ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Distributed auditing mechanism in order to strengthen user's control over data in Cloud computing Environment

Chandra Sekhar Murakonda
M.Tech Student,
Department of Computer Science Engineering,
NRI Institute of Technology,Guntur.

Hanumantha Rao Murukutna, MCA,M.Tech,
Assistant Professor,
Department of Computer Science Engineering,
NRI Institute of Technology,Guntur.

Abstract:

Cloud computing is is typically defined as a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. In cloud computing, the word cloud (also phrased as "the cloud") is used as a metaphor for "the Internet," so the phrase cloud computing means "a type of Internet-based computing," where different services — such as servers, storage and applications — are delivered to an organization's computers and devices through the Internet.

A major characteristic of the cloud services is that users' data are usually processed remotely in unknown machines that users do not operate. It can become a substantial roadblock to the wide adoption of cloud services.

To address this problem, we propose a highly decentralized answerability framework to keep track of the actual usage of the user's data in the cloud. The Cloud Information Accountability framework proposed in this work conducts automated logging and distributed auditing of relevant access performed by any entity, carried out at any point of time at any cloud service provider. It has two major elements: logger and log harmonizer.

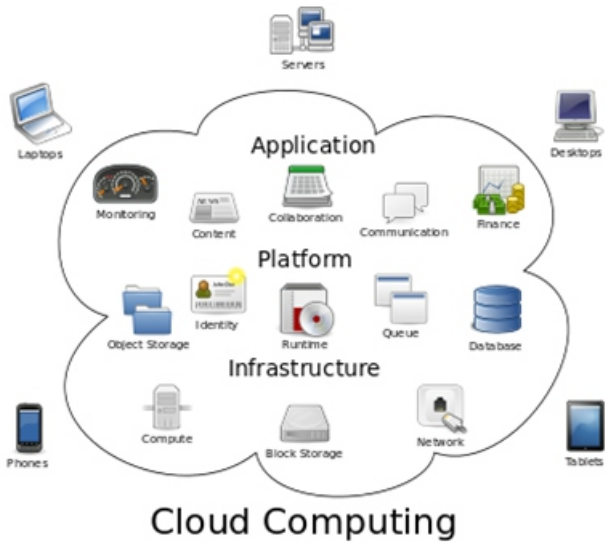
The proposed methodology will also take concern of the JAR file by converting the JAR into obfuscated code which will adds an additional layer of security to the infrastructure. Apart from that we are going to increase the security of user's data by provable data possessions for integrity verification.

Keywords:

Privacy, Cloud computing, data sharing, information accountability framework, Provable data possession.

Introduction:

In a cloud computing system, there's a significant workload shift. Local computers no longer have to do all the heavy lifting when it comes to running applications. The network of computers that make up the cloud handles them instead. Hardware and software demands on the user's side decrease. The only thing the user's computer needs to be able to run is the cloud computing system's interface software, which can be as simple as a Web browser, and the cloud's network takes care of the rest.



The National Institute of Standards and Technology's definition of cloud computing identifies "five essential characteristics":On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiringhuman interaction with each service provider. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Resource pooling:

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Rapid elasticity:

Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear unlimited and can be appropriated in any quantity at any time.

Measured service:

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

EXISTING SYSTEM:

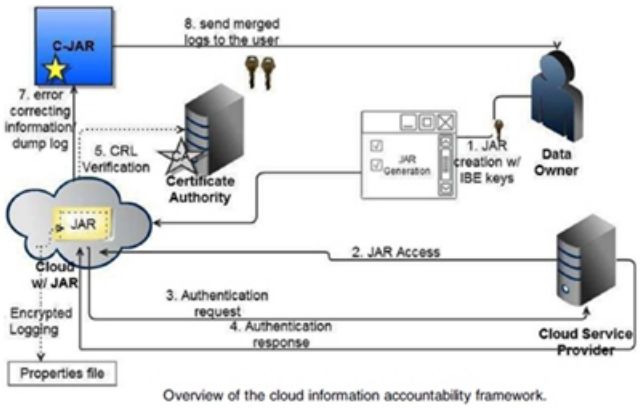
To allay users' concerns, it is essential to provide an effective mechanism for users to monitor the usage of their data in the cloud. For example, users need to be able to ensure that their data are handled according to the service level agreements made at the time they sign on for services in the cloud.

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.

PROBLEMS ON EXISTING SYSTEM:

First, data handling can be outsourced by the direct cloud service provider (CSP) to other entities in the cloud and theses entities can also delegate the tasks to others, and so on.

Second, entities are allowed to join and leave the cloud in a flexible manner. As a result, data handling in the cloud goes through a complex and dynamic hierarchical service chain which does not exist in conventional environments.



Overview of the cloud information accountability framework.

PROPOSED SYSTEM:

We propose a novel approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability. Unlike privacy protection technologies which are built on the hide-it-or-lose-it perspective, information accountability focuses on keeping the data usage transparent and trackable. Our proposed CIA framework provides end-to-end accountability in a highly distributed fashion. One of the main innovative features of the CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication.

By means of the CIA, data owners can track not only whether or not the service-level agreements are being honored, but also enforce access and usage control rules as needed. Associated with the accountability feature, we also develop two distinct modes for auditing: push mode and pull mode.

The push mode refers to logs being periodically sent to the data owner or stakeholder while the pull mode refers to an alternative approach whereby the user (or another authorized party) can retrieve the logs as needed.

Our main contributions are as follows:

- We propose a novel automatic and enforceable logging mechanism in the cloud.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

• 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 testbed. 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.

IMPLEMENTATION:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

MAIN MODULES:

1. Data Owner Module.

2. Jar Creation Module.

3. Cloud Service Provider Module.

4. Disassembling Attack.

5. Man-in-the-Middle Attack

MODULES DESCRIPTION:

1. 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.

2. 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.

3. 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

4. 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.

To compromise the confidentiality of the log files, the attacker may try to identify which encrypted log records correspond to his actions by mounting a chosen plaintext attack to obtain some pairs of encrypted log records and plain texts. However, the adoption of the Weil Pairing algorithm ensures that the CIA framework has both chosen ciphertext security and chosen plaintext security in the random oracle model. Therefore, the attacker will not be able to decrypt any data or log files in the disassembled JAR file.

Even if the attacker is an authorized user, he can only access the actual content file but he is not able to decrypt any other data including the log files which are viewable only to the data owner.1 From the disassembled JAR files, the attackers are not able to directly view the access control policies either, since the original source code is not included in the JAR files. If the attacker wants to infer access control policies, the only possible way is through analyzing the log file.

This is, however, very hard to accomplish since, as mentioned earlier, log records are encrypted and breaking the encryption is computationally hard. Also, the attacker cannot modify the log files extracted from a disassembled JAR. Would the attacker erase or tamper a record, the integrity checks added to each record of the log will not match at the time of verification, revealing the error. Similarly, attackers will not be able to write fake records to log files without going undetected, since they will need to sign with a valid key and the chain of hashes will not match.

5. 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.

Conclusion:

We introduced modern approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. Our approach allows the data owner to not only audit his content but also enforce strong back-end protection if needed. Apart from that we have enclosed PDP methodology to enhance the integrity of owner's data.

In future, we plan to refine our approach to verify the integrity of JRE. For that we will look into whether it is possible to leverage the advantage of secure JVM being developed by IBM and we would like to enhance our PDP architecture from user end which will allow the users to check data remotely in an efficient manner in multi cloud environment.

REFERENCES:

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

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

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

[5] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "A Logic for Auditing Accountability in Decentralized Systems," Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005.

[6] A. Squicciarini , S. Sundareswaran and D. Lin, " Preventing Information Leakage from Indexing in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing, 2010.

[7] Q. Wang, C. Wang, K. Ren, W. Lou and J. Li,"Enabling public auditability and data dynamics for storages security incloud computing", in INFOCOM.IEEE,2010,pp. 525-533.

[8] C.Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing,"in INFOCOM. IEEE, 2010, pp. 525–533.

[9] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Intl Cryptology Conf.

[10] HP Cloud Website.

[11] Advances in Cryptology,pp. 213-229, 2001.B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, 1993.

[12]http://en.wikipedia.org/wiki/Identity-based_encryption.