Data Sharing: Ensure Accountability Distribution in the Cloud

Suraj Mehta, Aakash Takale, Shreyas Tapale
B.E. Computer,
Pune University

***Abstract*— Cloud computing is highly emerging technology of the 21st century. Cloud enables the user to store the data also the data can be processed on any of the machine which user does not know. While enjoying the convenience brought by these new technologies of cloud computing, users fear of losing the hold on their data. A thorough research is needed for detective controls in the area of accountability and logging, where in some privacy protection techniques concentrate on preventive controls. We proposed system that suburbanized information accountability framework to keep track of user's data in cloud. We chose object-centered approach to highlight the security purpose by enclosing our logging mechanism together with user's data along with policies. The proposed system will also take care of the JAR file by converting it into nebulous code which will add the further layer of security to the architecture. We ensure accountability gets distributed and providing more reliable and secure system.**

1. INTRODUCTION

Use of cloud computing is very important in today's condition, because cloud concept enhance the view of distributed structure of data, also cloud computing is widely get used because its time and cost saving application. The aim our system is, in addition to the ensure accountability distribution. We use cloud for uploading data owner's data. Data owner who has uploaded his personal data on cloud, he is not sure about the safety of his data. So we store his data on the cloud by encrypting his data and then that data is wrapped into JAR file along with the access policies and then that JAR file is stored in the cloud. And then finally the user can access his data.

DES allow data owner to put their data on cloud in encrypted format. The DES encryption technique is fully based on two significant attributes of cryptography Permutation and Substitution. It uses block cipher. It encrypts the data in block size of 64 bits each. In this instead of decryption keys data owner issues attribute key to data user. To access the particular data on a cloud data user must have necessary attribute to satisfy that structure. After encryption and decryption, logging mechanism take place for JAR file. For storing and security for the respective JAR file we use RSA algorithm. This is public key encryption algorithm. It is most popular and asymmetric key cryptographic algorithm. The third algorithm is MD5 algorithm, used for authentication purpose.

In nowadays, a single server handles the multiple requests from user. The system has absences of automated logging mechanism in the cloud which increases time complexity. Better solution for this is use of decentralized distributed system by using above algorithms. To implement all the above stated algorithms we use two distinct modes for auditing namely as push mode and pull mode for control over data and authentication respectively.

2. PROPOSED SYSTEM

We propose a pleasantly different method namely ensuring Distributed Accountability based on notation of information accountability, to overcome the above problems. The main purpose of our proposed system is to design an accountability framework for modeling a highly scalable cloud environment. This will enables policies in a JAR file. Also the system features lies in its ability of maintaining powerful accountability that combines aspect of access control, authentication and usage control.

Data owner can upload data into the cloud server after encryption of data. Then a normal user can access the respecptive data with view, download, location based, timely access policies. The loggers and log harmonizer will have a track of the access logs and reports to the data owner to ensure the data owners data security. This security can be achieved by two distinct modes namely as push mode and pull mode. Pull mode relevant to an approach that user can retrieve the logs as needed while Push mode relevant to logs being regularly sent to the data owner.

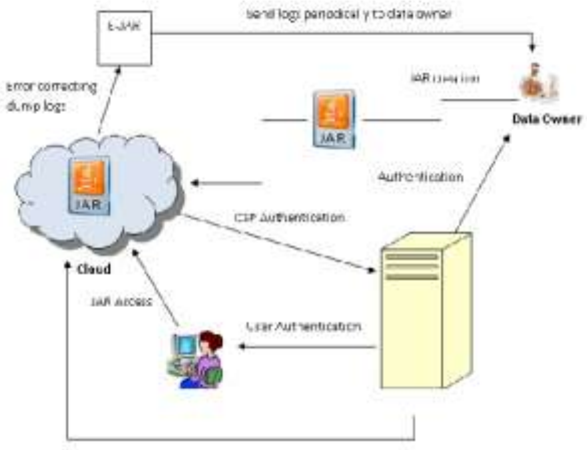

FIG1.SYSTEM ARCHITECTURE

The aim to develop logging and auditing technique which satisfy the following requirements.

1. Implement DES algorithm allowing data owner to generate security keys.
2. Implement data policies allowing data owner to mention the access control rules on paid/free user.
3. Every time user access data, log records must be generated in the form of text files, later encrypted and stored in log files.
4. Implement mechanism to keep the log record safe, to avoid missing records and data corruption.
5. Implement copying attack that free user or hacker cannot copy the data. By disabling keyboard and right click of mouse.
6. Implement dissembling attack that hacker cannot extract the data from the JAR file.
7. Implementing man in middle attack.

3. MATHEMATICAL MODEL

Problem Statement:

Cloud computing is highly emerging technology of the 21st century. Cloud enables the user to store the data also the data can be processed on any of the machine which user does not know. While enjoying the convenience brought by these new technologies of cloud computing, users fear of losing the hold on their data. A thorough research is needed for detective controls in the area of accountability and logging, where in some privacy protection techniques concentrate on preventive controls.

Mathematical Module:

Let S be the proposed System

S={Logger, IJ, OJ, LF, encrypted data, class files, LR, Log Harmonizer, pull(LR), push(LR), accesslog, purelog, master_Index}

1) Logger = {IJ, OJ, LF }

   IJ is the Inner-JAR

   OJ is Outer-JAR

   LF are the corresponding log files

2) IJ={encrypted data, ICF1, ICF2, ICF3, LF,public key}

   ICF1 is a class file for writing the log records

   ICF2 is a class file of the log harmonizer

   ICF3 is a class file for displaying or downloading the data

JAR file which is present on the cloud. The structure of master file will be as follows :

master_Index = {name of file, type of data, name of OuterJAR, name of InnerJAR}

IBE key pair (Public Key) for encrypting the log records

3) OJ ={IJ, OCF1, OCF2, OCF3, OCF4}

   OCF1: class file for authenticating the servers or the users

   OCF2: class file for finding the correct inner JAR,

   OCF3: class file which checks the JVM's validity,

   OCF4: class file is used for managing the GUI

4) Log Record

   LR=(r1,…..,rk)

   Where ri = (ID, Act, T, Loc, h ((ID, Act, T, Loc) |ri – 1 |…|r1), sig)

   Here, at time T, ri indicates an entity identified by ID has performed an action Act on the user's data at location Loc. The component h ((ID, Act, T, Loc) |ri – 1 |…|r1) corresponds to the checksum of the records preceding the newly inserted one, appended with the main content of the record itself. The signature of the record created by the server is denoted by sig component.

5) Log harmonizer = {CFs, error correction information, user's IBE decryption key}

   CFs: class files for a server and a client processes to allow the communication with its logger components.

6) Push (LR) - The logs are periodically pushed to the data owner (or auditor) by the harmonizer.

7) Pull (LR) - It allows auditors to retrieve the logs anytime when they want to check the recent access to their own data.

8) PureLog- records every access to the data.

9) AccessLog - It has two main tasks : logging actions and enforcing access control.

10) master_Index – It is the master file, which is created for performing indexing on the JAR files. The master file will contain an every

NP HARD OR NP COMPLETE:

Our project comes into the NP complete. Because in particular time it will give the result. For the decision problem, so that it will give the solution for the problem within polynomial time. The set of all decision problems whose solution can be provided into polynomial time by using the given algorithm.

4. ALGORITHM

MD5Algorithm:-

MD5 is abbreviation for Message-Digest 5 algorithm.MD5 algorithm is used for all users authentication. The MD5 is expressed in text format as 32 digit hexadecimal number. Also know as cryptographic hash function producing a 128-bit (16-byte) hash value. Wide variety of cryptographic applications is based on MD5 and is also ordinarily used to verify data integrity.

At the time of login, user name will be taken by MD5. MD5 gives a fixed-length 128bits output. Chunks of 512-bit blocks which are sixteen 32-bit words have been generated by broken of input message; the message is padded so that its length is divisible by 512. The working of padding is as follows: firstly at the end of the message, single bit 1 is appended. Then to bring the length of the message up to 64 bits less than a multiple of 512, followed by as many zeros as required. To representing the length of the original message modulo 264, the bits that are left unfilled are filled up with 64 bits. By this process it can check the authentication of that particular user.

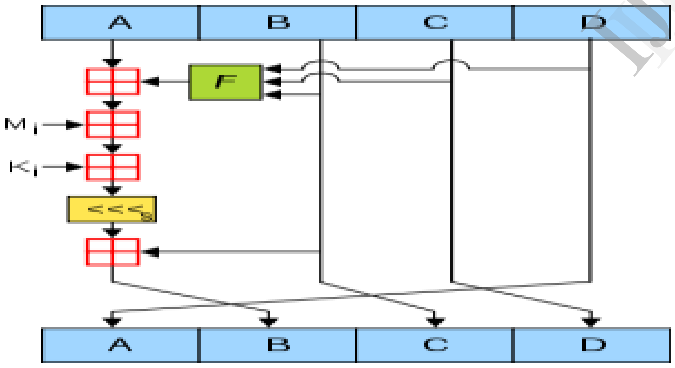

Fig.2 MD5 Algorithm

RSA ALGORITHM:-

We can use RSA algorithm as main purpose security algorithm. In this algorithm we automatically send command to the each cloud server and that command contains the JAR files assembling and dissembling also JAR file storage. This is public key encryption algorithm. To generate the public and private key based on mathematical fact and multiplying large numbers together it uses Prime number. The public key encryption takes place at inner JAR while private key encryption takes place at outer JAR respectively.

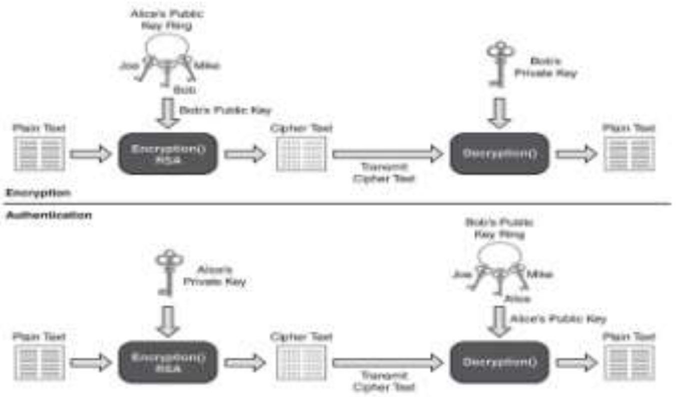

Fig.3 RSA Algorithm

DES ALGORITHM:-

We are using DES algorithm for encryption and re-encryption. It is symmetric key algorithm it can use the same keys or related keys for encryption and decryption of data. We are using one key it is given by data owner. Data owner sends keys to the data user via mail. Using this key data user decrypts the file. At a time 64 bit data is given as input and we get the 64 bit data as an output. If we used larger key size it becomes more difficult to get the encrypted data.

DES is the Data Encryption Standard algorithm for encrypting and decrypting data.

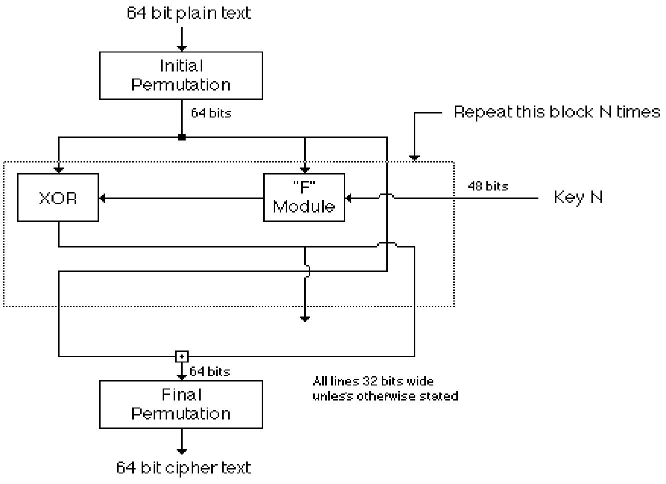

Fig.4 DES Algorithm

5. CONCLUSION

We proposed distributed accountability and auditing in cloud by automated logging mechanism. User is ensured about the safety of his data as we have used programmable JAR and data privacy. Also the data owner can monitor his data in an efficient way. This system is of great use to software engineers and to other industries as well. Specific needs are met in an efficient way and will serve the requirements of various industry people. In futures we will work on the authentication of JARs Files and to verify the integrity of the JER.

6. REFERENCES

[1] Smitha Sundareswaran, Anna C. Squicciarini, Dan Lin,‘‘Ensuring Distributed Accountability for Data Sharing in the Cloud’’,IEEE Transaction, Vol. 9, No.4, July/August 2012.

[2] Pankaj K. Singh, Ajit Kumar, R. Karthikeyan, “Ensuring Distributed Accountability for Data Sharing in the Cloud”, IJARCSSE, Vol.3, Issue 3, March 2013.

[3] Ashwini N. S., Mrs. Tamilarasi T., “Distributed Accountability and Auditing in Cloud ", International Journal of internal Computing, Vol.2, Issue 1, 2013.

[4] Aman Kumar, Dr. Sudesh Jakhar, Mr. Sunil Makkar., “Comparative Analysis between DES & RSA Algorithm’s”, IJARCSSE, Vol.2, Issue 7,July 2012.

[5] Mr.Indushree T G, Mr.Anand R, “Distributed Auditing for User Data in the Cloud”, CMRIT, Bangalore, India, IJCSMC, Vol.2, Issue 6, June 2013.

[6] M.Vanita, R. Raju,”Data Sharing: Efficient Distributed Accountability in Cloud Using Third Party Auditor”, IJITEE, Vol.2, Issue 5, April 2013.

[7] Mr.R.Kartik Ganesh, Ms. Aranya Hari, “Enhancing Privacy in Cloud by Avoiding Misuses of Files”, IJARCSSE, Vol.3, Issue 3, March 2013.