INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW

Searchable encryption is of increasing interest for protecting the data privacy in secure searchable cloud storage. In this paper, we investigate the security of a well-known cryptographic primitive, namely, public key encryption with keyword search (PEKS) which is very useful in many applications of cloud storage. Unfortunately, it has been shown that the traditional PEKS framework suffers from an inherent insecurity called inside keyword guessing attack (KGA) launched by the malicious server. To address this security vulnerability, we propose a new PEKS framework named dual-server PEKS (DS-PEKS). As another main contribution, we define a new variant of the smooth projective hash functions (SPHFs) referred to as linear and homomorphic SPHF (LH-SPHF). We then show a generic construction of secure DS-PEKS from LH-SPHF. To illustrate the feasibility of our new framework, we provide an efficient instantiation of the general framework from a Decision Diffie–Hellman-based LH-SPHF and show that it can achieve the strong security against inside the KGA.

LITERATURE REVIEW

2. LITERATURE REVIEW

2.1 LITERATURE REVIEW:

End users may not entirely trust the cloud storage servers and may prefer to encrypt their data before uploading them to the cloud server in order to protect the data privacy. This usually makes the data utilization more difficult than the traditional storage where data is kept in the absence of encryption.

One of the typical solutions is the searchable encryption which allows the user to retrieve the encrypted documents that contain the user-specified keywords, where given the keyword trapdoor, the server can find the data required by the user without decryption. Song et al. proposed keyword search on ciphertext, known as Searchable Symmetric Encryption (SSE).

Searchable symmetric encryption allows a party to outsource the storage of his data to another party in a private manner, while maintaining the ability to selectively search over it. This problem has been the focus of active research and several security definitions and constructions have been proposed. Although SSE schemes enjoy high efficiency, they suffer from complicated secret key distribution. Precisely, users have to securely share secret keys which are used for data encryption. Otherwise they are not able to share the encrypted data outsourced to the cloud.

To resolve this problem, Boneh et al. introduced a more flexible primitive, namely Public Key Encryption with Keyword Search (PEKS) that enables a user to search encrypted data in the asymmetric encryption setting. It allows users to search encrypted documents on an untrusted server without revealing any information. This notion is very useful in many applications and has attracted a lot of attention by the cryptographic research community. However, one limitation of all the existing PEKS schemes is that they cannot resist the Keyword Guessing Attack (KGA) launched by a malicious server.

2.2 EXISTING SYSTEM

- In a PEKS system, using the receiver's public key, the sender attaches some encrypted keywords (referred to as PEKS ciphertexts) with the encrypted data. The receiver then sends the trapdoor of a to-be-searched keyword to the server for data searching. Given the trapdoor and the PEKS ciphertext, the server can test whether the keyword underlying the PEKS ciphertxt is equal to the one selected by the receiver. If so, the server sends the matching encrypted data to the receiver.

- Baek et al. proposed a ew PEKS scheme without requiring a secure channel, which is referred to as a secure channel-free PEKS (SCF-PEKS).

- Rhee et al. later enhanced Baek et al.'s security model for SCF-PEKS where the attacker is allowed to obtain the relationship between the non-challenge ciphertexts and the trapdoor.

- Byun et al.introduced the off-line keyword guessing attack against PEKS as keywords are chosen from a much smaller space than passwords and users usually use well-known keywords for searching documents.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- Despite of being free from secret key distribution, PEKS schemes suffer from an inherent insecurity regarding the trapdoor keyword privacy, namely inside Keyword Guessing Attack (KGA). The reason leading to such a security vulnerability is that anyone who knows receiver's public key can generate the PEKS ciphertext of arbitrary keyword himself.

- Specifically, given a trapdoor, the adversarial server can choose a guessing keyword from the keyword space and then use the keyword to generate a PEKS ciphertext. The server then can test whether the guessing keyword is the one underlying the trapdoor. This guessing-then-testing procedure can be repeated until the correct keyword is found.

- On one hand, although the server cannot exactly guess the keyword, it is still able to know which small set the underlying keyword belongs to and thus the keyword privacy is not well preserved from the server. On the other hand, their scheme is impractical as the receiver has to locally find the matching ciphertext by using the exact trapdoor to filter out the non-matching ones from the set returned from the server.

2.3 PROPOSED SYSTEM

The contributions of this paper are four-fold.

- We formalize a new PEKS framework named Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) to address the security vulnerability of PEKS.
- A new variant of Smooth Projective Hash Function (SPHF), referred to as linear and homomorphic SPHF, is introduced for a generic construction of DS-PEKS.
- We show a generic construction of DS-PEKS using the proposed Lin-Hom SPHF.
- To illustrate the feasibility of our new framework, an efficient instantiation of our SPHF based on the Diffie-Hellman language is presented in this paper.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM:

- All the existing schemes require the pairing computation during the generation of PEKS ciphertext and testing and hence are less efficient than our scheme, which does not need any pairing computation.
- Our scheme is the most efficient in terms of PEKS computation. It is because that our scheme does not include pairing computation. Particularly, the existing scheme requires the most computation cost due to 2 pairing computation per PEKS generation.
- In our scheme, although we also require another stage for the testing, our computation cost is actually lower than that of any existing scheme as we do not require any pairing computation and all the searching work is handled by the server.

REQUIREMENT SPECIFICATIONS

3. REQUIREMENT SPECIFICATIONS

3.1 PRODUCT PERSPECTIVE:

The project DS-PEKS is a dependent application. DS-PEKS is a real-time application in the domain of JAVA

3.2 FUNCTIONAL REQUIREMENTS:

3.2.1 NUMBER OF MODULES:

After careful analysis the system has been identified to have the following modules:

1. System Construction Module
2. Semantic-Security against Chosen Keyword Attack
3. Front Server
4. Back Server

3.2.2 MODULES DESCRIPTION:

1. System Construction Module

In the first module, we develop the system with the entities required to provide our system. 1) Cloud User: the user, who can be an individual or an organization originally storing their data in cloud and accessing the data. 2) Cloud Service Provider (CSP): the CSP, who manages cloud servers (CSs) and provides a paid storage space on its infrastructure to users as a service. We propose a new framework, namely DS-PEKS, and present its formal definition and security models. We then define a new variant of smooth projective hash function (SPHF). A generic construction of DS-PEKS from LH-SPHF is shown with formal correctness analysis and security proofs. Finally, we present an efficient instantiation of DS-PEKS from SPHF.

2. **Semantic-Security against Chosen Keyword Attack**

In the module, we develop the semantic-security against chosen keyword attack which guarantees that no adversary is able to distinguish a keyword from another one given the corresponding PEKS ciphertext. That is, the PEKS ciphertext does not reveal any information about the underlying keyword to any adversary.

3. **Front Server**

After receiving the query from the receiver, the front server pre-processes the trapdoor and all the PEKS ciphertexts using its private key, and then sends some internal testing-states to the back server with the corresponding trapdoor and PEKS ciphertexts hidden.

4. **Back Server**

In this module, the back server can then decide which documents are queried by the receiver using its private key and the received internal testing-states from the front server.

3.3 NON-FUNCTIONAL REQUIREMENTS

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI'S at the top level have been categorized as Non-functional requirements concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. Here data is going to collect from multipath or any-path way to receiver side and send from sender side. Data must be secure while sending through network. Must not know to third party also that are going to store and process the data. Data can be search by specific keywords but they must be specified by data owner. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updating along with the extensive data search capabilities.

There are following non-functional requirements for proposed system:

- Two Server
- Scalability and speedup
- Better time and space efficiency
- Throughput
- Comparison with another related algorithms
- Minimum response time

3.4 HARDWARE AND SOFTWARE REQUIREMENTS:

3.4.1 HARDWARE INTERFACE:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

3.4.2 SOFTWARE INTERFACE:

- Operating system : Windows XP/7.
- Coding Language : JAVA/J2EE
- Data Base : MYSQL

3.4.3 DEVELOPMENT END:

- Technology : JAVA (J2EE), JSP
- Web Technologies :Html, JavaScript, CSS

TECHNOLOGIES USED

4. TECHNOLOGIES USED

4.1 JAVA TECHNOLOGY

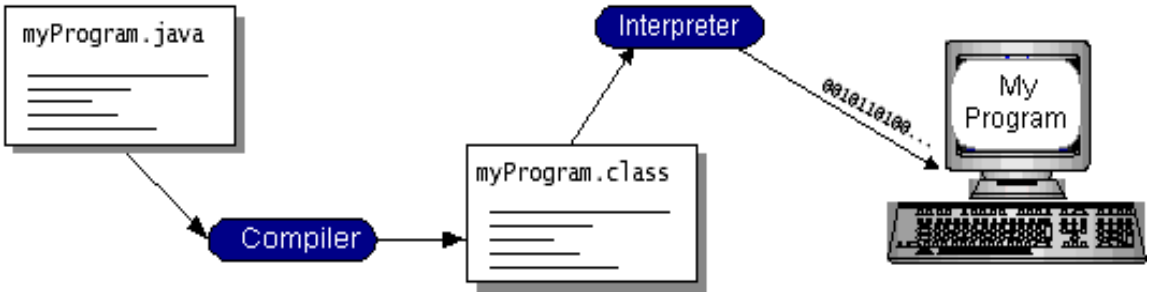
Java technology is both a programming language and a platform.

4.1.1 THE JAVA PROGRAMMING LANGUAGE

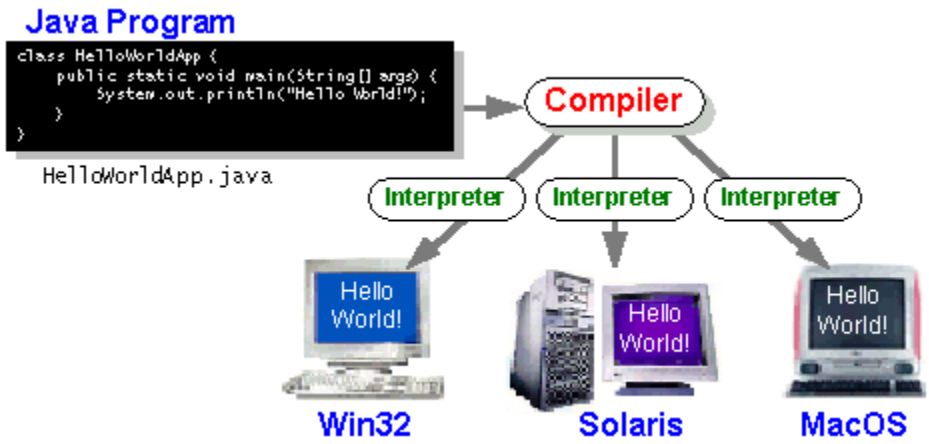
The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- 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.



4.1.2 THE JAVA PLATFORM

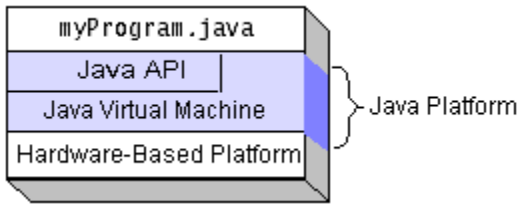
A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most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

4.1.3 WHAT CAN JAVA TECHNOLOGY DO?

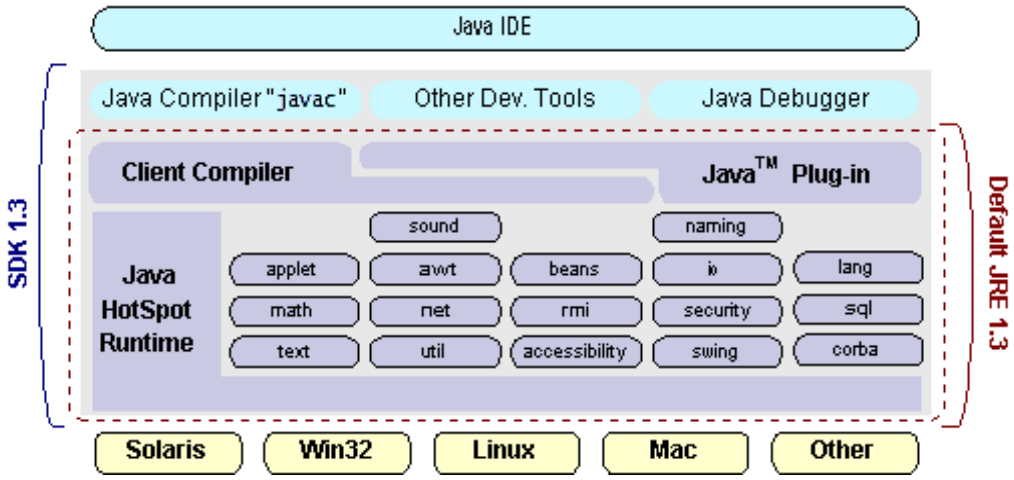
The most common types of programs written in the Java programming language are applets and applications. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser. However, the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



4.1.4 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.

4.2 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.

4.3 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.

4.3.1 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.

2. **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.

3. **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.

4. **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.

5. **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.

6. **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.

7. **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.

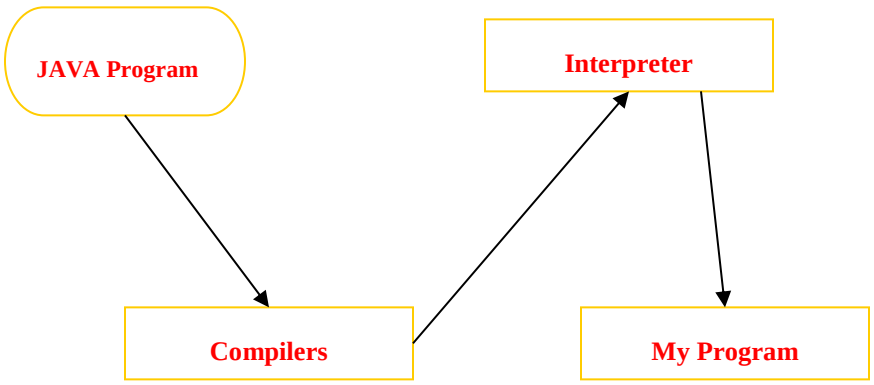
Java has two things: a programming language and a platform.

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.



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.

4.4 JAVA SERVER PAGES (JSP):

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types.

To deploy and run Java Server Pages, a compatible web server with a servlet container, such as Apache Tomcat is required. In our project we link java with jsp and we deploy it in apache tomcat server.

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. This tutorial will teach you how to use Java Server Pages to develop your web applications in simple and easy steps.

Java Server Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A Java Server Pages component is a type of Java servlet that is designed to fulfil the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages etc.

SYSTEM ARCHITECTURE DESIGN

5. SYSTEM ARCHITECTURE DESIGN

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

5.1 SYSTEM ARCHITECTURE FOR DS-PEKS:

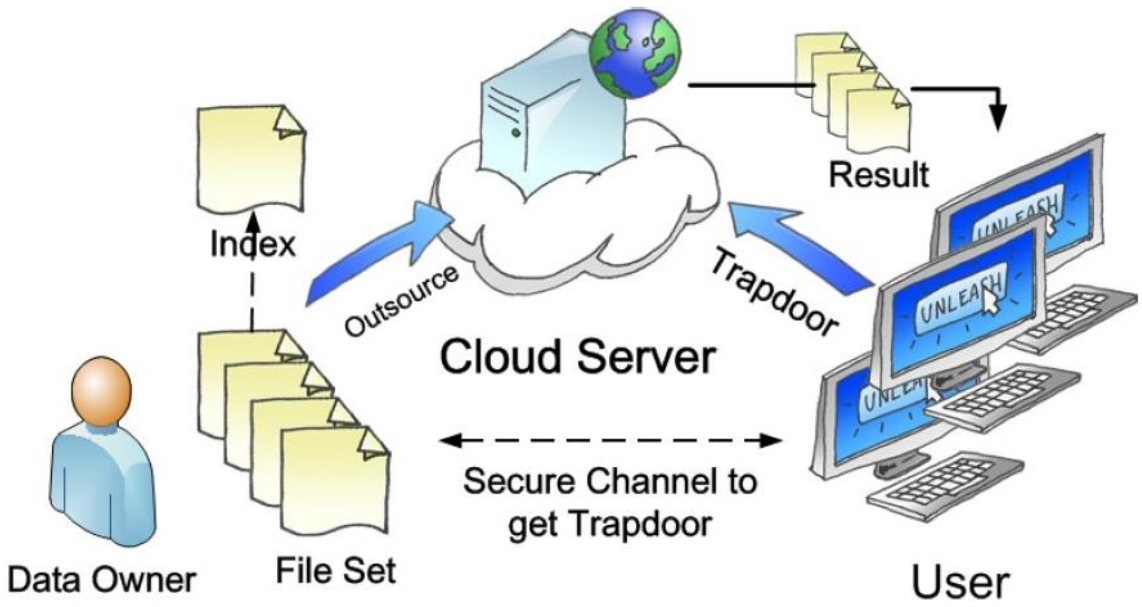


FIGURE 1: BLOCK DIAGRAM

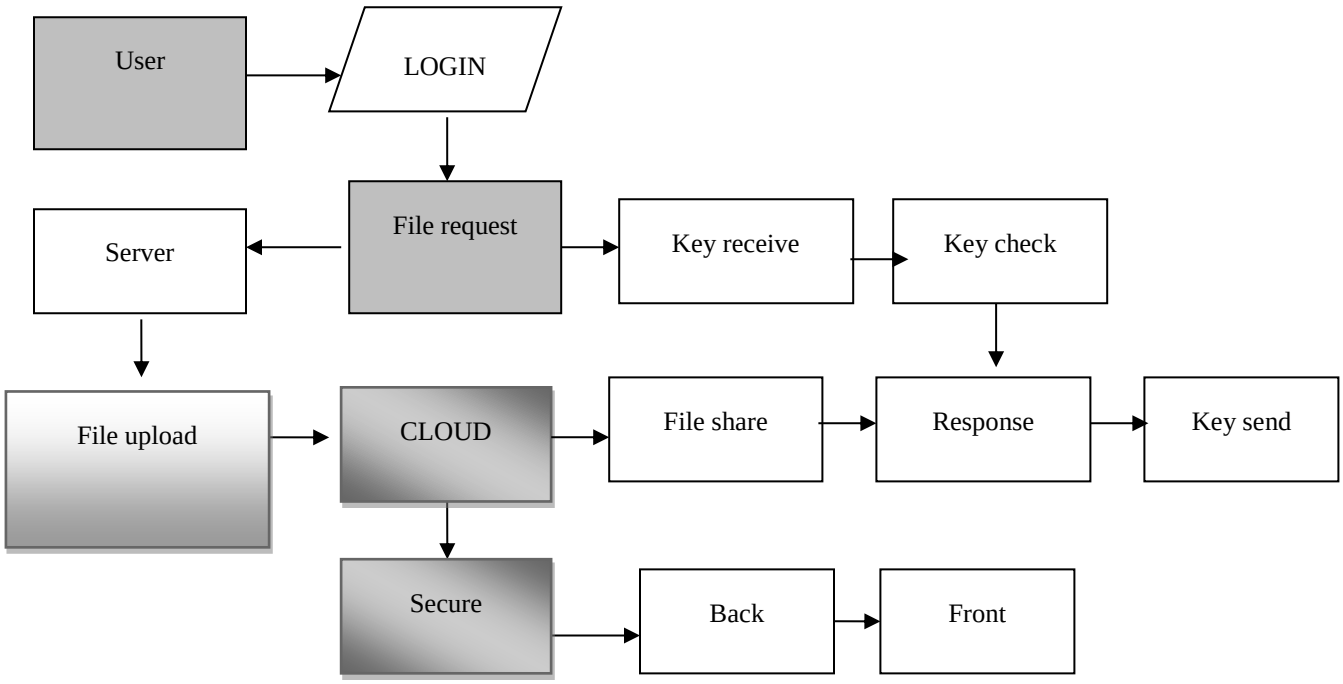
5.2 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



5.3 UML:

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system.

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

5.3.1 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 meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of OO tools market.

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

7. Integrate best practices.

5.4 UML DIAGRAMS:

5.4.1Use Case Diagram:

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.

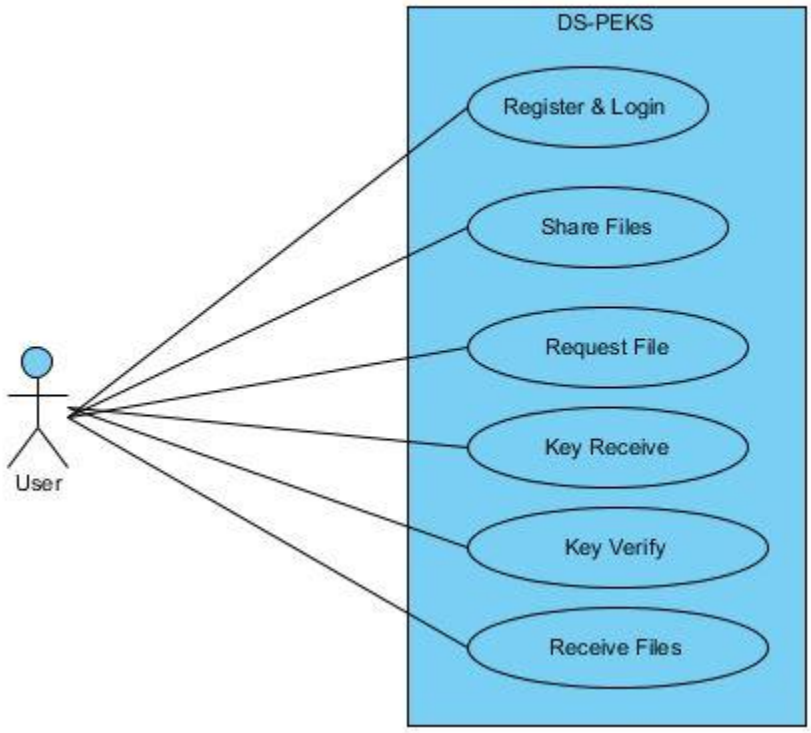


FIGURE 2: USE CASE DIAGRAM FOR USER

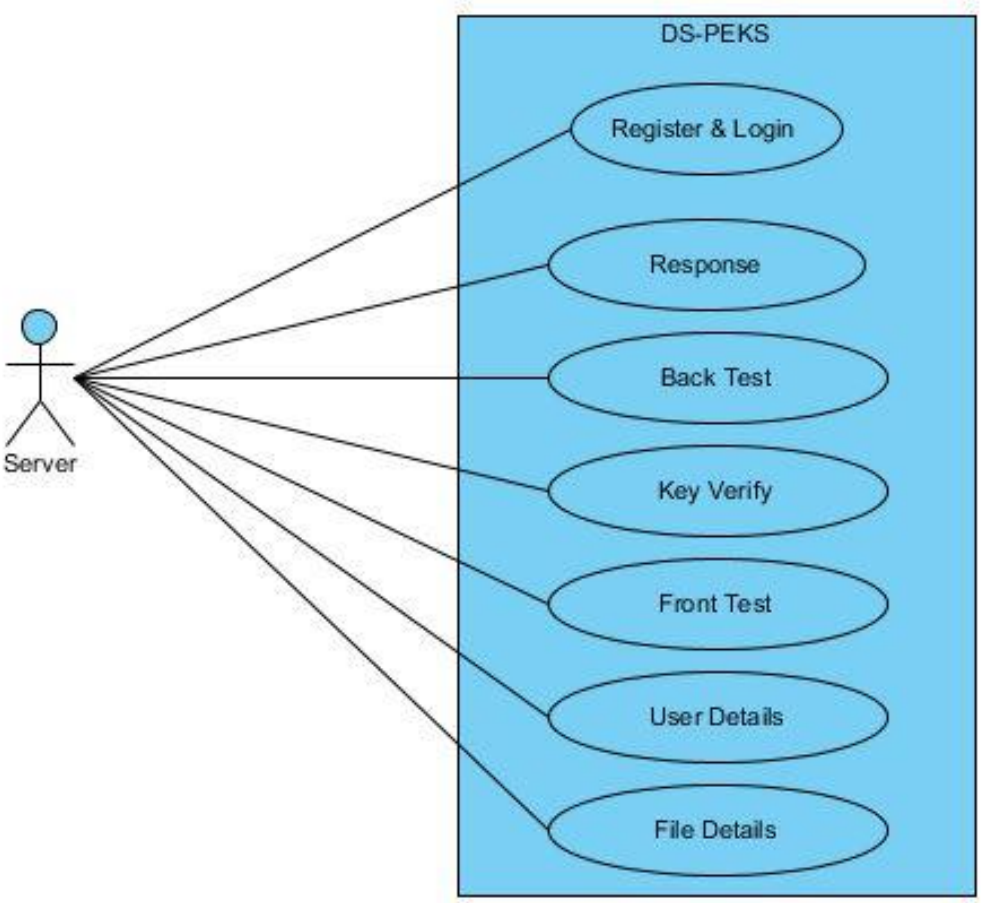


FIGURE 3: USE CASE DIAGRAM FOR SERVER

5.4.2 Class Diagram:

Class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. The Class diagram is one of the most widely used diagrams from the UML specification.

In this diagram we need to explain about the operations that are performed by user and Server

FIGURE 4: CLASS DIAGRAM

5.4.3 Object Diagram:

An object diagram shows a set of objects and their relationships. Object diagrams represent static snapshots of instances of the things found in class diagrams. These diagrams address the static design view or static process view of a system as do class diagrams, but from the perspective of real or prototypical cases.

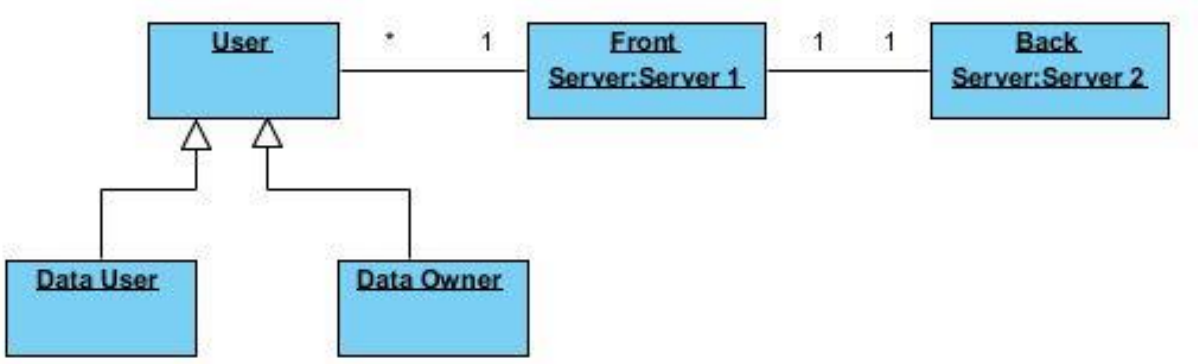


FIGURE 5: OBJECT DIAGRAM

5.4.4 Sequence Diagram:

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

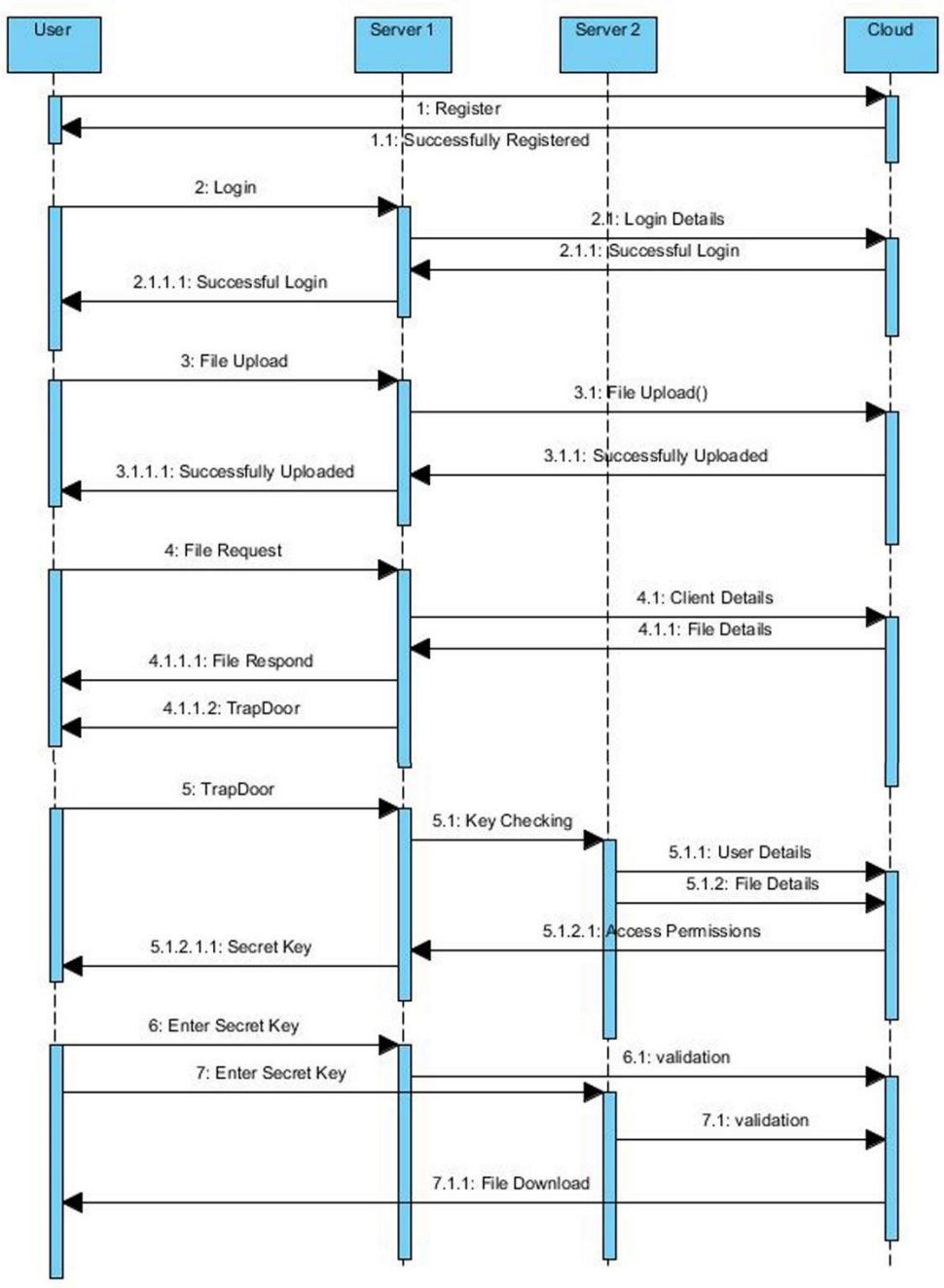


FIGURE 6: SEQUENCE DIAGRAM

5.4.5 Collaboration Diagram:

A collaboration diagram, also called a communication diagram or interaction diagram. A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time

They show the relationship between objects and the order of messages passed between them. The objects are listed as icons and arrows indicate the messages being passed between them. The numbers next to the messages are called sequence numbers. As the name suggests, they show the sequence of the messages as they are passed between the objects.

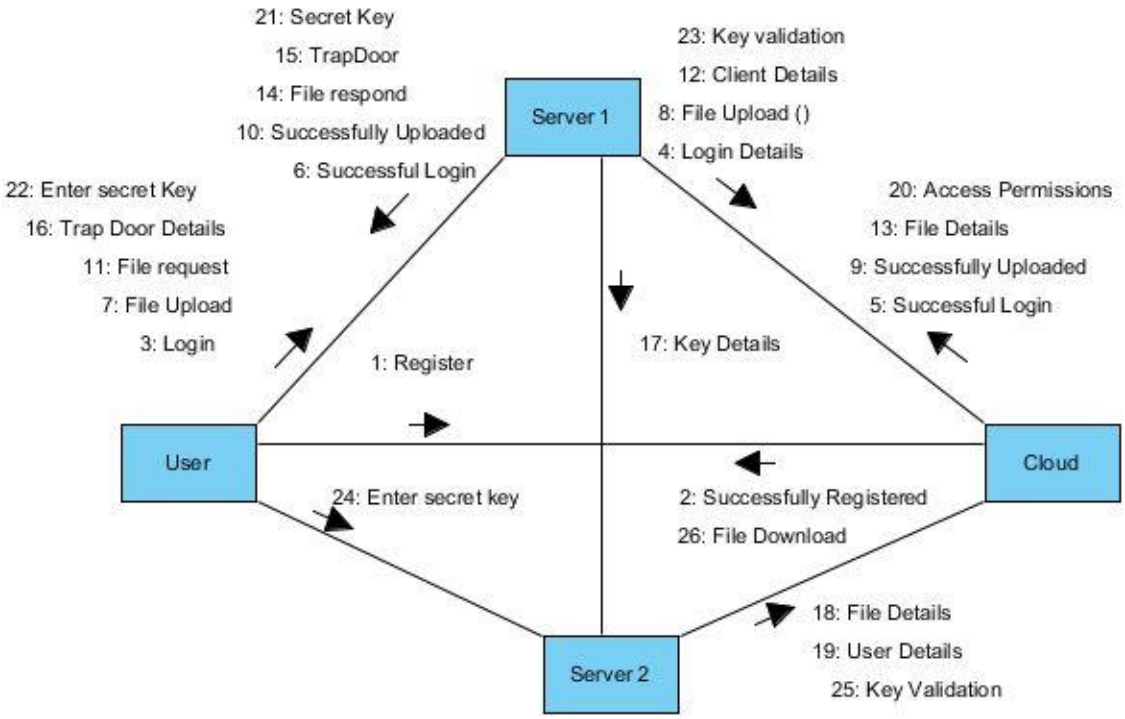


FIGURE 7: COLLABORATION DIAGRAM

5.4.6 Activity Diagram:

Activity diagrams are used to document workflows in a system, from the business level down to the operational level. The general purpose of Activity diagrams is to focus on flows driven by internal processing vs. external events.

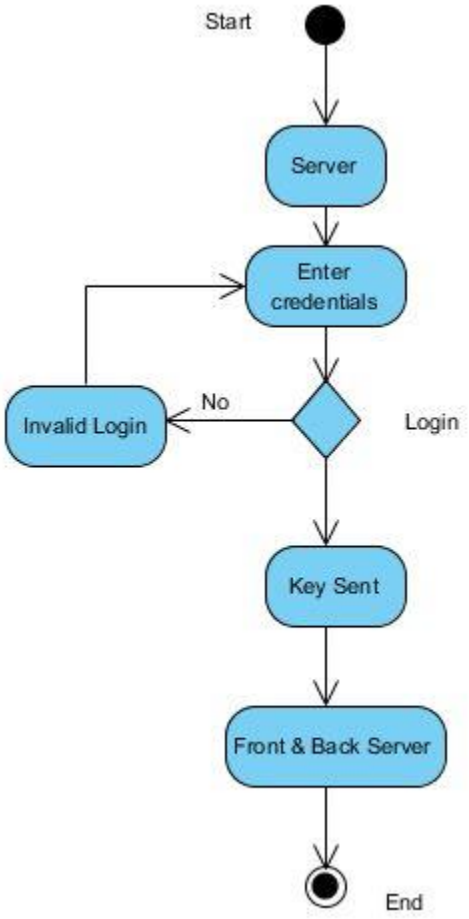


FIGURE 8: ACTIVITY DIAGRAM FOR SERVER

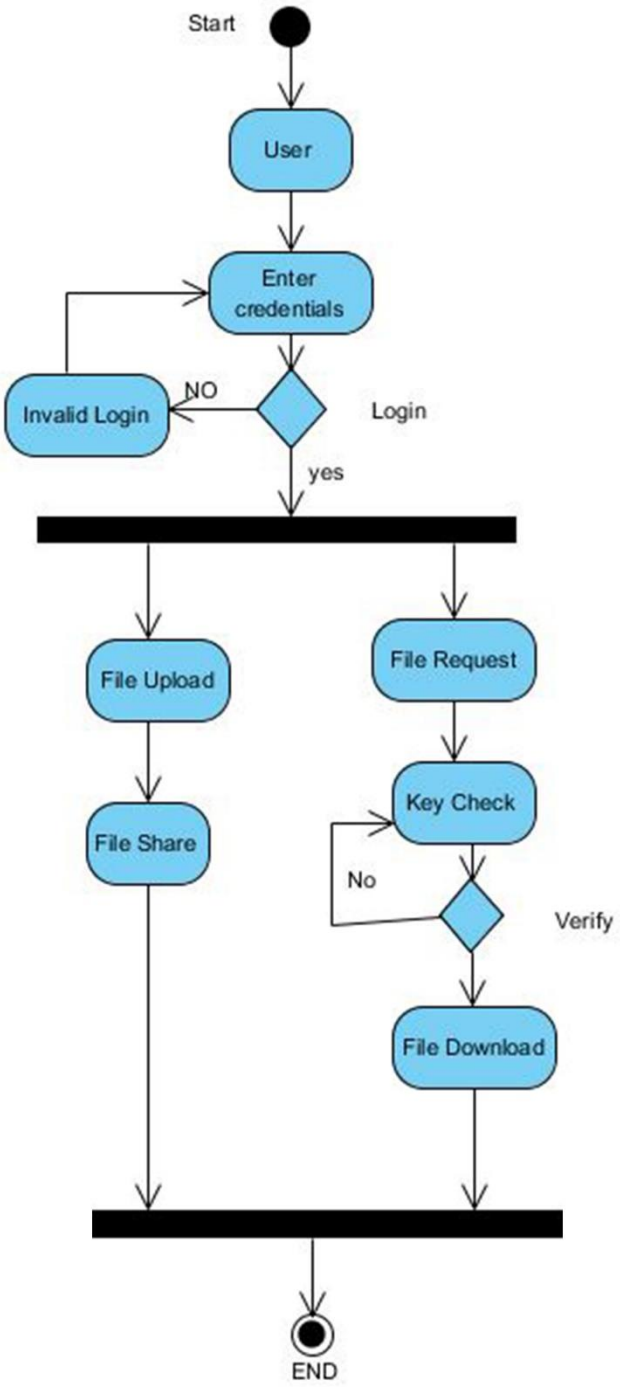


FIGURE 9: ACTIVITY DIAGRAM FOR THE USER

5.4.7 State Chart Diagram:

A state chart diagram shows a state machine, consisting of states, transitions, events, and activities. State chart diagrams address the dynamic view of a system. They are especially important in modeling the behavior of an interface, class, or collaboration and emphasize the event-ordered behavior of an object, which is especially useful in modeling reactive systems.

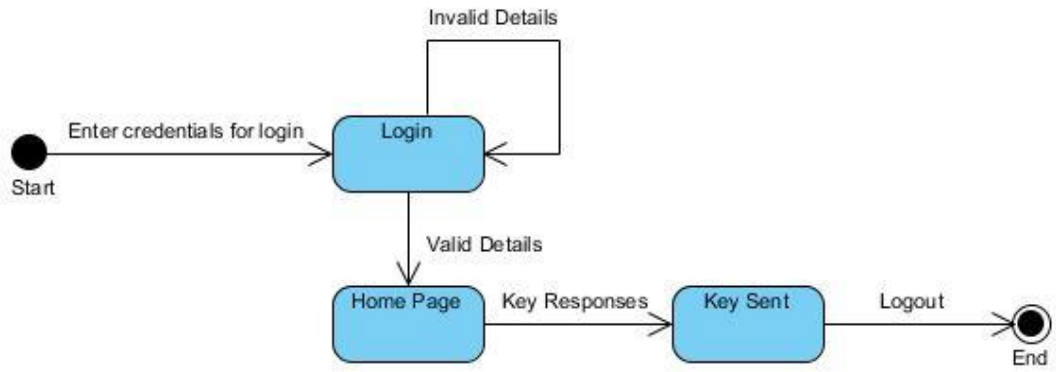


FIGURE 10: STATE CHART DIAGRAM FOR SERVER

In this diagrams, it shows the different states of the application and how this states changes based on the events occurring.

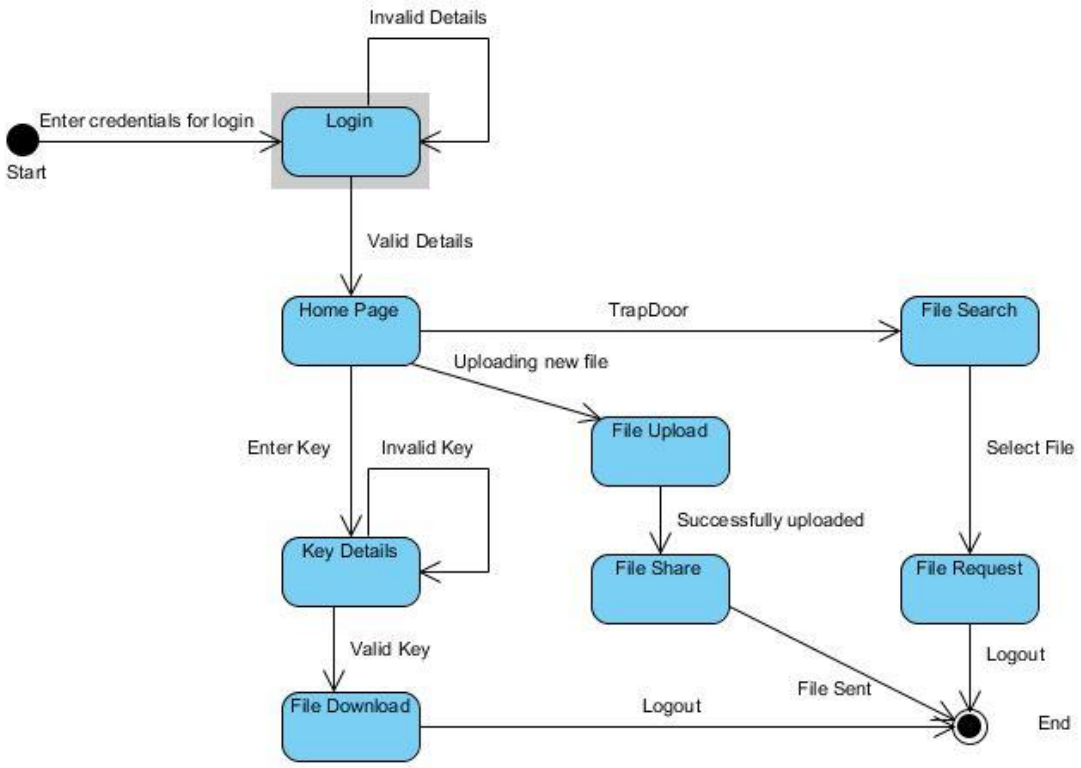


FIGURE 11: STATE CHART DIAGRAM FOR USER

5.4.8 Component Diagram:

The component diagram shows the total modules of the project and all the library files associated with it. In each component we describe all the functions. Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

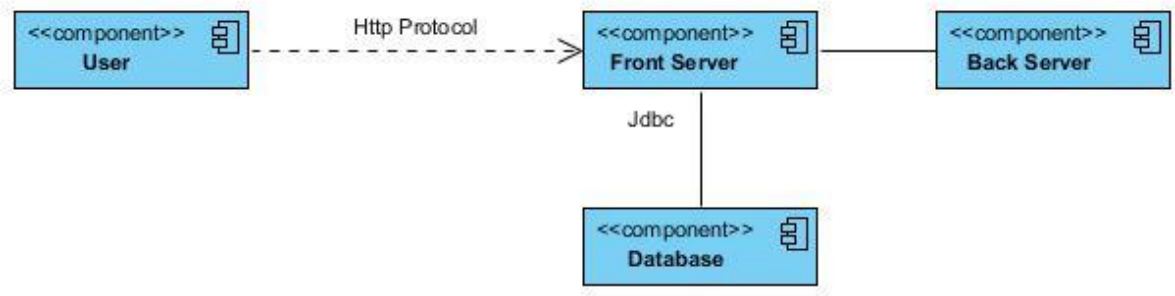


FIGURE 12: COMPONENT DIAGRAM

5.4.9 Deployment Diagram:

In this diagram it explains the different parts in which many clients can be connected to a server in which there is a web container (jasper and Catalina) and are internally connected to a database.

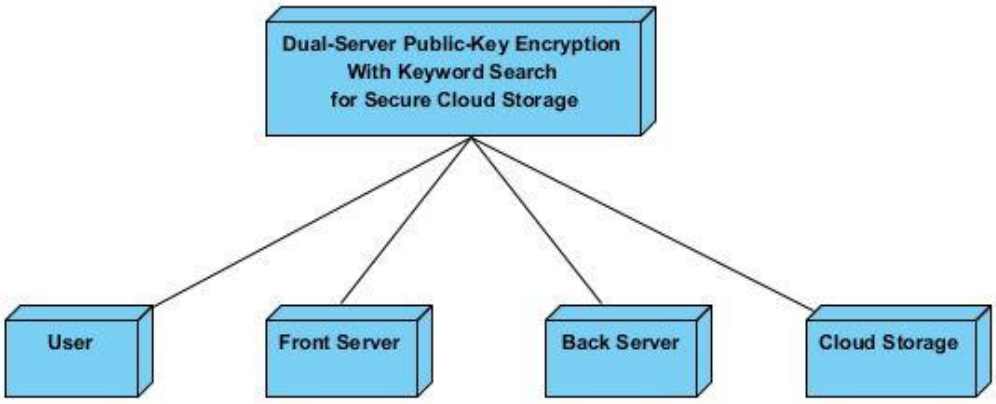


FIGURE 13: DEPLOYEMENT DIAGRAM

TESTING

6. TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

6.1 TESTING OBJECTIVES:

- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation
- To make sure that during the operation, incorrect input, processing and output will be detected
- To see that when correct inputs are fed to the system the outputs are correct
- To verify that the controls incorporated in the same system as intended
- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

6.2 TEST CASE DESIGN:

6.2.1 White box testing:

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

6.2.2 Black Box Testing:

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

6.2.3 Unit Testing:

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

6.2.4 Integration Testing:

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

6.2.5 Validation Testing:

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

6.2.6 System Testing:

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves:

- Implementing the system in a simulated production environment and testing it.
- Introducing errors and testing for error handling.

6.3 TEST CASES:

Test case	Check Item	Test Case Objective	Test Data/Input	Excepted Result	Actual Result	Results
TC-001	Registration Page	Leaving all fields Empty		By leaving the text field's empty and clicking Submit it should ask for the values	Getting Error message from JSP	FAIL
TC-002	Registration Page	Leaving all fields Empty		By leaving the text field's empty and clicking Submit it should ask for the values	By leaving the text field's empty and clicking Register it should ask for the values	PASS
TC-003	Registration Page	Leaving all fields Empty		By leaving the text field's empty and clicking Submit it should ask for the values	Creating Entry into the Database	FAIL
TC-004	Registration Page	Leaving all fields Empty		By leaving the text field's empty and clicking Submit it should ask for the values	By leaving the text field's empty and clicking Submit it should ask for the values	FAIL
TC-005	Registration Page	PhoneNo Field	PhoneNo: ABCDEF	By entering characters into fields and upon submit it should display as "Invalid Phone No"	It is allowing the users to get registered	FAIL

TC-006	**Registration Page**	**PhoneNo Field**	**PhoneNo: ABCDEF**	**By entering characters into fields and upon submit it should display as "Invalid Phone No"**	**By entering characters into fields and upon submit it should display as "Invalid Phone No"**	**PASS**
TC-007	**USER Login Page**	**Enter Invalid Username & Password**		**It should not allow User to Login and display as "Invalid Username/Password"**	**Not Allowing User to Login**	**PASS**
TC-008	**Password**			**The Password field should display the encrypted format of the text typed as (****)**	**Entered Text is displaying in text field**	**FAIL**
TC-009	**Password**			**The Password field should display the encrypted format of the text typed as (****)**	**Entered Text is displaying as encrypted format (****) in text field**	**PASS**
TC-010	**File Share Page**	**Uploading file to cloud**	**Select the file to upload**	**It should upload the file to the Cloud**	**File does not uploaded to the Cloud**	**FAIL**

TC-011	**File Share Page**	**Uploading file to cloud**	**Select the file to upload**	**It should upload the file to the Cloud**	**File uploaded to the Cloud**	**PASS**
TC-012	**File Share Page**	**Uploading file to cloud**	**Select the file to upload**	**File Should be Encrypted before uploading to the Cloud**	**File is not encrypted**	**FAIL**
TC-013	**File Share Page**	**Uploading file to cloud**	**Select the file to upload**	**File Should be Encrypted before uploading to the Cloud**	**File encrypted before uploading to the Cloud**	**PASS**
TC-014	**Search File Page**	**Entering Wrong keyword**	**Keyword: xYbshanX**	**By Entering wrong keyword it should not display any results**	**It does not displayed any results**	**PASS**
TC-015	**Search File Page**	**Entering Encrypted keyword**	**Keyword: GLhsouZ9GKb4hjixPuCQaWYly**	**By Entering encrypted keyword it should display the results**	**It does not displayed any results**	**FAIL**
TC-016	**Search File Page**	**Entering Encrypted keyword**	**Keyword: GLhsouZ9GKb4hjixPuCQaWYly**	**By Entering encrypted keyword it should display the results**	**By Entering encrypted keyword it should display the results**	**PASS**
TC-017	**Search File Page**	**Clicking Send Request button**		**By clicking request button it should send key request to the Servers**	**Key Request has been sent to the Servers**	**PASS**

TC-018	**Response Page Table**	**Clicking Send Response button**		**By clicking send button it should send secret key to the user Email id**	**No Email was sent to the user mail id**	**FAIL**
TC-019	**Response Page Table**	**Clicking Send Response button**		**By clicking send button it should send secret key to the user Email id**	**Secret key successfully sent to the user mail**	**PASS**
TC-020	**File Download Page**	**Secret Key Fields**	**Server1 Key :OQpwMU ZxbjelkBj Server2 Key :yBXmyiCg dAcrAZb**	**By Entering Secret Keys and clicking verify it should display the file to be download**	**By Entering Secret Keys and clicking verify it should display the file to be download**	**PASS**
TC-021	**File Download Page**	**Download Button**		**By clicking download button it should download the file in decrypted format**	**It downloaded the file in encrypted format**	**FAIL**
TC-021	**File Download Page**	**Download Button**		**By clicking download button it should download the file in decrypted format**	**By clicking download button it should download the file in decrypted format**	**PASS**

SCREEN SHOTS

7. SCREEN SHOTS

FIGURE 14: HOME PAGE SCREEN SHOT

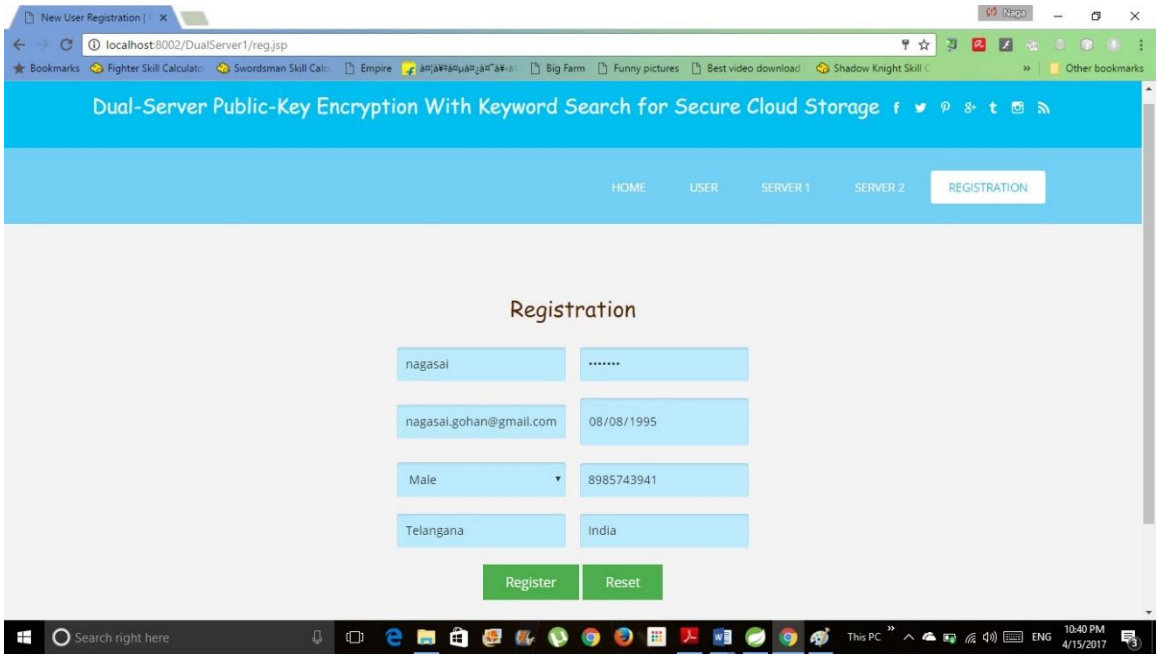


FIGURE 15: REGISTRATION PAGE SCREEN SHOT

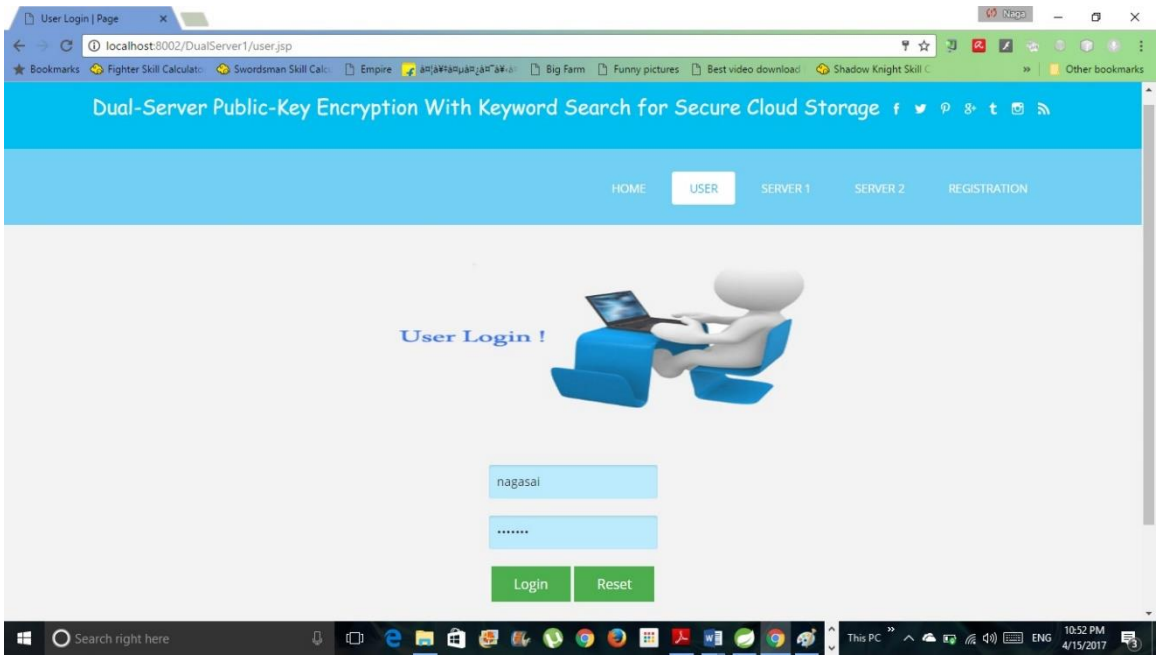


FIGURE 16: USER LOGIN PAGE SCREEN SHOT

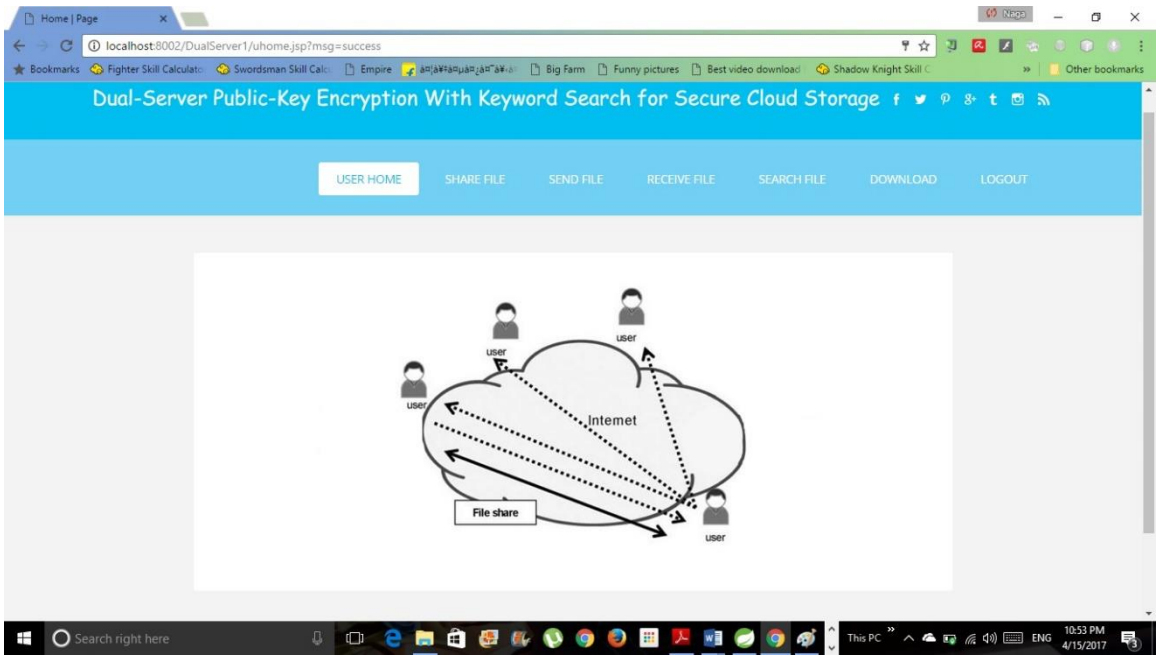


FIGURE 17: USER HOME PAGE SCREEN SHOT

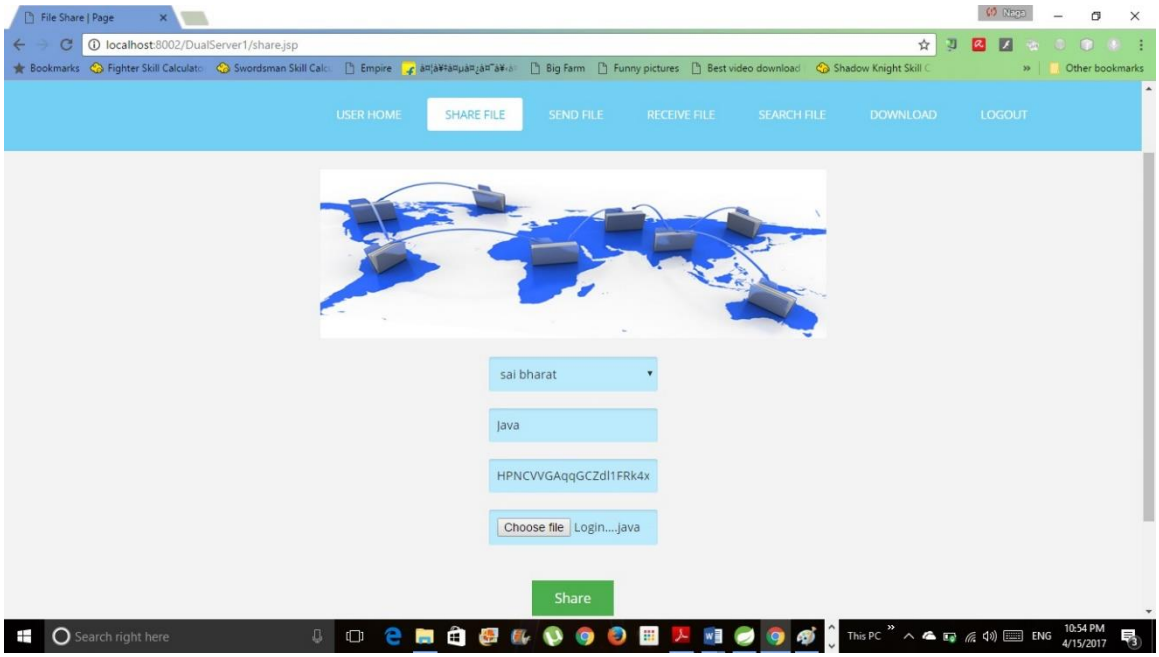


FIGURE 18: FILE SHARE PAGE SCREEN SHOT

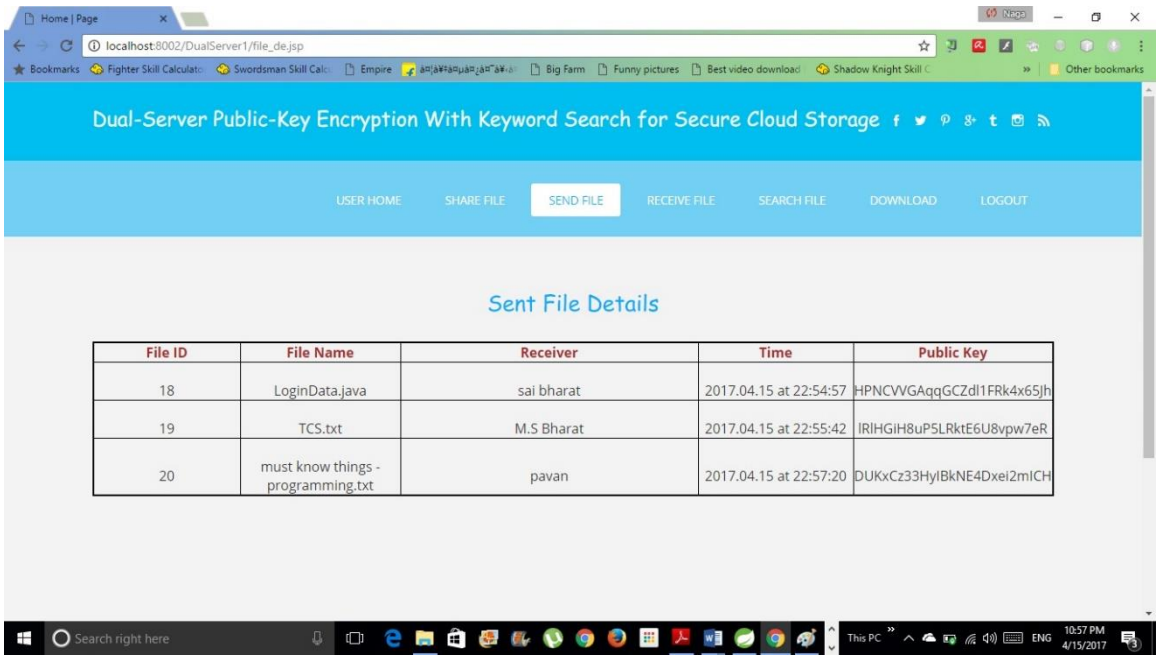


FIGURE 19: SENT FILE DETAILS PAGE SCREEN SHOT

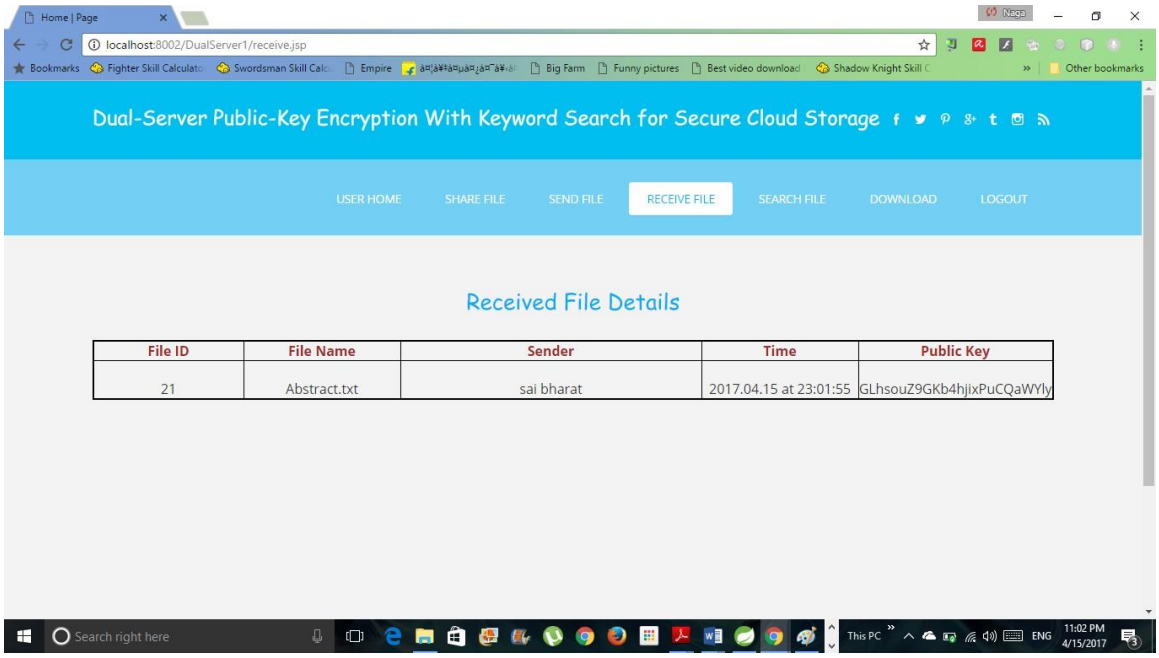


FIGURE 20: RECEIVED FILE DETAILS PAGE SCREEN SHOT

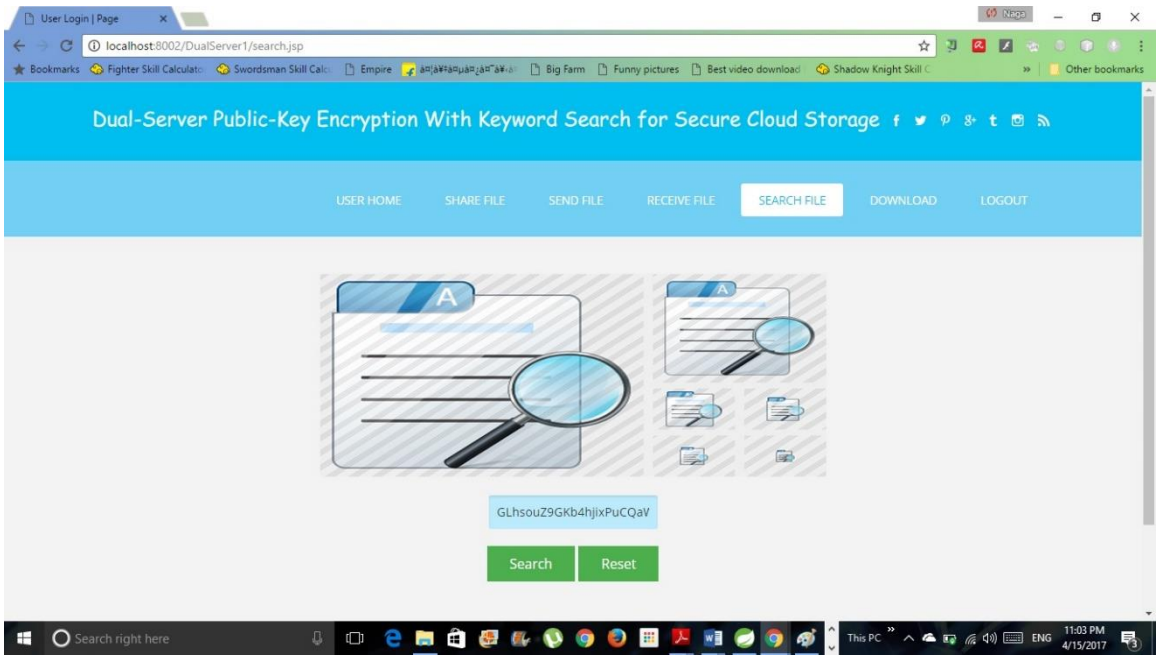


FIGURE 21: SEARCH FILE PAGE SCREEN SHOT

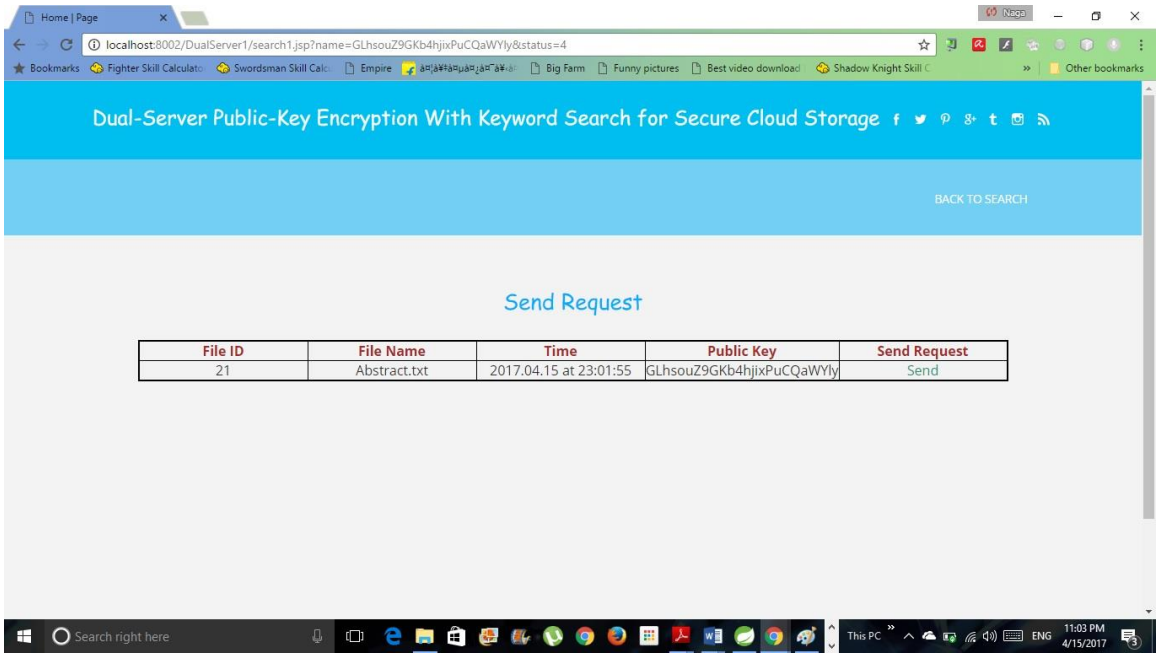


FIGURE 22: SEARCH RESULT PAGE SCREEN SHOT

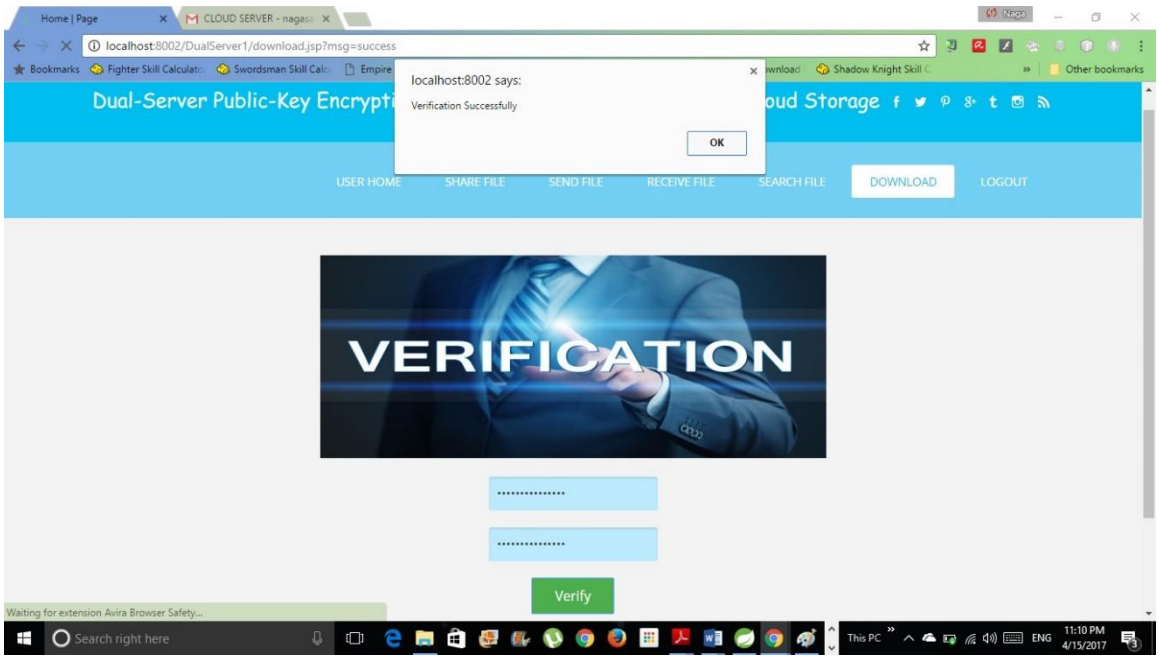


FIGURE 23: DOWNLOAD PAGE SCREEN SHOT

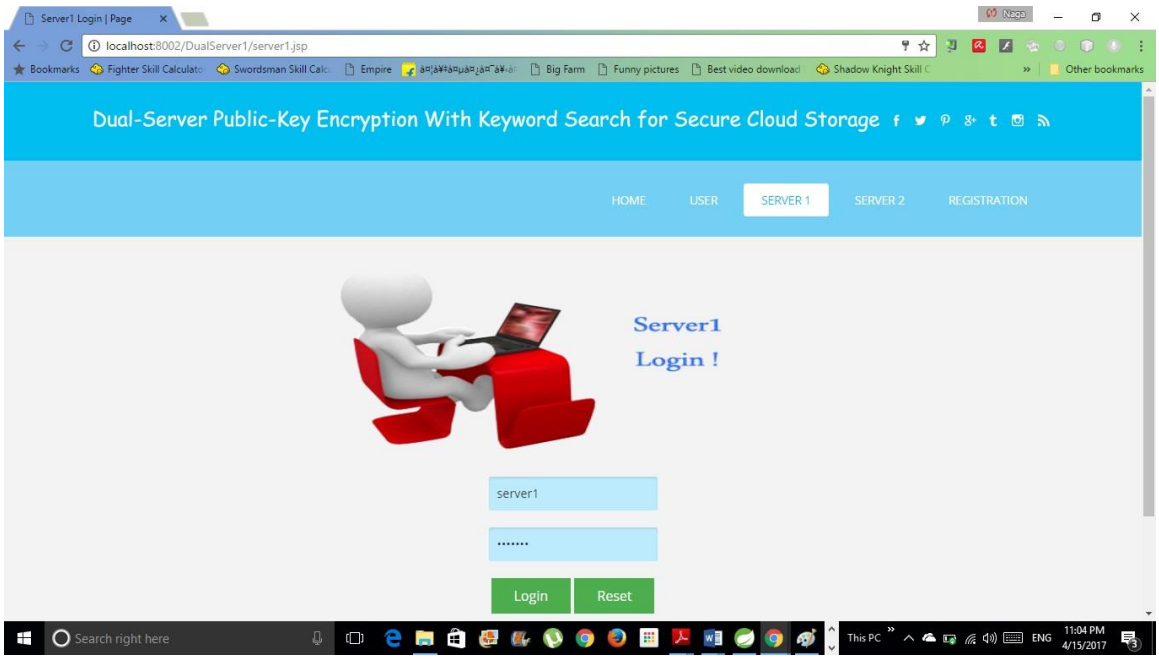


FIGURE 24: SERVER1 LOGIN PAGE SCREEN SHOT

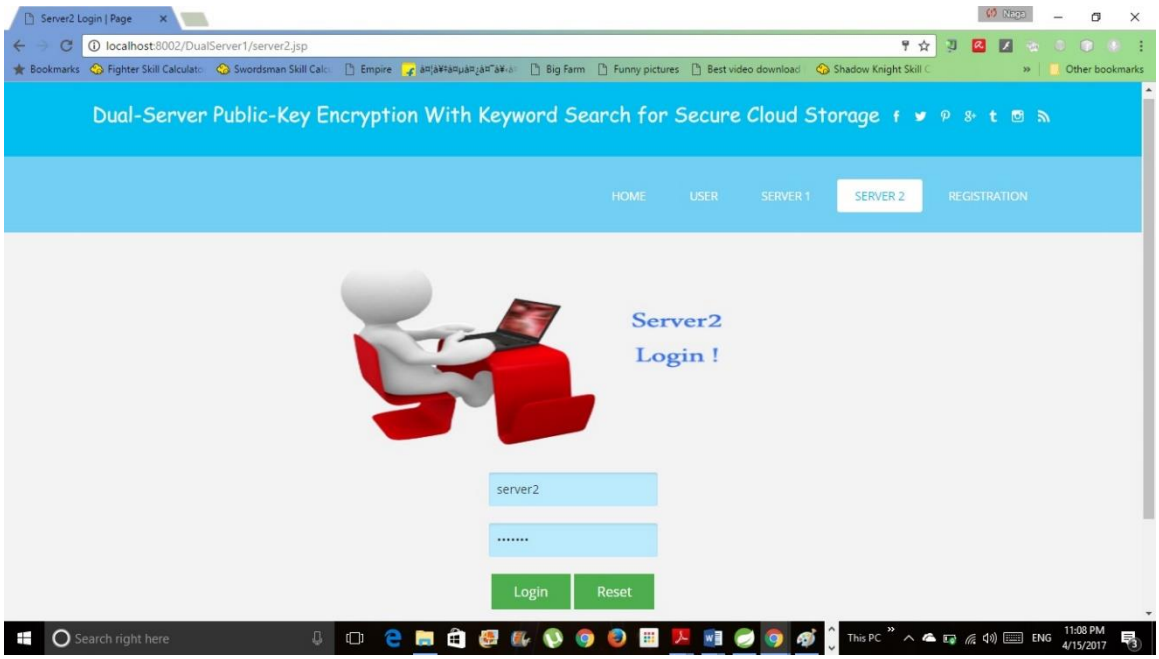


FIGURE 25: SERVER2 LOGIN PAGE SCREEN SHOT

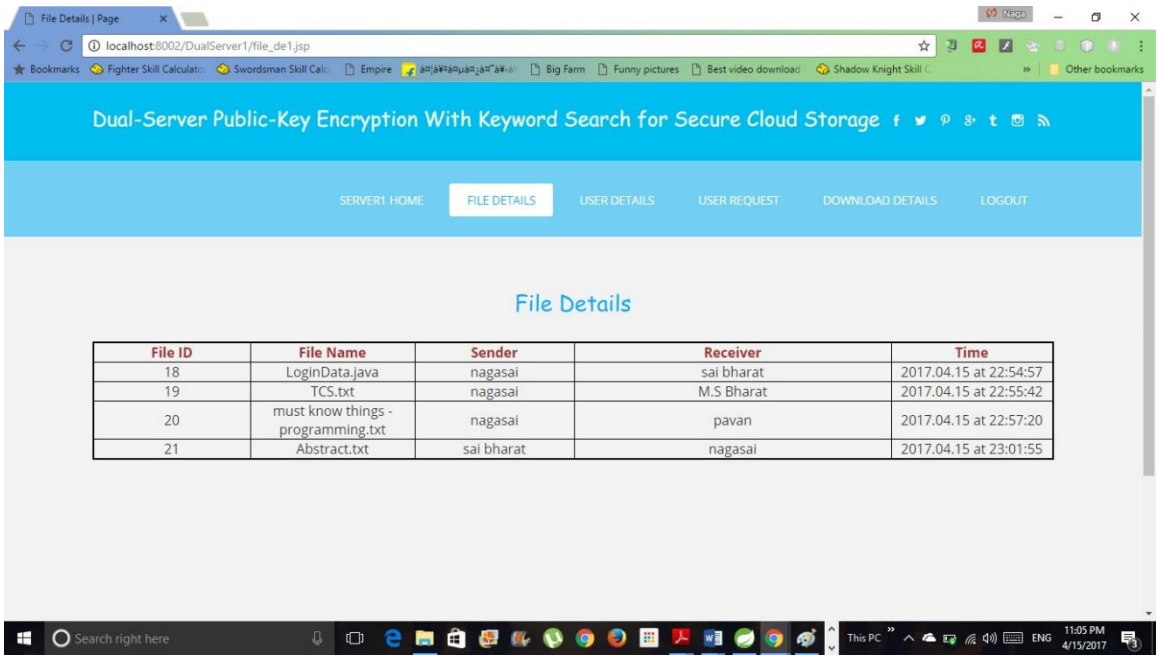


FIGURE 26: FILE DETAILS PAGE SCREEN SHOT

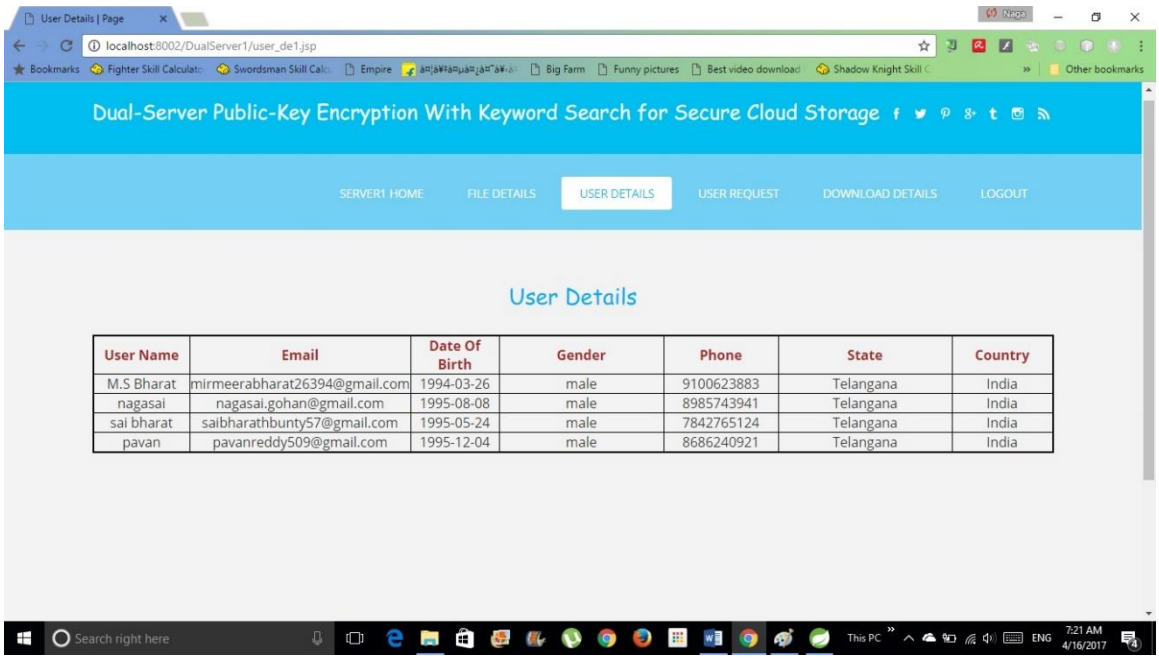


FIGURE 27: USER DETAILS PAGE SCREEN SHOT

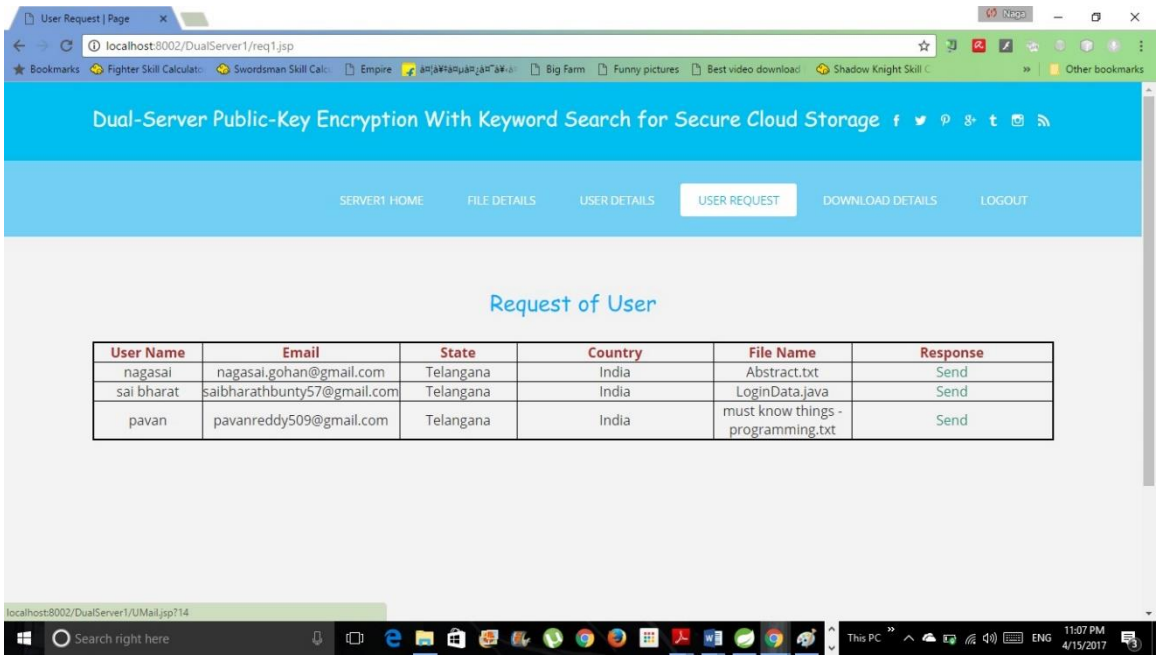


FIGURE 28: USER REQUEST PAGE SCREEN SHOT

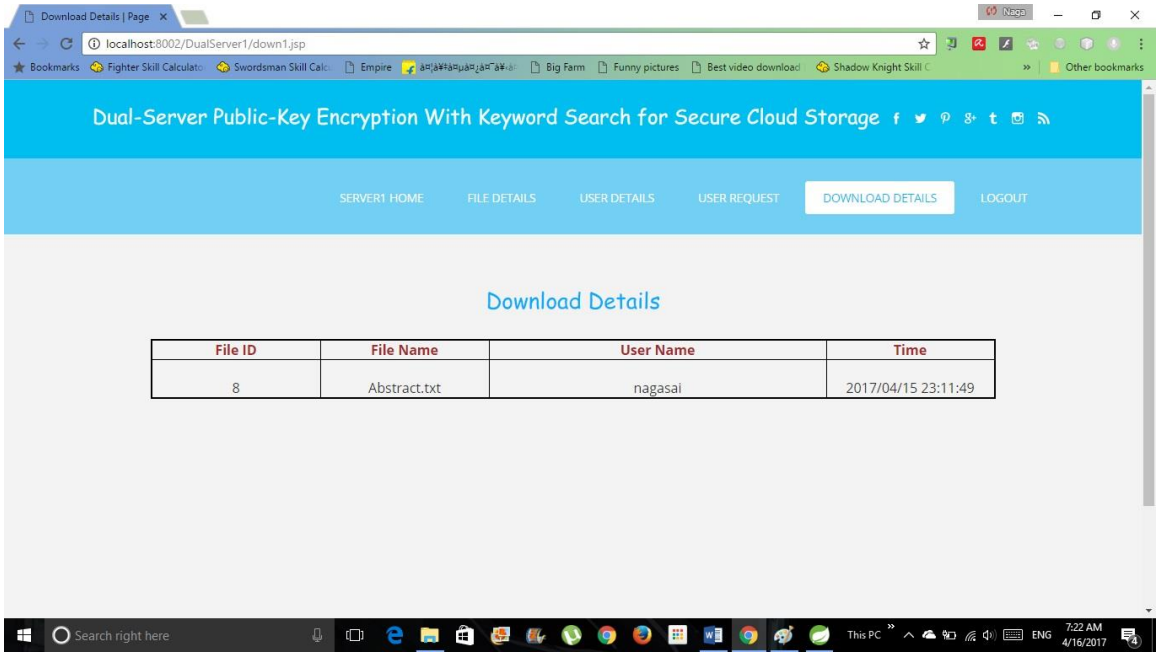


FIGURE 29: DOWNLOAD DETAILS PAGE SCREEN SHOT

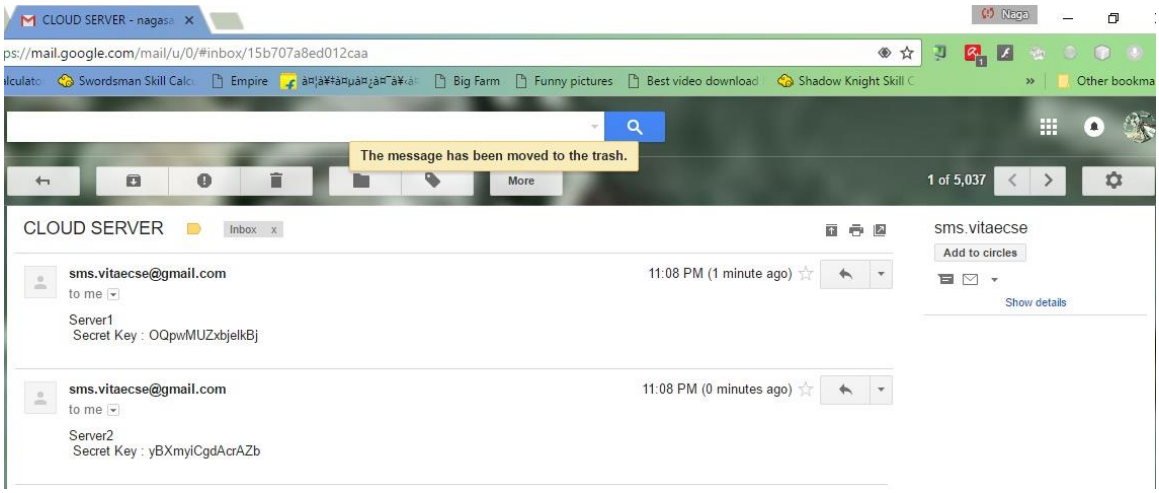


FIGURE 30: SECRET KEY SCREEN SHOT

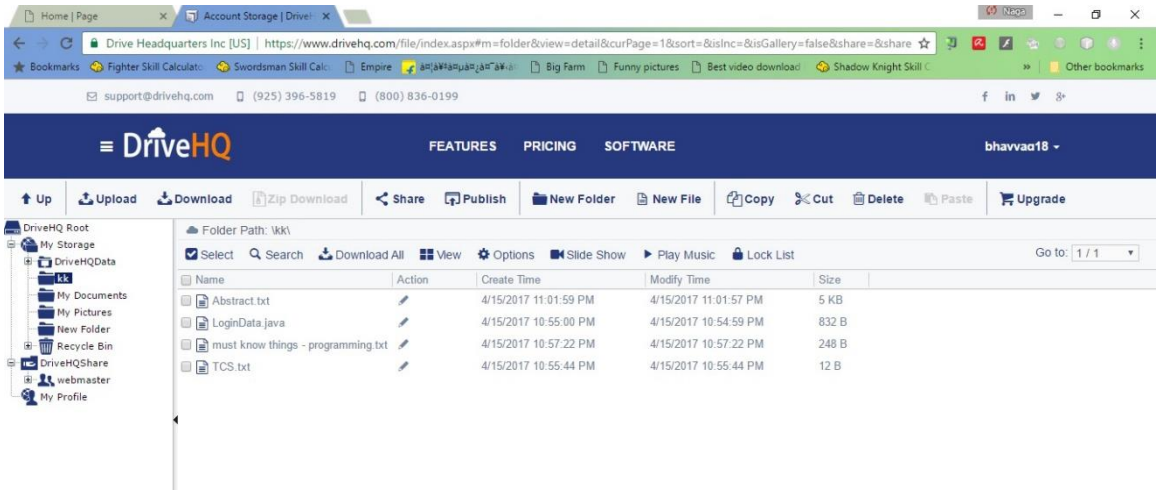


FIGURE 31: CLOUD STORAGE SCREEN SHOT

CONCLUSION

8. CONCLUSION

we proposed a new framework, named Dual-Server Public Key Encryption with Keyword Search (DS-PEKS), that can prevent the inside keyword guessing attack which is an inherent vulnerability of the traditional PEKS framework. We also introduced a new Smooth Projective Hash Function (SPHF) and used it to construct a generic DS-PEKS scheme. An efficient instantiation of the new SPHF based on the Diffie-Hellman problem is also presented in the paper, which gives an efficient DS-PEKS scheme without pairings.

BIBILIOGRAPHY

9. BIBILIOGRAPHY

[1] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in *Proc. 20th Australasian Conf. Inf. Secur. Privacy (ACISP)*, 2015, pp. 59–76.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, May 2000, pp. 44–55.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2004, pp. 563–574.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, 2006, pp. 79–88.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. EUROCRYPT*, 2004, pp. 506–522.

[6] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in *Proc. Int. Conf. EUROCRYPT*, 2003, pp. 524–543.

[7] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, "Building an encrypted and searchable audit log," in *Proc. NDSS*, 2004, pp. 1–11.

[8] M. Abdalla *et al.*, "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," in *Proc. 25th Annu. Int. Conf. CRYPTO*, 2005, pp. 205–222.

[9] D. Khader, "Public key encryption with keyword search based on K-resilient IBE," in *Proc. Int. Conf. Comput. Sci. Appl. (ICCSA)*, 2006, pp. 298–308.

[10] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," *IEEE Trans. Comput.*, vol. 62, no. 11, pp. 2266–2277, Nov. 2013.

[11] G. Di Crescenzo and V. Saraswat, "Public key encryption with searchable keywords based on Jacobi symbols," in *Proc. 8th Int. Conf. INDOCRYPT*, 2007, pp. 282–296.

[12] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Cryptography and Coding*. Cirencester, U.K.: Springer, 2001, pp. 360–363.

[13] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. Int. Conf. Comput. Sci. Appl. (ICCSA)*, 2008, pp. 1249–1259.

[14] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in *Proc. 4th Int. Symp. ASIACCS*, 2009, pp. 376–379.

[15] K. Emura, A. Miyaji, M. S. Rahman, and K. Omote, "Generic constructions of secure-channel free searchable encryption with adaptive security," *Secur. Commun. Netw.*, vol. 8, no. 8, pp. 1547–1560, 2015.

[16] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Proc. 3rd VLDB Workshop Secure Data Manage. (SDM)*, 2006, pp. 75–83.

[17] W.-C. Yau, S.-H. Heng, and B.-M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in *Proc. 5th Int. Conf. ATC*, 2008, pp. 100–105.

[18] J. Baek, R. Safavi-Naini, and W. Susilo, "On the integration of public key data encryption and public key encryption with keyword search," in *Proc. 9th Int. Conf. Inf. Secur. (ISC)*, 2006, pp. 217–232.

[19] H. S. Rhee, W. Susilo, and H.-J. Kim, "Secure searchable public key encryption scheme against keyword guessing attacks," *IEICE Electron. Exp.*, vol. 6, no. 5, pp. 237–243, 2009.

[20] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *J. Syst. Softw.*, vol. 83, no. 5, pp. 763–771, 2010.

[21] L. Fang, W. Susilo, C. Ge, and J. Wang, "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *Inf. Sci.*, vol. 238, pp. 221–241, Jul. 2013.

[22] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, "Constructing PEKS schemes secure against keyword guessing attacks is possible?" *Comput. Commun.*, vol. 32, no. 2, pp. 394–396, 2009.

[23] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Proc. Int. Conf. EUROCRYPT*, 2002, pp. 45–64.