Provenance Control Based Secure Cloud System Using Jar Accessibility

N. JYOSTHNA[1], P. LEENAPRIYA[2], D. BULLA RAO[3], P. NAGESWARA RAO[4]
DEPARTMENT OF CSE, SWETHA INSTITUTE OF TECHNOLOGY AND SCIENCE::TIRUPATHI
jyosthna21@gmail.com, leenapriya.p@gmail.com, bullaraodomathoti@gmail.com,
puttanr@reddiffmail.com

***Abstract*— Cloud computing is a set of services to be easily consumed over the Internet on an as-needed basis for user's. Cloud computing provides the highly scalable services, that enables to an end user over on a leased basis over a network. Though cloud computing provides many services to the user's, it as issues on the difficulty of how to provide proper security and privacy protection. The proper security and privacy protection in the cloud computing is very important but these problems are yet not solved. One of the important feature of the cloud computing services is to provides that user's data are processed remotely by the unknown machines so that the user do not own or operate their own data's. While enjoying the service provides by the cloud computing, the user fears of losing control of their data, it becomes the significant barrier in the services of the cloud computing. To overcome the above problems in the service provides by the cloud computing, we introduce new technique. In this paper, we propose a highly designed the cloud information accountability framework(CIA) for the data sharing in which procedural and technical solutions for the above problems and to keep track of the actual usage of the user's data in the cloud. In particularly, we approach that enables enclosing our logging mechanism together with users' data and policies. We leverage the JAR (Java ARchives) programmable capabilities to both create a dynamic and travelling object, and to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. We also propose the distributed auditing mechanisms for user's control to become the strong. We have prototyped the framework and deployed it on commercial cloud environment for experimental runs to test the efficiency and effectiveness of our approach and evaluate the performance of the implemented prototype.**

Keywords*— *Cloud computing, accountability, data sharing, Security, Privacy, user's control and JAR.

I. INTRODUCTION

Cloud computing is an emerging paradigm in the computer industry where the computing is moved to a cloud of computers. The cloud computing core concept is, simply, that the vast computing resources that we need will reside Somewhere out there in the cloud of computers and we'll connect to them and use them as and when needed. Cloud computing is the next general step in 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. To a large extent, cloud computing will be based on virtualized resources. The convenience and efficiency of this approach, however, comes with security risks and data privacy. A significant barrier to the adoption of cloud services is thus user fear of confidential data leakage and loss of privacy in the cloud. Privacy is a important and fundamental human right that encompasses the right to be left alone, many techniques are proposed under different systems and security models. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private audit ability and public audit ability.

The cloud computing is an emerging technology now a days. While the user enjoying the services provides by the cloud, at the same time they start worrying about the losing control of their data. In the cloud computing provides many services to the users. When the data processed on the clouds are often leading to the number of issues which is related to the accountability of the user and also related to the personal information (health, financial, etc). These will become the significant barrier to the services of cloud. To overcome the above issues the effective mechanism is required to monitor the user's data in the cloud services. If the user need to be able to ensure their data's are handled according to the agreements made at the time they sign on for the services in the cloud. Conventional access control approaches developed for closed domains such as databases and operating systems, or approaches using a centralized server in distributed environments, are not suitable, due to the following features characterizing cloud environments. As a result, data handling in the cloud goes through a complex and dynamic hierarchical service chain which does not exist in conventional environments. To overcome the problems, we propose a novel approach of Cloud Information Accountability (CIA) framework to keep track of the usage of the user's data namely Ensuring Distributed Accountability for Data Sharing in the Cloud. In the cloud services, information accountability focuses on keeping the data usage transparent and trackable. So that CIA framework is introduced for the end-to-end accountability in a highly distributed manner. The main features of the CIA frameworks are ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. In this paper, we also developed the two modes for the auditing mechanisms: one is push mode, this mode send the logs periodically to the data owner or stakeholder. The other is pull mode refers to an alternative approach whereby the user (or another authorized party) can retrieve the logs as needed. The design of the CIA framework leads to the issues like uniquely identifying CSPs, ensuring the reliability of the log, adapting to a highly decentralized infrastructure, etc. to addressing these issues our approach extend to the programmable capability of JAR(Java Archives) files to automatically log the usage of the users data by any entity in the cloud.

If any access will get trigger an automated ad authenticated to the JARs. We referred it has "strong binding" If exists the strong binding means it exists, the all copies of the JARs created and its leads to challenges on the ensuring the integrity of the logging. To overcome these, JAR is introduced with the central point of contact. It records the error correction information. If JAR is not connecting with central point means the enclosed data will be denied. In addition, our approach can handle personal identifiable information provided they are stored as image files. Our experiments demonstrate the efficiency, scalability and granularity of our approach. In addition, we also provide a detailed security analysis and discuss the reliability and strength of our architecture in the face of various nontrivial attacks, launched by malicious users or due to compromised Java Running Environment (JRE). Our main contributions of the paper are shown in the proposed system in detail. This paper is an extension of our previous conference paper [40]. We have made the following new contributions. First, we integrated integrity checks and oblivious hashing (OH) technique to our system in order to strengthen the dependability of our system in case of compromised JRE. We also updated the log records structure to provide additional guarantees of integrity and authenticity. Second, we extended the security analysis to cover more possible attack scenarios. Third, we report the results of new experiments and provide a thorough evaluation of the system performance. Fourth, we have added a detailed discussion on related works to prepare readers with a better understanding of background knowledge.

II. RELATED WORKS

In this section, we briefly discuss the works which is similar techniques as our approach but serve for different purposes.

Shraddha B. Toney and Sandeep U.Kadam , in this paper we review the cloud information accountability framework for the data sharing in which procedural and technical solutions are co-designed to demonstrate accountability by the various researchers to resolving privacy and security risks within the cloud. Usually cloud computing services are delivered by a third party provider who owns the infrastructure. It moves the application software and databases to the most data centres, where the controller of the data and services may not be fully trustworthy which have not been well understood. This paper presents a review on new way to supplement the current consumption and delivery model for IT services based on the Internet, by providing for dynamically scalable framework and often virtualized resources as a service over the Internet. By the time there are a number of notable commercial and individual cloud computing services, including Google, Microsoft and Yahoo. Details of the services provided are abstracted from the users who no longer need to be experts of technology infrastructure. Moreover, users may not know the machines which actually process and host own data.

Marco Casassa Mont, Ilaria Matteucci, Marinella Petrocchi, and Marco Luca Sbodio[48] proposed the paper for Enabling Data Sharing in the Cloud. In this paper Web interactions usually require the exchange of personal and confidential information for a variety of purposes, including enabling business transactions and the provisioning of services. A key issue affecting these interactions is the lack of trust and control on how data is going to be used and processed by the entities that receive this data. In the traditional world, this issue is addressed by using contractual agreements that are signed by the involved parties. This could be done electronically as well but there is currently a major gap between the definition of legal contracts, regulating the sharing of data and the software infrastructures required to support and enforce them. How to ensure that legal contracts can be actually enforced by the underlying IT infrastructure? How to ensure that a potentially enforceable version of the contract corresponds to the legal version of the contract? This article describes our work to address this gap through the usage of electronic Data Sharing Agreements (e-DSA). e-DSAs can be formally defined and analysed to identify inconsistencies and contradictory policies/constraints; they can then be deployed within the IT infrastructure and enforced.

They ensure that no one can add or remove entries in the middle of a provenance chain without detection, so that data are correctly delivered to the receiver. Differently, our work is to provide data accountability, to monitor the usage of the data and ensure that any access to the data is tracked. Since it is in a distributed environment, we also log where the data go. However, this is not for verifying data integrity, but rather for auditing whether data receivers use the data following specified policies. Along the lines of extended content protection, usage control [33] is being investigated as an extension of current access control mechanisms.

III. PROPOSED WORK

As organizations become increasingly reliant on cloud computing for servicing their data storage requirements, the need to govern access control at inner granularities becomes particularly important. This challenge is increased by the lack of policy supporting data migration across geographic boundaries and through organizations with divergent regulatory policies. In this paper, we present architecture for secure and distributed management of provenance, enabling its use in security-critical applications. Provenance, a metadata history detailing the derivation of an object, contains information that allows for expressive, policy-independent access control decisions. We consider how to manage and validate the metadata of a provenance-aware cloud system, and introduce protocols that allow for secure transfer of provenance metadata between end hosts and cloud authorities We propose a novel approach of Cloud Information Accountability (CIA) framework to keep track of the usage of the user's data namely Ensuring Distributed Accountability for Data Sharing in the Cloud. In the cloud services, information accountability focuses on keeping the data usage transparent and trackable. So that CIA framework is introduced for the end-to-end accountability in a highly distributed manner. The main features of the CIA frameworks are ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication. In this paper, we also developed the two modes for the auditing mechanisms: one is push mode, this mode send the logs periodically to the data owner or stakeholder. The other is pull mode.

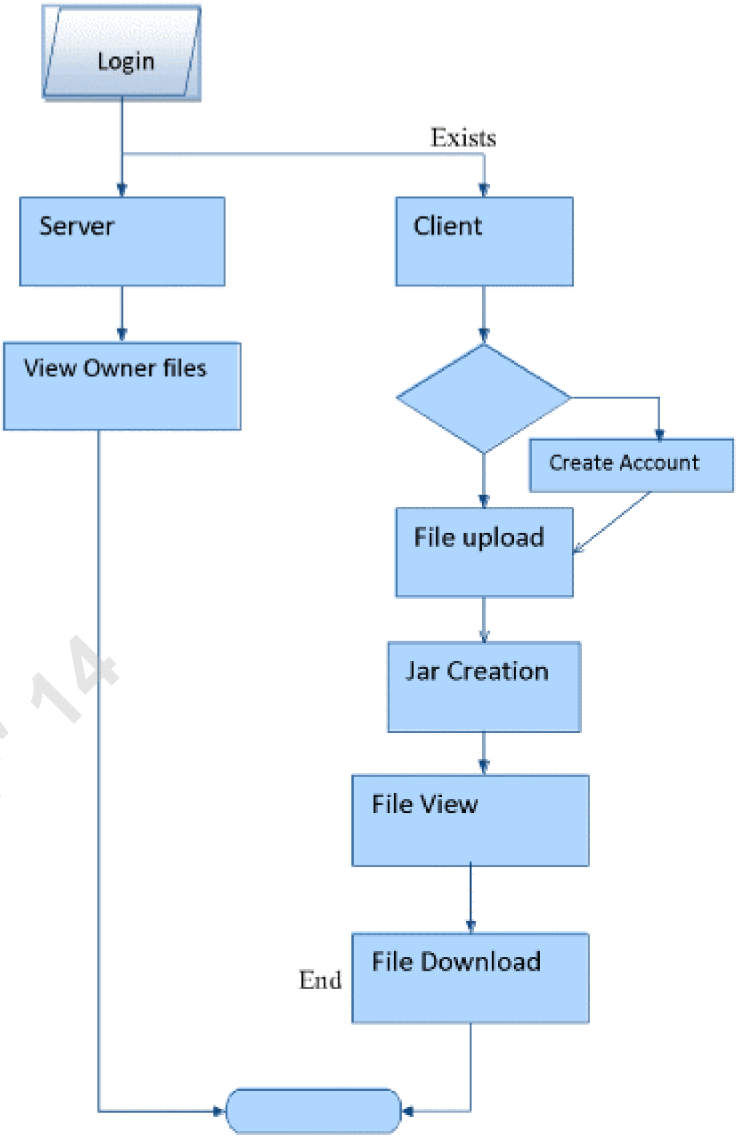

Figure 1: Flow of the proposed system

refers to an alternative approach whereby the user (or another authorized party) can retrieve the logs as needed. The design of the CIA framework leads to the issues like uniquely identifying CSPs, ensuring the reliability of the log, adapting to a highly decentralized infrastructure, etc. to addressing these issues our approach extend to the programmable capability of JAR(Java ARchives) files to automatically log the usage of the users' data by any entity in the cloud. Our experiments demonstrate the efficiency, scalability and granularity of our approach. In addition, we also provide a detailed security analysis and discuss the reliability and strength of our architecture in the face of various nontrivial attacks, launched by malicious users or due to compromised Java Running Environment (JRE). In summary, our main contributions are as follows:

We 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. Our proposed architecture is platform independent and highly decentralized; in that it does not require any dedicated authentication or storage system in place. We go beyond traditional access control in that we provide a certain degree of usage control for the protected data after these are delivered to the receiver. We conduct experiments on a real cloud test bed. The results demonstrate the efficiency, scalability, and granularity of our approach. We also provide a detailed security analysis and discuss the reliability and strength of our architecture.

IV. SIMULATION WORKS

We have simulated our system in JAVA. We implemented and tested with a system Configuration on Intel Dual Core processor, Windows XP and using NETBEANS IDE 7.0. We have used the following modules in our implementation part. The details of each module for this system are as follows:

A. Data owner module

In this module, the data owner uploads their data in the cloud server. The new users can register with the service provider and create a new account and so they can securely upload the files and store it. For the security purpose the data owner encrypts the data file and then store in the cloud. The Data owner can have capable of manipulating the encrypted data file. And the data owner can set the access privilege to the encrypted data file. To allay users' concerns, it is essential to provide an effective mechanism for users to monitor the usage of their data in the cloud. For example, users need to be able to ensure that their data are handled according to the service level agreements made at the time they sign on for services in the cloud.

B. Jar Creation Module

In this module we create the jar file for every file upload. The user should have the same jar file to download the file. This way the data is going to be secured. The logging should be decentralized in order to adapt to the dynamic nature of the cloud. More specifically, log files should be tightly bounded with the corresponding data being controlled, and require minimal infrastructural support from any server. Every access to the user's data should be correctly and automatically logged. This requires integrated techniques to authenticate the entity who accesses the data, verify, and record the actual operations on the data as well as the time that the data have been accessed. Log files should be reliable and tamper proof to avoid illegal insertion, deletion, and modification by malicious parties. Recovery mechanisms are also desirable to restore damaged log files caused by technical problems. The proposed technique should not intrusively monitor data recipients' systems, nor it should introduce heavy communication and computation overhead, which otherwise will hinder its feasibility and adoption in practice.

C. Cloud Service Provider Module

The cloud service provider manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud with the jar file created for each file for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them.

D. Disassembling Attack

In this module we show how our system is secured by evaluating to possible attacks to disassemble the JAR file of the logger and then attempt to extract useful information out of it or spoil the log records in it. Given the ease of disassembling JAR files, this attack poses one of the most serious threats to our architecture. Since we cannot prevent an attacker to gain possession of the JARs, we rely on the strength of the cryptographic schemes applied to preserve the integrity and confidentiality of the logs. Once the JAR files are disassembled, the attacker is in possession of the public IBE key used for encrypting the log files, the encrypted log file itself, and the *.class files. Therefore, the attacker has to rely on learning the private key or subverting the encryption to read the log records.

E. Man-in-the-Middle Attack

In this module, an attacker may intercept messages during the authentication of a service provider with the certificate authority, and reply the messages in order to masquerade as a legitimate service provider. There are two points in time that the attacker can replay the messages. One is after the actual service provider has completely disconnected and ended a session with the certificate authority. The other is when the actual service provider is disconnected but the session is not over, so the attacker may try to renegotiate the connection. The first type of attack will not succeed since the certificate typically has a time stamp which will become obsolete at the time point of reuse. The second type of attack will also fail since renegotiation is banned in the latest version of

OpenSSL and cryptographic checks have been added.

V.CONCLUSION AND FUTURE WORKS

It is 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 the problems.

we proposed the automatically logging any access to the data by using the auditing mechanisms. In this approach the data owner to not only audit and also has the strong back-up protection. The main features of our proposed work are that enables the data owner to audit even those copies of its data that were made without his knowledge. We also discussed the cloud information accountability framework for data sharing in the cloud.

In the future, we plan to refine our approach to verify the integrity of the JRE and the authentication of JARs. For example, we will investigate whether it is possible to leverage the notion of a secure JVM [18] being developed by IBM. This research is aimed at providing software tamper resistance to Java applications. In the long term, we plan to design a comprehensive and more generic object- oriented approach to facilitate autonomous protection of travelling content. We would like to support a variety of security policies, like indexing policies for text files, usage control for executables, and generic accountability and provenance controls.

VI. REFERENCES

[1] P. Ammann and S. Jajodia, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.

[3] E. Barka and A. Lakas, "Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks.

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

[5] R. Bose and J. Frew, "Lineage Retrieval for Scientific Data Processing: A Survey," ACM Computing Surveys, vol. 37, pp. 1-28, Mar. 2005.

[6] P. Buneman, A. Chapman, and J. Cheney, "Provenance Management in Curated Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '06), pp. 539-550, 2006.

[7] B. Chun and A.C. Bavier, "Decentralized Trust Management and Accountability in Federated Systems," Proc. Ann. Hawaii Int'l Conf. System Sciences (HICSS),2004.

[8] [8] OASIS Security Services Technical Committee, "Security
Assertion Markup Language (saml) 2.0,"

[9] 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.

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

[11] Y. Chen et al., "Oblivious Hashing: A Stealthy Software Integrity Verification Primitive," Proc. Int'l Workshop Information Hiding, F. Petitcolas, ed., pp. 400-414, 2003.

[12] S. Etalle and W.H. Winsborough, "A Posteriori Compliance Control," SACMAT '07: Proc. 12th ACM Symp. Access Control Models and Technologies, pp. 11-20, 2007.

[13] X. Feng, Z. Ni, Z. Shao, and Y. Guo, "An Open Framework for Foundational Proof-Carrying Code," Proc. ACM SIGPLAN Int'l Workshop Types in Languages Design and Implementation, pp. 67-78, 2007.

[14] Flickr, http://www.flickr.com/, 2012.

[15] R. Hasan, R. Sion, and M. Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance," Proc. Seventh Conf. File and Storage Technologies, pp. 1-14, 2009

[16] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57- 66, Aug. 2001.

[17] J.W. Holford, W.J. Caelli, and A.W. Rhodes, "Using Self- Defending Objects to Develop Security Aware Applications in Java," Proc. 27th Australasian Conf. Computer Science,vol. 26, pp. 341-349, 2004.

[18] Trusted Java Virtual Machine IBM, http://www.almaden.ibm. com/cs/projects/jvm/, 2012.

[19] P.T. Jaeger, J. Lin, and J.M. Grimes, "Cloud Computing and Information Policy: Computing in a Policy Cloud?," J. Information Technology and Politics, vol. 5, no. 3, pp. 269-
283,
2009
.

[20] R. Jagadeesan, A. Jeffrey, C. Pitcher, and J. Riely,
"Towards a Theory of Accountability and Audit," Proc.
14th European Conf. Research in Computer Security
(ESORICS), pp. 152-167,
2009.

[21] R. Kailar, "Accountability in Electronic Commerce
Protocols," IEEE Trans. Software Eng., vol. 22, no. 5, pp.
313-328, May
1996.

[22] W. Lee, A. Cinzia Squicciarini, and E. Bertino.

[23] J.H. Lin, R.L. Geiger, R.R. Smith, A.W. Chan, and S.Wanchoo, Method for Authenticating a Java Archive (jar)for Portable Devices, US Patent 6,766,353, July 2004.

[24] F. Martinelli and P. Mori, "On Usage Control for Grid Systems," Future Generation Computer Systems, vol. 26, no. 7, pp. 1032-1042, 2010.

[25] T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice), first ed. O' Reilly, 2009.