# DATA PROTECTION IN THE CLOUD BY USING JAR GENERATION

**Hanusha.K ,G.Priyanka ,Dr.S S Sridhar**

UG Student, SRM UNIVERSITY

hanusha04@gmail.com , maahipriyanka111@gmil.com , sridhar.s@ktr.srmuniv.ac.in

**ABSTRACT:** Cloud computing enables highly scalable services to be easily consumed over the Internet on an as-needed basis. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. While enjoying the convenience brought by this new emerging technology, users' fears of losing control of their own data (particularly, financial and health data) can become a significant barrier to the wide adoption of cloud services. **Methods:** To address this problem, in this paper, we propose a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. In particular, we propose an object-centered approach that enables enclosing our logging mechanism together with users' data and policies. **Findings:** We leverage 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. **Improvements:** We provide extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches.

***Keywords***: Cloud, JAR, CIA, LOGGER

## 1. 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. 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 on1. 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 environments2.

## 2. PROBLEM DEFINITION

### *Existing System*

- Data handling in the cloud goes through a complex and dynamic hierarchical service chain
- This does not exist in conventional environments.
- Ordinary web framework
- Uses web services for request and responses3

### *Disadvantages*

- No security for user‘s data. No authentication or security provided
- High resource costs needed for the implementation.
- Not suitable for small and medium level storage users[4].

### *2. 2.Proposed System*

- We propose a novel approach, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability.
- Our proposed CIA framework provides end-to end accountability in a highly distributed fashion

- It provides a detailed security analysis and discusses the reliability and strength of our architecture in the face of various nontrivial attacks by implementing Java Running Environment.
- In addition to a class file for authenticating the servers or the users, another class file finding the correct inner JAR, a third class file which checks the JVM‘s validity using oblivious hashing.
- Timer mechanism for limiting the accessing time for security purpose
- Secure JVM for making software tamper resistance capabilities to JAR file. It provides integrity, confidentiality to JAR.

### *2.2.1. Advantages*

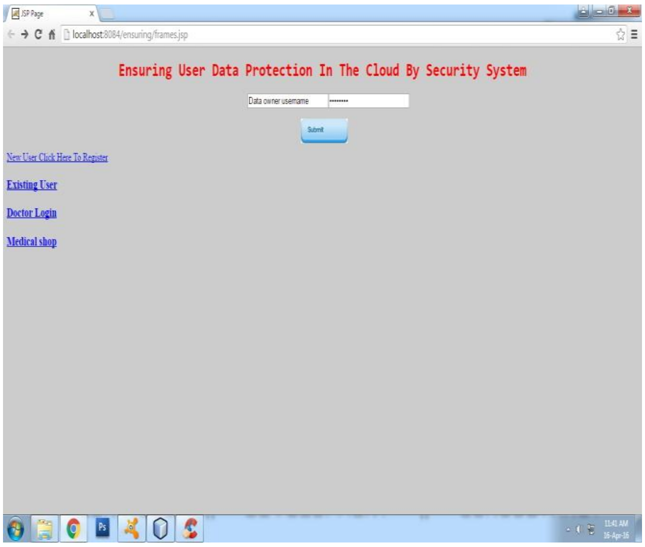
- One of the main innovative features of the CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication.
- Providing defenses against man in middle attack, dictionary attack, Disassembling Attack, Compromised JVM Attack.
- It‘s Suitable for limited and large number of storages.

## *3.* Implementation

### *JAR Generation*

- The JAR file contains a set of access control rules specifying whether and how the cloud servers and possibly other data interested party (users, companies) are authorized to access the content itself. Depending on the configuration settings defined at the time of creation, the JAR will provide usage control associated with logging, or will provide only logging functionality.
- 3.2. New user registration
- In this module, the data owner uploads their data in the cloud server. The new users can register with the service provider and create a new account and so they can securely upload the files and store it. For the security purpose the data owner encrypts the data file and then store in the cloud. The Data owner can have capable of manipulating the encrypted data file. And the data owner can set the access privilege to the encrypted data file 5-7.

## SCREENSHOTS



Ensuring User Data Protection In The Cloud By Security System

Data owner username ••••••••

JSP Page
localhost:8084/ensuring/frames.jsp
Submit
New User Click Here To Register
Existing User
Doctor Login
Medical shop

### *3. 3.Cloud service provider module*

- In Figure 1, The cloud service provider manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud with the jar file created for each file for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them

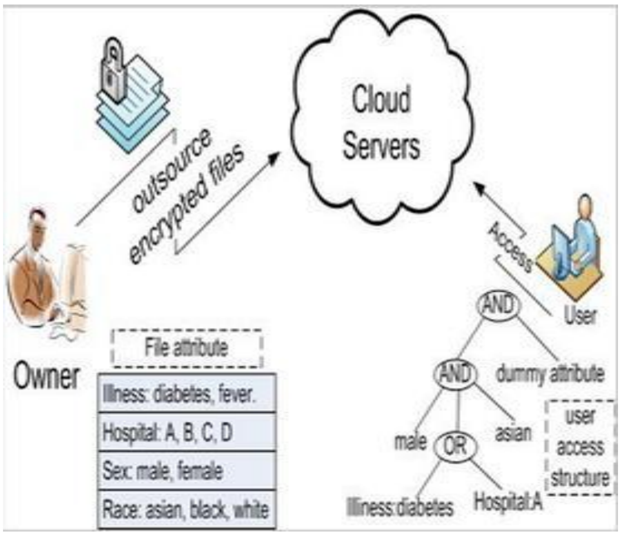


**Fig 1. Cloud Server**

### *Log Record Generation*

We leverage the programmable capability of JARs to conduct automated logging. A logger component is a nested Java JAR file which stores a user‘s data items and corresponding log files. The main responsibility of the outer JAR is to handle authentication of entities which want to access the data stored in the JAR file..

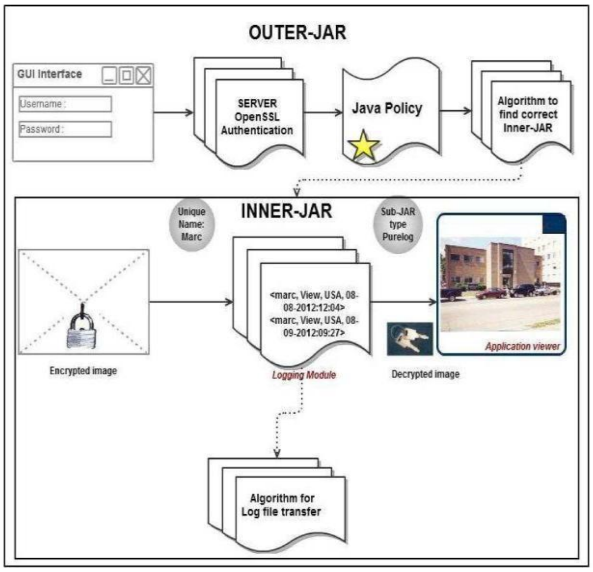
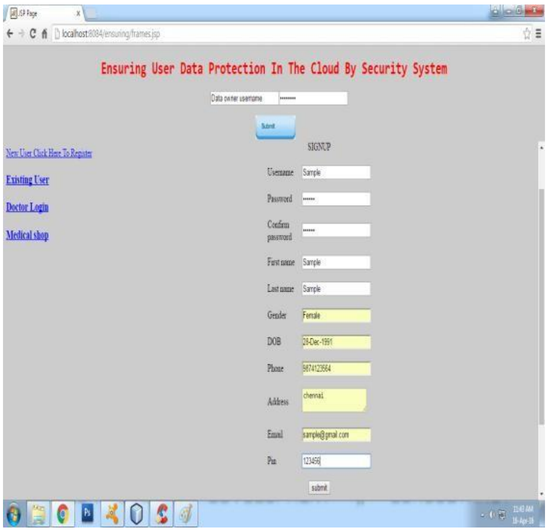

OUTER-JAR
GUI Interface
Username:
Password:
SERVER OpenSSL Authentication
Java Policy
Algorithm to find correct Inner-JAR
INNER-JAR
Unique Name: Marc
Sub-JAR type Purelog
<marc, View, USA, 08-08-2012:12:04>
<marc, View, USA, 08-09-2012:09:27>
Logging Module
Encrypted image
Decrypted image
Application viewer
Algorithm for Log file transfer

**Fig 2. Components of JAR**

The data owner can specify the permissions in user-centric terms as opposed to the usual code-centric security offered by Java, using Java Authentication and Authorization Services. Moreover, the outer JAR is also in charge of selecting the correct inner JAR according to the identity of the entity who requests the data is represented in Figure 2 .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 sequentially8**SCREENSHOTS**



Ensuring User Data Protection In The Cloud By Security System

Data owner username ••••••••

New User Click Here To Register
Existing User
Doctor Login
Medical shop

SIGNUP

Username Sample
Password ••••••
Confirm password ••••••
First name Sample
Last name Sample
Gender Female
DOB 28-Dec-1991
Phone 9874123564
Address chennai
Email sample@gmail.com
Pin 123456

submit

### *Mode Setting*

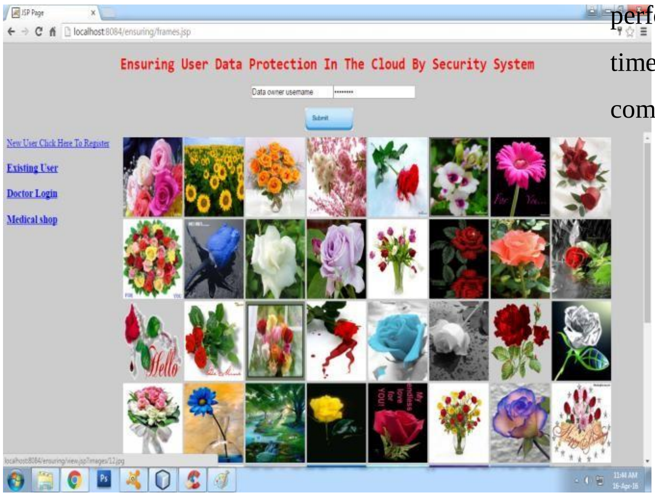
To allow users to be timely and accurately informed about their data usage, our distributed logging mechanism is complemented by an innovative auditing mechanism. We support two complementary auditing modes: 1) push mode; 2) pull mode.

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 a 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.

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

## SCREENSHOTS


Ensuring User Data Protection In The Cloud By Security System

Data owner username ••••••••

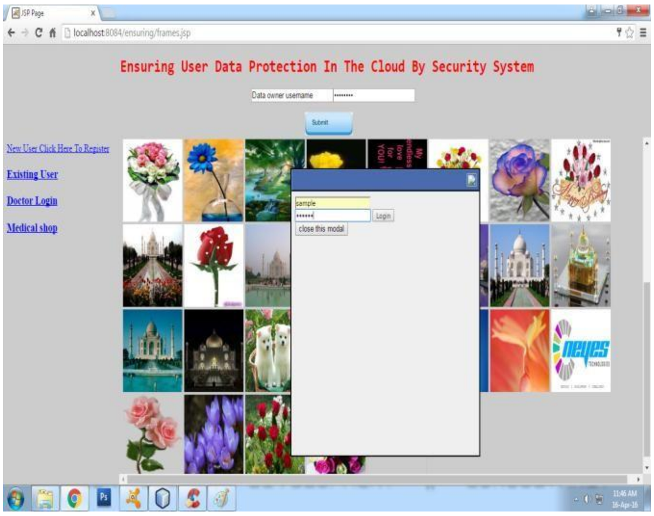
Submit

New User Click Here To Register

Existing User

Doctor Login

Medical shop


Ensuring User Data Protection In The Cloud By Security System

Data owner username ••••••••

Submit

New User Click Here To Register

Existing User

Doctor Login

Medical shop

sample ••••••• Login

close this modal

## 4. Technique in Proposed System

There are two major components of the CIA, the first being the logger, and the second being the log harmonizer.The log harmonizer forms the central component which allows the user access to the log files. The logger is strongly coupled with users data.

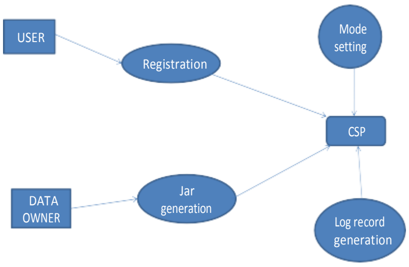
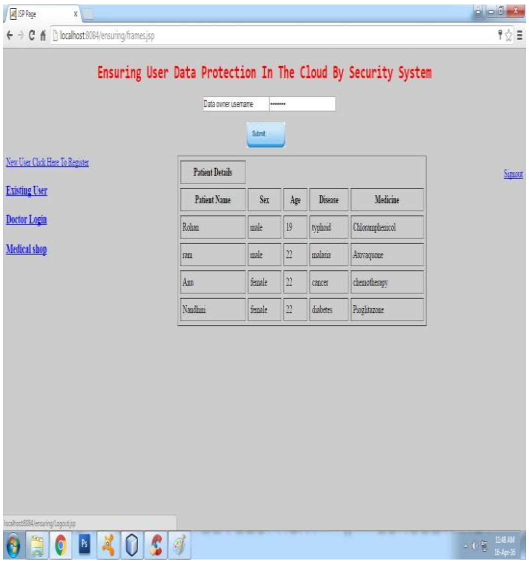


**Figure 3.Data Flow Diagram**

The Cloud Information Accountability (CIA) framework proposed in this work conducts automated logging and distributed auditing of relevant access performed by any entity, carried out at any point of time at any cloud service provider. It has two major components: logger and log harmonizer.

- There are two major components of the CIA, the first being the logger, and the second being the log harmonizer.
- The logger is the component which is strongly coupled with the user‘s data, so that it is downloaded when the data are accessed, and is copied whenever the data are copied. It handles a particular instance or copy of the user‘s data and is responsible for logging access to that instance or copy.
- The log harmonizer forms the central component which allows the user access to the log files. The logger is strongly coupled with user‘s data (either single or multiple data items). Its main tasks include automatically logging access to data items that it contains, encrypting the log record using the public key of the content owner, and periodically sending them to the log harmonizer.

## SCREENSHOTS


Ensuring User Data Protection In The Cloud By Security System

Data owner username ••••••••

Submit

New User Click Here To Register

Existing User

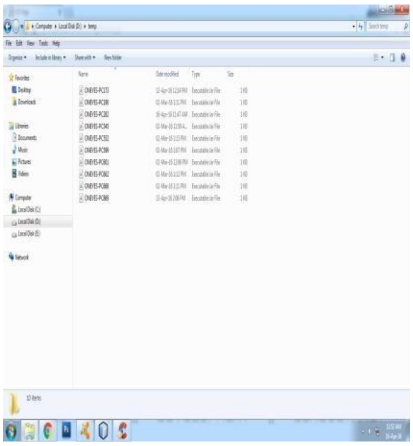
Doctor Login

Medical shop

Signout

Patient Details

| Patient Name | Sex | Age | Disease | Medicine |
|---|---|---|---|---|
| Rohan | male | 19 | typhoid | Chloramphenicol |
| ram | male | 22 | malaria | Atovaquone |
| Ann | female | 22 | cancer | chemotherapy |
| Nandhini | female | 22 | diabetes | Pioglitazone |



| Name | Date modified | Type | Size |
|---|---|---|---|
| ONEYES-PC173 | 13-Apr-16 12:34 PM | Executable Jar File | 1 KB |
| ONEYES-PC198 | 03-Mar-16 1:31 PM | Executable Jar File | 1 KB |
| ONEYES-PC202 | 16-Apr-16 11:47 AM | Executable Jar File | 1 KB |
| ONEYES-PC345 | 03-Mar-16 11:58 A... | Executable Jar File | 1 KB |
| ONEYES-PC352 | 02-Mar-16 1:15 PM | Executable Jar File | 1 KB |
| ONEYES-PC398 | 02-Mar-16 1:07 PM | Executable Jar File | 1 KB |
| ONEYES-PC801 | 03-Mar-16 12:06 PM | Executable Jar File | 1 KB |
| ONEYES-PC862 | 02-Mar-16 1:12 PM | Executable Jar File | 1 KB |
| ONEYES-PC888 | 03-Mar-16 1:31 PM | Executable Jar File | 1 KB |
| ONEYES-PC889 | 13-Apr-16 2:08 PM | Executable Jar File | 1 KB |

10 items

### *4.1.Algorithm in proposed system*

- Here we are using Advanced Encryption Standard to encrypt and decrypt the data. The Advanced Encryption Standard (AES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001.

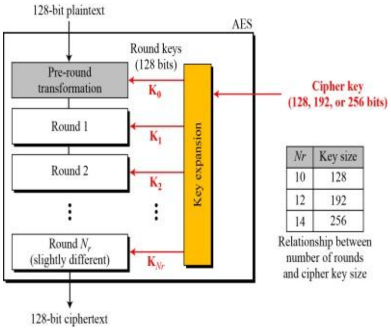
- The criteria defined by NIST for selecting AES fall into three areas:

- 1. Security

- 2. Cost

- 3. Implementation

- AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the number of rounds.



| Nr | Key size |
|---|---|
| 10 | 128 |
| 12 | 192 |
| 14 | 256 |

Relationship between number of rounds and cipher key size

**Figure 4.AES Encryption**

## 5.Conclusion

- This system 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. Ammann and S. Jajodia, –Distributed Timestamp Generation in Planar Lattice Networks,‖ ACM Trans. Computer Systems, vol. 11, 1993 Aug. pp. 205-225.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, –Provable Data Possession at Untrusted Stores,‖ Proc. ACM Conf. Computer and Comm. Security, 2007, pp. 598- 609.

[3] E. Barka and A. Lakas, –Integrating Usage Control with SIP-Based Communications,‖ J. Computer Systems, Networks, and Comm., vol. 2008, pp. 1-8.

[4] D. Boneh and M.K. Franklin, –Identity-Based Encryption from the Weil Pairing,‖ Proc. Int'l Cryptology Conf. Advances in Cryptology, 2001, pp. 213-229.

[5] R. Bose and J. Frew, –Lineage Retrieval for Scientific Data Processing: A Survey,‖ ACM Computing Surveys, vol. 37, 2005 Mar. pp. 1- 28.

[6] P. Buneman, A. Chapman, and J. Cheney, –Provenance Management in Curated Databases,‖ Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '06), 2006, pp. 539-550

[7].Yan Zhu,From RBAC to ABAC: Constructing lexible DataAccess Control for Cloud Storage Services, IEEE transactions on services computing, vol. 8, 2015 July, No 4

[8]Lan Zhou,–Trust Enhanced Cryptographic Role-Based Access Control for Secure Cloud Data Storage‖, IEEE transactions on information forensics and security, vol. 10, 2015 November, pp 11.