ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Review on Techniques to Ensure Distributed Accountability for Data Sharing in the Cloud

H.Arun[1], R.Nilam[2], R.Namrata[3], S.Purva[4]

Professor, Department of Computer Engineering, ZES's Dnyanganga College of Engineering & Research, Pune, India[1]

Student, Department of Computer Engineering, ZES's Dnyanganga College of Engineering & Research, Pune, India[2,3,4]

Abstract:Cloud computing is being paid a great attention among users, from individuals at home to the government bodies making it possible for us to access our information from anywhere at any time. An important aspect of this technology is that user's data are handled in unknown machines that users do not own or operate. So users might have fear of losing confidential data and henceforth losing their privacy in the cloud. This can become a substantial hindrance to the wide adoption of cloud services. But this drawback can be overcome by a novel approach, namely Cloud Information Accountability (CIA) framework.

The CIA framework conducts automated logging and distributed auditing of appropriate access performed by any entity, carried out at any point of time at any cloud service provider using programmable capabilities of JAR.

Keywords: Cloud computing, accountability, data sharing, logging mechanism, and auditing mechanism.

I. Introduction

Cloud computing is an emerging paradigm in the computer industry that puts entire computing infrastructure hardware and software, applications etc. online. It uses internet and remote, central servers to maintain data and applications. The cloud computing core concept is simply, a geographical shift in the location of our data from personal computers to a centralized server or 'cloud'. Cloud Computing is a step towards the evolution of on demand information technology services and products .Cloud computing is a means by which highly scalable and fully technology based services can be easily consumed over the internet on an as-needed basis. Although there are many benefits to adopt Cloud Computing but still there are some significant barriers to adoption. The convenience and efficiency of this technology comes with security risks and data privacy. Significant barrier to the adoption of cloud services is the user's fear of losing confidential data and privacy in the cloud. Privacy is an important and basic human right that encompasses the right to be left alone, to provide this right many techniques are proposed under different systems and security models. There are possibilities that data can be outsourced by the direct cloud service provider (CSP) to other entities in the cloud and these entities can also allocate the task of data management to others, and so on. Being flexible in nature entities are allowed to join and leave cloud on their wish. This cause's data handling takes place through a complex and dynamic hierarchical service chain which does not exist in conventional environments.

To overcome the above stated problems, a new approach called Cloud Information Accountability (CIA) framework was proposed, based on the concept of information accountability [1].This CIA framework provides end-to-end accountability in a highly distributed fashion. By means of the CIA[2], data owners can track whether the service-level agreements are being honored, but also make obligatory access and usage control rules as needed. Associated with the accountability feature two distinct modes for auditing: push mode and pull mode have also been coupled with this approach.

II. Literature Survey

In this section review related works addressing security in cloud. Security issue is very important in cloud there are many techniques available so here is review of all these.

A. *Security and Privacy issues in cloud*

The author of [5] provides a language which allowsto provide data with policies by agent; here agent has to prove their action and authorization to use particular data. In this technique data owner attach Policies with data that contain a description of which actions are allowed with which data, but there is the problem of Continuous auditing of agent. But there work has also provided solution that monitors incorrect behaviour of agent and agent has to give justification for their action, after that authority will check the justification. A privacy manager mechanism was given by S. Pearson in which the user's data is in encrypted form in cloud and evaluation is done on encrypted data. The privacy manager makes readable data from result of evaluation manager to get the correct result. As in obfuscation data is not present on Service provider's machine so there is no risk with data and hence data is safe on cloud. But this solution is not suitable when input data is large where this method can still require a large amount of memory [3]. The authors of [4] presents mechanism in which policies are decided by the parties that use, store or share the data irrespective of the jurisdiction in which information is processed. But data processed on service provider is in unencrypted at the time of processing so there is a risk of data leakage which becomes disadvantage of this mechanism.

There is a method of dynamic auditing protocol proposed by authors in [7] which supports the dynamic operations of the data on the cloud servers, but this method may leak the data content to the auditor as it requires the server to send the linear combinations of data blocks to the auditor. In [8], the authors extended their dynamic auditing scheme to be privacy preserving and support the batch auditing for multiple owners. However, due to the large number of data tags, their auditing protocols may incur a heavy storage overhead on the server.

In [6], authors have given a three layer architecture which protects information leakage from cloud; it provides three layers to protect data. In first layer the service provider is not allowed to view confidential data, in second layer service provider should cannot do the indexing of data, in third layer user specify use of his data and indexing in policies. In this way policies always travel with data.

In paper [1], the authors propose a novel automatic and enforceable logging mechanism in the cloud. To our knowledge, this is the first time a systematic approach to data accountability through the novel usage of JAR files is proposed. Their proposed architecture is platform independent and highly decentralized, in that 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.

B. *Identity based Encryption (IBE)*

In paper [9], the authors proposed identity-based encryption scheme (IBE). The scheme has chosen cipher text security in the random oracle model assuming an alternative of the computational Die-Hellman problem. This system is based on bilinear maps between groups. The example of such map is Weil pairing on elliptic curves. They have given precise definitions for secure identity based encryption schemes and given several applications for such systems.

Using standard techniques from threshold cryptography the PKG in their scheme master-key is being distributed so that this key is never available in a single location. Unlike common threshold systems, the authors showed that robustness for their system's distributed PKG is free.

III. **CIA SCHEMA**

There is need to provide technique which will audit data in cloud. On the basis of accountability, Cloud Information Accountability (CIA) framework is one mechanism which keeps use of data transparent means data owner should get information about usage of his data. This mechanism support accountability in distributed environment. There being two major components of this framework-logger and log harmonizer.

A. *Logger Component*

The CIA framework consist a logger component which is basically a nested Java JAR file that stores a user's data items and corresponding log files. JAR file consists of one outer JAR enclosing one or more inner JARs. Each inner JAR consists of encrypted data, class files to assist retrieval of log files and a log file for each encrypted item. The encryption is done using Weil-pairing-based IBE scheme. Outer JAR consist multiple inner JARs, access policy and class file that authenticates the server or the users and one more class file for finding the correct inner JAR. This enables logger component to handle authentication of entities which want to access the data stored in the JAR file.

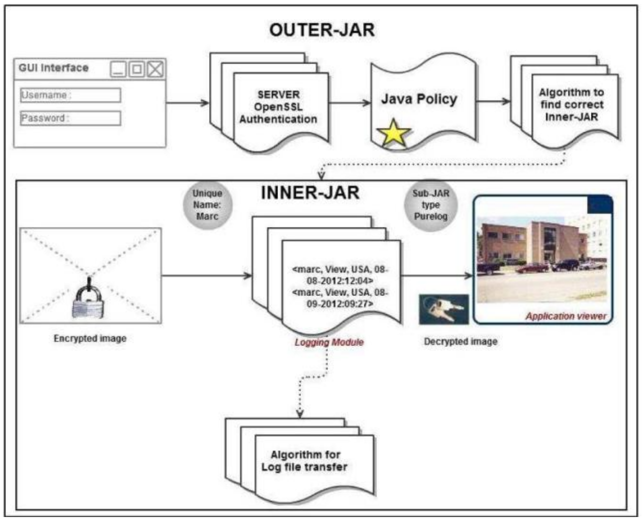

Fig.1. Structure of logger

Logging occurs at any access to the data in the JAR, 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 = (id, action, T, loc, h((id, action, T, loc)ri-1…r1), sig)
Where,
rk = log record
id = user identification

action = perform on user's data
T = Time at location loc
loc = Location
h((id, action, T, loc)ri-1…r1) = checksum component
sig = Signature of record by server

Checksum of each record is calculated and it is stored with data. Checksum is computed using hash function [37].

B. Log Harmoniser

A log harmonizer which has two main responsibilities: to deal with copies of JARs and to recover corrupted logs. The harmonizer is implemented as a JAR file. 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 user's IBE decryption key, to decrypt the log records and handle any duplicate records .Duplicate records result from copies of the user's data Jars. Since user's data are strongly coupled with the logger component in a data JAR file, the logger will be copied together with the user's data.

The log harmonizer is accountable for auditing. It supports two auditing strategies: *push and pull.*

With the Push mode the logs are periodically send to the data owner (or auditor) by the harmonizer. When the size of log file exceeds or when the time passes away for a certain period according to the temporal timer inserted as part of the JAR file.

In Pull mode data owner can retrieve log files whenever they want to check the recent access to their own data.

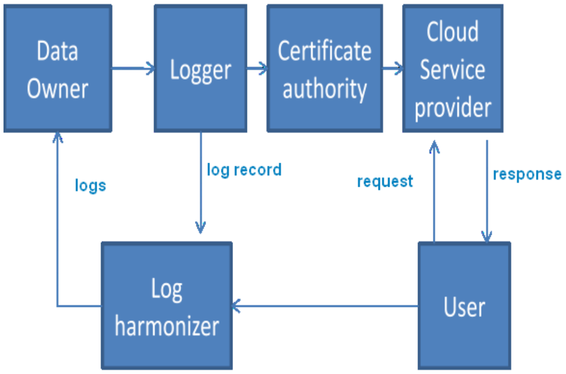

Fig.2. Auditing mechanism in the cloud

C. Generation of Encryption Keys

The conventional method to protect sensitive data from being outsourced to third parties is to store encrypted data on servers, while the decryption keys are disclosed to authorize users only.

An identity-based encryption scheme is specified by four randomized algorithms: Setup, Extract, Encrypt, Decrypt:

- Setup ( ): takes a security parameter k and returns params (system parameters) and master-key. The system parameters include decryption of a finite message space M, and a description of a finite cipher text space C. Intuitively, the system parameters will be publicly known, while the master-key will be known only to the "Private Key Generator" (PKG).

- Extract (): takes as input params, master-key, and an arbitrary ID € {0, 1}*, and returns a private key d. Here ID is an arbitrary string that will be used as a public key, and d is the corresponding private decryption key. The Extract Algorithm extracts a private key from the given public key.

- Encrypt (): takes as input params, ID, and M € M. It returns a ciphertext C € C.

- Decrypt (): takes as input params, ID, C € C, and a private key d. It return M € M.

These algorithms must satisfy the standard consistency constraint, namely when d is the private key generated by algorithm Extract when it is given ID as the public key, then

For all M€ M: Decrypt (params; ID; C; d) = M
Where C = Encrypt (params; ID; M)

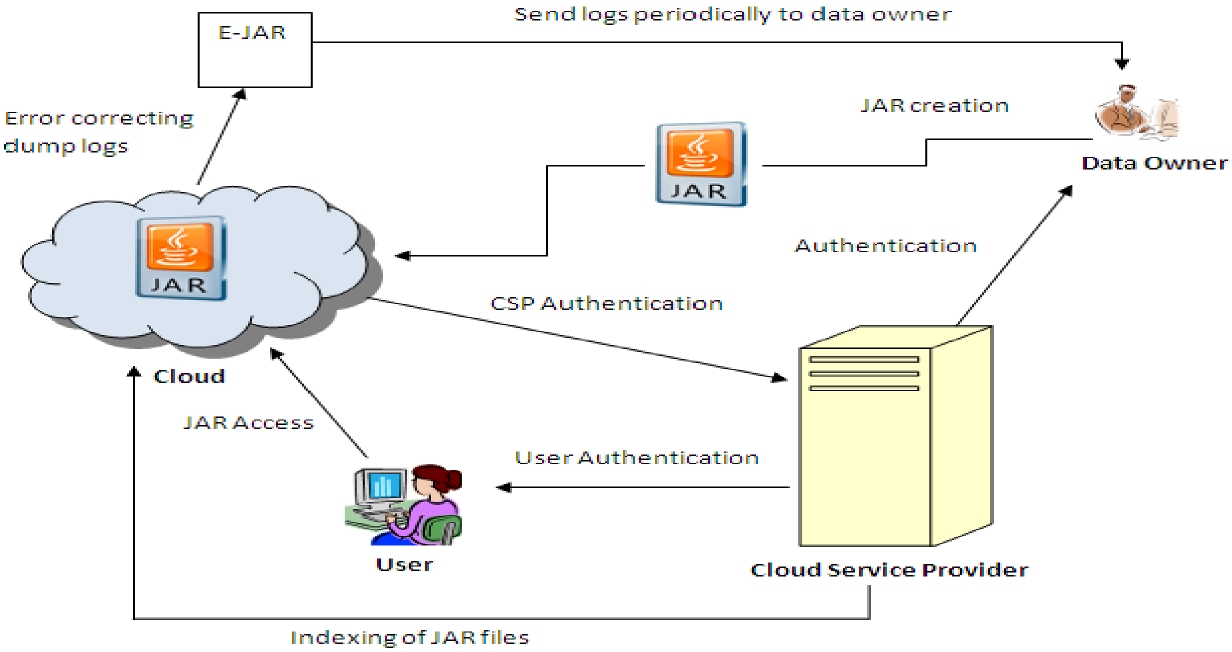

Fig.3. Working of CIA framework

D. System Flow

The overall CIA framework, combining data, users, logger and harmonizer is sketched in Fig. 3

At first when data owner will want to upload data he will first generate public and private keys using IBE Weil pairing scheme described by author [9]. The data is going to be uploaded on cloud in JAR format which first authenticates CSP and then users who are accessing it. This authentication is possible because Jar file includes a set of simple access control rules specifying whether and how the cloud server and possibly other data stakeholders (users, companies) are authorized to access the content itself.

With each access to data a log record containing details of user's location, access time, id etc will be automatically recorded and then is encrypted using public key. This record will be appended to log file present in JAR file. As many users will be viewing or downloading the data so many JAR copies will be created. Now here the role of log harmonizer comes to visible as its job is to hash together individual records to create a chain structure, able to quickly detect possible errors or missing records. The encrypted log files can later be decrypted as log harmoniser holds the private-key and integrity of log files can be verified. In addition to this, any error correction details will be sent to the log harmonizer to control possible log file corruption. This file is then periodically sent to data owner or can be pulled by data owner whenever he wants to see it for auditing purposes.

IV. SECURITY DISCUSSION

We now analyze possible attacks to our framework. We assume that attackers may have sufficient Java programming skills to disassemble a JAR file and prior knowledge of our CIA architecture. We first assume that the JVM is not corrupted, followed by a discussion on how to ensure that this assumption holds true.

A. 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.

B. 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.

V. Conclusion

It is more and more important to defend and preserve people's privacy on the Internet, against unwanted and unauthorized disclosure of their confidential data. Despite laws, legislations and technical attempts to solve this problem, at the moment there are no solutions to address. Throughout this paper, the authors have systematically studied and review the security and privacy issues in cloud computing. This paper presents effective mechanism, which performs automatic authentication of users and create log records of each data access by the user. Data owner can audit his content on cloud, and he can get the confirmation that his data is safe on the cloud. Data owner also able to know the duplication of data made without his knowledge. Data owner should not worry about his data on cloud using this mechanism and data usage is transparent, using this mechanism.

ACKNOWLEDGMENT

The authors would like to thanks the Department of Computer Engineering, ZES's Dnyanganga College of Engineering & Research, Pune, Indiafor the guidance and cooperation.

References

[1] SmithaSundareswaran, Anna C. Squicciarini and Dan Lin, "Ensuring Distributed Accountability for Data Sharing in the Cloud,", IEEE Transaction on dependable a secure computing, VOL. 9, NO. 4, pg 556-568, 2012.

[2] B.Crispo and G.Ruffo, "Reasoning about Accountability within Delegation" Proc. Third Int'l Conf. Information and Comm. Security (ICICS), pp. 251-260, 2001.

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

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

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

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

[8] 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.

[9] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf.

[10] HP Cloud Website

[11] Advances in Cryptology,pp. 213-229, 2001.B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source
Code in C. John Wiley & Sons, 1993.

[12] http://en.wikipedia.org/wiki/Identity-based_encryption