PROJECT REPORT ON

A THREE LAYER PRIVACY PRESERVING CLOUD STORAGE SCHEME USING FOG COMPUTING

The Project Report submitted in partial fulfillment of the requirement for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND SYSTEMS ENGINEERING

Submitted by

KUNDRAPU YASESWANI	**316114110024**
LAVETI TEJASWINI	**316114110025**
MANEPALLI SAI POOJITHA	**316114110027**
PONDURU JYOTHSNA VANI	**316114110039**

Under the esteemed guidance of

Ms. D. Lalitha Kumari

Assistant Professor

DEPARTMENT OF CS & SE, AUCEW

Department of Computer Science and Systems Engineering

Andhra University College of Engineering for Women

CERTIFICATE

ANDHRA UNIVERSITY COLLEGE OF ENGINEERING FOR WOMEN

VISAKHAPATNAM

This is to certify that this project entitled "**A THREE LAYER PRIVACY PRESERVING CLOUD STORAGE SCHEME ALONG WITH FOG COMPUTING**" is a bonafide workcarried out **KUNDRAPU YASESWANI (316114110024), LAVETI TEJASWINI (316114110025), MANEPALLI SAI POOJITHA (316114110027), PONDURU JYOTHSNA VANI (316114110039)**submitted in partial fulfillment of the requirements for the award of Degree of Bachelor Technology in Computer Science and Systems Engineering during the year December 2019 to April 2020.

Signature of Project Guide	Signature of Head of Department
Ms. D.Lalitha Kumari	**Prof. B Prajna**
Assistant Professor	Head of the Department
Department of CS & SE	Department of CS & SE
AUCEW	AUCEW

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved guide **Ms. D. Lalitha Kumari,** Assistant professor, Department of Computer Science and Systems Engineering, Andhra University College of Engineering for Women for the valuable guidance and suggestions, keen interest and thorough encouragement extended throughout the period of project work.

We express our deep sense of gratitude to our beloved Head of the Department **Prof. B Prajna**, Andhra University College of Engineering for Women for the valuable guidance and for permitting us to carry out this project. With immense pleasure, we record our deep sense of gratitude to our beloved principal **Prof. M Prameela Devi** for permitting us to carry out this project. We express our deep sense of gratitude to our beloved parents and friends for supporting us throughout the project.

We consider ourselves lucky enough to get such a project. This project would add as an asset to our academic profile. We express our thanks to the all those who contributed for the successful completion of our project work.

With gratitude,

KUNDRAPU YASESWANI(316114110024)

LAVETI TEJASWINI(316114110025)

MANEPALLI SAI POOJITHA (316114110027)

PONDURU JYOTHSNA VANI(316114110039)

DECLARATION

We hereby, declare that project entitled “**A THREE LAYER PRIVACY PRESERVING CLOUD STORAGE SCHEME ALONG WITH FOG COMPUTING**”is an authenticate record of our own work carried out at **ANDHRA UNIVERSITY COLLEGE OF ENGINEERING FOR WOMEN, VISAKHAPATNAM** as requirements of project term for the award of Degree of **Bachelor of Technology(COMPUTER SCIENCE AND SYSTEMS ENGINEERING)**.

We also here by , declare that this project is the result of our effort and that it has not been submitted to any other university for the award of any Degree.

KUNDRAPU.YASESWANI(316114110024)

LAVETI TEJASWINI(316114110025)

MANEPALLI SAI POOJITHA (316114110027)

PONDURU JYOTHSNA VANI(316114110039)

INDEX

Contents **Pg.No.**

ABSTRACT

Recent years witness the development of cloud computing technology. With the explosive growth of unstructured data, cloud storage technology gets more attention and better development. However, in current storage schema, user’s data is totally stored in cloud servers. In other words, users lose their right of control on data and face privacy leakage risk. Traditional privacy protection schemes are usually based on encryption technology, but these kinds of methods cannot effectively resist attack from the inside of cloud server. In order to solve this problem, we propose a three-layer storage framework based on fog computing. The proposed framework can both take full advantage of cloud storage and protect the privacy of data. Besides, Hash-Solomon code algorithm is designed to divide data into different parts. Then, we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme.

LIST OF FIGURES

1.INTRODUCTION

1.1 Existing System

Since the 21stcentury, computer technology has developed rapidly. Cloud computing, an emerging technology, was first proposed in SES 2006 (Search Engine Strategies 2006) by San Jose and defined by NIST (National Institute of Standardsand Technology). Since it was proposed, cloud computing has attracted great attention from different sectors of society. Cloud computing has gradually matured through so many people's efforts. Then there are some cloud-based technologies deriving from cloud computing. Cloud storage is an important part of them. With the rapid development of network bandwidth, the volume of user's data is rising geometrically. User's requirement cannot be satisfied by the capacity of local machine any more. Therefore, people try to find new methods to store their data. Pursuing more powerful storage capacity, a growing number of users select cloud storage. Storing data on a public cloud server is a trend in the future and the cloud storage technology will become wide spread in a few years. Cloud storage is a cloud computing system which provides data storage and management service. With a cluster of applications, network technology and distributed file system technology, cloud storage makes a large number of different storage devices work together co-ordinately. Now-a-days there are a lot of companies providing a variety of cloud storage services, such as Drop box, Google Drive, iCloud, Baidu Cloud, etc. These companies provide large capacity of storage and various services related to other popular applications,which in turn leads to their success in attracting humorous subscribers.

1.2 Proposed System

However, cloud storage service still exists a lot of security problems. The privacy problem is particularly significant among those security issues. In history, there were some famous cloud storage privacy leakage events. For example, Apples iCloud leakage event in 2014, numerous Hollywood actresses private photos stored in the clouds were stolen. This event caused an uproar, which was responsible for the users' anxiety about the privacy of their data stored in cloud server. As shown in Fig. 1, user uploads data to the cloud server directly. Subsequently, the Cloud Server Provider (CSP) will take place of user to manage the data. In consequence, users do not actually control the physical storage of their data, which results in the separation of ownership and management of data. The CSP can freely access and search the data stored in the cloud. Meanwhile the attackers can also attack the CSP server to obtain the user’s data. The above two cases both make users fell into the danger of information leakage and data loss. Traditional secure cloud storage solutions for the above problems are usually focusing on access restrictions or data encryption. These methods can actually eliminate most part of these problems. However, all of these solutions cannot solve the internal attack well, no matter how the algorithm improves. Therefore, we propose a TLS scheme based on fog computing model and design a Hash-Solomon code based on Reed-Solomon code. Fog computing is an extended computing model based on cloud computing which is composed of a lot of fog nodes. These nodes have a certain storage capacity and processing capability. In our scheme, we split user’s data into three parts and separately save them in the cloud server, the fog server and the user’s local machine. Besides, depending on the property of the Hash-Solomon code, the scheme can ensure the original data cannot be recovered by partial data. On another hand, using Hash-Solomon code will produce a portion of redundant data blocks which will be used in decoding procedure. Increasing the number of redundant blocks can increase the reliability of the storage, but it also results in additional data storage. By reasonable allocation of the data, our scheme can really protect the privacy of user’ data. The Hash-Solomon code needs complex calculation, which can be assisted with the Computational Intelligence (CI). Paradigms of CI have been successfully used in recent years to address various challenges, for example, the problems in Wireless sensor networks (WSNs) field. CI provides adaptive mechanisms that exhibit intelligent behaviour in complex and dynamic environments like WSNs. Thus, we take advantage of CI to do some calculating works in the fog layer. Compared with traditional methods, our scheme can provide a higher privacy protection from interior, especially from the CSPs.

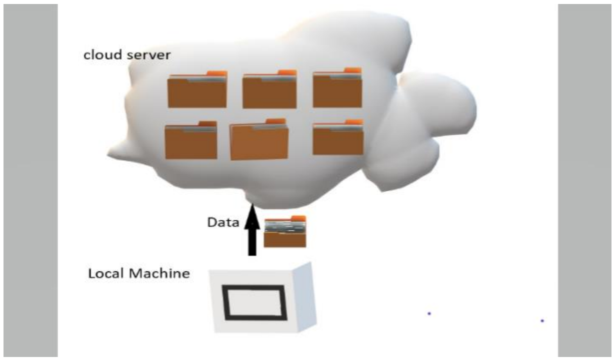


Fig1: Storage process in cloud server provider(CSP)

2. SYSTEM STUDY

2.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3. REQUIREMENTS SPECIFICATION

A software requirements specification is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use case that describes user interactions that the software must provide to the user for perfect interaction.

3.1 Minimum Hardware Requirements:

Hardware Requirements are the most common set of requirements defined by any operating system or software application.

System: Pentium Dual Core

Hard Disk: 120 GB

Monitor: 15 inches LED

Video graphics adaptor **:**16 bit VGA

Input Devices: Keyboard, Mouse

RAM: 4 GB

3.2 Minimum Software Requirements:

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on the computer to provide optimal functioning of an application.

Operating System: windows XP

Coding Language **:** Java/J2EE (JSP,Servlet)

Front End **:** J2EE

Back End **:** MySQL

3.3 FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS:

Functional Requirements

Functional requirements show the operation and activities the system must be able to perform. The functional requirements of Timely Call Blocker is that the user;

i. should be able to maintain the privacy of the data.

ii. should be able to efficiently store the data in different servers.

iii. shall be able to accurately encode and decode the data being stored while updating and retrieving the data respectively.
iv. shall be able to retrieve the same data being stored after combining from the different servers.
v. should be able to update and download in sufficient times.

Non Functional Requirements

Non-functional requirements are constraints upon the system behaviour or quality attributes of a system. Consequently, the non-functional requirements of Timely Call Blocker system are that the system:

i. should be developed to be simple and efficient for the end users.
ii. should be easy for interpretation.
iii. shall be able to provide best security to the data being stored.
iv. should be easy to maintain and provide reliability.
v. shall be compatible to any hardware.

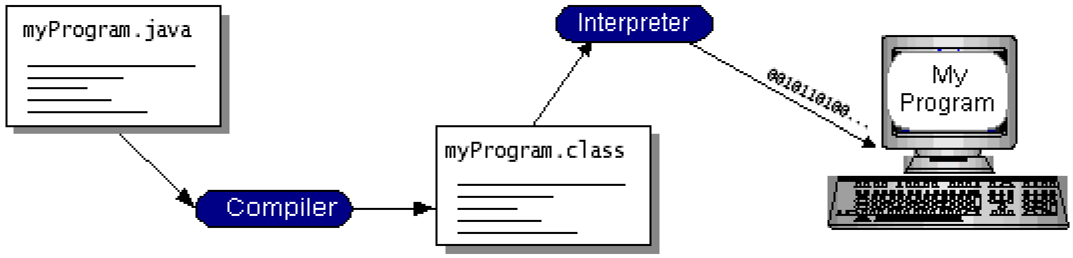
4. TECHNOLOGIES INVOLVED IN THIS PROJECT

JAVA TECHNOLOGY

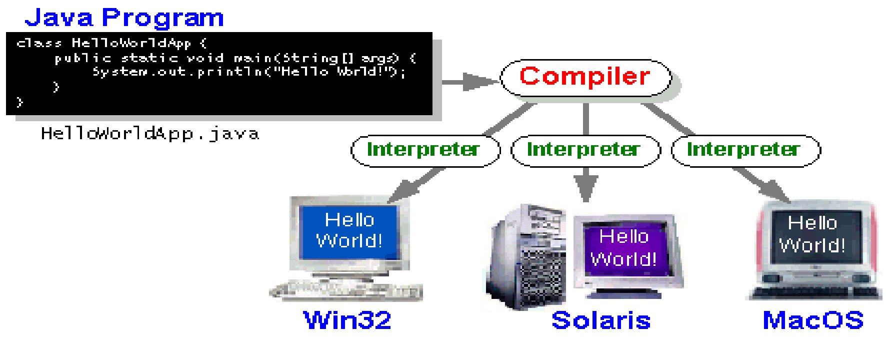
Java technology is both a programming language and a platform.The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- 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.

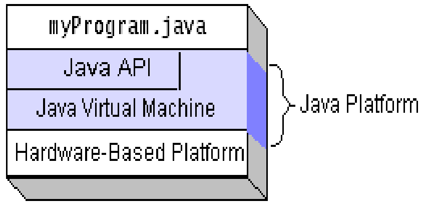


Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can 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.



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Java VM is the base for the Java platform and is ported onto various hardware-basedplatforms. 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 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.

The most common types of programs written in the Java programming language are *applets* and *applications*. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser. 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.

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.

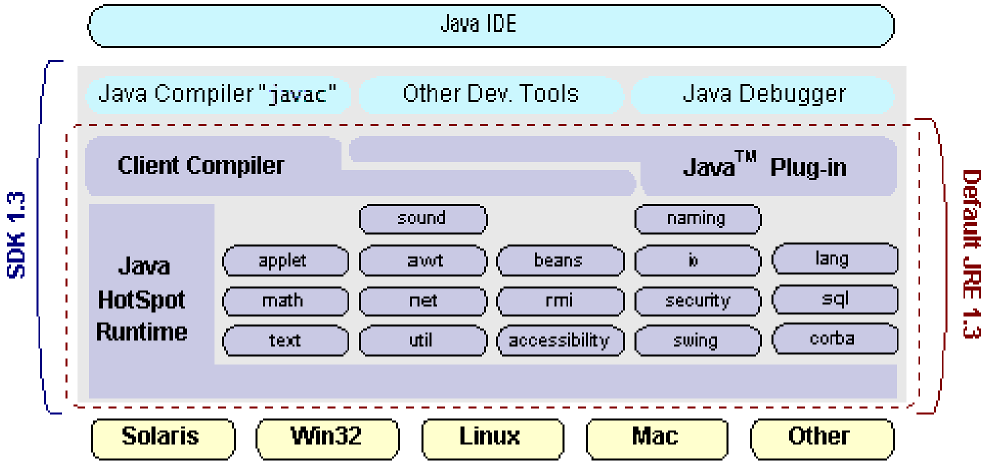
Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

Security: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

Software components: Known as JavaBeans, can plug into existing component architectures.

Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

Java Database Connectivity (JDBC^{TM}): Provides uniform access to a wide range of relational databases.



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, ODBC is used so 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.

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.

2. **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.

3. **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.

4. **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.

5. **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.

6. **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.

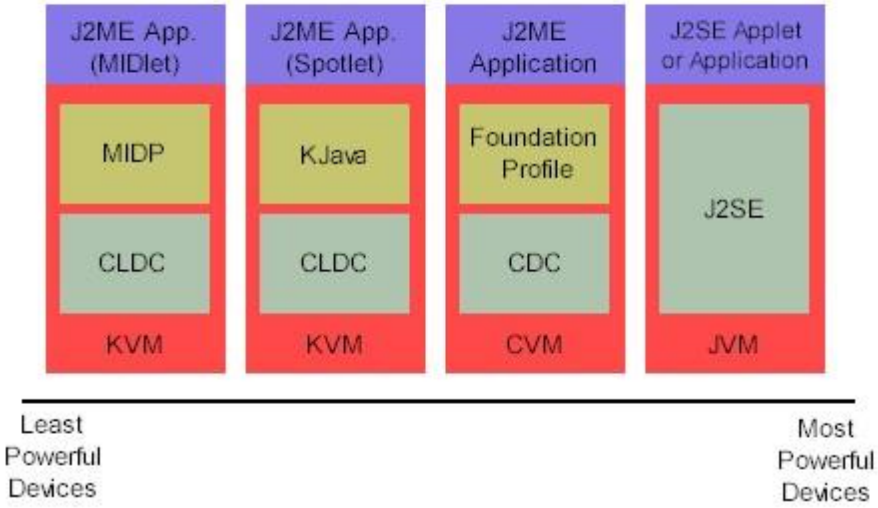
Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

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

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 Theprofile 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. Its size limitations make CLDC more interesting and challenging applications (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

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: K Java and MIDP. Both K Java 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: K Java

K Java is Sun's proprietary profile and contains the K Java API. The K Java profile is built on top of the CLDC configuration. The K Java virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. K Java contains a Sun-specific API that runs on the Palm OS. The K Java 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 K Java 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

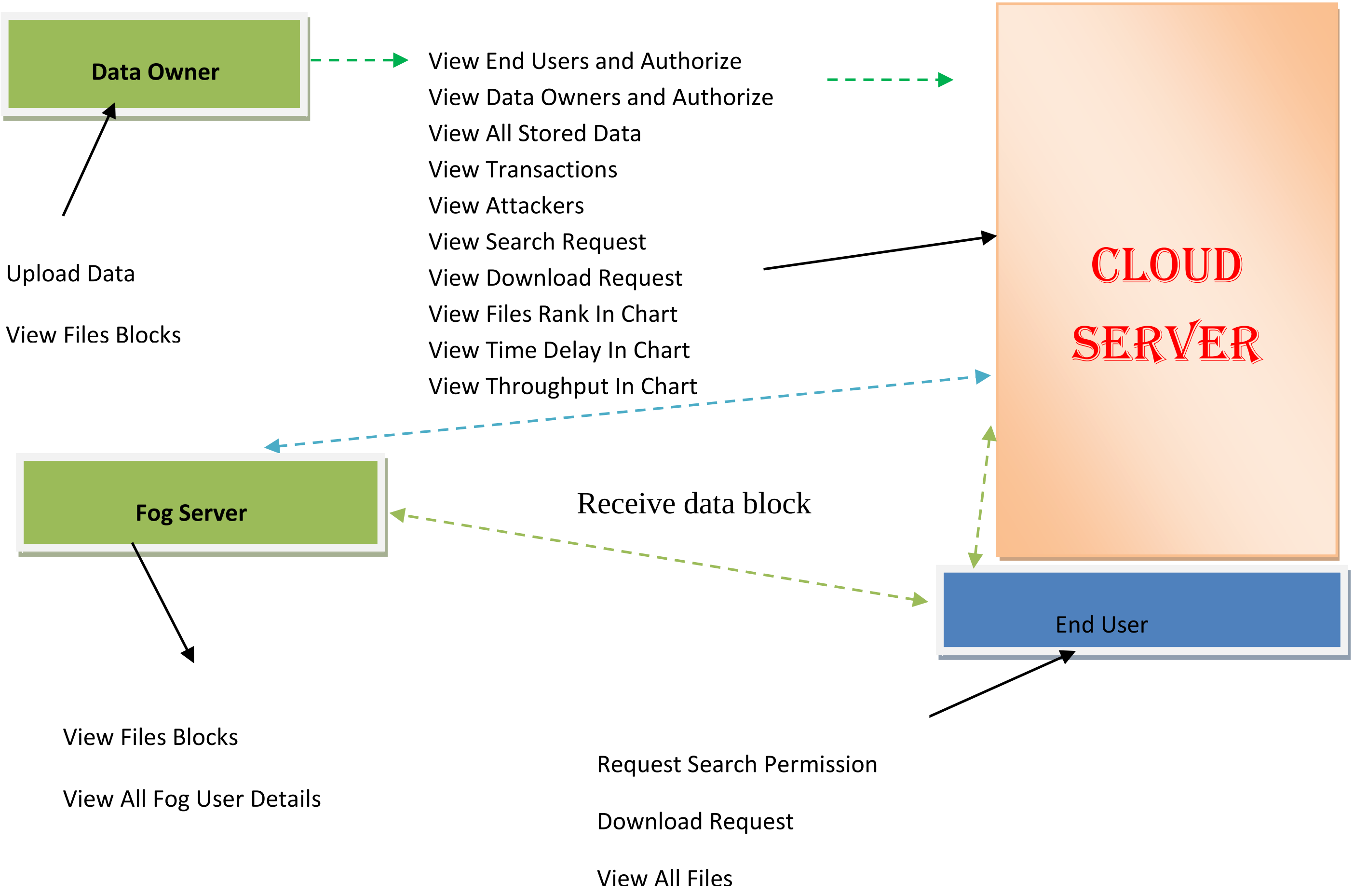
* javax.microedition.midlet

* javax.microedition.rms

5. SYSTEM DESIGN

System design is the transformation of an analysis model into a system design model. During system design, developers define the design goals of the project and decompose the system into smaller subsystems that can be realized by individual teams. The result of system design is a model that includes subsystem decomposition and a clear description of each of these strategies.

5.1. ARCHITECTURE



Here, this architecture states that there are 2 types of users and 2 different servers they are: data owner and end user, cloud server and fog server totally 4 modules . Data owner can upload data and can view the file blocks ,where as other user i.e, End users wants to access the data then he/she requests the cloud server to give permission to access the data. Cloud sever is responsible for all the users including data owner that can get access of data i.e search, read and write . Data owner and Cloud server can see his data blocks, search requests , download requests, file ranks, time delay and throughput .Fog server gives access data to cloud server when it gives access to the users and vies the file blocks and all fog user details. End user can request for search permission, download permission and can view all the files that are authorized. End user gets the data from the cloud server and fog server.

5.2. MODULES

There are 4 modules we used in this project they are : Data Owner ,End User ,Cloud server and Fog server.

- **Data Owner**

  In this module, he logs in by using his/her user name and password. After Login the owner Uploads Data, View Files Blocks.

- **End User**

  In this module, he logs in by using his/her user name and password. After Login the user will do some operations such as Request Search Permission, Download Request ,View All Files, Download File.

- **Fog Server**

  In this module, the Fog Server can do following operations such as View Files Blocks,View All Fog User Details and process the end user operations to send data block.

- **Cloud Server**

  The Cloud server as a server to provide data storage service and can also do the following operations such as View End Users and Authorize ,View Data Owners and Authorize, View All Stored Data, View Transactions ,View Attackers, View Search Request, View Download_Request,View Files Rank In Chart, View Time Delay In Chart, View Throughput In Chart.

5.3. UML DIAGRAMS

Unified Modeling Language (UML) as the name implies, is a modeling language. It may be used to visualize, specify, construct, and document the artifacts of a software system. It provides a set of notations to create a visual mode of the system. UML has been designed for a broad range of applications. Hence, it provides constructs for a broad range of systems and activities (e.g., distributed systems, analysis, system design, and deployment). System development focuses on three different models of the system:

• The functional model, represented in UML with use case diagrams, describes the functionality of the system from the user’s point of view. • The object model, represented in

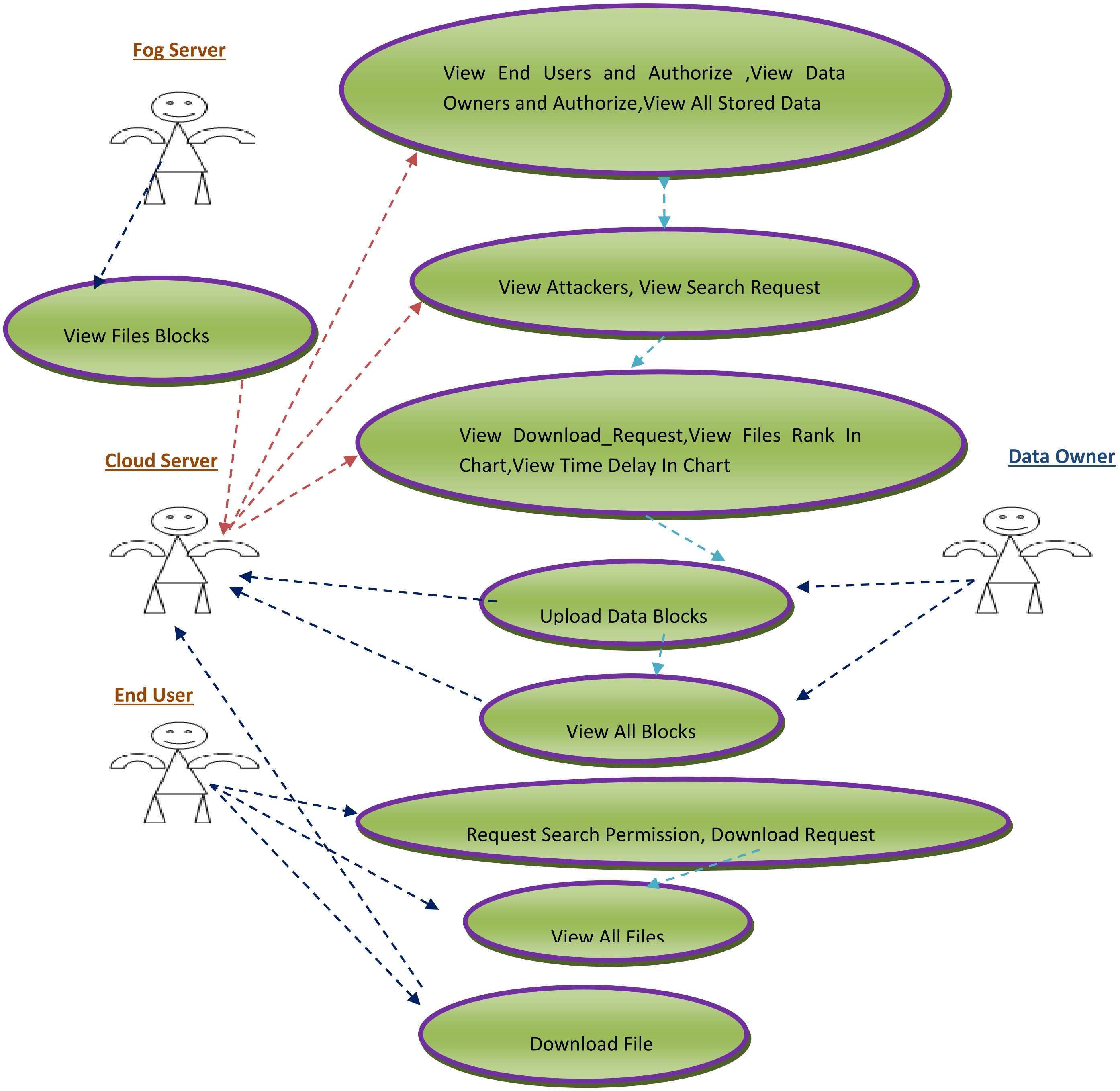
UML with class diagrams, describes the structure of the system in terms of objects, attributes, associations, and operations.

- The dynamic model, represented in UML with interaction diagrams, state machine diagrams, and activity diagrams, describes the internal behaviour of the system.

Interaction diagrams describe behavior as a sequence of messages exchanged among a set of objects, whereas state machine diagrams describe behavior in terms of states of an individual object and the possible transitions between states. Activity diagrams describe behavior in terms control and data flows.

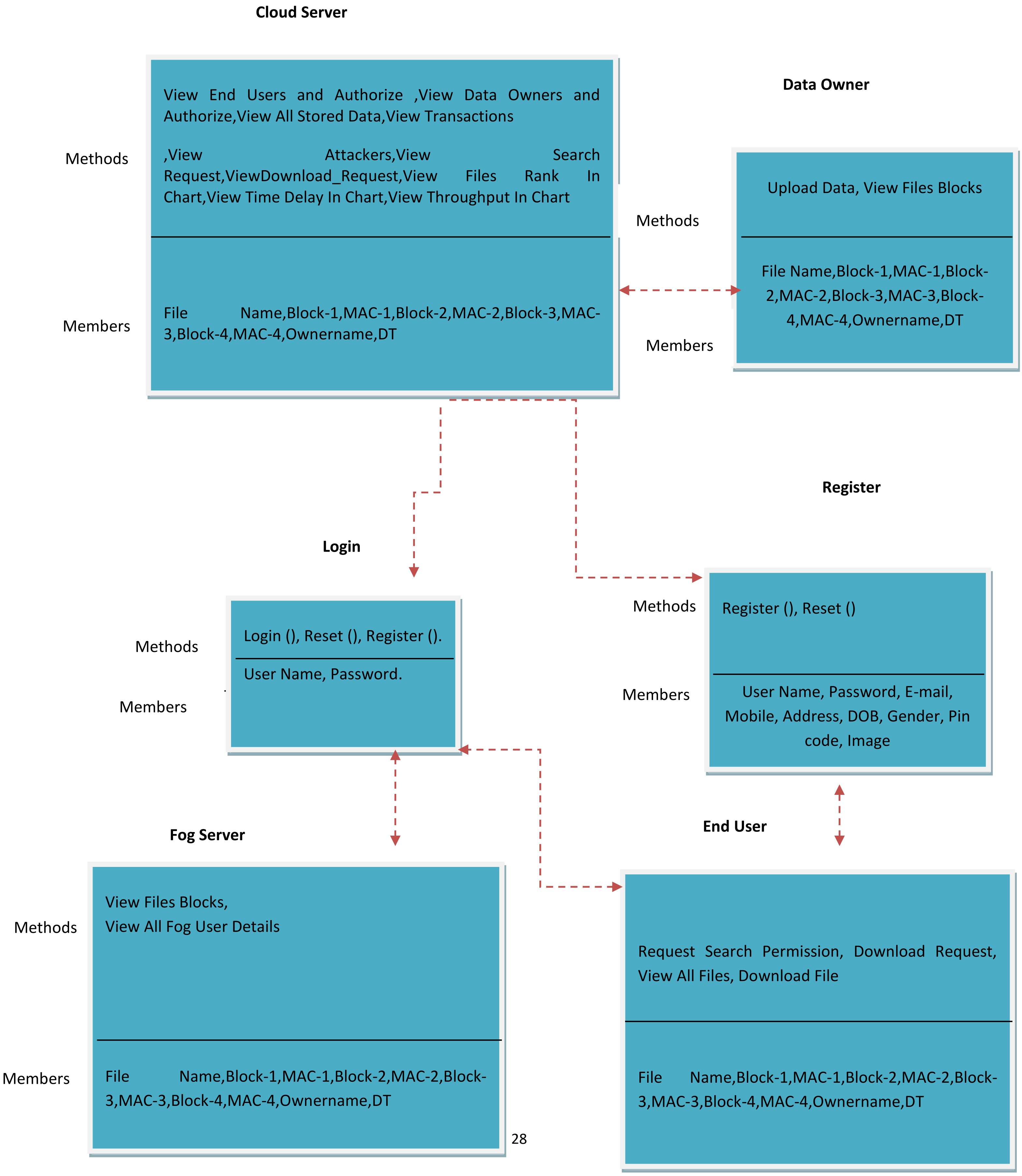
5.3.1. Use case Diagram

Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behaviour of the system from an external point of view. The identification of actors and use cases results in the definition of the boundary of the system. The actors are outside the boundary of the system, whereas the use cases are inside the boundary of the system


Fog Server
View End Users and Authorize ,View Data Owners and Authorize,View All Stored Data
View Files Blocks
View Attackers, View Search Request
View Download_Request,View Files Rank In Chart,View Time Delay In Chart
Cloud Server
Data Owner
Upload Data Blocks
View All Blocks
End User
Request Search Permission, Download Request
View All Files
Download File

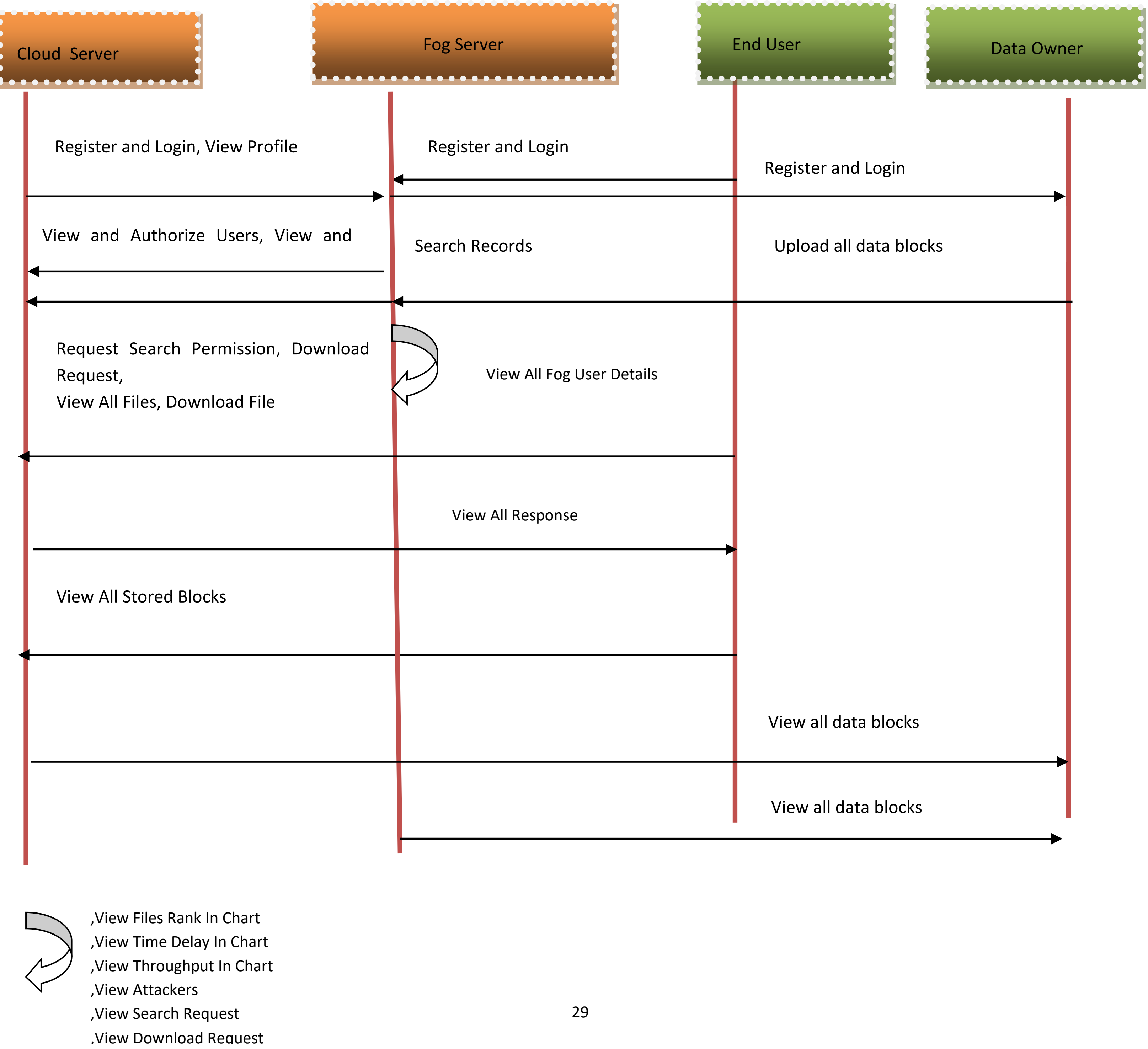
5.3.2. Class diagram

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.


Cloud Server
Methods
View End Users and Authorize ,View Data Owners and Authorize,View All Stored Data,View Transactions
,View Attackers,View Search Request,ViewDownload_Request,View Files Rank In Chart,View Time Delay In Chart,View Throughput In Chart
Members
File Name,Block-1,MAC-1,Block-2,MAC-2,Block-3,MAC-3,Block-4,MAC-4,Ownername,DT
Data Owner
Methods
Upload Data, View Files Blocks
Members
File Name,Block-1,MAC-1,Block-2,MAC-2,Block-3,MAC-3,Block-4,MAC-4,Ownername,DT
Register
Methods
Register (), Reset ()
Members
User Name, Password, E-mail, Mobile, Address, DOB, Gender, Pin code, Image
Login
Methods
Login (), Reset (), Register ().
Members
User Name, Password.
Fog Server
Methods
View Files Blocks,
View All Fog User Details
Members
File Name,Block-1,MAC-1,Block-2,MAC-2,Block-3,MAC-3,Block-4,MAC-4,Ownername,DT
End User
Request Search Permission, Download Request, View All Files, Download File
File Name,Block-1,MAC-1,Block-2,MAC-2,Block-3,MAC-3,Block-4,MAC-4,Ownername,DT

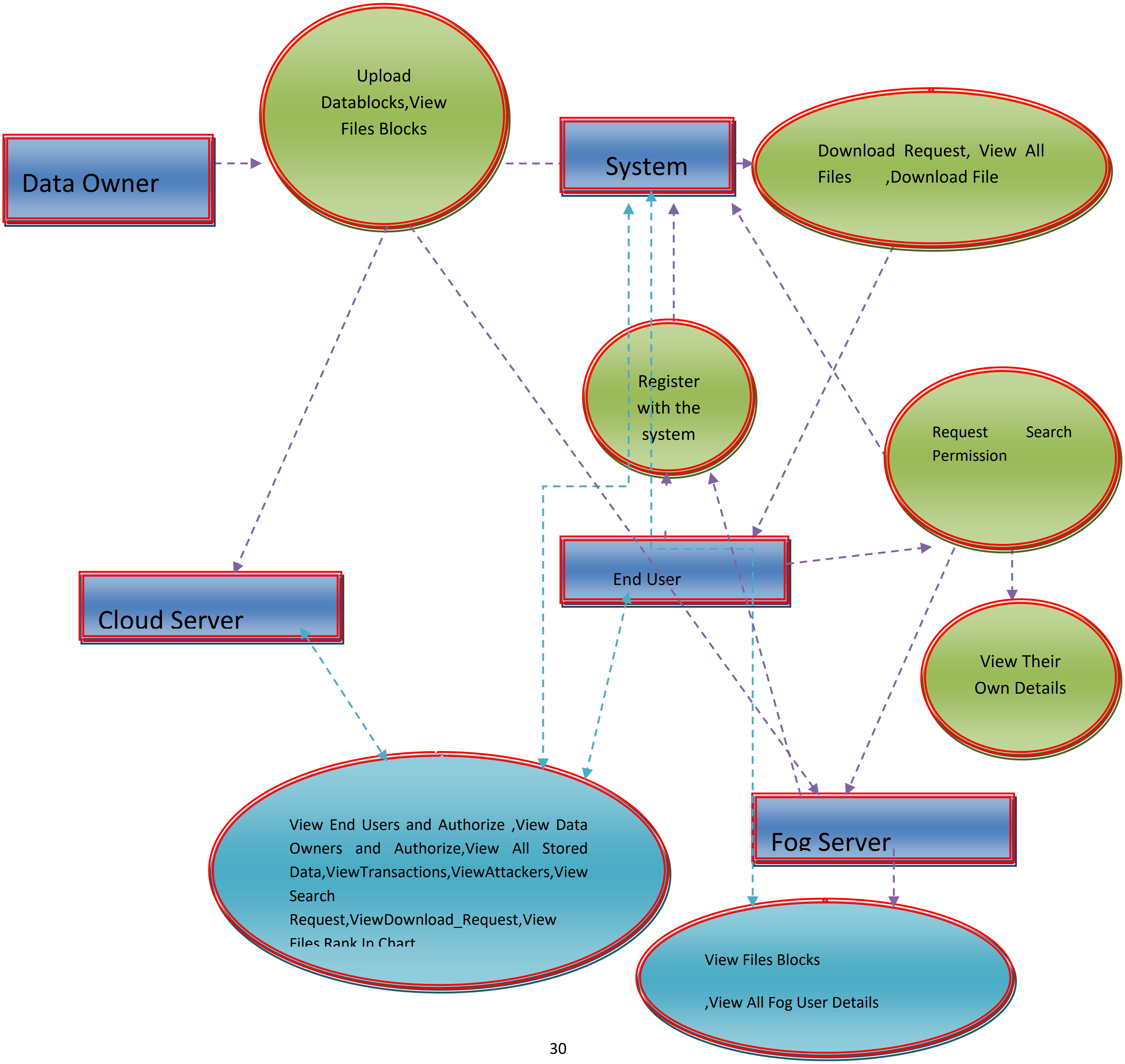
5.3.3. Sequence Diagram

Sequence diagrams represent the objects participating in the interaction horizontally and time vertically. An object interacts with another object by sending messages. Arguments may be passed along with a message and are bound to the parameters of the executing method in the receiving object.



5.3.4. Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system. The data-flow diagram provides the information about the outputs and the inputs of each entity and the process itself.


Data Owner
Upload Datablocks,View Files Blocks
System
Download Request, View All Files ,Download File
Register with the system
Request Search Permission
Cloud Server
End User
View Their Own Details
View End Users and Authorize ,View Data Owners and Authorize,View All Stored Data,ViewTransactions,ViewAttackers,View Search Request,ViewDownload_Request,View Files Rank In Chart
Fog Server
View Files Blocks
,View All Fog User Details

6.SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.

6.1. Sample Code

I. HTML Code

```
<html>
<head>
<title>Transaction Results</title>
<script type="text/javascript" src="sources/jscharts.js"></script>
</head>
<body>
<div id="graph">Loading graph...</div>
<script type="text/javascript">
var myData=new Array();
var colors=[];
<%      int i=0;
        String s1=null;
        while(rs.next())
        {
         s1=rs.getString(1);
        int s3=rs.getInt(2);
        %>
        myData["<%=i%>"]=["<%= s1%>",<%= s3%>];
        <%
        i++;}
        %>
        var myChart = new JSChart('graph', 'bar');
        myChart.setDataArray(myData);
        myChart.setBarColor('#42aBdB');
        myChart.setBarOpacity(0.8);
        myChart.setSize(700,320);
        myChart.setBarBorderColor('#D9EDF7');
        myChart.setBarValues(true);
        myChart.setTitleColor('#8C8383');
        myChart.setAxisColor('#777E89');
        myChart.setAxisValuesColor('#777E81');
        myChart.draw();
</script>
</body>
</html>
```

II. Java Code

```
<%@ page import="java.sql.*"%>
<%@ page import="java.util.*" %>
<%
        Connection connection = null;
        try {



                Class.forName("com.mysql.jdbc.Driver");
                connection                                                      =
DriverManager.getConnection("jdbc:mysql://localhost:3306/3layer","root","root");
        String sql="";

        }
        catch(Exception e)
        {
                System.out.println(e);
            }
    %>
```

6.2.Algorithm Implementation

Hash-Solomon Algorithm

It was introduced by Irving S. Reed and Gustave Solomon in 1960 .Hash-Solomon is the algorithm designed using Reed-Solomon. It is one of the oldest but still widely used and is a linear cyclic systematic non-binary block code. It is capable of correcting both burst errors and erasures. Wide range of applications in digital communications and storage.

After being encoded by Hash-Solomon code, the data will be divided into k parts and generates m redundant data. Hash-Solomon code has such property, in these k+m parts of data, if someone has at least k parts, he can recover the complete data. In other word, nobody can recover the complete data with less than k parts of data. According to this property of Hash-Solomon code, in our scheme, we let no more than k-1 parts of data be stored in higher server which has larger storage capacity and let the remainder be stored in the lower server. In this way, the stealer cannot recover the complete data even if one of the three layers' data was stolen. Thus we can ensure the privacy of user's data. Then we consider the value of k and m. Assuming that we want to save r% data on the fog server. In the Hash-Solomon code, we have definitions as follows:

Definition 1 Invalid Ratio: The ratio of the number of failure data blocks to the number of data blocks which will be used in encoding. In other words, the ratio of the number of data blocks stored in lower server to the number of data blocks stored in the upper server. For example, the ratio of the number of data blocks stored in the local machine to the number of data blocks stored in the fog server. In the same way, the ratio of the number of data blocks stored in the fog server to the number of data blocks stored in the cloud server.

Definition 2 Maximal Invalid Ratio: The maximal invalid ratio is the ratio of the number of invalid data to the number of all data blocks when the upper server can just recover the complete data by the data blocks stored in them. If there was one more invalid data blocks, the upper server can't recover the complete data anymore. In Hash-Solomon code, the Maximal Invalid Ratio can be expressed as $\frac{m}{k+m}$.For convenience, we just consider two layers situation. Assuming that there is x MB data which is prepared to save. After encoding, there will be $\frac{k+m}{m} * x$ data.Wepreparetoo save r% in the lower server. In order to avoid the upper server recovers the data, the value of k, m and r must satisfy the relationship:

$$\frac{m}{k+m} \leq \frac{k+m}{k} * r \qquad (1)$$

Through functional transformation, the relationship between k ,m and r can be expressed as formula (2). We can see that if the parameter r is determined, the parameter k can be expressed by m. So we can only consider the ratio and the number of data blocks when we use our scheme.

$$k = \frac{(m-2mr)+\sqrt{(2mr-m)^2-4m^2r^2}}{2r} \qquad (2)$$

The parameter k is the number of blocks after data being divided, the parameter m is the number of redundant data blocks and the parameter r is the storage ratio of different servers. Besides ,the fog server includes Computational Intelligence which can help the system with calculating the results of the values of k and m, because of the nodes in the fog server having its own computing power.

RSA Algorithm

The acronym RSA is the initial letters of the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman, who publicly described the algorithm in 1977. Clifford Cocks, an English mathematician working for the British intelligence agency Government Communications Headquarters (GCHQ), had developed an equivalent system in 1973, which was not declassified until 1997.

A user of RSA creates and then publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but only someone with knowledge of the prime numbers can decode the message. Breaking RSA encryption is known as the RSA problem. Whether it is as difficult as the factoring problem is an open question. There are no published methods to defeat the system if a large enough key is used.

But, RSA is a relatively slow algorithm, and because of this, it is less commonly used to directly encrypt user data. We cannot use this algorithm here in our project because we have to divide the text into three parts and then it should be encrypted and generates the signature keys. It can be not more efficient than Hash-Solomon algorithm we used.

Efficiency

We have discussed the relationship of k and m. we find that the ratio of k and m is decided once the storage ratio is decided. It means that if we set the storage ratio as 20%, k = 3m. Then we set k = 3, m = 1. In the real scenario, data blocks cannot be stored partly. In the above example, the lower server must store at least 2 blocks, so that the real storage ratio is 50%, which is far from the 20%. In order to reduce error, we can let k or m be a large number. However, with the increasing of k, the encoding and decoding efficiency will decrease, which will be proved by experiments. In this section, we will discuss how to balance the storage efficiency and the coding efficiency. At last, we propose a comprehensive index of the whole efficiency of the scheme. The storage efficiency is an important index for a storage related algorithm. A good system with high storage efficiency can save storage capacity as much as possible. Storage Industry Networking Association defines the storage efficiency as:

Storage Efficiency = DataSpace/(DataSpace+ CheckSpace) (3)

In our scheme, storage efficiency can be expressed as $\frac{k}{k+m}$.Then we can get the following formulas (4,5).We can see that the storage efficiency will increase with the increment to the ratio of k and m. From Fig. below we know that when the ratio of k and m increase, the number of data blocks (k) also increase, which influences the coding efficiency.

$$E_S = \frac{k}{k+m} = \frac{\frac{k}{m}}{\frac{k}{m}+1}. \qquad (4)$$

$$\lim_{\frac{k}{m}\to\infty} = \frac{\frac{k}{m}}{\frac{k}{m}+1} = 1 \qquad (5)$$

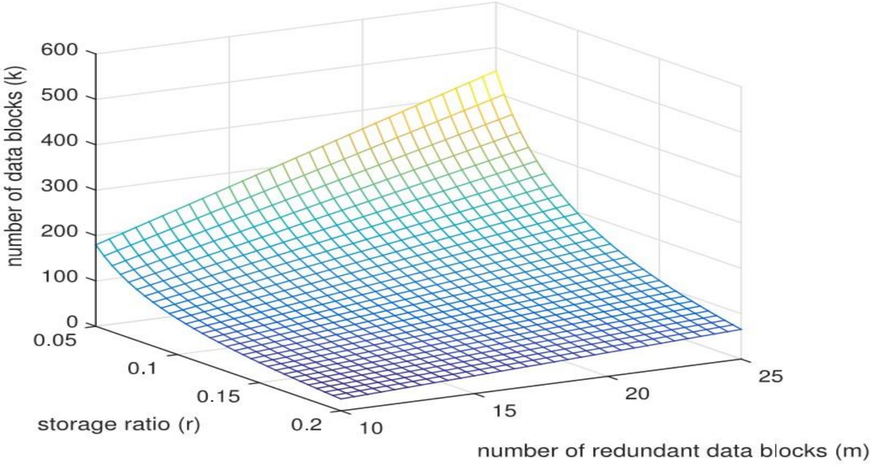
The coding efficiency is related to the operation on Galois field. We consider the influence of different bits of coding which is related to the ω of the Galois field. The relationship of ω, k and m satisfy the equation 2^ω >k+m. When ω increases, the consume of RAM increases. Therefore, we let the reciprocal of ω to present the coding efficiency and it can be expressed as

$$E_c = \frac{\ln(k+m)}{\ln 2} \qquad (6)$$

The change of storage efficiency and the coding efficiency when the number of k increases. The value of m is set to 2. Apparently, the tendency of storage efficiency is contrary to the tendency of coding. It means there must be a value of k which can achieve a best efficiency of the whole system. Therefore we should design a new index to take both of the storage efficiency and coding efficiency into consideration. The comprehensive efficiency of the scheme can be expressed as

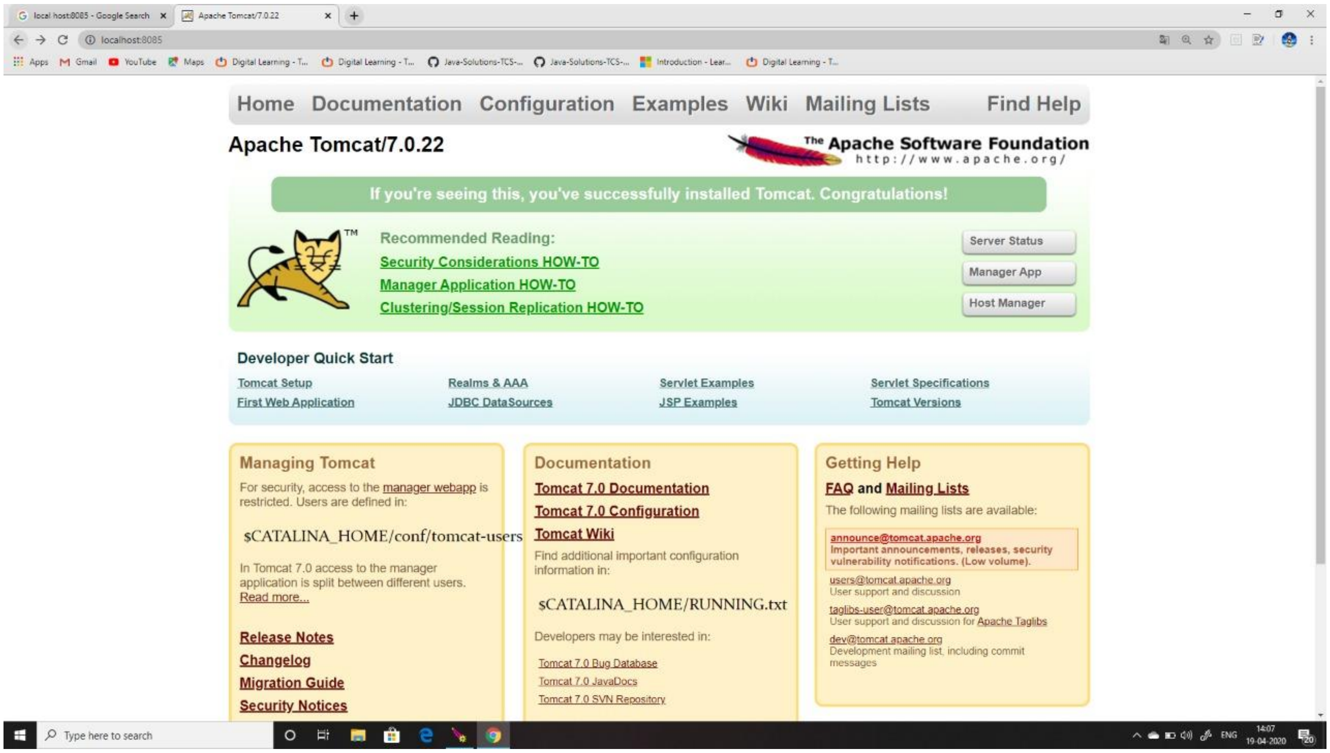
$$E_w = C_1 \frac{\ln(k+m)}{\ln 2} + C_2 \frac{k}{k+m} \qquad (7)$$

The parameter C1 and C2 are related to the storage ratio. For example, we set the value of m to 2, then the value of C1 is set as 0.6, the value of C2 is set as 0.4. The comprehensive efficiency of the scheme increases at first and decreases after it achieve the summit of the functional graph. We can consider the value of k which corresponds to the summit is the most suitable value for the whole efficiency of the scheme.



The efficiency of using Hash-Solomon algorithm is of 96% efficient. Where as with the usage of RSA, we can see the less efficiency about 70-75%. RSA algorithm consider the whole data in terms of bits it is not useful for dividing the data into data blocks. But the Hash-Solomon algorithm is special, it helps to divide the data into data blocks and also for encryption and decryption by generating the signature keys.

6.3. OUTPUT SCREENS



Apache Tomcat/7.0.22

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Servlet Examples
JSP Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

$CATALINA_HOME/conf/tomcat-users

In Tomcat 7.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

$CATALINA_HOME/RUNNING.txt

Developers may be interested in:

Tomcat 7.0 Bug Database
Tomcat 7.0 JavaDocs
Tomcat 7.0 SVN Repository

Getting Help

FAQ and Mailing Lists

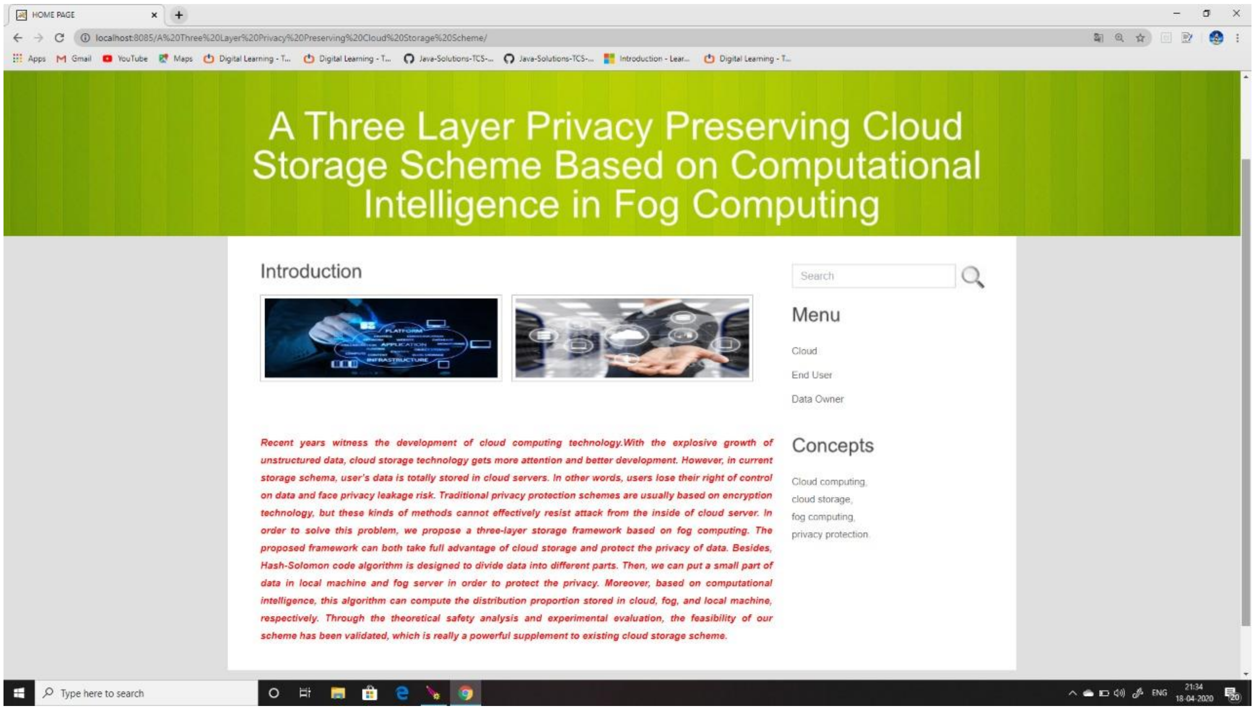
The following mailing lists are available:

announce@tomcat.apache.org
Important announcements, releases, security vulnerability notifications. (Low volume).

users@tomcat.apache.org
User support and discussion

taglibs-user@tomcat.apache.org
User support and discussion for Apache Taglibs

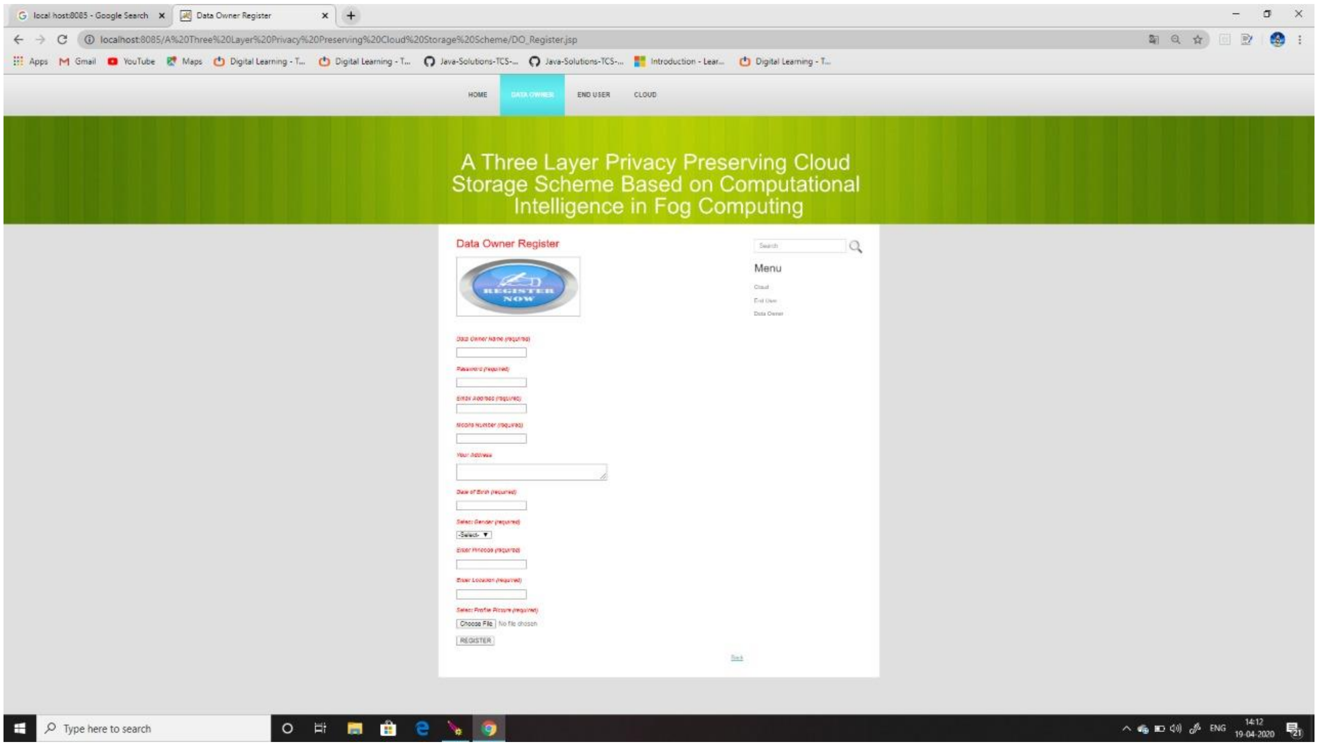
dev@tomcat.apache.org
Development mailing list, including commit messages



A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Introduction

Recent years witness the development of cloud computing technology.With the explosive growth of unstructured data, cloud storage technology gets more attention and better development. However, in current storage schema, user's data is totally stored in cloud servers. In other words, users lose their right of control on data and face privacy leakage risk. Traditional privacy protection schemes are usually based on encryption technology, but these kinds of methods cannot effectively resist attack from the inside of cloud server. In order to solve this problem, we propose a three-layer storage framework based on fog computing. The proposed framework can both take full advantage of cloud storage and protect the privacy of data. Besides, Hash-Solomon code algorithm is designed to divide data into different parts. Then, we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme.


localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/DO_Register.jsp
HOME
DATA OWNER
END USER
CLOUD
Search
Menu
Cloud
End User
Data Owner
REGISTER NOW
Type here to search

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Data Owner Register

Data Owner Name (required)

Password (required)

Email Address (required)

Mobile Number (required)

Your Address

Date of Birth (required)

Select Gender (required)

-Select- ▼

Enter Pincode (required)

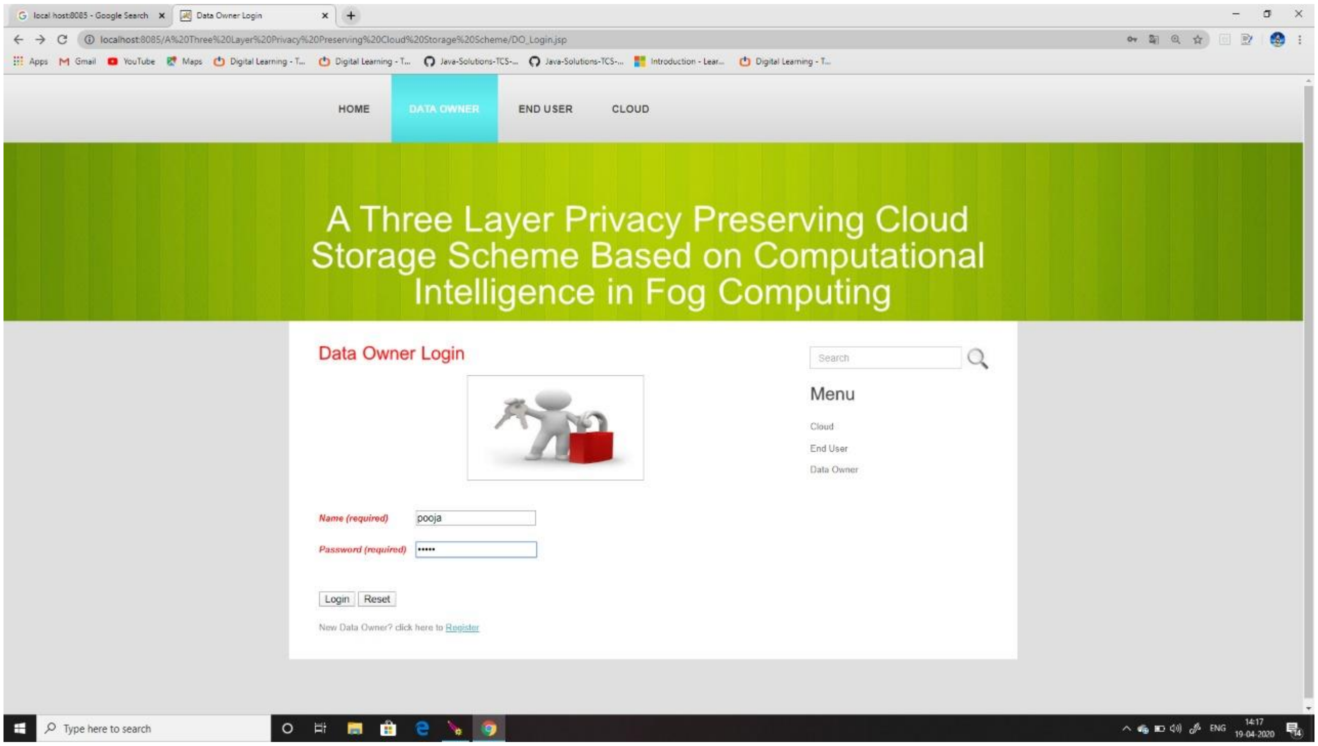
Enter Location (required)

Select Profile Picture (required)

Choose File No file chosen

REGISTER

Back


localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/DO_Login.jsp
HOME
DATA OWNER
END USER
CLOUD
Search
Menu
Cloud
End User
Data Owner
Type here to search

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

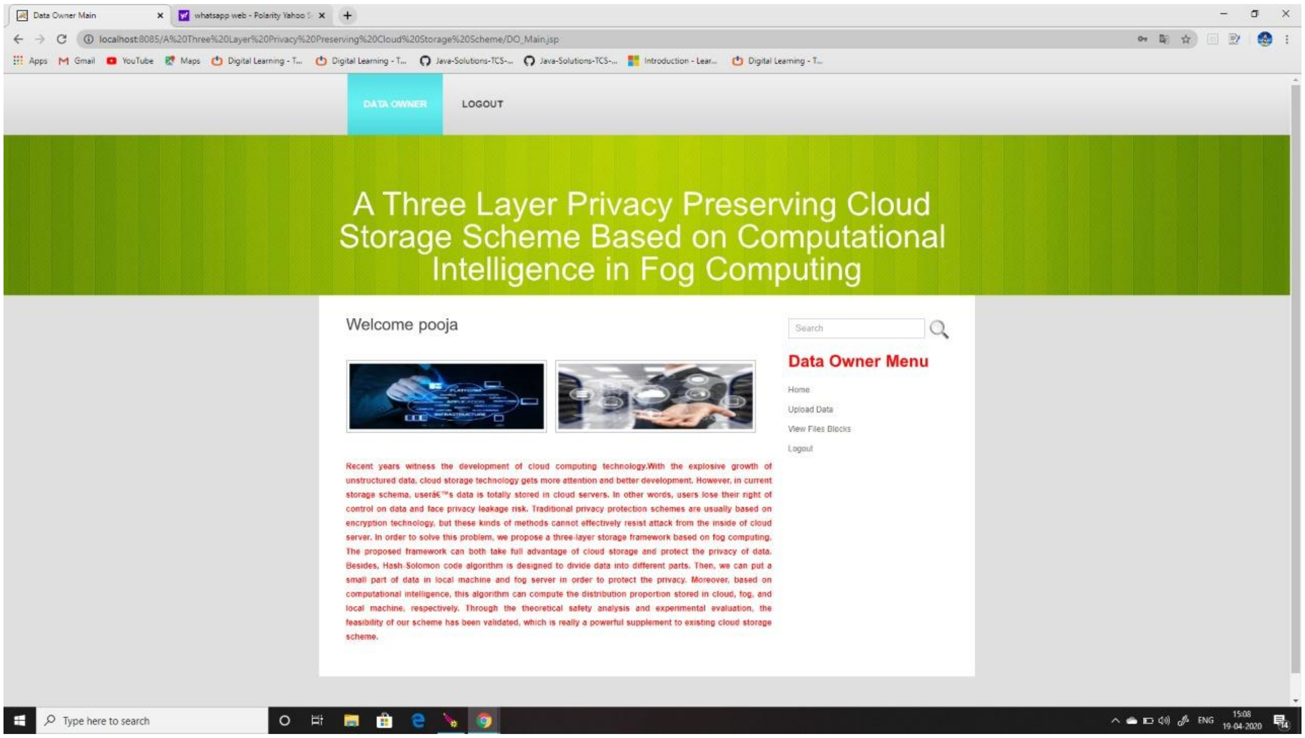
Data Owner Login

Name (required) pooja

Password (required) •••••

Login Reset

New Data Owner? click here to Register


Data Owner Main
whatsapp web - Polarity Yahoo S
localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/DO_Main.jsp
DATA OWNER
LOGOUT
Search
Type here to search
15:08
19-04-2020

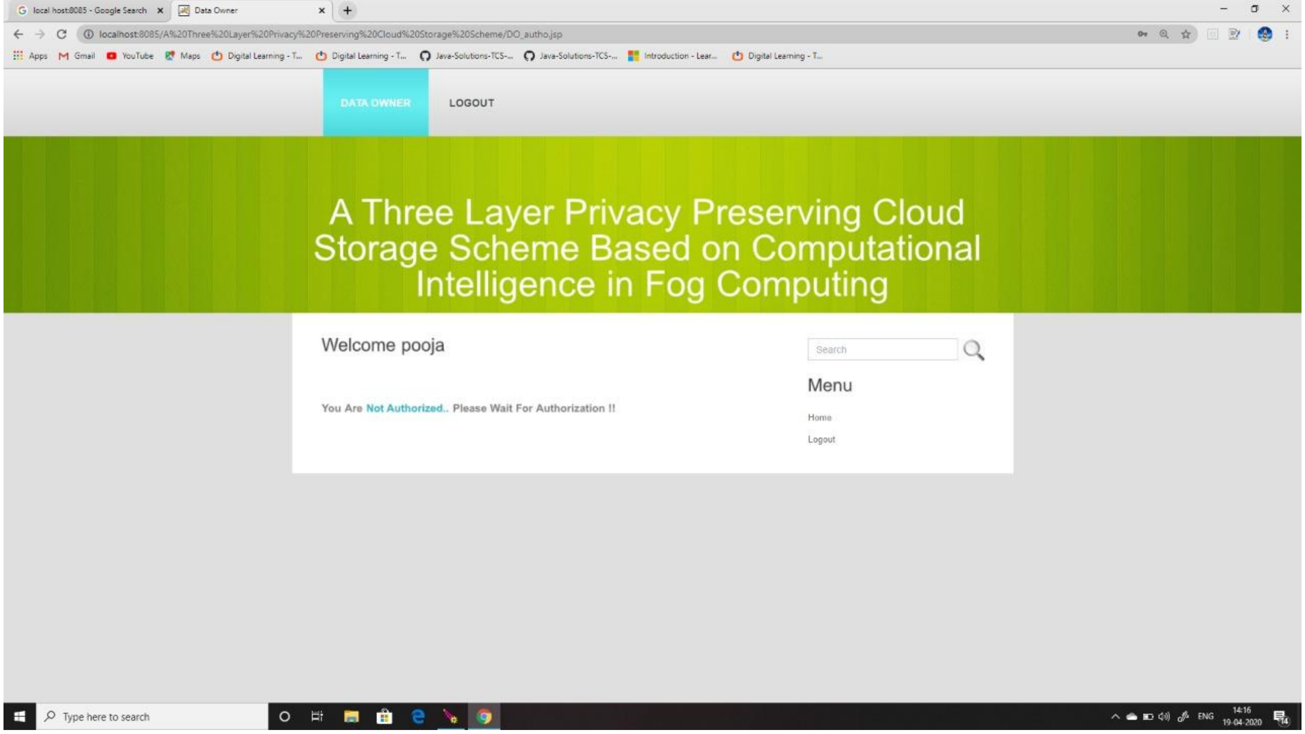
A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Welcome pooja

Data Owner Menu

- Home
- Upload Data
- View Files Blocks
- Logout

Recent years witness the development of cloud computing technology.With the explosive growth of unstructured data, cloud storage technology gets more attention and better development. However, in current storage schema, userâ€™s data is totally stored in cloud servers. In other words, users lose their right of control on data and face privacy leakage risk. Traditional privacy protection schemes are usually based on encryption technology, but these kinds of methods cannot effectively resist attack from the inside of cloud server. In order to solve this problem, we propose a three-layer storage framework based on fog computing. The proposed framework can both take full advantage of cloud storage and protect the privacy of data. Besides, Hash-Solomon code algorithm is designed to divide data into different parts. Then, we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme.


local host:8085 - Google Search
Data Owner
localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/DO_autho.jsp
DATA OWNER
LOGOUT
Search
Type here to search
14:16
19-04-2020

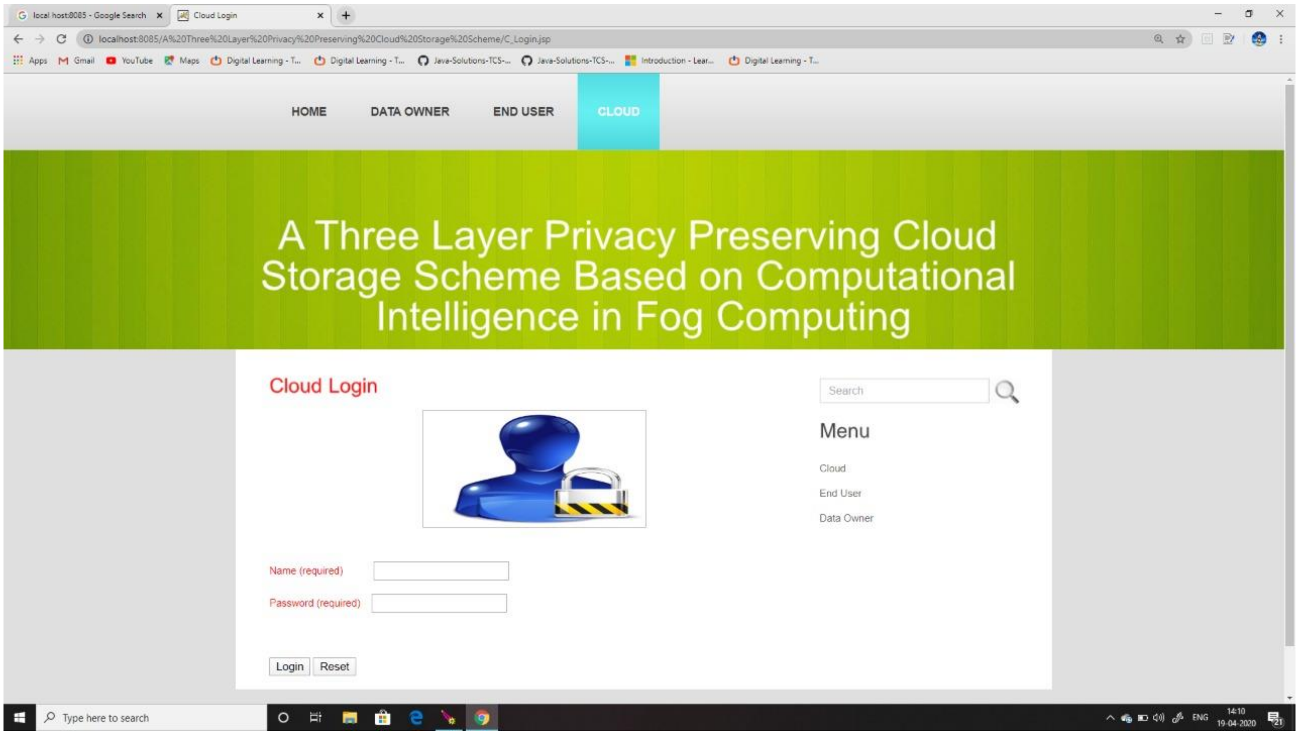
A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Welcome pooja

You Are Not Authorized.. Please Wait For Authorization !!

Menu

- Home
- Logout

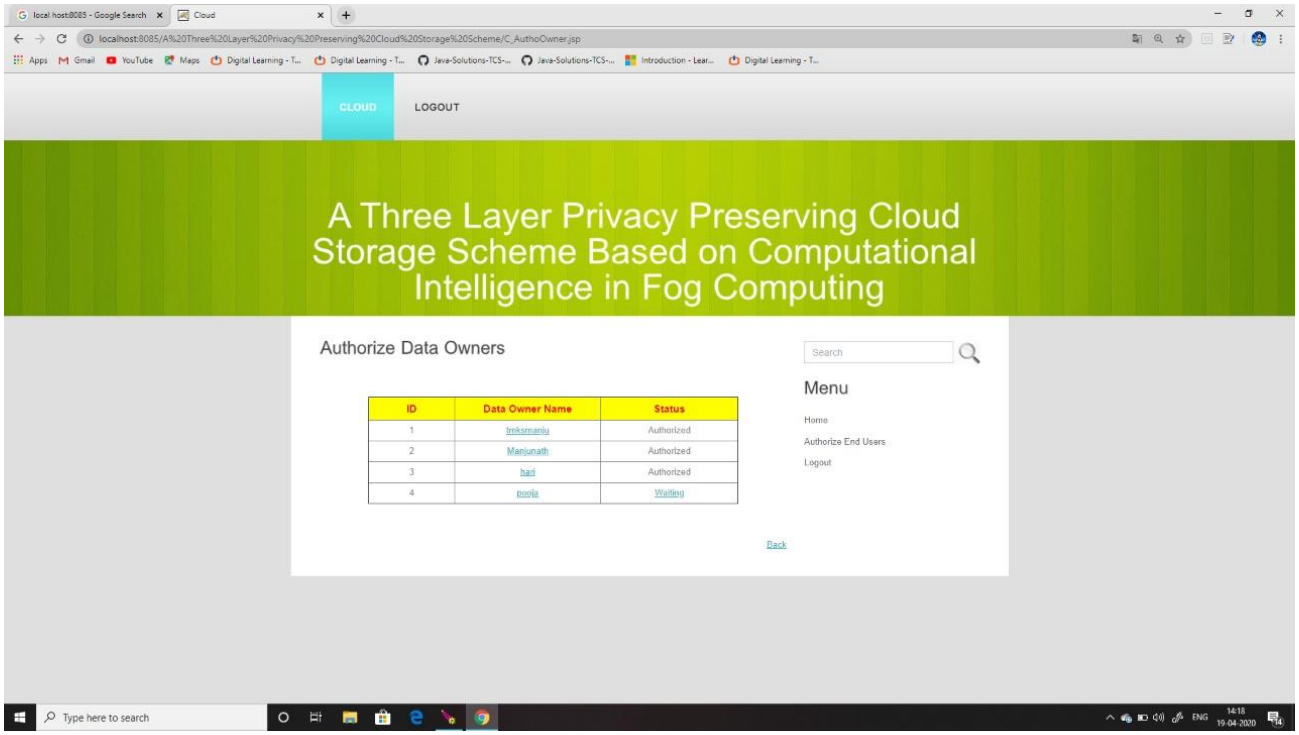


A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Cloud Login

Name (required)

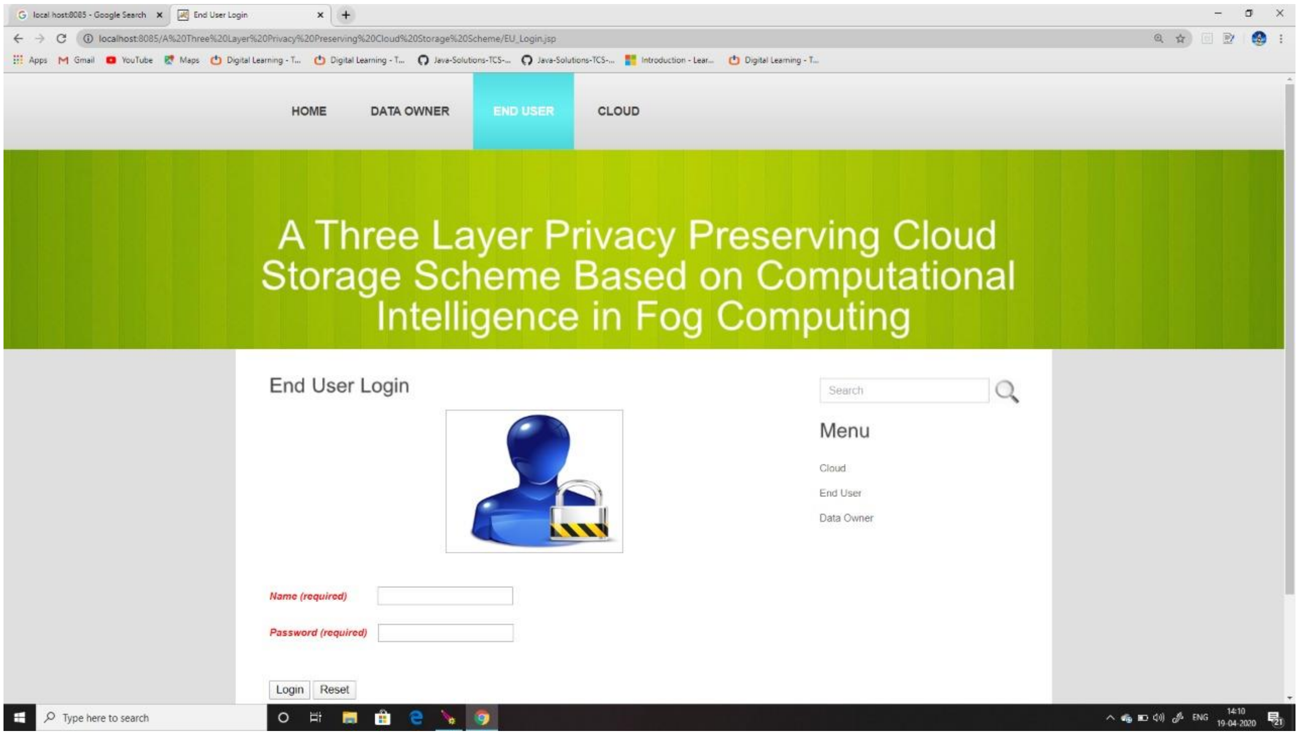
Password (required)



A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Authorize Data Owners

ID	Data Owner Name	Status
1	lmksmanju	Authorized
2	Manjunath	Authorized
3	hari	Authorized
4	pooja	Waiting


localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/EU_Login.jsp
HOME
DATA OWNER
END USER
CLOUD

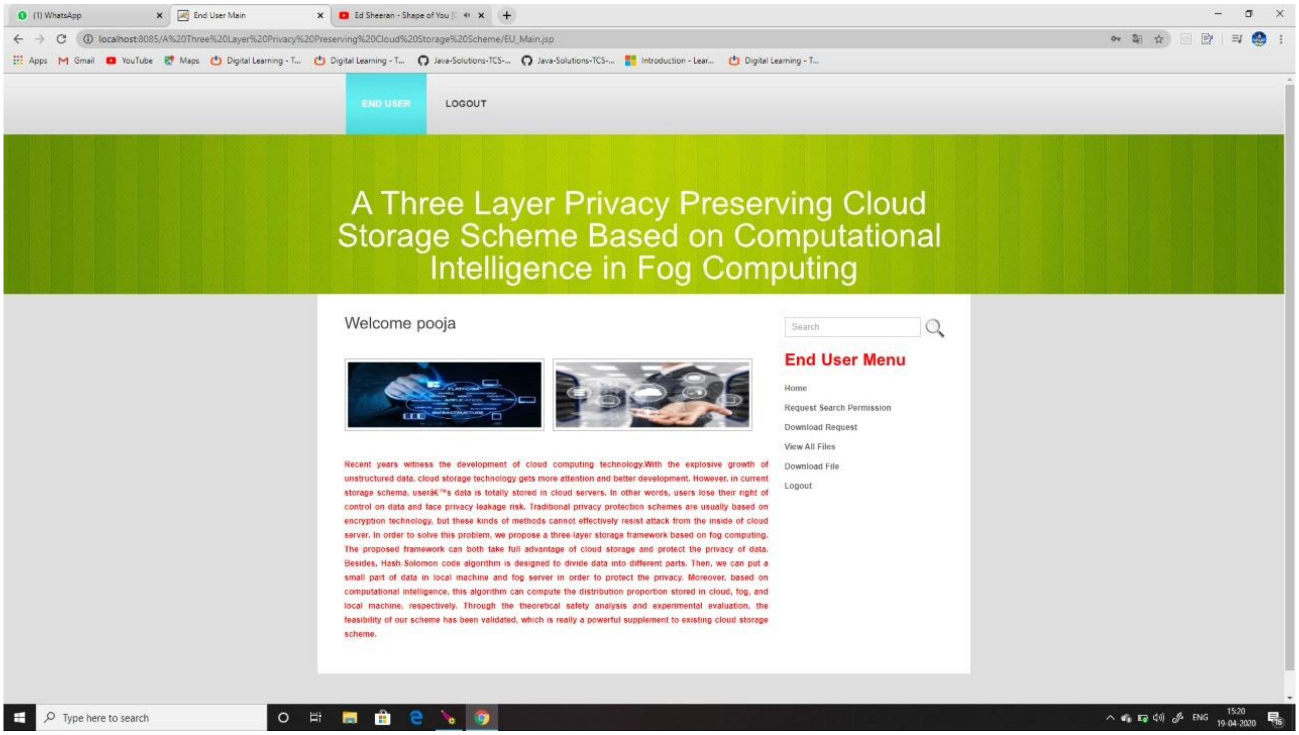
A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

End User Login

Name (required)

Password (required)

Login
Reset
Search
Menu
Cloud
End User
Data Owner

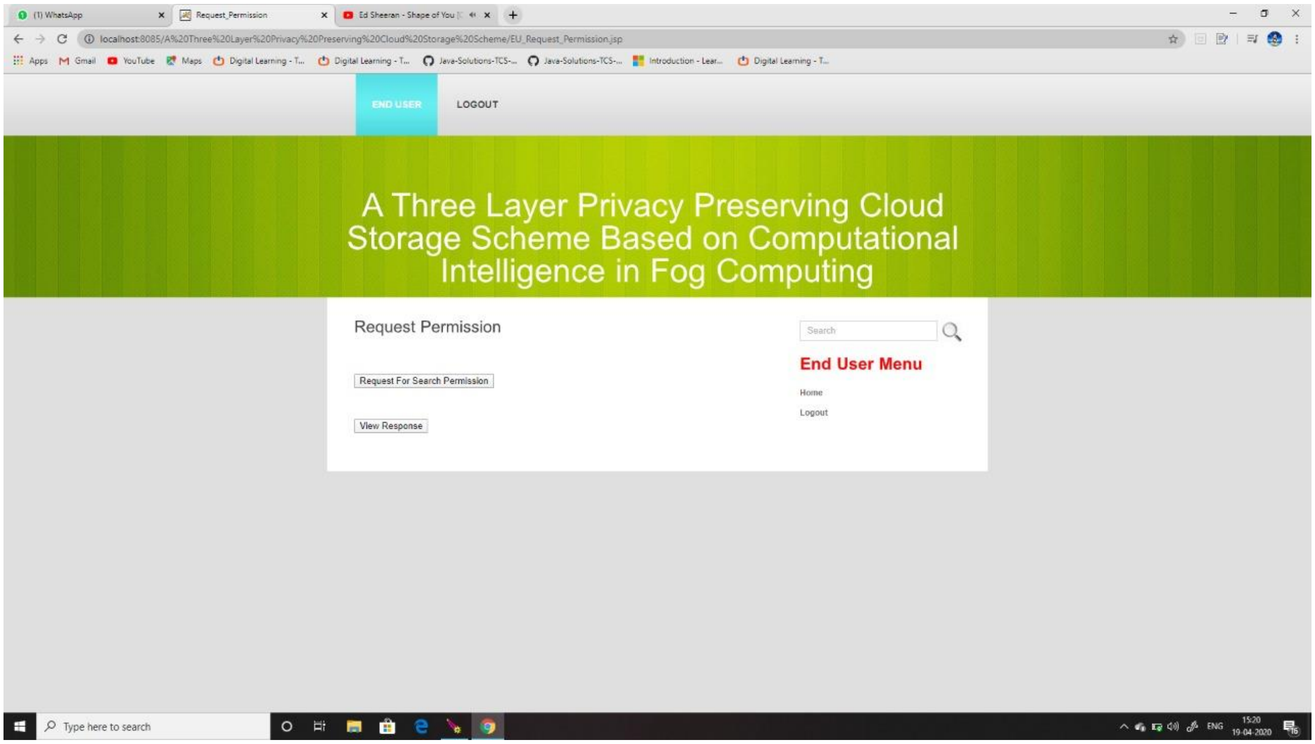

localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/EU_Main.jsp
END USER
LOGOUT

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Welcome pooja

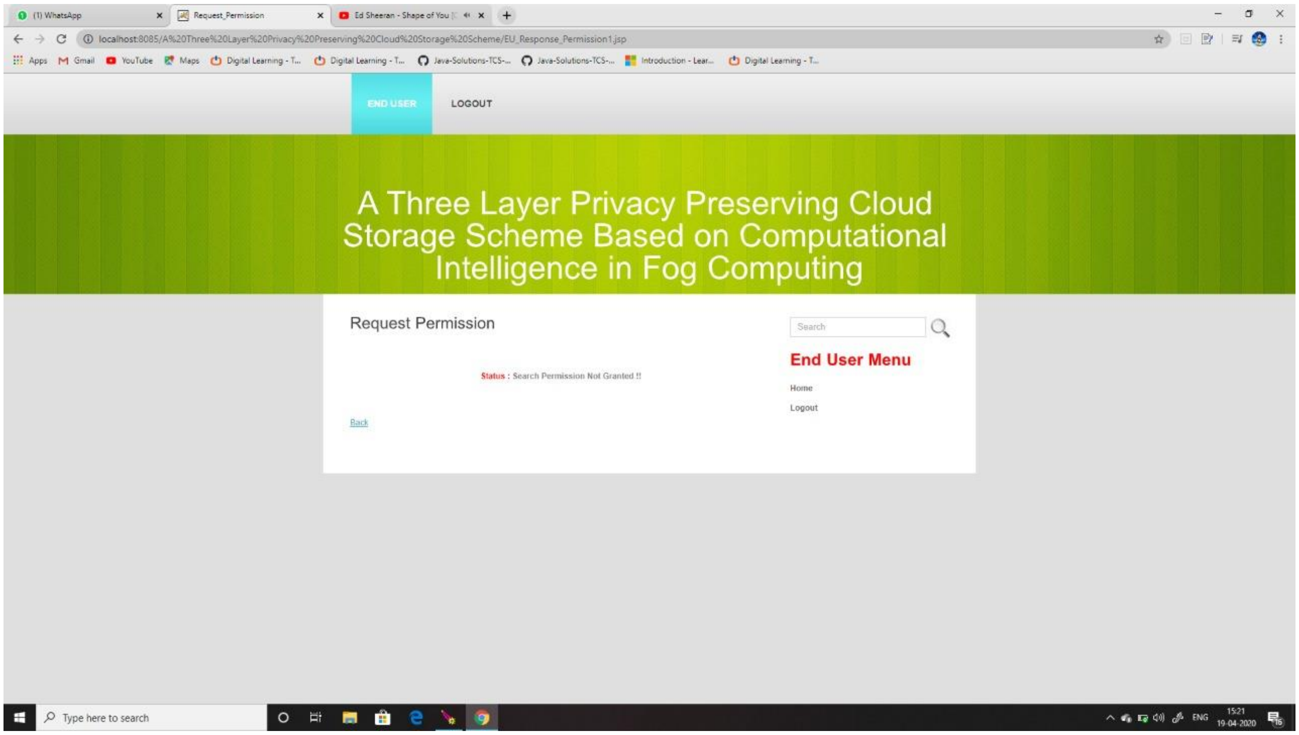
Search
End User Menu
Home
Request Search Permission
Download Request
View All Files
Download File
Logout

Recent years witness the development of cloud computing technology.With the explosive growth of unstructured data, cloud storage technology gets more attention and better development. However, in current storage schema, userâ€™s data is totally stored in cloud servers. In other words, users lose their right of control on data and face privacy leakage risk. Traditional privacy protection schemes are usually based on encryption technology, but these kinds of methods cannot effectively resist attack from the inside of cloud server. In order to solve this problem, we propose a three-layer storage framework based on fog computing. The proposed framework can both take full advantage of cloud storage and protect the privacy of data. Besides, Hash-Solomon code algorithm is designed to divide data into different parts. Then, we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme.


localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/EU_Request_Permission.jsp
END USER
LOGOUT
Search
End User Menu
Home
Logout
Request For Search Permission
View Response
Type here to search

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

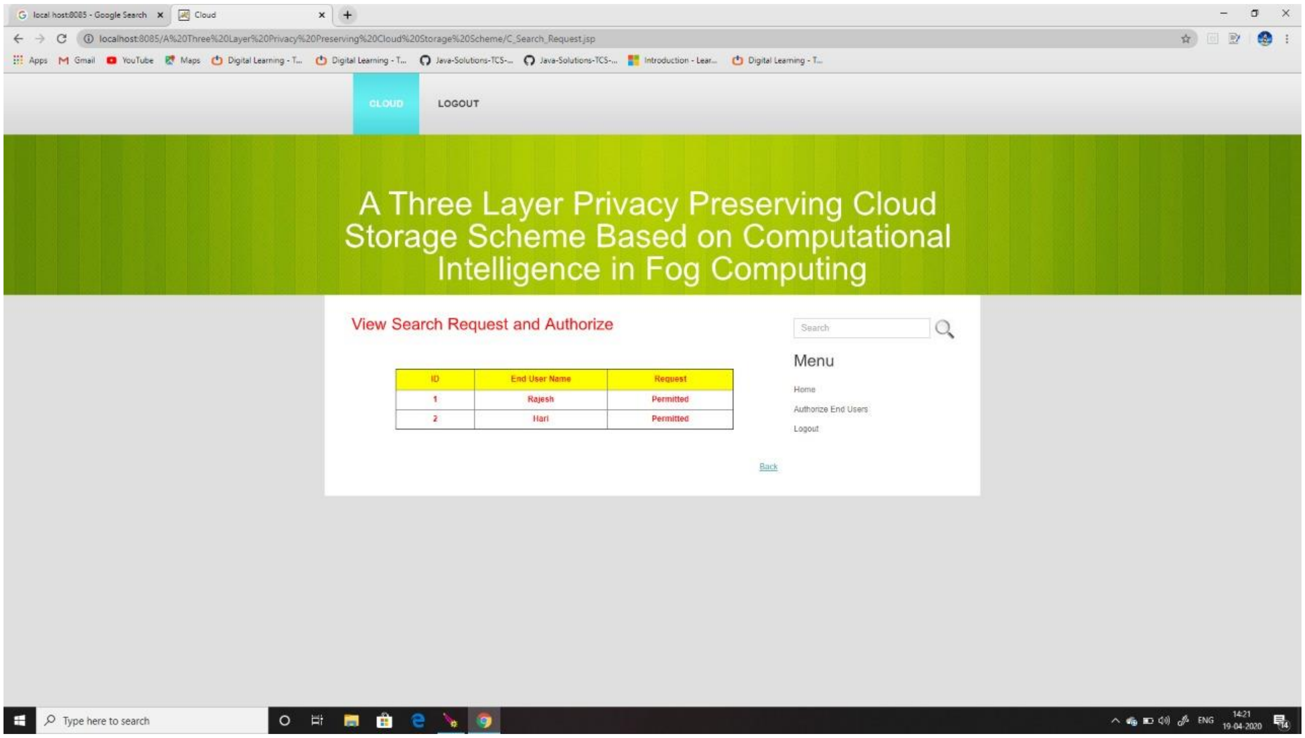
Request Permission


localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/EU_Response_Permission1.jsp
END USER
LOGOUT
Search
End User Menu
Home
Logout
Back
Type here to search

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Request Permission

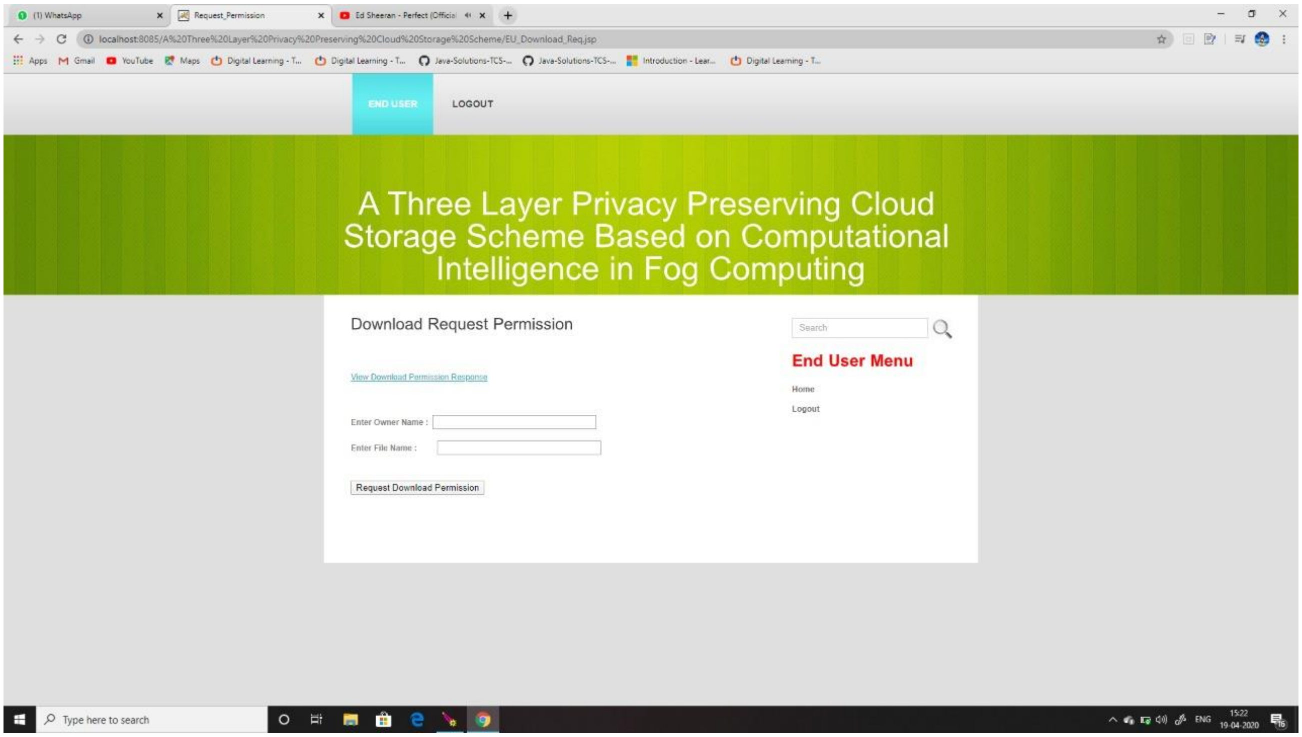
Status : Search Permission Not Granted !!



A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

View Search Request and Authorize

ID	End User Name	Request
1	Rajesh	Permitted
2	Hari	Permitted

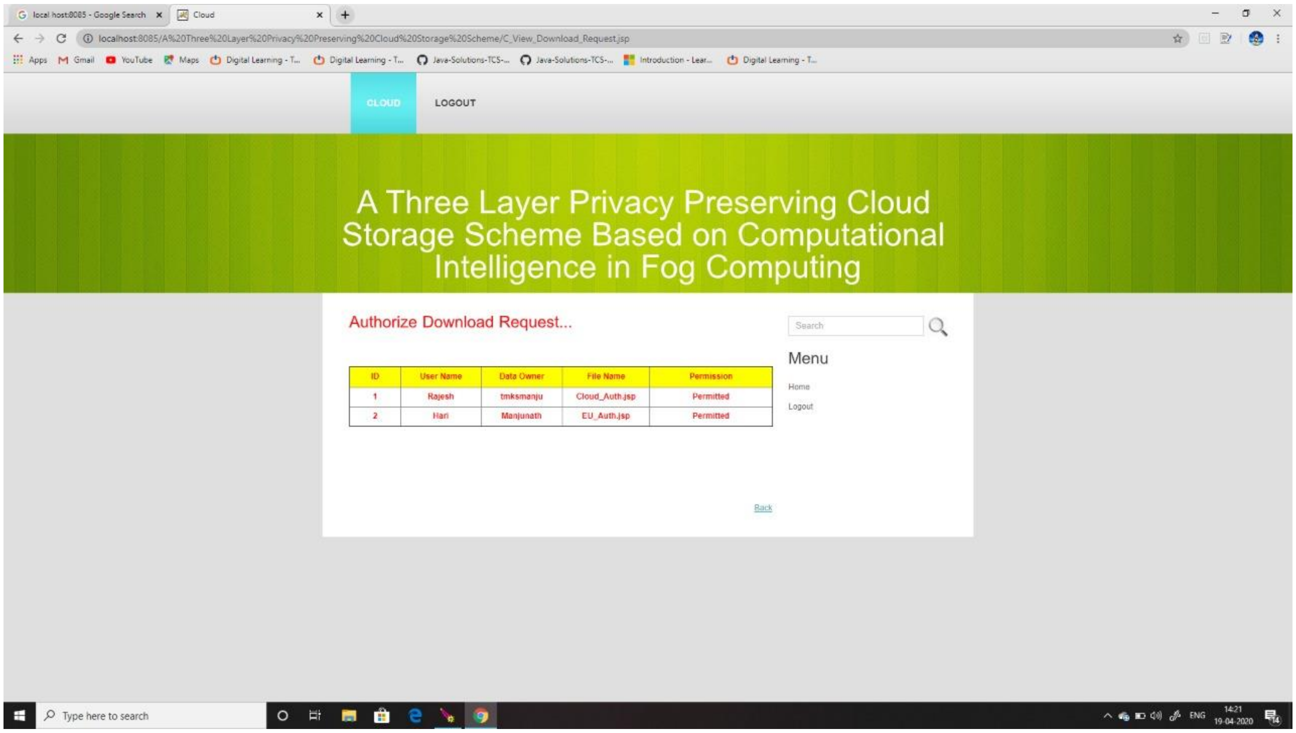

END USER
LOGOUT
Search
End User Menu
Home
Logout
View Download Permission Response
Request Download Permission

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Download Request Permission

Enter Owner Name :

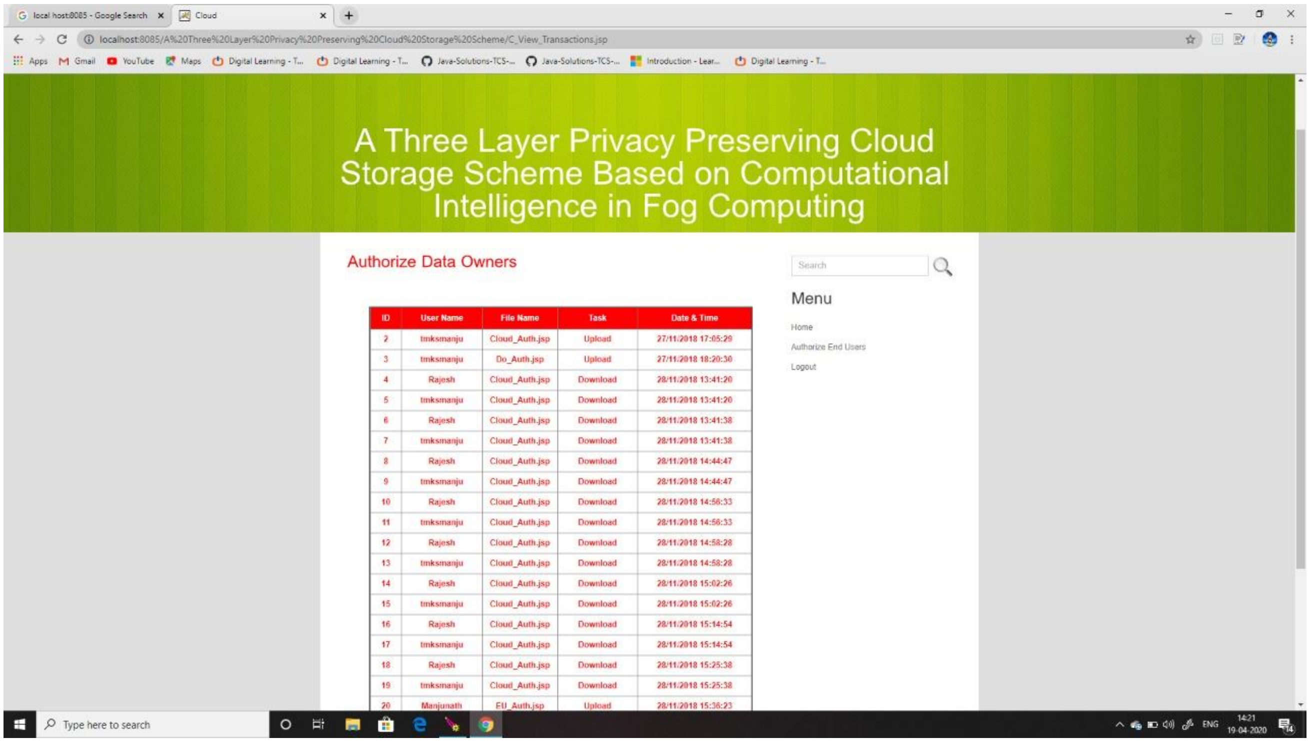
Enter File Name :



A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Authorize Download Request...

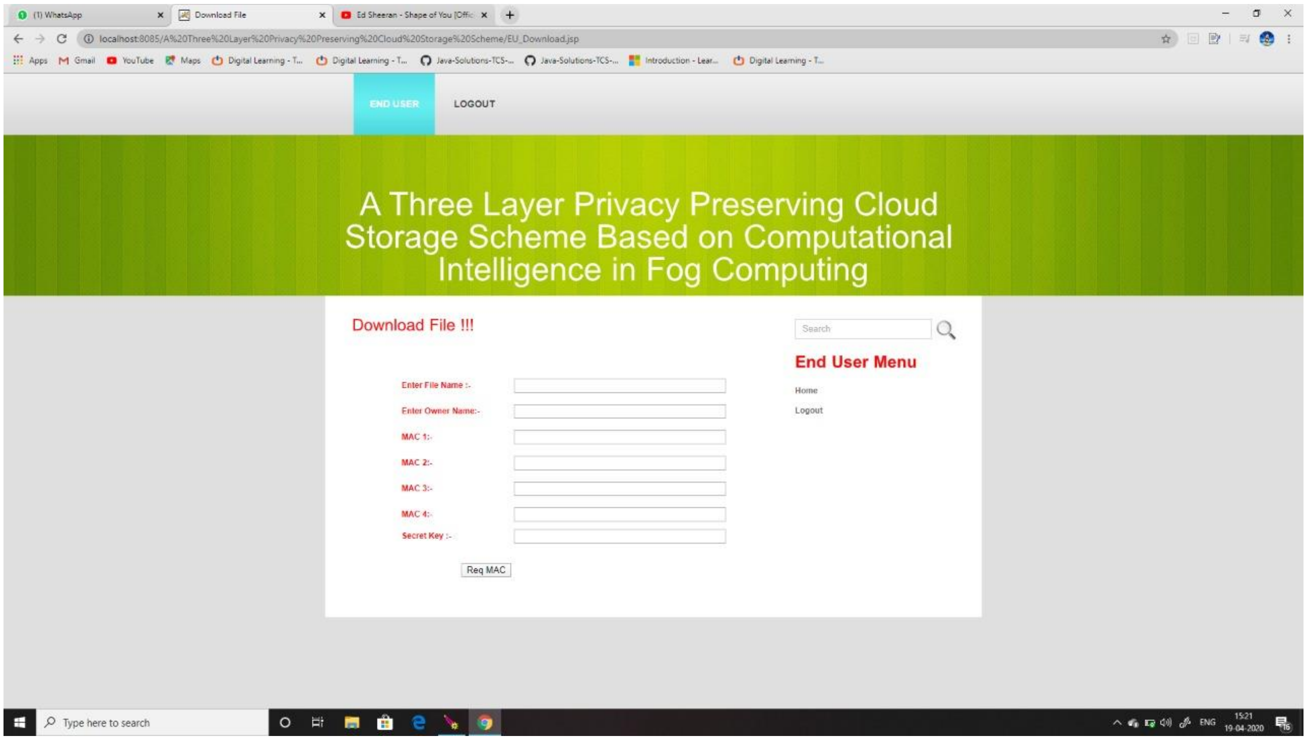
ID	User Name	Data Owner	File Name	Permission
1	Rajesh	tmksmanju	Cloud_Auth.jsp	Permitted
2	Hari	Manjunath	EU_Auth.jsp	Permitted



A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Authorize Data Owners

ID	User Name	File Name	Task	Date & Time
2	tmksmanju	Cloud_Auth.jsp	Upload	27/11/2018 17:05:29
3	tmksmanju	Do_Auth.jsp	Upload	27/11/2018 18:20:30
4	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 13:41:20
5	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 13:41:20
6	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 13:41:38
7	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 13:41:38
8	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 14:44:47
9	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 14:44:47
10	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 14:56:33
11	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 14:56:33
12	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 14:58:28
13	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 14:58:28
14	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 15:02:26
15	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 15:02:26
16	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 15:14:54
17	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 15:14:54
18	Rajesh	Cloud_Auth.jsp	Download	28/11/2018 15:25:38
19	tmksmanju	Cloud_Auth.jsp	Download	28/11/2018 15:25:38
20	Manjunath	EU_Auth.jsp	Upload	28/11/2018 15:36:23



localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/EU_Download.jsp
END USER
LOGOUT
Search
End User Menu
Home
Logout
Req MAC

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

Download File !!!

Enter File Name :-

Enter Owner Name:-

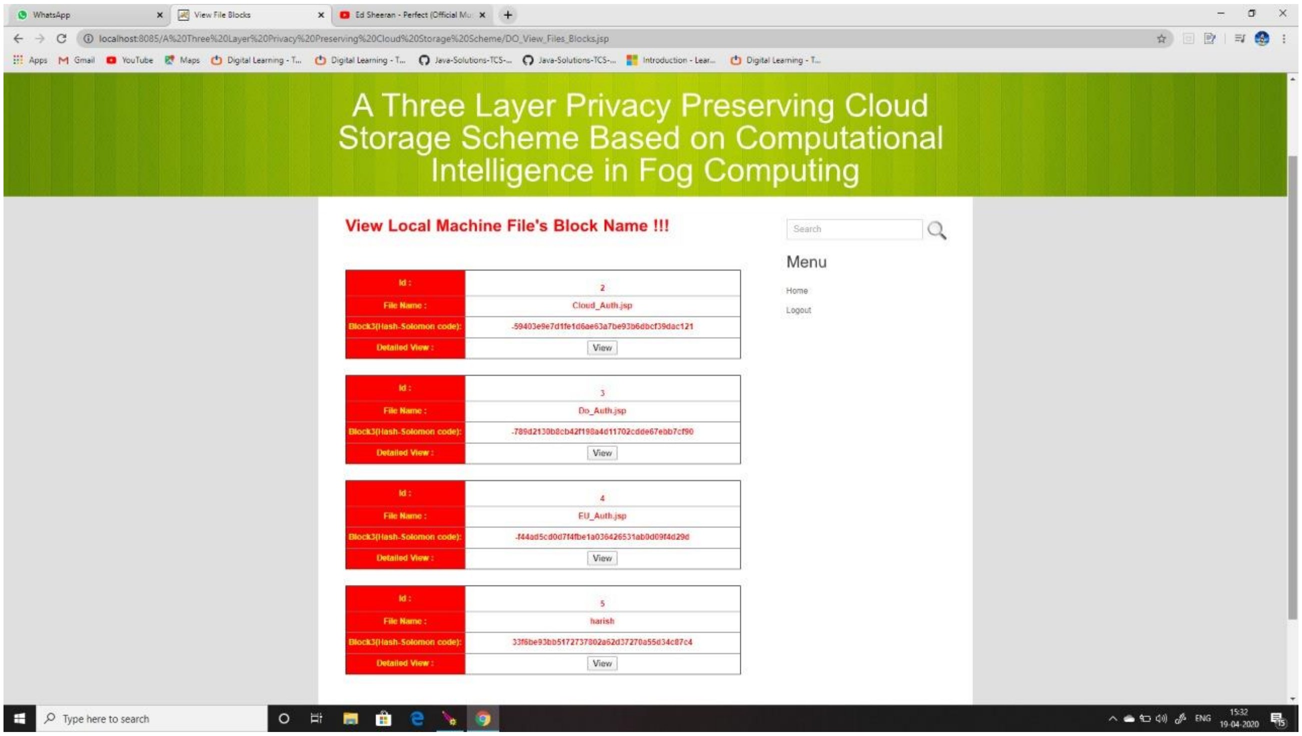
MAC 1:-

MAC 2:-

MAC 3:-

MAC 4:-

Secret Key :-



localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/DO_View_Files_Blocks.jsp
Search
Menu
Home
Logout

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

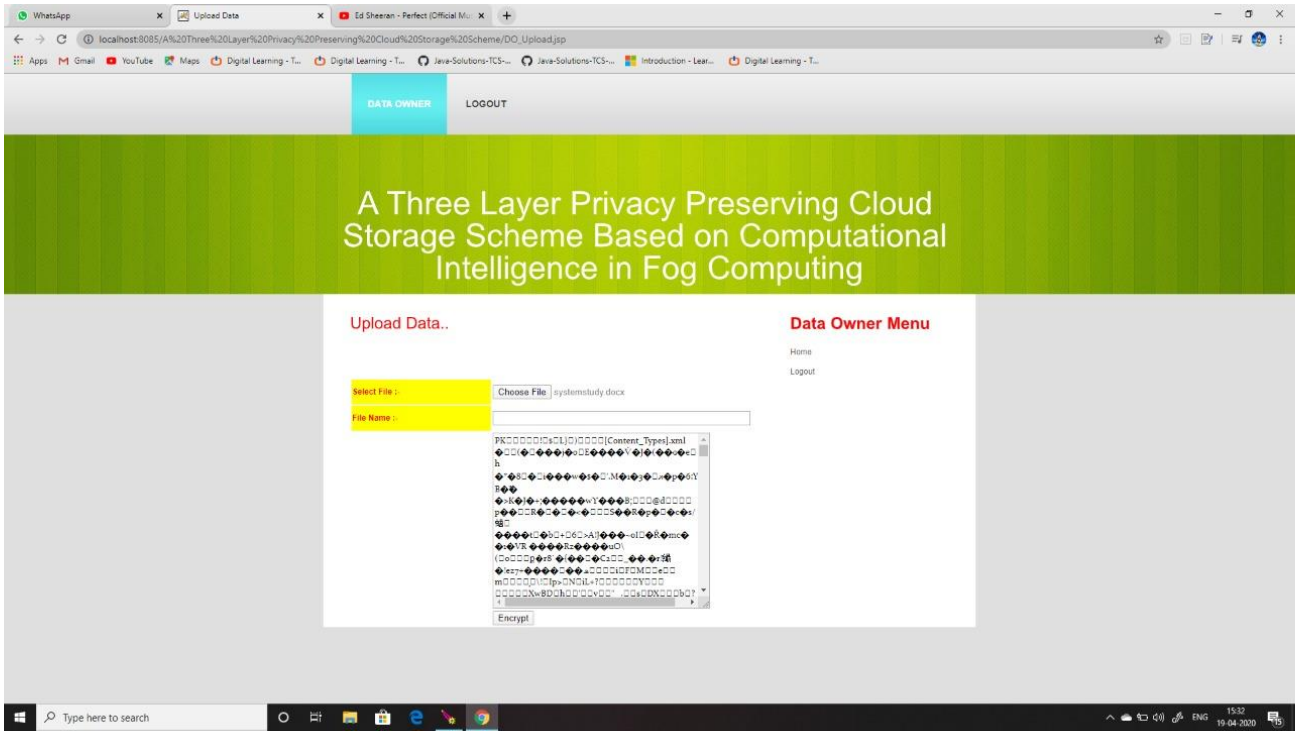
View Local Machine File's Block Name !!!

Id :	2
File Name :	Cloud_Auth.jsp
Block3(Hash-Solomon code):	-59403e9e7d1fe1d6ae63a7be93b6dbcf39dac121
Detailed View :	View

Id :	3
File Name :	Do_Auth.jsp
Block3(Hash-Solomon code):	-789d2130b8cb42f198a4d11702cdde67ebb7cf90
Detailed View :	View

Id :	4
File Name :	EU_Auth.jsp
Block3(Hash-Solomon code):	-f44ad5cd0d7f4fbe1a036426531ab0d09f4d29d
Detailed View :	View

Id :	5
File Name :	harish
Block3(Hash-Solomon code):	33f6be93bb5172737802a62d37270a55d34c87c4
Detailed View :	View



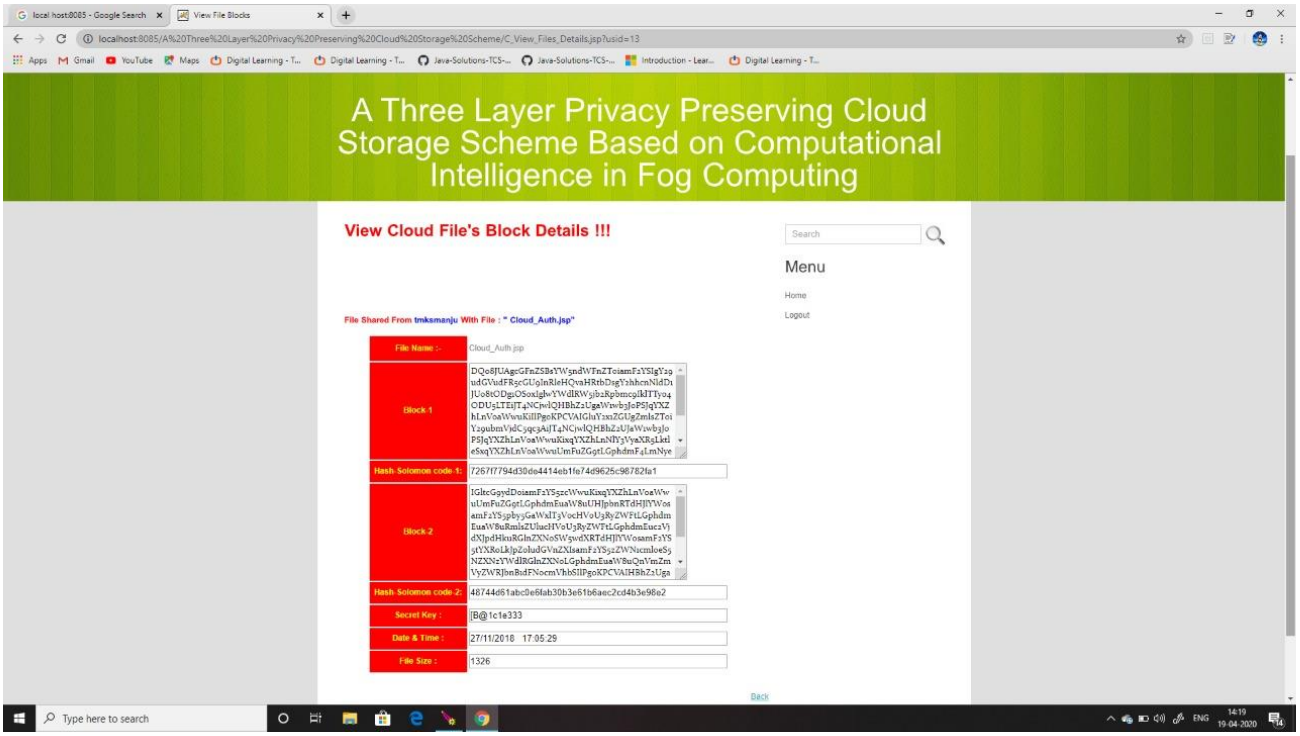
WhatsApp
Upload Data
Ed Sheeran - Perfect (Official Mu
localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/DO_Upload.jsp
Apps
Gmail
YouTube
Maps
Digital Learning - T...
Java-Solutions-TCS-...
Introduction - Lear...
DATA OWNER
LOGOUT
A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing
Data Owner Menu
Home
Logout
Type here to search
ENG
15:32
19-04-2020

Upload Data..

Select File :-	Choose File systemstudy.docx
File Name :-	

```
PK□□□□□!□s□L)□)□□□□[Content_Types].xml
�□□(�□���)�o□E����Ŷ�J�(��o�e□
h
�"�8□�□i���w�s�□'.M�1�3�□ж�p�6:Y
B��
�>K�J�+;�����wY���B;□□□@d□□□□
p��□□R�□�□�<�□□□S��R�p�□�c�s/
蝂□
����t□�b□+□6□>A!J���~oI□�Ř�mc�
�1�VR ����Rz����uO\
(□o□□□p�r8`�{��□�C2□□_��.�r鞝
�!ez7+����□��ѧ□□□□i□F□M□□e□□
m□□□□,□\!□Ip>□N□iL+?□□□□□□Y□□□
□□□□□XwBD□h□□'□□v□□" .□□s□DX□□□b□?
```

Encrypt

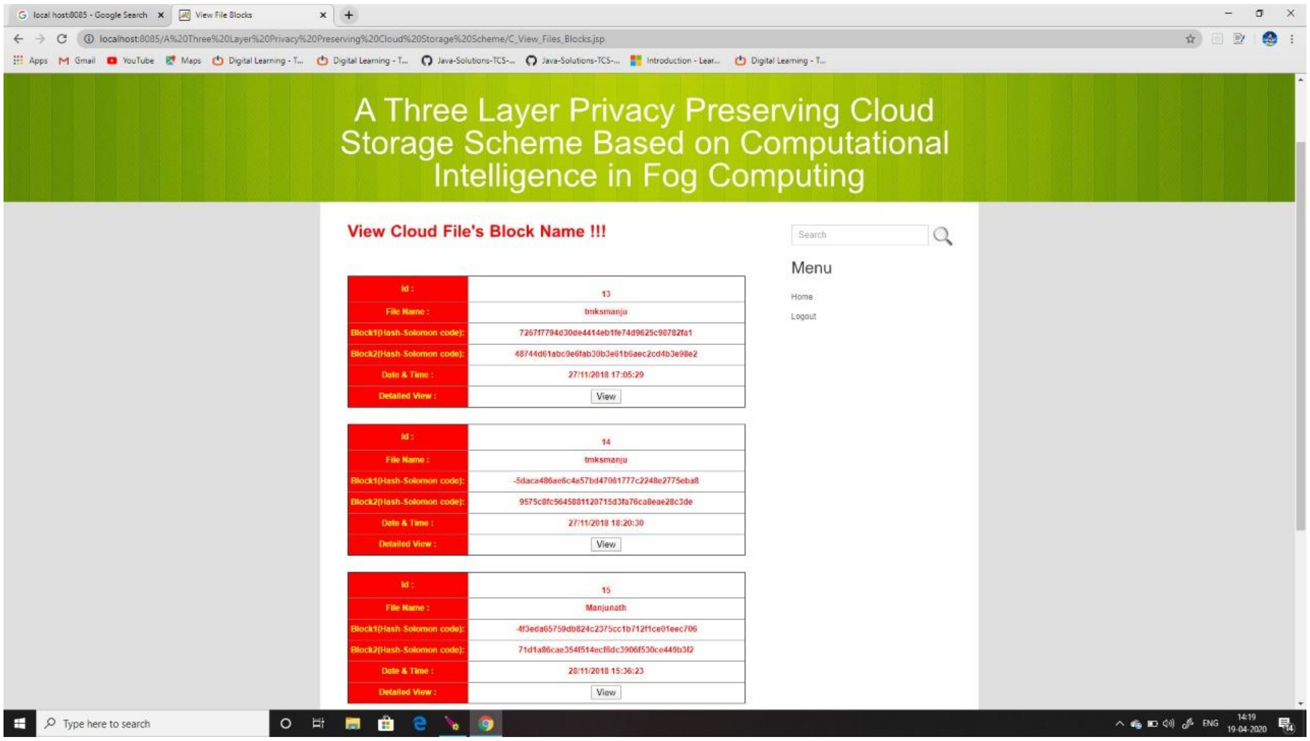


local host:8085 - Google Search
View File Blocks
localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/C_View_Files_Details.jsp?usid=13
Apps
Gmail
YouTube
Maps
Digital Learning - T...
Java-Solutions-TCS-...
Introduction - Lear...
A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing
Search
Menu
Home
Logout
Back
Type here to search
ENG
14:19
19-04-2020

View Cloud File's Block Details !!!

File Shared From tmksmanju With File : " Cloud_Auth.jsp"

File Name :-	Cloud_Auth.jsp
Block-1	DQo8JUAgcGFnZSBsYW5ndWFnZToiamF2YSIgY29 udGVudFR5cGU9InRleHQvaHRtbDsgY2hhcnNldD1 JUo8tODg1OSoxIglwYWdlRW5jb2Rpbmc9IkITTyo4 ODU5LTEiJT4NCjwlQHBhZ2UgaW1wb3JoPSJqYXZ hLnVoaWwuKiIlPg0KPCVAIGluY2x1ZGUgZmlsZToi Y29ubmVjdC5qc3AiJT4NCjwlQHBhZ2UJaW1wb3Jo PSJqYXZhLnVoaWwuKixqYXZhLnNlY3VyaXR5Lktl eSxqYXZhLnVoaWwuUmFuZG9tLGphdmF4LmNye
Hash-Solomon code-1:	7267f7794d30de4414eb1fe74d9625c98782fa1
Block-2	IGltcG9ydDoiamF2YS5zcWwuKixqYXZhLnVoaWw uUmFuZG9tLGphdmEuaW8uUHJpbnRTdHJlYWos amF2YS5pby5GaWxlT3VocHVoU3RyZWFtLGphdm EuaW8uRmlsZUlucHVoU3RyZWFtLGphdmEuc2Vj dXJpdHkuRGlnZXNoSW5wdXRTdHJlYWosamF2YS 5tYXRoLkJpZoludGVnZXIsamF2YS52ZWN1cmloeS5 NZXNzYWdlRGlnZXNoLGphdmEuaW8uQnVmZm VyZWRJbnB1dFNocmVhbSIlPg0KPCVAIHBhZ2Uga
Hash-Solomon code-2:	48744d61abc0e6fab30b3e61b6aec2cd4b3e98e2
Secret Key :	[B@1c1e333
Date & Time :	27/11/2018 17:05:29
File Size :	1326



localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/C_View_Files_Blocks.jsp

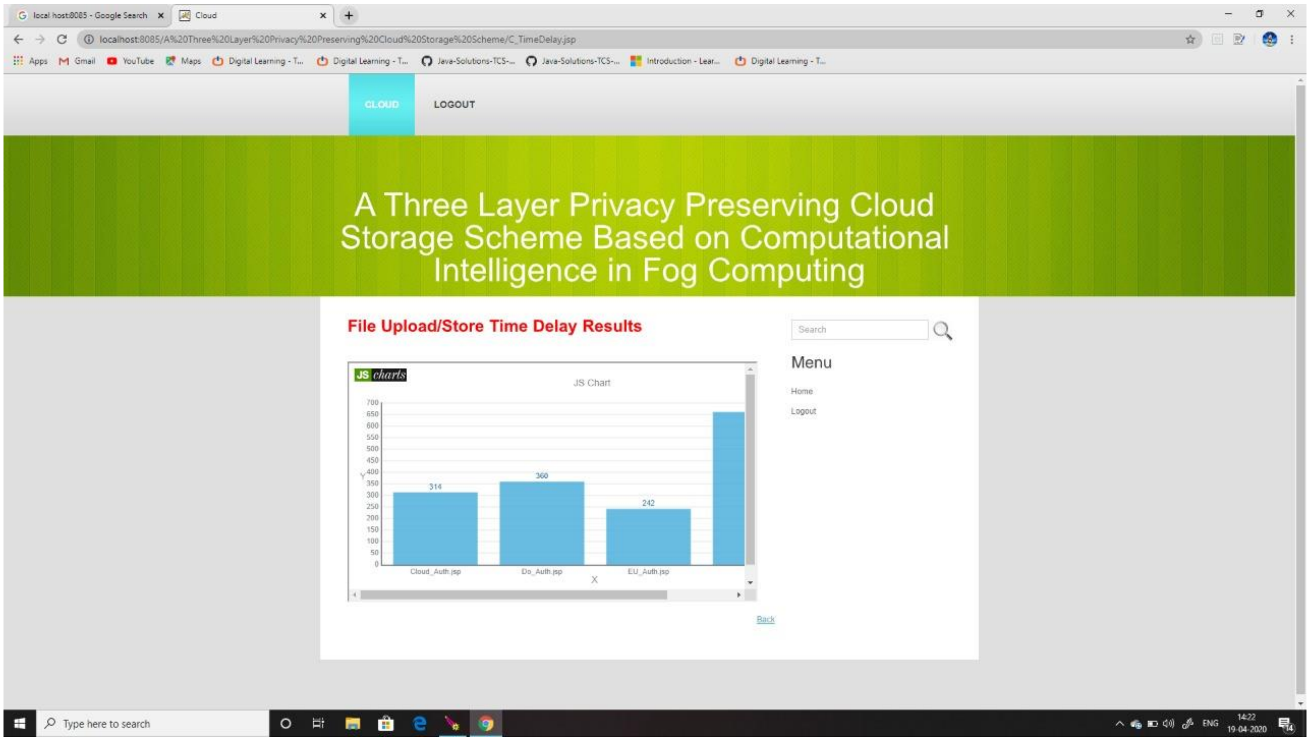
A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

View Cloud File's Block Name !!!

Id :	13
File Name :	tmksmanju
Block1(Hash-Solomon code):	7267f7794d30de4414eb1fe74d9625c98782fa1
Block2(Hash-Solomon code):	48744d61abc0e6fab30b3e61b6aec2cd4b3e98e2
Date & Time :	27/11/2018 17:05:29
Detailed View :	View

Id :	14
File Name :	tmksmanju
Block1(Hash-Solomon code):	-5daca486ae6c4a57bd47081777c2248e2775eba8
Block2(Hash-Solomon code):	9575c8fc5645881120715d3fa76ca8eae28c3de
Date & Time :	27/11/2018 18:20:30
Detailed View :	View

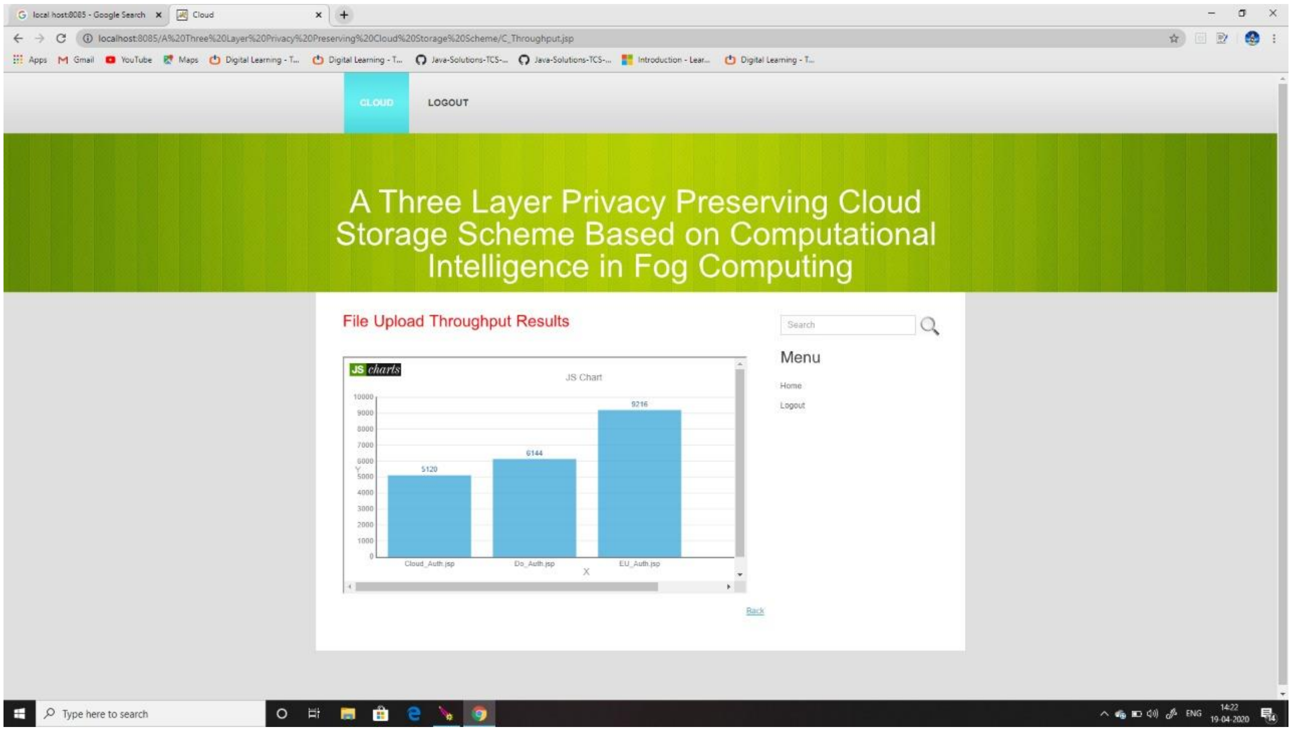
Id :	15
File Name :	Manjunath
Block1(Hash-Solomon code):	-4f3eda65759db824c2375cc1b712f1ce01eec706
Block2(Hash-Solomon code):	71d1a86cae354f514ecf6dc3906f530ce449b3f2
Date & Time :	28/11/2018 15:36:23
Detailed View :	View



localhost:8085/A%20Three%20Layer%20Privacy%20Preserving%20Cloud%20Storage%20Scheme/C_TimeDelay.jsp

A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

File Upload/Store Time Delay Results



A Three Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing

File Upload Throughput Results

7. SYSTEM TESTING

7.1 Testing Methodologies

The following are the Testing Methodologies:

- Unit Testing.
- Integration Testing.
- User Acceptance Testing.
- Output Testing.
- Validation Testing.

7.1.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

7.1.2 Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

7.1.3 User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

7.1.4 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

7.1.5 Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

7.2 User Training

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

7.3. Maintenance

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

Testing Strategy :

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

System Testing

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

Unit Testing

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.The future holds a lot to offer to the development and refinement of this project.

7.4.Test Cases

SNO	Test case	Requirement specification	Expected output	Observed output	Status P=pass F=fail
1	Register with a high storage capacity picture	Certain capacity of storage to get registered	Login page will be shown	Web page will become blank	F
2	Login with just user id and password by the user	Cloud authourization	User's page will be shown on the screen	Still shows the login page	F
3	Data blocks can be seen in cloud server	User id and password	Data blocks will be shown	Data blocks will be shown	T
4	Data blocks in can be seen using the authourization of cloud by End User	Signature key	Parts of data will be shown on the screen	Parts of data will be shown on the screen	T
5	Searching for a file in server with a single click by any one.	Search permission by cloud	Required file will be seen in page	There will be no change in the page	F
6	Fog server can see all the parts of data	Fog server's user id and password	Only the part that stored in fog server	Whole file that is separated into three parts can be seen on screen	T

8.CONCLUSION

The development of cloud computing brings us a lot of benefits. Cloud storage is a convenient technology which helps users to expand their storage capacity. However, cloud storage also causes a series of secure problems. When using cloud storage, users do not actually control the physical storage of their data and it results in these partition of ownership and management of data. In order to solve the problem of privacy protection in cloud storage, we propose a TLS framework based on fog computing model and design a Hash-Solomon algorithm. Through the theoretical safety analysis, the scheme is proved to be feasible. By allocating the ratio of data blocks stored in different servers reasonably, we can ensure the privacy of data in each server. On another hand, cracking the encoding matrix is impossible theoretically. Besides, using hash transformation can protect the fragmentary information. Through the experiment test, this scheme can efficiently complete encoding and decoding without influence of the cloud storage efficiency. Furthermore, we design a reasonable comprehensive efficiency index, in order to achieve the maximum efficiency, and we also find that the Cauchy matrix is more efficient in coding process.

9.REFERENCES

[1] P. Mell and T. Grance, “The NIST definition of cloud computing,” Nat. Inst. Stand. Technol., vol. 53, no. 6, pp. 50–50, 2009.

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: Architecture, applications, and approaches,” Wireless Commun. Mobile Comput., vol. 13, no. 18, pp. 1587–1611, 2013.

[3] J. Chase, R. Kaewpuang, W. Yonggang, and D. Niyato, “Joint virtual machine and bandwidth allocation in software defined network (sdn) and cloud computing environments,” in Proc.IEEE Int .Conf. Commun., 2014, pp. 2969–2974.

[4] H. Li, W. Sun, F. Li, and B. Wang, “Secure and privacy-preserving data storage service in public cloud,” J. Comput. Res. Develop., vol. 51, no. 7, pp. 1397–1409, 2014.

[5] Y.Li,T.Wang,G.Wang,J.Liang,andH.Chen,“Efficientdatacollection in sensor-cloud system with multiple mobile sinks,” in Proc. Adv. Serv. Comput., 10th Asia-Pac. Serv. Comput. Conf., 2016, pp. 130–143.

[6] L. Xiao, Q. Li, and J. Liu, “Survey on secure cloud storage,” J. Data Acquis. Process., vol. 31, no. 3, pp. 464–472, 2016.

[7] R. J. McEliece and D. V. Sarwate, “On sharing secrets and reed-solomon codes,” Commun. ACM, vol. 24, no. 9, pp. 583–584, 1981.

[8] J. S. Plank, “T1: Erasure codes for storage applications,” in Proc. 4th USENIX Conf. File Storage Technol., 2005, pp. 1–74.