Security and Access Control Evaluation for Cloud Data Centers

P.BALASUBRAMANIAN
II ME (CSE)
Nandha Engineering College
Erode

***Abstract*-Cloud computing is a computing paradigm that enables highly scalable services to be consumed over the Internet on a shared basis. Cloud service providers place data in data centers which is distributed as shared data to the users. Users' data are usually processed remotely in unknown machines that users do not ownership. Centralized monitoring applications are not suitable in the cloud environment and hence data access is provided on a shared basis by cloud service providers. These shared data values are monitored by different cloud auditing schemes. Cloud Information Accountability is an efficient framework for this auditing. This combines the data aspects of access control, usage control and authentication. CIA uses JAR(Java Archives) files for auditing functionality. Two modes of auditing are push mode and pull mode. The push mode refers to logs that are sent to the data owner in a periodic fashion. The pull mode refers to the scenario wherein any authorized person can access the logs when needed. JAR files log the data values in a periodic manner. These data are sent along with access control policies and logging policies enclosed in JAR files, to cloud service providers. When the data are accessed by any external entity, logging mechanism is automatically triggered. In the proposed system, JAR authentication is provided and JRE integration verification is also provided. The CIA model is enhanced with authentication and integrity analysis models. It is a platform independent accountability management model.**

***Keywords*-Cloud computing, accountability, auditing, data sharing, JAR files, Cloud Information Accountability**

I. Introduction

Cloud computing is a means by which highly scalable, technology-enabled services can be easily consumed over the Internet on a need basis. Cloud computing is a computing environment wherein computing resources which may be hardware or software are delivered as a service via a network. Cloud computing entitles resource sharing to achieve best utility over a network. There are a large number of commercial cloud computing services, including Amazon, Google, Microsoft, Yahoo. End users access cloud-based applications through a web browser or a light-weight desktop or mobile app while the business software and user's data are stored on servers at a remote location. But these users may not know where their data are processed and hosted. Hence, users face lots of problems in losing control of their own data. This is because the data that are processed on cloud service providers are often outsourced, which leads to a number of issues associated with accountability, handling of personally identifiable information. These problems hinder the wide adoptability of cloud services over the network. Therefore, the relative security of cloud computing services is a contentious issue that may be delaying its adoption.

To take care of these concerns, there must be provided some mechanism that allows the users to monitor their data usage in cloud environment. Users must be able to know who all access their data in a timely fashion. Users should also be able to ensure that their data are handled exactly based on the service level agreements that are made at the time they sign with the service provider. Traditional approaches that are designed for databases or operating systems or distributed environment are not suitable for this concern. To overcome those problems, a new approach, namely Cloud Information Accountability (CIA) framework, based on the concept of information accountability is adopted. Information accountability focuses on keeping the data usage transparent and traceable. This entitles the ability to easily access and work with data no matter where they are located or what application created them. It also brings the assurance that data being reported are accurate and are coming from the official source. The proposed CIA framework also provides end-to end accountability. By end to end accountability, we mean that one can check whether the policies that govern data manipulations and inferences were in face adhered to. Moreover, it has the ability to maintain lightweight and powerful accountability that combines aspects of access control, usage control and authentication. Thus by using CIA, data owners can track not only whether the service-level agreements are being obeyed, but can enforce access and usage control rules also.

The CIA framework possesses several challenges. This is related with unique identification of CSPs, assurance of reliability of the log, adaptation to a highly decentralized infrastructure, etc. These issues can be addressed by extending the programming capability of JAR (Java Archives) files to automatically log the usage of the users' data by any entity in the cloud. When the user's data is accessed by any entity any time, it is logged by JAR. Initially, users provide their data along with some policies like access policies which they want to abide with to the cloud service providers. These policies are also enclosed in JAR files. When any data is accessed, an automated and authenticated logging mechanism local to the JARs is initiated. This is referred to as "strong binding". The user will have control over his data at any location by having copies of JARs. The JARs are provided with a central point of contact so as to form a link between them and the user. It records the error correction information sent by the JARs. This information results from the loss of any logs from any of the JARs. Moreover, if a JAR is not able to contact its central point, any access to its enclosed data will also be denied. Basically, JAR is an archive file format typically used to aggregate many Java class files and associated metadata and resources (text, images and so on) into one file to distribute application software or libraries on the Java platform. This is now utilized for logging the usage of data.

II. Problem Domain

The common requirements are initially identified and several guidelines to achieve data accountability in the cloud need to be developed. When a user wants to subscribe to a particular cloud service, he usually needs to send his/her data as well as associated access control policies or logging policies to the cloud service provider. After the data are received by the cloud service provider, the service provider will have granted access rights, such as read, write, and copy, on the data. Once these access rights are granted, the data will be fully available at the service provider only.

The following requirements need to be met to provide efficient functionality. In order to adapt to the dynamic nature of the cloud, the logging should be made decentralized. This involves distributing the administrative functions or powers of a central authority among several local authorities. In addition, log files should be strongly coupled with the corresponding data being controlled, and require minimal infrastructural support from any cloud service provider. Next requirement is that each access to the user's data should be correctly and automatically logged by some effective mechanism. Another requirement is that log files should be reliable and tamper proof. This prevents any illegal insertion, deletion, or modification of information by any malicious parties outside. In cases where log files are damaged by some technical or intruder means, recovery mechanisms are a best policy to cope up with. Log files should be sent back to their data owners periodically to inform them about the current usage of their data. Data owners are the ultimate personnel and they must be informed about their data usage in an effective manner. More importantly, log files should be able to be retrieved anytime by their data owners when needed regardless of the location where the files are stored. This is yet another requirement. How Cloud Information Accountability can be used to achieve these goals is discussed herewith.

III. Solution Framework

Cloud Information Accountability framework is considered as the solution for the problem statements here. It performs automated logging and distributed auditing of relevant access performed by some entity, carried out at any point of time at any cloud service provider.

A. *Cloud Information Accountability*

Accountability is defined as the obligation and/ or willingness to demonstrate and take responsibility for performance in light of agreed-upon expectations. Cloud Information Accountability (CIA) has two major components: 1)logger and 2)log harmonizer.

i)Logger: The logger is the entity that is heavily coupled with data of the user. Therefore, when the data are accessed, it is downloaded and is copied whenever the data are copied. The main functionalities of logger includes automatically logging access to data items that it contains, encrypting the log record using the public key of the content owner, and periodically sending them to the log harmonizer. It handles a particular instance or copy of the user's data and is responsible for logging access to that instance or copy.

Logger is usually configured to ensure that access and usage control policies associated with the data are honored. The logger requires only minimal support from the server for it to be deployed. The tight coupling between data and logger, results in a highly distributed logging system, which is very useful. Furthermore, since the logger does not need to be installed on any system or require any special support from the server, it is not very intrusive in its actions.

Moreover, the logger is also responsible for generating the error correction information for each log record and sends the same to the log harmonizer, which is the second component in CIA. This error correction information combined with the encryption and authentication mechanism provides a robust and reliable recovery mechanism.

ii)Log harmonizer: The log harmonizer performs auditing function. It forms the central component which allows the user to access the log files. Log harmonizer is considered as a trusted entity and therefore generates a master key. It holds on to the decryption key for the IBE key pair, as it is responsible for decrypting the logs. Alternatively, the decryption can be carried out on the client end if the path between the log harmonizer and the client is not trusted. In this case, the harmonizer sends the key to the client in a secure key exchange.

B.Auditing

Auditability is also an enabler of (retrospective) accountability: It allows an action to be reviewed against a pre-determined policy to decide if the action was compliant, and if it was not, to hold accountable the person or organization responsible for the action. To allow users to be informed about their data usage timely and accurately, distributed logging mechanism is complemented by an innovative auditing mechanism. Two auditing modes are supported in the system: 1) push mode 2) pull mode.

i)Push mode: In this mode, the logs are 'pushed' to the data owner by the log harmonizer in an automated fashion periodically. The push action will be triggered under two conditions: one is when the time elapses for a period of time, other is when the size of JAR file exceeds the size mentioned by the owner at the time of creation. The period of time mentioned in the first condition is based on the temporal timer inserted as part of the JAR file. After the logs are sent to the data owner, the log files will be dumped, so as to free the space for future access logs. Along with the log files, the error correcting information for those logs is also dumped.

The push mode serves two essential functions in the logging architecture: 1) it ensures that the size of the log files does not explode and 2) it enables timely detection and correction of any loss or damage to the log files. The auditor, upon receiving the log file, will verify its cryptographic guarantees, by checking the records' integrity and authenticity. By construction of the records, the auditor, will be able to quickly detect forgery of entries, using the checksum added to each and every record.

ii)Pull mode: Pull mode is an on-demand approach. It allows auditors to retrieve the logs anytime when they want to check the recent access to their own data. The pull message consists simply of an FTP pull command. This pull command can be issues from the command line. The request will be sent to the harmonizer, and the user will be informed of the data's locations and obtain an integrated copy of the authentic and sealed log file.

C. Procedure

The overall CIA framework, which combines data, users, logger and log harmonizer is shown in the figure.

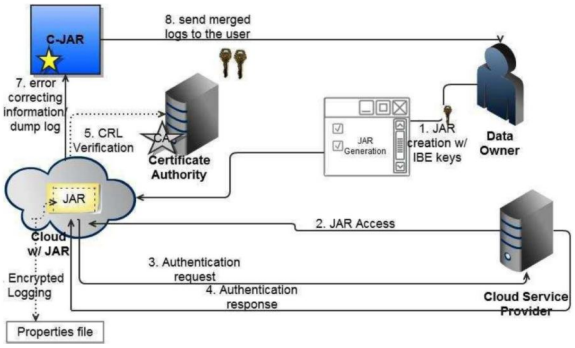

Fig 1 CIA framework

Initially, each user creates a pair of public and private keys using Identity-Based Encryption. The user will create a logger component which is a JAR file using this key, to store its data items. The JAR file includes a set of simple access control rules specifying whether and how the cloud servers and possibly other data stakeholders are authorized to access the content itself. Then, he sends the JAR file to the cloud service provider that he subscribes to. To authenticate the CSP to the JAR, it uses Open SSL based certificates, wherein a trusted certificate authority certifies the CSP. In the event that the access is requested by a user, this employs SAML-based authentication, wherein a trusted identity provider issues certificates verifying the user's identity based on his username. Once the authentication succeeds, the service provider will be allowed to access the data enclosed in the JAR file.

Depending on the configuration settings defined at the time of creation, the JAR will provide usage control associated with logging, or will provide only logging functionality. Each time there is an access to the data, the JAR will automatically generate a log record, encrypt it using the public key distributed by the data owner, and store it along with the data. The encryption of the log file prevents

unauthorized changes to the file by attackers. The data owner can either reuse the same pair of keys for all JARs or can create different key pairs for separate JARs. Usage of separate keys can enhance the security without introducing any overhead except in the initialization phase. In addition, some error correction information will be sent to the log harmonizer to handle any corruption of log file. To ensure trustworthiness of the logs, each record is signed by the entity accessing the content. This is similar to a digital signature. Further, individual records are hashed together to create a chain structure, thus detecting possible errors or missing records. The encrypted log files can later be decrypted and their integrity can be verified. They can be accessed by the data owner or other authorized stakeholders at any time for auditing purposes with the help of the log harmonizer, which is a part of the framework.

IV. Security and Access Control Evaluation Schemes

The system is designed to perform data center management and access control activities. Decentralized access control monitoring is provided in the system. Object based access monitoring is performed for the data owners. The system is divided into six major modules. They are Data owner, Cloud data center, client, JAR authentication, Security and access control, and Attack verification.

A..Data Owner

The data owner shares the data files to the clients. Data files are provided with different access permissions. Access permissions are assigned by the data owner based on the user group. The system is designed with multiple data owners.

B. Cloud Data Center

The cloud data center provides storage spaces for the cloud users. Shared data files provided by data owners are uploaded to the cloud data centers. Client requests are processed by the data centers and access logs are also maintained under the cloud data centers.

C. Client

The client application is designed to access the data files under the cloud environment. The data owner assigns the client access levels. Data files are provided with reference to the access levels. Client collects the data files from the data centers.

D. *JAR Authentication*

The JAR files are distributed from the data centers with the data files. The classes in the JAR components are authenticated by the data centers. The JAR execution is initiated after the access Authentication methods are used to control anonymous JAR component access.

E. Security and Access Control

The security and access control methods are used to verify the JAR components. Data access levels are monitored and verified with client permissions. Client monitoring codes are provided with different access levels. Access level based functions are integrated in the monitoring component.

F. Attack Verification

The attack verification is carried out with integrity checking methods. Data and runtime integrity checking methods are used in the system. The data integrity verification is used to check the data transmission process. The runtime verification is performed to verify the code execution process.

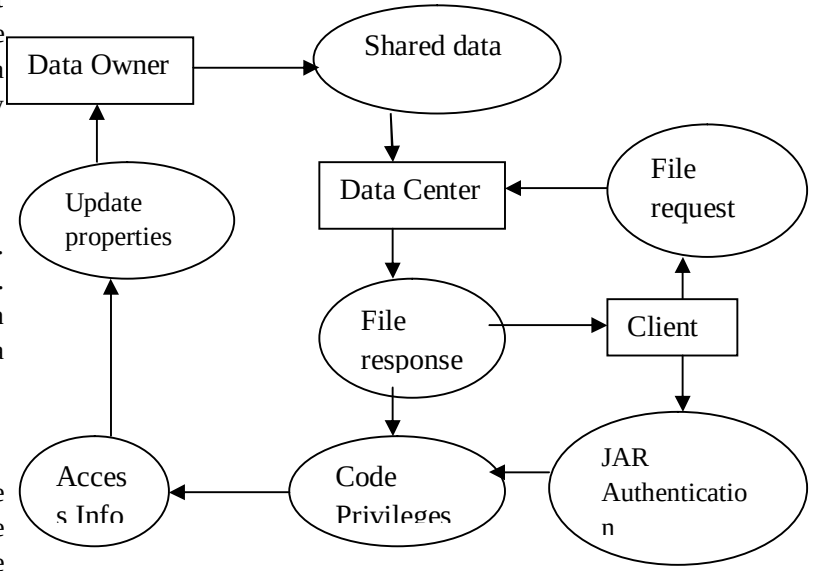

Fig 2 Cloud Data Accounting System

V. Conclusion

The data centers are used to share the data around cloud nodes. Cloud Information Accountability (CIA) framework is used to perform data access monitoring process. This paper proposes a new automatic and enforceable logging mechanism in the cloud. This uses JAR files for data

accountability which enhances the easiness and usage. The CIA model is enhanced with authentication and integrity analysis models. The system security is ensured with data and executable access control mechanism. Accountability monitoring is carried out under the usage environment. Policy based model integrates security and accounting process. The proposed architecture is platform independent and highly decentralized, in that it does not require any dedicated authentication or storage system in place. This incorporates JAR programmable capabilities to both create a dynamic and traveling object. Distributed auditing mechanisms are also provided to strengthen user's control.

REFERENCES

[1] A.Pretschner, F. Schuo¨ tz, C. Schaefer, and T. Walter, *"Policy Evolution in Distributed Usage Control,"* Electronic Notes Theoretical Computer Science, vol. 244, pp. 109-123, 2009.

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

[3] SmithaSundareswaran, Anna C. Squicciarini, and Dan Lin, *"Ensuring Distributed Accountability for Data Sharing in the Cloud"*.IEEE Transactions on Dependable And Secure Computing, Vol. 9, No. 4, July/August 2012.

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

[5] M.C. Mont, S. Pearson, and P. Bramhall, *"Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services,"* Proc. Int'l Workshop Database and Expert Systems Applications (DEXA), pp. 377-382, 2003.

[6] M. Xu, X. Jiang, R. Sandhu, and X. Zhang, *"Towards a VMMBased Usage Control Framework for OS Kernel Integrity Protection,"* SACMAT '07: Proc. 12th ACM Symp. Access Control Models and Technologies, pp. 71-80, 2007.

[7] 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.

[8] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, *"Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing,"* Proc. European Conf. Research in Computer Security (ESORICS), pp. 355-370, 2009.

[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] RagibHasan, RaduSion and Marianne Winslett, *"Preventing History Forgery with Secure Provenance"*, May 24, 2009.

[11] 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.

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