Middle-East Journal of Scientific Research 20 (6): 702-704, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.20.06.11389

Ensuring Distributed Accountability for Data Sharing in Cloud

K. Karthick, P. Jennifer and A. Muthukumaravel

Department of MCA, Bharath University,
Selaiyur, Chennai-73, Tamil Nadu, India

Abstract: 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. 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 JAR 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 JARs. 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.

Key words: Cloud computing enables · Emerging technology · Programmable capabilities · Trigger authentication

INTRODUTION

Project Description: CLOUD computing 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 [1-3]. To date, there are a number of notable commercial and individual cloud computing services, including Amazon, Google, Microsoft, Yahoo and Sales force. Details of the services provided are abstracted from the users who no longer need to be experts of technology infrastructure [4-6]. 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.To allay users' concerns [7] 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. 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.

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.

Corresponding Author: K. Karthick, Department of MCA, Bharath University, Selaiyur, Chennai-73, Tamil Nadu, India.

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-toned 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 [8]. 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.

System 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 [9].

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.

Modules Description:

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 [10].

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:

- 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 [10].
- 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.
- 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.

CONCLUSION

We proposed innovative 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. 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. For example, we will investigate whether it is possible to leverage the notion of a secure JVM being developed by IBM. This research is aimed at providing software tamper resistance to dot net applications. In the long term, we plan to design a comprehensive and more generic object-oriented approach to facilitate autonomous protection of traveling content. We would like to support a variety of security policies, like indexing policies for text files, usage control for executables and generic accountability and provenance controls.

Future Enhancements: For future work, we aim to provide a framework to supply a secure cloud database that will guarantee to prevent security risks facing the cloud computing community. This framework will apply multi-clouds and the secret sharing algorithm to reduce the risk of data intrusion and the loss of service availability in the cloud and ensure data integrity. In relation to data intrusion and data integrity, assume we want to distribute the data into three different cloud providers and we apply the secret sharing algorithm on the stored data in the cloud provider. An intruder needs to retrieve at least three values to be able to find out the real value that we want to hide from the intruder. This depends on Shamir's secret sharing algorithm with a polynomial function technique which claims that even with full knowledge of (k – 1) clouds, the service provider will not have any knowledge of vs (vs is the secret value). We have used this technique in previous databases-as-a-serves research [5]. In other words, hackers need to retrieve all the information from the cloud providers to know the real value of the data in the cloud. Therefore, if the attacker hacked one cloud provider's password or even two cloud provider's passwords, they still need to hack the third cloud provider (in the case where k = 3) to know the secret which is the worst case scenario. Hence, replicating data into multi-clouds by using a multi-share technique [5] may reduce the risk of data intrusion and increase data integrity. In other words, it will decrease the risk of the Hyper-Visor being hacked and Byzantine fault-tolerant data being stolen from the cloud provider. Regarding service availability risk or loss of data, if we replicate the data into different cloud providers, we could argue that the data loss risk will be reduced. If one cloud provider fails, we can still access our data live in other cloud providers. This fact has been discovered from this survey and we will explore dealing with different cloud provider interfaces and the network traffic between cloud providers.

REFERENCES

1. Ammann, P. and S. Jajodia, 1993. Distributed Timestamp Generation in Planar Lattice Networks, ACM Trans. Computer Systems, 11: 205-225.
2. Ateniese, G., R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, 2007. Provable Data Possession at Untrusted Stores, Proc. ACM Conf. Computer and Comm. Security, pp: 598-609.
3. Barka, E. and A. Lakas, 2008. Integrating Usage Control with SIP-Based Communications," J. Computer Systems, Networks and Comm., pp: 1-8.
4. Boneh, D. and M.K. Franklin, 2001. Identity-Based Encryption from the Weil Pairing, Proc. Int'l Cryptology Conf. Advances in Cryptology, pp: 213-229.
5. Bose, R. and J. Frew, 2005. Lineage Retrieval for Scientific Data Processing: A Survey, ACM Computing Surveys, 37: 1-28.
6. Kerana Hanirex, D. and K.P. Kaliyamurthie, 2013. Multi-classification approach for detecting thyroid attacks, International Journal of Pharma and Bio Sciences, 4(3): B1246-B1251.
7. Khanaa, V., K. Mohanta and T. Saravanan, 2013. Comparative study of uwb communications over fiber using direct and external modulations, Indian Journal of Science and Technology, 6 (suppl 6): 4845-4847.
8. Kumar Giri, R. and M. Saikia, 2013. Multipath routing for admission control and load balancing in wireless mesh networks, International Review on Computers and Software, 8(3): 779-785.
9. Kumaravel, A. and K. Rangarajan, 2013. Routing alogrithm over semi-regular tessellations, 2013 IEEE Conference on Information and Communication Technologies, ICT 2013.
10. Kumaravel, A. and K. Rangarajan, 2013. Algorithm for automaton specification for exploring dynamic labyrinths, Indian Journal of Science and Technology, 6(suppl 6).