ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Data Privacy and security for Distributed Accountabilitty Using Logger in Cloud

U.Harshavardhini[1], M.Sayee kumar[2], T.Aravind[3]

ME-CSE (Final year), Muthayammal Engineering College, Rasipuram, India[1]

Assistant Professor, Muthayammal Engineering College, Rasipuram, India[2]

Assistant Professor, Muthayammal Engineering College, Rasipuram, India[3]

ABSTRACT: Cloud computing has emerged as one of the most influential paradigms in the IT industry for last few years. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines. While enjoying the convenience brought by this new emerging technology, users' fears of losing control of their own data. To address this problem, in this work, we propose a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. In particular, we propose an object-centered approach that enables enclosing our logging mechanism together with users' data and policies.JAR programmable capabilities are leveraged to both create a dynamic and traveling object, and to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. To strengthen user's control,distributed auditing mechanisms are enhanced. I enhance my work by providing authentication of JAR. This allows the developer to develop powerful applications even they can modify the code and audit the code of the copied code by the attacker because the applications have more access to the computer resources of the client machine.

KEYWORDS: Cloud computing, Privacy and security, JAR files, Cloud Information Accountability

I.INTRODUCTION

Cloud computing enables highly scalable services to be easily consumed over the Internet on an as-needed basis. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. While enjoying the convenience brought by this new emerging technology, users' fears of losing control of their own data (particularly, financial and health data) can become a significant barrier to the wide adoption of cloud services. However, allowing cloud service providers (CSPs), which are not in the same trusted domains as enterprise users, to take care of confidential data, may raise potential security and privacy issues. To keep the sensitive user data confidential against untrusted CSPs, a natural way is to apply cryptographic approaches, by disclosing decryption keys only to authorized users. Achieve flexible and fine-grained access control but those schemes are only applicable to systems in which data owners and the service providers are within the same trusted domain. Normally data owners and service providers are not in the same trusted domain in cloud computing. Enterprises usually store data in internal storage and install firewalls to protect against intruders to access the data. They also standardize data access procedures to prevent insiders to disclose the information without permission. In cloud computing, the data will be stored in storage provided by service providers. Service providers should not be a trusted one anyhow they are all third party.

II.SYSTEM MODEL

Cloud Information Accountability (CIA) framework, based on the notion of information accountability. Privacy protection technologies are built on the hide-it-or-lose-it perspective, information accountability focuses on keeping the data usage transparent and tractable. CIA framework provides end-to end accountability in a highly distributed fashion.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

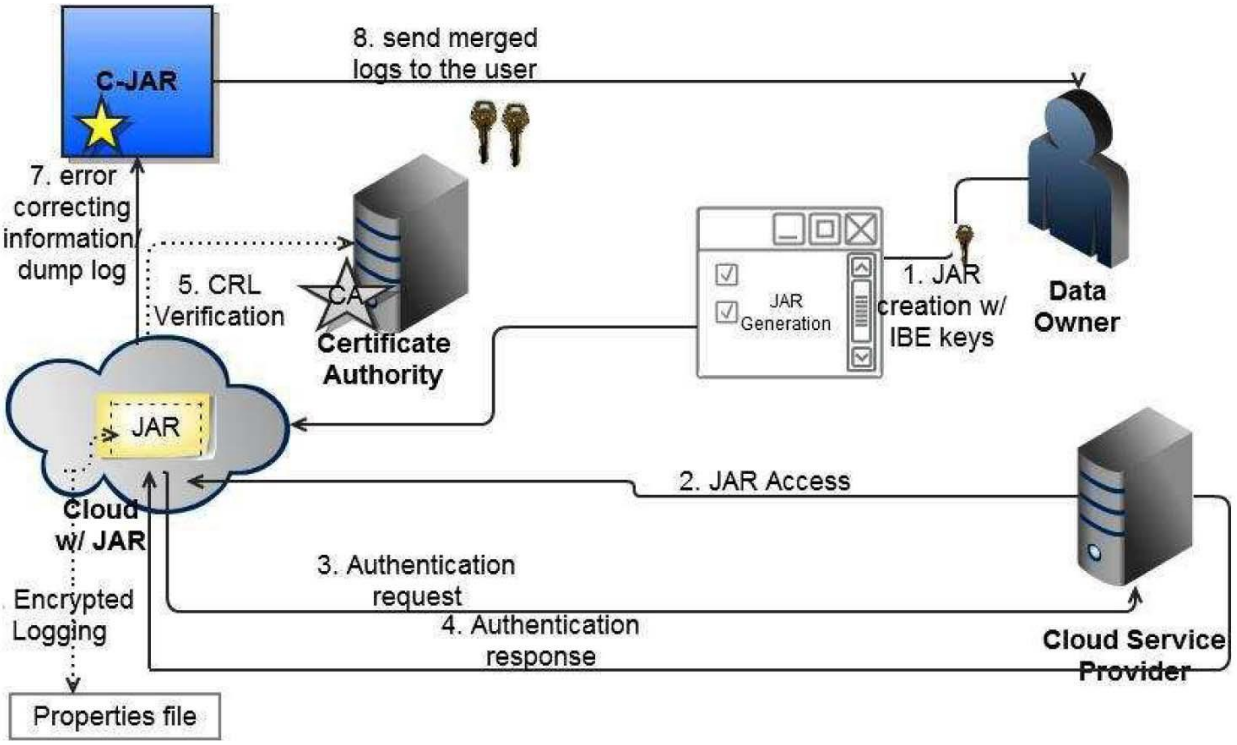


Fig 1 CLOUD INFORMATION ACCOUNTABILITY FRAMEWORK

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. Verifying the authenticity of data has emerged as a critical issue in storing data on untrusted servers.

CIA prevents storage servers from misrepresenting or modifying data by providing authenticity checks when accessing data. However, archival storage requires guarantees about the authenticity of data on storage, namely that storage servers possess data. It is insufficient to detect that data have been modified or deleted when accessing the data, because it may be too late to recover lost or damaged data. Archival storage servers retain tremendous amounts of data, little of which are accessed.

They also hold data for long periods of time during which there may be exposure to data loss from ad- ministration errors as the physical implementation of storage evolves, e.g., backup and restore, data migration to new systems, and changing memberships in peer-to-peer systems. Archival network storage presents unique performance demands. Reading an entire archive, even periodically, greatly limits the scalability of network stores. (The growth in storage capacity has far outstripped the growth in storage access times and bandwidth). they can achieve cost savings and productivity enhancement by using cloud-based services to manage projects, to make collaborations Furthermore, I/O incurred to establish data possession interferes with on-demand band- width to store and retrieve data. Normally data owners and service providers are not in the same trusted domain in cloud computing. Enterprises usually store data in internal storage and install firewalls to protect against intruders to access the data.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

III.DEVELOPING A CLOUD ENVIRONMENT

Initially the basic network model for the cloud data storage is developed in this module. Four different network entities can be identified as follows: Client(Data Owner): an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations; Cloud Storage Server (CSS): an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data; Certificate Authority: an entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

The log harmonizer forms the central component which allows the user access to the log files. The logger is strongly coupled with user's data (either single or multiple data items). Its main tasks include 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 may also be configured to ensure that access and usage control policies associated with the data are honored. The logger is also responsible for generating the error correction information for each log record and sends the same to the log harmonizer. The error correction information combined with the encryption and authentication mechanism provides a robust and reliable recovery mechanism, therefore meeting the third requirement. The log harmonizer is responsible for auditing. Being the trusted component, the log harmonizer generates the master key. It holds on to the decryption key for the IBE key pair, as it is responsible for decrypting the logs. The log harmonizer is also responsible for handling log file corruption. In addition, the log harmonizer can itself carry out logging in addition to auditing. Separating the logging and auditing functions improves the performance.

IV.PROPOSING A CLOUD INFORMATION ACCOUNTABILITY (CIA) FRAMEWORK

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 components: logger and log harmonizer. The logger is the component which is strongly coupled with the user's data, so that it is downloaded when the data are accessed, and is copied whenever the data are copied. It handles a particular instance or copy of the user's data and is responsible for logging access to that instance or copy.

V.DATA ACCESS IN CLOUD INFORMATION ACCOUNTABILITY

The development of JAR file includes a set of simple access control rules specifying whether and how the cloud servers and possibly other data owners 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, we use CA wherein a trusted certificate authority certifies the CSP. In the event that the access is requested by a user, we employ authentication, wherein a trusted identity provider issues certificates verifying the user's identity based on his username.

Once the authentication succeeds, the user will be allowed to access the data enclosed in the JAR. 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. As for the logging, 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.

VI.DEVELOP AUTOMATED LOGGING MECHANISM

In the automated logging mechanism, the main responsibility of the JAR is to handle authentication of entities which want to access the data stored in the JAR file. In our context, the data owners may not know the exact CSPs that are going to handle the data. Hence, authentication is specified according to the servers' functionality. Each JAR contains the encrypted data, class files to facilitate retrieval of log files and display enclosed data in a suitable format, and a log file for each encrypted item.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

If the access request is granted, the JAR will additionally record the access information along with the duration for which the access is allowed. In the current system, we support four types of actions, i.e., Act has one of the following four values: view, download, timed-access, and Location-based access. For each action, we propose a specific method to correctly record or enforce it depending on the type of the logging module. And finally we design the end to end mechanism for auditing using push or pull mode configure for log files send to the data owner.

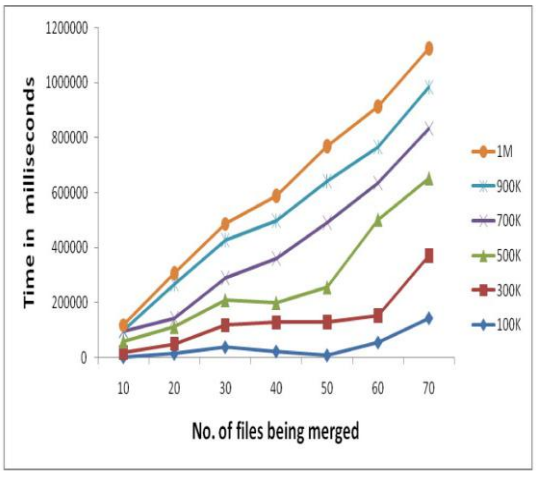
VII.PERFORMANCE EVALUATION

The performance can be examined by the time taken to create a log file and then measure the overhead in the system. With respect to time, the overhead can occur at three points: during the authentication, during encryption of a log record, and during the merging of the logs.

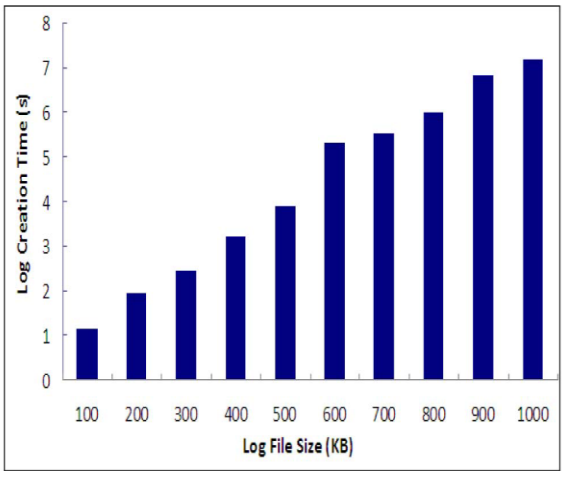
Also, with respect to storage overhead, we notice that our architecture is very lightweight, in that the only data to be stored are given by the actual files and the associated logs. We can evaluate our performance by the following parameters they are Log Creation Time, Time Taken to Perform Logging, and Log Merging Time Size of the Data JAR Files. If the access request is granted, the JAR will additionally record the access information along with the duration for which the access is allowed. Each inner JAR contains the encrypted data, class files to facilitate retrieval of log files and display enclosed data in a suitable format, and a log file for each encrypted item.

VIII.CONCLUSION AND FUTURE ENHANCEMENT

An object-centered approach that enables enclosing the logging mechanism together with users' data and policies, automatically logging any access to the data in the cloud together with an auditing mechanism and it allows the data owner to not only audit his content but also enforce strong back-end protection if needed. The data owner to audit even those copies of its data that were made without his knowledge and user's control also provide distributed auditing mechanisms the efficiency and effectiveness of the proposed approaches. We can evaluate our performance by the following parameters they are Log Creation Time, Time Taken to Perform Logging, Log Merging Time Size of the Data JAR Files.



Time to merge log files.



Time to create log files of different sizes

In the future, the approach can be refined to verify the integrity of the JRE and the authentication of JARs. . This research is aimed at providing software tamper resistance to Java applications. CIA can support a variety of security policies, like indexing policies for text files, usage control for executables, and generic accountability and provenance controls.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

REFERENCES

1. Ammann.P and Jajodia .S, "Distributed Timestamp Generation in Planar Lattice Networks," ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.
2. Ateniese.G, R., Burns.R, Curtmola,.J, Herring. L, Kissner,.Z. Peterson.K and Song.D, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, pp. 598- 609, 2007
3. Chun.B and Bavier.A.C, "Decentralized Trust Management and Accountability in Federated Systems," Proc. Ann. Hawaii Int'l Conf. System Sciences (HICSS), 2004
4. Corin.R, Etalle.S, Hartog.F, Lenzini.G, and Staicu.H, "A Logic for Auditing Accountability in Decentralized Systems," Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005
5. Hightower.J and Borriello.G, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57-66, Aug. 2001
6. Jaeger.P, Lin.T, and Grimes.J, "Cloud Computing and Information Policy: Computing in a Policy Cloud?," J. Information Technology and Politics, vol. 5, no. 3, pp. 269-283, 2009.
7. Mont.M.C, Pearson.K, and Bramhall.O, "Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services," Proc. Int'l Worksh
8. Pretschner.A, Hilty.M, and Basin.D, "Distributed Usage Control," Comm. ACM, vol. 49, no. 9, pp. 39-44, Sept. 2006op Database and Expert Systems Applications (DEXA), pp. 377-382, 2003.
9. 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.
10. Wang.Q, Wang.Q, Li.I, Ren.K, and Lou.K, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. European Conf. Research in Computer Security (ESORICS), pp. 355-370, 2009.