**A SEMINAR REPORT ON**

# Distributed Accountability for Data Sharing in Cloud


*Submitted By*

***LAKSHMIPRIYA R***

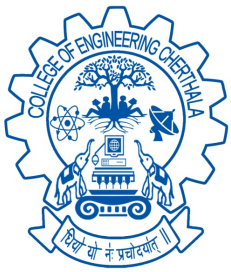
**Reg. No . CEC17CS035**

*under the esteemed guidance of*

*Mrs. ANITHA M*

*Assistant Professor*

*Computer Science & Engineering*


**APRIL 2021**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**COLLEGE OF ENGINEERING,CHERTHALA**
**PALLIPPURAM P O, ALAPPUZHA-688541,**
**PHONE: 0478 2553416, FAX: 0478 2552714**
**http://www.cectl.ac.in**

**A SEMINAR REPORT ON**

# Distributed Accountability for Data Sharing in Cloud

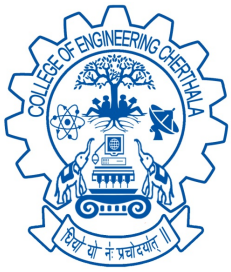
*Submitted By*

***LAKSHMIPRIYA R (Reg. No . CEC17CS035)***

*under the esteemed guidance of*

*Mrs. ANITHA M*

*In partial fulfillment of the requirements for the award of the degree*
*of*
*Bachelor of Technology*
*in*
*Computer Science and Engineering*
*of*
*Cochin University Of Science And Technology*




**APRIL 2020**

**Department of Computer Science and Engineering**
**College of Engineering, Pallippuram P O, Cherthala,**
**Alappuzha Pin: 688541,**
**Phone: 0478 2553416, Fax: 0478 2552714**
**http://www.cectl.ac.in**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**COLLEGE OF ENGINEERING CHERTHALA**

**ALAPPUZHA-688541**

# C E R T I F I C A T E

This is to certify that, the seminar report titled ***DISTRIBUTED ACCOUNTABILITY FOR DATA SHARING IN CLOUD*** is a bonafide record of the **CS 17L4 SEMINAR** presented by **LAKSHMIPRIYA R** (Reg.No.CEC17CS035), Seventh Semester B. Tech. Computer Science & Engineering student, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **B. Tech. Computer Science & Engineering** of **Cochin University of Science & Technology**.

| **Guide** | **Co-ordinator** | **HoD** |
|---|---|---|
| **Mrs. Anitha M** | **Mr. Jayakrishnan** | **Dr. Priya S** |
| Assistant Professor | Assistant Professor | Professor |
| Computer Science & Engg. | Computer Science & Engg. | Computer Science & Engg. |

# ACKNOWLEDGEMENT

This work would not have been possible without the support of many people. First and foremost, I give thanks to Almighty God who gave me the inner strength, resources, and ability to complete my seminar successfully.

I would like to thank **Dr. Mini M G**, The Principal, who has provided with the best facilities and atmosphere for the seminar completion and presentation. I would also like to thank HoD **Dr. Priya S** (Professor, Computer Science and Engineering) and my seminar guide **Mr. Muhammed Ilyas H** (Assistant Professor, Computer Science and Engineering), my seminar coordinator **Mr. Jayakrishnan** (Assistant Professor, Computer Science and Engineering) for the help extended and also for the encouragement and support are given to me while doing the seminar.

I would like to thank my dear friends for extending their cooperation and encouragement throughout the seminar work, without which I would never have completed the seminar this well. Thank you all for your love and also for being very understanding.

# ABSTRACT

Cloud computing is a technology, which uses the internet and remote servers to stored data and applications. Cloud computing provides on-demand services. Cloud computing technology provides advantages to end-users and business organizations. Few notable advantages are cost efficiency, increased storage capacity, backup and recovery, continuous resource availability, and location independence. Data owners host their private data in the cloud and worry about unauthorized access to their data. While the data owner will store his/her data on the cloud, he must get confirmation that his/her data is safe on the cloud. Any unauthorized users accessing the owner's private data leads to accountability issues.

To solve the above problem in this paper we provide an effective mechanism to track usage of data using accountability. We design a trusted monitoring framework, which provides a chain of trust that excludes the untrusted privileged domain, as well as utilizing the trusted computing technology to ensure the integrity of the monitoring environment. Accountability is checking of authorization policies and it is important for transparent data access. We provide automatic logging mechanisms using JAR programming which improves the security and privacy of data in the cloud. Using this mechanism data owner may know his/her data is handled as per his requirement or service level agreement.

Though the Cloud Service Provider (CSP) gives the privacy and veracity of the data; this tracks the actual usage of the user's data in the cloud by using novel extremely decentralized framework data and policies. An object-centered approach that enables enclosing our logging mechanism together with users' data and policies. By leveraging the JAR programmable aptitude to both create a dynamic and peripatetic object, it focuses on transparent data usage.
.

**Keywords–** ***Cloud computing, trusted monitoring framework, Security, Chain of trust.***

# Contents

# List of Figures

# Chapter 1

# INTRODUCTION

Cloud computing is a technology which uses the internet and remote servers to store data and applications. In the cloud there is no need to install particular hardware, software on the user machine, so the user can get the required infrastructure on his the machine in cheap charges/rates. Cloud computing is an infrastructure which provides useful, on-demand network services to use various resources with less effort. Features of Cloud computing is, huge access to data, application, resources and hardware without installation of any software, user can access the data from any machine or anywhere in the world, a business can get resource in one place, that's means cloud computing provides scalability in on-demand services to the business

Cloud Computing technology provides advantages to end-users and business organizations. Few notable advantages are cost efficiency, increased storage capacity, backup, and recovery, continuous resource availability and location independence.

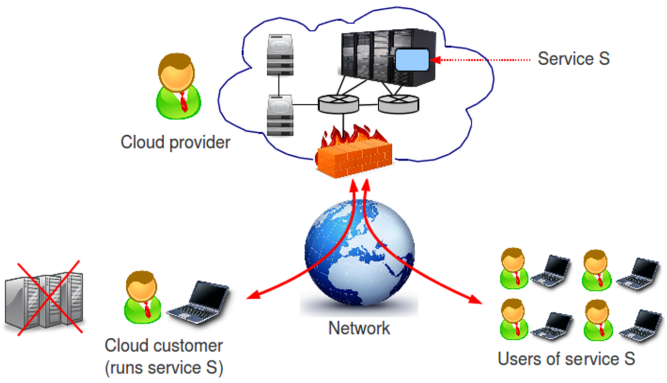



Fig. 1.1: CLOUD COMPUTING EXAMPLE

# Chapter 2

# CLOUD COMPUTING AND ACCOUNTABILITY

## 2.1 CLOUD STORAGE: FEAR OF USERS

Cloud computing is scalable services. Cloud computing is a computing platform that resides in a large data center and can dynamically provide servers the ability to address a wide range of needs, ranging from scientific research to e-commerce. Cloud computing is expanding rapidly as a service used by a great many individuals and organizations internationally, policy issues related to cloud computing. Details of the services provided are abstracted from the users who no longer need to be experts of technology infrastructure

Despite these advantages, the biggest issue with the cloud is the "Security. Though there are several advantages with cloud, it also imposes several security threats related to outsourced user's data. Since private data is hosted in the cloud and they are being processed at remote machines and are administered by the cloud service providers (CSP), the users are worried about the loss of data control in the cloud. There are various reasons for the CSP to involve in unfaithful disclosure or leakage of user's data to any external entity that may turn out to be a serious privacy and security concern for any user towards his/her data. Cloud Users do not know the machines where their data are processed, so they start bothering about losing control over their data. There are no specific mechanisms to check if the service level agreements made between the data owner and the end-users have been preserved or not. Data is often

being outsourced in the cloud, leading to accountability issues and manipulation of personally identifiable information.

Therefore despite being introduced to cloud storage facilities, statistics state that only 10 percent of the total business population is only using the cloud. which thus still leads to the above fears. since nowadays many similar products are available if the storage is not safe, then it might lead to misuse of private data even if they do not use the same, ideas can be misused, which leads to serious accountability issues.

## 2.2 MODELS OF CLOUD STORAGE

### 2.2.1 INFRASTRUCTURE AS A SERVICE (IAAS)

IAAS is a form of cloud computing that provides virtualized computing resources over the internet. In an IAAS model, a third-party provider hosts hardware, software, servers, storage, and other infrastructure components on the behalf of its users. IAAS providers also host users' applications and handle tasks including system maintenance backup and resiliency planning. IAAS platforms offer highly scalable resources that can be adjusted on-demand which makes it a well-suited for workloads that are temporary, experimental, or change unexpectedly. Other characteristics of IAAS environments include the automation of administrative tasks, dynamic scaling, desktop virtualization, and policy-based services. Other characteristics of IAAS include the automation of administrative tasks, dynamic scaling, desktop virtualization, and policy-based services.

Technically, the IaaS market has a relatively low barrier of entry, but it may require substantial financial investment to build and support the cloud infrastructure. Mature open-source cloud management frameworks like OpenStack are available to everyone and provide strong a software foundation for companies that want to build their private cloud or become a public cloud provider.

### 2.2.2 PLATFORM AS A SERVICE

Platform as a Service (PAAS) is a cloud computing model that delivers applications over the internet. In a PAAS model, a cloud provider delivers hardware and software tools, usually those needed for application development, to its users as a service. A PAAS provider hosts the hardware and software on its infrastructure. As a result, PAAS frees users from having to install in-house hardware and software to develop or run a new application.

PAAS doesn't replace a business's entire infrastructure but instead, a business relies on PAAS providers for key services, such as Java development or application hosting. A PAAS provider, however, supports all the underlying computing and software; users only need to log in and start using the platform-usually through a Web browser interface. PAAS providers then charge for that access on a per-user basis or a monthly basis.

### 2.2.3 SOFTWARE AS A SERVICE

Software as a Service(SAAS) is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. SAAS has become an increasingly prevalent delivery model as underlying technologies that support Web services and service-oriented architecture (SOA) mature and new development approaches, such as Ajax, become popular. SAAS is closely related to the ASP (Application service provider) and on-demand computing software delivery models. IDC identifies two slightly different delivery models for SAAS namely the hosted application model and the software development model. Some of the core benefits of using the SAAS model are:

- Easier administration.
- automatic updates and patch management.
- compatibility: all users will have the same version of the software.
- easier collaboration, for the same reason.
- global accessibility.

## 2.3 ACCOUNTABILITY IN CLOUD

To solve the security issues in the cloud; other users can't read the respective users' data without having access. The data owner should not bother about his data, and should not get fear about the damage of his data by the hacker; there is a need for a security mechanism that will track the usage of data in the cloud. Accountability is necessary for monitoring data usage, in this all actions of users like sending of the file are cryptographically linked to the server, that performs them, and the server maintain a secured record of all the actions of past and the server can use the records to know the correctness of the action. It also provides reliable information about the usage of data and it observes all the records, so it helps in making trust, relationship, and reputation. So accountability is for verification of authentication and authorization. It is a powerful tool to check the authorization policies. Accountability describes authorization requirements for data usage policies. Accountability mechanisms, which rely on upon after the fact verification, are an attractive means to enforce authorization policies

### 2.3.1 PHASES OF ACCOUNTABILITY

There are 7 phases of accountability

- Policy setting with data of data by users
- Logging
- Merge logs
- Error correctness in log
- Auditing
- Rectify and improvement.

These phases may change as per framework First the data owner will set the policies with data and send it to cloud service provider (CSP), data will be use by users and logs of each record will be created, then log will be merged and error correction in log has been done and in auditing logs are checked and in last phase improvement has been done

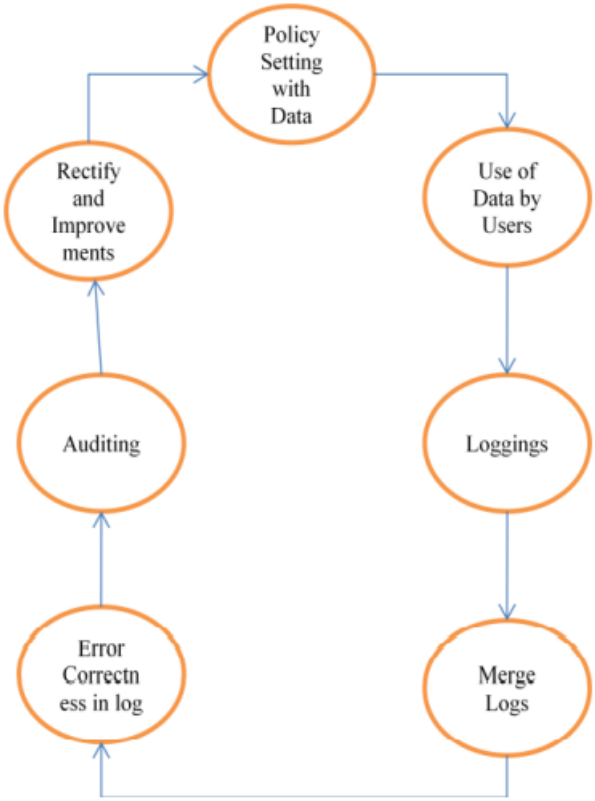



Fig. 2.1: PHASES OF ACCOUNTABILITY

# Chapter 3

# EXISTING SYSTEM

## 3.1 PRIVACY MANAGER MECHANISM:

privacy manager mechanism in which user's data is safe on the cloud, in this technique the user's data is in encrypted form in the cloud and evaluating is done on encrypted data, the privacy manager make readable data from the result of the evaluation manager to get the correct result. In obfuscation, data is not present on the Service provider's the machine so there is no risk with data, so data is safe on the cloud, But this solution is not suitable for all cloud application, when input data is large this method can still require a large amount of memory

## 3.2 THREE TIER MECHANISM:

In this mechanism these policies are decided by the parties that use, store or share that data irrespective of the jurisdiction in which information is processed. But it has the limitation that data processed on SP is in unencrypted at the point of processing so there is a risk of data leakage, the author gives a the language which permits to serve data with policies by the agent; the agent should prove their action and authorization to use particular data. In this logic, data owner attach Policies with data, which contain a description of which actions are allowed with which data, but there is the problem of Continuous auditing of the agent, but they provide a solution that incorrect behavior. Should monitor and agent give justification for their action, after that authority will check the justification.

It gives a three-layer architecture which protects information leakage from the cloud, it provides three-layer to protect data, in the first layer the service provider should not view confidential data in second layer service provider should not do the indexing of data, in the third layer user specify the use of his data and indexing in policies, so policies always travel with data.It present accountability in a federated system to achieve trust management. The trust in the use of resources are accomplished through accountability so to resolve the problem for trust management in the federated system they have given three layers of architecture, in the first layer is the authentication and authorization in this authentication does use public key cryptography. The second layer is accountability which performs monitoring and logging. The third layer is anomaly detection which detects misuse of resources. This mechanism requires third party services to observe network resources

## 3.3 VIRTUALIZATION

Virtualization is a pillar technology in cloud computing for multiplexing computing resources on a single cloud platform for multiple cloud tenants. Monitoring the behavior of virtual machines (VMs) on a cloud platform is a critical requirement for cloud tenants. Existing monitoring mechanisms on virtualized platforms either take a complete VM as the monitoring granularity, such that they cannot capture the malicious behaviors within individual VMs, or they focus on specific monitoring functions that cannot be used for heterogeneous VMs concurrently running on a single cloud node. Furthermore, the existing monitoring mechanisms have made an assumption that the privileged domain is trusted to act as expected, which causes the cloud tenants' concern about security because the privileged domain in fact could not act as the tenants' expectation. We design a trusted monitoring framework, which provides a chain of trust that excludes the untrusted privileged domain, by deploying an independent guest domain for the monitoring purpose, as well as utilizing the trusted computing technology to ensure the integrity of the monitoring environment. The user is more concerned with the privacy and security issues in the cloud environment.

## 3.4 PROBLEM DEFINITION:

Cloud Computing is a technology that provides network-based services on-demand. Data owners host their private data in the cloud and worry about unauthorized access to their data. They feel uncomfortable about any user misusing their private data. This insecure feeling of data owners holds them back from using cloud services. Any unauthorized users accessing the owner's private data leads to accountability issues. To solve the accountability issue, a mechanism to monitor the actual data usage is proposed. This approach grants access rights to users based on their role and also monitors every access to the owner's data, verifying that the service level agreements have been violated or not.

Though there are several advantages with cloud, it also imposes several security threats related to outsourced user's data. There are various reasons for the CSP to involve in unfaithful disclosure or leakage of user's data to any external entity that may turn out to be a serious privacy and security concern for any user towards his/her data

The data processed on clouds are often outsourced, leading to several issues related to accountability, including the handling of personally identifiable information. It is necessary 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 according to the service level agreements made at the time. They sign on for services. Conventional access control approaches developed for closed domains such as Databases and operating systems, or approaches using a centralized server in distributed environments, are not suitable, due to the following features characterizing cloud environments. First, data handling can be outsourced by the direct cloud service provider (CSP) to other entities in the Cloud and these entities can also delegate tasks to others, and so on. Outsourcing of data processing invariably raises governance and accountability questions. Second, entities are allowed to join and flexibly leave the cloud. As a result, data handling in the cloud goes through a complex and dynamic hierarchical service chain which does not exist in conventional environments.

## 3.5 SUGGESTIONS BY AUTHORS:

### 3.5.1 TOWARD PUBLICLY AUDITABLE SECURE CLOUD DATA STORAGE SERVICES

publicly auditable cloud data storage can help this nascent cloud economy become fully established. With public audit ability, a trusted entity with expertise and capabilities data owners do not possess can be delegated as an external audit party to assess the risk of outsourced data when needed. Such an auditing service not only helps save data owners' computation resources but also provides a transparent yet cost-effective method for data owners to gain trust in the cloud. The author describes approaches and system requirements that should be brought into consideration, and outline challenges that need to be resolved for such a publicly auditable secure cloud storage service to become a reality.

### 3.5.2 ONLINE DATA STORAGE USING IMPLICIT SECURITY

implicit security architecture suited for the application of online storage. In this scheme, data is partitioned in such a way that each partition is implicitly secure and does not need to be encrypted. These partitions are stored on different servers on the network which are known only to the user. Reconstruction of the data requires access to each server and the knowledge as to which servers the data partitions are stored.

# Chapter 4

# PROPOSED MODEL

## 4.1 CLOUD DATA ACCOUNTABILITY FRAMEWORK

There is a need to provide a technique that will audit data in the cloud. Based on accountability, we proposed one mechanism which keeps the use of data transparent means data owner should get information about the usage of his data. This mechanism support accountability in a distributed environment Data owner should not bother about his data, he may know his data is handled according to service level agreement and his data is safe on the cloud. The data owner will decide the access rules and policies and the user will handle data using this rule and logs of each data access have been created. In this mechanism, there are two main components i.e. logger and log harmonizer.

The logger is with the data owner's data, it provides logging access to data and encrypts log record by using the public the key which is given by the data owner and send it to log harmonizer. The log harmonizer is performing the monitoring, rectifying, and merging. The logs are decrypted by the log harmonizer using the master key for integrity verification of logs and send to the data owner. In this mechanism, the data owner will create the private key and public key, using the generated key owner will create a logger which is a JAR file (JAVA Archives), it includes his policies like access policies and logging policies with data sent to the cloud service provider. .

These access control policies help the CSP in authenticating the user and also in granting access rights to the users. These access control policies are fixed by the data owners based

on the user's role to be played in accessing the cloud data. The users are authenticated by the cloud service provider and the requested JAR file access will be granted to the user based on the previously specified access control rights based on the user's role, whereas Authentication of cloud service provider has been done using open SSL based certificates. To certify the CSP to the JAR (FIG.1) we tend to use open SSL- primarily based certificates, whereby a trustworthy certificate authority certifies the CSP. Within the event that the access is requested by a user, we tend to use SAML-based authentication, whereby a reliability identity provider tribulations certificates confirmative the user's identity supported his username automated logging mechanism is initiated for every data access and the log generator begins to generate log records for every data access in the cloud

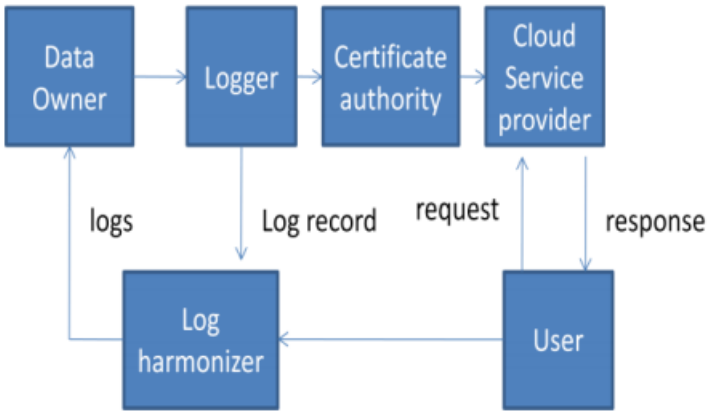



Fig 2: Accountability Mechanism in cloud

Fig. 4.1: ACCOUNTABILITY MECHANSIM

working of accountability mechanism in the cloud is given in this when user will access data then log of each access is created by the logger and periodically sent to log harmonizer, log harmonizer send this logs to the data owner and the data owner can see logs and take appropriate action if he wants

## 4.2 INDIVIDUAL MODES FOR AUDITING

The two auditing modes help the data owner to monitor the usage of his/her data hosted in the cloud.

### 4.2.1 PUSH MODE

The push mode refers to logs being periodically sent to the data owner or stakeholder. the size of the log record and the maximum time that should elapse before dumping of log files are fixed by the data owner. If anyone of these condition occurs, logs are being sent to the data owner

### 4.2.2 PULL MODE

Pull mode refers to an alternative approach whereby the user (Or another authorized party) can retrieve the logs as needed. the request mode can be used by the data owners if he/she suspects of data being misused. This mode helps the owner to monitor the data usage immediately

## 4.3 JAR FILE STRUCTURE

### 4.3.1 OUTER JAR

The logger is a nested JAR file storing user data and respective log files. The outer JAR contains access control policies for each user based on their role and is responsible for user authentication for a particular JAR file based on the specified access rights. The outer JAR may contain one or more inner JARs. The outer JAR helps in identifying the correct inner JAR based on user requests.The outer jar is responsible for validating JVM too

### 4.3.2 INNER JAR

The inner JAR consists of the data in encrypted form, class files initiating log generation, displaying data in required format and generation of log file for each encrypted data

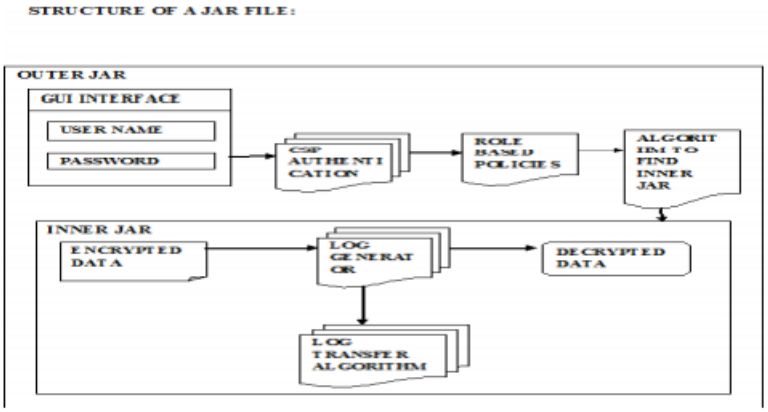



Fig 2 Structure of the JAR File

Fig. 4.2: JAR FILE STRUCTURE

## 4.4 FLOW OF DATA

The figure tells the design of the entire system where a jar file is created with all kinds of rights of the users who are registered. It also explains the way by which the cloud user could interact with the system, where the user rights are granted only if the rule is present in the protected jar file

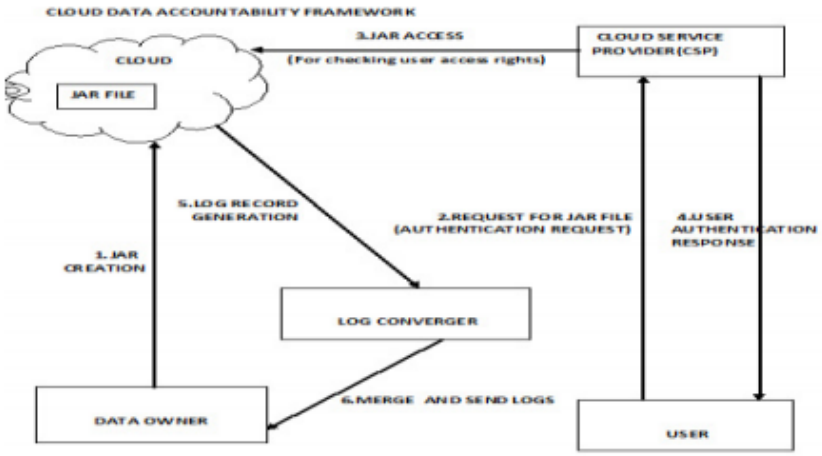



Fig 1 Cloud data accountability framework

Fig. 4.3: FRAMEWORK

## 4.5 TECHNIQUES PROPOSED WITH LOGGING AND AUDITING

The logging should be decentralized to adapt to the dynamic nature of the cloud

- Every access to the user's data should be correctly and automatically logged.
- Log files should be reliable and tamper-proof to avoid illegal insertion, deletion, and modification by malicious parties.
- Log files should be sent back to their data owners periodically to inform them of the current usage of their data.
- The proposed technique should not intrusively monitor data recipients' systems

## 4.6 MODULE RELATED TO DATA PROPRIETOR

In this module, the data proprietor 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 proprietor can have capable of manipulating the encrypted data file. And the data owner can set the access privilege to the encrypted data file. To allay 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 according to the service level agreements made at the time they sign on for services in the cloud

## 4.7 MODULE RELATED TO JAR FILE CREATION

we create the jar file for every file upload. The user should have the same jar file to download the file. This way the data is going to be secured. The logging should be decentralized to adapt to the dynamic nature of the cloud. More specifically, log files should be tightly bonded with the corresponding data being controlled, and require minimal infrastructural support from any server. Our basic approach toward addressing these issues is to leverage and extend the

programmable capability of JAR (Java Archives) files to automatically log the usage of the users' data by any entity in the cloud.

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. JAR will provide user control associated with logging (or) will provide only logging associated with logging functionality. Users will send their data along with any policies such as access control policies and logging policies that they want to enforce, enclosed in JAR files, to cloud service providers. Any access to the data will trigger an automated and authenticated logging mechanism local to the JARs. Every access to the user's data should be correctly and automatically logged. This requires integrated techniques to authenticate the entity that accesses the data, verify, and record the actual operations on the data as well as the time that the data have been accessed. Log files should be reliable and tamper-proof to avoid illegal insertion, deletion, and modification by malicious parties. Recovery mechanisms are also desirable to restore damaged log files caused by technical problems. The proposed technique should not intrusively monitor data recipients' systems, nor it should introduce heavy communication and computation overhead, which otherwise will hinder its feasibility and adoption in practice.

## 4.8 CLOUD SERVICE PROVIDER MODULE

The cloud service provider manages a cloud to provide data storage services. 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

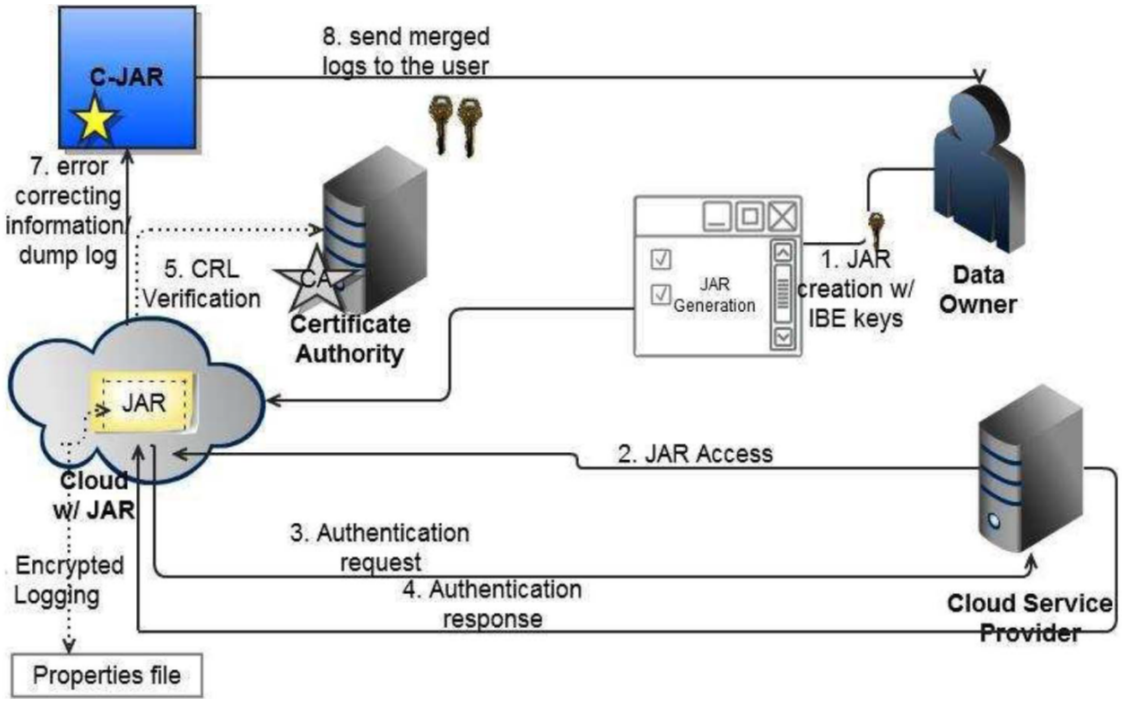



Fig. 4.4: FLOW OF DATA

## 4.9 TRANSITION DIAGRAM

- 0: Unsuccessful
- 1: Successful
- Transition are:
- S0: Data Owner will send data to the logger.
- S1: Data Owner will create a logger which is a jar
- file to store data and policies.
- S2: Authentication of CSP to JAR file.
- S3: Authentication of the user.
- S4: owner can see merge log

- Input: = 0, 1
- Representation of
- A= (S0, S1 , S2 , S3 , S4, 0, 1, , S0 , S4 )
- Input given 11011011
- Expected output
- (S0,1) = S1
- (S1,1) = S2
- (S2,1) = S3
- (S3,1) = S4
- (S4,1) = S0

In accountability mechanisms the log records are generated as access of data in jar happened then it create log record log rec (Lr).

- Lr = r1, r2, r3, r4... rk. item Parameters uses for log record are rk = ( id, action, T, loc, h((id, action, T, loc)ri-1...r1), sig ) Where,
- rk = log record
- id = user identification
- action = perform on user's data
- T = Time at location loc
- loc = Location
- h((id, action, T, loc)ri-1...r1) = checksum component
- sig = Signature of record by server

Checksum of each record is calculated and it is stored with data. Checksum is computed using hash function

- H[i] = f(H[i  1] ,m[i]),
- Where,
- Compression function is f = 0, 1 n x 0, 1b 0,1n H[i] = hash value of ith log record [10], [11].

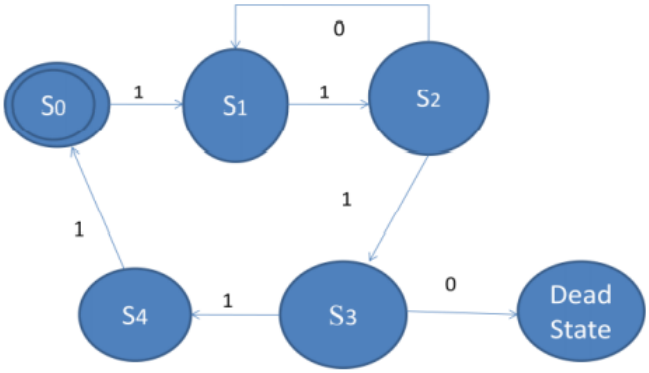



Fig 3: State Transition Diagram

Fig. 4.5: TRANSITION DIAGRAM

# Chapter 5

# POSSIBLE ATTACKS

## 5.1 COPYING ATTACK

this forms “strong binding” since the policies and the logging mechanism travel with the data. This strong binding exists even when copies of the JARs are created; thus, the user will have control over his data at any location. Such a decentralized logging mechanism meets the dynamic nature of the cloud but also imposes challenges on ensuring the integrity of the logging The Jar copy attack copies the JAR files and assumes that it will allow him to access the JAR file data without being noticed by the data owner. But every access to a JAR file creates a log record and being sent to the data owner for auditing. Even for additional copies of the JAR, log records will be generated and sent to the data owner.

If any copy of a JAR file is moved to a location that could not be accessed by the log harmonizer, . To cope with this issue, we provide the JARs with a central point of contact which forms a link between them and the user. It records the error correction information sent by the JARs, which allows it to monitor the loss of any logs from any of the JARs. Moreover, if a JAR is not able to contact its central point, any access to its enclosed data will be denied.

## 5.2 DIASSEMBLING ATTACK

A disassembling attack is another attack possible in our scenario. Once the Jar files have been obtained by the hacker, he/she tries to disassemble the JAR file. But the JAR file contains

data and logs in an encrypted format. The hacker knows only the public key used for encrypting the logs, there are no possibilities to obtain the master key used for decrypting the logs. The attacker cannot modify the log file content after disassembling the JAR.the attacker will not be able to decrypt any data or log files in the disassembled JAR file. Even if the attacker is an authorized user, he can only access the actual content file but he is not able to decrypt any other data including the log files which are viewable only to the data owner.

From the disassembled JAR files, the attackers are not able to directly view the access control policies either, since the source code is not included in the JAR files. If the attacker wants to infer access control policies, the only possible way is through analyzing the log file. This is, however, very hard to accomplish since, as mentioned earlier, log records are encrypted, and breaking the encryption is computationally hard. Also, the attacker cannot modify the log files extracted from a disassembled JAR. Would the attacker erase or tamper a record, the integrity checks added to each record of the log will not match at the time of verification, revealing the error. Similarly, attackers will not be able to write fake records to log files without going undetected, since they will need to sign with a valid key and the chain of hashes will not match.

The integrity checking mechanism for logs will detect any modification in the log records since the integrity checks added to each record will not match at the time of verification by the log Harmonizer. The Reed-Solomon encoding is for this checking mechanism using which the log harmonizer can easily detect corrupted logs.

## 5.3 MAN-IN-THE-MIDDLE ATTACK

an attacker may intercept messages during the authentication of a service provider with the certificate authority, and reply to the messages to masquerade as a legitimate service provider. There are two points in time that the attacker can replay the messages. One is after the actual service provider has completely disconnected and ended a session with the certificate authority. The other is when the actual service provider is disconnected but the session is not over, so the attacker may try to renegotiate the connection. The first type of attack will not succeed since the certificate typically has a timestamp which will become obsolete at the time point of reuse.

The second type of attack will also fail since renegotiation is banned in the latest version of open SSL and cryptographic checks have been added.

## 5.4 COMPROMISED JVM ATTACK

here the attacker tries to compromise the JVM. Our attack is against a JVM that permits untrusted code to execute after it has used its bytecode verifier to check that the code is type-safe and therefore respects its interfaces. The goal of our attack applet1 is to obtain two pointers of incompatible types that point to the same location. This permits circumvention of the Java type system. Once the type system is circumvented, it is straightforward to write a function that reads and writes arbitrary memory locations in the program address space, and hence executes arbitrary code s The attack works by sending the Java Virtual Machine a program (which the JVM will type-check using the bytecode verifier) and waiting for a memory error. The program type-checks; when it runs, it arranges the memory so that memory errors allow it to defeat the type system.

# Chapter 6

# CONCLUSION AND FUTURE ENHANCEMENT

This paper presents an effective mechanism, which performs automatic authentication of users and creates log records of each data access by the user. The data owner can audit his content on the cloud, and he can get the confirmation that his data is safe on the cloud. The data owner is also able to know the duplication of data made without his knowledge. The data owner should not worry about his data on the cloud using this mechanism and data usage is transparent, using this mechanism.d. This specification allows the data proprietor to not only audit his content but also put into effect strong back-end protection if needed. Moreover, one of the main features of our work is that it enables the data proprietor to audit even those copies of its data that were made without his knowledge.

In the future we would like to develop a cloud, on which we will install JRE and JVM, to do the authentication of JAR ie; various VMs for various cloud users thereby enhancing the security of the cloud and by using various security algorithm to protect the data. Try to improve the security of store data and to reduce log record generation time. we premeditated to refine our approach to verify the integrity and the authentication of JARs. To improve different types of approaches for automatically logging any access to the data in the cloud together with an auditing mechanism. We would like to sustain a variety of security mechanisms, indexing policies for text files, usage control for executables, time to time logging, and closing of section.

# GLOSSARY

**JAR**: Java Archive File

**IOt**: Internet of Things

**JVM**: Java Virtual Machine

**JRE**: Java Runtime Enivironment

**CSP**: Cloud Servuce Provider

**SSL**: Secure Socket Layer

**SOA**: Service Oriented Architecture

**SAML**: Security Assertion Markup Language

**IAAS**: Infrastructure As A Service

**PAAS**: Platform As A Service

**SAAS**: Software As A Service

# REFERENCES


[1] Smitha Sundareswaran, Anna C. Squicciarini and Dan Lin, "Ensuring Distributed Accountability for Data Sharing in the Cloud,", IEEE Transaction on dependable a secure computing, VOL. 9, NO. 4, pg 556-568, 2012.

[2] S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud, " Proc First Int'l conf. Cloud Computing, 2009

[3] Ryan K L Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing" HP Laboratories, pp 1 – 7, HPL-2011

[4] R. Kalaiprasath, R. Elankavi and Dr. R. Udayakumar. Cloud Information Accountability (CIA) Framework Ensuring Accountability of Data in Cloud and Security in End to End Process in Cloud Terminology, International Journal of Civil Engineering and Technology, 8(4), 2017

[5] S. Nanda, T. Chiueh, A survey on virtualization technologies, Technical Report, ECSL-TR-129, SUNY at Stony Brook, 2005

[6] S. Nanda, T. Chiueh, A survey on virtualization technologies, Technical Report, ECSL-TR-129, SUNY at Stony Brook, 2005

[7] http://www.bic-trust.eu/files/2013/01/Papanikolaou$_{A}ccountabilityInCloudComputing_{J}une$2012.