**Crypt Cloud+: Secure and Expressive Data Access Control for Cloud Storage**

**Alternative Title: Attribute based data management in crypt cloud**

**Aim:**

The main aim of this project is to provide integrity of an organization data which is in public cloud.

**Synopsis:**

Data owners will store their data in public cloud along with encryption and particular set of attributes to access control on the cloud data. While uploading the data into public cloud they will assign some attribute set to their data. If any authorized cloud user wants to download their data they should enter that particular attribute set to perform further actions on data owner’s data. A cloud user wants to register their details under cloud organization to access the data owner’s data. Users want to submit their details as attributes along with their designation. Based on the user details Semi-Trusted Authority generates decryption keys to get control on owner’s data. An user can perform a lot of operations over the cloud data. If the user wants to read the cloud data he needs to be entering some read related attributes, and if he wants to write the data he needs to be entering write related attributes. Foe each and every action user in an organization would be verified with their unique attribute set. These attributes would be shared by the admins to the authorized users in cloud organization. These attributes will be stored in the policy files in a cloud. If any user leaks their unique decryption key to the any malicious user data owners wants to trace by sending audit request to auditor and auditor will process the data owners request and concludes that who is the guilty.

**Existing System:**

In existing system the CP-ABE may help us prevent security breach from outside attackers. But when an insider of the organization is suspected to commit the “crimes” related to the redistribution of decryption rights and the circulation of user information in plain format for illicit financial gains, how could we conclusively determine that the insider is guilty? Is it also possible for us to revoke the compromised access privileges? In addition to the above questions, we have one more which is related to key generation authority. A cloud user’s access credential (i.e., decryption key) is usually issued by a semi-trusted authority based on the attributes the user possesses. How could we guarantee that this particular authority will not (re-)distribute the generated access credentials to others.

**Proposed System:**

In this work, we have addressed the challenge of credential leakage in CP-ABE based cloud storage system by designing an accountable authority and revocable Crypt Cloud which supports white-box traceability and auditing (referred to as Crypt Cloud+). This is the first CP-ABE based cloud storage system that simultaneously supports white-box traceability, accountable authority, auditing and effective revocation. Specifically, Crypt Cloud+ allows us to trace and revoke malicious cloud users (leaking credentials). Our approach can be also used in the case where the users’ credentials are redistributed by the semi-trusted authority.

**Modules:**

* **Organization profile creation & Key Generation**
* **Data Owners File Upload**
* **File Permission & Policy File Creation**
* **Tracing who is guilty**

**1. Organization profile creation & Key Generation**

User has an initial level Registration Process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database. Now the Accountable STA (semi-trusted Authority) generates decryption keys to the users based on their Attributes Set (e.g. name, mail-id, contact number etc..,). User gets the provenance to access the Organization data after getting decryption keys from Accountable STA.

**2. Data Owners File Upload**

In this module data owners create their accounts under the public cloud and upload their data into public cloud. While uploading the files into public cloud data owners will encrypt their data using RSA Encryption algorithm and generates public key and secret key. And also generates one unique file access permission key for the users under the organization to access their data.

**3. File Permission & Policy File Creation**

Different data owners will generate different file permission keys to their files and issues those keys to users under the organization to access their files. And also generates policy files to their data that who can access their data. Policy File will split the key for read the file, write the file, download the file and delete the file.

**4. Tracing who is guilty**

Authorized DUs are able to access (e.g. read, write, download, delete and decrypt) the outsourced data. Here file permission keys are issued to the employees in the organization based on their experience and position. Senior Employees have all the permission to access the files (read, write, delete, & download). Fresher’s only having the permission to read the files. Some Employees have the permission to read and write. And some employees have all the permissions except delete the data. If any Senior Employee leaks or shares their secret permission keys to their junior employees they will request to download or delete the Data Owners Data. While entering the key system will generate attribute set for their role in background validate that the user has all rights to access the data. If the attributes set is not matched to the Data Owners policy files they will be claimed as guilty. If we ask them we will find who leaked the key to the junior employees.

**Software Requirements**

* Windows 7 and above
* JDK 1.7
* J2EE
* Tomcat 7.0
* MySQL

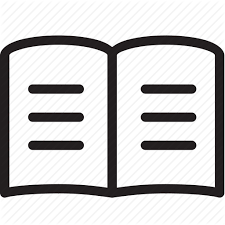
**Hardware Requirements**

* Hard Disk : 80GB and Above
* RAM : 4GB and Above
* Processor : P IV and Above

**Technology Used**

* + - J2EE (JSP, Servlets), JavaScript, HTML, CSS, AJAX.

**Architecture:**



STA

File Upload

Data Owners

Public Cloud

Cloud Users

File Permission Key

Encrypt Files

Policy File Creation

Registration & Login

File Read, Write, Download, Delete

Generates Attribute based decryption keys

Enter Decryption key

File permission key

True User

Key Leakage

Account Blocked

Informs to Data Owners

Data owners have all rights to delete and edit their data