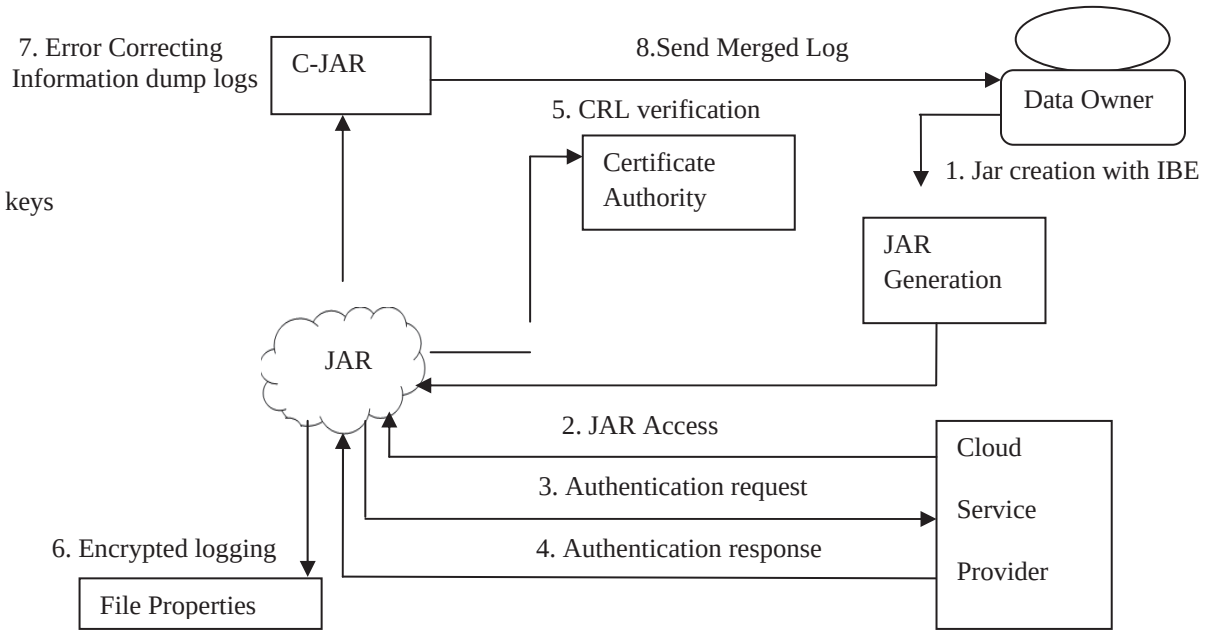

Figure 2.1 Cloud Information Accountability Frameworks [3, 4, 5, 6, 7, 8, and 9].

III. PROPOSED SYSTEM ARCHITECTURE

The Proposed system architecture is platform independent and decentralized. The System Architecture lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. By means, the data owners can track not only data access but also ensure access control list is being honored. Apart from that system is enhancing the integrity of log record. As log record is very important data while accounting the end user's activity. This System architecture can also enforce access control and usage control rules if needed.

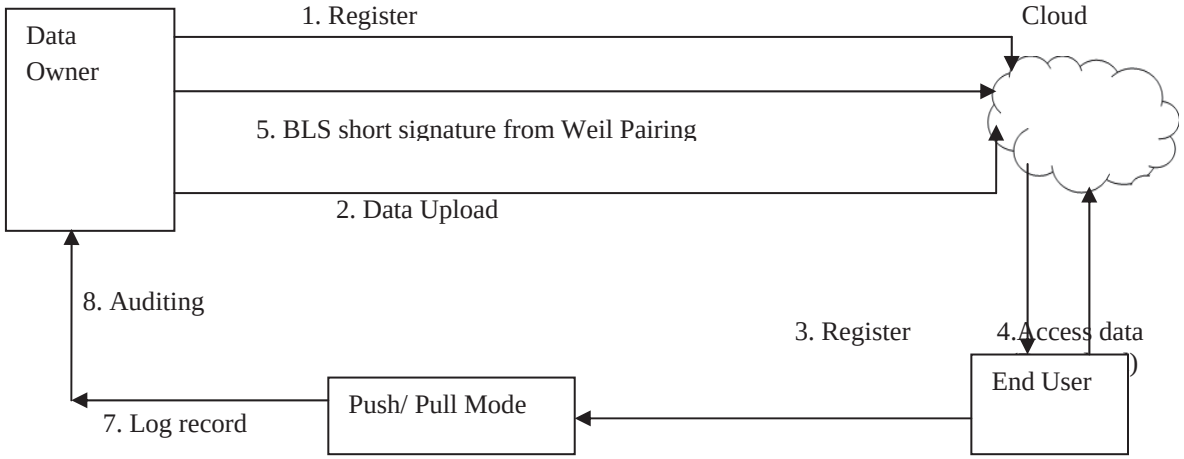

Fig.3.1 Overview of System Architecture

3.1 Main Modules

1. Data Owner registration: - Data owner must register their details in the cloud server. Cloud server stores the data owner details. The authentication is done by verifying the password stored in the database.
2. Data upload: - . Data Owner uploads the data in the cloud server.
3. End User registration: - . Every user must register their details in clouds server. The cloud server stores data owner details.
4. Access Data (download):- When the end user request to get data. The authentication is done by verifying the password stored in the database. As soon as user access the data and perform any activity like read, edit or download logger will create corresponding log file. This log file contains the user session details.
5. Data owner auditing: - The log files get decrypted at the data owner site. And merged log record is used for auditing and analysis purpose.
6. Cloud server: The cloud service provider manages a cloud data storage service. User's data and encrypted log files with their public key is saved in the cloud server.

3.2 Major Component

1. Logger: - The loggers have the details of the data owner and user who accessing the cloud. Like which user/data owner accessing the cloud server, accessed at the particular time and the ip address from which the data is requested by user.
2. Log harmonizer: - The log harmonizer has two main responsibilities to deal with copies of log records. It does not contain the user's data items being audited, but consists of class files for both a server and a client processes to allow it to communicate with its logger components. Since user's data are coupled with the logger component in a data, the logger will be copied with the user's data.
3. Auditing modes:-The log harmonizer is accountable for auditing. There are two auditing modes.

a. In push mode:- Log file send to the data owner as soon as session disconnected from end user side so that data owner may be able to who are all the accessing their data at that particular time period. This mode logs are periodically send log details to data owner
b. In pull mode: - Logs are send on demand basis to the data owner meanwhile it will be saved in cloud server. Data owner can be retrieve log files whenever they want to audit the recent access to their own data.

IV. IMPLEMENTATION

4.1 Cloud Setup (Ulteo Open Virtual Desktop)

Ulteo Open Virtual Desktop (OVD) is a virtual desktop and application delivery platform that supports Windows Remote Desktop Services and Linux hosted desktop and application sessions. OVD enables organizations to integrate and seamlessly deliver them as a secure service to clients based on Windows, Linux, MacOS, Android and IOS platforms. Ulteo OVD is open source virtual desktop. Ulteo OVD is all about mixing various applications sources into a consistant stream that can be delivered to users, depending on their needs. It's also been designed to be integrated in heterogeneous environments and inter-operate with various technologies. Web application archive file can be integrated in the Ulteo open virtual desktop to deliver the application.

4.2 Logger

Any access to data will trigger the generation of log files corresponding to uploaded file. The encryption is done using BLS short signature from weil-pairing-based cryptography. It enables logger component to handle authentication of entities which want to access the data stored.

Logging occurs at any access to the data, and new log entries are appended sequentially, in order of creation LR= (r1, r2 . . . rk). Each record ri is encrypted individually and appended to the log file. In particular, a log record takes the following form:
rk = (User_name, file_name, activity, ip_address, date_time, h((User_name, file_name, activity, ip_address, date_time)ri-1…r1), sig,)
Where,
rk = log record
User_name = user identification
file_name= name of the file
activity= access activity perform on user's data i.e.(read, edit, download)
ip_address= Internet protocol address of the system from which user access the data
date_time = date and Time at which activity performed
h((User_name, action, loc,T)ri-1…r1) = checksum component
sig = Signature of record created by the server
Examples :- suppose the end user with User_name Shubhangi, read the file pro from ip_address at 4:52 pm on October 20, 2015. The corresponding log record is given below.
<Shubhangi, pro.txt, read, 192.168.2.4, 2015-10-20, 16:52:30>
Here IP lookup table and use the range of the IP address to find the most probable location for improved readability

4.3 Logger harmonizer

A log harmonizer which has two main responsibilities: to deal with copies of log files and to recover corrupted logs. It does not contain the user's data items being audited, but consists of class files for both a server and a client processes to allow it to communicate with its logger components. The harmonizer stores error correction information sent from its logger components, as well as the public key, to decrypt the log records and handle any duplicate records. The duplicate records result from copies of the data owner's data. Since data are strongly coupled with the logger component, the logger will be copied together with that. The log harmonizer is accountable for auditing. It supports two auditing strategies such as push and pull mode. With the Push mode the logs are periodically send to the data owner (or auditor) by the harmonizer. In pull mode data owner can retrieve log files as per their requirement for auditing purpose.

IV. SECURITY DISCUSSION

We now analyze possible attacks to our framework. We assume that attackers may have sufficient Java programming skills to disassemble a log file and prior knowledge of our architecture.

1. Copy attack

The common attack tries to copy the log file without being noticed by data owner. But such attack found out by auditing. If attacker moves copies of log files to places where the harmonizer can't connect, copies of log files will soon become inaccessible. This because redundancy information to the harmonizer periodically. Thus logger component provides more transparency than conventional log file encryption. However if someone tries to download the log files, the actions are recorded by the logger and the log record is sent to the user. By this the data owner will be aware of his/her file download.

2. Disassembling attack

If some unauthorized person tries to access the data, first of all it is impossible as his/her integrity is checked by the authentication system before giving the access to actual data. Let us consider the person intercept between the actual user and the system and tries to hack the data. But he will receive the disassembled log record which is encrypted and if he/she need to decrypt it to get the actual data, and also breaking the encryption is computationally complex. Even if attacker is authorized user, he can access the actual content file but he is not able to decrypt any log files which are viewable only to the data owner. Adoption of Weil pairing algorithm ensures both chosen cipher text security and Plaintext security.

3. Integrity of logs

As log files contain very much important information for data access accountability. It should not be tamper in transit. To overcome this we are protecting those by generating short signature from Weil pairing. In weil paring key pair is based upon bilinearity which is computationally difficult to break.

V. EXPERIMENTAL RESULT

1. Data owner can perform the auditing by analyzing the log files. Each log record contained the information like the name of user (User_name), name of the file (file_name), access activity performed by user i.e. read, edit or download as per the permission set, the network ip_address from is the location from which user is accessing the data as well as the date and time on which user access the data in real world time. Each access to data will generate log records as shown below.

User_name, file_name, activity, ip_address, date_time

Shubhangi, per, read, 192.168.2.4, 2015-10-12, 16:52:30

nilesh, notes, edit, 192.168,3.5,2014-10-11,17:03:23

akash, assignment, download,192.168.5.1,2015-10-16,12:09:14

2. The log record can summarize the data access and usage accountability. Statistical output is the number of read, edit and downloads activity performed by the end user. On the X axis uploaded by data owner files vs. on the Y axis total number of access activity performed by end user on that file.

Fig. 4 Files access summary

VI. CONCLUSION

Cloud makes IT resources as a service. But security breached is key barrier to wide acceptance of cloud computing. To solve such trust issues some preventive controls as well as detective controls are need to be focused or researched. Preventive controls are used to lessen the occurrences of action from continuing further or taking place at all like access policy that issued to govern the user to modify or read a file or database, or network and host firewall used to blocks all unauthorized sites but allowable activity etc. Accountability and auditability strategies make cloud more trusted by lessen the security loopholes. Data usage and access control can be monitored and audited with help of accountability as well as auditability. And integrity of log files is also important factor need to be achieved.

REFERENCES

[1] Ryan K L Ko, Peter Jagadpramana, Mirandra Mowbray, Siani Pearson, Marhus Kirchberg, Qianhui Liang, Bu sung Lee, *"Trust cloud: A framework for accountability and Trust in Cloud Computing."* HP Laboratories, HPL-2011- 38

[2] K L Ryan Ko, Peter Jagadpramana, Bu Sung Lee, *"Flogger: A File-Centric Logger for Monitoring File Access and Transfers within Cloud computing environment"* HP Laboratories, HPL-2011-119

[3] Smitha Sundareswaran, Anna C.Squicciarini, member, IEEE, and Dan *"Ensuring distributed Accountability for Data Sharing in the Cloud."* Lin, IEEE transaction on dependable and secure computing Vol. 9 No.4 Year 2012

[4] Snehal Suryawanshi, Anant M. Bagade, *"Distributed Accountability for Data Sharing in Cloud"* International Journal of computer Applications (0975-8887) Volume 59- No.8, December 2012

[5] Nilupal Bose, G. Manimala *"Secure Framework For Data Sharing in Cloud Computing Environment"*, international journal of emerging technology and advanced engineering, vol-3, special issue 1 , January 2013

[6] H.Arun, R.Namarata, S,purva *"Review on Techniques to ensure Distributed Accountability for Data Sharing in the Cloud"* , international journal of advanced research in computer and communication engineering Vol.2,Issue10, October 2013

[7] H.Arun, R.Nilam, S.Purva, Pankaj Kumar Singh, Ajit Kumar & R.Krthikeyan *"Ensuring Distributed Accountability for Data Sharing in the Cloud"*, International Journal of Advanced Research in Computer Science and Software Engineering, volume 3, March 2013,www.ijarcsse.com.

[8] D. Dhivya & S.Chinnadurai, *"Cloud Information Accountability Framework for auditing the Data usage in Cloud environment*", international journal of computational engineering research, volme.03, ISSN- 2250-3005, November 2013.

[9] Xianghan Zheng1, Hui Ye1, Chunming Tang2, Chunming Rong3, Guolong Chen1 1 College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China 2 College of Mathematics and Computer Science, Guangzhou University, Guangzhou, 510006. China, 3 Department of Computer Science and Electronic Engineering, University of Stavanger, 4036, Norway *"A Survey on Cloud Accountability*" International Conference on Cloud Computing and Big Data, 630627978-1-4799-2830-9/14 2014 IEEE DOI 10.1109/CLOUDCOM-ASIA.2013.29 2013

[10] Shital A. hande, Prof Sunil B.mane *"An analysis on Data accountability and security in cloud"* International conference on Industrial Instrumentation and control college of engineering Pune , May2015