

DATA SHARING IN IT ENTERPRISES USING CLOUD COMPUTING

R.GNANAJEYARAMAN*

S.KARTHIKRAJA *

Abstract

Cloud computing has been envisioned as the de-facto solution to the rising storage costs of IT Enterprises. With the high costs of data storage devices as well as the rapid rate at which data is being generated it proves costly for enterprises or individual users to frequently update their hardware. Apart from reduction in storage costs data outsourcing to the cloud also helps in reducing the maintenance. Cloud storage moves the user's data to large data centers, which are remotely located, on which user does not have any control. However, this unique feature of the cloud poses many new security challenges which need to be clearly understood and resolved. We provide a scheme which gives a proof of data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This proof can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA).

* Asso.Prof, Dept. of CSE, SBM College of Engg. & Tech.

I. INTRODUCTION

In cloud computing we presents a new way to supplement the current consumption and delivery model for IT services based on the Internet, by providing for dynamically scalable and often virtualized resources as a service over the Internet. To date, there are a number of notable commercial and individual cloud computing services, including Amazon, Google, Microsoft, Yahoo, and Salesforce [19]. 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 their data. While enjoying the convenience brought by this new technology, users also start worrying about losing control of their own data. The data processed on clouds are often outsourced, leading to a number of issues related to accountability, including the handling of personally identifiable information. Such fears are becoming a significant barrier to the wide adoption of cloud services [3].

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. First, data handling can be outsourced by the direct cloud service provider (CSP) to other entities in the cloud and these 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.

II. BASIC SURVEY

As data generation is far outpacing data storage it proves costly for small firms to frequently update their hardware whenever additional data is created. Also maintaining the storages can be a difficult task. It transmitting the file across the network to the client can consume heavy bandwidths. The problem is further complicated by the fact that the owner of the data may be a small device, like a PDA (personal digital assist) or a mobile phone, which have limited CPU power, battery power and communication bandwidth. 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. The disadvantage of the existing is: The main drawback of this scheme is the high resource costs it requires for the implementation. Also computing hash value for even a moderately large data files can be computationally burdensome for some clients (PDAs, mobile phones, etc). Data encryption is large so the disadvantage is small users with limited computational power (PDAs, mobile phones etc.).

III. PROPOSED SYSTEM:

To address this problem, in this paper, 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. We leverage the XML programmable capabilities 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 XMLs. To strengthen user's control, we also provide distributed auditing mechanisms. We provide extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches.

One of the important concerns that need to be addressed is to assure the customer of the integrity i.e. correctness of his data in the cloud. As the data is physically not accessible to the user the cloud should provide a way for the user to check if the integrity of his data is maintained or is compromised. In this paper we provide a scheme which gives a proof of data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This proof can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA). It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted.

The advantage of the proposed system is:

1. Apart from reduction in storage costs data outsourcing to the cloud also helps in reducing the maintenance.
2. Avoiding local storage of data.
3. By reducing the costs of storage, maintenance and personnel.
4. It reduces the chance of losing data by hardware failures.

5. Not cheating the owner.

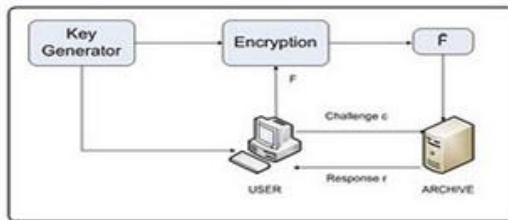


Fig:Architecture Diagram

AUTOMATED LOGGING MECHANISM

In this section, we first elaborate on the automated logging mechanism and then present techniques to guarantee dependability.

The Logger Structure

The programmable capability of JARs to conduct automated logging. A logger component is a nested Java JAR file which stores a user's data items and corresponding log files. As shown in Fig. 2, our proposed JAR file consists of one outer JAR enclosing one or more inner JARs.

JARs Availability

To protect against attacks perpetrated on offline JARs, the CIA includes a log harmonizer which has two main responsibilities: to deal with copies of JARs and to recover corrupted logs. Each log harmonizer is in charge of copies of logger components containing the same set of data items. The harmonizer is implemented as a JAR file. It does not contain the user's data items being audited, but consists of class files for both a server and a client processes to allow it to communicate with its logger components.

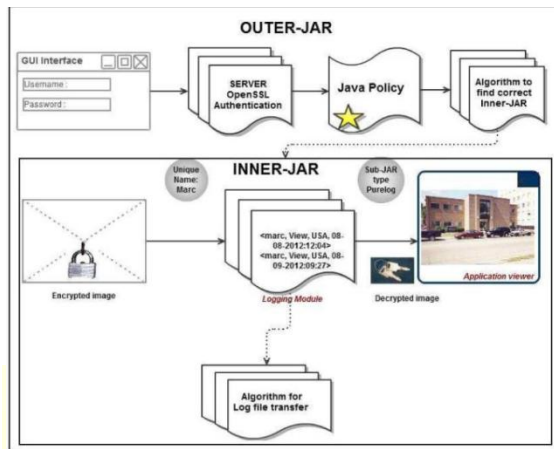


Fig. 2. The structure of the JAR file.

The main responsibility of the outer JAR is to handle authentication of entities which want to access the data stored in the JAR file.

Log Record Generation:

The Log records are generated by the logger component.

Logging occurs at any access to the data in the JAR,

and new log entries are appended sequentially, in order

of creation $LR \frac{1}{4} hr_1; \dots; r_{k_i}$. Each record r_i is encrypted

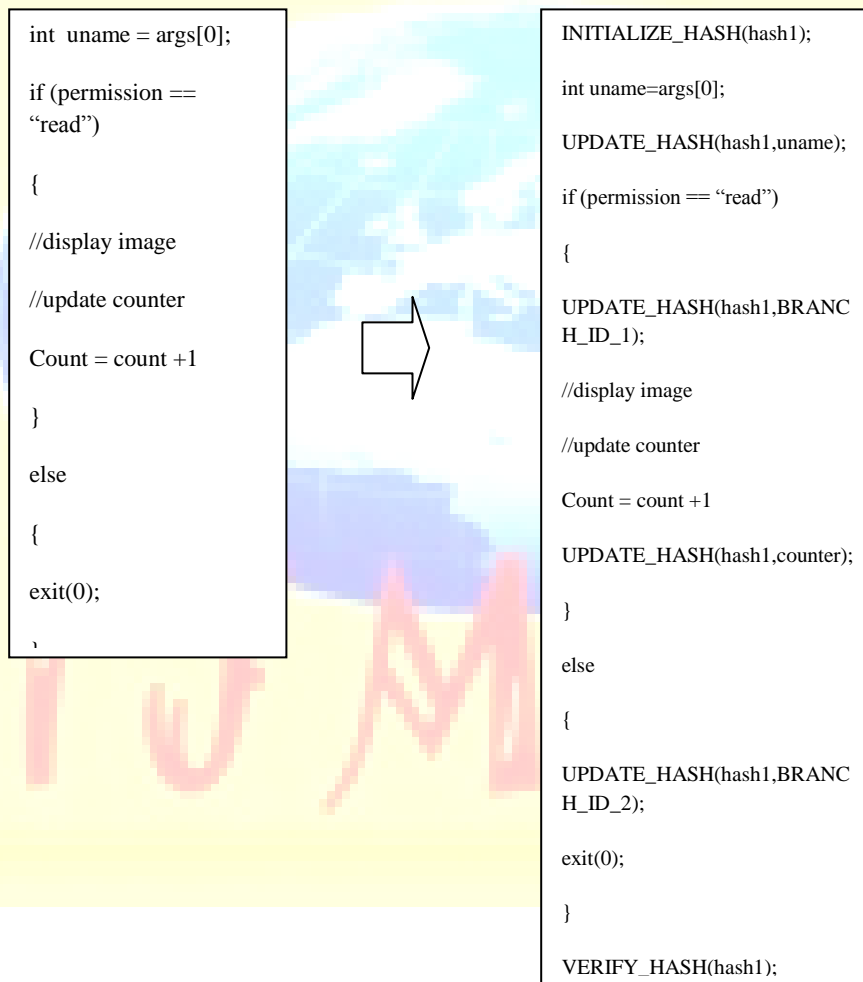
individually and appended to the log file. Here, r_i indicates that an entity identified by ID has performed an action Act on the user's data at time T at location Loc. The component corresponds to the checksum of the records preceding the newly inserted one, concatenated with the main content of the record itself (we use I to denote concatenation). The checksum is computed using a collision-free hash function. The component sig denotes the signature of the record created by the server.

Dependability of Logs

In this section, how we are going to ensure the dependability of logs. In particular, we aim to prevent the following two types of attacks. First, an attacker may try to evade the auditing mechanism by storing the JARs corrupting the JAR, or trying to prevent them from communicating with the user. Second, the attacker may try to compromise the JRE used to run the JAR files.

Log Correctness

For logs to be correctly recorded, it is essential that the JRE of the system on which the logger components are running remain unmodified. To verify the integrity of the logger component, we rely on a two-step process: 1) we repair the JRE before the logger is launched and any kind of access is given, so as to provide guarantees of integrity of the JRE. 2) We insert hash codes, which calculate the hash values of the program traces of the modules being executed by the logger component.



Original code

Hashed code

Hashed code

Fig. 3. Obvious hashing applied to the logger.

IV. MODULES

1. Cloud Information Accountability (CIA) Framework.
2. Distinct mode for auditing
3. Logging and auditing Techniques
4. Major components of CIA

Cloud Information Accountability (CIA) Framework:

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.

Distinct mode for auditing:

Push mode:

The push mode refers to logs being periodically sent to the data owner or stakeholder.

Pull mode:

Pull mode refers to an alternative approach whereby the user (Or another authorized party) can retrieve the logs as needed.

Logging and auditing Techniques:

1. 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.
2. 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.
3. 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.
4. Log files should be sent back to their data owners periodically to inform them of the current usage of their data. More importantly, log files should be retrievable anytime by their data owners when needed regardless the location where the files are stored.

5. 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.

Major components of CIA

There are two major components of the CIA, the first being the logger, and the second being the log harmonizer. 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. For example, a data owner can specify that user X is only allowed to view but not to modify the data. The logger will control the data access even after it is downloaded by user X. The log harmonizer forms the central component which allows the user access to the log files. The log harmonizer is responsible for auditing.

V. CONCLUSION

In the cloud we proposed innovative approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. This approach allows the data owner to not only audit his content but also enforce strong back-end protection if needed. Moreover, one of the main features of our work is that it enables the data owner to audit even those copies of its data that were made without his knowledge. In the future, we plan to refine our approach to verify the integrity of the JRE and the authentication of JARs [1].

VI.REFERECENES

- [1].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.
- [2].A. Pretschner, M. Hilty, F. Schuster, C. Schaefer, and T. Walter, "Usage Control Enforcement: Present and Future," IEEE Security & Privacy, vol. 6, no. 4, pp. 44-53, July/Aug. 2008.
- [3].A. Pretschner, F. Schuster, C. Schaefer, and T. Walter, "Policy Evolution in Distributed Usage Control," Electronic Notes Theoretical Computer Science, vol. 244, pp. 109-123, 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].A. Squicciarini, S. Sundareswaran, and D. Lin, "Preventing Information Leakage from Indexing in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing,2010.
- [6] .S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang, "Promoting Distributed Accountability in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing,2011.
- [7] .Eucalyptus Systems, <http://www.eucalyptus.com/>, 2012.
- [8].Emulab Network Emulation Testbed, www.emulab.net, 2012.