ACCOUNTABLE DATA SHARING IN CLOUD USING LOGGING MECHANISM

R.RaamPriya, L.Vinothini
PSG College of Technology, Coimbatore,

Abstract **--In Cloud Computing the scalable services are easily consumed in an on-demand basis via internet. In this the User's data are processed remotely in an unknown machine, where there is a fear of losing their own data. In order to overcome this problem, a decentralized accountability framework is proposed to keep track of the actual usage of data. In this paper, we propose a logging mechanism, along with the user data and policies to keep track of the actual usage of the data. The data is uploaded using JAR. To ensure that any access to the users data will trigger authentication and automated logging we develop a distributed auditing mechanism, which strengthen the user's control and also used for handling log files.**

Keywords **--Cloud Computing ,accountability ,data sharing.**

I.INTRODUCTION

As suggested in plenty of articles across the scientific literature, cloud computing is envisaged to be the next key computing paradigm. This new approach is expected to bring availability, scalability and ubiquitous access as never seen before. However despite this growing interest in cloud computing there are many voices pointing to the lack of an accepted for this computing paradigm[1].

Cloud computing is a major change in how we store information and run applications. Instead of running programs and data on an individual desktop computer, everything is hosted in the "cloud" –a nebulous assemblage of computers and servers accessed via internet. Cloud computing lets you access all your application and documents from anywhere in the world freeing you from the confines of the desktop and making it easier for group members on different locations to collaborate. The emergence of cloud computing is the computing equivalent of the electricity revolution of a century ago. Before the advent of electrical utilities, every farm and business produced its own electricity from freestanding generators. After the electrical grid was created, farms and businesses shut down their generators and bought electricity from utilities, at a much lower price than they could produce on their own [6].

Key to the definition of cloud computing is the "cloud" itself. For our purpose, the cloud is a large group of interconnected computers. These computers can be personal computers or network of servers; they can be public or private. For example, Google hosts a cloud that consists of both smallish PCs and large servers. Google's cloud is private one that is publicly accessible. This cloud of computers extends beyond a single company or enterprise. The applications and data served by the cloud are available to broad group of users, cross-enterprise and cross platform. Access via the internet. Any authorized user can access these doc and app from any computer over internet connection. And, to the user, the technology and infrastructure behind the cloud is invisible. Its apparent whether cloud services are based on HTTP , HTML ,XML, JavaScript or other specific technologies.

Cloud computing consist of two types of models. First is the service model which is of three types namely SaaS ,IaaS, PaaS. Software as a service(SaaS) is probably the most common type of cloud service development. With SaaS ,a single application is delivered to thousands of users from the vendor's servers. Customers don't pay for owning the software; rather, they pay for using it. Users access an application via API accessible over the web. Platform as a service, in this variation of SaaS ,the development environment is offered as a service. The developer uses the "building blocks" of the vendor's development environment to create his own custom application. It's kind of like creating an application using Legos; building the app is made easier by use of these predefined block of code, even if the resulting app is somewhat constrained by the types of code. Infrastructure as a service IaaS is the most basic and each higher model abstracts from the details of the lower models. Second is the deployment model which is of four types-Public cloud, Private cloud, Community cloud and Hybrid cloud.

II. RELATED WORK

In this section we briefly discuss works which adopt similar techniques as our approach but serves for different purpose.

2.1 Media Cloud: An open cloud computing middleware for content management.

There is an increasing interest in sharing the media files with family and friends. However, UPnP or DLNA were not designed for media distribution beyond the boundaries of a local network and management of such media files through web application is tedious. To overcome this problem, we propose media cloud which is a middleware between the user and data.

2.2 Social cloud computing: A vision for socially motivated resource sharing.

Online relationships in social networks are often based on real world relationships and can therefore be used to infer a level of trust between users. In this the users share heterogeneous resources within the context of social network. The resource access permission is outsourced. So that we go for posted price allocation.

2.3 HASBE:A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing.

Users entrust valuable data to the cloud providers which are often been outsourced. In this ASBE is used for access control. But complex access control implementation is difficult. For that we use HASBE by extending cipher text policy of ASBE. This enables the scalability and flexibility. HASBE has access expiration time.

2.4 Scalable and secure sharing of personal health records in cloud computing using ABE.

PHR is outsourced to be stored by third party. This assures patients to control over access to their own PHR by encrypting the PHR. Attribute Based Encryption (ABE)is used to encrypt the contents of the record. This enables the modification of access policy or file attribute.

2.5 Secure Logging As a Service-Delegating Log Management to the Cloud.

Securely maintaining log records over extended periods of time is very important to the proper functioning of any organization. Integrity of the log files and that of the logging process need to be ensured at all times. In addition, as log file soften contain sensitive information, confidentiality and privacy of log records are equally important. However, deploying a secure logging infrastructure involves substantial capital expenses that many organizations may find overwhelming. Delegating log management to the cloud appears to be a viable cost saving measure. In this paper, we identify the challenges for a secure cloud-based log management service and propose a framework for doing the same.

III. PROBLEM STATEMENT

This begins by creating a cloud, where the user uploads the data in the cloud. But the data on cloud are often outsourced which leads to security and privacy issues. Also the actual usage of the data is unknown, so a logging mechanism is developed to track the actual usage of the data and all the operation on data are governed by the log generator. Some access control policies are also uploaded along with the data. The Cloud Service Provider(CSP) will receive the data and grants the access rights. To access the data, users have to register in the cloud. Free registration leads to access only some data. To access full data user has to pay and get the rights from the CSP. After the registration, different user id and password are given for all the users and now they can access the data on the cloud. The data on cloud are kept in the encrypted format. Algorithms like MD5 and SHA1 are used for encrypting the data. But sometimes the data on cloud can be decrypted by third party using the decryption keys. When more number of log files is generated auditing of log files becomes tedious. And the merging of log files takes more time and these log files requires more space.

IV. PROPOSED SYSTEM

Along with this logging mechanism, we propose an auditing mechanism with two auditing modes-push mode and pull mode. In push mode all the log files are sent periodically to the data owner by the harmonizer. In pull mode all the log files are sent on an on-demand basis. This mode allows auditors to retrieve the logs anytime when they want to check the recent access to their own data. The log files contains the name of the person who access the data, type of action performed, date, time, etc. In current system we propose three types of actions i.e. view, download and time accessed.

Example: Suppose a cloud service provider with ID James, located in UK, downloads an image in a JAR file, at 6:20pm on July 19,2012. The log record is:

<James, download, 18:20:32, UK, jhg8376k,356vfsgh8>

The location is converted from the IP address for improved readability.

In order to avoid the easy decryption of data either encrypt the data using AES algorithm or generate the decryption keys during run time. We use a log retrieval algorithm for the easy retrieval of log files. For easy auditing we can reduce the log file merge time by using the corresponding queries and by storing the files in the database, by doing so we can also reduce the storage space.

V. SYSTEM DESIGN

This system consists of five modules. First is module is the formation of cloud where the data is stored. The second module is the user and owner creation where the admin creates a user and owner and grants the access rights. The third module is the uploading the data to cloud, in which an authorized user can upload the files. Fourth module is generating the log files and the fifth module is the sending the log files to the owner.

i. Cloud Formation

A novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. The end user should store the data in the cloud. The user can access the stored data where ever needed. 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 we use a cloud database called xeround cloud is used instead of a mysql database, in this each user is provided with a user id and password, through which they can login and access the data available in cloud at any time via internet.

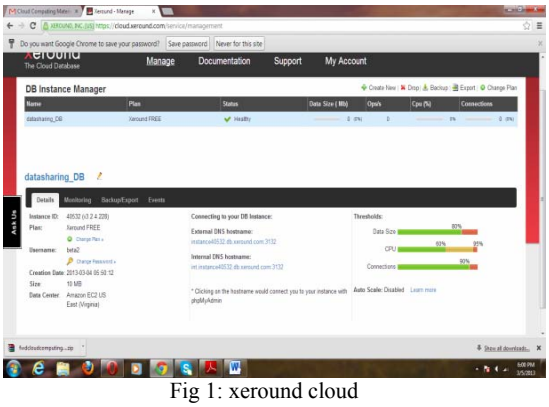

Fig 1: xeround cloud

ii.User and owner creation

To access the cloud data the users have to register in the cloud formation. The free registration leads to access some of the data only. To access full data the user has to pay and get the rights from the cloud. After the registration using their id and password they can login and the access their data. The user is given the rights to access the files available in the cloud and he can view his account details. The owner is given the rights to view the visitors history, registered users and also can upload data in the cloud.

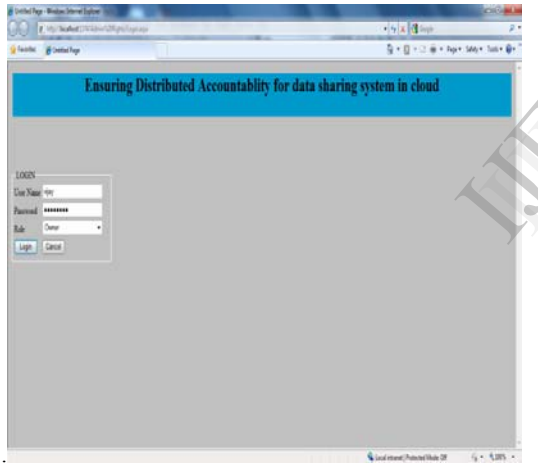

Fig 2: User creation

iii.Upload Data into cloud

We can upload the data to data access in the cloud. The uploaded file should be uploading through JAR (Java Achieves). And the uploaded data is in encrypted format. So that no is able to hack the data. Users send the data along with access control policies. The user will have control over his data at any location. The owner can upload data like images and documents, they can also upload two or more files at the same time.

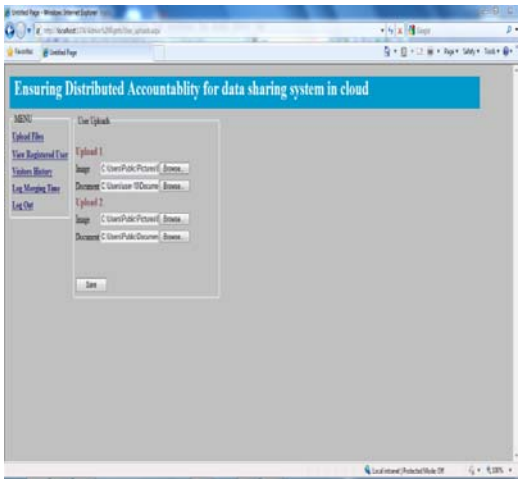

Fig 3:File uploading

The users registered in the cloud can also view the files available in the cloud by just selecting the file which they want, and then they can click on view button.

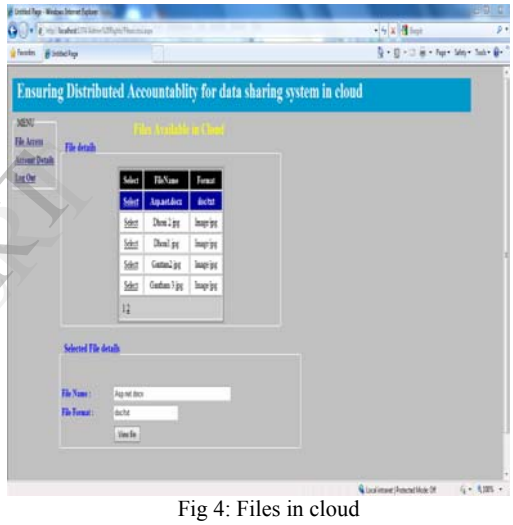

Fig 4: Files in cloud

iv. Creating Log during Data Access

To strengthen user's control, we also provide distributed auditing mechanisms. All the details should be maintained in the log. It stores all the details for authorized users as well as unauthorized users. The unauthorized users can access only limited amount of data. There are two types of sending: On-demand and Usual method.In usual method all the transactions should be sent to the owner. In on demand method the owners have to buy the details purposely from the cloud. While creating log we can also reduce the log merging time.

Fig 5: Log files

v.Reporting to the Owner

At the end of each day the owner view the details of data storage and the details of upload and download details. The owner can view the weekly details and monthly details also. The owner will receive a log file to his email, so that the owner can keep track of his own data. The generated log file is an excel file, which contains the name of the user, the file he accessed,access date and access time.

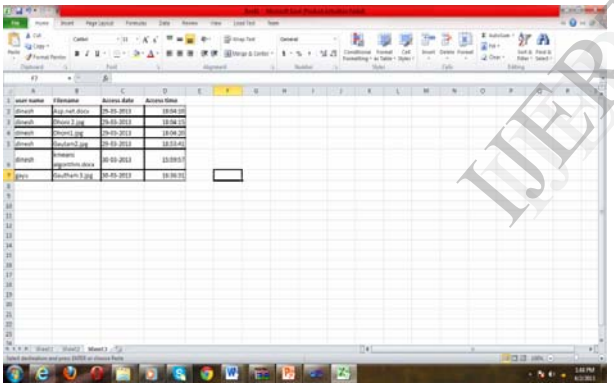

Fig 6:Log in Excel sheet

GRAPH:

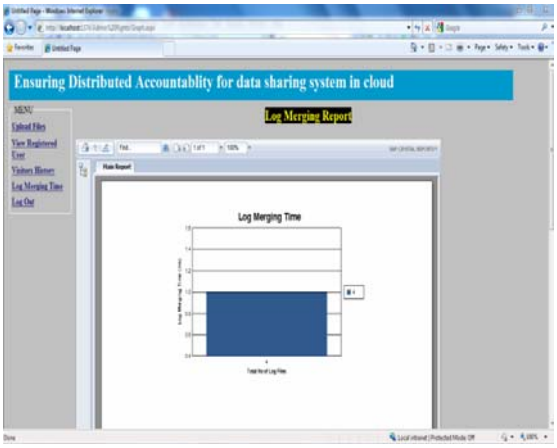

Fig 7:This graph shows the time take to merge the log file

6. CONCLUSION

The main feature of our proposed system is that it enables the data owner to audit even those copies of its data that were made without his knowledge. This paper proposes a secured sharing of data in the cloud by proposing a decentralized accountability framework called logging mechanism. Thus the data owners can protect and audit their content and also the copies of the data that were made without the owner knowledge can also be audited.

Hence we conclude that the data on cloud can be securely shared and audited by generating log files during the data access in cloud and these log files are the send to the data owner at the end of each day, by doing so the owner can keep track of the actual usage of data. This avoids the access and modification by unauthorized users and also avoids data. The log files merge time is also reduced.

7 FUTURE ENHANCEMENT

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 Java 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.

References

1. *Daniel Díaz-Sánchez, Member, IEEE, FlorinaAlmenarez, Member, IEEE, Andrés Marin, Member, IEEE, Davide Proserpio, Member, IEEE, and Patricia Aria Cabarcos, Member, IEEE ," Media Cloud: An Open Cloud Computing Middleware or Content Management", IEEE Transactions on Consumer Electronics, Vol. 57, No. 2, May 2011 .*

2. *KyleChard , Member, IEEE, Kris Bubendorfer ,Member, IEEE, Simon Caton, Member, IEEE,and Omer Rana, Member , IEEE," Social Cloud Computing: A Vision for Socially Motivated Resource Sharing", IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 55, NO. 3, MONTH 2010.*

3. *Zhiguo Wan, Jun'e Liu, and Robert H. Deng, Senior Member, IEEE," HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Controling Cloud Computing",INFORMATION FOR ENSICS AND SECURITY, VOL. 7, NO. 2, APRIL 2012.*

4. *Abdulrahman A. Almutairi and Muhammad I. Sarfraz, Purdue University*

SalehBasalamah,Umm Al-Qura University Walid G. Aref and ArifGhafoor, Purdue University," A Distributed Access Control Architecture for Cloud Computing", published by the IEEE computer society March/April 2012.

5. *Ming LiMember, IEEE,ShuchengYu,Member, IEEE, Yao Zheng, Student Member, IEEE, KuiRen, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE," Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Student Member, IEEE, KuiRen, Encryption", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS VOL. 6, 2012.*

6. *Michael Miller,"Cloud Computing", web-based application that change the way you work and collaborate online.*