Proficiently Organizing the Protection & Integrity of Users Data in a Cloud Computing

[1]Dinari Ravi Kumar, [2]Goda.Srinivasarao
[1,2]Dept. of CSE, Krishnachaitanya Institute of Technology and Sciences, Markapur, India

Abstract

Cloud computing requires companies and individuals to transfer some or all control of computing resources to Cloud Service Providers (CSPs). Such transfers naturally pose concerns for company decision makers. In a recent 2010 survey by Fujitsu Research Institute [1] on potential cloud customers, it was found that 88% of potential cloud consumers are worried about who has access to their data, and demanded more awareness of what goes on in the backend physical server.

To overcome the above problems, we propose a novel approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability, apart from this ARIES algorithm is used to recover loging information.

Keywords

Cloud Computing, Accountability, Data Sharing

I. Introduction

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing is a means by which highly scalable, technology-enabled services can be easily consumed over the Internet on an as-needed basis [1]. The convenience and efficiency of this approach, however, comes with privacy and security risks [2]. A significant barrier to the adoption of cloud services is thus user fear of confidential data leakage and loss of privacy in the cloud. Furthermore, the cross-jurisdictional nature of clouds presents a new challenge in maintaining the data protection required by current legislation including restrictions on cross-border data transfer.

At the broadest level, privacy is a fundamental human right that encompasses the right to be left alone, although an analysis of the term is complex [3]. In the commercial, consumer context, privacy entails the protection and appropriate use of the personal information of customers, and the meeting of expectations of customers about its use. For organizations, privacy entails the application of laws, policies, standards and processes by which Personally Identifiable Information (PII) of individuals is managed.

We can 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. 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.

To overcome the above problems, we propose a new approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability. 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.

II. Existing System

Pearson et al. have proposed accountability mechanisms to address privacy concerns of end users [9] and then develop a privacy manager [10]. Their basic idea is that the user's private data are sent to the cloud in an encrypted form, and the processing is done on the encrypted data. The privacy manager to reveal the correct result. However, the privacy manager provides only limited features in that it does not guarantee protection once the data are being disclosed.

In [7], the authors present a layered architecture for addressing the end-to-end trust management and accountability problem in federated systems. The authors' focus is very different from ours, in that they mainly leverage trust relationships for accountability, along with authentication and anomaly detection. Further, their solution requires third-party services to complete the monitoring and focuses on lower level monitoring of system resources.

III. Proposed System

Example for Identifying Problem:

Ram, is a professional Writer, plans to sell his stories by using the Mediafire Cloud Services. For her business in the cloud, she has the following requirements:

- His stories are downloaded only by users who have paid for her services.
- Potential buyers are allowed to view his stories first before they make the payment to obtain the download right.
- Due to the nature of some of his works, only users from certain countries can view or download some sets of photographs.
- For some of his works, users are allowed to only view them for a limited time, so that the users cannot reproduce his work easily.
- In case any dispute arises with a client, he wants to have all the access information of that client.
- He wants to ensure that the cloud service providers of

Mediafire do not share his data with other service providers, so that the accountability provided for individual users can also be expected from the cloud service providers.

According to above scenario in mind, we identify the common requirements and develop several guidelines to achieve data accountability in the cloud. A user who subscribed to a certain cloud service, usually needs to send his/her data as well as associated access control policies (if any) to the 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. Using conventional access control mechanisms, once the access rights are granted, the data will be fully available at the service provider.

We aim to develop novel logging and auditing techniques which satisfy the following requirements:

- 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.
- Log files should be sent back to their data owners periodically to inform them of the current usage of their data.

There are two main components of the CIA, the first being the logger, and the second being the 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. The log harmonizer forms the central component which allows the user access to the log files.

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

A. CIA Framework Architecture

- The overall CIA framework, combining data, users, logger and harmonizer is sketched in fig. 1.
- At the beginning, each user creates a pair of public and private keys based on Identity-Based Encryption. This IBE scheme is a Weil-pairing-based IBE scheme, which protects us against one of the most prevalent attacks to our architecture. Using the generated key, the user will create a logger component which is a JAR file, 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 (users, companies) 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 OpenSSLbased certificates, wherein a trusted certificate authority certifies the CSP. In the event that the access is requested by a user, we employ SAML-based authentication [8], wherein a trusted identity provider issues certificates verifying the user's identity based on his username.
- Once the authentication succeeds, the service provider (or 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.
- The data owner could opt to reuse the same key pair for all JARs or create different key pairs for separate JARs. Using 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 possible log file corruption.
- To ensure trustworthiness of the logs, each record is signed by the entity accessing the content. Further, individual records are hashed together to create a chain structure, able to quickly detect possible errors or missing records.
- The encrypted log files can later be decrypted and their integrity verified. They can be accessed by the data owner or other authorized stakeholders at any time for auditing purposes with the aid of the log harmonizer.

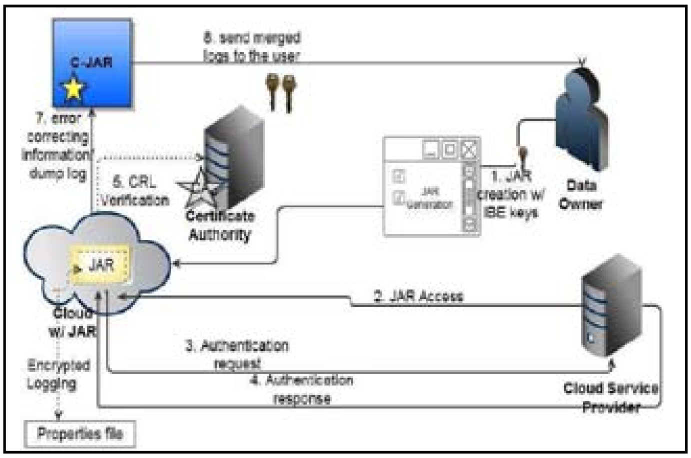
C-JAR
8. send merged logs to the user
7. error correcting information/ dump log
5. CRL Verification
Certificate Authority
JAR Generator
1. JAR creation w/ IBE keys
Data Owner
JAR
Cloud w/ JAR
2. JAR Access
3. Authentication request
4. Authentication response
Encrypted Logging
Cloud Service Provider
Properties file

Fig. 1: CIA Framework Architecture

B. Logging Recovery Procedure – Aries Algorithm

For the ARIES algorithm to work a number of log records have to be created during the operation of the database. Log entries are sequentially numbered with Sequence Numbers.

Usually the resulting log file is stored on so-called "stable storage", that is a storage medium that is assumed to survive crashes and hardware failures. To gather the necessary information for the logging two data structures have to be maintained: the Dirty Page Table (DPT) and the Transaction Table (TT).

The dirty page table keeps record of all the pages that have been modified and not yet written back to disc and the first Sequence Number that caused that page to become dirty. The transaction table contains all transactions that are currently running and the Sequence Number of the last log entry they caused.

We create log records of the form (Sequence Number, Transaction ID, Page ID, Redo, Undo, Previous Sequence Number). The Redo and Undo fields keep information about the changes this log record saves and how to undo them. The Previous Sequence Number is a reference to the previous log record that was created for this transaction. Using the Previous Sequence Numbers it is for example possible to traverse to log file in reverse order and undo all the actions of a specific transaction in case it aborts.

Every time a transaction begins or commits we write a Begin

Transaction entry or, respectively, an End Of Log entry for that transaction.

C. Recovery

The recovery works in three phases. The first phase, Analysis, computes all the necessary information from the log file. The Redo phase restores the database to the exact state at the crash, including all the changes of uncommited transactions that were running at that point in time. The Undo phase then undoes all uncommited changes, leaving the database in a consistent state.

D. Analysis

During the Analysis phase we restore the DPT and the TT as they were at the time of the crash. We run through the logfile (from the beginning or the last checkpoint) and add all transactions for which we encounter Begin Transaction entries to the TT. Whenever an End Log entry is found, the corresponding transaction is removed. The last Sequence Number for each transaction is of course also maintained. During the same run we also fill the dirty page table by adding a new entry whenever we encounter a page that is modified and not yet in the DPT. This however only computes a superset of all dirty pages at the time of the crash, since we don't check the actual database file whether the page was written back to the storage.

E. Redo

From the DPT we can compute the minimal Sequence Number of a dirty page. From there we have to start redoing the actions until the crash, in case they weren't persisted already.

Running through the log file we check for each entry whether the modified page is in the DPT table and whether the Sequence Number in the DPT is smaller than the Sequence Number of the record (i.e. whether the change in the log is newer than the last version that was persisted). If it is we fetch the page from the database storage and check the Sequence Number on the actual if it is smaller than the Sequence Number on the log record. That check is necessary because the recovered DPT is only a conservative superset of the pages that really need changes to be reapplied. Lastly we reapply the redo action and store the new Sequence Number on the page. It is also important for recovery from a crash during the Redo phase, as the redo isn't applied twice to the same page.

F. Undo

After the Redo phase the database reflects the exact state at the crash. However the changes of uncommited transactions have to be undone to restore the database to a consistent state.

For that we run backwards through the log for each transaction (those runs can of course be combined into one) using the Previous Sequence Number fields in the records. For each record we undo the changes (using the information in the Undo field) and write a compensation log record to the log file. If we encounter a Begin Transaction record we write an End Log record for that transaction.

The compensation log records make it possible to recover during a crash that occurs during the recovery phase. That isn't as uncommon as one might think, as it is possible for the recovery phase to take quite long. CLRs are read during the Analysis phase and redone during the Redo phase.

IV. Conclusion

The key barrier to widespread uptake of cloud computing is the lack of trust in clouds by potential customers. While preventive controls for security and privacy measures are actively being researched, there is still little focus on detective controls related to cloud accountability and audit ability. 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.

References

[1] P.Ammann, S.Jajodia,"Distributed Timestamp Generation in Planar Lattice Networks", ACM Trans. Computer Systems, Vol. 11, pp. 205-225, Aug. 1993.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song,"Provable Data Possession at Untrusted Stores", Proc. ACM Conf. Computer and Comm. Security, pp. 598- 609, 2007.

[3] E. Barka, A. Lakas,"Integrating Usage Control with SIP-Based Communications", J. Computer Systems, Networks, and Comm., Vol. 2008, pp. 1-8, 2008.

[4] D. Boneh, M.K. Franklin,"Identity-Based Encryption from the Weil Pairing", Proc. Int'l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001.

[5] R. Bose, J. Frew,"Lineage Retrieval for Scientific Data Processing: A Survey", ACM Computing Surveys, Vol. 37, pp. 1-28, Mar. 2005.

[6] P. Buneman, A. Chapman, J. Cheney,"Provenance Management in Curated Databases", Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '06), pp. 539-550, 2006.

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

[8] OASIS Security Services Technical Committee,"Security Assertion Markup Language (saml) 2.0", [Online] Available: http://www.oasis-open.org/committees/tc home.php?wg abbrev=security, 2012.

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

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

Dinari Ravi Kumar received his B.Tech degree in Civil Engineering From J.N.T.U.College of Anantapur, AP, in 1996. The m.tech. Degree in CSE From Krishna Chaitanya Institute of Science & Technologies Markapuram Prakasam (Dt), in 2013. At present, He is engaged in"Proficiently Organizing the Protection & Integrity of Users Data in a Cloud Computing "

GODA.SRINIVASARAO received the MTech degree from K L C E (Koneru Lakshmaiah College Of Engineering), Guntur, AP in 2009. Currently he is working as Professor & HOD IN CSE Dept In Krishna Chaitanya Institute Of Science & Technologies Markapuram Prakasam(Dt) Andhra Pradesh, India. He has 12 years of experience in Teaching. Previously he has worked with Rise Group of Institutions at Ongole as a Associate Professor., His research interest are is cloud computing.