



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 Issue: VII Month of publication: July 2017

DOI:

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

An Efficient Scheme for Ensuring Distributed Accountability for Data Sharing in the Cloud

Ms. Rakshiya K. Siddiqui¹, Prof. Ritesh Kumar Yadav², Dr. Varsha Namdeo³

^{1,2,3} Department of Computer Science & Engineering, RKDF IST, Bhopal, RGPV University, India

Abstract: Cloud computing is that the use of computing of sources (hardware and software) that area unit delivered as a service over a network (typically the internet). It permits extremely climbable services to be simply consumed over the web on required basis. A serious characteristic of the cloud services is that users' information area unit is typically processed remotely in unknown machines that users don't operate. It will become a considerable roadblock to the wide adoption of cloud services. To handle this downside, a extremely localized accountability framework to stay track of the particular usage of the user's information within the cloud is proposed. The Cloud Information Accountability framework planned during this work conducts machine-driven work and distributed auditing of relevant access performed by any entity, meted out at any purpose of your time at any cloud service supplier. The two major elements of proposed system are lumberjack and log harmonizer. The proposed system will also take concern of the JAR file by changing the JAR into obfuscated code which is able to add a further layer of security to the infrastructure. Further the protection of user's information is increased by obvious information possessions for integrity verification.

Keywords: Cloud computing, data sharing, JAR files, information accountability framework, Provable data possession.

I. INTRODUCTION

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model that provides dynamically scalable and often virtualized resources as a service over the Internet. Currently, large numbers of commercial and individual cloud computing services are available. Users of these services are abstracted from the details of the service provider and users may not know the machines which process and host their data. Due to this, users have started worrying about losing control of their own data. The data processed on clouds are often outsourced, leading to several issues related to accountability, including the handling of personally identifiable information. Such threats are becoming a significant barrier to the emerging and new technology of cloud services.

To remove the threats of cloud services and reduce the users concerns, 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 per the service level agreements made at the time they sign on for services in the cloud. As a solution to overcome the above problems, a novel approach, named Cloud Information Accountability framework, based on the notion of information accountability is proposed in this paper.

The Cloud Information Accountability framework proposed in this paper conducts automated logging and distributed auditing of relevant access performed by any entity, taken out at any point of time at any cloud service provider. It has two major components: logger and log harmonizer. The JAR file includes a set of simple access control rules specifying whether and how the cloud servers and possibly other data stakeholders are authorized to access the content itself. Apart from that we are going to check the integrity of the JRE on the systems on which the logger components is initiated. This integrity checks are carried out by using oblivious hashing. The proposed methodology will also take concern of the JAR file by converting the JAR into obfuscated code which will adds an additional layer of security to the infrastructure. In addition to this, the security of user's data is also maintained. This is done by provable data possessions for integrity verification. Based on the configuration settings defined at the time of creation, the JAR will give usage control associated with logging, or will give only logging functionality. As for the logging, every time there is an access to the data, the JAR will automatically produce a log record.

II. EXISTING SYSTEM

Cloud computing is the delivery of computing as a service rather than a product, by which shared resources, software, and information are given to computers and other devices as a utility like the electricity grid over a network (typically the Internet). In these days a single server deals with the multiple requests from the user. Here the server has to operate the both the request from the

user simultaneously, so the processing time will be high. This may leads to deficit of data and packets may be delayed and corrupted and also the Data Management and the Services are not Trust Worthy. While enjoying the convenience brought by this new technology, users also start bothering about losing control of their own data. The data operated on clouds are often outsourced, which lead to a number of issues related to accountability, including the management of personally identifiable information. To allay users' concerns, it is necessary to provide an effective mechanism for users to monitor the usage of their data in the cloud. For example, users required 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 made for closed domains such as databases and operating systems, or approaches with a centralized server in distributed environments, are not suitable, because of the following features characterizing cloud environments.

III.PROPOSED SYSTEM

To overcome the above problems, we propose a novel method, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability. Data Owner can upload the data into the cloud server after encrypted the data. User can subscribe into the cloud server with certain access polices such as read, write and copy of the original data. The Loggers and Log Harmonizer will have a track of the access logs and reports to the data owner. This Process ensures security.

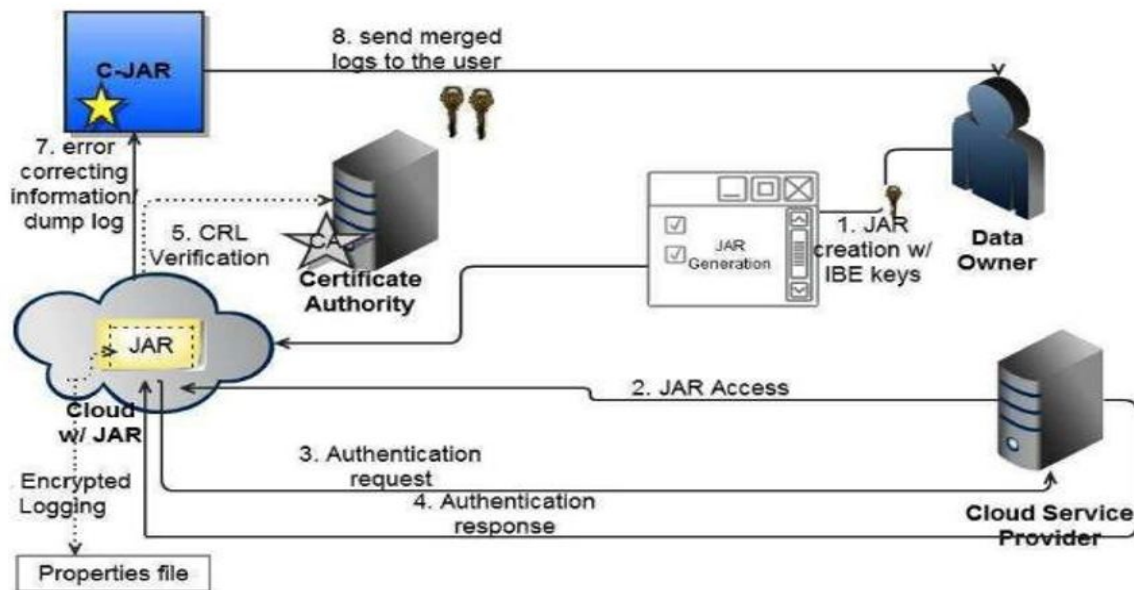


Fig. 3.1 System Architecture

The proposed system has following modules

A. User/Data Owner

User is the person is going to see or download the data from the Cloud server. To access the data from the Cloud server, the users have to be registered with the cloud server. So that the user have to register their details like username, password and a set of random numbers. This is the information that will store in the database for the future authentication. Data Owner is the Person who is going to upload the data in the Cloud Server. In order to upload the data into the Cloud server, the Data Owner have to be registered in the Cloud Server. Once the Data Owner registered in cloud server, the space will be assigned to the Data Owner.

B. Cloud Servers

Cloud Server is the area where the user going to request the data and also the data owner will upload their data. Once the user send the request regarding the data they want, the request will ne first send to the Cloud Server and the Cloud Server will forward your

request to the data owner. The data Owner will send the data the data the user via Cloud Server. The Cloud Server will also manage the Data owner and Users information in their Database for future purpose.

C. Logger

The Logger is maintained by the Cloud Server. Loggers have the details of the data owner and users who are accessing the Cloud Server. So the Logger will be more useful for many purposes. Like which user / data owner accessing the Cloud Server, accessed at the particular time and the IP address from which the data is requested by user etc.

D. Certificate Authority

The Certificate Authority is used to verify the Cloud Server is recognized or not. The Cloud Server has to be recognized by the certificate authority. If not recognized, the Cloud Server is a Fraudulent Server. The data owner can check the whether the recognized or not. Because the data owner is going to upload their data in the Cloud Server.

E. Access Privileges

The access privileges are set by the data owner for accessing their data. Some Owners will provide read only, some of them will allow read and download. The Cloud Server will send the dynamic intimation when the user is accessing the data beyond their limits. This increases more security while sharing the data in the Cloud.

F. Push and Pull Concept

For the every periodical time the Cloud Server will send the access details of the user to the data owner. So that the Data Owner may able to know who're all the accessing their data at the particular time period. During the registration phase, the Data owner will ask by the Cloud Server whether they're choosing the push or pull method. In the Pull method, the data owner has to send the request to the Cloud Server regarding the access details of their data up to the particular time. Then the Cloud Server will send the response to the Data Owner regarding the user's access details.

G. Random Set Generation and Verification

When the user request the data to be downloaded from the Cloud Server, the user have to enter the Random number set. If it is matched, the user is allowed to download the data. The Random number sets will be provide to the user during the registration Phase itself. Each and Every time the Random number set will vary. This ensures security while downloading the data.

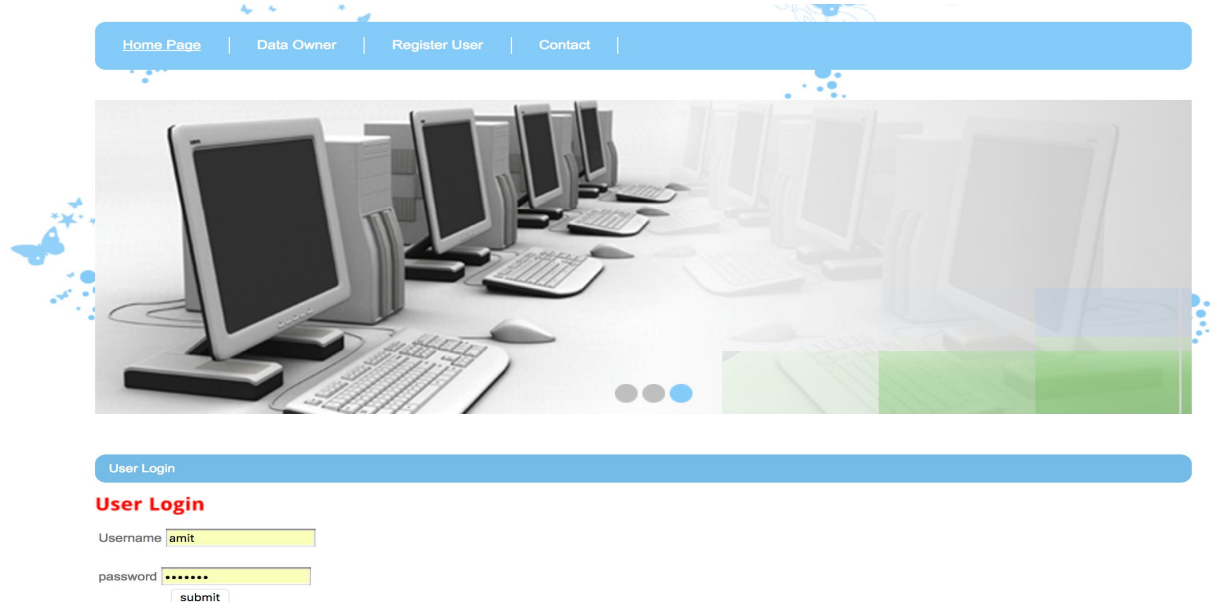
IV. TECHNIQUES USED

The algorithm here used is Log Retrieval Algorithm for push and pull modes. The algorithm presents logging and synchronization steps with the harmonizer in case of Pure Log. First, the algorithm checks whether the size of the JAR has exceeded a stipulated size or the normal time between two consecutive dumps has elapsed. The size and time threshold for a dump are specified by the data owner at the time of development of the JAR. The algorithm also determines whether the data owner has requested a dump of the log files. If none of these events has happened, it proceeds to encrypt the record and write the error-correction information to the harmonizer. The interaction with the harmonizer begins with a simple handshake. If no response is received, the log file records an error. The data owner is then alerted via emails, if the JAR is configured to send error notifications. Once the handshake is done, the interaction with the harmonizer proceeds, using a TCP/IP protocol. If either of the mentioned events (i.e., there is request of the log file or the size or time exceeds the threshold) has happened, the JAR simply dumps the log files and resets all the variables, to make a space for new records. In case of Access Log, the above algorithm is modified by adding an additional check after step 6. Precisely, the Access Log checks whether the CSP accessing the log satisfies all the conditions specified in the policies pertaining to it. If the conditions are fulfilled, access is granted; otherwise, access is declined. Irrespective of the access control outcome, the attempted access to the data in the JAR file will be logged. Our auditing mechanism has two fundamental advantages. First, it guarantees a high level of availability of the logs. Second, the usage of the harmonizer minimizes the amount of workload for human users in going through long log files sent by different copies of JAR files.

V. SYSTEM RESULTS

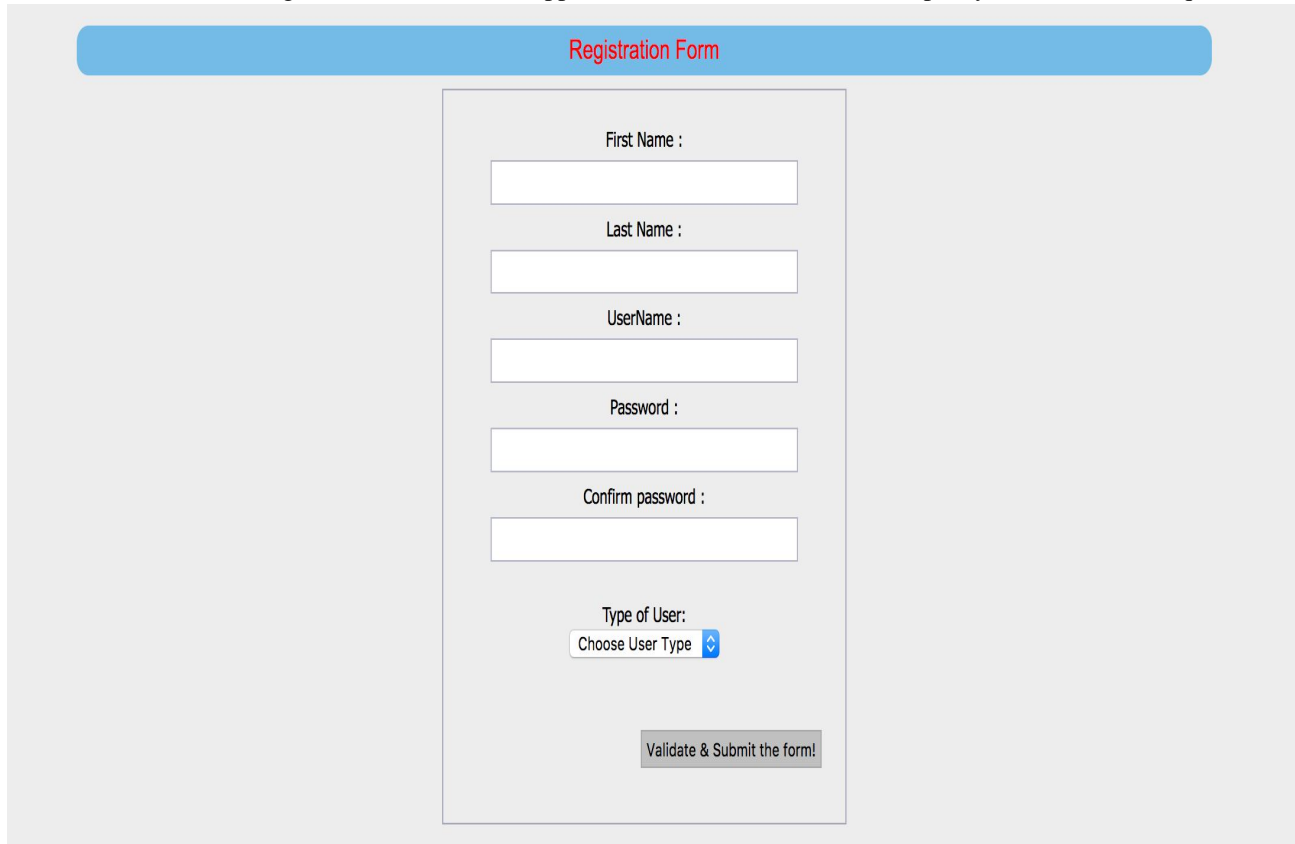
Following are some of the screenshots of the implemented hybrid book recommendation system.

A. The screenshot shows the first page i.e. welcome page of the application. This is the first page user first encounters in order to login. To perform this operation, user has to enter his/her username and password used during registration phase.



The screenshot shows the 'User Login' page of the application. At the top, there is a navigation bar with links: Home Page, Data Owner, Register User, and Contact. Below the navigation bar is a large image of several computer monitors on a desk. Underneath the image, there is a 'User Login' section with the title 'User Login' in red. It contains two input fields: 'Username' with the value 'amit' and 'password' with masked characters '*****'. A 'submit' button is located below the password field.

B. The screenshot shows the registration screen of the application. User has to feed data as per system validation requirements.



The screenshot shows the 'Registration Form' page of the application. The title 'Registration Form' is displayed in red at the top. The form contains several input fields: 'First Name', 'Last Name', 'UserName', 'Password', and 'Confirm password'. Below these fields is a 'Type of User' dropdown menu with the text 'Choose User Type' and a blue arrow icon. At the bottom of the form is a button labeled 'Validate & Submit the form!'.

C. The screenshot is of the data owner login screen. The validation of data owner's credentials is very important leading to access to further operations.



DataOwner Login

Admin Login

Admin:

password:

D. This is the screenshot of page wherein file uploading operations takes place.

Welcome To Uploading

Information about your collection

Name of Collection :

Information about Collection:

Select Country:

Specify your File : No file chosen

Visible: ☐ Yes ☐ No

Enter the Duration :

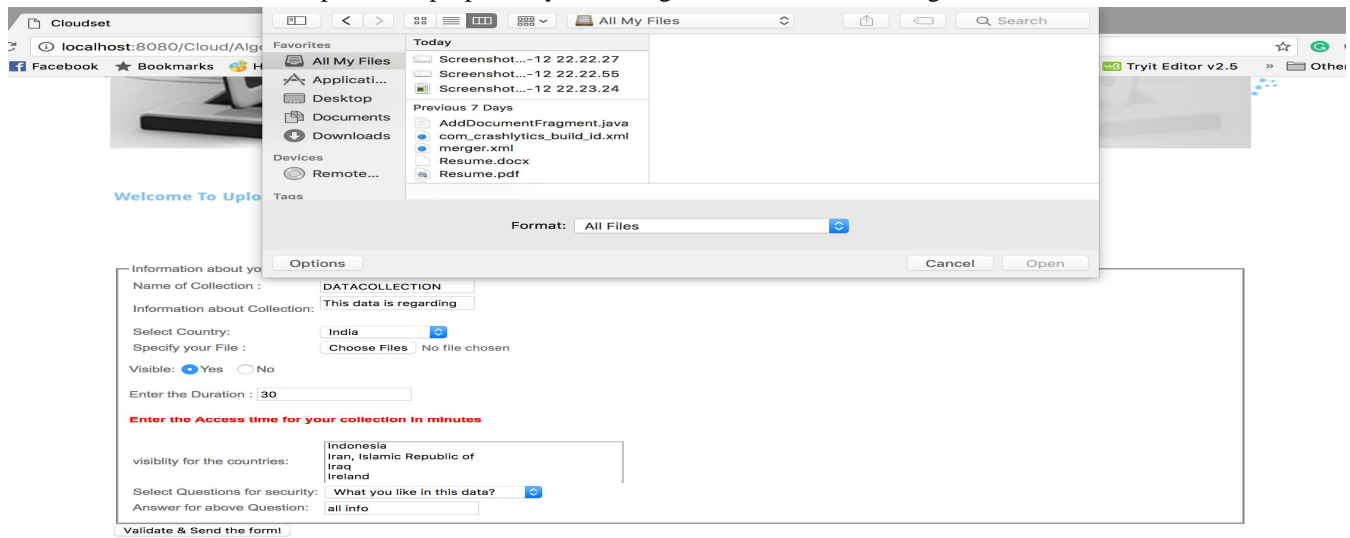
Enter the Access time for your collection in minutes

visibility for the countries:

Select Questions for security:

Answer for above Question:

E. The screenshot shows the output of the proposed system using the above mentioned algorithms.



CONCLUSION

We introduced modern 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. Apart from that we have enclosed PDP methodology to enhance the integrity of owner's data. In future, we plan to refine our approach to verify the integrity of JRE. For that we will look into whether it is possible to leverage the advantage of secure JVM being developed by IBM and we would like to enhance our PDP architecture from user end which will allow the users to check data remotely in an efficient manner in multi cloud environment.

REFERENCES

- [1] Ensuring Distributed Accountability for Data Sharing in the Cloud Author, Smitha Sundareswaran, Anna C.Squicciarini, Member, IEEE, and Dan Lin, IEEE Transactions on Dependable and Secure Computing ,VOL 9,NO,4 July/August 2012.
- [2] Hsio Ying Lin,Tzeng.W.G, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding ",IEEE transactions on parallel and distributed systems,2012.
- [3] Yan Zhu, Hongxin Hu, Gail Joon Ahn, Mengyang Yu, "Coopera-tive Provable Data Possession for Integrity Verification in Multi-Cloud Storage" , IEEE transactions on parallel and distributed systems,2012. 3. Generic Website
- [4] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.
- [5] Eucalyptus Systems, <http://www.eucalyptus.com/>, 2012.
- [6] Emulab Network Emulation Testbed, www.emulab.net, 2012.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)