International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 3 Issue: 5
ISSN: 2321-8169
3413 - 3416

Provision of Data Accountability and Security in cloud

Shital A. Hande
Department of Computer Engg.
College Of Engineering Pune.
handesa13.comp@coep.ac.in

Prof. Sunil B. Mane
Department of Computer Engg.
College Of Engineering Pune.
Sunilbmane@gmail.com

Abstract:- Cloud computing is best emerging paradigm in computer industry. This technology hides the details of services from user. Moreover, users may not know the machines which actually process and host their data and also that data is outsourced to other entities which cause issues related to accountability. So there is need of approach which allows users to keep track of their data in the cloud. To solve security related issues we propose a Cloud Data Security and Accountability (CDMA) framework which is based on Information Accountability. This framework allows user to keep track of data in the cloud. It is an extension of existing Cloud Information Accountability (CIA) with improved performance. Its main feature is lightweight Accountability with high security.

Keywords: *- Cloud Computing, Logging, Auditing Data Accountability, Data Security.*

*****

1. Introduction

Cloud computing is computation in which many groups of servers are connected together to allow the various services, and online access to computer services or resources. The main reason for cloud popularity is its virtual nature, flexibility cost and speed. By using this technology, user has access to the resources they required for their particular task. . In this technology service details are hided from the data user. Moreover, user is unaware of the machine which process and hosts their data. there are also some problems while enjoying this new technology that includes losing control over users own data, processed data on cloud also outsourced which causes different issues related to Accountability including handling personal identification information because of which user start worrying. To deal with user concerns, there is need of effective mechanism that monitors usage of user's data in cloud. For example, when user log on to the services in the cloud they need to ensure that their data is operated as per service level agreement. There are different conventional access control approaches that are not suitable like approaches for centralized server and closed domain such as operating system, because of some features of cloud system .First, Cloud Service Provider (CSP) can outsource user's data toother entities in the cloud and these entities to other, and so on. Second Flexibility to entities, they can enter and leave the cloud as they want. So because of this, processing data in the cloud becomes complex. To solve these problems, we propose a framework namely Cloud Data security and Accountability (CDMA) which is based on Information Accountability. It is an extension of CIA framework where we provide efficient data accountability with high security. This framework provides end to end accountability in distributed fashion. Its best feature is ability of maintaining powerful and lightweight accountability that combines different aspects of access control, usage control and authentication.

2. Literature survey

Cloud is a computing model providing web-based software, middleware and computing resources on demand. Cloud computing changes the way we think about technology. By deploying technology as a service, user accesses only to the resources they need for a particular task. This prevents them from paying for idle computing resources. Cloud computing can also go beyond cost savings by allowing users to access the latest software and infrastructure offerings to foster business innovation. In literature we are going to focus mostly on security issues of the cloud. We are going to discuss security in cloud and data accountability in the following sections.

2.1 Security in cloud

Today there are many techniques available that handles security issues in cloud. Review of some of the techniques is as follows: The main building block between cloud service providers and data owners is trust. But unfortunately it is not completely done by providers. **Jensen and et al.** [1] has given a approach of data anonymization.

This mechanism improves trust between both the parties. Its main idea is cut off link between user and its data and provides usage to the cloud based on group signature and ring signature. But this approach is limited to some fixed number of users and also there is need of data security for detecting activities of data and data leakage across the cloud Suen[9] has given end to end data centric mechanism. That mechanism is S2Logger which allows cloud shake holders to trace the data across the cloud. S2Logger performs analysis of kernel space data event at file as well as block level. It logs the data activities. By using that information detection of security related threats, violation of data policy and data leakage is possible.

G. Lenzini and et al. [5] proposes the mechanism in which it is possible for agent to provide data with policies. If agents want to access data then agents has to prove their authentication and action. The actions user can do with the data are specified in the policies attached with data. This requires continuous auditing of agent.

But this system provided solution which monitors wrong behavior of agent and for that agent has to give justification. After that this justification is checked by authority. *S. Pearson* was given privacy mechanism in which data is in encrypted form in cloud. Further evaluations are carried on encrypted data. Data on cloud is safe because obfuscation data is not on service provider machine. But this is not a proper solution for larger input data. This method requires large amount of memory for larger size data [3].

S. Pearson and et al. [4] presents a mechanism which improves performance. This is a mechanism in which parties decides policies that store, use and share the data. They do not consider the jurisdiction in which data is processed. But at the time of processing processed data is in unencrypted form. So there is possibility of data leakage. This becomes disadvantage of provided mechanism.

Wang and et al. [7] proposed a method of Dynamic auditing protocol for auditing. On cloud server this method allows to perform dynamic operations of the data. Here also there is possibility of data leakage to the auditor. Because the linear combinations of data blocks to the auditor are sent by the server.

Dynamic auditing scheme extended to improve privacy. For multiple owners this scheme becomes privacy preserving and also supports batch auditing. This auditing protocol requires large number of data tags. So on the cloud this scheme may incur a heavy storage overhead [8].

Sundareswaran and et al. [6] proposed a three layer architecture which protects data leakage. In first layer, view to confidential data is not allowed to the service providers. In second layer indexing of data is prohibited. In last layer, user describes use of his data along with policies.

First time **Dan Lin and et al.** [2] proposes the mechanism of automatic logging in the cloud was proposed. This mechanism focuses on data accountability by using concept of JAR file. It is highly decentralized and platform independent. But in this method there is possibility of data changing attack while data is travelled on network. Also Because of multiple jar files it requires more execution time and more latency.

3. CDSA Framework

In existing CIA model, during data transmission on network there is possibility of data changing attack. So to overcome this we are presenting here our Proposed Cloud Data Security and Accountability Framework (CDSA Framework). With this framework data owner tracks service level agreement and also enforce usage and access control rules. For auditing we develop two distinct modes: pull mode and push mode, push mode for periodically sending logs to data owner of the data and pull mode allows data owner to retrieve the logs as needed. This Framework provides JAR verification module which protects the Data Changing attack while traveling on the network. If file altered on the network the data user will be prompted after verification so that the user will not access that JAR. There is very less overhead of code on the data. Light weight module is developed to handle JAR data.

3.1 Main Components

It consists of two main components: Logger and Log Server (Log Harmonizer):

Logger

It is strongly coupled with uploaded user's data. When anyone accessed user's data then logger get downloaded. Its main purpose is automatically logging. When data is accessed then log record gets created by logger and encrypts it using public key of data owner. After that it sends them to the Log server. Logger requires less support from server. There is no need to install logger on any system. It results in highly distributed system because of strong coupling between data and logger. The structure shown in Figure 1 works as, when user accesses Jar file it will ask for the authentication credentials. After providing correct credentials user will get chance to verify authentication of data owner and correctness of file. If verification is successful then only user will get chance to access the JAR file otherwise not. At the same time Logger becomes active and it will generate Log record and transfer it to the Log server. Now Jar file consist of encrypted data file and user has to decrypt it using private key for accessing the data.

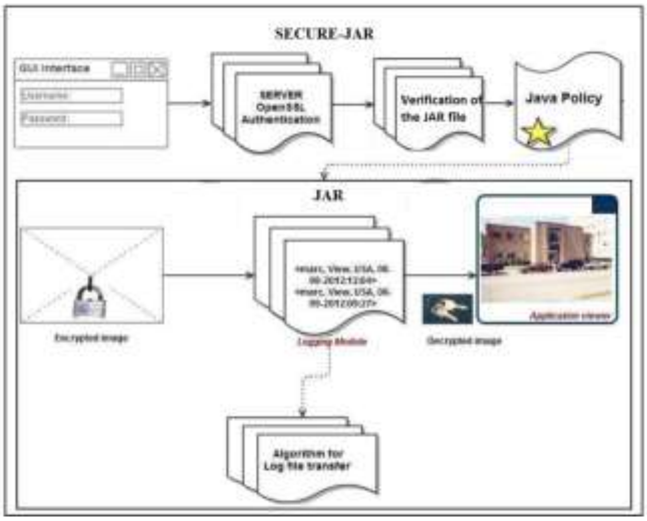

Figure 1.Secure JAR File Structure

Log Server (Log Harmonizer)

Its main purpose is auditing. Log server receives logging information from Logger. User can access log files from Log server. It holds decryption key and using that it decrypts log record which is received from logger. Alternatively It is possible to carry out decryption at client and. It provides two strategies for auditing pull and push. Push mode periodically transfers log file to the data owner. Alternatively pull mode transfers log file on demand.

3.2 Data Flow

In the figure 2, we have given the overview of Proposed Cloud Data Security and Accountability Framework. In Step 1 Data owner need to login to the portal. In Step 2 Data owner need to Generate Key pair which will be used for the file encryption. Step 3 Data owner Encrypt all the files that are to be placed on the cloud. In Step 4 user Uploads Encrypted files.

In Step 5 Data owner will create the JAR file by using the public key generated in the step 2. All the files uploaded in the in step 4 will be enclosed in the JAR file. In step 6 Data owner need to Sign the created JAR file so that signature of the all data files is enclosed in the JAR file itself. Data owner now can upload the Signed JAR file on the Cloud. User will share the generated private key, username and password with the authorized or intended user. We have not considered key exchange mechanism here. It is assumed that the keys and other credentials are with the authenticate data users

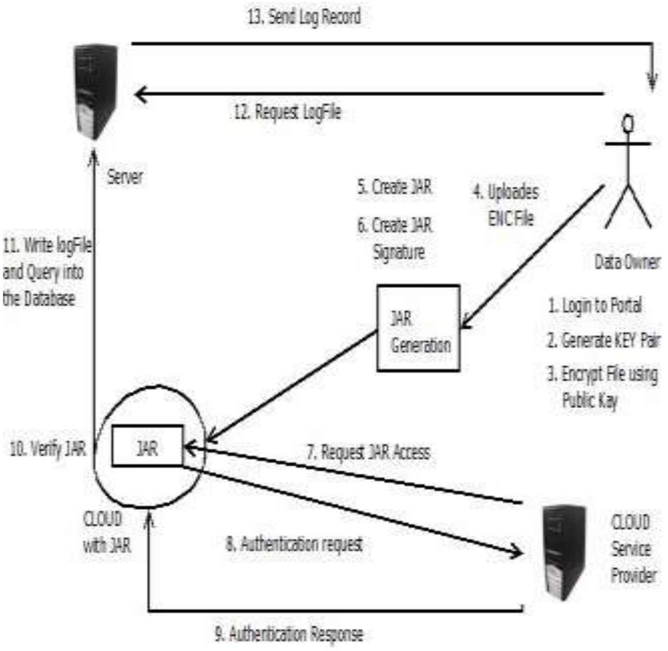

Figure 2.CDSA Framework

Step 7 if somebody want to access the files of particular data owner firstly he obtains the credentials from him. On cloud he can request JAR file access. Step 8 JAR file will ask for authentication credential. In Step 9 the Data User will provides the correct credential to access that JAR file. If he is not authorized person to access that JAR he will not be permitted to access that JAR. Step 10 data user verifies the JAR file if it is not tampered during sending on the network its signature verifies and prompt Jar verified message. If it got damaged on the network the jar verification module prompts the verification failed message. So that user will not access that file.

Step 11 if the data user verification is successful then the log file for that particular JAR file is created on the server. At the same time the logs information is also inserted to the SQL database. Step 12 Data owner can request for the log file. Step 12 The Server will send log information to that particular user.

4. Graphical Representation of Experimental results

Proposed CDSA framework is light weight model to handle JAR file. For testing purpose, we have considered some image files whose size varies between 55 KB to 1500 KB. For these files Jar File size ranges from 100 KB to 1900 KB. Increasing size of file will increase Encryption time as well as JAR creation time. The results are shown in tabular (Table 1) and graphical format (figure. 3 and 4).

Table 1: File size & Time

Test Cases	Size (KB)	Encrypted file Size (KB)	JAR file size (KB)	Time to Encrypt (ms)	Jar creation time (ms)
Case 1	55	95	108	265	4.8
Case 2	132	229	208	546	4.95
Case 3	320	559	458	1295	5.47
Case 4	452	790	634	1888	4.7
Case 5	652	1112	900	2746	5.05
Case 6	975	1661	1291	4009	4.98
Case 7	1478	2461	1901	5928	5.51

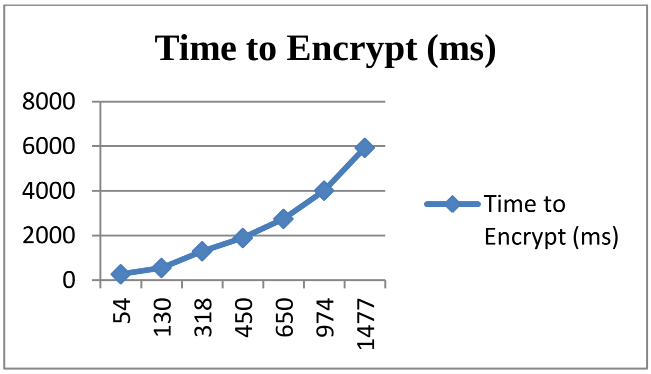

Figure 3: File size (x axis) & Encrypt time(y axis)

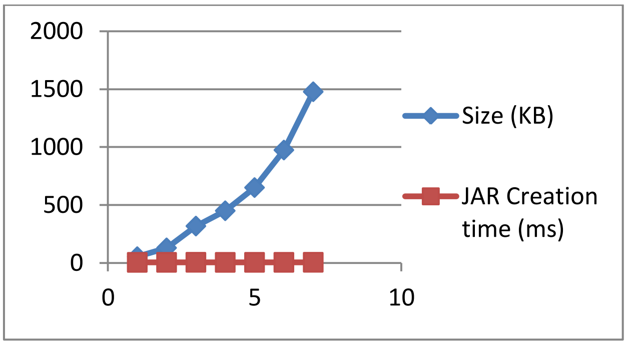

Figure 4: JAR File size(KB) and JAR Creation Time

5. Conclusion

We proposed innovative approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. Our approach allows the data owner to not only audit his content but also enforce strong back-end protection if needed. Moreover, one of the main features of our work is that it enables the data owner to audit even those copies of its data that were made without this knowledge. Our Cloud Data Security and Accountability (CDSA) Framework provides JAR verification module which protects the Data Changing attack while travelling on the network. If file altered on the network the data user will be prompted after verification so that the user will not access that JAR. There is very less overhead of code on the data. Light weight module is developed to handle JAR data. The file encryption time increases as the size of the data file increases. The Jar creation time is nearly equal for the files we have tested. In the future, work can be extended to secure key and user credentials exchange algorithm between data owner and data user. This can be achieved through trusted third party key exchange store.

References

[1] Meiko Jensen, Sven Sch¨age, J¨org Schwenk, “Towards an Anonymous Access Control and Accountability Scheme for Cloud Computing”, *IEEE 3rd International conference,2013.*

[2] Smitha Sundareswaran, Anna C. Squicciarini and Dan Lin, (July/August 2012) "Ensuring Distributed Accountability for Data Sharing in the Cloud,", *IEEE Transaction on dependable a secure computing*, VOL. 9, NO. 4

[3] S. Pearson, Y. Shen, and M. Mowbray," A privacy Manager for Cloud Computing,*" Proc. Int'l Conf. Cloud Computing* , 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 in cloud 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] sChun Hui Suen, Ryan K L Ko, Yu Shyang Tan, Peter Jagadpramana, Bu Sung Lee “S2Logger: End-to-End Data Tracking Mechanism for Cloud Data Provenance”, *IEEE Conference,*12th 2013.

[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.