IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

A New Method for Object Management in Cloud Computing Application

K Venkata Chalapathi[1], Hemalatha T[2]

Computer Science and Engineering, S V U College Engineering, Tirupati, India[1,2]

Abstract: Cloud computing presents a new way to development the present use and release model for IT services based on the Internet, by providing for actively scalable and often virtualized assets as a service over the Internet. It enables extremely scalable services to be with no trouble consumed over the Internet on an as-needed basis. A major feature of the cloud services is that user's data are normally processed vaguely in unknown machines that users do not own or operate. While enjoying the skill brought by this new budding technology, we suggest a new really decentralized in series responsibility structure to keep path of the real practice of the user's data in the cloud. In particular, we offer an object-centered move toward that enables enclosing our sorting mechanism jointly with user's data and policies. We leverage the sorting mechanism to both create a active and roaming object, and to ensure that any access to user's data will start validation and automated sorting. To make stronger user's control, we also present spread audit mechanisms. We provide broad trial studies that show the good organization and success of the planned out come.

Keywords: cloud computing, authentication, logging, audit.

I. INTRODUCTION

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. To date, there are a number of notable commercial and individual cloud computing services, including Amazon, Google. 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. To allay users' concerns, it is essential to provide an effective mechanism for users to monitor the usage of their data 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, propose a novel approach, namely Cloud Information Accountability (CIA) framework [1], based on the notion of information accountability.

II. RELATED WORK

Cloud computing is a computing platform that resides in a large data centre and is able to dynamically provide servers the ability to address a wide range of needs, ranging from scientific research to e-commerce. The provision of computing resources as if it were a utility such as electricity, while potentially revolutionary as a computing service, presents many major problems of information policy, including issues of privacy, security, reliability, access, and regulation. This paper explores the nature and potential of cloud computing, the policy issues raised, and research questions related to cloud computing and policy. Ultimately, the policy issues raised by cloud computing are examined as a part of larger issues of public policy attempting to respond to rapid technological evolution. The issue of how to provide appropriate privacy protection for cloud computing is important, and as yet unresolved. In this paper we propose an approach in which procedural and technical solutions are co-designed to demonstrate accountability [2] as a path forward to resolving jurisdictional privacy and security risks within the cloud.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

III. PROPOSED WORK

Here proposed CIA framework provides end-to end accountability [3] 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.

In this basic design approach I would like to describe functionalities of data owner, cloud server, users. Owner uploads the encrypted files to cloud server. Authentication [4,8] is performed by the user. Verify the users and what files are exist in the owner side all the files information and users information is also available.

Cloud is the server inside this server all the encrypted [5,7] files are available users search on this cloud server information. In the side of cloud server we can see the file but we can't copy the data or file .User can access the cloud server information first he has to made subscription for the file and then by using the subscription details file will be accessed to the user.

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. User interface design or user interface engineering is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design may be utilized to support its usability [6]. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered around their expertise, whether that be software design, user research, web design, or industrial design.

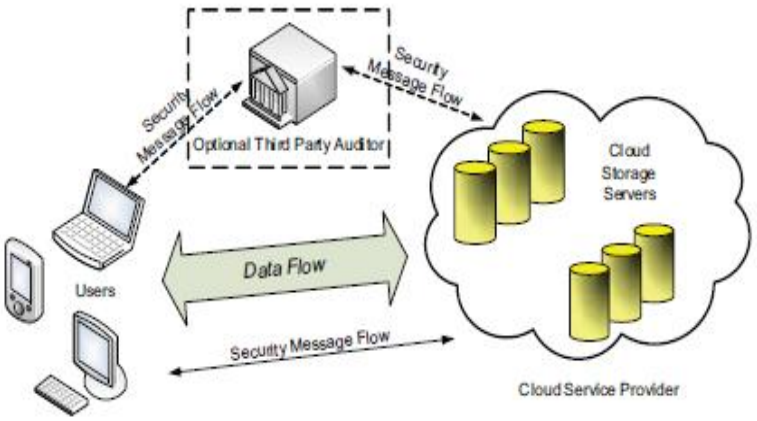

Fig1: Cloud Service framework

IV. MODULES

VIEW:
The entity (e.g., the cloud service provider) can only read the data but is not allowed to save a raw copy of it anywhere permanently. For this type of action, the Pure Log will simply write a log record about the access, while the Access Logs will enforce the action through the enclosed access control module. Recall that the data are encrypted and stored in the inner JAR. When there is a view-only access request, the inner JAR will decrypt the data on the fly and create a temporary decrypted file. The decrypted file will then be displayed to the entity using the Java application viewer in case the file is displayed to a human user. Presenting the data in the Java application, viewer disables the copying functions using right click or other hot keys such as Print Screen. Further, to prevent the use of some screen capture

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

software, the data will be hidden whenever the application viewer screen is out of focus. The content is displayed using the headless mode in Java on the command line when it is presented to a CSP.

DOWNLOAD:
The entity is allowed to save a raw copy of the data and the entity will have no control over this copy neither log records regarding access to the copy. If Pure Log is adopted, the user's data will be directly downloadable in a pure form using a link. When an entity clicks this download link, the JAR file associated with the data will decrypt the data and give it to the entity in raw form. In case of Access Logs, the entire JAR file will be given to the entity. If the entity is a human user, he/she just needs to double click the JAR file to obtain the data. If the entity is a CSP, it can run a simple script to execute the JAR.

TIME ACCESS:
This action is combined with the view-only access, and it indicates that the data are made available only for a certain period of time. The Pure log will just record the access starting time and its duration, while the Access Log will enforce that the access is allowed only within the specified period of time. The duration for which the access is allowed is calculated using the Network Time Protocol. To enforce the limit on the duration, the Access Log records the start time using the NTP, and then uses a timer to stop the access. Naturally, this type of access can be enforced only when it is combined with the View access right and not when it is combined with the Download.

V. EXPERIMENTAL RESULTS

In the experiments, we first examine 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. Further, JAR act as a compressor of the files that it handles. In particular, as introduced in Section 3, multiple files can be handled by the same logger component. To this extent, we investigate whether a single logger component, used to handle more than one file, results in storage overhead.

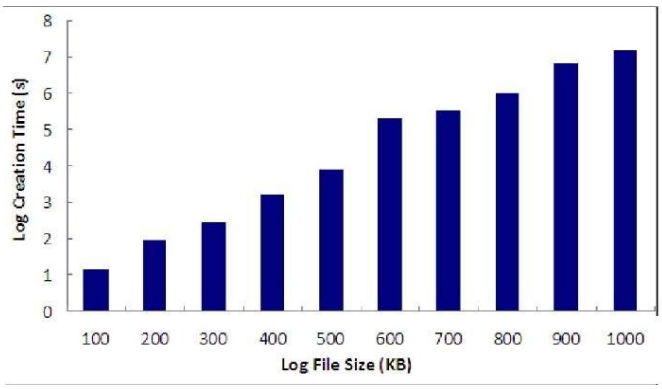

Fig 2: Time to create log files of different sizes

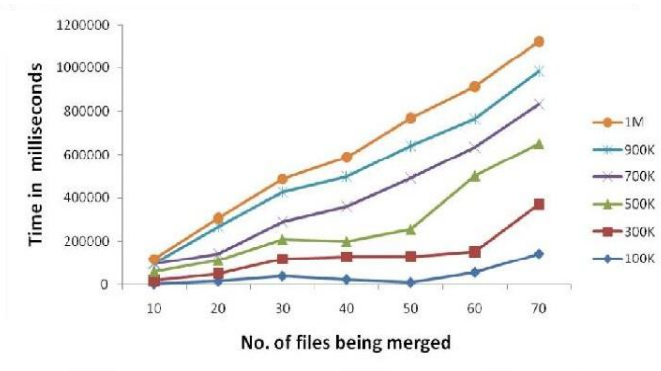

Fig 3: Time to merge log files

DOI10.17148/IJARCCE.2017.6606

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 6, June 2017

VI. 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.

REFERENCES

[1] P.T. Jaeger, J. Lin, and J.M. Grimes, "Cloud Computing and Information Policy: Computing in a Policy Cloud?," J. Information Technology and Politics, vol. 5, no. 3, pp. 269-283, 2009.
[2] R. Jagadeesan, A. Jeffrey, C. Pitcher, and J. Riely, "Towards a Theory of Accountability and Audit," Proc. 14th European Conf. Research in Computer Security (ESORICS), pp. 152-167, 2009.
[3] R. Kailar, "Accountability in Electronic Commerce Protocols," IEEE Trans. Software Eng., vol. 22, no. 5, pp. 313-328, May 1996.
[4] M.C. Mont, S. Pearson, and P. Bramhall, "Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services," Proc. Int'l Workshop Database and Expert Systems Applications (DEXA), pp. 377-382, 2003.
[5] X. Feng, Z. Ni, Z. Shao, and Y. Guo, "An Open Framework for Foundational Proof-Carrying Code," Proc. ACM SIGPLAN Int'l Workshop Types in Languages Design and Implementation, pp. 67-78, 2007.
[6] R. Kailar, "Accountability in Electronic Commerce Protocols," IEEE Trans. software Eng., vol. 22, no. 5, pp. 313-328, May 1996.
[7] W. Lee, A. Cinzia Squicciarini, and E. Bertino, "The Design and Evaluation of Accountable Grid Computing System," Proc. 29th IEEE Int'l Conf. Distributed Computing Systems (ICDCS '09), pp. 145-154, 2009.
[8] S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud," Proc. First Int'l Conf. Cloud Computing, 2009.

BIOGRAPHIES

K. Venkata Chalapathi received B.Tech degree in Computer Science and Engineering from Sri Vidyanikethan Engineering College, JNTUA, Anantapur, A.P, India in 2010 and M.Tech, Computer Science and Engineering, final semester, from Sri Venkateswara University College of Engineering, Tirupati, A.P, India. His interested areas are cloud computing and Data Mining. He attended Two National Conferences during 2013 and 2014.

DOI10.17148/IJARCCE.2017.6606