Comparative Analysis for Ensuring Distributed Accountability of Data Sharing in Cloud

Mr Vineet Saxena1, Simran Ali2
1scholar Ccsit, Teerthanker Mahaveer University, Moradabad
2assistant professor Ccsit, Teerthanker Mahaveer University, Moradabad, India
[1] vineet.saxena84@gmail.com
[2]simranalitmu@gmail.com

Abstract— **The ability to hold cloud service providers accountable for transactions is important when opting for cloud services. Cloud computing is a new term where users can interact directly with the virtualized resources. Through a cloud service, users can store their confidential data on cloud servers and access it from anywhere because the cloud servers are remote machines so the user does not have to be present in the same location as the storage device holding his data. Moreover, users may not have any knowledge about the machines that host and process their data. While this technology has gifted convenience to the users, users have started worrying about the protection of their sensitive data because if any entity accesses their data and tries to alter its originality then they would never come to know about it and this is the issue to which the solution is the feature of accountability. Accountability tracks every important aspect of any data sharing or data usage in the cloud where it is answerable for every action in the system as it can be traced back to some entity while assuring the protection of data from loss and theft. In this paper we review the cloud information accountability framework in which procedural and technical solutions are co-designed to exhibit accountability by the various researches to resolve privacy and security risks within the cloud and presents a review on a new approach to complement the current consumption and delivery model for IT services based over the Internet, by providing a dynamically scalable framework.**

Keywords— **Cloud computing, accountability, data sharing, security, privacy**

I. INTRODUCTION

Cloud computing is a rising pattern in the computer industry that places the complete computing infrastructure over the web. It stores the data and applications of the users in remote servers with the help of internet. The whole idea of cloud computing states that the personal data that the users used to store in their computers can now be stored on centralized servers. It is an initiative taken in the direction of the development of a model in which computing services like servers, databases, storage, networking and software are made available over the internet to the user on his demand so in this way we can imagine the internet as "the cloud"[7][2]. There are benefits associated with cloud computing due to which it has become so popular. Firstly, it eliminates the cost of purchasing the components required for building the cloud computing infrastructure and its maintenance. Secondly, immense amounts of computing resources can be delivered within minutes. Thirdly, the capability to tolerate significant increases in throughput or other potentially limiting factors in a flexible manner which means making available the right amount of computing power, storage and bandwidth from the appropriate location as per the requirement. Fourthly, because all the data of the users is stored in the cloud the backup and recovery procedures are much easier and cheap to implement as compared to storage on a physical device and this feature ensures reliability in the cloud computing environment. Fifthly, with cloud computing one can access his data and work from anywhere and all that one needs is a stable internet connection for that. But even though we have good reasons to adopt cloud computing the benefits provided by it bring along with it major security issues.

When users allow cloud service providers(CSPs) to take care of confidential data, it may give rise to security and privacy issues because they are basically submitting all their sensitive information to a third party which can pose a great risk. The user data is stored and processed in unknown remote machines which the user has no knowledge about. Anyone from the CSP (cloud service provider) company can access the user data and perform any sort of operations on it and the user will not even come to know that his data has been accessed and manipulated by an unauthorized user. The CSP cannot be held accountable if the user loses his data. In this way we can see that user has no actual control over his data. So, in order to address the above stated issues an approach known as cloud information accountability was proposed and it is based on information accountability.

II. PROPOSED SYSTEM

Keeping in mind the security issues in the cloud computing environment (CIA) and the incomplete

and inefficient mechanisms of data protection a new approach has been devised. The proposed approach is called cloud information accountability (CIA) framework[1][7] and it provides accountability at both the ends in a distributed manner. It carries out automated logging and distributed auditing of accesses performed by any entity with the help of jar files. The jar files consist of two components: logger and log harmonizer. The jar files also comprise of access control rules that define the manner in which the cloud service providers and data stakeholders are permitted to access and use the data. The proposed approach will introduce an additional level of security by performing obfuscation on the JAR thus converting it into obfuscated code. For verifying integrity, a technique of provable data possessions will be employed. Each time the data file is accessed the JAR will automatically create a log record[11][12]. The JAR's programmable capabilities are used for creating a dynamic and traveling object, and to ensure that any access to the user's data will generate authentication and automated logging and in this way, the user would be kept informed about the accesses made to his file by anyone else apart from him. It provides data transparency which means that the user will get information about the usage of his data and ensures that the data can be tracked[3][6][5][7][13]. This framework is platform independent and highly decentralized without the need of a dedicated authentication or storage system. It enables a user to enforce access and usage control rules as needed. At the time when the user is adopting the services of the cloud service provider he frames an agreement which states the following:

- The parameters and minimum levels required for each factor of the service, as well as solutions in case of failure to meet those requirements.
- Affirmation of your institution's ownership of its data stored on the cloud service provider's servers, and specifies the user's rights to get it back.
- Details of the system infrastructure and security standards to be maintained by the CSP, along with the rights to review their fulfillment of the standards.
- States the cost to continue and discontinue using the cloud service.
- This agreement is known as service-level agreement and it is important as it maintains the minimum level of service and defines the expectations for both the parties. CIA ensures that this agreement is honored.
- This framework incorporates JVM (java virtual machine) that provides and maintains the highest level of security in the cloud environment. It executes byte code instructions.

III. UTPUT [5][6OUTPUT [5][6]1939ERSITY262MAJOR COMPONENTS

The Cloud Information Accountability framework (CIA) provides the mechanism of automated logging and distributed auditing of access made to the user data by any entity, at any particular time with the help of the two major components:

A. Logger
B. Log Harmonizer

A. Logger

The logger is the component which is strongly attached to the user's data, so whenever the data is accessed it gets downloaded and whenever data is copied it also gets copied and this is because it is bind to the user data. It is responsible for monitoring access to the user data and recording operations that are carried out on user data and it uses a copy of the user's data for this purpose. Another important task that it handles is encryption of log record with the help of public key of the data owner and sending them to the log harmonizer at regular intervals. The deployment of the logger requires very less effort to be put in by the server.

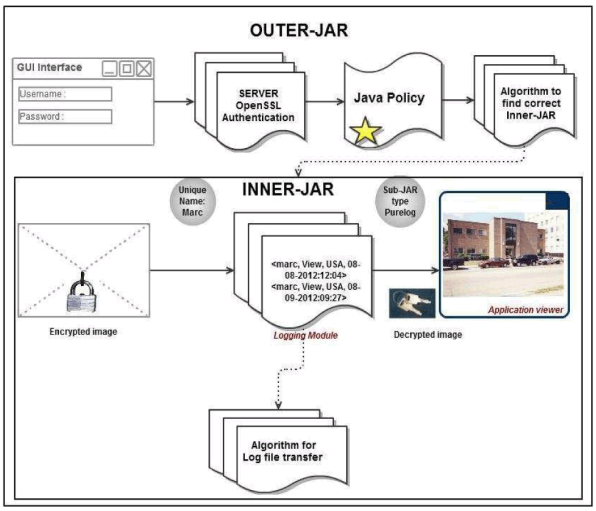

Fig. 1 Structure of logger

The outcome of the tight coupling between the logger and user data is a highly distributed logging system and this satisfies the first design requirement. Moreover, the logger does not have to be installed on any system and it does not need any exclusive support from the server, its actions are not invasive in nature which leads to the fulfillment of the fifth requirement. Another important task performed by the logger is for every log record it generates error correction information combined with encryption and authentication mechanism and sends it to the log harmonizer. This makes the recovery mechanism a robust and reliable one thus fulfilling the third requirement.

This was all about the functions of the logger and the benefits associated with it. Now the detailed working of it will be discussed.

The logger component is a nested java JAR file that holds user data and related log files. The JAR file consists of an outer jar that holds one or more inner JARs. Inner JAR has data in encrypted form and class files which help in acquiring log files and a log file for each encrypted data item. Weil-pairing-based IBE scheme is used to perform encryption. The outer JAR is composed of multiple inner JARs, access policy and class file that performs the task of verifying the server or users and then there's one more class file that locates the correct inner JAR. In this manner, the logger component performs verification of the authenticity of the entities who want to access the user data stored in the JAR file[3][7][2].

B. Log Harmonizer

The log harmonizer allows the user to access the log files and is responsible for auditing. As it is a trusted component, the log harmonizer generates the master key. It retains the decryption key as it holds the responsibility of decrypting the log files. In case if it is known that the channel between the log harmonizer and client is not secure then the process of decryption can be carried out on the client side after the log harmonizer sends the key to the client using a secure key exchange mechanism.

Two auditing methods are supported by the log harmonizer: push and pull.

The push method pushes back the log file to the data owner from time to time and this operation is carried out automatically by the log harmonizer.

The pull method makes the log file available to the data owner as often as requested so in this way it follows an on-demand approach.

If at any particular time, there is more than one logger for the same data set then the log harmonizer merges the log records before sending it to the data owner. If a situation arises in which the log file gets corrupted then the log harmonizer handles the issue of recovering the log files. In addition, the log harmonizer can even carry out the function of logging all by itself along with auditing. Keeping the logging and auditing mechanisms separate leads to improved performance.

The harmonizer is implemented as a JAR file. It does not contain the user's data being audited but consists of class files that allow the server and client processes to communicate with the logger components. The harmonizer stores error correction information send to it by the logger component and the user's decryption key to decrypt the log records. Various copies of user's data JAR file may lead to duplicate records so the log harmonizer also deals with the duplicate records.

The logger and the log harmonizer are both implemented as lightweight and portable JAR files and this manner of implementation serves automatic logging mechanism[3][6][4].

IV. BUILDING MODULES

The building modules of the CIA framework are as follows:

A. Data Owner Module
B. Jar Creation Module
C. Cloud Service Provider
D. Disassembling Attack
E. Man-In-The-Middle Attack

A. *Data Owner Module*

In this module, a user registers with cloud service provider and creates an account. After his account has been created the user can upload important data and store it in a secure way. To provide protection to the data the user encrypts the data and then stores it on the cloud. The user can even perform manipulation on the encrypted data and can also set the access privileges in which the extent to which an entity can use, read, write to or perform other operations on the data is well defined.

To address the security concerns of the users a mechanism has been proposed to monitor the usage of the data in the cloud[10].

For example, users should be able to stay assured that their data is being handled according to the SLA (service-level agreement).

A. Jar Creation Module

In this module, every time a user uploads a data file a jar file for it is created so the user would be required to use the same jar file in order to download the data file that he had uploaded[10]. This ensures the protection of data on the cloud. The log files are tightly bound to the user data. The logger does not require much support from the server and every access to the user data is monitored appropriately and automatically with the help of techniques that verify the authenticity of the entity who accesses the data. The logger also records operations carried out on the data and the time of access. Log files should be as such that no malicious entity is able to perform illicit operations on the data. It carries out recovery mechanisms for corrupt log files due to technical faults. The proposed framework does not act in an intrusive manner while monitoring the user data systems nor does it incur heavy computation overhead which makes its adoption easy and reliable.

B. Cloud Service Provider Module

The role of a cloud service provider is to organize a cloud and provide a secure service for data storage to the registered users. The user encrypts his data file and uploads it to the cloud along with the jar file created for that file and whenever he requires the data file, he can download the encrypted file using the same jar file and once it downloads he can decrypt and use it[10].

C. Disassembling Attack

In this module, it is shown how the system is made secure against an attack in which the jar file of the logger is disassembled and useful information is extracted from it or log records are damaged. The JAR file is quite easy to disassemble and this can pose a serious threat to the cloud architecture and because we cannot stop an attacker from taking control over the JAR file we depend on the strength of the cryptographic schemes applied[1][7] with full confidence to protect the integrity and confidentiality of the logs.

D. Man-in-the-middle attack

In this module, the attacker pretends to be a genuine service provider by obstructing the messages during the authentication of a service provider with the certificate authority and replies the messages either when the actual service provider has ended the session with the certificate authority or when the service provider got disconnected even when the session has not ended in this way negotiating the connection all over again[10]. But in both the situations, the attacker will not succeed because in the first case the certificate has a time stamp and will turn obsolete when the time of reuse comes and in the second case because of OpenSSL and addition of cryptographic checks the renegotiation will fail.

V. DATA FLOW

At first, the user generates a pair of public and private keys using the identity-based encryption scheme which is a Weil-pairing-based scheme. With the help of generated key, the user creates a cia component called logger which is actually a jar file to store data. The JAR file consists of access control rules in which it is precisely stated whether and how the cloud servers, other users or companies are authorized to access the data content. The jar file is then sent to the cloud service provider to which the user has signed up with. In order to authenticate the CSP (cloud service provider) OpenSSL-based certificates are used in which a trusted certificate authority certifies the CSP. In case when a user requests access, SAML (a data format which assists the identity provider and service provider to exchange authentication and authorization) enables the identity provider to issue certificate verifying the user's identity. After authenticating the user will be allowed to access the data in the JAR file. Every time an access is made to the data the JAR creates a log record, encrypts it using an encryption key and appends it with the data. The encryption of the log file protects it from the attackers. The owner of the data can choose to reuse the same key pair for all JARs or can even create distinct key pairs for each separate JAR. Security would thus improve if different keys are used without leading to any unnecessary overhead. Log harmonizer would manage copies of JAR created at the time when users view or download the data. It would create a chain structure by hashing together individual records which would enable it to quickly detect errors or locate missing records. Moreover, if in case a log file gets corrupted due to any reason the logger sends error correction information to the log harmonizer and in this way the log harmonizer deals with the log file corruption issues. The encrypted log files can be later decrypted using the private key that the log harmonizer holds and the integrity of log files can be verified. The log files are sent to the data owner from time to time or can be even requested by the data owner whenever he needs it. Fig 2 shows the working of the CIA framework.

VI. CONCLUSIONS

As the security issues are increasing day by day it is becoming more important to protect the confidentiality of user data on the Internet against unwanted and unauthorized usage and access.

Despite framing and implementing laws and technical mechanisms to address the security concerns at the moment, there are no complete solutions to solve the problem with the help of various resources, security and privacy issues have been studied and an effective mechanism known as cloud information accountability(CIA) has been proposed which incorporates automatic logging and auditing mechanisms that monitor access made to the user data and creates log records of access and usage of user's data so in this way the owner of the data would be kept informed about the usage and access made to the data and he can stay assured that the data is being handled as per the SLA(service level agreement).

Future research would have to be conducted concentrating more on JAR authentication and integrity of JRE. A generic java application is required to facilitate autonomous protection of traveling data.

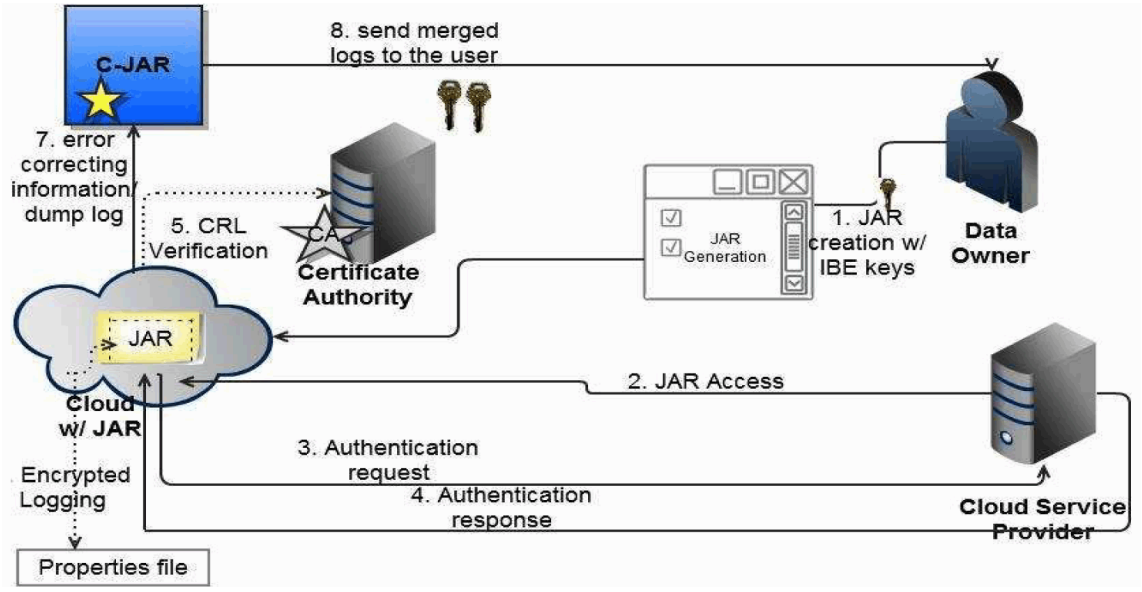

Fig 2. Process of data flow

REFERENCES

[1] S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang, "Promoting Distributed Accountability in the Cloud", Proc. IEEE Int'l Conf. Cloud Computing, 2011.

[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] V.G. Vineetha and T. Mary Daphne, "CIA Framework for Ensuring Accountability of Data in Cloud", International Conference on Applied Mathematics and Theoretical Computer Science 2013.

[4] R.N.Heames and Dr.P.Sudhakar, "Data accountability in cloud using reliable log files forwarding", International Journal on Recent and Innovation Trends in Computing and Communication Vol: 1 Issue: 4.

[5] Thejaswini M U, "Effective Distributed Accountability For Data Sharing In Cloud", International Conference On Advances in Computer & Communication Engineering (ACCE - 2015).

[6] S. Khairnar, P. khare and S. Khan, "Ensuring Distributed Accountability for Data Sharing On the Cloud", IJIR Vol-2, Issue-6, 2016.

[7] H.Arun , R.Nilam, R.Namrata and S.Purva, "Review on Techniques to Ensure Distributed Accountability for Data Sharing in the Cloud", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 10, October 2013.

[8] R. K Ganesh and Aranya Hari, "Enhancing Privacy in Cloud by Avoiding Misuses of Files", International Journal of Advanced Research in Computer Science and Software Engineering, Vol 3, Issue 3, March 2013.

[9] Hitendra and Dr. Sandhya Tarar, "Obfuscation as a Security Measure in Cloud Computing", International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016.

[10] M.S. Rani and M.J Lakshmi, "Ensuring Distributed Accountability for Data Sharing in the Cloud", International Journal of Scientific Engineering and Technology Research Volume.03, Issue No.34, November-2014.

[11] E. Madhavarao, M. Parimala and C. JayaRaju, "Data Sharing In The Cloud Using Distributed Accountability", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 6, June 2013.

[12] A. M. Bagade and S. Suryawanshi, "Distributed Accountability for Data Sharing in Cloud", International Journal of Computer Applications (0975 – 8887) Vol 59– No.8, Dec 2012.

[13] V.V Lonare and Prof .J.N. Nandimath, "Ensuring Distributed Accountability for Data Sharing in Cloud Using AES and SHA", (IJCSIT), Vol. 6 (1) , 2015.

[14] T. Praveenkumar , K. Narsimhulu, "Secure and Accountable Data Sharing In the Cloud", IJCTT– volume 4 Issue 7–July 2013.

[15] E. Madhavarao , M. Parimala and C. JayaRaju, "Data Sharing In The Cloud Using Distributed Accountability", IJARCET, Volume 2, Issue 6, June 2013.