**A Cross Tenant Access Control (CTAC) Model for Cloud Computing: Formal Specification and Verification**

**ABSTRACT:**

Sharing of resources on the cloud can be achieved on a large scale since it is cost effective and location independent. Despite the hype surrounding cloud computing, organizations are still reluctant to deploy their businesses in the cloud computing environment due to concerns in secure resource sharing. In this paper, we propose a cloud resource mediation service offered by cloud service providers, which plays the role of trusted third party among its different tenants. This paper formally specifies the resource sharing mechanism between two different tenants in the presence of our proposed cloud resource mediation service. The correctness of permission activation and delegation mechanism among different tenants using four distinct algorithms (Activation, Delegation,Forward Revocation and Backward Revocation) is also demonstrated using formal verification. The performance analysis suggest that sharing of resources can be performed securely and efficiently across different tenants of the cloud.

**SYSTEM ARCHITECTURE:**



**MODULES:**

* Data owner Module
* User and Physician Module
* Cloud Service Provider (CSP)
* Authority Module
* File hierarchy System

**MODULES DESCRIPTION:**

**Data owner Module:**

In the first module, we develop the Data Owner Module. Owner Will Signup and Wait for the approval Key of admin. After Getting key Owner can login using the key, and upload any records related to users medical Information on the cloud.

In this module, data owner will check the progress status of the file upload by him/her.It has large data needed to be stored and shared in cloud system. In our scheme, the entity is in charge of defining access structure and executing **Encrypt** operation. And it uploads ciphertext to CSP. After the completion, owner logout the session

**User and Physician Module:**

In this module, we develop the User Module. User Will registries and login on the user's page. We develop the module, such that, the User will search for his/her medical records by given user medical record id on the page.User will get search results of the medical records related to the idand he/she will request admintoaccess the document which is encrypted one by the admin.After Getting decrypt key from the admin,he/she can access to the medical records.User logouts the session.It wants to access a large number of data in cloud system. The entity first downloads the corresponding ciphertext. Then it executes **Decrypt** operation of the proposed scheme.

**Cloud Service Provider (CSP)**

It is a semi-trusted entity in cloud system. It can honestly perform the assigned tasks and return correct results. However, it would like to find out as much sensitive contents as possible. In the proposed system, it provides ciphertext storage and transmission services. In this module, we also develop admin module process. Admin Will Login on the admin's page.He/she will check the pending requests of any of the above person.After accepting the request from the above person,he/she will generate master key for encrypt andSecret key for decrypt.

**Authority Module:**

It is a completely trusted entity and accepts the user enrollment in cloud computing. And it can also execute **Setup** and **KeyGen**operations of the proposed scheme.The Researcher will registries and login on the researcher's page.Researcher will search for any medical records by the disease category(i.e Cancer,Hernia..etc..). Researcher will Request for decrypt key to the admin.After getting the key from admin,researcher will access to the medical records of patient without their personal details.After the process, Researcher logouts the session.

**File hierarchy System:**

The large number of classes in the Java IO package is overwhelming and annoying. However, if we use Java, we still need to understand those classes. In fact, the classes in Java IO package is not very complex, but we need a good way to learn those. There are two important factors for understanding the classes:

1). Java io class hierarchy diagram

2). Decorator pattern

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’LED
* Input Devices : Keyboard, Mouse
* Ram : 1GB.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows 7.
* Coding Language : JAVA/J2EE
* Tool : Netbeans 7.2.1
* Database : MYSQL