**TITLE: IDENTIFY-BASED ENCRYPTION TRANSFORMATION FOR FLEXIBLE SHARING**

**TEAM MEMBERS**

**K.DHEERAJ (19S41A0546)**

**L.PRASHANTH (19S41A0547)**

**J.LAXMINIVAS REDDY (19S41A0536)**

**G.KOMURAVELLY (19S41A0531)**

*Under the Guidance of*

**Mr. A.MAHENDER**

Associate Professor



**Department of Computer Science & Engineering**

**VAAGESWARI COLLEGE OF ENGINEERING**

**(Affiliated to JNTU Hyderabad &Approved by AICTE New Delhi)**

**Ramakrishna colony, Karimnagar-505481**

**Department of Computer Science & Engineering**

**VAAGESWARI COLLEGE OF ENGINEERING**

**(Affiliated to JNTU Hyderabad &Approved by AICTE New Delhi)**

**Ramakrishna colony, Karimnagar-505481**

****

#### CERTIFICATE

This is to certify that the project report entitled “**: IDENTIFY-BASED ENCRYPTION TRANSFORMATION FOR FLEXIBLE SHARING”** submitted by the following students in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in CSE, and is a bonafide record of the work performed by.

The work embodied in this project report has not been submitted to any other institution for the award of the degree

**K.DHEERAJ (19S41A0546)**

**L.PRASHANTH (19S41A0547)**

**J.LAXMINIVAS REDDY (19S41A0536)**

**G.KOMURAVELLY (19S41A0531)**

**Internal Guide Head Of The Department**

**Mr. MD SIRAJUDDIN Dr. N. CHANDRAMOULI**

**Associate Professor Associate Professor**

**Principal External Examiner**

**Dr. Ch Srinivas**

**ACKNOWLEDGEMENT**

We wish to pay our sincere thanks to **Dr.Ch. Srinivas**, Principal, Vaageswari College of Engineering, Karimnagar, for providing all required facilities and his support during the project work.

We would like to thank **Dr. N.Chandramouli,** Associate. Professor and HOD of the Computer Science and Engineering Department for his valuable suggestions during the project work.

We sincerely extend our thanks to the project guide, **Mr.MD. SIRAJUDDIN,** Associate Professor of Computer Science and Engineering Department, for sparing his valuable time in guiding the project work and giving feedback with many useful suggestions during the project work**.**

We are also conveying our heartfelt thanks to the Institute authority, Department, Library, and Laboratory staff of Vaageswari College of Engineering for their co-operation during our project. We thank our beloved friends for their help and encouragement regarding the concepts and experimentation

# CHAPTER 1

# INTRODUCTION

* 1. **Problem statement:**

**Cloud computing** is the use of [computing](http://en.wikipedia.org/wiki/Computing) resources (hardware and software) that are delivered as a service over a network (typically the [Internet](http://en.wikipedia.org/wiki/Internet)). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

## Motivation:

The goal of cloud computing is to apply traditional [supercomputing](http://www.webopedia.com/TERM/S/supercomputer.html), or [high-performance computing](http://www.webopedia.com/TERM/H/High_Performance_Computing.html) power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses [networks](http://www.webopedia.com/TERM/N/network.html) of large groups of [servers](http://www.webopedia.com/TERM/S/server.html) typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared [IT](http://www.webopedia.com/TERM/I/IT.html) infrastructure contains large pools of systems that are linked together. Often, [virtualization](http://www.webopedia.com/TERM/V/virtualization.html) techniques are used to maximize the power of cloud computing.

**Characteristics and Services Models:**

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

* **On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service’s provider.
* **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
  1. **Objective:**

In general, we can divide these approaches into four categories: simple cipher text access control, hierarchical access control, access control based on fully homomorphism encryption and access control based on attribute-based encryption (ABE). All these proposals are designed for non-mobile cloud environment

Tysowski et al. considered a specific cloud computing environment where data are accessed by resource-constrained mobile devices, and proposed novel modifications to ABE, which assigned the higher computational overhead of cryptographic operations to the cloud provider and lowered the total communication cost for the mobile user.

# Proposed System:

We propose a Lightweight Data Sharing Scheme (LDSS) for mobile cloud computing environment.

The main contributions of LDSS are as follows:

We design an algorithm called LDSS-CP-ABE based on Attribute-Based Encryption (ABE) method to offer efficient access control over cipher text.

We use proxy servers for encryption and decryption operations. In our approach, computational intensive operations in ABE are conducted on proxy servers, which greatly reduce the computational overhead on client side mobile devices. Meanwhile, in LDSS-CP-ABE, in order to maintain data privacy, a version attribute is also added to the access structure. The decryption key format is modified so that it can be sent to the proxy servers in a secure way.

We introduce lazy re-encryption and description field of attributes to reduce the revocation overhead when dealing with the user revocation problem.

Finally, we implement a data sharing prototype framework based on LDSS.

# Advantages of proposed system:

The experiments show that LDSS can greatly reduce the overhead on the client side, which only introduces a minimal additional cost on the server side.

Such an approach is beneficial to implement a realistic data sharing security scheme on mobile devices.

The results also show that LDSS has better performance compared to the existing ABE based access control schemes over cipher text.

# CHAPTER 2

# TECHNOLOGIES LEARNT

**Software Environment**

## Java Technology

Java technology is both a programming language and a platform.

### The Java Programming Language

### **The Java programming language is a high-level language that can be characterized by all of the following buzzwords:**

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

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



### The Java Platform

A platform is the hardware or software environment in which a program runs. We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and Mac OS. Most platforms can be described as a combination of the operating system and hardware. 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.

The Java platform has two components:

* The Java Virtual Machine (Java VM)
* The Java Application Programming Interface (Java API)

You’ve already been introduced to the Java VM. It’s the base for the Java platform and is ported onto various hardware-based platforms.

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 of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that’s running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

## What Can Java Technology Do?

The most common types of programs written in the Java programming language are applets and applications. If you’ve surfed the Web, you’re probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

* **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
* **Applets**: The set of conventions used by applets.
* **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
* **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
* **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
* **Software components**: Known as JavaBeansTM, can plug into existing component architectures.
* **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
* **Java Database Connectivity (JDBCTM)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



## How Will Java Technology Change My Life?

We can’t promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

* **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it’s easy to learn, especially for programmers already familiar with C or C++.
* **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
* **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people’s tested code and introduce fewer bugs.
* **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
* **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure JavaTM Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
* **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
* **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded “on the fly,” without recompiling the entire program.

### ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn’t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn’t as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

**JDBC**

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

### JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. **SQL Level API**

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

1. **SQL Conformance**

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

1. **JDBC must be implemental on top of common database interfaces**  
   The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.
2. **Provide a Java interface that is consistent with the rest of the Java system**

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

1. **Keep it simple**

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

1. **Use strong, static typing wherever possible**

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

1. **Keep the common cases simple**

Because more often than not, the usual SQL calls used by the programmer are simple SELECT’s, INSERT’s, DELETE’s and UPDATE’s, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

Java ha two things: a programming language and a platform.

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

Simple Architecture-neutral

Object-oriented Portable

Distributed High-performance

Interpreted multithreaded

Robust Dynamic

Secure

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

**JavaProgram**

**Compilers**

**Interpreter**

**My Program**

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make “write once, run anywhere” possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

## Networking

### TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

### IP datagram’s

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

### UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

### TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

### Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

### Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

### Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

### Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

### Total address



The 32 bit address is usually written as 4 integers separated by dots.

### Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

### Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

#include <sys/types.h>

#include <sys/socket.h>

int socket(int family, int type, int protocol);

Here "family" will be AF\_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

**JFree Chart**

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, [free software](http://www.gnu.org/philosophy/free-sw.html). It is distributed under the terms of the [GNU Lesser General Public Licence](http://www.gnu.org/licenses/lgpl.html) (LGPL), which permits use in proprietary applications.

## 1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

## 2. Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

## 3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

## 4. Property Editors

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

**J2ME (Java 2 Micro edition):-**

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

**1. General J2ME architecture**



J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

**2.Developing J2ME applications**

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role preverification plays in this process.

**3.Design considerations for small devices**

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

\* Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.

\* Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.

\* Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

**4.Configurations overview**

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

\* **Connected Limited Device Configuration (CLDC)**is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

\* **Connected Device Configuration (CDC)**is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

**5.J2ME profiles**

**What is a J2ME profile?**

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

**Profile 1: KJava**

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava contains a Sun-specific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is com.sun.kjava. We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

**Profile 2: MIDP**

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile application

development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

\* java.lang

\* java.io

\* java.util

\* javax.microedition.io

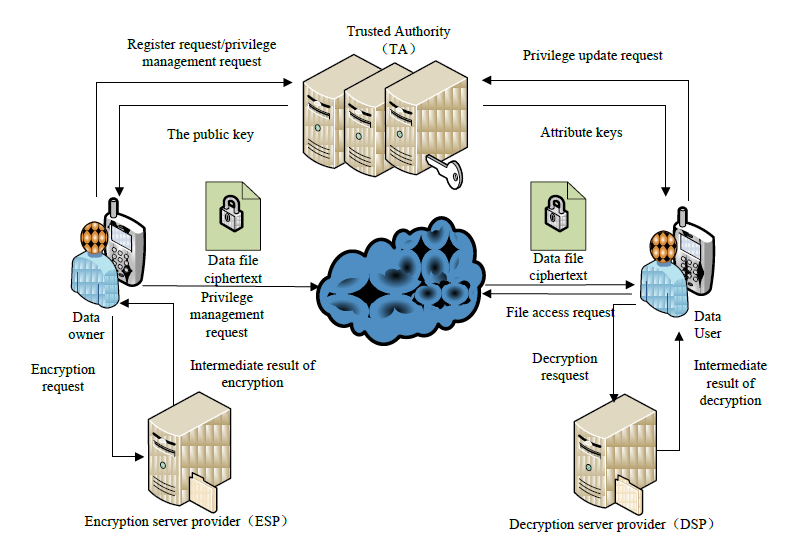
\* javax.microedition.lcdui

\* javax.microedition.midlet

\* javax.microedition.rms

# CHAPTER 3SYSTEM DESIGN

**3.1 System Architecture**

****

**3.2Moduledescription**

**System Framework:**

The development of cloud computing and the popularity of smart mobile devices, people are gradually getting accustomed to a new era of data sharing model in which the data is stored on the cloud and the mobile devices are used to store/retrieve the data from the cloud.In these applications, people (data owners) can upload their documents and other files to the cloud and share these data with other people (data users) they like to share. CSPs also provide data management functionality for data owners. Since personal data files are sensitive, data owners are allowed to choose whether to make their data files public or can only be shared with specific data users. Clearly, data privacy of the personal sensitive data is a big concern for many data owners. We propose LDSS, a framework of lightweight data sharing scheme in mobile cloud. It has the following six components. (1)Data Owner (DO) (2) Data User (DU) (3) Trust Authority (TA) (4) Encryption Service Provider (ESP) (5) Decryption Service Provider (DSP) (6) Cloud Service Provider (CSP).

**Data Owner (DO):**

When the data owner (DO) registers on TA, TA runs the algorithm Setup() to generate a public key PK and a master key MK. PK is sent to DO while MK is kept on TA itself. DO defines its own attribute set and assigns attributes to its contacts. All these information will be sent to TA and the cloud. TA and the cloud receive the information and store it. DO uploads data to the mobile cloud and share it with friends. DO determines the access control policies. DO sends data to the cloud. Since the cloud is not credible, data has to be encrypted before it is uploaded. The DO defines access control policy in the form of access control tree on data files to assign which attributes a DU should obtain if he wants to access a certain data file.

**Data User (DU):**

DU logins onto the system and sends, an authorization request to TA. The authorization request includes attribute keys (SK) which DU already has.TA accepts the authorization request and checks the request and a generate attribute keys (SK) for DU. DU sends a request for data to the cloud.Cloud receives the request and checks if the DU meets the access requirement. DU receives the ciphertext, which includes ciphertext of data files and ciphertext of the symmetric key. DU decrypt the ciphertext of the symmetric key with the assistance of DSP. DU uses the symmetric key to decrypt the ciphertext of data files.

**Trusted Authority:**

To make LDSS feasible in practice, a trusted authority (TA) is introduced. It is responsible of generating public and private keys, and distributing attribute keys to users. With this mechanism, users can share and access data without being aware of the encryption and decryption operations. We assume TA is entirely credible, and a trusted channel exists between the TA and every user. The fact that a trusted channel exists doesn’t mean that the data can be shared through the trusted channel, for the data can be in a large amount. TA is only used to transfer keys (in a small amount) securely between users. In addition, it’s requested that TA is online all the time because data users may access data at any time and need TA to update attribute keys.

**Cloud Service Provider:**

CSP stores the data for DO. It faithfully executes the operations requested by DO, while it may peek over data that DO has stored in the cloud. DU sends a request for data to the cloud. Cloud receives the request and checks if the DU meets the access requirement. If DU can’t meet the requirement, it refuses the request; otherwise it sends the ciphertext to DU. CSP manages the Uploaded Files.

# SystemSpecification

* + 1. **SoftwareRequirements**

Functional requirements for a secure cloud storage service are straightforward:

1. The service should be able to store the user’sdata;
2. The data should be accessible through any devices connected to theInternet;
3. The service should be capable to synchronize the user’s data between multiple devices (notebooks, smart phones,etc.);
4. The service should preserve all historical changes(versioning);
5. Data should be shareable with otherusers;
6. The service should support SSO;and
7. The service should be interoperable with other cloud storage services, enabling data migration from one CSP toanother.

Operating System : Windows XP

Application Server : Tomcat 5.0

Front End : HTML, Java, Jsp

Scripts : JavaScript.

Server side Script : Java Server Pages.

Database : Mysql 5.0

Database Connectivity: JDBC.

# HardwareRequirements:

* + Processor : Intel Core2Duo
  + Seed : 2.4GHz
  + RAM : 2GB(minimum)
  + HardDisk : 180 GB

Cloud computing has three fundamental models, these are:

* 1. **DetailedDesign**

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it’s one of the most popular [business process modeling techniques](https://tallyfy.com/business-process-modeling-techniques).

It is based on **diagrammatic representations** of software components. As the old proverb says: “a picture is worth a thousand words”. By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at [Rational Software](https://en.wikipedia.org/wiki/Rational_Software). It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates.

## GOALS:

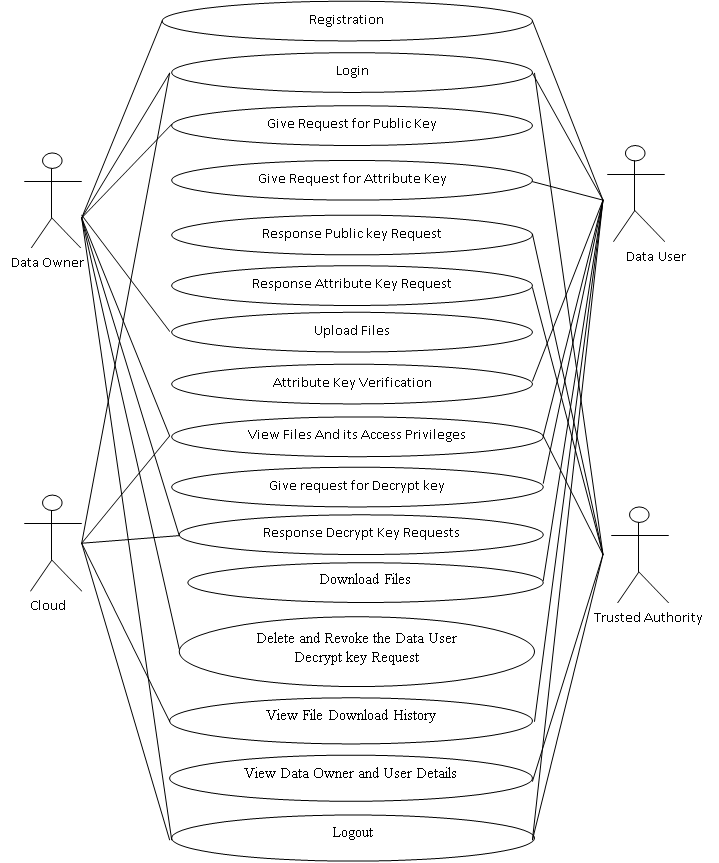
The Primary goals in the design of the UML are as follows:

* 1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningfulmodels.
  2. Provide extendibility and specialization mechanisms to extend the coreconcepts.
  3. Be independent of particular programming languages and developmentprocess.
  4. Provide a formal basis for understanding the modelinglanguage.
  5. Encourage the growth of OO toolsmarket.

6 Support higher level development concepts such as collaborations, frameworks, patterns and components.

1. Integrate best practices.
   * 1. **USE CASEDIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



# SEQUENCEDIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



# CLASSDIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) 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 the classes. It explains which class contains information.

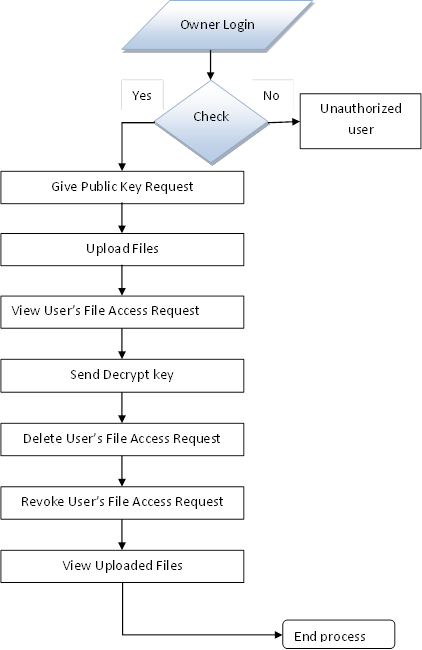


**Data Flow diagram :-**

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow..

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

****

# CHAPTER 4IMPLEMENTATION

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@page import ="java.util.\*"%>

<%@ include file="connect.jsp" %>

<%@page import ="java.util.\*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>

<%@ page import="java.sql.\*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileInputStream,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.BufferedInputStream" %>

<%@ page import ="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>

<%@page import ="java.util.\*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutputStream,java.io.PrintStream"%>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>CLOUD SERVICE PROVIDERS </title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-titillium-250.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style32 {

font-size: 12px;

font-weight: bold;

color: #FF0000;

}

.style33 {font-size: 12px; color: #6d6d6d; }

.style34 {

font-size: 24px;

font-weight: bold;

}

.style35 {

color: #FF0000;

font-weight: bold;

}

.style36 {font-weight: bold}

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header\_resize">

<div class="logo">

<h1><a href="index.html" class="style34">A Distributed Trust Evaluation Protocol<br />

with Privacy Protection for Intercloud</a></h1>

</div>

<div class="menu\_nav">

<ul>

<li><a href="C\_Main.jsp">CLOUD SERVICE PROVIDERS</a></li>

<li><a href="C\_Login.jsp"><span>Logout </span></a></li>

<li></li>

</ul>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"><a href="#"><img src="images/slide1.jpg" width="960" height="360" alt="" /></a><a href="#"><img src="images/slide2.jpg" width="960" height="360" alt="" /></a><a href="#"><img src="images/slide3.jpg" width="960" height="360" alt="" /></a></div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content\_resize">

<div class="mainbar">

<div class="article">

<h2> View File Details !!! </h2>

<p>&nbsp;</p>

<%

String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8="",s9="",s10,s11,s12,s13;

int i=0,j=0,k=0;

int uid = Integer.parseInt(request.getParameter("usid"));

try

{

String query="Select \*from cloudserver where id='"+uid+"'";

Statement st=connection.createStatement();

ResultSet rs=st.executeQuery(query);

while ( rs.next() )

{

i=rs.getInt(1);

s2=rs.getString(2);

s3=rs.getString(3);

s4=rs.getString(4);

s5=rs.getString(5);

s6=rs.getString(6);

s7=rs.getString(7);

s8=rs.getString(8);

s9=rs.getString(9);

}

%>

<table width="549" border="1" style="border-collapse:collapse" cellpadding="0" cellspacing="0" align="center">

<tr>

<td width="142" height="36" bgcolor="#FFFF00"><div align="center" class="style29 style32">

<div align="left"> ID :</div>

</div></td>

<td width="394"><label>

<input readonly="readonly" name="textfield" type="text" value="<%=i%>" size="49" />

</label></td>

</tr>

<tr>

<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">

<div align="left">File Name : </div>

</div></td>

<td><input readonly="readonly" name="textfield2" type="text" size="49" value="<%=s2%>" /></td>

</tr>

<tr>

<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">

<div align="left">Index : </div>

</div></td>

<td><input readonly="readonly" name="textfield22" type="text" size="49" value="<%=s3%>" /></td>

</tr>

<tr>

<td height="186" bgcolor="#FFFF00"><div align="center" class="style32">

<div align="left">Contents :</div>

</div></td>

<td><label>

<textarea readonly="readonly" rows="10" cols="50" name="textarea"><%=s4%></textarea>

</label></td>

</tr>

<tr>

<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">

<div align="left">MAC :</div>

</div></td>

<td><input readonly="readonly" name="textfield3" type="text" value="<%=s5%>" size="49" /></td>

</tr>

<tr>

<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">

<div align="left">Rank : </div>

</div></td>

<td><input readonly="readonly" name="textfield32" type="text" value="<%=s7%>" size="49" /></td>

</tr>

<tr>

<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">

<div align="left">Date &amp; Time :</div>

</div></td>

<td><input readonly="readonly" name="textfield33" type="text" value="<%=s8%>" size="49" /></td>

</tr>

<tr>

<td height="36" bgcolor="#FFFF00"><div align="center" class="style33">

<div align="left" class="style35">USERS: </div>

</div></td>

<td><input readonly="readonly" name="textfield34" type="text" value="<%=s9%>" size="49" /></td>

<%

connection.close();

}

catch(Exception e)

{

out.println(e.getMessage());

}

%>

</tr>

</table>

<p align="right"><a href="C\_ViewFiles.jsp">Back</a></p>

</div>

</div>

<div class="sidebar">

<div class="gadget">

<h2 class="star">Menu</h2>

<div class="clr"></div>

<ul class="sb\_menu style36">

<li><a href="C\_Main.jsp">Home</a></li>

<li><a href="C\_Login.jsp">Logout</a></li>

</ul>

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="footer">

<div class="footer\_resize">

<p class="lf">&nbsp;</p>

<p class="rf">&nbsp;</p>

<div style="clear:both;"></div>

</div>

</div>

</div>

<div align=center></div>

</body>

</html>

<%@page import="java.util.\*"%>

<%@ include file="connect.jsp"%>

<%@page

import="java.util.\*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>

<%@ page

import="java.sql.\*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileInputStream,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.BufferedInputStream"%>

<%@ page

import="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>

<%@page

import="java.util.\*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutputStream,java.io.PrintStream"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Attack</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-aller.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style1 {font-size: 24px}

.style2 {color: #0000FF}

.style3 {color: #5E6D4E}

.style4 {font-weight: bold}

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header\_resize">

<div class="logo">

<h1><a href="index.html" class="style1">An Efficient Multi-User Searchable Encryption Scheme without Query Transformation Over Outsourced Encrypted Data

</a></h1>

</div>

<div class="menu\_nav">

<ul>

<li class="active"><a href="index.html"><span>Home</span> Page </a></li>

<li><a href="DataOwnerLogin.html">Data Owner </a></li>

<li><a href="EndUserLogin.html">End User</a></li>

<li><a href="CloudServerLogin.html">Cloud Server</a>

</ul>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"><a href="#"><img src="images/slide1.jpg" width="960" height="320" alt="" /></a><a href="#"><img src="images/slide2.jpg" width="960" height="320" alt="" /></a><a href="#"><img src="images/slide3.jpg" width="960" height="320" alt="" /></a></div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content\_resize">

<div class="mainbar">

<div class="article">

<h2><span>WELCOME TO HOME PAGE </span></h2>

<p class="infopost"><a href="#" class="com"><span>11</span></a></p>

<div class="clr"></div>

<div class="clr"></div>

</div>

<div class="article">

<h2><span></span><form action="Attack1.jsp" method="post" name="form1" id="form1">

<table width="616" border="0" align="left">

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td width="333"><span class="style1">Enter File Name :-</span></td>

<td width="273"><label><input required name="t1"

type="text" size="40" /></label></td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td><span class="style1">Ur Name :-</span></td>

<td><input name="ot" type="text" size="40" /></td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>

<div align="right"><input type="submit" name="Submit"

value="Attack" /></div>

</td>

</tr>

</table>

</form></h2>

<p class="infopost">&nbsp;</p>

<div class="clr"></div>

<div class="clr"></div>

</div>

</div>

<div class="sidebar">

<div class="searchform">

<form id="formsearch" name="formsearch" method="post" action="#">

<span>

<input name="editbox\_search" class="editbox\_search" id="editbox\_search" maxlength="80" value="Search our ste:" type="text" />

</span>

<input name="button\_search" src="images/search.gif" class="button\_search" type="image" />

</form>

</div>

<div class="clr"></div>

<div class="gadget">

<h2 class="star"><span>Home</span> Menu</h2>

<div class="clr"></div>

<ul class="sb\_menu">

<li><a href="index.html">Home Page </a></li>

<li><a href="DataOwnerLogin.html">Data Owner </a></li>

<li><a href="EndUserLogin.html">End User</a></li>

<li><a href="CloudServerLogin.html">Cloud Server</a></li>

</ul>

</div>

<div class="gadget">

<h2 class="star">&nbsp;</h2>

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="fbg">

<div class="fbg\_resize">

<div class="clr"></div>

</div>

</div>

<div class="footer">

<div class="footer\_resize">

<div style="clear:both;"></div>

</div>

</div>

</div>

</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import ="java.util.\*"%>

<%@page import ="java.sql.\*"%>

<%@page import ="java.util.\*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>

<%@ page import="java.sql.\*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileInputStream,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.BufferedInputStream" %>

<%@ page import ="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>

<%@page import ="java.util.\*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutputStream,java.io.PrintStream"%>

<%@ include file="connect.jsp" %>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>CLOUD SERVICE PROVIDERS </title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-titillium-250.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style21 {font-size: 14px}

.style24 {color: #FF0000}

.style28 {font-size: 24px}

.style29 {font-size: 14px; color: #FF0000; font-weight: bold; }

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header\_resize">

<div class="logo">

<h1><a href="index.html" class="style28">A Distributed Trust Evaluation Protocol<br />

with Privacy Protection for Intercloud</a></h1>

</div>

<div class="menu\_nav">

<ul>

<li><a href="C\_Main.jsp">CLOUD SERVICE PROVIDERS</a></li>

<li><a href="C\_Login.jsp"><span>Logout </span></a></li>

<li></li>

</ul>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"><a href="#"><img src="images/slide1.jpg" width="960" height="360" alt="" /></a><a href="#"><img src="images/slide2.jpg" width="960" height="360" alt="" /></a><a href="#"><img src="images/slide3.jpg" width="960" height="360" alt="" /></a></div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content\_resize">

<div class="mainbar">

<div class="article">

<h2><span>Grant Search Control </span></h2>

<p>&nbsp;</p>

<table width="565" border="2" cellpadding="0" style="border-collapse:collapse" cellspacing="0" align="center">

<tr>

<td width="99" height="30" bgcolor="#FFFF00"><div align="center" class="style21 style24"><strong> ID </strong></div></td>

<td width="174" bgcolor="#FFFF00"><div align="center" class="style29">User Name </div></td>

<td width="252" bgcolor="#FFFF00"><div align="center" class="style29">Permission</div></td>

<%

String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8,s9="",s10,s11,s12,s13;

int i=0,j=0,k=0;

try

{

String query="select \* from request";

Statement st=connection.createStatement();

ResultSet rs=st.executeQuery(query);

while ( rs.next() )

{

i=rs.getInt(1);

s2=rs.getString(2);

s3=rs.getString(3);

%>

</tr>

<tr>

<td height="28"><div align="center" class="style24"><%=i%></div></td>

<td><div align="center" class="style24"><a href="C\_UserSGDetails.jsp?uname=<%=s2%>"><%=s2%></a></div></td>

<%

if(s3.equalsIgnoreCase("Requested"))

{

%>

<td><div class="style24">

<div align="center"><a href="C\_GrantSearch1.jsp?usid=<%=i%>"><%=s3%></a></div>

</div></td>

<%

}else

{

%>

<td width="28"><div class="style24">

<div align="center"><%=s3%></div>

</div></td>

</tr>

<%

}

}

connection.close();

}

catch(Exception e)

{

out.println(e.getMessage());

}

%>

</table>

<p>&nbsp;</p>

<p align="right"><a href="C\_Main.jsp">Back</a></p>

</div>

</div>

<div class="sidebar">

<div class="gadget">

<h2 class="star">Menu</h2>

<div class="clr"></div>

<ul class="sb\_menu">

<li><strong><a href="C\_Main.jsp">Home</a></strong></li>

<li><strong><a href="C\_Login.jsp">Logout</a></strong></li>

</ul>

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="fbg"></div>

<div class="footer">

<div class="footer\_resize">

<p class="lf">&nbsp;</p>

<p class="rf">&nbsp;</p>

<div style="clear:both;"></div>

</div>

</div>

</div>

<div align=center></div>

</body>

</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@page import ="java.util.\*"%>

<%@page import ="java.sql.\*"%>

<%@page import ="java.util.\*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>

<%@ page import="java.sql.\*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileInputStream,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.BufferedInputStream" %>

<%@ page import ="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>

<%@page import ="java.util.\*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutputStream,java.io.PrintStream"%>

<%@ include file="connect.jsp" %>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>CLOUD SERVICE PROVIDERS </title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-titillium-250.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style21 {font-size: 14px}

.style24 {color: #FF0000}

.style28 {font-size: 24px}

.style29 {font-size: 14px; color: #FF0000; font-weight: bold; }

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header\_resize">

<div class="logo">

<h1><a href="index.html" class="style28">A Distributed Trust Evaluation Protocol<br />

with Privacy Protection for Intercloud</a></h1>

</div>

<div class="menu\_nav">

<ul>

<li><a href="C\_Main.jsp">CLOUD SERVICE PROVIDERS</a></li>

<li><a href="C\_Login.jsp"><span>Logout </span></a></li>

<li></li>

</ul>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"><a href="#"><img src="images/slide1.jpg" width="960" height="360" alt="" /></a><a href="#"><img src="images/slide2.jpg" width="960" height="360" alt="" /></a><a href="#"><img src="images/slide3.jpg" width="960" height="360" alt="" /></a></div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content\_resize">

<div class="mainbar">

<div class="article">

<h2><span>Grant Search Control </span></h2>

<p>&nbsp;</p>

<table width="565" border="2" cellpadding="0" style="border-collapse:collapse" cellspacing="0" align="center">

<tr>

<td width="99" height="30" bgcolor="#FFFF00"><div align="center" class="style21 style24"><strong> ID </strong></div></td>

<td width="174" bgcolor="#FFFF00"><div align="center" class="style29">User Name </div></td>

<td width="252" bgcolor="#FFFF00"><div align="center" class="style29">Permission</div></td>

<%

String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8,s9="",s10,s11,s12,s13;

int i=0,j=0,k=0;

try

{

String query="select \* from request";

Statement st=connection.createStatement();

ResultSet rs=st.executeQuery(query);

while ( rs.next() )

{

i=rs.getInt(1);

s2=rs.getString(2);

s3=rs.getString(3);

%>

</tr>

<tr>

<td height="28"><div align="center" class="style24"><%=i%></div></td>

<td><div align="center" class="style24"><a href="C\_UserSGDetails.jsp?uname=<%=s2%>"><%=s2%></a></div></td>

<%

if(s3.equalsIgnoreCase("Requested"))

{

%>

<td><div class="style24">

<div align="center"><a href="C\_GrantSearch1.jsp?usid=<%=i%>"><%=s3%></a></div>

</div></td>

<%

}else

{

%>

<td width="28"><div class="style24">

<div align="center"><%=s3%></div>

</div></td>

</tr>

<%

}

}

connection.close();

}

catch(Exception e)

{

out.println(e.getMessage());

}

%>

</table>

<p>&nbsp;</p>

<p align="right"><a href="C\_Main.jsp">Back</a></p>

</div>

</div>

<div class="sidebar">

<div class="gadget">

<h2 class="star">Menu</h2>

<div class="clr"></div>

<ul class="sb\_menu">

<li><strong><a href="C\_Main.jsp">Home</a></strong></li>

<li><strong><a href="C\_Login.jsp">Logout</a></strong></li>

</ul>

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="fbg"></div>

<div class="footer">

<div class="footer\_resize">

<p class="lf">&nbsp;</p>

<p class="rf">&nbsp;</p>

<div style="clear:both;"></div>

</div>

</div>

</div>

<div align=center></div>

</body>

</html>

**CHAPTER – 5**

**TEST RESULTS**

**Types of Testing**

**The basic levels of Testing:**

Client needs acceptance testing



Requirements system testing



Design  integration testing



Code unit testing

**Functional Testing**

Real tests give efficient protests that functions tested are attainable as specific by the business and technical requirements, system documentation, and user manuals.

: known categories of application outputs should be exercised.

Systems/Procedures: interfacing systems or procedures should be invoked.

**TYPES OF TESTS**

**Unit testing:**

A unit is the smallest piece of source code that can be tested. It is also known as a module which consists of numerous lines of code that are processed by a single programmer. The key purpose of performing unit testing is to expose that a particular unit doesn’t satisfy the specified functional requirements and also to show that the structural implementation is not like to the projected structure designed.

**Integration testing:**

tests are intended to test incorporated programming segments to figure out whether they really keep running as one system. Testing is occasion driven and is more worried with the fundamental result of screens or fields. Reconciliation tests exhibit that in spite of the fact that the parts were separately fulfillment, as appeared by effectively unit testing, the blend of segments is right and comprised. Integration testing is specifically aimed at revealing the problems that rise from the mixture of components.

**Functional test**

Functional tests give efficient challenges that capacities tried are accessible as determined by the business and specialized necessities, framework documentation, and client manuals Functional testing is centered on the following items:

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Association and arrangement of practical tests is centered around prerequisites, key capacities, or unique experiments. Likewise, efficient scope relating to recognize Business procedure streams; information fields, predefined procedures, and progressive procedures must be considered for testing. Before utilitarian testing is finished, extra tests are distinguished and the powerful estimation of current tests is resolved.

**System Test**

System testing guarantees that the entire coordinated programming framework meets prerequisites. It tests a design to guarantee known and unsurprising results. A sample of framework testing is the arrangement situated framework combination test. Framework testing depends on procedure portrayals and streams, stressing pre-driven procedure connections and mix focuses.

**White Box Testing**

It is a testing in which the product analyzer has information of the internal workings, structure and dialect of the product, or if nothing else its motivation. It is reason. It is utilized to test ranges that can't be gotten a handle on from a discovery level.

**Black Box Testing**

It is the testing the product with no information of within workings, structure or dialect of the part being tried. Discovery tests, as most different sorts of tests, must be composed from a complete source report, for example, prerequisite or necessities archive, for example, determination or necessities record. It is a trying in which the product under test is dealt with, as a discovery .you can't "see" into it. The test gives inputs and reacts to yields without considering how the product functions.

**Unit Testing:**

Unit testing is by and large appeared as a major aspect of a joined code and unit test period of the product lifecycle, in spite of the fact that it is not exceptional for coding and unit testing to be directed as two unmistakable stages.

**Test strategy and approach**

Ground testing will be done physically and functional tests will be inscribed in detail.

**Test objectives**

* All field admissions essentially work appropriately.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Validate that the accesses are of the correct format
* No duplicate entries should be allowed
* Entire links must gross the user to the accurate page.

**Integration Testing**

Software integration testing is the incremental combination analysis of two or more joint software components on a single platform to generate failures created by boundary faults.

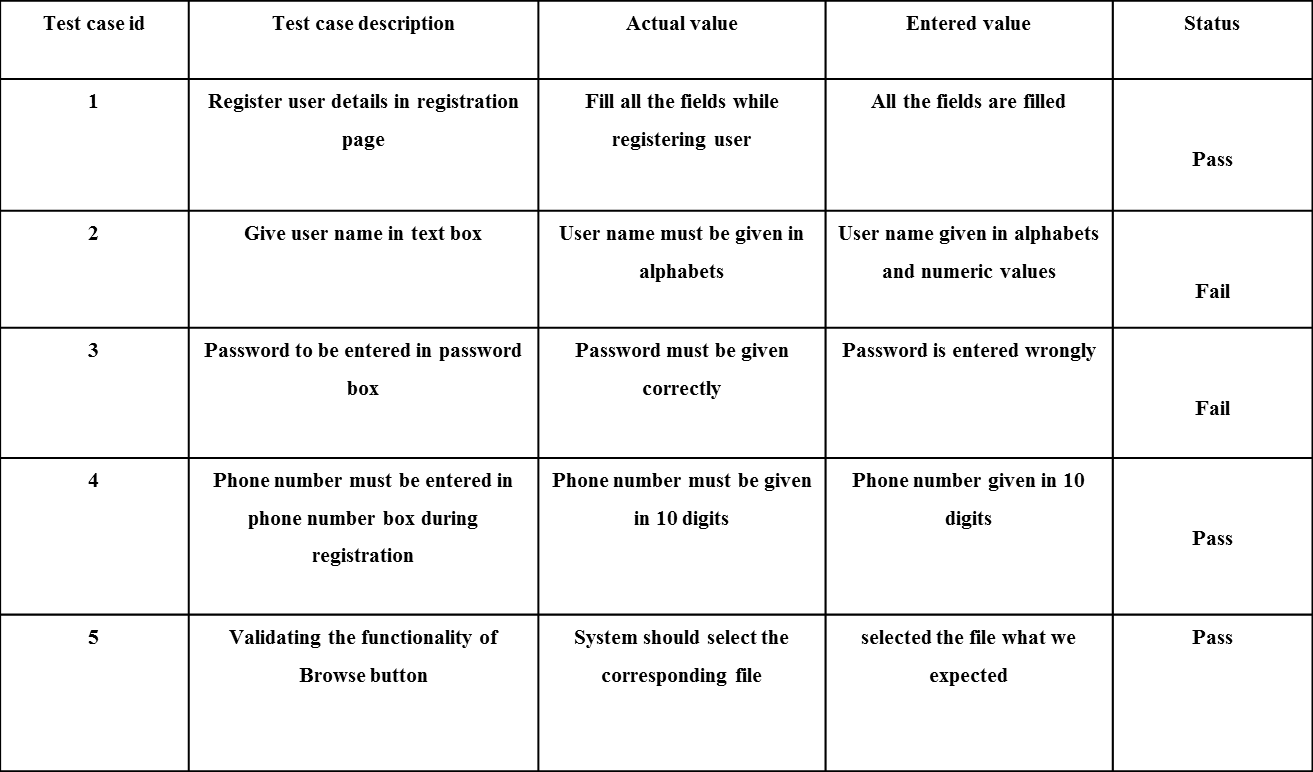
The task of the integration test is to design those components or s/w applications, e.g. modules in a software system or – one step up – software presentations at the company level

– interact without faults.

**Test Results:** All the test cases stated above passed effectively. No defects met.

**Acceptance Testing**

User Acceptance Testing is a serious phase of any project and needs important contribution by the end user. It also guarantees that the system encounters the functional requirements.

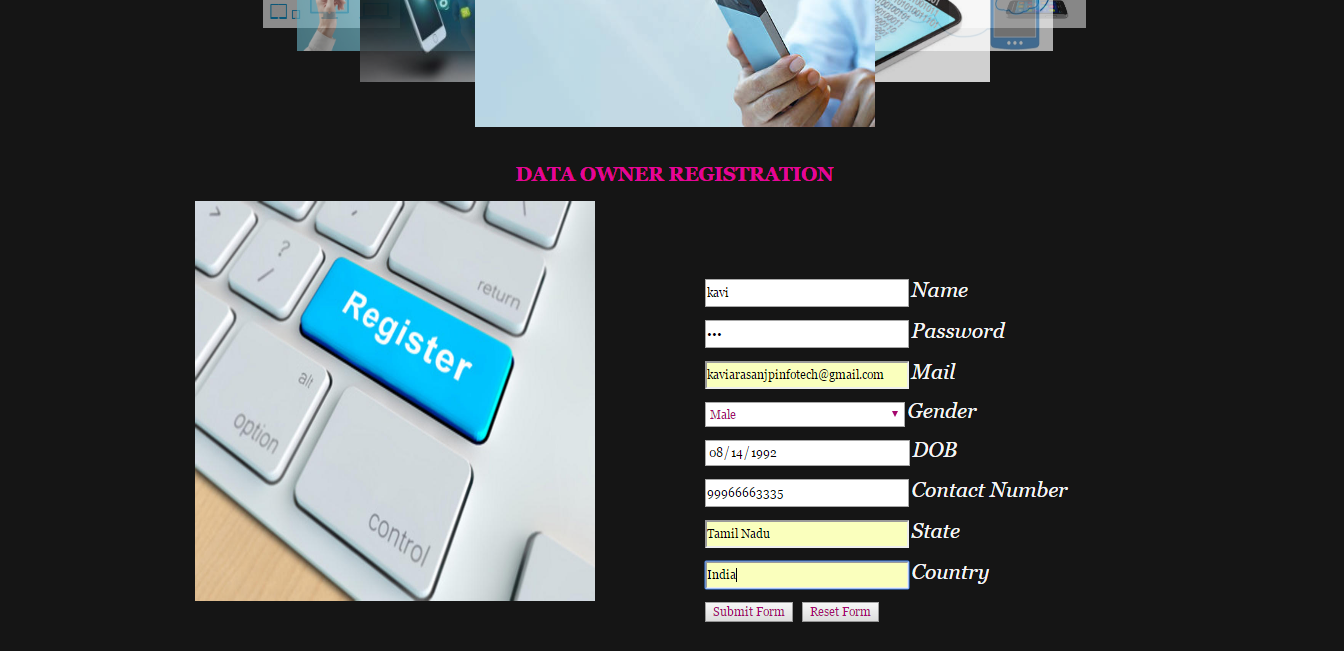


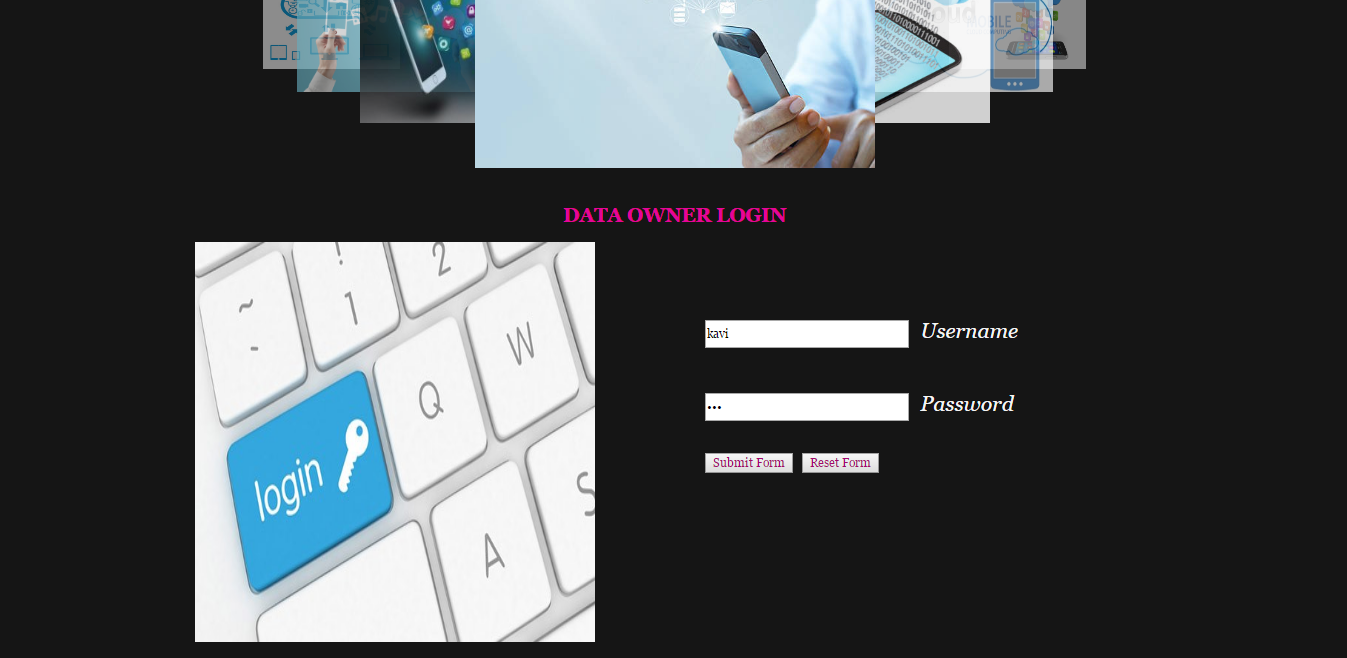
**Test Results:** All the test cases stated above passed effectively. No defects met.

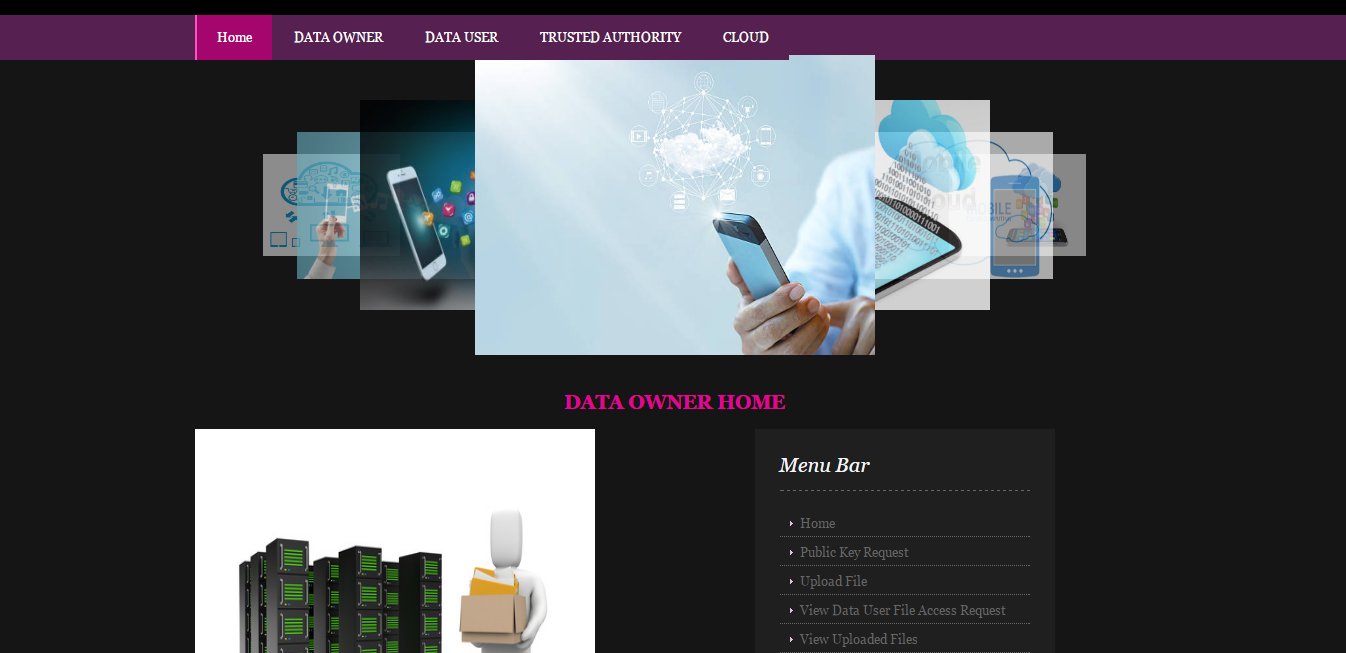
# CHAPTER 6

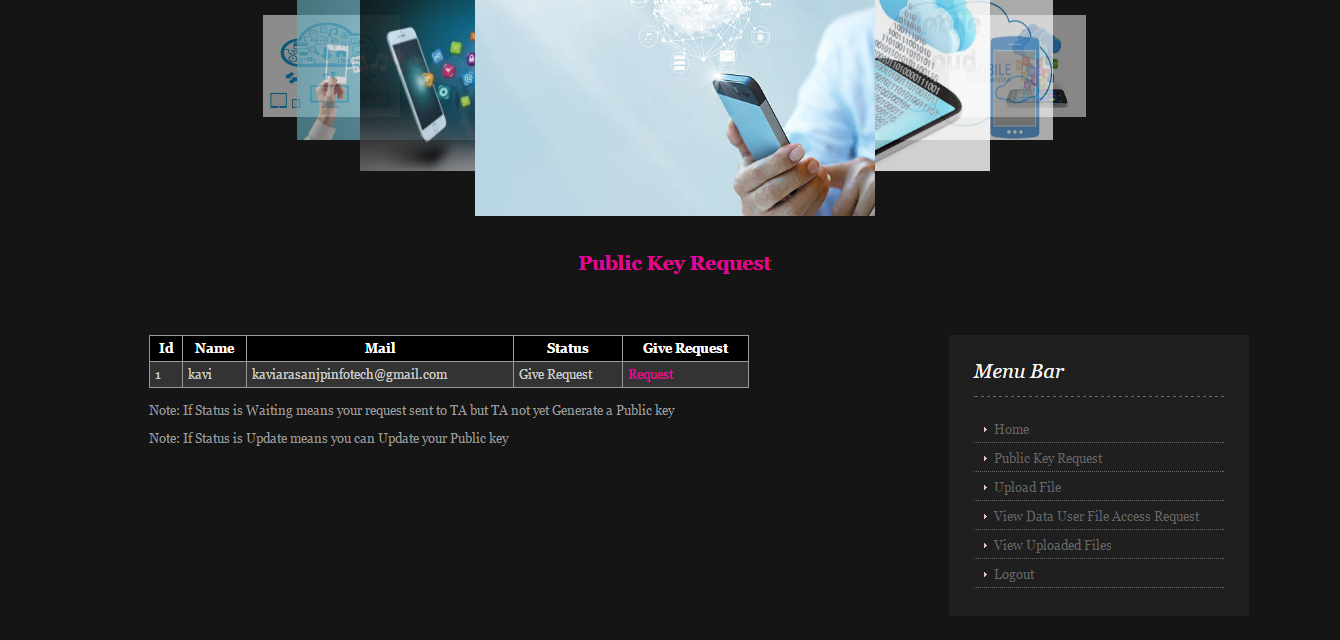
# RESULTS

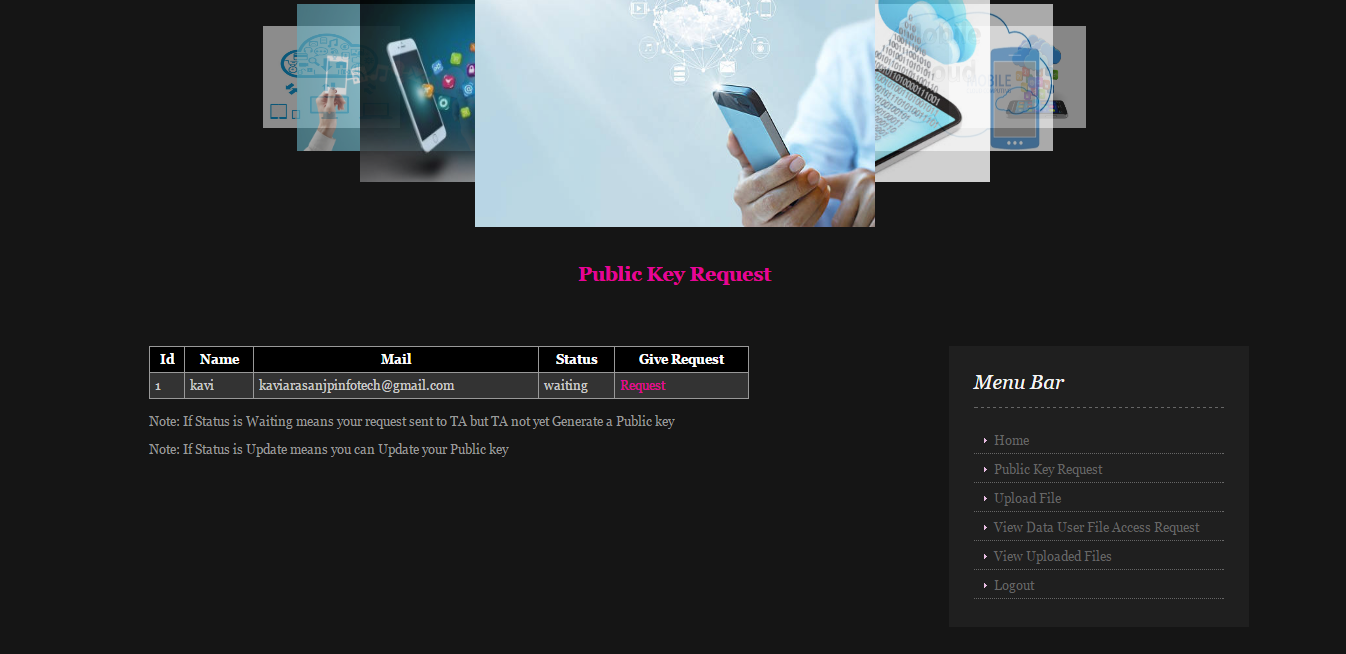


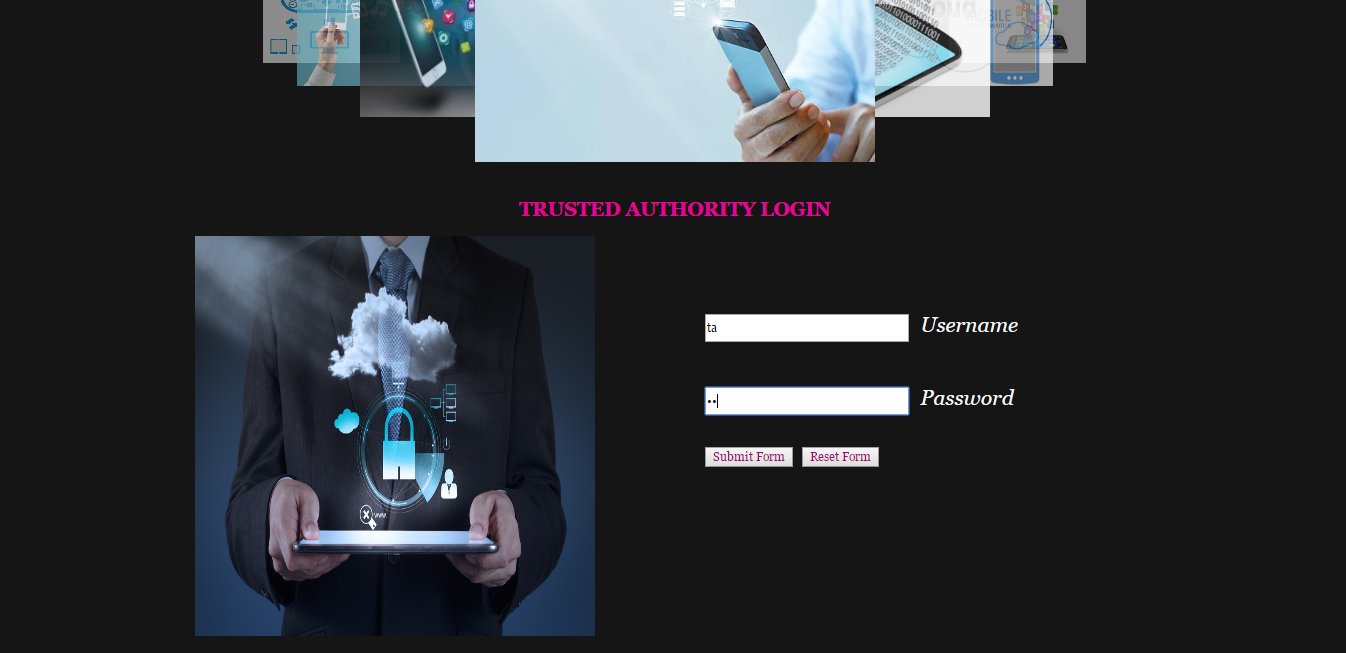


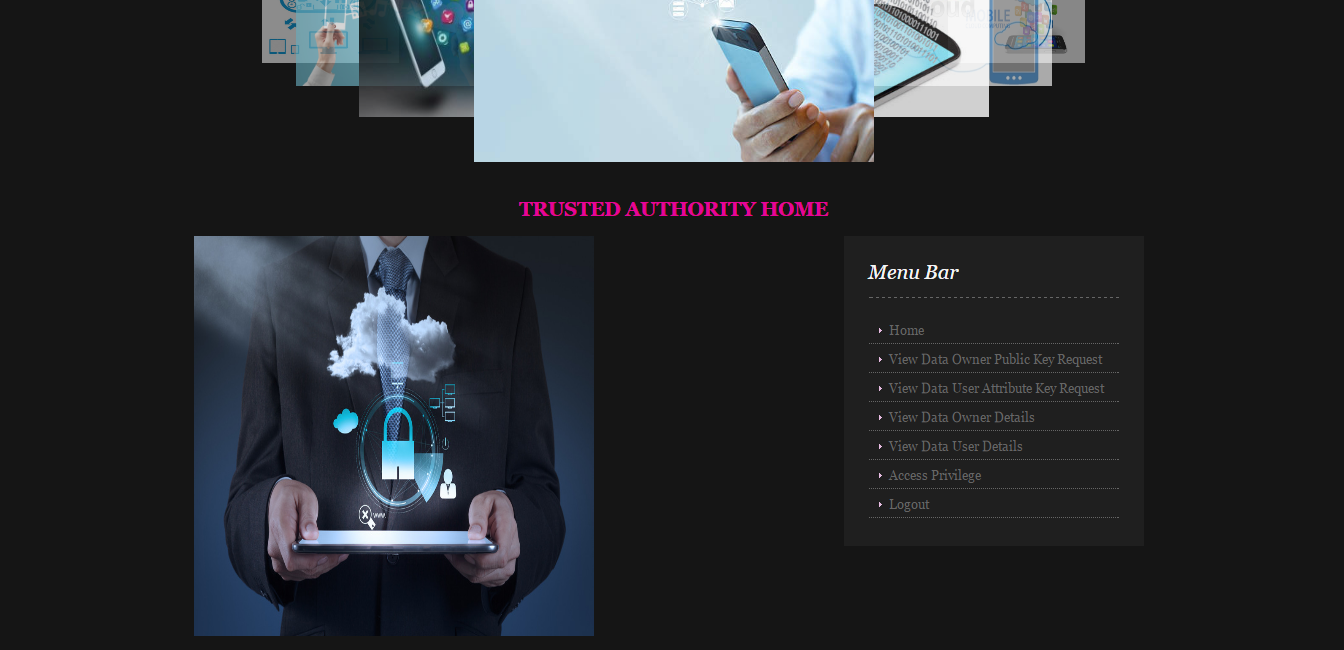


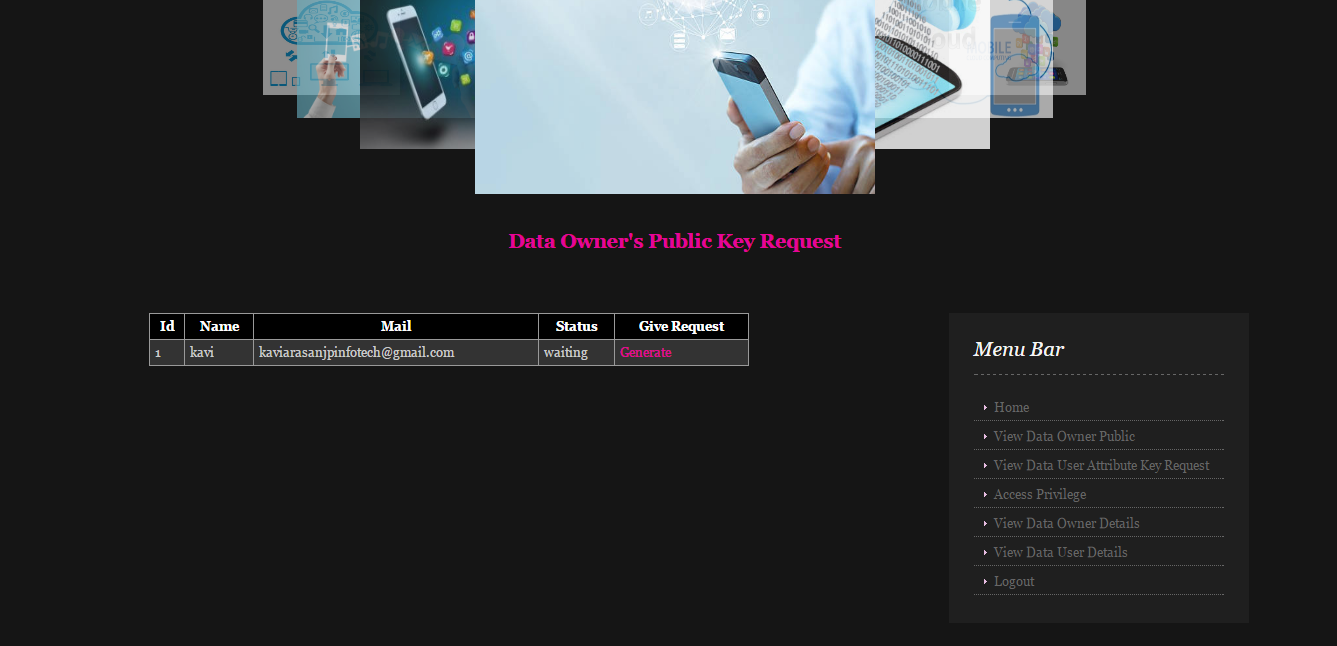


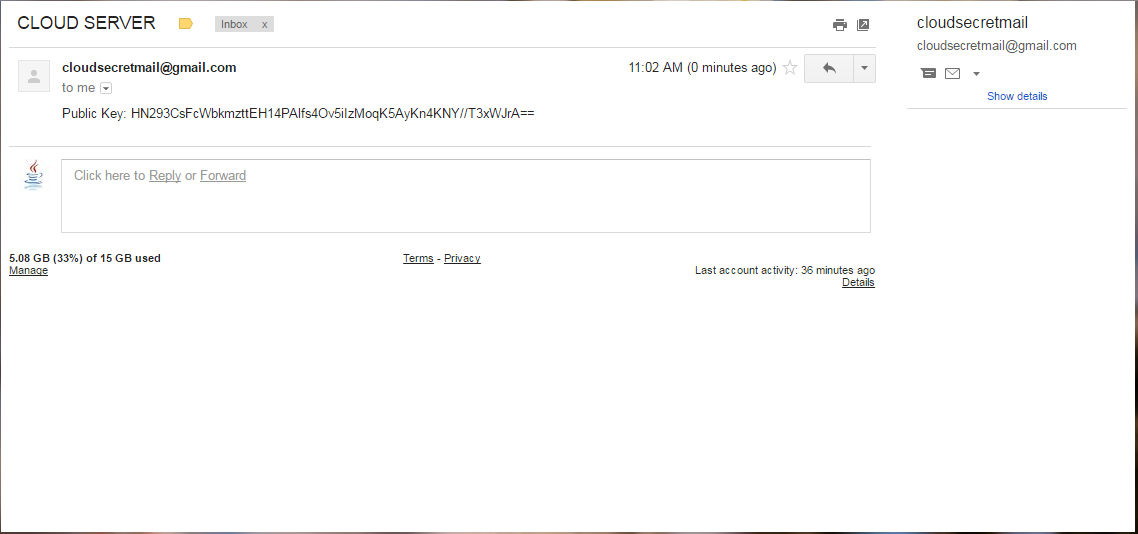












# CHAPTER 7

**CONCLUSION & FUTURE WORK**

**CONCLUSION**

In recent years, many studies on access control in cloud are based on attribute-based encryption algorithm (ABE). However, traditional ABE is not suitable for mobile cloud because it is computationally intensive and mobile devices only have limited resources. In this paper, we propose LDSS to address this issue. It introduces a novel LDSS-CP-ABE algorithm to migrate major computation overhead from mobile devices onto proxy servers, thus it can solve the secure data sharing problem in mobile cloud. The experimental results show that LDSS can ensure data privacy in mobile cloud and reduce the overhead on users’ side in mobile cloud. In the future work, we will design new approaches to ensure data integrity.

# Future Work

To further tap the potential of mobile cloud, we will also study how to do ciphertext retrieval over existing data sharing schemes.

# CHAPTER-8REFERENCES

[1] Gentry C, Halevi S. Implementing gentry’s fully-homomorphic encryption scheme. in: Advances in Cryptology–EUROCRYPT 2011. Berlin, Heidelberg: Springer press, pp. 129-148, 2011.

[2] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. in: Proceeding of IEEE Symposium on Foundations of Computer Science. California, USA: IEEE press, pp. 97-106, Oct. 2011.

[3] Qihua Wang, Hongxia Jin. "Data leakage mitigation for discertionary access control in collaboration clouds". the 16th ACM Symposium on Access Control Models and Technologies (SACMAT), pp.103-122, Jun. 2011.

[4] Adam Skillen and Mohammad Mannan. On Implementing Deniable Storage Encryption for Mobile Devices. the 20th Annual Network and Distributed System Security Symposium (NDSS), Feb. 2013.

[5] Wang W, Li Z, Owens R, et al. Secure and efficient access to outsourced data. in: Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, USA: ACM pp. 55-66, 2009.

[6] Maheshwari U, Vingralek R, Shapiro W. How to build a trusted database system on untrusted storage. in: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4. USENIX Association, pp. 10-12, 2000.

[7] Kan Yang, Xiaohua Jia, Kui Ren: Attribute-based fine-grained access control with efficient revocation in cloud storage systems. ASIACCS 2013, pp. 523-528, 2013.

[8] Crampton J, Martin K, Wild P. On key assignment for hierarchical access control. in: Computer Security Foundations Workshop. IEEE press, pp. 14-111, 2006.

[9] Shi E, Bethencourt J, Chan T H H, et al. Multi-dimensional range query over encrypted data. in: Proceedings of Symposium on Security and Privacy (SP), IEEE press, 2007. 350- 364

[10] Cong Wang, Kui Ren, Shucheng Yu, and Karthik Mahendra Raje Urs. Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data. IEEE INFOCOM 2012, Orlando, Florida, March 25-30, 2012

[11] Yu S., Wang C., Ren K., Lou W. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. INFOCOM 2010, pp. 534-542, 2010

[12] Kan Yang, Xiaohua Jia, Kui Ren, Bo Zhang, Ruitao Xie: DACMACS: Effective Data Access Control for Multiauthority Cloud Storage Systems. IEEE Transactions on Information Forensics and Security, Vol. 8, No. 11, pp.1790-1801, 2013.

[13] Stehlé D, Steinfeld R. Faster fully homomorphic encryption. in: Proceedings of 16th International Conference on the Theory and Application of Cryptology and Information Security. Singapore: Springer press, pp.377-394, 2010.

[14] Junzuo Lai, Robert H. Deng ,Yingjiu Li ,et al. Fully secure keypolicy attribute-based encryption with constant-size ciphertexts and fast decryption. In: Proceedings of the 9th ACM symposium on Information, Computer and Communications Security (ASIACCS), pp. 239-248, Jun. 2014.

[15] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute based encryption in: Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP). Washington, USA: IEEE Computer Society, pp. 321-334, 2007.

[16] Liang Xiaohui, Cao Zhenfu, Lin Huang, et al. Attribute based proxy re-encryption with delegating capabilities. in: Proceedings of the 4th International Symposium on Information, Computer and Communications Security. New

York, NY, USA: ACM press, pp. 276-286, 2009.

[17] Pirretti M, Traynor P, McDaniel P, et al. Secure atrribute-based systems. in: Proceedings of the 13th ACM Conference on Computer and Communications Security. New York, USA: ACM press, pp. 99-112, 2006.

[18] Yu S., Wang C., Ren K., et al. Attribute based data sharing with attribute revocation. in: Proceedings of the 5th International Symposium on Information, Computer and Communications Security (ASIACCS), New York, USA: ACM press pp. 261-270, 2010.

[19] Sandhu R S, Coyne E J, Feinstein H L, et al. Role-based access control models. Computer, 29(2): 38-47, 1996.

[20] Tian X X, Wang X L, Zhou A Y. DSP RE-Encryption: A flexible mechanism for access control enforcement management in DaaS. in: Proceedings of IEEE International Conference on Cloud Computing. IEEE press, pp.25-32, 2009

[21] Di Vimercati S D C, Foresti S, Jajodia S, et al. Over-encryption: management of access control evolution on outsourced data. in: Proceedings of the 33rd international conference on Very large data bases. Vienna, Austria: ACM, pp. 123-134, 2007.

[22] Kan Yang, Xiaohua Jia, Kui Ren, Ruitao Xie, Liusheng Huang: Enabling efficient access control with dynamic policy updating for big data in the cloud. INFOCOM 2014, pp.2013-2021, 2014.

[23] Jia W, Zhu H, Cao Z, et al. SDSM: a secure data service mechanism in mobile cloud computing. in: Proceedings of 30th IEEE International Conference on Computer Communications. Shanghai, China: IEEE, pp. 1060-1065, 2011.

[24] D. Huang, X. Zhang, M. Kang, and J. Luo. Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication. in: Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering. Nanjing, China: IEEE, pp. 90-98, 2010.

[25] Benjamin Livshits, Jaeyeon Jung. Automatic Mediation of Privacy-Sensitive Resource Access in Smartphone Applications. USENIX Security, pp.113-130, Aug. 2013.

[26] Zhou Z, Huang D. Efficient and secure data storage operations for mobile cloud computing. in: Proceedings of 8th International Conference on Network and Service Management (CNSM 2012), Las Vegas, USA: IEEE, pp. 37-45, 2012.

[27] P. K. Tysowski and M. A.Hasan. Hybrid attribute- and reencryption-based key management for secure and scalable mobile applications in clouds. IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 172-186, Nov. 2013.

[28] Boneh D, Franklin M. Identity-based encryption from the Weil pairing. in: Proceedings of the Advances in Cryptology. Berlin, Heidelberg: Springer-Verlag, pp. 213−229, 2001.

[29] Sahai A, Waters B. Fuzzy identity based encryption. in: Proceedings of the Advances in Cryptology. Aarhus, Denmark: Springer-Verlag, pp.457-473, 2005.

[30] Shamir A. How to share a secret. Communications of the ACM,1979, 22 (11): 612-613.