



# **SECURE AND DATA ACCESS CONTROL FOR CLOUD STORAGE**

## **A PROJECT REPORT**

*Submitted by*

**V.ARUN PRASAD      111919205003**

**E.EZHILARASAN      111919205008**

**B.JASWANTH      111919205012**

*In partial fulfillment for the award of the degree*

*Of*

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

**S.A. ENGINEERING COLLEGE**

**CHENNAI – 600 077.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2023**

## **BONAFIDE CERTIFICATE**

**Certified that this project report “SECURE AND DATA ACCESS CONTROL FOR CLOUD STORAGE” is the bonafide work of “V.ARUN PRASAD (111919205003), E.EZHILARASAN (111919205008), B.JASWANTH (111919205012)” who carried out the project work under my supervision.**

**SIGNATURE**

**Dr.AHMED MUDASSAR ALI,M.E.,Ph.D**

**HEAD OF THE DEPARTMENT**

**PROFESSOR**

Dept. of Information Technology

S.A.Engineering College

Chennai-600 077.

**SIGNATURE**

**Mrs.A.M.SERMAKANI,M.E.,Ph.D**

**SUPERVISOR**

**ASSOCIATE PROFESSOR**

Dept. of Information Technology

S.A.Engineering College

Chennai-600 077.

Submitted to Project and Viva-Voce Examination held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**



**Global Techno Solutions<sup>®</sup>**  
Solutions unlimited

---

22/03/2023

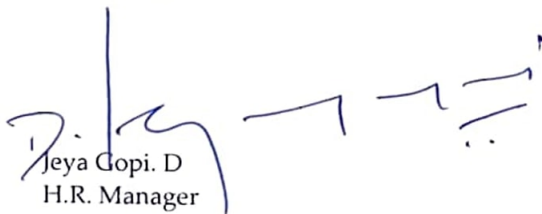
TO WHOMSOEVER IT MAY CONCERN

This is to certify that the following final year B.Tech (Information Technology) students of S.A Engineering College, Chennai has successfully completed their project work title **"Secure and Data Access Control for Cloud Storage"** during January, 2023 to March, 2023 (Except Sundays and Holidays) in our organization.

Mr. Arun Prasad. V	(Reg. No. 111919205003)
Mr. Ezhilarasan. E	(Reg. No. 111919205008)
Mr. Jaswanth. B	(Reg. No. 111919205012)

We wish them all success for their future endeavors.

For Global Techno Solutions

  
Jeya Gopi. D  
H.R. Manager



No.60/4, 11th Avenue, Ashok Nagar, Chennai - 600 083. INDIA.  
Tel : +91-44-4203 3422 Email : [career@globaltechnosolutions.net](mailto:career@globaltechnosolutions.net) Web : [www.globaltechnosolutions.net](http://www.globaltechnosolutions.net)

## ACKNOWLEDGEMENT

We express our gratitude to beloved founder (**Late**)**Thiru D.SUDHARSSANAM, Founder** and the honorable chairman **Thiru D.DURAIWAMY**, who gave us the opportunity to do this project.

We express our proud thanks and deep sense of gratitude to **Thiru S.AMARNATH**, Correspondent and We would like to sincerely thank **Thiru D.SABARINATH**., Director, for the facilities and support provided by them in the college.

We are thankful to **Dr.S.RAMACHANDRAN, M.E.,Ph.D., Principal**, for providing all necessary facilities for undertaking the project.

We express our deep sense of gratitude and heartfelt thanks to **Dr.AHMED MUDASSAR ALI, M.E.,Ph.D., Head of Department of Information Technology** for giving valuable suggestions, expert guidance and encouragement which paved the way for the successful completion of our project work.

We are deeply obliged to our Project Guide **Mrs.A.M.SERMAKANI , M.E., Ph.D., Associate Professor**, Department of Information Technology for offering her valuable guidelines and suggestions during course work of project. We also extend our sincere thanks to all faculty members and non-teaching staff members of Department of Information Technology, who have given their co- operation and helped us to complete the project successfully

Finally, we express our thanks to our beloved Parents and Family Members for their moral support to make our project a grand success.

## **ABSTRACT**

Secure cloud storage, which is an emerging cloud service, is designed to protect the confidentiality of outsourced data but also to provide flexible data access for cloud users whose data is out of physical control. Cipher text Policy Attribute-Based Encryption (CP-ABE) is regarded as one of the most promising techniques that may be leveraged to secure the guarantee of the service. However, the use of CP-ABE may yield an inevitable security breach which is known as the misuse of access credential (decryption rights), due to the intrinsic “all-or-nothing” decryption feature of CP-ABE. We investigate the two main cases of access credential misuse: one is on the semi-trusted authority side, and the other is on the side of cloud user. To mitigate the misuse, we propose the first accountable authority and revocable CP-ABE based cloud storage system with white-box traceability and auditing, referred to as Cloud. We also present the security analysis and further demonstrate the utility of our system via experiments.

## LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
3.3	Architecture Diagram	10
5.3	Working of Java	18
5.3.1	Java Platform	19
5.4	Network Diagram	23
5.4.1	Virtualized Cloud Model	26
7.1	Sequence diagram	34
7.2	Use case Diagram	35
7.3	Activity Diagram	36
7.4	Colloboration Diagram	37
7.5	Dataflow Diagram	38
7.6	Class Diagram	40

## LIST OF ABBREVIATIONS

- **JVM** Java Virtual Machine.
- **JDK** Java Development Toolkit.
- **J2EE** Java 2 Platform Enterprise Edition.
- **API** Java Application Programming Interface.
- **SeDaSC** Secure Data Sharing in Clouds
- **JSP** Jakarta Server Pages.
- **HTML** Hypertext Markup Language.
- **CSS** Cascading Style Sheet.
- **JMX** Java Management Extensions.
- **SQL** Structured Query Language.
- **XML** Extensible Markup Language.
- **AJAX** Asynchronous JavaScript And XML.
- **CP-ABE** Cipher text Policy Attribute - Based Encryption.
- **LAN** Local Area Network
- **WAN** Wide Area Network
- **VPN** Virtual Private Network
- **VLAN** Virtual Local Area Network
- **PaaS** Platform as a Service

- **SaaS**                      Software as a Service.
- **IaaS**                      Identity as a Service.
- **NaaS**                      Network as a Service.
- **SOA**                      Service – Oriented Architecture.
- **STA**                      Semi – Trusted Authority.
- **UML**                      Unified Modeling Language.
- **DFD**                      Data Flow Diagram.
- **DOs**                      Data Owners.
- **Dus**                      Data Users.
- **RSA**                      Rivest - Shamir – Adleman.
- **HTTP**                      Hyper Text Transfer Protocol.
- **ADT**                      Android Development Tool



# TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	<b>Abstract</b>	V
	<b>List of Figures</b>	VI
	<b>List of Abbreviations</b>	VII
<b>1</b>	<b>Introduction</b>	1
<b>2</b>	<b>Literature Survey</b>	2
	2.1 Introduction	2
	2.2 Cloud data integrity checking with an identity-based auditing mechanism from RSA.	2
	2.3 Encrypted data management with deduplication in cloud computing.	2
	2.4 Expressive, Efficient and Revocable Data Access Control for Multi-Authority Cloud.	3
	2.5 Large Universe Ciphertext-Policy Attribute -Based Encryption with White-Box Traceability.	3
	2.6 Leveraging software defined networking for security policy enforcement.	4

	2.7 Security and Privacy for Storage and Computation in Cloud Computing.	4
	2.8 Attribute Based Data Sharing with Attribute Revocation.	4
	2.9 SeDaSC: Secure Data Sharing in Clouds.	5
	2.10 Crypt Cloud+: Secure and Expressive Data Access Control for Cloud Storage.	5
	2.11 Crypt Cloud Secure and Expressive Data Access Control for Cloud Storage.	6
<b>3</b>	<b>System Design</b>	<b>7</b>
	3.1 Introduction	7
	3.2 Detailed description of Proposed of Modules	7
	3.3 System Architecture Diagram	10
<b>4</b>	<b>System Analysis</b>	<b>11</b>
	4.1 Overview of Existing System	11
	4.1.1 Disadvantage	11
	4.2 Overview of Proposed System	12
	4.2.1 Advantage	13

<b>5</b>	<b>Requirement Specification</b>	<b>14</b>
	5.1 Introduction	14
	5.2 Features of Java	15
	5.3 Working of Java	16
	5.3.1 Java Platform	19
	5.4 Introduction to Cloud Computing	22
	5.4.1 Virtualization	26
	5.4.2 Service – Oriented Architecture	27
	5.5 Hardware and Software Specifications	28
	5.5.1 Hardware Requirements	28
	5.5.2 Software Requirements	28
	5.6 Technologies Used	28
<b>6</b>	<b>Project Purpose and Scope</b>	<b>29</b>
	6.1 Purpose	29
	6.2 Project Scope	29

	6.3 Product Perspective	29
	6.4 System Features	30
	6.5 Design and Implementation Constraints	31
	6.6 Other Nonfunctional Requirements	32
	6.6.1 Performance Requirements	32
	6.6.2 Safety Requirements	33
<b>7</b>	<b>UML Diagrams</b>	34
	7.1 Sequence Diagram	34
	7.2 Use Case Diagram	35
	7.3 Activity Diagram	36
	7.4 Collaboration Diagram	37
	7.5 Data Flow Diagram	38
	7.6 Class Diagram	40
<b>8</b>	<b>Coding and Testing</b>	41
	8.1 Coding Standards	41
	8.2 Coding	43
	8.3 Test Procedure	48
	8.4 Types of Testing	48

	8.4.1 Unit Testing	48
	8.4.2 Functional Test	49
	8.4.3 Integration Testing	49
	8.5 Testing Techniques	50
	8.5.1 White Box Testing	51
	8.5.2 Black Box Testing	51
<b>9</b>	<b>Screenshots</b>	<b>52</b>
<b>10</b>	<b>Conclusion and Future Enhancement</b>	<b>58</b>
	10.1 Conclusion	58
	10.2 Future Enhancement	58
	<b>Reference</b>	<b>59</b>

# **CHAPTER 1**

## **INTRODUCTION**

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. For 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.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 INTRODUCTION**

The purpose of literature survey is to give the brief overview and also establish complete information about the reference papers. The goal of literature survey is to completely specify the technical details related to the main project in a concise and unambiguous manner.

#### **2.2 Cloud data integrity checking with an identity-based auditing mechanism from RSA.**

Author: Yong Yua, Liang Xuea, Man Ho Aub, Willy Susilo, Jianbing Ni, Yafang Zhanga, Athanasios V. Vasilakos, Jian Shene. Year – 2016.

Cloud data auditing is extremely essential for securing cloud storage since it enables cloud users to verify the integrity of their outsourced data efficiently. The computation overheads on both the cloud server and the verifier can be significantly reduced by making use of data auditing because there is no necessity to retrieve the entire file but rather just use a spot - checking technique.

#### **2.3 Encrypted data management with deduplication in cloud computing.**

Author : Trupti Rongare. Year – 2016.

To preserve cloud data confidentiality and user privacy, cloud data are often stored in an encrypted form. But duplicated data that are encrypted under different encryption schemes could be stored in the cloud, which greatly decreases the

utilization rate of storage resources, especially for big data. Several data reduplication schemes have recently been proposed.

## **2.4 Expressive, Efficient and Revocable Data Access Control for Multi-Authority Cloud Storage.**

Author : Chetan Bulla Akshata R. Patil, Priyanka B. Guttedar and Reshma G. Giddenavar. Year - Jun 2016.

Cipher text-Policy Attribute-based Encryption (CP-ABE) is regarded as one of the most suitable technologies for data access control in cloud storage, because it gives data owners more direct control on access policies. However, it is difficult to directly apply existing CP-ABE schemes to data access control for cloud storage systems because of the attribute revocation problem.

## **2.5 Large Universe Ciphertext-Policy Attribute-Based Encryption with White-Box Traceability.**

Author: Jianting Ning, Zhenfu Cao, Xiaolei Dong, Lifei Wei, and Xiaodong.  
Year - 2014

A Cipher text-Policy Attribute-Based Encryption (CP-ABE) system extracts the decryption keys over attributes shared by multiple users. It brings plenty of advantages in ABE applications. CP ABE enables ne-grained access control to the encrypted data for commercial applications. We have also proved the selective security of our new system in the standard model under q-type" assumption.



## **2.6 Leveraging software defined networking for security policy enforcement.**

Author:iaqiangLiu,YongLi,HuandongWang, DepengJin, LiSu, LieguangZeng, ThanosVasilakos. Year - 2015

Network operators employ a variety of security policies for protecting the data and services. However, deploying these policies in traditional network is complicated and security vulnerable due to the distributed network control and lack of standard control protocol. Software defined network provide a ideal paradigm to address these challenges by separating control plane and data plane, and exploiting the logically centralized control.

## **2.7 Security and Privacy for Storage and Computation in Cloud Computing.**

**Author :** K. Sharmila<sup>1</sup>, V. Vinoth Kumar. **Year** – 2014

The Secure Data Sharing in Clouds (SeDaSC) methodology that provides: data confidentiality and integrity, access control, data sharing (forwarding) without using compute-intensive re-encryption, insider threat security, and forward and backward access control. The SeDaSC methodology encrypts a file with a single encryption key.

## **2.8 Attribute Based Data Sharing with Attribute Revocation.**

Author : Shucheng, YuCong Wang, Kui Ren. Year - 2010

Ciphertext-Policy Attribute Based Encryption (CP-ABE) is a promising cryptographic primitive for fine-grained access control of shared data. In CP-ABE, each user is associated with a set of attributes and data are encrypted with access

structures on attributes. A user is able to decrypt a cipher text if and only if his attributes satisfy the cipher text access structure. Beside this basic property, practical applications usually have other requirements. In this paper we focus on an important issue of attribute revocation which is cumbersome for CP-ABE schemes.

## **2.9 : SeDaSC: Secure Data Sharing in Clouds.**

Author : Mazhar Ali, Revathi Dhamotharan Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos Keqin Li, Albert Y. Zomaya. Year - April 2015.

Cloud storage is an application of clouds that liberates organizations from establishing in-house data storage systems. However, cloud storage gives rise to security concerns. In case of group-shared data, the data face both cloud-specific and conventional insider threats. Secure data sharing among a group that counters insider threats of legitimate yet malicious users is an important research issue.

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

Author : Jianting Ning, Zhenfu Cao, Xiaolei Dong, Kaitai Liang. Year – 2018

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is regarded as one of the most promising techniques that may be leveraged to secure the guarantee of the service. However, the use of CP-ABE may yield an inevitable security breach which is known as the misuse of access credential, due to the intrinsic decryption feature of CP-ABE.

## **2.11: Crypt Cloud Secure and Expressive Data Access Control for Cloud Storage.**

Author : John Justin, Sybi Cynthia, Rohith Viswanathan G. Year – 2019

In this paper, we investigate the two main cases of access credential misuse: one is on the semi-trusted authority side, and the other is on the side of cloud user. To mitigate the misuse, we propose the first accountable authority and revocable CP-ABE based cloud storage system with white-box traceability and auditing, referred to as Crypt Cloud+.

## **CHAPTER 3**

### **SYSTEM DESIGN**

#### **3.1 INTRODUCTION**

Data owners (DOs) encrypt their data under the relevant access policies prior to outsourcing the (encrypted) data to a public cloud (PC). PC stores the outsourced (encrypted) data from DOs and handles data access requests from data users (DUs). Authorized DUs are able to access (e.g. download and decrypt) the outsourced data. Semi-trusted authority (AT) generates system parameters and issues access credentials (i.e., decryption keys) to DUs. Auditor (AU) is trusted by other entities; takes charge of audit and revoke procedures, and returns the trace and audit results to DOs and DUs.

#### **3.2 DETAILED DESCRIPTION OF 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..). A cloud user wants to register their details under cloud

organization to access the data owner's data. 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. Data Owners upload the file in private cloud owner must do permission and policy who wants to access the file. It's must be safe to store in private cloud.

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

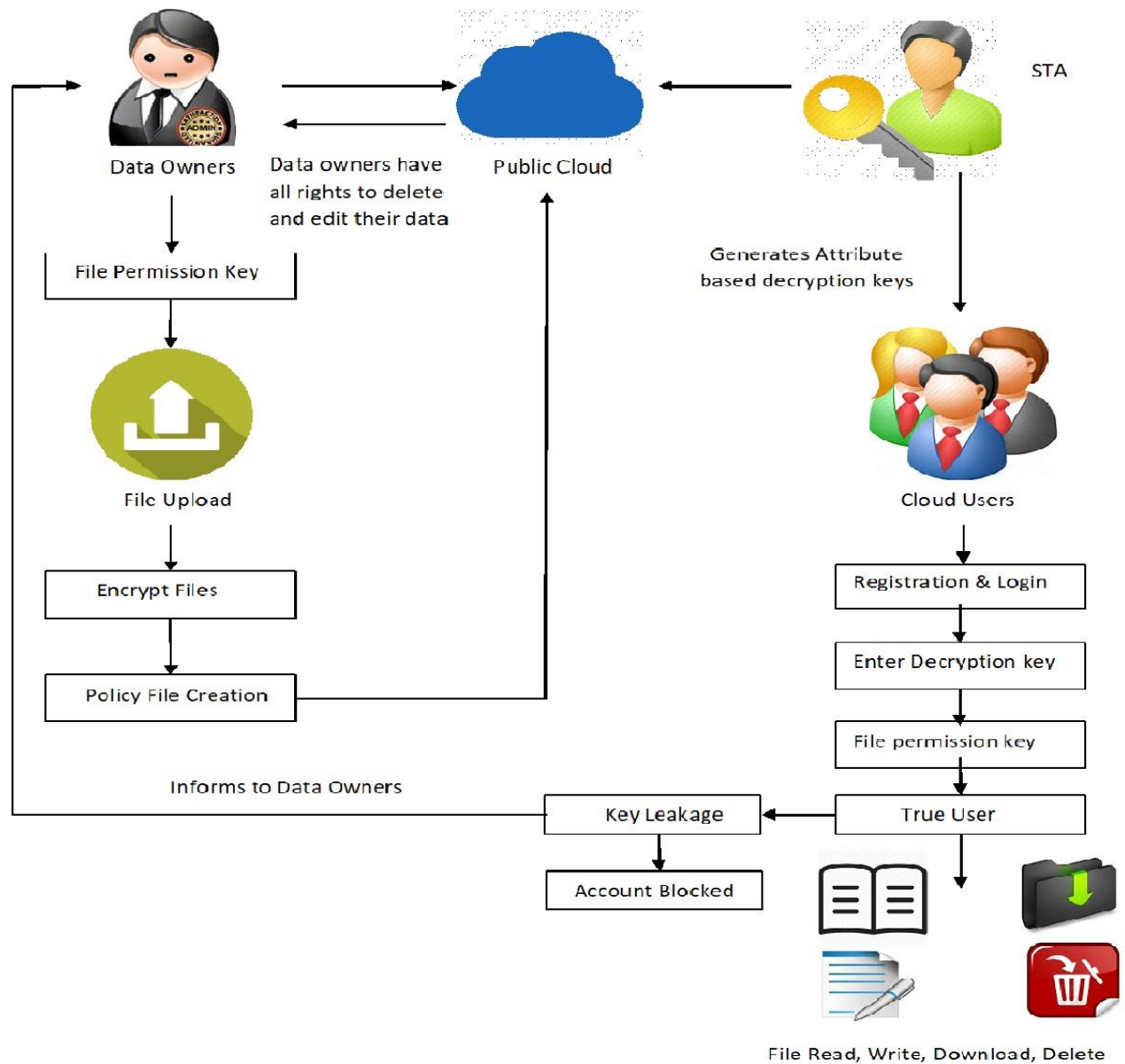
Different data owners will generate different file permission keys to their files and issues those keys to users under the organization to access their files. And also generates policy files to their data that who can access their data. Policy File will split the key for read the file, write the file,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

Authority generates decryption keys to get control on owner's 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. For 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.

#### **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 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.

### 3.3 SYSTEM ARCHITECTURE DIAGRAM



**Fig: 3.3 Architecture Diagram**

## **CHAPTER 4**

### **SYSTEM ANALYSIS**

#### **4.1 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.

##### **4.1.1 Disadvantages**

- 1) The existing CP-ABE based cloud storage systems fail to consider the case where access credential is misused.
- 2) Don’t Provided in file access policy. Users’ who leak their access credentials can’t be traced and identified.



3) Key Generation – In Existing System the key generation is Manual technique. We want to enter manually. So it is time consuming and also hard to remember the key to use it.

## **4.2 Proposed System**

In this work, 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. This is the CP-ABE based cloud storage system that simultaneously supports white-box traceability, accountable authority, auditing and effective revocation. Specifically, Cloud allows us to trace and revoke malicious cloud users (leaking credentials).

AT produces the system parameters initially in order to configure the system and then communicates the full system parameters (including public and private parameters) with AU. The public parameters are then published. AT also produces access credentials (decryption keys) for DUs based on their IDs and qualities. DOs encrypt their data using access controls they choose, and then outsource the encrypted data to PC. To gain access to the underlying data, any authorized DU can decode the outsourced ciphertexts. A DU is approved if his or her collection of attributes fulfils the access policy imposed over the outsourced data.

Our Cloud was created with safe cloud storage in mind. Our approach can be also used in the case where the users' credentials are redistributed by the semi-trusted authority.

### **4.2.1 Advantages**

- 1) Reduce IT infrastructure costs - By utilizing cloud software, you eliminate the need to invest in bigger numbers of more powerful servers, as well as the requirement for IT personnel to manage such powerful servers.
- 2) Reduced Software Costs - It saves money on software since you don't have to buy individual software packages for each machine in the firm.
- 3) Key Generation – In our project the key generation is automatic. You don't need to type the attribute keys. Just click on it.
- 4) Data Integrity - Reducing cloud users' burden of storage management and equipment maintenance. Avoiding investing a large amount of hardware and software. Enabling the data access independent.
- 5) Security and Privacy - High accessibility, availability and reliability make cloud computing a better solution for interoperability problems.
- 6) SeDaSC - Data confidentiality and integrity; Access control; Insider threat security.

## CHAPTER 5

### REQUIREMENT SPECIFICATIONS

#### 5.1 Introduction

The widespread usage of cloud computing may pose an indirect risk to the security of outsourced data and the privacy of cloud users. A major problem here is ensuring that only authorized individuals have access to the data that has been outsourced to the cloud, at any time and from any location. One naïve method is to encrypt the data before uploading it to the cloud. Nevertheless, the method restricts additional data exchange and processing. This is because a data owner must download encrypted data from the cloud and re-encrypt it before sharing it suppose the data owner has no local copies of the data. In the context of cloud computing, fine-grained access control over encrypted data is desirable. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) may be an effective method for ensuring data confidentiality and providing fine-grained access control in this situation. Organizations (for example, an institution like the University of Texas at San Antonio) and individuals can use a CP-ABE-based cloud storage system. (e.g., students, faculty members and visiting scholars of the university). Approved cloud users are then given access credentials (i.e., decryption keys) matching to their attribute sets (e.g., student role, faculty member role, or guest role), which they may use to access the outsourced data. Being a strong one-to-many encryption system, CP- ABE not only protects data saved in the cloud, but it also allows for fine-grained access control over the data.

## 5.2 Features of JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak. The first implementation of Oak was in a PDA-type device called Star Seven (\*7) that consisted of the Oak language, an operating system called Green OS, a user interface, and hardware. The name \*7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by NCSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

### 5.3 Working of JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method. To output text from the program, we execute the **println** method of **System.out**, which is java's output stream. UNIX users will appreciate the theory

behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things:

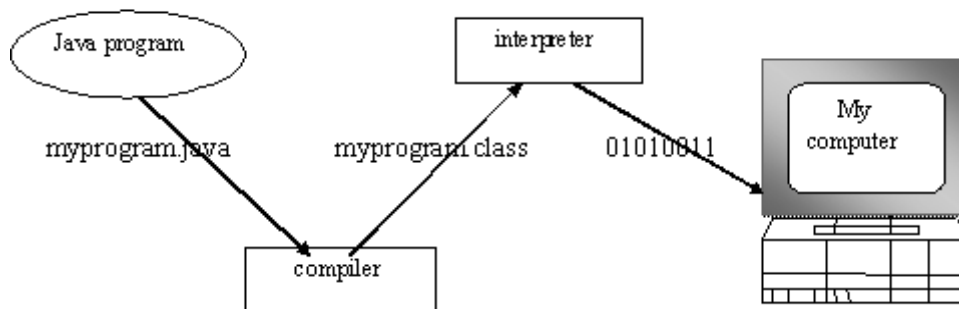
- Programming language
- Platform

Java is a high-level programming language that is all of the following:

- Simple
- Object-oriented
- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic

The code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:



**Fig.5.3 Working of JAVA**

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine (JVM)**. Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

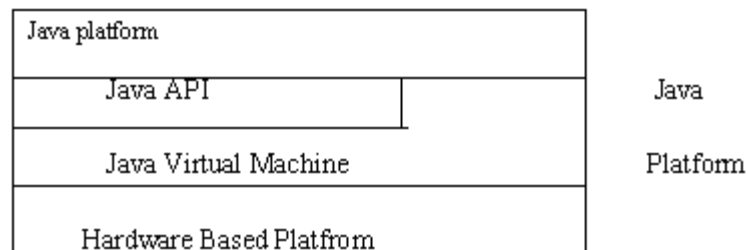
### 5.3.1 The JAVA Platform

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components:

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (**packages**) of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



**Fig.5.3.1 Java Platform**



As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

## **Apache Tomcat Server**

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top-level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

## **Environment**

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets. The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments. Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

## **Product features**

### Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector
- Java Management Extensions (JMX), JSP and Struts-based administration
- Tomcat 5.x
- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability
- native Windows and Unix wrappers for platform integration
- faster JSP parsing

## **History**

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation. Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of

and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

## **Algorithm**

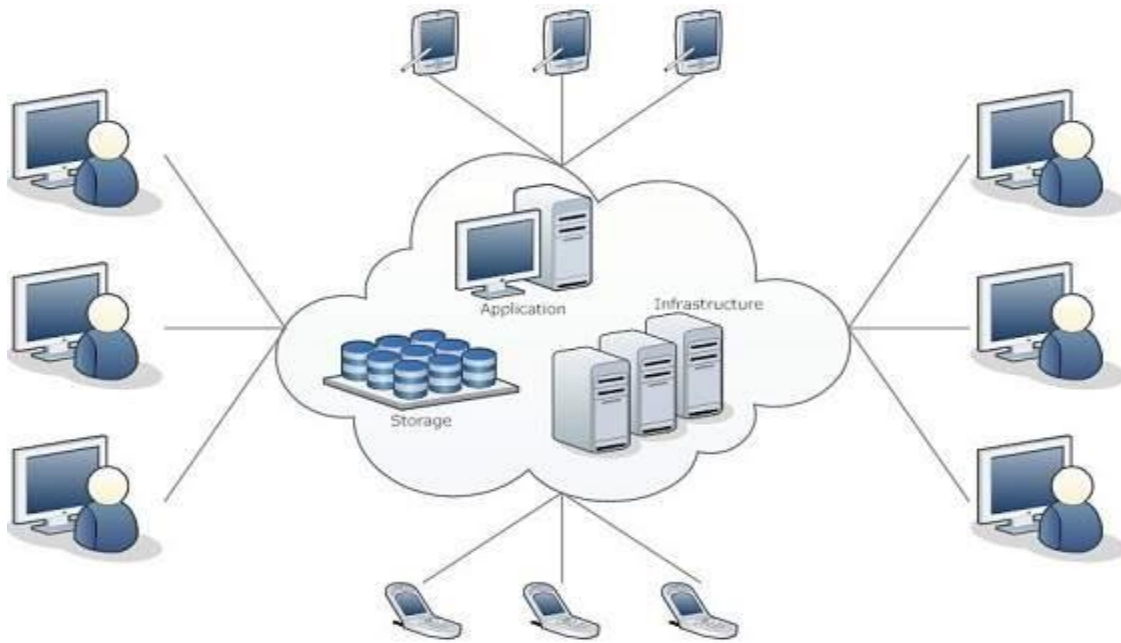
### **Binary search**

**Binary search** is a fast **search algorithm** with run-time complexity of  $O(\log n)$ . ... For this **algorithm** to work properly, the data collection should be in the sorted form. **Binary search** looks for a particular item by comparing the middle most item of the collection. If a match occurs, then the index of item is returned.

## **5.4 Introduction to Cloud Computing**

Cloud Computing provides us means of accessing the applications as utilities over the Internet. It allows us to create, configure, and customize the applications online.

The term Cloud refers to a Network or Internet. In other words, we can say that Cloud is something, which is present at remote location. Cloud can provide services over public and private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM) execute on cloud. Cloud Computing refers to manipulating, configuring, and accessing the hardware and software resources remotely. It offers online data storage, infrastructure, and application.



**Fig: 5.4 Network Diagram**

Cloud computing offers platform independency, as the software is not required to be installed locally on the PC. Hence, the Cloud Computing is making our business applications mobile and collaborative.

There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. Following are the working models for cloud computing:

- Deployment Models
- Service Models

### **Cloud Service Models**

**Infrastructure-as-a-Service** provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc. Apart from these resources, the IaaS also offers:

- Virtual machine disk storage
- Virtual local area network (VLANs)
- Load balancers
- IP addresses
- Software bundles

All of the above resources are made available to end user via server virtualization. Moreover, these resources are accessed by the customers as if they own them.

**Platform-as-a-Service** offers the runtime environment for applications. It also offers development and deployment tools required to develop applications. PaaS has a feature of point-and-click tools that enables non-developers to create web applications. App Engine of Google and Force.com are examples of PaaS offering vendors. Developer may log on to these websites and use the built-in API to create web-based applications. But the disadvantage of using PaaS is that, the developer locks-in with a particular vendor. For example, an application written in Python against API of Google, and using App Engine of Google is likely to work only in that environment.

The following diagram shows how PaaS offers an API and development tools to the developers and how it helps the end user to access business applications.

**Software-as-a-Service (SaaS)** model allows providing software application as a service to the end users. It refers to software that is deployed on a host service and is accessible via Internet. There are several SaaS applications listed below:

- Billing and invoicing system
- Customer Relationship Management (CRM) applications

- Help desk applications
- Human Resource (HR) solutions

Some of the SaaS applications are not customizable such as Microsoft Office Suite. But SaaS provides us Application Programming Interface (API), which allows the developer to develop a customized application.

**Identity-as-a-Service (IDaaS)** Employees in a company require to login to system to perform various tasks. These systems may be based on local server or cloud based. Following are the problems that an employee might face:

- Remembering different username and password combinations for accessing multiple servers.
- If an employee leaves the company, it is required to ensure that each account of that user is disabled. This increases workload on IT staff.

To solve above problems, a new technique emerged which is known as **Identity-as-a-Service (IDaaS)**. IDaaS offers management of identity information as a digital entity. This identity can be used during electronic transactions.

**Network-as-a-Service** allows us to access to network infrastructure directly and securely. NaaS makes it possible to deploy custom routing protocols.

NaaS uses virtualized network infrastructure to provide network services to the customer. It is the responsibility of NaaS provider to maintain and manage the network resources. Having a provider working for a customer decreases the workload of the customer. Moreover, NaaS offers network as a utility. NaaS is also based on pay-per-use model.

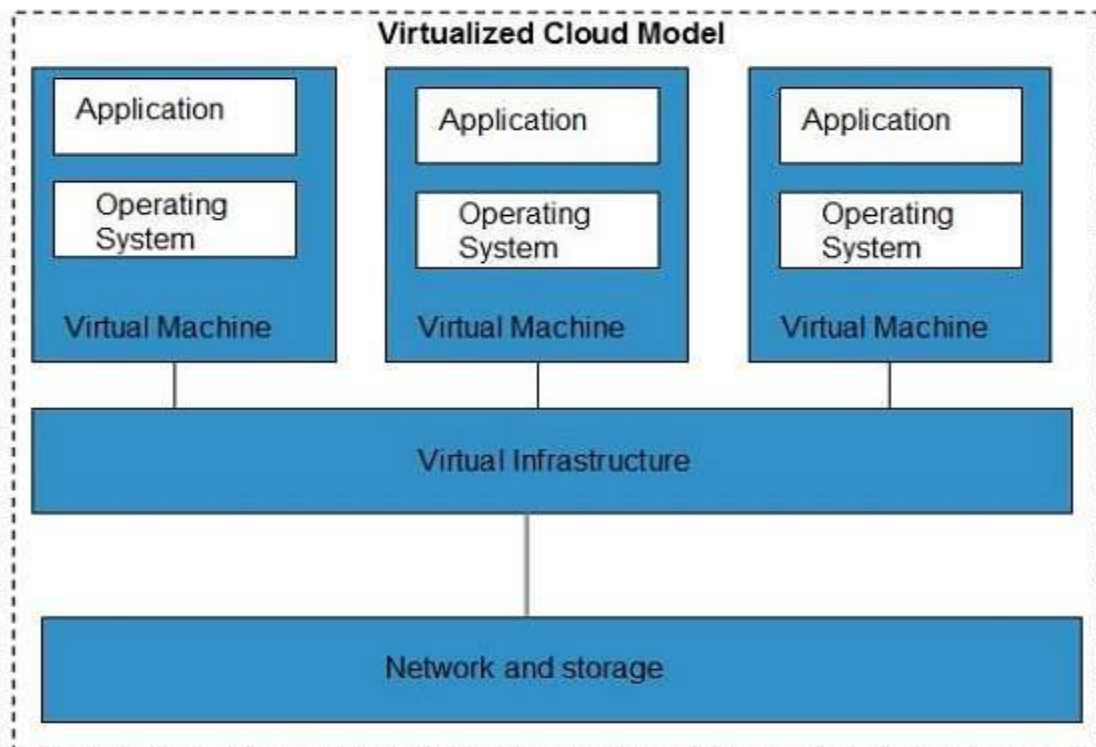
## Cloud Computing Technologies

There are certain technologies working behind the cloud computing platforms making cloud computing flexible, reliable, and usable. These technologies are listed below:

- Virtualization
- Service-Oriented Architecture (SOA)
- Grid Computing
- Utility Computing

### 5.4.1 Virtualization

**Virtualization** is a technique, which allows sharing single physical instance of an application or resource among multiple organizations or tenants (customers).

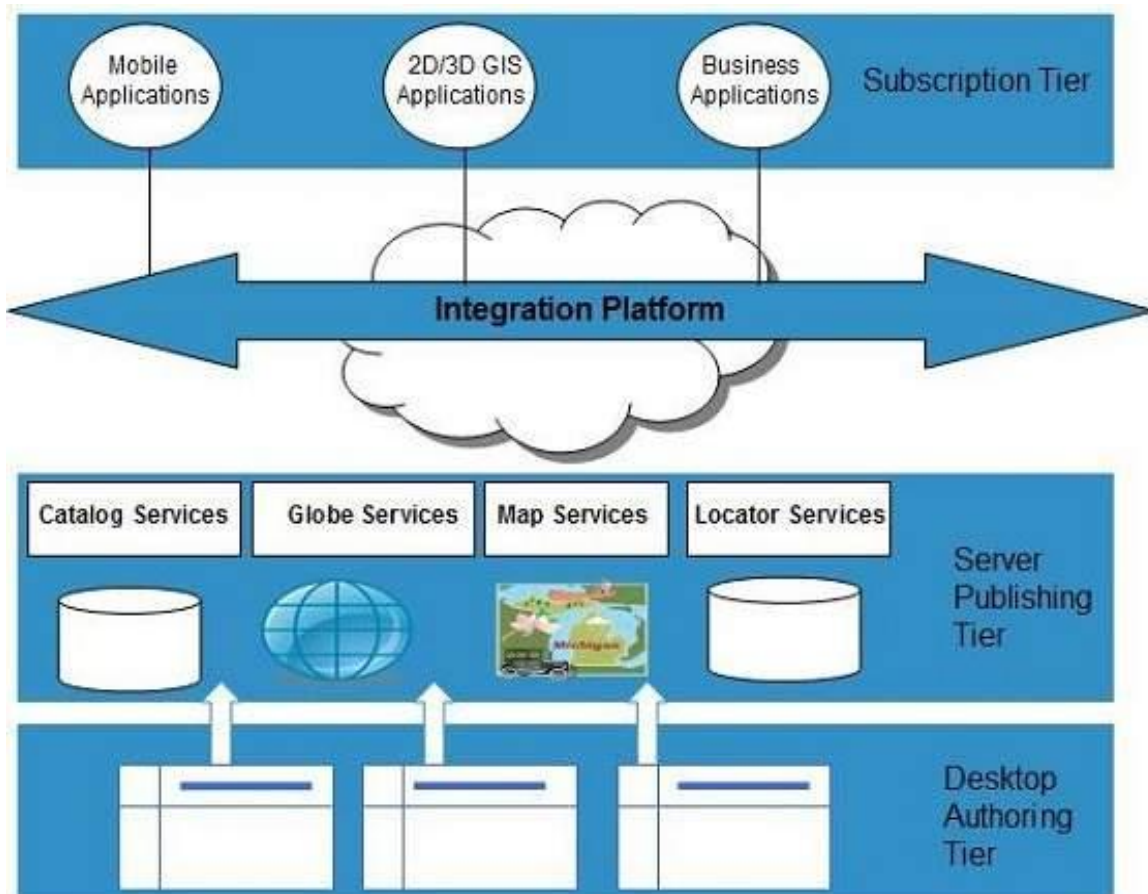


**Fig: 5.4.1 Virtualized Cloud Model**

The Multitenant architecture offers virtual isolation among the multiple tenants. Hence, the organizations can use and customize their application as though they each have their instances running.

#### 5.4.2 Service-Oriented Architecture (SOA)

Service-Oriented Architecture helps to use applications as a service for other applications regardless the type of vendor, product or technology. Therefore, it is possible to exchange the data between applications of different vendors without additional programming or making changes to services. The cloud computing service oriented architecture is shown in the diagram below.



**Fig: 5.4.2 SOA Architecture**



## **5.5 HARDWARE AND SOFTWARE SPECIFICATION**

### **5.5.1 HARDWARE REQUIREMENTS**

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

### **5.5.2 SOFTWARE REQUIREMENTS**

- Windows 7 and above
- JDK 1.7
- J2EE
- Java Virtual Machine (JVM)
- Tomcat 7.0
- MySQL

## **5.6 TECHNOLOGIES USED**

J2EE (JSP, Servlets), JavaScript, HTML, CSS, AJAX.

## **CHAPTER 6**

### **PROJECT PURPOSE AND SCOPE**

#### **6.1 Purpose**

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

#### **6.2 Project Scope**

The problem of credential leakage in CPABE- based cloud storage systems in this study by establishing a responsible authority and revocable Cloud that provides white-box traceability and auditing (referred to as Cloud). This is the first CP-ABE-based cloud storage solution that offers white-box traceability, responsible authority, auditing, and effective revocation all at the same time. Cloud, in particular, enables us to track down and deactivate rogue cloud users (leaking credentials). Our solution may also be employed when the semi-trusted authority redistributes the users' credentials.

#### **6.3 Product Perspective**

Data owners will store their data in public cloud along with encryption and particular set of attributes to access control on the cloud data. Cloud owners have all rights to download and delete their data whenever they want. 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. A 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. For 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.

## **6.4 System Features**

- 1) The capacity to track malevolent cloud users. Users who reveal their login credentials can be tracked down and identified.
- 2) Responsible power. It is possible to identify a semi-trusted authority who produces and distributes access credentials to unauthorized user(s) without sufficient authorization. This enables additional actions to be conducted (e.g. criminal investigation or civil litigation for damages and breach of contract).
- 3) Examining. An auditor can detect whether a (suspected) cloud user has leaked his or her access credential.

4) Tracing requires "almost" no storage. We employ a Paillier like encryption as an extractable commitment in tracking malevolent cloud users, and we don't need to keep a user identification table for tracing.

5) Revocation of malicious cloud users. Access credentials for individuals who have been tracked and judged to be "compromised" can be withdrawn. We devise two procedures to effectively revoke the "traitor(s)." The ATER-CP-ABE provides an explicitly revocation mechanism in which a revocation list is explicitly specified in the algorithm Encrypt, whereas the ATIR-CP-ABE provides implicit revocation in which the encryption does not need to know the revocation list but a key update operation is required on a regular basis.

## **6.5 Design and Implementation Constraints**

### **Constraints in Analysis**

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

### **Constraints in Design**

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions

- Determination of the Require Clauses
- Global actions and Constraint Realization

## **Constraints in Implementation**

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object-oriented methods.

## **6.6 Other Nonfunctional Requirements**

### **6.6.1 Performance Requirements**

The application at this side controls and communicates with the following three main general components.

- Embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following. Web

Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

### **6.6.2 Safety Requirements**

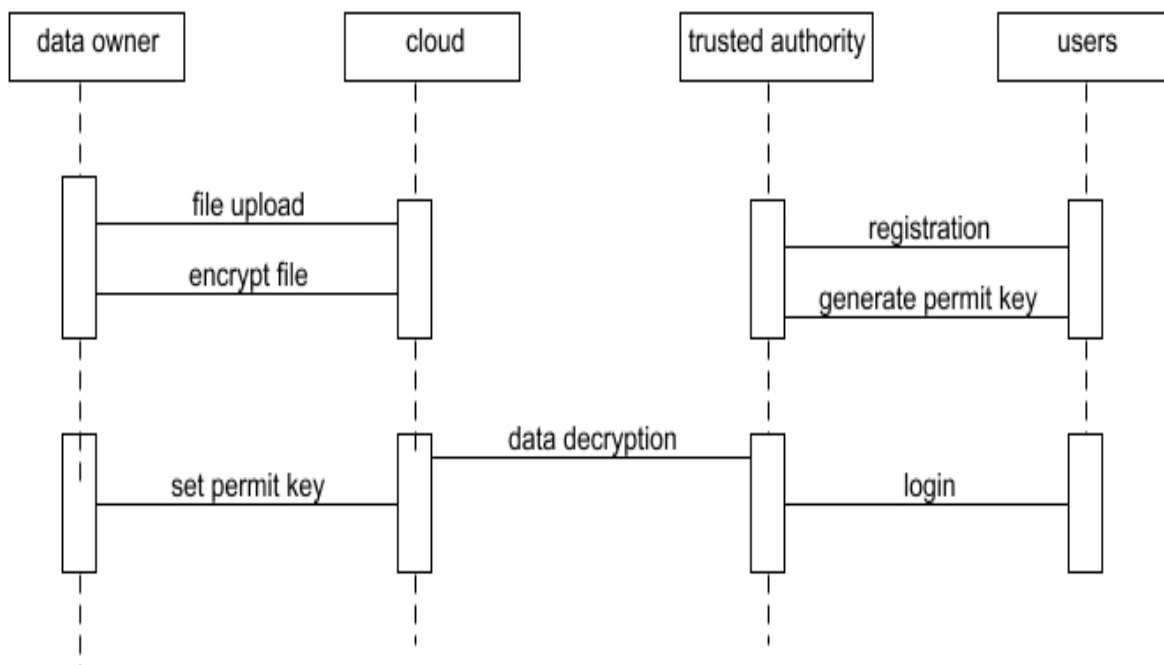
1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

## CHAPTER 7

### UML DIAGRAMS

#### 7.1 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



**Fig:7.1 SEQUENCE DIAGRAM**

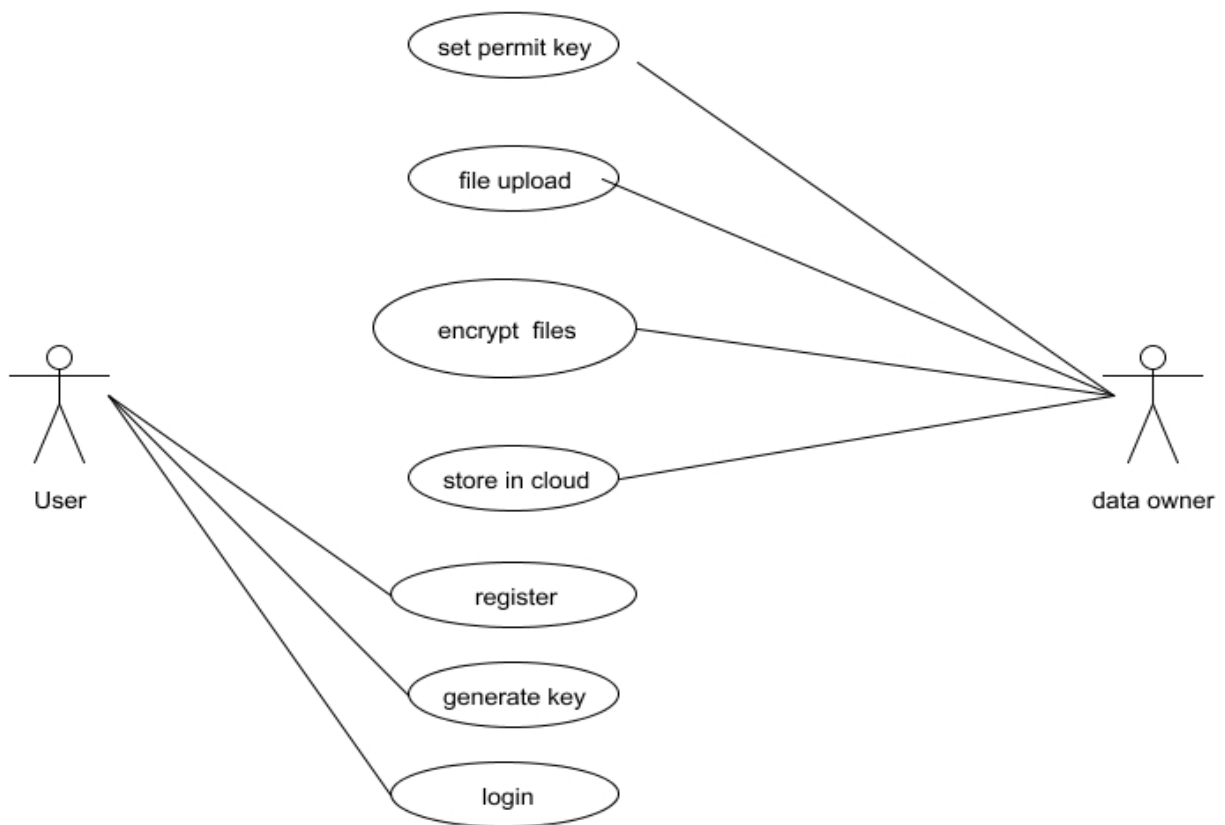
## 7.2 Use Case Diagram:

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

**Use case:** A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actor:** An actor is a person, organization or external system that plays a role in one or more interaction with the system.



**FIG:7.2 USECASE DIAGRAM**

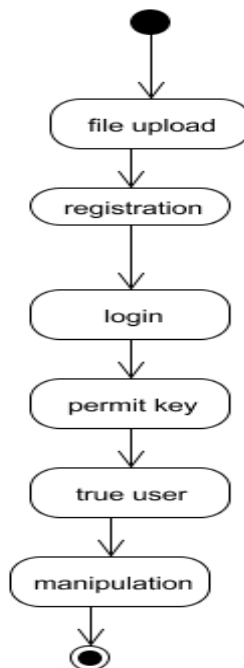


### 7.3 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

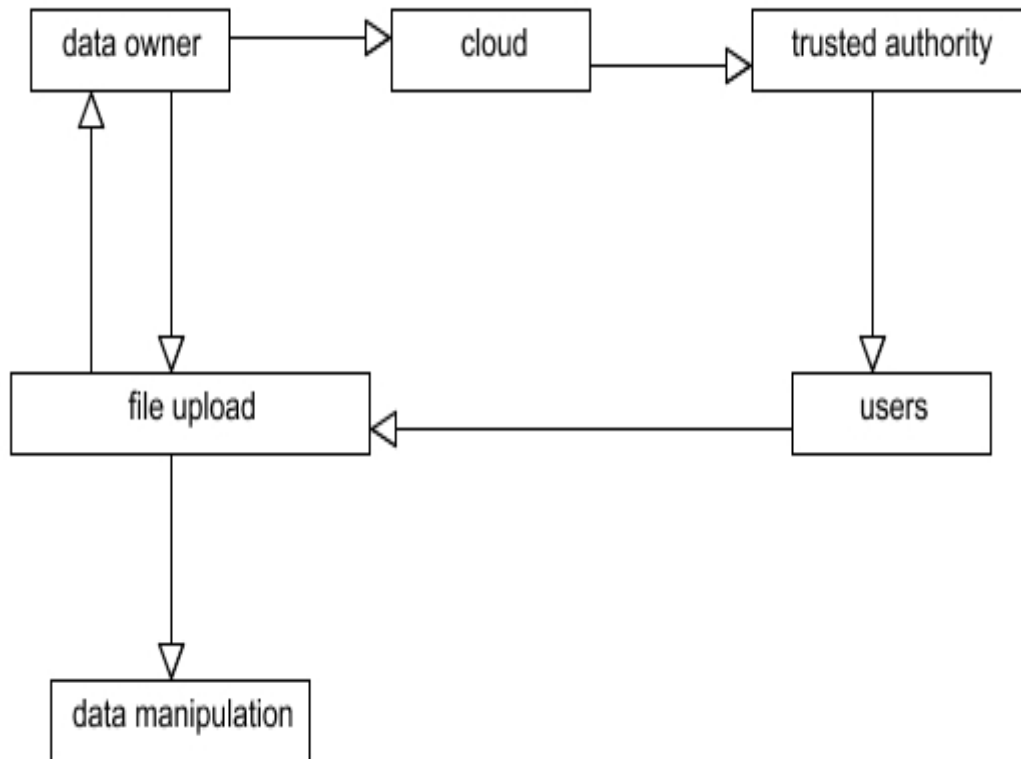
- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



**Fig:7.3 ACTIVITY DIAGRAM**

## 7.4 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

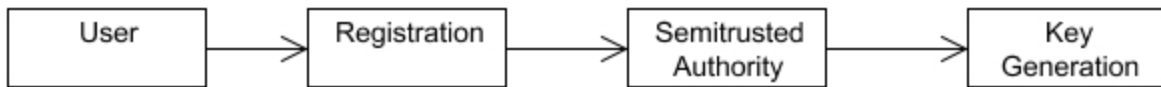


**Fig:7.4 COLLABORATION DIAGRAM**

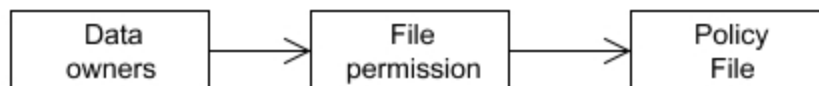
## 7.5 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

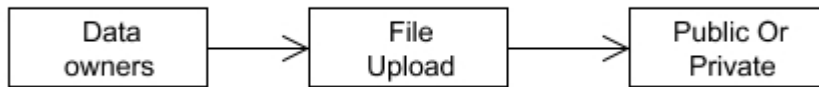
### Level 0:



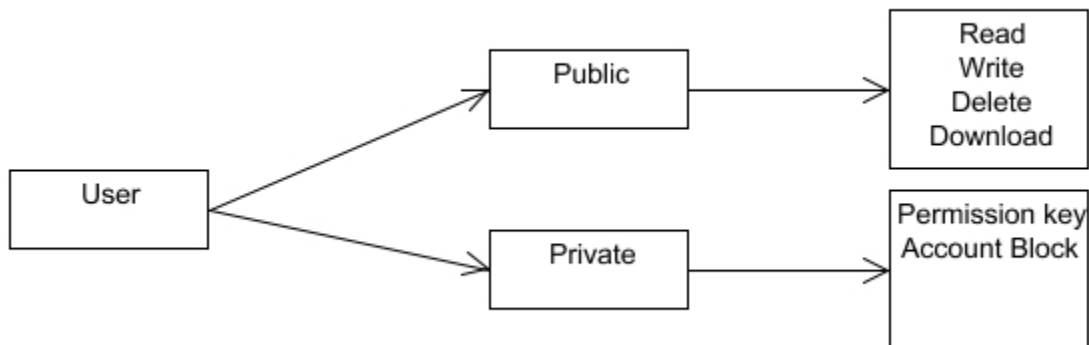
### Level 1:



## Level 2:



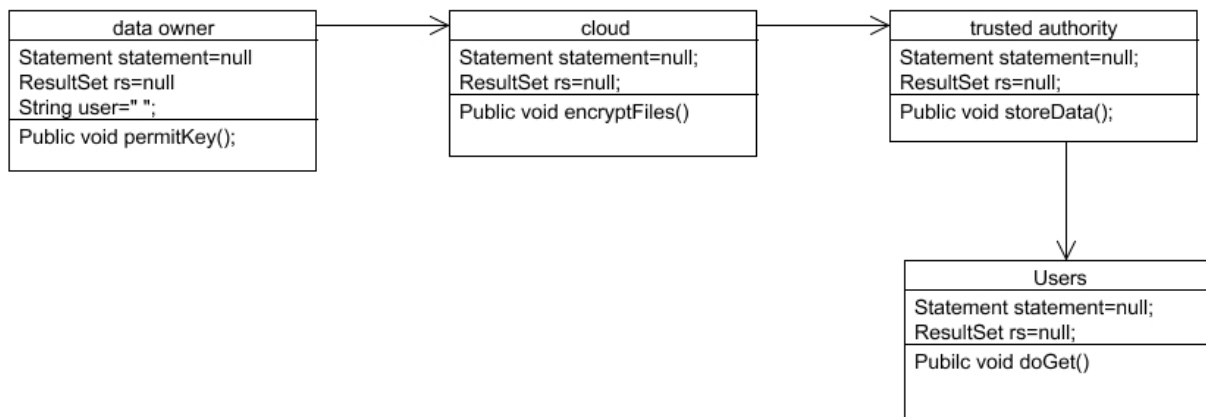
## Level 3:



**Fig:7.5 DATA FLOW DIAGRAM**

## 7.6 Class Diagram

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



**Fig:7.6 CLASS DIAGRAM**

## **CHAPTER 8**

### **CODING AND TESTING**

#### **8.1 CODING STANDARDS**

Once the design aspect of the system is finalized the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

1. Naming conventions
2. Value conventions
3. Script and comment procedure
4. Message box format
5. Exception and error handling

## **NAMING CONVENTIONS**

Naming conventions of classes, data member, member functions, procedures etc., should be self-descriptive. One should even get the meaning and scope of the variable by its name. The conventions are adopted for easy understanding of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

### **Class names**

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

### **Member Function and Data Member name**

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

### **Value Conventions**

Value conventions ensure values for variable at any point of time. This involves the following:

1. Proper default values for the variables.
2. Proper validation of values in the field.
3. Proper documentation of flag values.

## **SCRIPT WRITING AND COMMENTING STANDARD**

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

### **MESSAGE BOX FORMAT**

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

1. X – User has performed illegal operation.
2. ! – Information to the user.

## **8.2 CODING**

AccountabilityPojo.java

```
package logics;
```

```
public class AccountabilityPojo {
```

```
    privateString
```

```
    email,question1,answer1,question2,answer2,question3,answer3,question4,answer4,  
    owner;
```

```
    public String getOwner() {
```

```
        return owner;
```



```

    }

    public void setOwner(String owner) {

        this.owner = owner;

    }

    public String getEmail() {

        return email;

    }


    public void setEmail(String email) {

        this.email = email;

    }

    public String getQuestion1() {

        return question1;

    }

    public void setQuestion1(String question1) {

        this.question1 = question1;

    }

    public String getAnswer1() {

        return answer1;

```

```

    }

    public void setAnswer1(String answer1) {

        this.answer1 = answer1;

    }

    public String getQuestion2() {

        return question2;

    }

    public void setQuestion2(String question2) {

        this.question2 = question2;

    }

    public String getAnswer2() {

        return answer2;

    }

    public void setAnswer2(String answer2) {

        this.answer2 = answer2;

    }

```

EmpChanges.java

```
package logics;
```

```
import java.io.IOException;
```

```

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.http.servlet.HttpServlet;

public class EmpChanges extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String name=request.getParameter("name");

        String email=request.getParameter("email");

        String hemail=request.getParameter("hemail");

        String desig=request.getParameter("desig");

        String mobile=request.getParameter("mobile");

        int status=FileDao.editEmployee(name, email, desig, mobile, hemail);

        if(status>0)

```

```
{  
  
    System.out.println("Employee details changed in a database");  
  
    request.setAttribute("msg", "Employee details changed in a database");  
  
    RequestDispatcher rd=request.getRequestDispatcher("employees.jsp");  
  
    rd.forward(request, response);  
  
}  
  
else  
  
{  
  
    System.out.println("Employee details not changed in a database");  
  
    request.setAttribute("msg", "Employee details not changed in a database");  
  
    RequestDispatcher rd=request.getRequestDispatcher("employees.jsp");  
  
    rd.forward(request, response);  
  
        }  
  
        out.close();  
  
}
```

## **8.3 TEST PROCEDURE**

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

## **8.4 Types of Testing**

### **8.4.1 Unit Testing**

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

### **8.4.2 Functional Tests**

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

### **8.4.3 Integration Testing**

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

## **8.5 TESTING TECHNIQUES**

### **Testing**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Other-wise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

### **8.5.1 White Box Testing**

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

### **8.5.2 Black Box Testing**

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

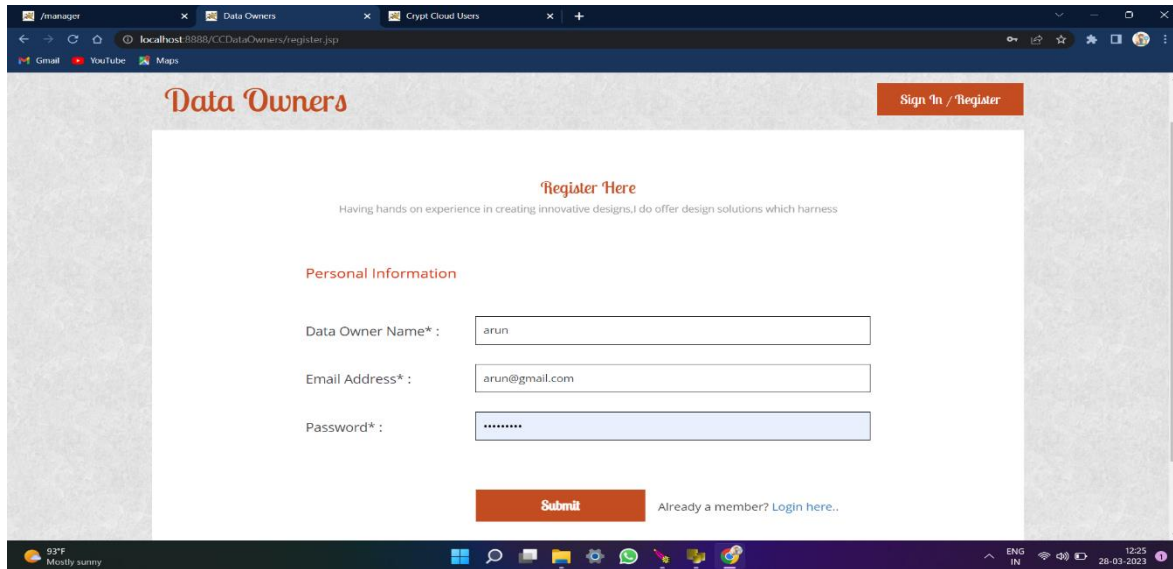
- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing



# CHAPTER 9

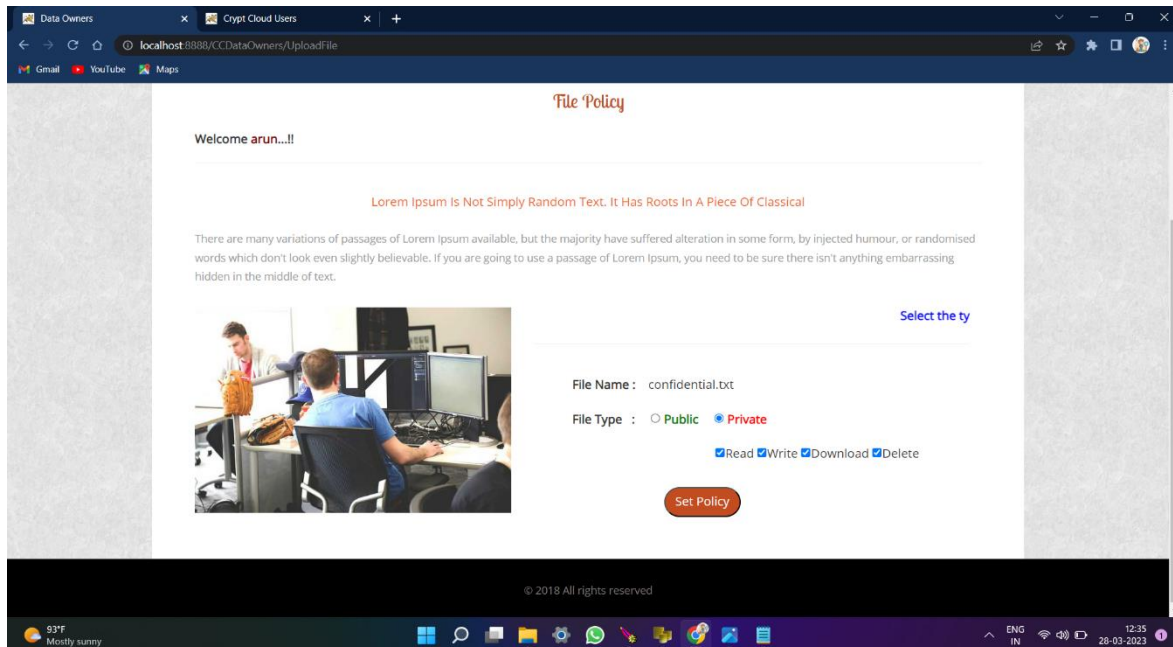
## SCREENSHOTS

### Profile Creation:

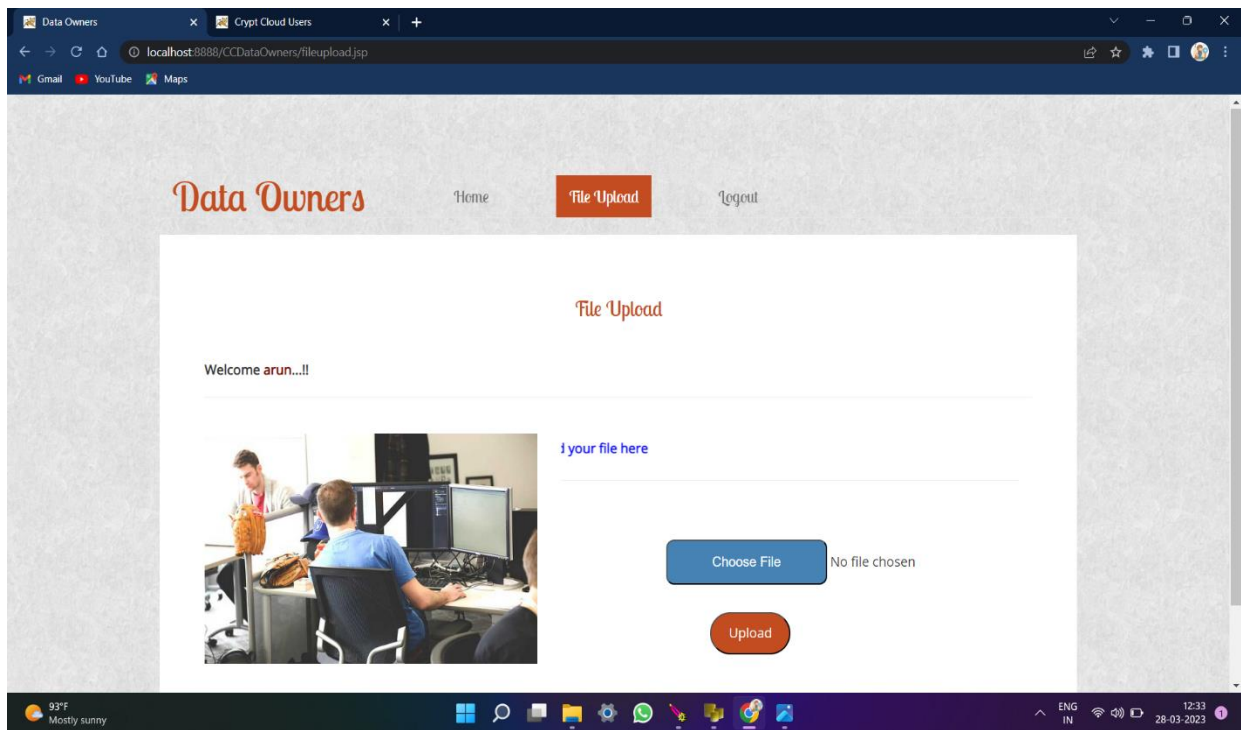


The screenshot shows a web browser window with the URL `localhost:8888/CCDataOwners/register.jsp`. The page has a header with the title "Data Owners" and a "Sign In / Register" button. The main content area is titled "Register Here" and includes a sub-header "Having hands on experience in creating innovative designs,I do offer design solutions which harness". Below this is a "Personal Information" section with three input fields: "Data Owner Name\*" (containing "arun"), "Email Address\*" (containing "arun@gmail.com"), and "Password\*" (containing "\*\*\*\*\*"). A "Submit" button is located below the fields, and a link "Already a member? Login here.." is to its right. The browser's taskbar at the bottom shows the system time as 12:25 on 28-03-2023.

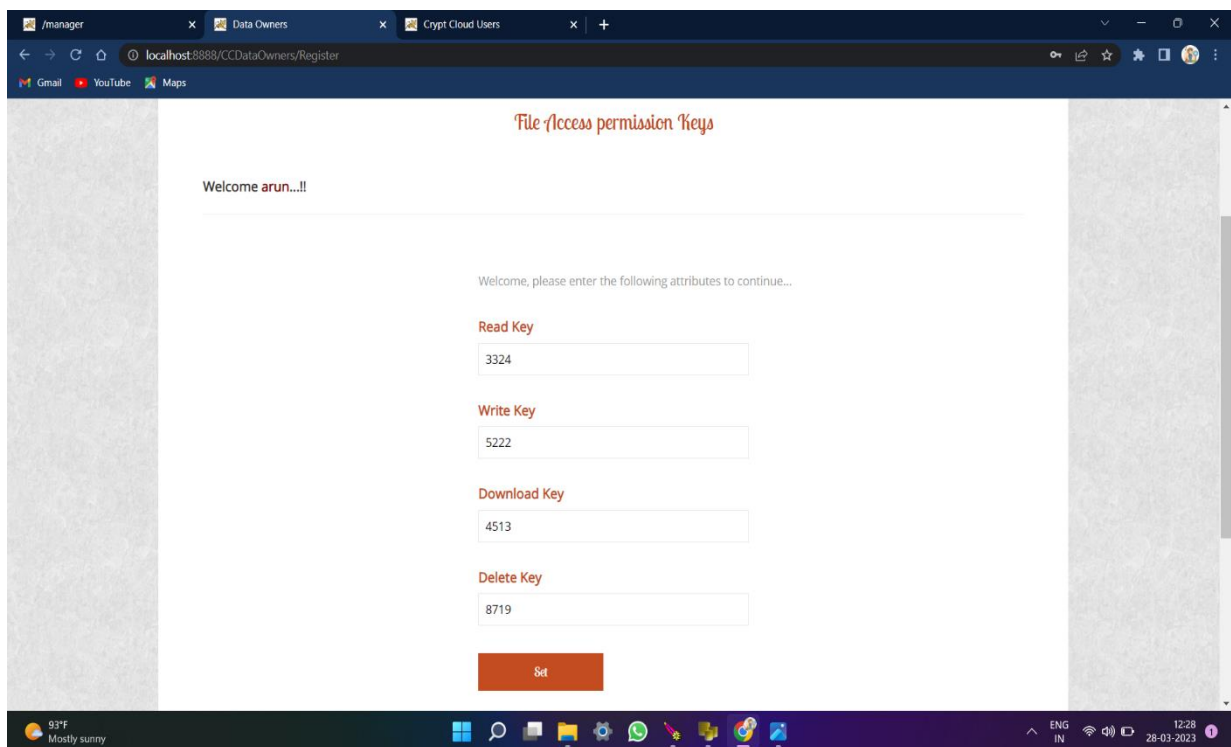
### Data Owners File Upload:



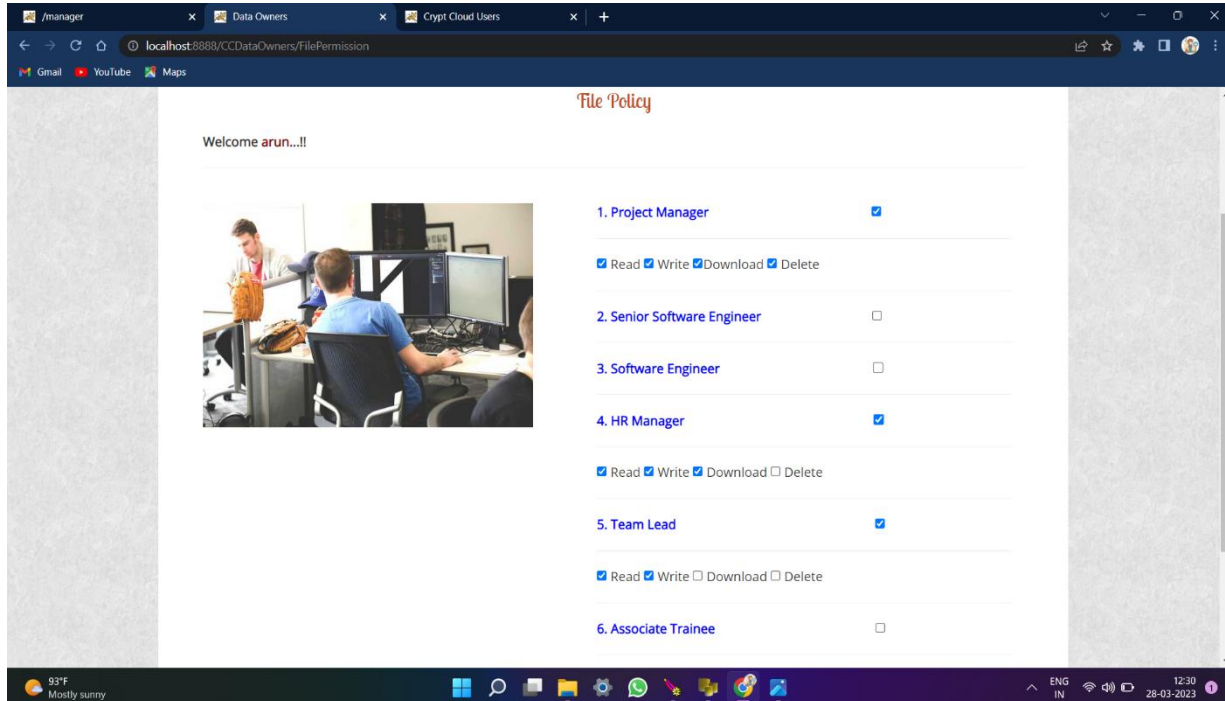
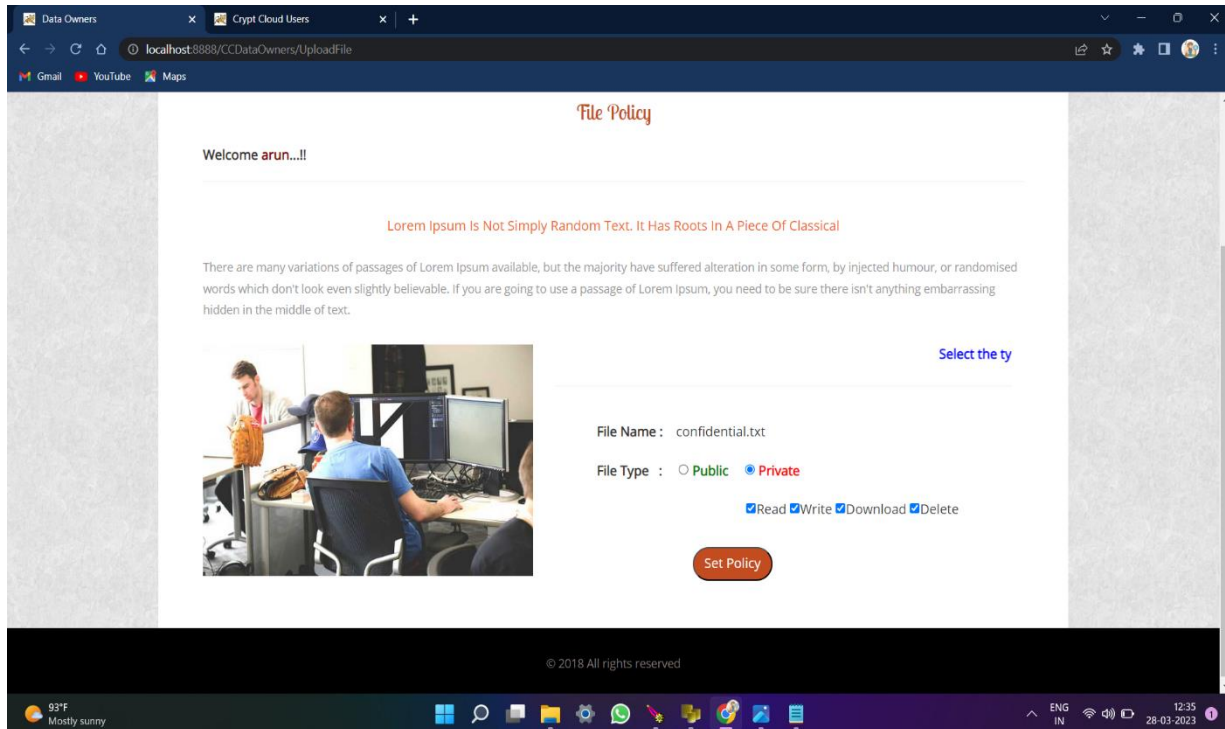
The screenshot shows a web browser window with the URL `localhost:8888/CCDataOwners/UploadFile`. The page has a header with the title "File Policy" and a "Welcome arun...!!" message. The main content area is titled "Lorem Ipsum Is Not Simply Random Text. It Has Roots In A Piece Of Classical" and includes a paragraph of Lorem Ipsum text. Below the text is a "Select the ty" button. To the left of the text is an image of two people working at a computer. To the right of the text is a form with the following fields: "File Name : confidential.txt", "File Type : ☐ Public ☒ Private", and a row of checkboxes: ☒ Read ☒ Write ☒ Download ☒ Delete. A "Set Policy" button is located below the form. The browser's taskbar at the bottom shows the system time as 12:35 on 28-03-2023.



## File Permission Keys:



# File Policy:



## User Sign up:

Quickly a Blogging Category Flai x Demesne a Real estate Category x +

localhost:8888/CCUsers/register.jsp

Gmail YouTube Maps

### User Sign up

Project Manager

jashmax

jashmax@gmail.com

\*\*\*\*\*

9878787879

Chennai

Tamil Nadu

India

Sign Up

Already registered? [Login here...](#)

93°F Mostly sunny

ENG IN 12:39 28-03-2023

Data Owners x Demesne a Real estate Category x Cloud Users x +

localhost:8888/CCUsers/KeyCheck

Gmail YouTube Maps

Welcome **Arasan...**!

Following are your file permissions

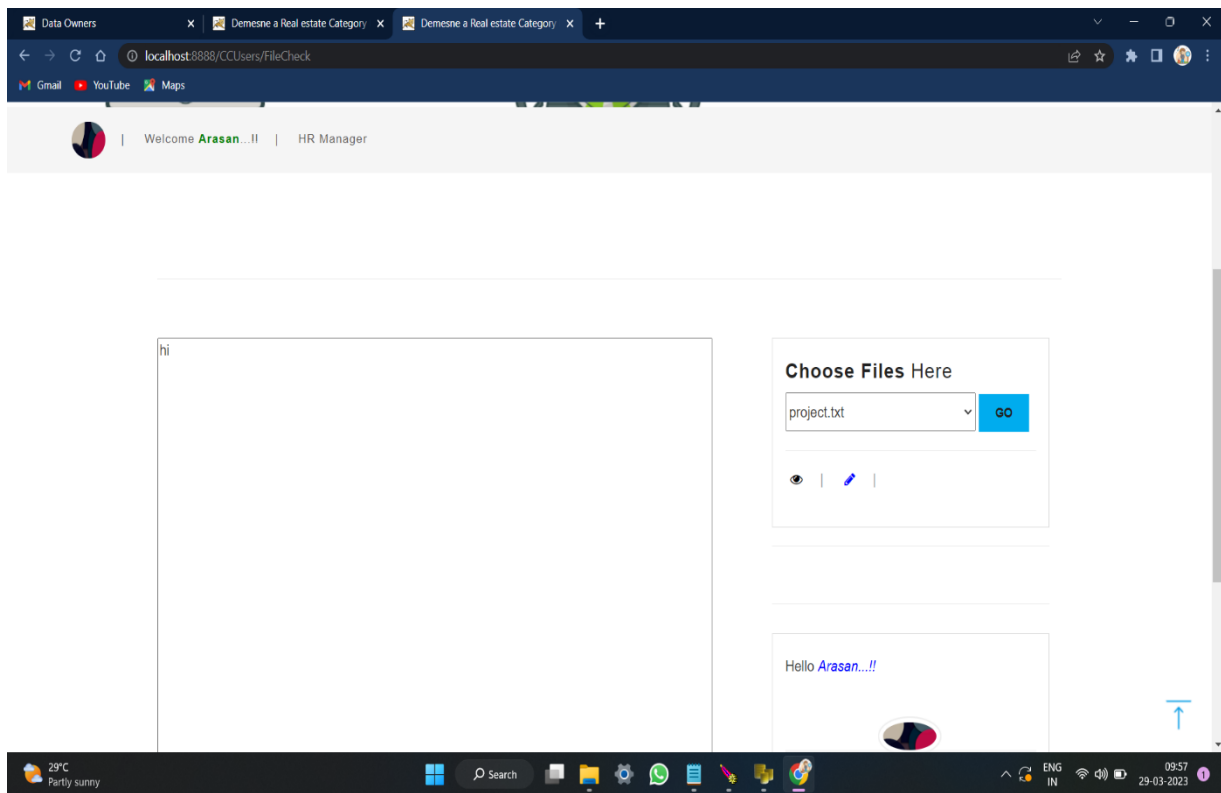
☒ READ ☒ WRITE

SUBMIT

29°C Partly sunny

Search

ENG IN 09:54 29-03-2023





## Semi-Trusted Authority:

The screenshot shows a web browser window with the address bar displaying 'localhost:8888/CCUsers/StatLog'. The page has a header with the title 'Semi-Trusted Authority'. Below the header, there is a green text label 'Generating Decryption Policies Based On User Attributes | Generating Decryption Policies Based On User AI'. The main content area displays a table with 5 columns: S.No, User Email, User Name, User Attributes, and Status. The table contains 8 rows of data. At the bottom right of the main content area, there is a small profile icon and a blue arrow pointing upwards.

S.No	User Email	User Name	User Attributes	Status
1	nitish@gmail.com	Nikitha	Nitishkumar Reddynitish@gmail.com9999999999aaa	Generated
2	jhon@gmail.com	Jhansi	Jhonjhon@gmail.com9999999999chennaiindia	Generated
3	ajith@gmail.com	ajith	ajithajith@gmail.com9856214785ootyTamil NaduIndia	Generated
4	dhoni@gmail.com	dhoni	dhonidhoni@gmail.com9856214785ChennaiTamil NaduIndia	Generated
5	bravo@gmail.com	bravo	bravobravo@gmail.com8596124785ChennaiTamil NaduIndia	Generated
6	manager@gmail.com	manager	managemanager@gmail.com9562314785ChennaiTamil NaduIndia	Generated
7	karthi@gmail.com	karthi	karthikarthi@gmail.com9150269816ChennaiTamil NaduIndia	Generated
8	jashmax@gmail.com	jashmax	jashmaxjashmax@gmail.com987878789ChennaiTamil NaduIndia	

## Tracing who is guilty:

The screenshot shows a web browser window with the URL `localhost:8888/CCDataOwners/theft.jsp`. The page title is "Key Theft List". It displays a welcome message "Welcome rrrr...!!" and a message "Welcome, here are the list of employees who were committed to key theft...". Below this is a table with the following data:

S.No	Employee	Designation	Time	Whose Key
1.	 karthi	associate trainee	22/3/23 11:43 AM	project manager
2.	 manager	hr manager	29/3/23 9:12 AM	project manager

The screenshot shows a web browser window with the URL `localhost:8888/CCUsers/UserLog`. The page title is "Cloud User". It displays a message "Your account is blocked!" with a link "Click here to unblock your account...". Below this is a "User Sign in" form with the following fields:

manag@gmail.com

\*\*\*\*\*

Sign In

Not yet registered? [Register here...](#)

## **CHAPTER 10**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **10.1 Conclusion**

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 Cloud which supports white-box traceability and auditing (referred to as Cloud. This is the first CP-ABE based cloud storage system that simultaneously supports white-box traceability, accountable authority, auditing and effective revocation. Specifically, 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. We note that we may need black-box traceability, which is a stronger notion (compared to white-box traceability), in Cloud. One of our future works is to consider the black-box traceability and auditing.

#### **10.2 Future Enhancement**

Our future work will include extending Cloud to provide “partial” and fully public traceability without compromising on performance.

## REFERENCES

- [1] Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos, Keqin Li, and Albert Y. Zomaya. Sedasc: Secure data sharing in clouds. *IEEE Systems Journal*, 11(2):395–404, 2017.
- [2] Mazhar Ali, Samee U. Khan, and Athanasios V. Vasilakos. Security in cloud computing: Opportunities and challenges. *Inf. Sci.*, 305:357–383, 2015.
- [3] Michael Armbrust, Armando Fox, R ean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
- [4] Nuttapong Attrapadung and Hideki Imai. Attribute-based encryption supporting direct/indirect revocation modes. In *Cryptography and Coding*, pages 278–300. Springer, 2009.
- [5] Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
- [6] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptology-CRYPTO’92*, pages 390–420. Springer, 1993.
- [7] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *EUROCRYPT - 2004*, pages 56–73, 2004.



- [8] Hongming Cai, Boyi Xu, Lihong Jiang, and Athanasios V. Vasilakos. Iot-based big data storage systems in cloud computing: Perspectives and challenges. *IEEE Internet of Things Journal*, 4(1):75–87, 2017.
- [9] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In *Advances in Cryptology - EUROCRYPT 2015*, pages 595–624, 2015.
- [10] Angelo De Caro and Vincenzo Iovino. jpbcc: Java pairing based cryptography. In *ISCC 2011*, pages 850–855. IEEE, 2011.