

Accountability of Data usage in Cloud Computing

Nitin P. Doiphode¹, Hemlata Channe²

^{1,2} Computer Engineering Department, Pune Institute of Computer Technology, PICT, Pune, India

Abstract: *Cloud computing is a distributed model which provides shared pool of resources and services to the customers. Cloud computing can easily provide services through network and these are on demand services to the users. User stores different kinds of data on the cloud which can be sensitive or private and should not be disclose to unauthorized users. That is why the data owner needs to know that who is accessing his data without his required permission and whether his data is handled according to the service level agreement which is made at the time when user registers for the services in cloud. For that data owner should know the current usage of his data in cloud through accountability. In this paper, we are focusing on accountability of outsourced data in cloud. Accountability is verification of authentication and authorization policies. It consists of automated logging and auditing of the data usage. In automated logging, any access to the owner's data will trigger authentication and log is generated automatically. These logs generated are sent to the data owner so that he can audit his data usage by analyzing the log records. After analyzing the log records, if data owner finds any unauthorized access to the outsourced data then he will convey this to the cloud service provider in form of audit file. Cloud service provider will read the audit file and take the decisions according to the type of access made to the outsourced data.*

Keywords: Cloud computing, accountability, logs, authentication, audit, security.

1. Introduction

Cloud computing is a model which provides shared pool of resources and services to the customers. Cloud computing can easily provide services through network and these are on demand services to the users. Many users from large organizations to individuals take services from Cloud. Now days, there are commercial and individual cloud computing services are available, which are Amazon, Google, Microsoft, Yahoo, Salesforce, etc.

User stores different kinds of data on the cloud which can be sensitive or private and should not be disclose to unauthorized users. For example, personal healths care data, financial data, etc. This outsourced data processed on cloud and data owner didn't know that on which machine in cloud his data resides and who are accessing his data without prior his knowledge. That's why the data owner should know that who are using or accessing his data without his permission and whether his data is handled according to the service level agreement which is made at the time, when user registers for the services in cloud he sign on for services in cloud [1]. For that the data owner should know the current usage of his data.

To increase the trust between the cloud and its users, all actions performed on user's data should be made transparent to the data owner. This transparency can be achieved by using accountability mechanism. Accountability can help cloud users to monitor their outsourced data and also the data owner can able to check whether their data is accessed by authorized or unauthorized user. Accountability can ensure the integrity of the outsourced data in the cloud.

In this paper, we are focusing on accountability of outsourced data usage. Accountability is verification of authentication and authorization policies. It consists of automated logging and auditing of the data usage. In automated logging, any access to the owner's data will trigger authentication and log is generated automatically. These generated logs are sent to the data owner so that he can audit his data usage by analyzing the log records. This approach can help to identify unauthorized access to the outsourced data. The components

which perform logging and merging of actions on data are logger and log harmonizer. These components are JAR files and it makes usage of JAR capabilities.

The rest of the paper is organized as follows: Section 2 describes literature survey. Section 3 specifies the problem statement. Section 4 represents proposed idea of accountability of data usage in cloud. Section 5 discusses the related mathematical work and Section 6 discusses about the security issues and solutions. Finally, Section 7 gives idea about experimental setup and section 8 shows results achieved, followed by the conclusion of the paper.

2. Related Work

In this section we review some cloud data storage privacy and security related issues and problems. Then, we discussed about the accountability techniques adopted in cloud. Some discussion is also based on the privacy and security issues in cloud.

Cloud computing has many issues in security and privacy of outsourced data storage. Zhifeng Xiao and Yang Xiao identified these issues in [14], with the help of some attributes like availability of services, integrity of data, confidentiality, and accountability of cloud. Ramgovind S. et al discussed security considerations and current challenges to the cloud computing [15].

In [16], Daniele Catteddu et al presented a model of accountability in which they have defined the accountability as data governance, differentiating among accountability attributes, practices, mechanisms and tools. S. Pearson in [11] described accountability as core concept and how it helps to increase the trust in cloud. In [10], Ryan K L Ko et al presented trust related scenario to achieve accountability in cloud. In [8] and [9], discusses privacy and accountability in cloud in more detail. In paper [19], authors have done literature survey on accountability of data usage.

Smitha Sundareswaran et al presented a new approach for accountability for data sharing in cloud [1], in which they

have used JAR functionalities to create the log records and to evaluate the records. In [2], Cong Wang et al presented an idea for auditing of cloud storage, in which they have introduced one third party auditor (TPA) to check integrity of outsourced data in cloud. In [12], Boyang Wang et al proposed privacy preserving mechanism to audit the shared data storage on cloud. Yan Zhu also proposed audit service to verify the integrity of outsourced data in cloud [17]. Kan Yang proposed an auditing protocol for cloud data storage [18]. Ming Li et al presented a new patient centric framework; this uses attribute-based encryption (ABE) technique to encrypt each patient's health record files [13]. S. Pearson et al represent technical solution to the accountability to solve security risks in cloud and in this mechanism the policies are decided by the users that access or share the data [3]. To make outsourced data more secure on cloud, S. Pearson et al developed privacy manager software which helps the user to secure his data on cloud [4]. This data is in encrypted form on cloud and this privacy manager makes readable data from encrypted data. Ryan K L Ko et al presents TrustCloud framework addresses accountability and discusses key challenges in achieving a trusted cloud by using detective controls [5]. R. Corin et al proposed procedural and technical solutions to demonstrate accountability to minimize privacy and security risks in cloud [6].

3. Problem Statement

In this section, we are specifying problem statement by considering an example from which we can get the objectives for our proposed system.

We consider a health care management system, because now days most of the hospitals and clinics make use of cloud to share the health information among the patients, health care staff and insurance company. In this system, doctors need to manage patient's health records carefully so that no one can make misuse of it, hence this approach can be the hospital-centric. In this, doctors or any administrative faculty of hospital be the data owner of patient's health records. So, hospital wants to do audit of patients health records usage. For this the owner of these health records needs the following requirements:

- Health records can be downloaded by only authorized health care staff members and insurance company.
- For critical diseases, hospital faculties needs to set authorization policies for health record users.
- If the disease is nominal then other people can also allowed viewing the health records.
- According to the designation of the data user in hospitals and insurance company, patient can allow to view or download the health record.
- Hospital management system wants to ensure that the cloud service provider is not sharing patient's health records with anyone else.

With the help of this above requirements we identify some needs and we reach to following objectives of accountability in cloud.

- Creating the log records of all the actions performed on outsourced data in cloud.

- To help the data owner to monitor his data by analyzing the log records.
- To identify the unauthorized access to the owner's data.
- Ensuring the confidentiality and integrity of log records created.

4. Proposed Work

In this section, we are focusing on accountability of data usage in cloud and description of how we are going to record the actions performed on the data in cloud. Accountability framework in this architecture consists of automated logging and auditing of user's accesses to the owner's data in the cloud. Logging has two important components which are logger and log harmonizer. In Fig. 1, working of accountability mechanism in cloud is shown. When user will access the data, log of each access is created by logger and it will send these generated log record to log harmonizer. Log harmonizer will merge all these log records and checks for error correction and recovery of logs. Then log harmonizer will send the log file to the data owner periodically or after on demand of the data owner. Data owner receive the log file and analyze the log records in it. If data owner finds any unauthorized access to the outsourced data then he will convey this to the cloud service provider through audit file. So cloud service provider can take the appropriate action according to the type of access made to the outsourced data. Logger component is a JAR file performs automated logging and verification of authentication and authorization of users and cloud service provider. This JAR contains the data file and the access policies for authentication and authorization of the CSP and data users. Logger generates log records of each action performed on the outsourced data and encrypts these log records to provide confidentiality. The created log records are sent to the log harmonizer periodically or immediately after log creation by the logger.

Log harmonizer is also a JAR file which collects log records from the logger periodically, so that harmonizer can know that logger is active. Log harmonizer is also responsible for error correction of log files and recovery. Log harmonizer supports two modes to send log files to the data owner which are push and pull mode. In push mode, harmonizer periodically sends log files to the data owner in automated fashion. In pull mode, data owner requests for the log file to the harmonizer and log harmonizer send log file to the owner as response.

Following are the steps of accountability-

1. Creation of JAR file by setting authentication and authorization policies with data.
2. Actions performed on the data by cloud users like view or download data files.
3. Logging of each action performed on data by logger.
4. Logger send log records periodically to the log harmonizer.
5. Log harmonizer receives log record from logger and merges all these log records.
6. Error correction in logs is performed by log harmonizer and generated log file is sent to data owner.
7. Data owner receives log files and performs analysis of log files.

8. After analysis completion, if data owner found unauthorized access to the data then he will create audit file of it.
9. Data owner send generated audit file to the cloud service provider.
10. Cloud service provider will take appropriate actions according to the access type to the outsourced data.

Initially data owner will set the access policies with data and then he will create the JAR file and submits it to the cloud service provider. When this data is accessed by the users then logger will create one log record and this action will be recorded in this log record. Then after recording each and every action, logger will generate error correction information for each record and send them to the log harmonizer. Log harmonizer merges all these records in one log file and sends this file to the data owner. Data owner will analyze the log records and perform audit on it. If data owner finds any unauthorized access to the data then he will create audit file to store this information and send this audit file to the cloud service provider. After receiving the audit file, cloud service provider will read it and take the appropriate actions according to the access type made by the users.

Following system block diagram shows the different entities of accountability mechanism in cloud and interactions among them.

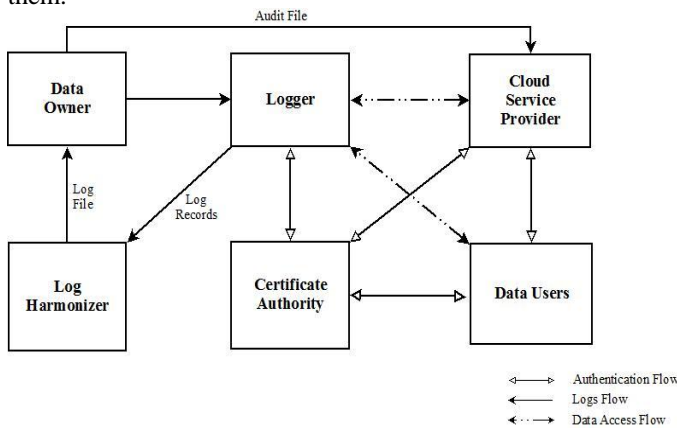


Figure 1: System Flow Diagram

5. Mathematical Model

$S = \{ s, e, X, Y, D, J, L, A, O, C, U, f \mid \phi_s \}$

Let, S is the system.

- s - Start state of the system
- e - End state of the system
- X - Set of inputs to the system i.e. JAR file (J) consisting data and access control policies.
 - $X = \{ J \}$
- Y - Output of the system i.e. log records (L) created.
 - $Y = \{ L, A \}$
- D - Set of data files to outsource to cloud.
 - $D = \{ D_a \in D \mid D_a \text{ is the data file to outsource.} \}$
- J - Set of JAR files created for every data file which is to outsource.
 - $J = \{ J_l, J_h \}$

Where,

- J_l - Logger which performs logging of data.
- J_h - Log harmonizer used to perform audit.

- L - Set of log records i.e. log file in system.

- $L = \{ L_1, L_2, L_3, \dots, L_m \}$
Where, m is the no. of log records.
- A - Audit file generated after analysis of log file
- O - Owner of the outsourced data.
- C - Cloud service provider.
- U - Set of users which accesses outsourced data.
 - $U = \{ U_1, U_2, U_3, \dots, U_r \}$
Where, r is the no. of users.
- f - Set of functions.
 - $f = \{ f_c, f_u, f_l, f_e, f_h, f_a \}$
- f_c - Function to create the JAR file by setting the policies with data and submit this created JAR file to the cloud provider.
 - $f_c: J(D) \rightarrow C$
- f_u - Function to represent use or access of data by users.
 - $f_u: U_r \rightarrow J(D)$
- f_l - Function to create log records of each access to the outsourced data.
 - $f_l: J_l \rightarrow L_m$
- f_e - Logger will perform error correctness in log records and submits log records to the log harmonizer.
 - $f_e: J_l(D) \rightarrow J_h$
- f_h - Log harmonizer merge all log records from logger and send it to the data owner.
 - $f_h: J_h \rightarrow (L_1 \cup L_2 \cup L_3 \cup \dots \cup L_m)$
- f_a - Data owner performs auditing of data by analyzing the log records.
 - $f_a: (O \rightarrow f_m(L)) \rightarrow A$
- ϕ_s - Constraints on system:
 1. If JAR file creation on JVM is failed.
 - f_{sc} - function represents failure of JAR file creation.
 2. If authentication of CSP and data user failed.
 - f_{sa} - function represents failure of authentication of CSP and data users
 3. If logger is not able to create log records.
 - f_{sl} - function represents failure of logger
 4. If log harmonizer is not able to merging of log records and submitting log files to the data owner and CSP.
 - f_{sh} - function represents failure of log harmonizer.
 5. Network failure i.e. physical connection problems.
 - f_{nw} - function represents network failure in system.
- Success - Log records are successfully created and unauthorized accesses to the data are avoided by the logger.
 - $(f_c \cap f_u \cap f_l \cap f_e \cap f_h \cap f_a) \rightarrow L_m$
- Failure - Logs are not created and data owner is not able to audit the outsourced data.
 - $(f_l \cup f_h \cup f_a) \neq L_m$

6. Security Discussion

In this section, we are discussing some security attacks and solutions for them. We need certificate authority to certify authentication requests for CSP and data users to access the data in cloud. The JAR file which contains the data and the

log files in it, so the attacker can try to learn the information from these log files. Attackers may have that much of knowledge of the structure of JAR files and our accountability framework. We are considering here that the JVM platform that we are using is not corrupted, and on this assumption we are trying to avoid the different attacks in our system.

6.1 Coping Attack

The most dangerous attack is that attacker may copy the entire JAR file and try to get extra information from it [7]. We can able to detect this kind of attacks because every JAR file needs to send redundancy information periodically to the log harmonizer. So this approach can also detect the files which are created without knowledge of the data owner.

6.2 Account or Service Hijacking

Always reused credentials and passwords may results impact of such attacks. If an attacker gets access to the credentials of the user, then he can eavesdrops on user transactions and operations, also he can manipulate the log records. This can be avoided by powerful authentication and proactive monitoring to detect unauthorized access. In our proposed idea, we can use OpenSSL-based certificates to authenticate cloud service provider and SAML-based authentication to authenticate cloud users.

6.3 Data loss or leakage

Attack on JAR files may results to data loss or leakage because attacker may try to learn information from the log files. Loss of encoding key may affects integrity and confidentiality of data files. This can be detected by strong API access control policies and strong key generations for CSP and data users.

6.4 Man-in-the-Middle Attack

An attacker may try to manipulate the messages among the certificate authority, data owner and CSP while authentication. Attacker may try to establish a fake connection among these entities, but it can be avoided by proper session handling.

6.5 Attacks on JVM

Attacker may try to do settlement with the JVM. We can use hashing techniques like oblivious hashing to check the integrity of the JVM, so we can check the correctness of the JRE on which we are executing our JAR files.

7. Experimental Setup

In this section, we introduce the settings for the test environment in which the required configuration is specified. The test environment may consist of minimum number of 2 nodes in which one will be the cloud storage and another will be the client machine. On these nodes we are configuring OpenStack cloud which is open source cloud and can be easily available [19]. We will use Linux-based machines for

the implementation and each node will have minimum configuration as Intel Core i3 processor, 4GB RAM, 500GB Hard Drive and network connectivity. Each node should have the JVM environment.

8. Results

Results can be shown with the help of parameters like log creation time, authentication time, logging time, log merging time and size of the JAR files created. Log creation time and merging time are less than the time required for logging. Authentication time is considered in logging time only. Size of the JAR file created can be less than the actual data file, due to compression facility provided by the JAR files. Following graph shows the original data file size versus created JAR file size.

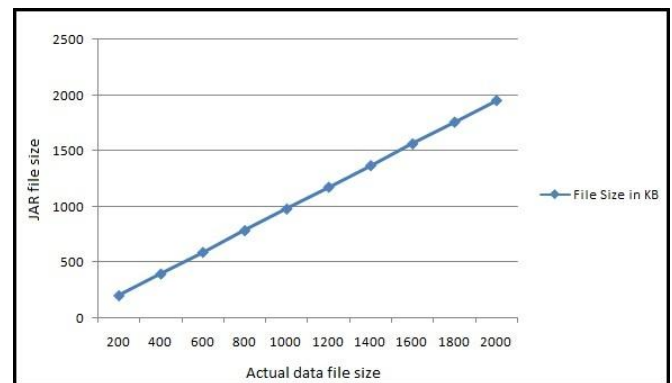


Figure 2: Size of the Logger component

Log file consist of number of log records from loggers. Log records are the tuples of 6 elements which are collected by the logger and written to single log record. Log record (L) shown as,

$$L = \langle \text{CSP, User, Act, Perm, F, T} \rangle$$

1. CSP - Cloud service provider
2. User - Identity of user who accessed data
3. Act - Action performed on data
4. Perm - Permission granted or denied
5. F - File on which access attempted
6. T - Time and date of action performed

< csp, nitin, View, Granted, file.txt, 2015-03-11 16:52:30 >

User *nitin* is granted to view a file *file.txt* under *csp* cloud service provider at *2015-07-25 16:52:30*.

9. Conclusion

In this paper, we proposed an approach for accountability of data usage in cloud. It will help data owner to audit his outsourced data by creating the log records of all actions performed on his data. We can provide confidentiality and integrity to the generated logs. By analyzing the audit files, cloud service provider can restrict or control the unauthorized access to the outsourced data. Finally, this approach can help data owner to trust more on cloud service providers as it can increase the transparency among data owner and his data under the control of cloud service provider.

In the future, we will try to provide more security to the JVM environment.

References

- [1] Smitha Sundareswaran, 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] Cong Wang, S. M. Chow, Qian Wang, KuiRen, Wenjing Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Transaction on Coputers, 2013.
- [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] S. Pearson, Y. Shen, and M. Mowbray, "A Privacy Manager for Cloud Computing," Proc. Int'l Conf. Cloud Computing (CloudCom), pp. 90-106, 2009.
- [5] Ryan K L Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui Liang, Bu Sung Lee, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing," 2nd IEEE Cloud Forum for Practitioners, 2011.
- [6] 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.
- [7] S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang, "Promoting Distributed Accountability in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing, 2011.
- [8] Marco Casassa Mont, Siani Pearson and Pete Bramhall, "Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services," Proc. Int'l Workshop Database and ExpertSystems Applications (DEXA), pp. 377-382, 2013.
- [9] Marianthi Theoharidou, Nick Papanikolaou, Siani Pearson and DimitrisGritzalis, "Privacy Risk, Security, Accountability in Cloud Platforms," IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 177 – 184, 2013.
- [10] Ryan K. L. Ko, Bu Sung Lee, Siani Pearson, "Towards Achieving Accountability, Auditability and Trust in Cloud Computing," Springer Advances in Computing and Communications Communications in Computer and Information Science, Volume 193, pp 432-444, 2011.
- [11] Siani Pearson, "Toward Accountability in the Cloud," View from the Cloud, IEEE Internet Computing, IEEE Computer Society, July/August issue, vol. 15, no. 4, pp. 64-69, 2011.
- [12] Boyang Wang, Baochun Li, Hui Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," IEEE 5th International Conference on Cloud Computing, 2014.
- [13] Ming Li, Shucheng Yu, Yao Zheng, KuiRen, Wenjing Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," IEEE Transactions on Parallel And Distributed Systems, 2013.
- [14] Zhifeng Xiao and Yang Xiao, "Security and Privacy in Cloud Computing," IEEE Communications Surveys & Tutorials, Vol. 15, No. 2, 2013.
- [15] Ramgovind S, Eloff MM and Smith E, "The Management of Security in Cloud Computing," IEEE Information Security for South Africa (ISSA), 2010.
- [16] Daniele Catteddu et al. "Towards a model of accountability for cloud computing services," Proceedings of the DIMACS/BIC/A4Cloud/CSA International Workshop on Trustworthiness, Accountability and Forensics in the Cloud (TAFC), 2013.
- [17] Yan Zhu, Gail-Joon Ahn, Hongxin Hu and Yau, S.S., "Dynamic Audit Services for Outsourced Storages in Clouds," Services Computing, IEEE Transactions on Volume 6 , Issue 2, 2013.
- [18] Yang, Kan and Xiaohua Jia. "An efficient and secure dynamic auditing protocol for data storage in cloud computing," Parallel and Distributed Systems, IEEE Transactions on Volume 24, Issue 9, 2013.
- [19] Nitin P. Doiphode and Prof. H. P. Channe, "A Survey on Accountability of Data usage in Cloud Computing," International Journal of Science, Engineering and Technology Research (IJSETR), Vol. 4, no. 3, pp. 514-516, March 2015
- [20] OpenStack systems, <http://www.openstack.org/>. [Accessed: July 15, 2015].

Author Profile

Nitin P. Doiphode received B.E. degree in Computer Science Engineering from K.I.T.'s College of Engineering, Kolhapur and currently persuing his M.E. degree in Computer Engineering from Pune Institute of Computer Technology, Pune. His main is in distributed systems.

Prof. H. P. Channe is an Assistant Professor in Computer Engineering Department at Pune Institute of Computer Technology, Pune. Her interest is in networks and security.