BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE)
ISSN 2348-0513
Vol. 2, Issue 3, Mar 2014, 37-46

"SECURE JAR" ENSURING DISTRIBUTED DATA SHARING AND SECURITY IN CLOUD

SNEHAL R. BADHE, PRIYA D. KADAM, POONAM M. JADHAV & AMRUTA A. BARGE

JSMP's Jayawantrao Sawant College of Engineering, Pune University, Maharashtra, India

ABSTRACT

This paper describes an application which maintains log of data usage stored on cloud. It also provides security to the data by compiling jar of that file. As we know that, once data stored on cloud user may not know that where his data is used and who is using the data. By using this application data owner will get all the information about who is using data. In this paper we have provided policies to the data like user can view, download the data based on his wish. As data is stored by data owner, data owner will get information of his data periodically. Data owner also can get the information of the data whenever he wants. For this purpose we are using pull and push mode. To secure the data from unauthorized user authentication is provided.

KEYWORDS: Internet, Cloud Services, Storage, Social Networking Sites

INTRODUCTION

Introduction for Cloud

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. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models. Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. Doing so may give rise to certain privacy implications. Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks , computer processing power, and specialized corporate and user applications.

Privacy Risks

While there are benefits, there are privacy and security concerns too. Data is travelling over the Internet and is stored in remote locations. In addition, cloud providers often serve multiple customers simultaneously. There is a major risk in cloud is security for data stored on cloud.

Project Concept

At the beginning each user creates a pair of public and private keys based on Identity-Based Encryption (IBE). These keys are used to protect us against one of the most prevalent attacks such as detecting illegal copies of user's data. These keys are sending to the users through mobile message or mail. Using the generated key the user will create a logger component which is 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 are authorized to access the content itself. Then he sends the JAR file to the cloud service provider that he subscribes to. Once the authentication succeeds, the service provider will be allow to access the data enclose in the JAR.As for the logging, each time there is an access to the data, the JAR will automatically generate a log record, encrypt it using public key.

The encryption of log file prevents unauthorized changes to the file by attackers. The data owner could have two options as to reuse the same key pair for all JARs or to create different key pair for separate JARs. Using separate keys can enhance the security without introducing any overhead except in the initialization phase. To ensure trustworthiness of 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. Then the encrypted log files are decrypted at that time their integrity is verified. After decryption of data, this data can be used by owner or any authorised user.

Project Objectives

A main objective of this project is to propose a different cryptographic algorithm which can protect & secure cloud data more efficiently than existing cryptographic algorithms. The main objective of this project is to generate the log record & stores all log data which is given to data owner from cloud server. The project objective will be achieved by examining the current approaches to securing data in storage and the way it may be processed by cloud-based applications through the implementation of homomorphic encryption. Another objective is to provide confidentiality, integrity & reliability as the data remains in the encrypted form at the time of processing like in storage, for retrieve, in process & when particular output if produced.

LITERATURE SURVEY

Internet is the required thing in today's world. As whole information about everything is available on internet, it is occupying the whole world. There are various application services which user can use through internet. Internet is transmission medium and transform all technologies, software services, cloud services, etc. cloud computing could be defined as the integration of virtual resources according to user requirements, flexibly combining resources including hardware, development platforms and various applications to create services. Generally speaking, cloud computing applications are demand-driven, providing various services according to user requirements, and service providers charge by metered time, instances of use, or defined period.

In cloud computing data is stored on cloud by data owner. Data owner will provide accessibility according to his need like view, copy, download. Logger is provided so that data owner will know the location, time, access type. Encryption decryption is good cryptography method to provide security.

PROBLEM STATEMENT

Example 1

Alice, a professional photographer, plans to sell her photographs by using the Sky High Cloud Services. For her business in the cloud, she has the following requirements:

- Her photographs are downloaded only by users who have paid for her services.
- Potential buyers are allowed to view her pictures first before they make the payment to obtain the download right.
- Due to the nature of some of her works, only users from certain countries can view or download some sets of photographs.

- For some of her works, users are allowed to only view them for a limited time, so that the users cannot reproduce her work easily.
- In case any dispute arises with a client, she wants to have all the access information of that client.
- She wants to ensure that the cloud service providers Of Sky High do not share her data with other service providers, so that the accountability provided for Individual users can also be expected from the cloud Service providers.

With the 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.

ARCHITECTURE

Existing System

From the showing of data security, which has always been an important aspect of quality of service, Cloud Computing avoid less poses new challenging security threats for number of reasons.

- Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging.
- Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance.

As a complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited.

PROPOSED SYSTEM

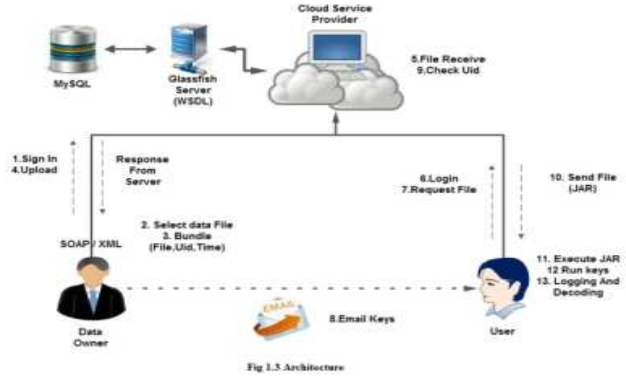

Figure 1

- Data owner will register himself on CSP. After registering data owner will login on his account to host his services on cloud. If he is authorised user then server will give response to him.
- After getting response from CSP data owner will select the file to upload it on cloud.
- After selecting files he will bundle it with his UID and time in jar format. i. e it will compile jar.
- After bundling all this data, he will upload it on cloud service provider.
- CSP will receive the files in jar format. Data owner will encrypt data and upload it to CSP. Those files will be converted in WSDL format through Glassfish server. Then this data is stored in database.
- Now if user wishes to access the data from cloud. So for that first he has to register on CSP. After registration he will log in to account.
- After login successfully, he will request files.
- Data owner will provide keys based on access control that he want to give to that user. Email keys are composite keys containing access of the data.
- After checking that user is authorised or not, CSP will provide the required file to user in jar format with private key.
- After getting jar file on user's machine, he will execute that file.
- By using keys provided by data owner user will decrypt that file and he will get original data.
- Log record is created at the time of execution and it will send back to CSP in encrypted form.
- By using pull or push mode data owner can get usage of his data from CSP.

AUTOMATED AUDITING MECHANISM

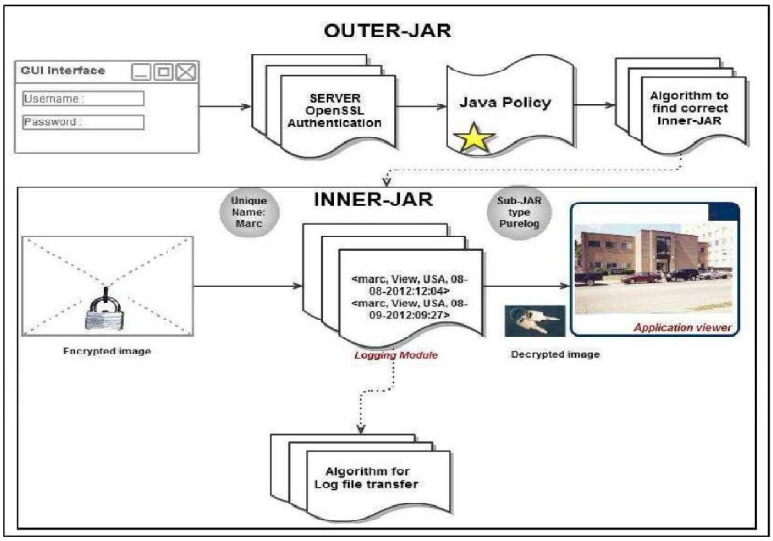

Figure 2

Logging Structure

In the auditing mechanism there is one outer jar and remaining are inner jar. Outer jar contains class files for authentication of server or user another class file for finding correct inner jar, third class file is to check VM's validity using hashing function. Inner jar contain class file for writing log records, another class file which corresponds with the log harmonizer, the encrypted data and third class file to display and download data and public key of the IBE key pair that is necessary for encrypting log records.

Working is as follow

- Authentication is done in first step according to user name and password.
- After authentication is succeeded access control policies are given to user.
- Once proper access is given, there is algorithm which redirect user to correct inner jar.
- In inner jar user will get data which he want, in encrypted format.
- Log record is generated according to access given by data owner.
- Then data will be available in original format i.e. decrypted data. Log records are sent to cloud.
- From cloud data will be sent to data owner by pull or push method.

Generation of Log Record

Log records are generated by the logger component.

Logging occurs at any access to the data in the JAR, and new log entries are appended sequentially, in order of creation LR= <r1; . . . rk>. Each record ri is encrypted individually and appended to the log file. In particular, a log record takes the following form:

Ri=<ID, Act, T, Loc, h (ID, Act, T, Loc) |ri _ 1|. . . |r1),sig>

Here, ri indicates that an entity identified by I D has performed an action Act on the user's data at time T at location Loc. The component h(ID, Act, T, Loc, |, ri _ 1|. . .|jr1) corresponds to the checksum of the records preceding the newly inserted one, concatenated with the main content of the record itself (we use I to denote concatenation). The checksum is computed using a collision-free hash function.

The component sig denotes the signature of the record created by the server. If more than one file is handled by the same logger, an additional Obj ID field is added to each record. An example of log record for a single file is shown below.

Example 4

Suppose that a cloud service provider with ID Kronos, located in USA, read the image in a JAR file (but did not download it) at 4:52 pm on May 20, 2011. The corresponding log record is

(Kronos, View, 2011-05-29 16:52:30, USA, 45rftT024g, r94gm30130ff)

In current system we provide four types of actions:

View, download, timely-access, location _ based _ access.

Auditing Modes

There are two Auditing modes

- Push Mode
- Pull Mode

Push Mode

In this mode, the logs are periodically pushed to the data owner (or auditor) by the harmonizer. The push action will be triggered by either type of the following two events: one is that the time elapses for certain period according to the temporal timer inserted as part of the JAR file; the other is that the JAR file exceeds the size stipulated by the content owner at the time of creation. After the logs are sent to the data owner, the log files will be dumped, so as to free the space for future access logs. Along with the log files, the error correcting information for those logs is also dumped.

Pull Mode

This mode allows auditors to retrieve the logs anytime when they want to check the recent access to their own data.

Algorithm

Size*:* Maximum size of log file specified by the data owner, *time:* maximum time allowed to elapse before log file is dumped, ***t beg:*** timestamp at which last dump occurred, ***log****:* current log file, ***pull:*** indicates whether a command from the data owner is received.

- *Let TS(NTP) be the Network Time Protocol timestamp*
- *pull := 0*
- *rec := < UID, OID, AccessType, Result, Time, Loc>*
- *curtime := TS(NTP)*
- *lsize := sizeof(log) // current size of the log*
- *if ((cutime-tbeg)< time)&&(lsize<size)&&(pull==0) then*
- *log := log + ENCRYPT(rec) // ENCRYPT is the encryption function used to encrypt the record*
- *PING to JAR // send a PING to the harmonizer to check if it is alive*
- *if PING-JAR then*
- *PUSH RS(rec) // write the error correcting bits*
- *else*
- *EXIT(1) //error if no PING is received*
- *end if*
- *end if*
- *if ((cutime-tbeg)>time)||(lsize>= size) ||(pull!=0) then*

- *// check if PING is received*
- *if PING-JAR then*
- *PUSH(log) // write the log file to the harmonizer*
- *RS(log) := NULL //reset the error correction records*
- *tbeg := TS(NTP) // reset the tbeg variable*
- *pull := 0*
- *else*
- *EXIT(1) //error if no PING is received*
- *end if*
- *end if*

MATHEMATICAL MODEL

Mathematical modelling has been used to solve problems not only in engineering and physics, but also in biology and health care. Here is a general guideline for how to build a mathematical model.

Let

Assumption= {Proper internet connection is available, JVM is available}

Input= {Uid, Passwd}.

Output= { }.

Success= {Data is provided to authenticated user only}.

Failure= {Server Down, disturbance in connection, unsuccessful authentication}.

Graph Theory

Table 1

Edges	Vertex
e1= User registration. e2=Login. e3=File selection. e4=Attribute assignment. e5=Jar creation (Inner). e6=Encryption(Log Creation) e7=Outer jar creation. e8=Upload to cloud. e9=Request file access. e10= Data owner generates password. e11=Transmit to data user. e12=Download e13=Decrypt using ABE.	V1=User (Data owner) V2=Server. V3=File system. V4=Jar creator. V5=Encryptor. V6=Storage cloud. V7=Data user.

Set Theory

Table 2

Functions	Abbreviation
f (Reg) = {U}. f (login)={Uid, Passwd}. f (SHA)={Passwd}. e=f(encrypt)={Uid, Public key, Data}. f(compress)={data}. f(upload)={e, Uid}. f(download)={e, Uid}. f(decompress)={data}. F(decrypt)={uid, private key, data}.	Let S be the system for Cloud Security. Let S = {U,Sr,E,D,C,De} U is set of users. U={U1,U2,……..,Un}. Sr is set of services. Sr={Sr1,Sr2,……Srn}. E is Encryptor. De is Decompressor.

CONCLUSIONS

We have presented Ensuring distributed data sharing & security in cloud in which we are working on data security in cloud by using the JAR files. We are working for to store user's data on cloud and allow clients to download that data by taking private key from the data owner. If data owner allow that client to access the data then only client can access that data. For the security purpose we can use the one time password from the server. The log record is automatically generated on cloud & sends it to the data owner, by using that log records data owner can checks all information. We are using encryption algorithm to secure data.

REFERENCES

1. Smitha Sundarswaran, Anna C. Squicciarini, member, IEEE, and Dan Lin "Ensuring Distributed Accountability for Data Sharing in the Cloud" IEEE transactions on dependable and secure computing vol.9 no.4, year IEEE 2012.
2. OASIS Security Services Technical Committee, "Security Assertion Mark up language (saml) 2.0," http://www.oasis-open.org/ committees/tchome.php?wg abbrev=security, 2012.
3. S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud," Proc. First Int'l Conf. Cloud Computing, 2009.
4. S. Pearson, Y. Shen, and M. Mowbray, "A Privacy Manager for Cloud Computing," Proc. Int'l Conf. Cloud Computing (Cloud Com), pp. 90-106, 2009.
5. 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.
6. R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "A Logic for Auditing Accountability in Decentralized Systems," Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005.
7. S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud," Proc.First Int'l Conf. Cloud Computing, 2009.
8. S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang, "Promoting Distributed Accountability in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing, 2011.

9. D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigen-baum, J. Hendler, and G.J. Sussman, “Information Accountability,” Comm. ACM, vol. 51, no. 6, pp. 82-87, 2008.

AUTHORS DETAILS

Snehal Raju Badhe- BE Computer, JSPM Pune University, Pune, Maharashtra, India

Priya Devidas Kadam- BE Computer, JSPM Pune University, Pune, Maharashtra, India

Poonam Mohan Jadhav- BE Computer, JSPM Pune University, Pune, Maharashtra, India

Amruta Arvind Barge- BE Computer, JSPM Pune University, Pune, Maharashtra, India