Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.581 – 587

RESEARCH ARTICLE

Secure Data Sharing in the Cloud by Maintaining Integrity using Logging Mechanism

K.Velammal[1], Mrs.L.Sheela[2]

[1]PG Student, Embedded System Technologies, Regional Centre of Anna University, Tirunelveli, India
[2]Associate Professor, Embedded System Technologies, Regional Centre of Anna University, Tirunelveli, India
[1] kiruthika.vel03@gmail.com; [2] l_sheela79@yahoo.com

Abstract—Cloud Computing is the delivery of computing and storage capacity as a service. but there is lack of confidence in trusting , because user's data are processes remotely in unknown servers. To overcome this problem we provide an object-centered technique to extend owners' full control over his own data. In particular, a logging mechanism is provided for user's data and ensures that any access to their data will trigger authentication which is used to protect user's data and also monitor the actual usage of data in the cloud. We also provide distributed auditing mechanism.

Keywords— Cloud Computing, auditing, data sharing, logging

I. INTRODUCTION

Cloud computing technology is flexible, scalable and enables services that can be easily consumed over the Internet on as-needed basis. The convenience and efficiency of this technology, however comes with privacy and security risks [6], [7]. The biggest advantage of cloud computing is the elimination of the investment in stand-alone software or servers by the user. In cloud computing technology, one can easily save the cost of data storage, software updates, management, A major Advantage of the cloud services is that users' data are processed remotely in unknown machines that users don't need to own or operate. While enjoying the convenience brought by this new technology the users' fear of losing control of their own data, because the data processed on clouds are outsourced, which leads

to a number of issues related to accountability. Data represents an extremely important asset for any organization, especially personal information, confidential data it is individual and precious to each of us if they are disclosed they will face serious consequences. Thus, cloud users want to make sure that their data are kept confidential to outsiders, including the cloud provider and their potential competitors. So it is necessary to provide flexible and fine-grained access control for service-oriented cloud computing model.

Data owner get fear about damage of his data by hacker; so there is need of security mechanism which will track usage of data in the cloud. It is necessary for monitoring data usage, in this all actions of users and server are recorded and the record is securely maintained in the server. So it helps in make trust. Currently, we focus on image files since images represent a very common content type for end users and organizations (as is proven by the popularity of Flickr) and are increasingly hosted in the cloud as part of the storage.

In our Framework we concentrate on accountability, first the data owner will set the policies for the data which he/she wants to place in cloud and send it to cloud service provider (CSP) enclosed in JAR files, any access to the data will be automatically checked for its authentication and logs record for each data item will be created and sent to data owner so that the data owner can monitor usage of his data on the cloud.

II. RELATED WORK

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. In this section we review some related works concerned with security and privacy issues in cloud. Also, we briefly discuss the work which adopt similar techniques as our approach but serve for different purposes.

A. Security and Privacy issues in cloud

Security and Privacy Risks for the related data in Cloud Computing also increases. Concern arise as in cloud it is not always clear to an individual why their personal information is requested or how it will be used or passed on to others parties. Till today, little work has been done regarding accountability and auditing in cloud and lot to be researched. Pearson et al. have proposed accountability mechanisms to address privacy concerns of end users [3] and then develop a privacy manager [2]. Their basic idea is that the user's private data are sent to the cloud in an encrypted form, and the processing is performed on the encrypted data. However, the privacy manager provides only limited features in that it does not guarantee protection once the data are being disclosed. In [6], the authors present a layered architecture for addressing the end to-end trust management and accountability problem in federated systems. First layer in this architecture is authentication and authorization layer and second layer is accountability layer. It requires third-party services to monitor. In paper [5], the authors propose a novel automatic and enforceable logging mechanism in the cloud. This is a systematic approach to data accountability through the novel usage of JAR files is proposed. Their proposed architecture is platform independent and highly decentralized ,it does not require any dedicated authentication or storage system in place but here multiple jar files (inner jars),

takes lot of time to execute and latency is noticed by data users. In [11], the authors proposed a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, but this method content may leak to the auditor because it requires the server to send the linear combinations of data blocks to the auditor. In [4], the authors extended their dynamic auditing scheme to be privacy preserving and support the batch auditing for multiple owners. However, their auditing protocols may incur a heavy storage overhead on the server. And our way of collecting user's information or auditing is simple.

B. Attribute Based Encryption

The notion of ABE was first introduced by Sahai and Waters [8] as a new method for fuzzy identity-based encryption. The primary drawback of the scheme in [8] is that its threshold semantics lacks expressibility. Several works has been done to solve the expressibility problem. In the ABE scheme, ciphertexts are not encrypted to one particular user as in traditional public key cryptography. Rather, both ciphertexts and users' decryption keys are associated with a set of attributes or a policy over attributes. A user is able to decrypt a ciphertext only if there is a match between his decryption key and the ciphertext. depending how attributes and policy are associated with ciphertexts and users' decryption keys[11]. CP-ABE is conceptually closer to traditional access control models such as Role-Based Access Control (RBAC) [10]. Thus, it is more natural to apply CP ABE, instead of KP-ABE, to enforce access control of encrypted data [11]. In a CP-ABE scheme, decryption keys support user attributes that are organized logically as a single set, so users can use all possible combinations of attributes in a single set issued in their keys to satisfy policies. To solve this problem, Bobba [9] introduced cipher text-policy attribute-based encryption (CP-ABE or ASBE for short). By all these advantages mentioned above, we are using CP-ABE technique for access control.

III. ARCHITECTURE FOR SECURE SHARING OF DATA IN CLOUD

We take CIA (Cloud information accountability) framework [5], 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.

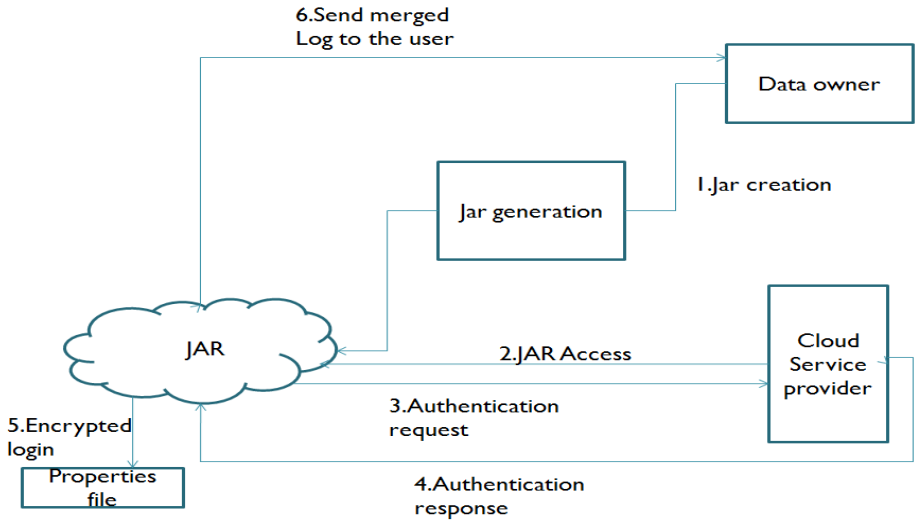

Fig 1: Architecture for accounting and auditing

At the beginning, each user creates a pair of public and private keys based on Identity-Based Encryption. Using the generated key, the user will create JAR file, to store its data items. The JAR file includes a set of simple access control rules specifying whether and how the cloud servers and possibly other data stakeholders (users, companies) are authorized to access the data. Then, he sends the JAR file to the cloud service provider that he subscribes to. To authenticate the CSP to the JAR (steps 3-4 in Fig.1), we use OpenSSL based certificates, wherein a trusted certificate authority certifies the CSP. In the event when the data is access by a user, we employ SAML-based authentication. Once the authentication succeeds, the service provider (or the user) will be allowed to access the data enclosed in the JAR. The JAR will automatically generate a log record, and it is encrypted by public key distributed by the data owner, and sends this to the data owner periodically.

IV. IMPLEMENTATION

The various modules are as follows:

A. **Jar Generation**

The JAR file contains a set of access control rules specifying whether and how the cloud servers and possibly other data interested party (users, companies) are authorized to access the content itself. Depending on the configuration settings, the JAR will provide usage control.

B. **Logger Creation**

We implement the programmable capability of JARs to conduct automated logging. A logger component is a JAR file which stores a user's data items and corresponding log files. The main responsibility of the outer JAR is to handle authentication of entities which want to access the data stored in the JAR file. The data owners may not know which CSPs that are going to handle their data. Hence, authentication is specified according to the servers'.

C. Log Record Generation

Log records are generated by the logger component. Logging occurs if there is any access to the data in the JAR, and new log records appended sequentially, in order of creation LR = r_1; . . . ; r_k. Each record r_i is encrypted individually and appended to the log file.

D. Mode Setting

To allow users to be timely and accurately informed about their data usage, our distributed logging mechanism is combined with auditing mechanism. We implement two auditing modes:

a. **Push mode**. In this mode, the logs are periodically pushed to the data owner

b. **Pull mode**. This mode allows auditors to retrieve the logs anytime when they want to check the recent access to their own data.

V. Performance Study

Here we have developed our framework in java platform. We tested the framework by uploading and downloading image files, as image files are common content type for users and organizations nowadays (for example Flickr). The framework is light because the data's are stored in cloud and to do that time taken is less. Next, the JAR module acts as compressor by compressing the files which reduces the memory space.

Few snapshots of our framework

Logging window

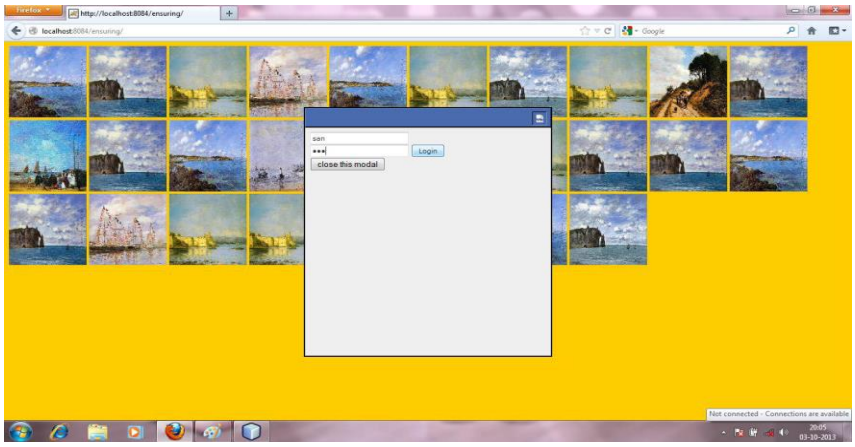

Fig 2. Logging window

Fig.2 shows logging window. Every access to the user's data should be correctly and automatically logged. This requires integrated techniques to authenticate the entity that accesses the data, and identify the user who is requesting the data. If we intend to download the image a logging window will appear contain user name and password, this authentication request is send to the server, After verifying the username and password authentication response send by the server.

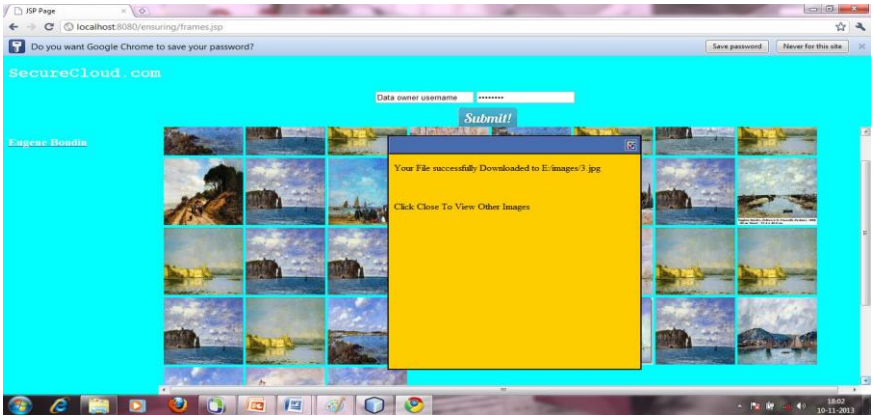

Fig 3: File Download

Fig.2 shows file download. After verification if person is an authenticated one, then file can be download in client's system.

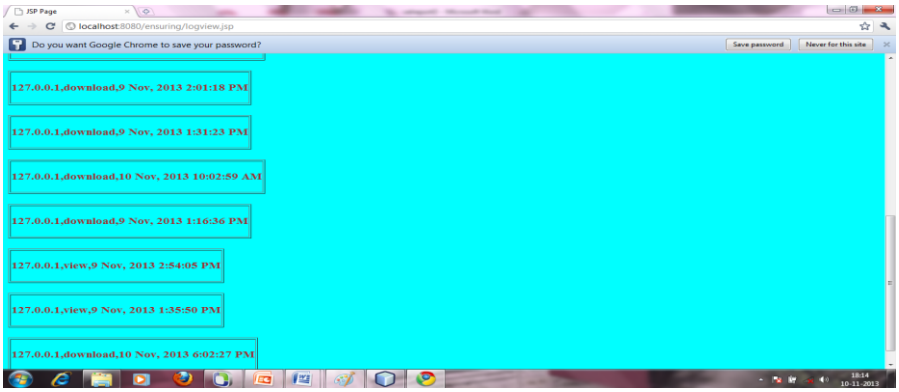

Fig 3: Client's Log Information

Fig.3 shows the log information of the particular owner's data. The log records are placed in log monitor server, if owner want to see the log record, user needs to enter user name and password and required key that is given by log monitor server to the owner. After authenticated request is given by the owner he can view the log records related to his files. So he can identify the usage made on his data.

A. **Security study**

In this section let's analyze some possible attacks on our framework.

1) Attacks on JAR files:

The common attack that we can assume is accessing the data in JAR file without being noticed. But such attack can be found out by auditing. However if someone tries to download the JAR 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 JAR file download.

2) Unauthorized user:

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's consider the person intercept between the actual user and the system and tries to hack the data. But he will receive the disassembled Jar file and 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.

VI. CONCLUSION

In this paper we have proposed distributed accountability and auditing in cloud for automated logging mechanism and also monitoring the data usage by auditing. This framework ensures the user by protecting their data using programmable JAR and also by keeping user's data privacy by limiting the data consumer by giving access policies. The importance of this framework is data owner can monitor their data usage by effective auditing mechanism.

For future enhancement we would like to concentrate more on JAR authentication and we have started to work on social media's like Facebook and twitter for user information auditing by using social media integration technique.

REFERENCES

[1] A.Squicciarini, S.Sundareswaran and D.Lin, "Preventing Information Leakage from Indexing in the Cloud," Proc.IEEE Int'l Conf. Cloud Computing, 2010.

[2] S. Pearson , Y. Shen, and M. Mowbray," A privacy Manager for Cloud Computing," Proc. Int'l Conf. Cloud Computing (cloudcom), pp.90-106,2009

[3] S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud, " Proc First Int'l conf. Cloud Computing, 2009.

[4] C.Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM. IEEE, 2010, pp. 525–533.

[5] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "A Logic for Auditing Accountability in Decentralized Systems," Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005.

[6] HP Cloud website.

[7] S.Pearson, "Taking Account of Privacy when Designing Cloud Computing Services".

[8] As A. Sahai and B. Waters, "Fuzzy identity based encryption," in Proc.Acvances in Cryptology—Eurocrypt,2005.

[9] R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in Proc ESORICS, Saint Malo, France, 2009

[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in Proc. IEEE Symp. Security and Privacy, Oakland, CA, 2007.

[11] Q. Wang, C. Wang, K. Ren, W. Lou and J. Li,"Enabling public auditability and data dynamics for storages security in cloud computing", in INFOCOM.IEEE,2010,pp. 525-533.

[12] ZhiguoWan, Jun'e Liu,Robert H. Deng, "HASBE: A Hierarchical Attribute-Based Solution for flexible andScalable Access Control in Cloud Computing".

