**A**

**PROJECT REPORT**

**ON**

**SECURE FILE STORAGE IN CLOUD COMPUTING USING HYBRID CRYPTOGRAPHY ALGORITHM**

**AT**

**CONSCIENCE TECHNOLOGIES**

**IN PARTIAL FULFILLMENT FOR THE AWARD OF**

**MASTER OF COMPUTER APPLICATION**

**SUBMITTED BY**

**R. RAHUL**

**(150920862065)**

**UNDER THE GUIDANCE OF**

**Mr. Dr. B. BHUJANGA REDDY**

**PROFESSOR**



**MANPOWER DEVELOPMENT COLLEGE**

**ECIL ‘X’ ROAD, S.P NAGAR, MOULA-ALI, SECUNDERABAD**

**(AFFILIATED TO OSMANIA UNIVERSITY)**

**2019-2022**

**DECLARATION**

I, the undersigned hereby declare that the project report titled “**SECURE FILE STORAGE IN CLOUD COMPUTING USING HYBRID CRYPTOGRAPHY ALGORITHM**” carried out at **CONSCIENCE TECHNOLOGIES** is my original work written and submitted by me in partial fulfillment of Master Degree in Computer Application of Osmania University, Hyderabad.

I also declare that this project has not been submitted earlier in any other University or Institution.

**Date: R. RAHUL**

**ACKNOWLEDGEMENT**

I take this opportunity to extend my profound thanks and deep sense of gratitude to the authorities of **CONSCIENCE TECHNOLOGIES** for giving me the opportunity to undertake this project work in their organization.

My sincere thanks to **Mrs. Dr. V. N. SANTHISREE**, Principal of Manpower Development College for her cooperation.

My sincere thanks to **Mr. Dr. B. BHUJANGA REDDY**, Head of the Department, MCA of Manpower Development College.

I also thanks to My Project Guide **Mr. Dr. A. JAKULIN SAFIA**, Professor for his support and cooperation in completing this project work.

Lastly, I take this opportunity to thank my parents and other family members for their whole hearted support during my project work.

**R. RAHUL**

**ABSTRACT**

Now a day’s cloud computing is used in many areas like industry, military colleges etc to storing huge amount of data. We can retrieve data from cloud on request of user. To store data on cloud we have to face many issues .To provides the solution to these issues there are n number of ways. Cryptography and steganography techniques are more popular now a day’s for data security. Use of a single algorithm is not effective for high level security to data in cloud computing. In this paper we have introduced new security mechanism using symmetric key cryptography algorithm and steganography .In this proposed system AES, blowfish, RC6 and BRA algorithms are used to provide block wise security to data. All algorithm key size is 128 bit.LSB steganography technique is introduced for key information security. Key information contains which part of file is encrypted using by which algorithm and key .File is splited into eight parts. Each and every part of file is encrypted using different algorithm. All parts of file are encrypted simultaneously with the help of multithreading technique. Data encryption Keys are inserted into cover image using LSB technique. Stego image is send to valid receiver using email .For file decryption purpose reverse process of encryption is applied.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **1.** | **INTRODUCTION** | **1-3** |
| **2.** | **SYSTEM ANALYSIS** | **3-4** |
| **3.** | **SYSTEM REQUIREMENTS** | **5** |
| **4.** | **IMPLEMENTATION** | **6-7** |
| **5.** | **SYSTEM DESIGN** | **8-16** |
| **6.** | **INPUT & OUTPUT DESIGN** | **17-18** |
| **7.** | **LITERATURE SURVEY** | **18-20** |
| **8.** | **SOFTWARE ENVIRONMENT** | **21-39** |
| **9.** | **SYSTEM STUDY** | **40-42** |
| **10.** | **SYSTEM TEST** | **42-51** |
| **11.** | **SAMPLE CODE** | **51-59** |
| **12.** | **SCREENSHOTS & TEST CASES** | **60-66** |
| **13.** | **CONCLUSION** | **67** |
| **14.** | **REFERENCES** | **67** |

**SECURE FILE STORAGE IN CLOUD COMPUTING USING HYBRID CRYPTOGRAPHY ALGORITHM**

**INTRODUCTION**

Cryptography technique translates original data into unreadable form. Cryptography technique is divided into symmetric key cryptography and public key cryptography. This technique uses keys for translate data into unreadable form. So only authorized person can access data from cloud server. Cipher text data is visible for all people.

Symmetric key cryptography algorithms are AES,DES,3DES,IDEA ,BRA and blowfish. The main issue is deliver the key to receiver into multi user application. These algorithm require low delay for data encode decode but provides low security. Public key cryptography algorithm is RSA and ECC algorithm. Public and private keys are manipulated into public key cryptography algorithms. These algorithms accomplished high level security but increase delay for data encode and decode. Steganography hide the secret data existence into envelope. In this technique existence of data is not visible to all people. Only valid receiver knows about the data existence. Text steganography technique is used to produce high security for data. Secret data of user hide into text cover file. After adding text into text cover file it looks like normal text file. If text file found by illegitimate user than also cannot get sensitive data. If illegitimate user try to recover original data than large amount of time is essential.DES algorithm is used for text encode and decode. Advantage of text steganography technique is provide security to text. Minimum space is essential for text steganography as compare to image steganoraphy.

Three bit LSB technique used for image steganography. This system is suggested by author R.T.Patil .Sensitive data of user hide into cover image. We can hide huge amount of into image using LSB steganography technique .The author Klaus Hafmann has implemented high throughput architecture for cryptography algorithm.AES is symmetric key cryptography algorithm. It supports three types of keys. For 128 bit key require 10 rounds, 192 bit key require 12 rounds and 256 bit key require 14 rounds. In improved AES algorithm encryption and decryption time is reduced .Advantage of modified AES algorithm is provides better performance in terms of delay.

New symmetric key cryptography algorithm is presented by author M. Nagle. It applies a single key for texts encode and decode. Size of key is 128 bit. In this algorithm many steps are executed randomly so illegitimate user can even guess the steps of algorithm. Provide high throughput is one of the advantage of symmetric key cryptography algorithms. Improved DES algorithm uses 112 bit key size for data encode and decode. For data encode purpose two keys are used.128 bit input of DES algorithm is divided into two parts .That two parts are executed at a same time.DES algorithm has one weakness. That is less key size.3DES algorithm essential large amount of time for encryption and decryption .Improved DES algorithm have capability of provide better performance as compare to DES and 3DES. Name Based Encryption Algorithm is work on one byte at a time. It uses secret key for encryption and decryption .Key generation process is done using random key generation technique. It provides security to data. Disadvantage of this algorithm is essential maximum time for converting data into cipher text because it operate on single byte at a time To solve data storage and security issues author has new security model .In this model private and public cloud storage areas are used for increase security level of data. On private cloud secure data is stored and unnecessary data is stored on public cloud. Because public cloud any one can access. The main reason behind this system is reduce storage cost .Private cloud is more secure than the public cloud. To enhance security of file in cloud computing .Source file is break into different into different part. Every part of file is encrypted and stored on more than one cloud. Information about file is stored on cloud server for decryption purpose. If attacker try to recover original file than he will get only a single part of file. Elliptic Curve cryptography algorithm is used to accomplish high level security .Key managing complications are removed using access management and identity.ECC algorithm need maximum amount of time for file encode and decode. File is converted into unreadable format using AES algorithm. Encrypted file is stored on cloud.AES algorithm is less secure than public key cryptography algorithms.

AES and 3DES algorithms are merge into hybrid algorithm to accomplish confidentiality. It is harder for attacker to recover secret file of user. It consumes maximum amount of delay to translate data into decode and encode form.

In existing system single algorithm is used for data encode and decode purpose. But use of single algorithm is not accomplish high level security. If we use single symmetric key cryptography algorithm than we have to face security problem because in this type of algorithm applies a single key for data encode and decode. So key transmission problem occur while sharing key into multiuser environment. Public key cryptography algorithms accomplish high security but maximum delay is needed for data encode and decode. To solve above issues we have introduced new security mechanism.

Cloud owner and cloud user are included into system architecture. Cloud owner upload the data on cloud server. File is split into octet. Every part of file is encoded simultaneously using multithreading technique. Encoded file is stored on cloud server. Keys used for encryption are stored into cover image. Cloud computing is the multi user environment .In this more than one user can access file from cloud server. Cloud user request for file. On request of file user also get stego image using email which consist of key information. Reverse process is used for decode the file.

**SYSTEM ANALYSIS**

**Existing System:**

The customers who store their data with cloud service providers are liable to several threats. In our work, we have considered four different varieties of threat models. First, the single point of failure, which affects the information availability that would occur in case if a server of any cloud service provider has failed. Information availability is again a crucial issue which can also be affected, if in case the cloud service provider i.e., CSP, has run out of the service. Secondly, one major threat is data integrity. Integrity can be a degree of confidence that the information within the cloud that is what's imagined to be there, and has the protection against any accidental or intentional changes without proper authorization. Such worries are always present; so, in that case, a cloud service’s customer cannot completely rely upon a cloud service provider to ensure the proper storage of his important data

**Disadvantages:**

* Requires an active internet connection to connect with cloud server.

**Proposed System:**

The proposed model is liable to meet the required security needs of data center of cloud. Blowfish used for the encryption of file slices takes minimum time and has maximum throughput for encryption and decryption from other symmetric algorithms. The idea of splitting and merging adds on to meet the principle of data security. The hybrid approach when deployed in cloud environment makes the remote server more secure and thus, helps the cloud providers to fetch more trust of their users. For data security and privacy protection issues, the fundamental challenge of separation of sensitive data and access control is fulfilled. Cryptography technique translates original data into unreadable form. Cryptography technique is divided into symmetric key cryptography and public key cryptography. This technique uses keys for translate data into unreadable form. So only authorized person can access data from cloud server. Cipher text data is visible for all people.

**Advantages:**

* The stored image file is completely secured, as the file is being encrypted not by just using one but three encryption algorithm which are AES, DES and RC6.
* The key is also safe as it embeds the key in image using LSB.
* The system is very secure and robust in nature.
* Data is kept secured on cloud server which avoids unauthorized access.

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

* System : Pentium IV 3.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 14’ Colour Monitor.
* Mouse : Optical Mouse.
* Ram : 1 GB.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows Family.
* Coding Language : J2EE (JSP,Servlet,Java Bean)
* Data Base : MS Access.
* Web Server : Tomcat 6.0

**IMPLEMENTATION**

**MODULES**

1. Data Owner
2. Cloud Server
3. END User
4. Android User

**MODULAR DESCRIPTION**

* **Data Owner**

In this module, the data provider uploads their encrypted data in the Cloud server. For the security purpose the data owner encrypts the data file and then store in the server. The Data owner can have capable of manipulating the encrypted data file and performs the following operations Browse and enc and Uploads files, Grant Permission to cloud consumer / End user

* **Cloud Server**

The **Cloud** server manages which is to provide data storage service for the Data Owners. Data owners encrypt their data files and store them in the Server for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the Server and then Server will decrypt them. The server will generate the aggregate key if the end user requests for file authorization to access and performs the following operations such as View all User Files, Give privileges to user ,View Search Transaction, View all attackers ,View all End Users, View all Data Owners, Create Index on searched data and provide all related data related to corresponding keyword, View all android users

* **END User**

In this module, the user can only access the data file with the secret key. The user can search the file for a specified keyword. The data which matches for a particular keyword will be indexed in the cloud server and then response to the end user.

* **Android User**

We can easily use this application. This application user has to install in a mobile. Before using this application user should register, after registration he should login by using authorized user name and password. After login successful he will do some operations such as searching keyword in the cloud server to find KNN data and viewing cloud attackers in the android mobiles.

**SYSTEM DESIGN**

**SYSTEM ARCHITECTURE**

**1) View all User Files**

**2) Give privileges to user**

**3) View Search Transaction**

**4) View all attackers   
5).View all End Users  
6). View all Data Owners**

**7). Create Index on searched data and provide all related data related to corresponding keyword**

**8).View all android users**

**CLOUD SERVER**

**Browse and enc and Uploads files**

**1) View user Details**

**2) View Attacker Details**

**3) Unblock User**

**Data**

**Owner**

**1) Registers & Logins**

**1) Grant Permission to cloud consumer / End user**

**1) Searches for files based on Content’s keyword**

**2) Requests for Skey**

**3) Requests for downloading files  
4) Find the file search ratio**

**5) Find all K Nearest Neighbor search documents**

**1) Registers & Login**

**Web User / Android User**

**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling 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 Modeling 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 modeling 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.

**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.

**USE 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.

Upload File

Encrypt File

Upload Response

Browse File

**Data Owner**

Store Encrypted Data

asdfa

**Cloud Server**

View Attackers

asdfa

View Owner Files

asdfa

Give privileges to userasdfa

Create Index on searched dataasdfa

**Searches for files based on Content’s keyword**

**End User**

**Requests for Skey**

**Requests for downloading files**

Authorize user

Retrieve and store data

**Fig. Use Case Diagram**

**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

Members

Methods

Register

Register, Reset

Name, Password, DOB, Gender, Address, City, Country, Email, Mobile

Cloud Server

Data Owner

View all Owner Files, View Attacks,Store\_Files,Authorize\_Files,Authorize users, View all User Files,Give privileges to user ,View Search Transaction,View all attackers ,View all End Users,View all Data Owners,Create Index ,View all android users

File ID, File Name, Owner Name, Secret Key, User Details, File Access Details, View User Property, Hackers, Exit

Browse File, Upload File, Encrypt File, View All, Exit, View\_owner\_Files

Select File Name, Owner Name, Owner File

Methods

Methods

Members

Members

Login

Login, Register, Reset

User Name, Password

Methods

Members

End User

Search File Download, Search ‘N’ Ranked data, Exit, Register, and Login

Fname,n rank,username,secret\_key

Methods

Members

**SEQUENCE DIAGRAM:**

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

**End user**

**Cloud server**

**Data provider**

Browse & Encrypt uploads file

File sent confirmation

Request secret key

Gives secret key

Searching file based on Keyword

Check files name and Secret key

Search response

Request Access Permission

Access permission Response

Checks file names based on keyword and creating index of related files

Request SK

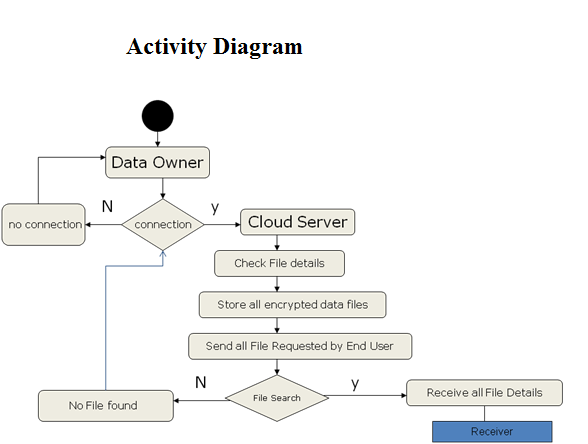
Search response

Download file

Save files

View all data provider files

**ACTIVITY DIAGRAM:**



**COMPONENT DIAGRAM:**

Data Owner

N:1

Cloud Server

N:N

Receiver

**Data Flow Diagram:**

**ATTACKER**

**Data Owner**

**CLOUD   
 SERVER**

**END USER**

**Android User**

**Fig. Data Flow Diagram**

**Flow Chart:**

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 the input data to the system, various processing carried out on these data, and the output data is generated by the system.

**User**

User Register

Start

Login

Yes No

View ‘N’ ranked data

Username & Password Wrong

Yes No

View Data Details

No Data

Select data

Yes No

Update Ranking

Post User Details

**Cloud Server**

**Cloud Server**

**Start**

User Login

Yes No

**Accept End user search**

**No Username Exists**

No

Yes

Search user Req

**Update Rank**

**Description**

**No data found**

Encrypt data

**Send Data to End User**

**Update Rank**

**View Rankings**

**End User**

**INPUT & OUTPUT DESIGNS**

**INPUT DESIGN**

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

**OUTPUT DESIGN**

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

**LITERATURAL SURVEY**

**Software protection and simulation on oblivious rams**

Software protection is one of the most important issues concerning computer practice. There exist many heuristics and ad-hoc methods for protection, but the problem as a whole has not received the theoretical treatment it deserves. In this paper, we provide theoretical treatment of software protection. We reduce the problem of software protection to the problem of efficient simulation on oblivious RAM.A machine is oblivious if thhe sequence in which it accesses memory locations is equivalent for any two inputs with the same running time. For example, an oblivious Turing Machine is one for which the movement of the heads on the tapes is identical for each computation. (Thus, the movement is independent of the actual input.) What is the slowdown in the running time of a machine, if it is required to be oblivious? In 1979, Pippenger and Fischer showed how a two-tape oblivious Turing Machine can simulate, on-line, a one-tape Turing Machine, with a logarithmic slowdown in the running time. We show an analogous result for the random-access machine (RAM) model of computation. In particular, we show how to do an on-line simulation of an arbitrary RAM by a probabilistic oblivious RAM with a polylogaithmic slowdown in the running time. On the other hand, we show that a logarithmic slowdown is a lower bound.

**Practical techniques for searches on encrypted data**

It is desirable to store data on data storage servers such as mail servers and file servers in encrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client wishes to retrieve only documents containing certain words, it was not previously known how to let the data storage server perform the search and answer the query, without loss of data confidentiality. We describe our cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. Our techniques have a number of crucial advantages. They are provably secure: they provide provable secrecy for encryption, in the sense that the untrusted server cannot learn anything about the plaintext when only given the ciphertext; they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for an arbitrary word without the user's authorization; they also support hidden queries, so that the user may ask the untrusted server to search for a secret word without revealing the word to the server. The algorithms presented are simple, fast (for a document of length n, the encryption and search algorithms only need O(n) stream cipher and block cipher operations), and introduce almost no space and communication overhead, and hence are practical to use today.

**Computationally private information retrieval with polylogarithmic communication**

We present a single-database computationally private information retrieval scheme with polylogarithmic communication complexity. Our construction is based on a new, but reasonable intractability assumption, which we call the Φ-Hiding Assumption (ΦHA): essentially the difficulty of deciding whether a small prime divides Φ(m), where m is a composite integer of unknown factorization.

**Single database private information retrieval implies oblivious transfer**

A Single-Database Private Information Retrieval (PIR) is a protocol that allows a user to privately retrieve from a database an entry with as small as possible communication complexity. We call a PIR protocol non-trivial if its total communication is strictly less than the size of the database. Non-trivial PIR is an important cryptographic primitive with many applications. Thus, understanding which assumptions are necessary for implementing such a primitive is an important task, although (so far) not a well-understood one. In this paper we show that any non-trivial PIR implies Oblivious Transfer, a far better understood primitive. Our result not only significantly clarifies our understanding of any non-trivial PIR protocol

**5) Public Key Encryption with keyword Search**

We study the problem of searching on data that is encrypted using a public key system. Consider user Bob who sends email to user Alice encrypted under Alice's public key. An email gateway wants to test whether the email contains the keyword \urgent" so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. We de\_ne and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word \urgent" is a keyword in the email without learning anything else about the email. We refer to this mechanism as PublicKey Encryption with keyword Search. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using our mechanism Alice can send the mail server a key that will enable the server to identify all messages containing some specific keyword, but learn nothing else. We define the concept of public key encryption with keyword search and give several constructions.

**SOFTWARE ENVIRONMENT**

## **Java Technology**

Java technology is both a programming language and a platform.

### **The Java Programming Language**

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

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

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



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



### **The Java Platform**

A platform is the hardware or software environment in which a program runs. We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and 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.

## **What Can Java Technology Do?**

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

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

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

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

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

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



## **How Will Java Technology Change My Life?**

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

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

### **ODBC**

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

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

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

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

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

**JDBC**

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

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

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

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

### **JDBC Goals**

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

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

1. **SQL Level API**

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

1. **SQL Conformance**

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

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

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

1. **Keep it simple**

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

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

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

1. **Keep the common cases simple**

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

Finally we decided to proceed the implementation using Java Networking.

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

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

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

Simple Architecture-neutral

Object-oriented Portable

Distributed High-performance

Interpreted multithreaded

Robust Dynamic

Secure

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

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

**Java Program**

**Compilers**

**Interpreter**

**My Program**

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

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

## **Networking**

### **TCP/IP stack**

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

****

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

### **IP datagram’s**

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

### **UDP**

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

### **TCP**

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

### **Internet addresses**

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

**Network address**

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

**Subnet address**

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

**Host address**

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

### **Total address**

****

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

### **Port addresses**

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

### **Sockets**

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

#include <sys/types.h>

#include <sys/socket.h>

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

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

**JFree Chart**

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

* A consistent and well-documented API, supporting a wide range of chart types;
* A flexible design that is easy to extend, and targets both server-side and client-side applications;
* Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);
* JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

## **Map Visualizations**

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

* Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);
* Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;
* Testing, documenting, testing some more, documenting some more.

## **2. Time Series Chart Interactivity**

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

## **3. Dashboards**

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

## **4. Property Editors**

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

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

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

**1. General J2ME Architecture**



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

**2. Developing J2ME applications**

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

**3. Design considerations for small devices**

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

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

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

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

**4. Configurations overview**

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

* **Connected Limited Device Configuration (CLDC)** is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.
* **Connected Device Configuration (CDC)** is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

**5. J2ME profiles**

**What is a J2ME profile?**

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

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

**Profile 1: KJava**

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

**Profile 2: MIDP**

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

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

\* java.lang

\* java.io

\* java.util

\* javax.microedition.io

\* javax.microedition.lcdui

\* javax.microedition.midlet

\* javax.microedition.rms

**SYSTEM STUDY**

**PRELIMINARY INVESTIGATION**

The first and foremost strategy for development of a project starts from the thought of designing a mail enabled platform for a small firm in which it is easy and convenient of sending and receiving messages, there is a search engine ,address book and also including some entertaining games. When it is approved by the organization and our project guide the first activity, ie. preliminary investigation begins.

The activity has three parts:

* **Request Clarification**
* **Feasibility Study**
* **Request Approval**

**REQUEST CLARIFICATION**

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Here our project is basically meant for users within the company whose systems can be interconnected by the Local Area Network (LAN). In today’s busy schedule man need everything should be provided in a readymade manner. So taking into consideration of the vastly use of the net in day to day life, the corresponding development of the portal came into existence.

**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

* **OPERATIONAL FEASIBILITY**
* **ECONOMIC FEASIBILITY**
* **TECHNICAL FEASIBILITY**

###### **OPERATIONAL FEASIBILITY**

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

###### **ECONOMIC FEASIBILITY**

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

###### **TECHNICAL FEASIBILITY**

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and Web Logic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

**REQUEST APPROVAL**

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, it cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched.

**SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

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

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### **TESTING METHODOLOGIES**

The following are the Testing Methodologies:

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

**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.

**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.

1. **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 Software 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.

**OTHER TESTING METHODOLOGIES**

**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.

**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.

**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.

**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.

**MAINTAINENCE**

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.

**SAMPLE CODE**

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

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

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

<%@ page import="com.oreilly.servlet.\*,java.lang.\*,java.text.SimpleDateFormat,java.io.\*,javax.servlet.\*, javax.servlet.http.\*" %>

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

<%@ page import="org.bouncycastle.util.encoders.Base64"%>

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

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

<head>

<title> Cloud Server </title>

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

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

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

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

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

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

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

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

<style type="text/css">

<!--

.style1 {font-size: 16px}

.style2 {font-size: 14px}

.style3 {font-size: 24}

.style3 {font-size: 26px; }

.style4 { font-size: 24;

font-weight: bold;

color: #666666;

}

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header\_resize">

<div class="searchform">

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

<span>

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

</span>

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

</form>

</div>

<div class="menu\_nav">

<ul>

<li class="active"></li>

<li class="style2">

<div align="left"><a href="User.html"><span>USER</span></a></div>

</li>

<li><a href="Register.html"><span>REGISTER</span></a></li>

<li><a href="DataOwner.html"><span>DATA OWNER</span></a></li>

<li><a href="CloudServer.html"><span>CLOUD SERVER</span></a></li>

<li><a href="About.html"><span>ABOUT SYSTEM</span></a></li>

</li>

</ul>

</div>

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

<div class="logo">

<h1><a href="index.html">Endas<span></span><small>Efficient Encrypted Data Search as a Mobile Cloud Service</small></a></h1>

</div>

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

<div class="slider">

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

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

</div>

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

</div>

</div>

<div class="content">

<div class="content\_resize">

<div class="mainbar">

<div class="article">

<h2><span>Attackers ::</span></h2>

<p>&nbsp;</p>

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

<table width="960" border="1" align="center">

<tr>

<td width="178" height="31"><div align="center"><span class="style3">User Name </span></div></td>

<td width="252"><div align="center"><span class="style3">File Name </span></div></td>

<td width="223"><div align="center"><span class="style3">Secret Key </span></div></td>

<td width="279"><div align="center"><span class="style3">Date & Time</span></div></td>

</tr>

<%

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

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

try

{

String query="select \* from Attacker ";

Statement st=connection.createStatement();

ResultSet rs=st.executeQuery(query);

while ( rs.next() )

{

i=rs.getInt(1);

s2=rs.getString(2);

s3=rs.getString(3);

s4=rs.getString(4);

s5=rs.getString(5);

%>

<tr>

<td height="29"><div align="center"><%=s2%></div></td>

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

<td><div align="center"><%=s4%></div></td>

<td><div align="center"><%=s5%></div></td>

</tr>

<%

}

connection.close();

}

catch(Exception e)

{

out.println(e.getMessage());

}

%>

</table>

</p>

<p> BACK <a href="CloudServerMain.html">CLOUD SERVER MENU</a></p>

</div>

</div>

<div class="sidebar">

<div class="searchform">

</div>

</div>

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

<ul class="ex\_menu">

<li></li>

</ul>

</div>

</div>

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

</div>

</div>

<div class="fbg">

<div class="fbg\_resize">

<div class="col c1">

<h2><span>Image</span> Gallery</h2>

<a href="#"><img src="images/gal1.jpg" width="75" height="75" alt="" class="gal" /></a> <a href="#"><img src="images/gal2.jpg" width="75" height="75" alt="" class="gal" /></a> <a href="#"><img src="images/gal3.jpg" width="75" height="75" alt="" class="gal" /></a> <a href="#"><img src="images/gal4.jpg" width="75" height="75" alt="" class="gal" /></a> <a href="#"><img src="images/gal5.jpg" width="75" height="75" alt="" class="gal" /></a> <a href="#"><img src="images/gal6.jpg" width="75" height="75" alt="" class="gal" /></a> </div>

<div class="col c2">

<h2><span>Trapdoor generation process</span></h2>

<p>The trapdoor

is traditionally generated by the provider to provide

data security. To perform a search

request, the user first authenticates with the provider. During authentication, the provide would send its secret

key to the user to decrypt the documents stored in cloud.</p>

</div>

<div class="col c3">

<h2 class="style3">Efficient Search Algorithm</h2>

<p class="style4">The efficient search algorithm proposed by EnDAS relies

on a binary search tree structure to accelerate indexing.

This paper introduce the conventional

privacy-preserving index construction procedures, including index construction.</p>

<p class="contact\_info style2">&nbsp;</p>

</div>

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

</div>

</div>

<div class="footer">

<div class="footer\_resize">

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

</div>

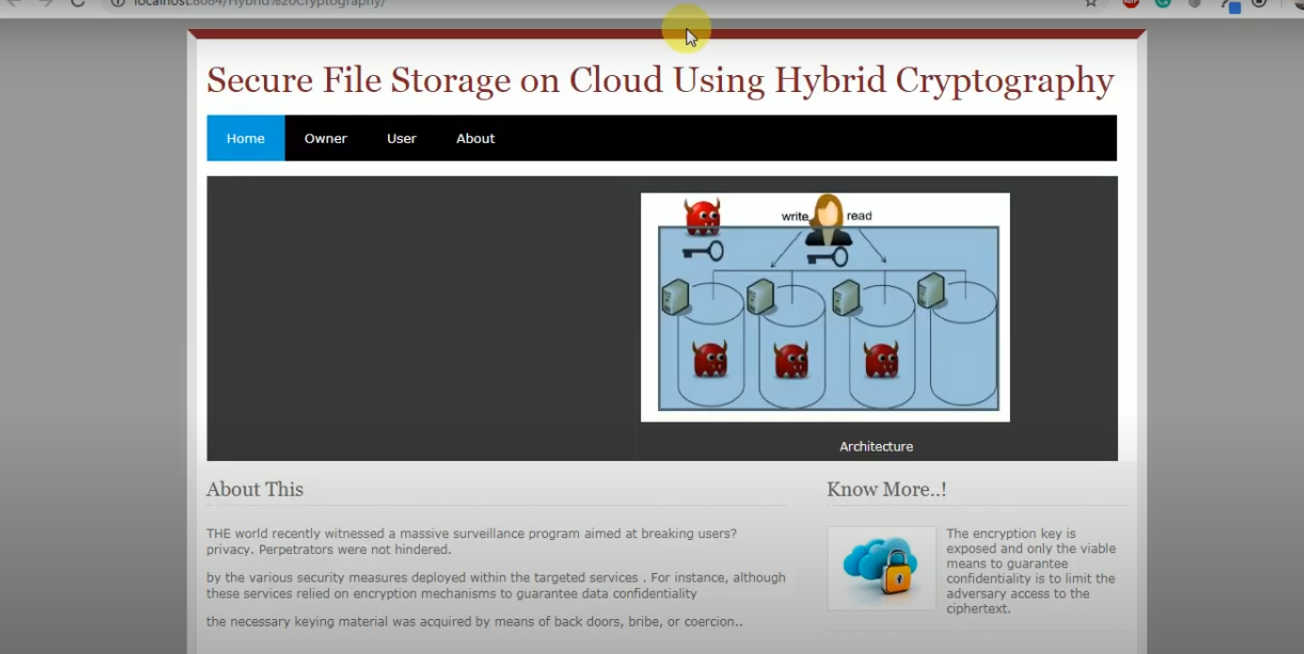
</div>

</div>

</body>

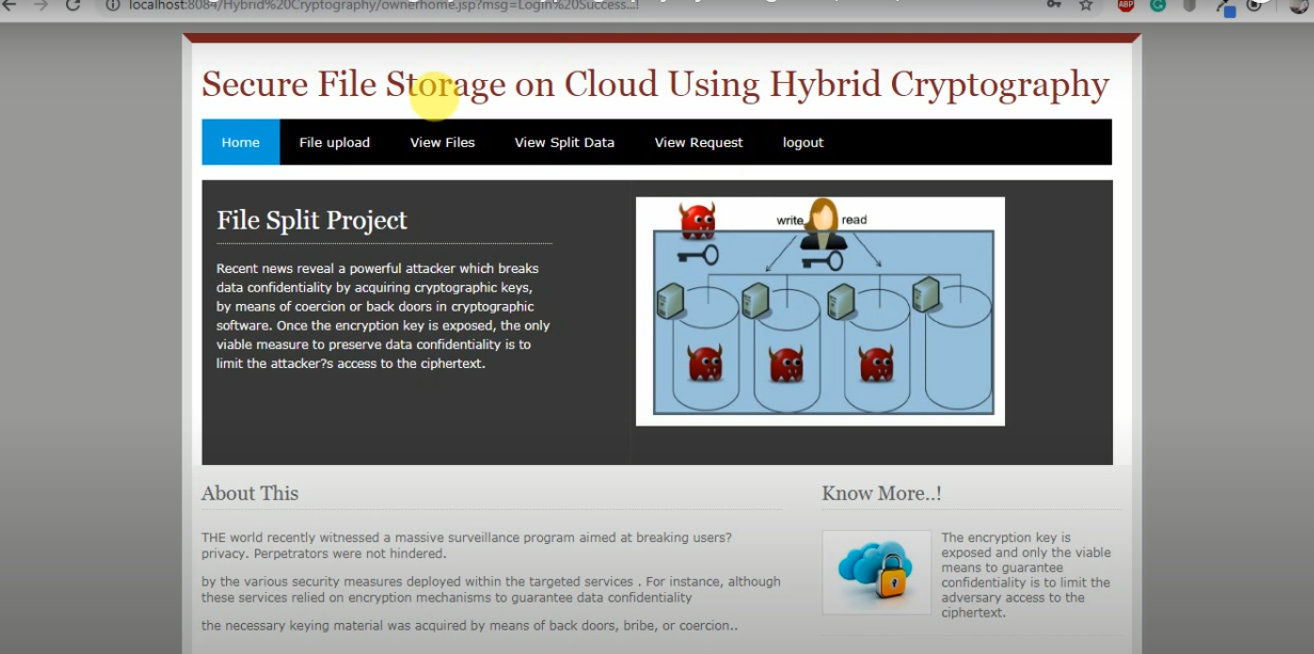
</html>

**SCREEN SHOTS**

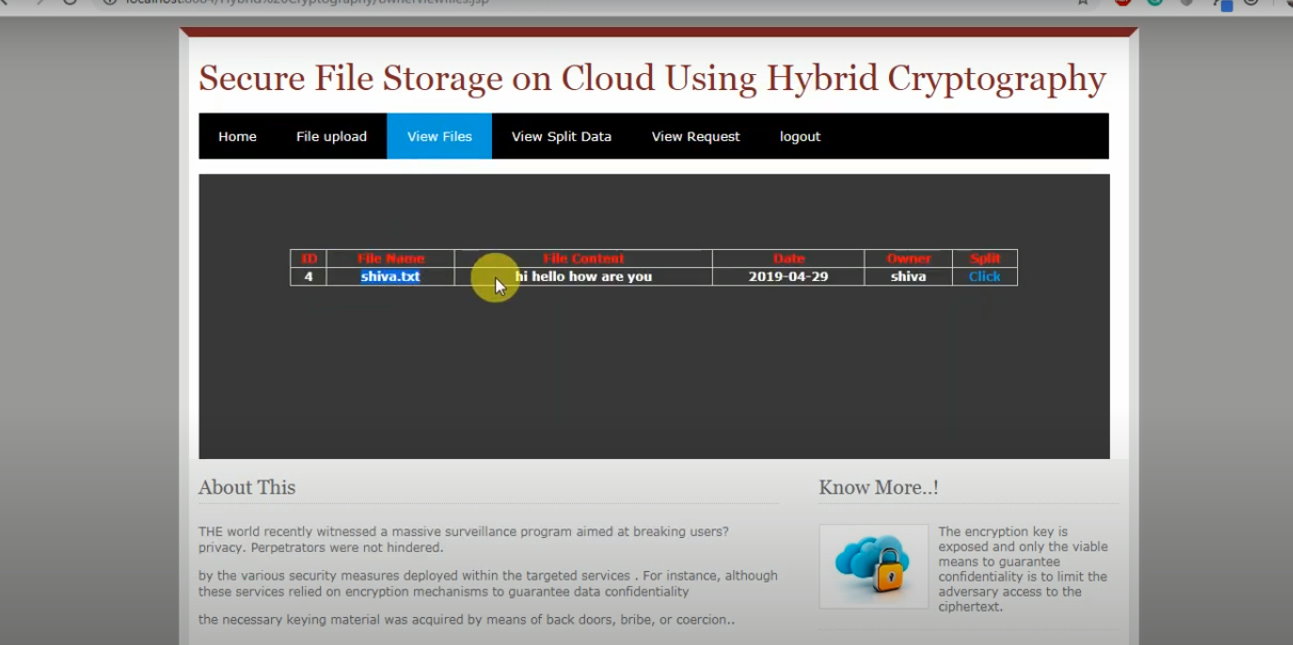




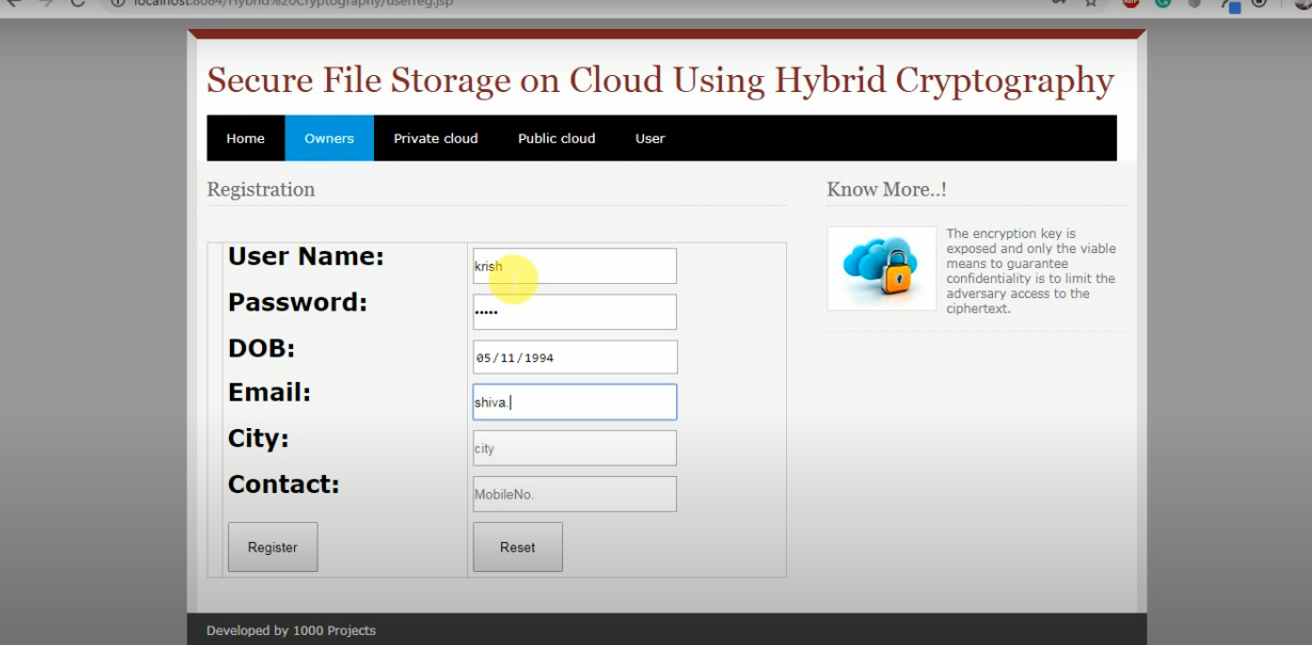


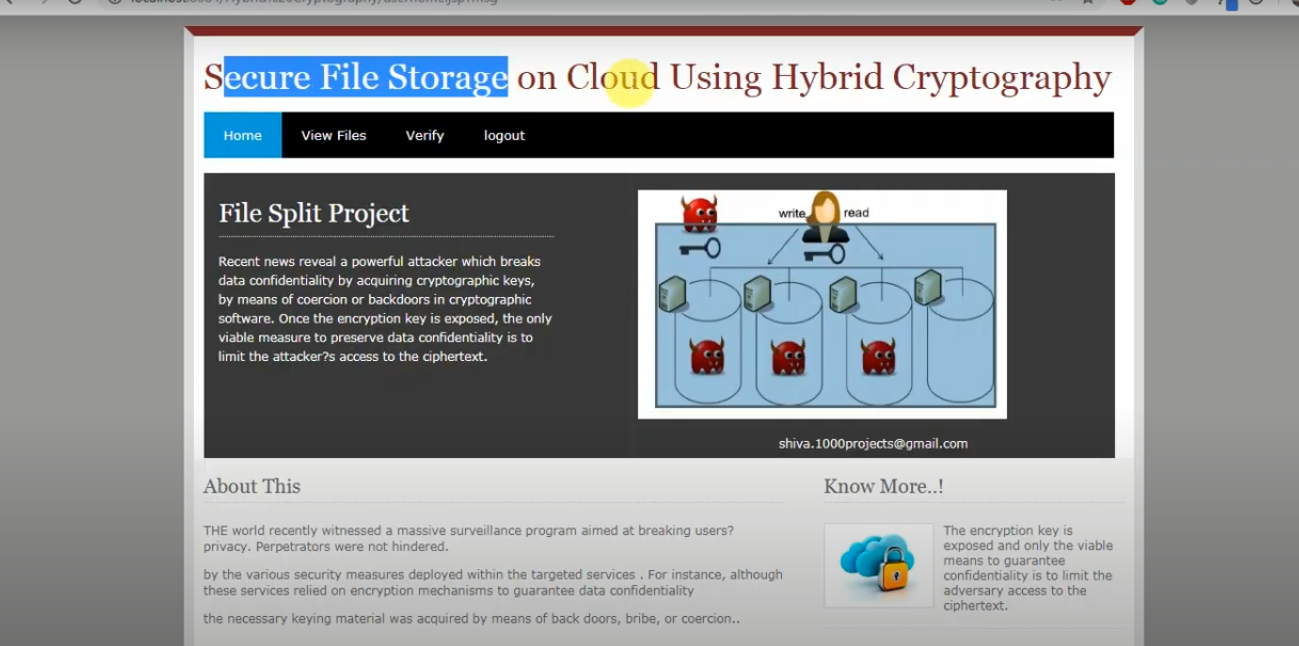


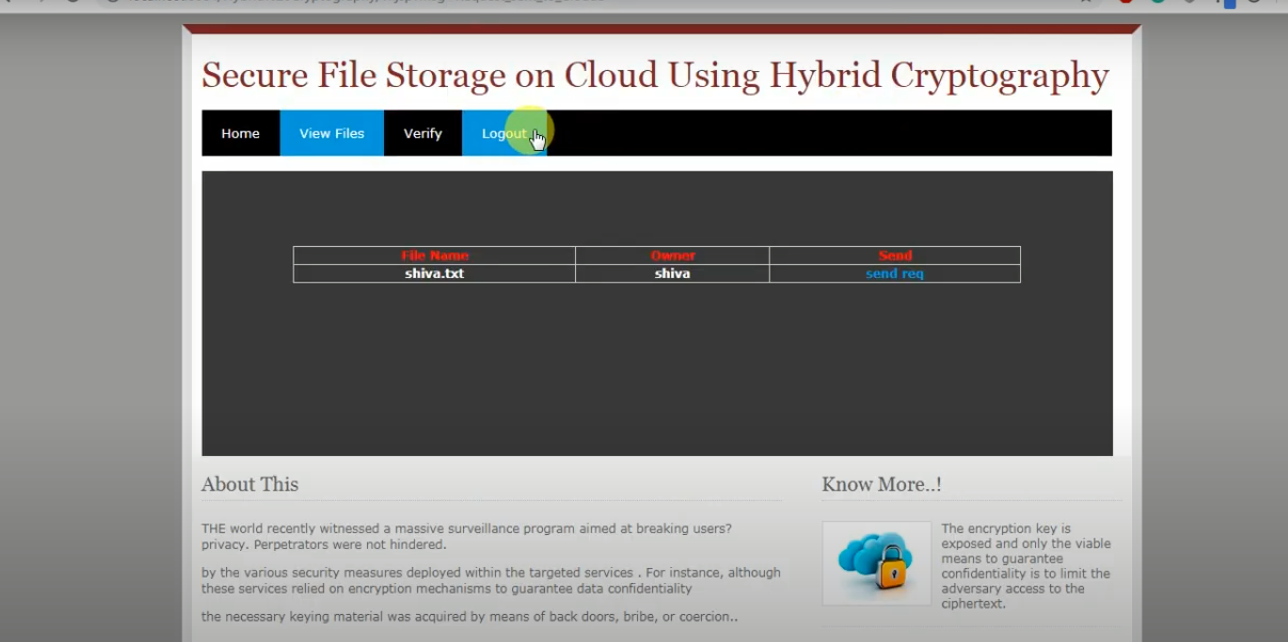


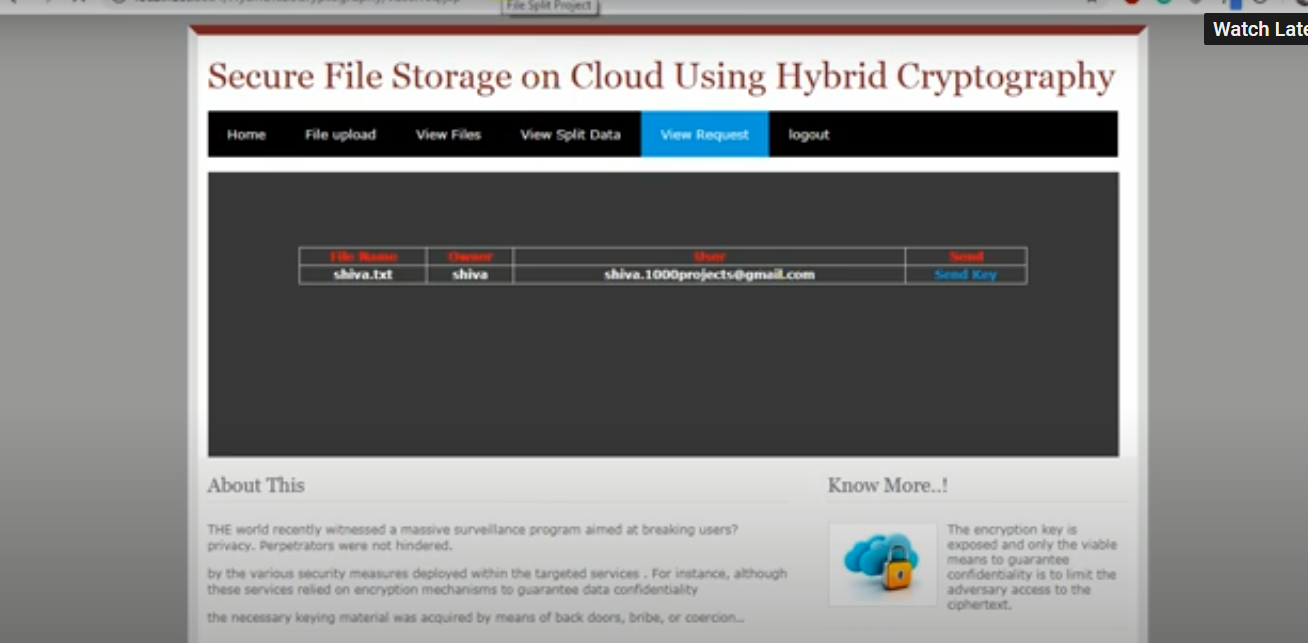














**TESTCASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Id** | **Test Name** | **Input** | **Output** | **Expected Result** | **Status** |
| 1 | Owner browsing the data | file | Data stored | Data stored in encrypted form | PASS |
|  | Owner browsing data | data | Data is not Present in directory | Data stored in encrypted form | FAIL |
| 2 | Owner Verification of Files | File Name with Uploaded MAC In Cloud and Auditor | File Attacked or Safe | Attacked or Safe | PASS |
|  | Owner Verification of Files | No Inputs | No Mac to Compare File | Attacked or Safe | FAIL |
| 3 | User Access File MAC & SK | File Name | Get MAC and SK if Cloud Permitted | Get SK and MAC | PASS |
|  | User Access File MAC & SK | File Name | Should not Get MAC and SK if Cloud not Permitted | Get SK and MAC | FAIL |
| 4 | User Downloads file content | File Name | Get decrypted content if Cloud Permitted | Download Content | PASS |
|  | User Downloads file content | File Name | Should not Get decrypted content if Cloud not Permitted | Download Content | FAIL |
| 5 | End user downloading data from server | Server IP address | Downloaded Data from the server | Data should be downloaded | PASS |
|  | End user downloading data from server | Invalid server IP address | Cant Downloaded Data from the server | Data should be downloaded | FAIL |
| 6 | Attacker (Hacking) | File Name | Get File Details | Hack the content | PASS |
|  | Attacker(Hacking)  With Same credential of Attacker | File Name | Get File Details | Hack the content | Fail |

**CONCLUSION**

Cloud storage issues are solved using cryptography and steganogarphy techniques.. Block wise Data security is achieved using AES, RC6, Blowfish and BRA algorithms. Key information security is accomplished using LSB technique. Data integrity is accomplished using SHA1 hash algorithm. Low delay parameter is achieved using multithreading technique. With the help of proposed security mechanism data integrity, high security, low delay, authentication and confidentiality parameters are accomplished. Using proposed Text file encryption need 17% to 20% less time as compare to AES algorithm. For AES text decryption needs 15% to 17% maximum time as compare to proposed system. In Blowfish for encryption need 12% to 15% maximum time as compare to proposed hybrid algorithm. Text file decryption using hybrid algorithm need 10% to 12% less time with respect to Blowfish algorithm. In future, try to accomplish high level security using hybridization of public key cryptography algorithms.

**REFERENCES**

1. V.S. Mahalle , A. K. Shahade, “Enhancing the Data Security in Cloud by Implementing Hybrid (Rsa & Aes) Encryption Algorithm”, *IEEE , INPAC*,pp 146-149,Oct .2014.
2. Abu Marjan, Palash Uddin, “Developing Efficient Solution to Information Hiding through text steganography along with cryptography”,IEEE, IFOST,pages 14-17, October 2014.
3. P. S. Bhendwade and R. T. Patil, “Steganographic Secure Data Communication”,IEEE, *International Conference on Communication and Signal Processing*, pages 953-956,April 2014.
4. S. Hesham and Klaus Hofmann , “High Throughput Architecture for the Advanced Encryption Standard Algorithm” *IEEE,International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, pages 167- 170,April 2014.