INTEGRATED PROJECT REPORT

ON

DECENTRALIZED

SECURED FILE STORAGE

COMPUTER SCIENCE AND ENGINEERING

Batch-2016

Submitted By:-

Akshit Gupta

1610991077

Mast Ram Sharma

1610991508

Vinayak Dev Sharma

1610991957

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHITKARA UNIVERSITY

PUNJAB

CERTIFICATE

This is to be certified that the project entitled “DECENTRALIZED SECURED FILE STORAGE” has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester January 2019- May-2019 is a bonafide piece of project work carried out by “Akshit Gupta (1610991077), Mast Ram Sharm (1610991508), Vinayak Dev Sharma (1610991957) ” towards the partial fulfilment for the award of the course Integrated Project ( ) under the guidance of “Mr. Jatin Gupta” and supervision.

Sign: ___________
Mr. Jatin Gupta
Associate Professor
Department Of CSE

CANDIDATE'S DECLARATION

We, **Akshit Gupta (1610991077), Mast Ram Sharm (1610991508), Vinayak Dev Sharma (1610991957) ,** B.E.-2016 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled **"DECENTRALIZED SECURED FILE STORAGE"** is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

Sign.____________	**Sign.____________**	**Sign. ___________ _**
Akshit Gupta	Mast Ram Sharma	Vinayak Dev Sharma
1610991077	1610991508	1610991957

Place:
Date:

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behaviour and acts during the course of study.

We express our sincere gratitude to all for providing us an opportunity to undergo Integrated Project as the part of the curriculum.

We are thankful to “Mr Jatin Gupta” for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

Lastly, We would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improve our quality of work.

Akshit Gupta	Mast Ram Sharma	Vinayak Dev Sharma
1610991077	1610991508	1610991957

Contents

Introduction

Cloud storage offers a convenient alternative to offline storage devices, but cloud providers are not necessarily the most secure option available to consumers. Even though data is encrypted but still the encryption depends on individual servers to protect data as it travels between the user's computer and the cloud. Therefore, it can be broken or weakened by a server that has been compromised by a hacker, or doesn't support the latest version of encryption, potentially allowing a bad actor to steal your data or login information. Not only are they susceptible to identity theft, financial theft and even potential for physical harm. Cloud servers are also prone to natural/manmade disasters, accidents and attacks like DOS or DDOS.

DE - CLOUD is a distributed cloud storage system which splits the data in large numbers of chunks and then encrypting it to send into a diverse network of storage devices.

Encrypted small spitted chunks of data ensure that even if a cloud storage gets compromised, the attacker will only get a small incomplete encrypted form of a large file which is useless until you gather and combine all chunks of data. Random sending of data in diversified storage locations makes it practically impossible for any attacker to find all the chunks of data to combine them into complete file while making the server invincible to DOS and DDOS attacks.

Working

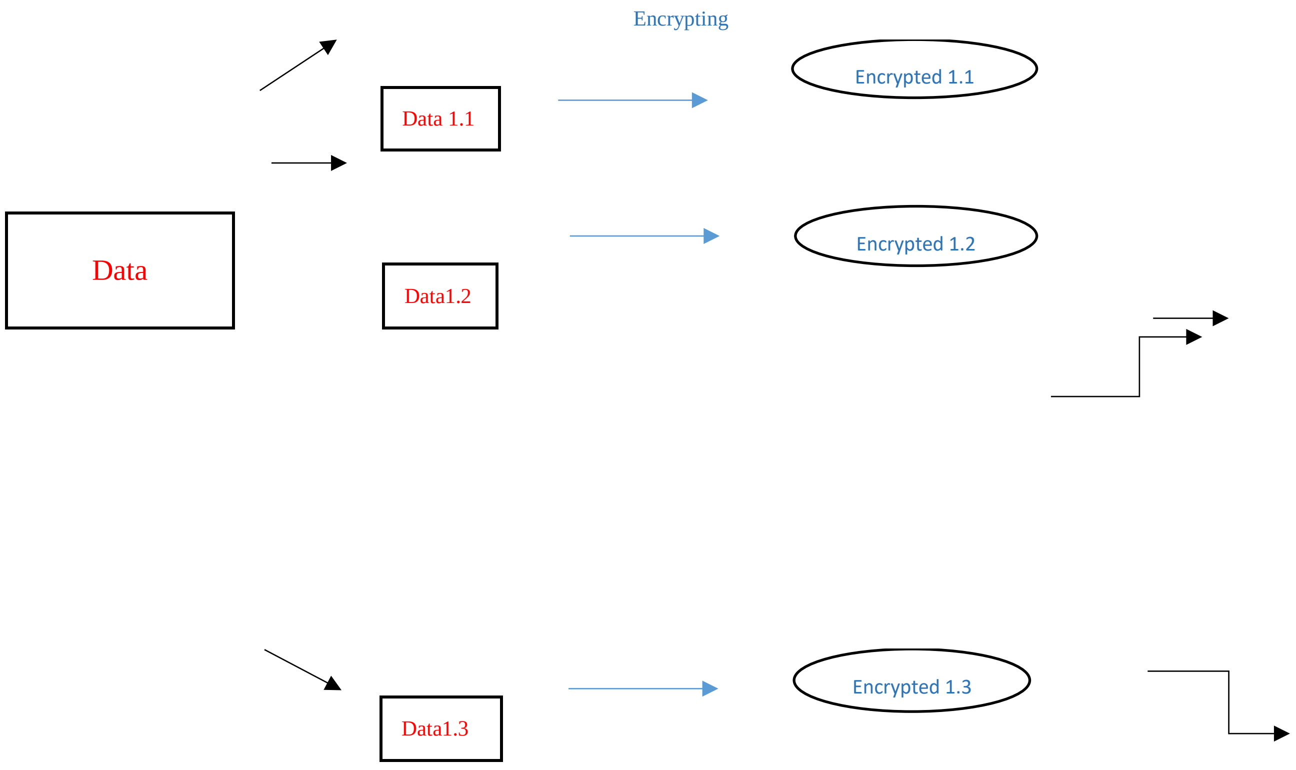
Uploading

File is first divided into small chunks of random numbers and then encrypted using AES encryption. A unique key is generated and then encrypted chunks are sent to the suitable random storage location.

Downloading

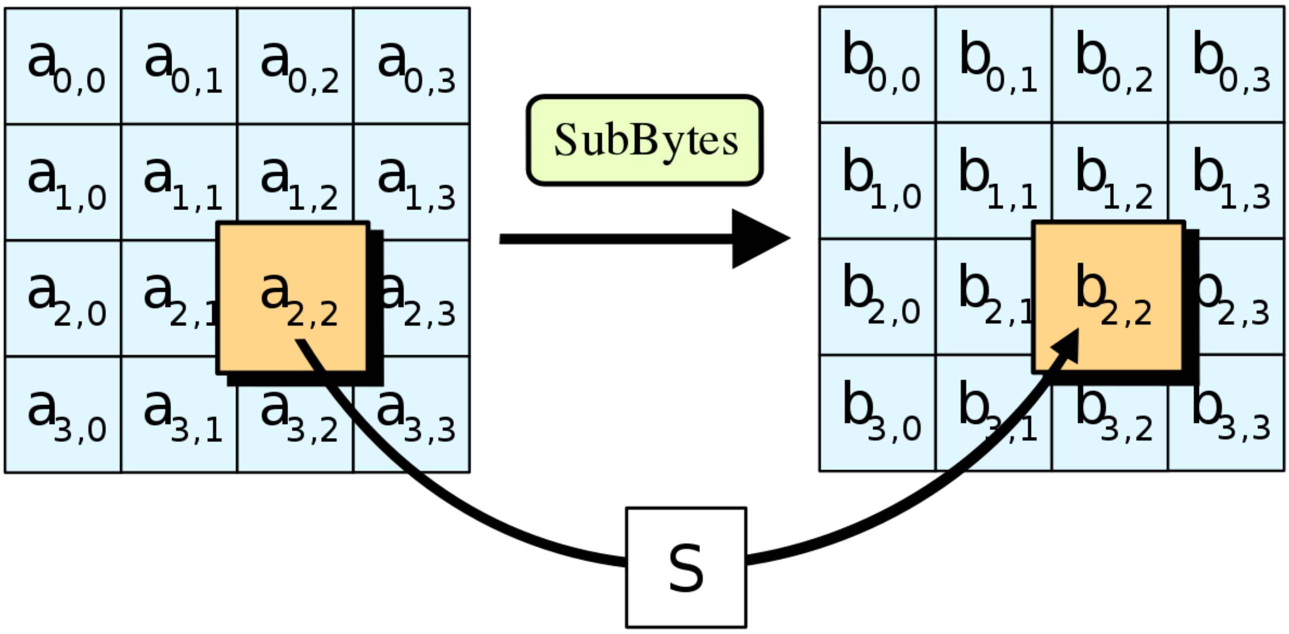
User supplies his unique key. Unique algorithm locates the file and then check its integrity using hash functions. If the file has been modified then system generate error and no merging of file is performed.

If hash verification passes then encrypted chunks from all over the locations are gathered and then decrypted using given key.



Technology Used

AES encryption

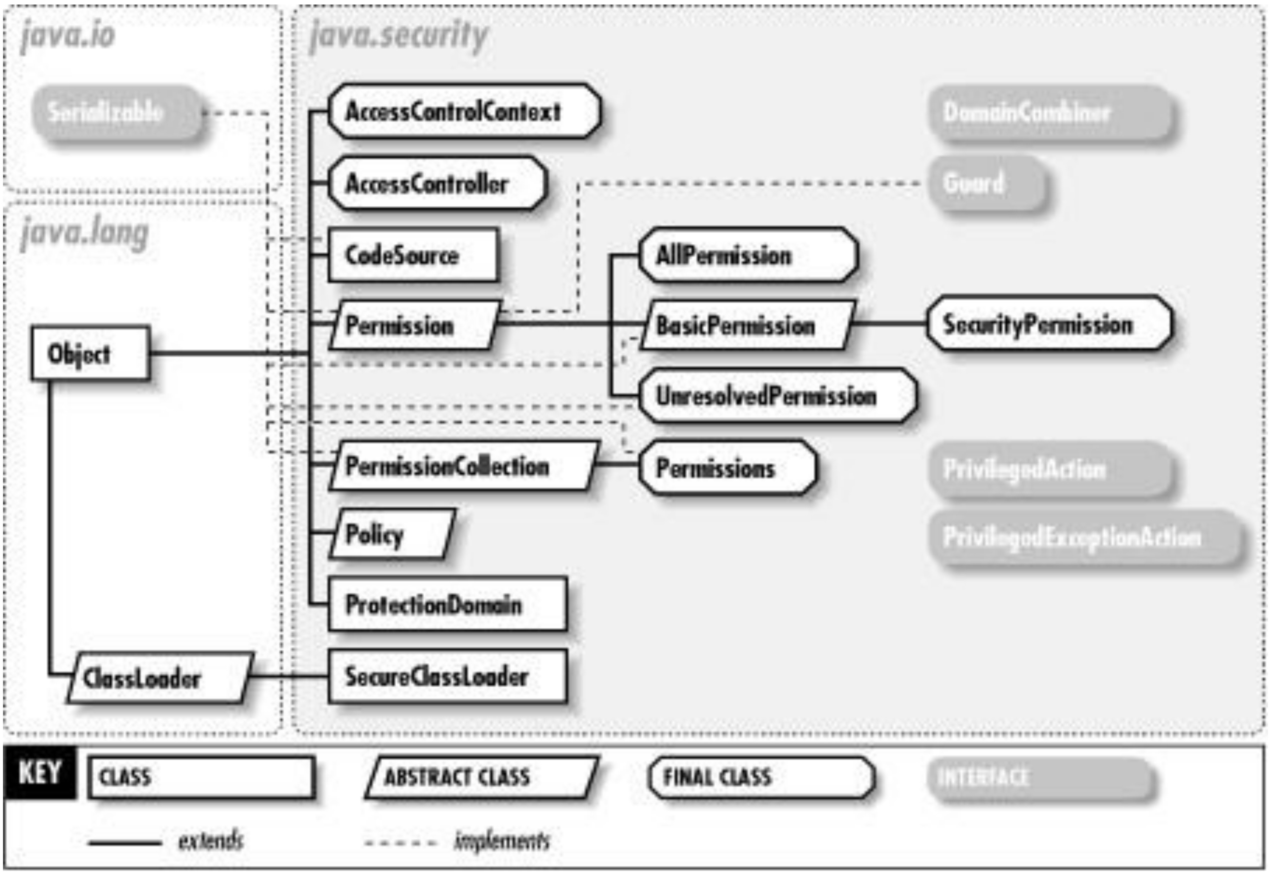

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

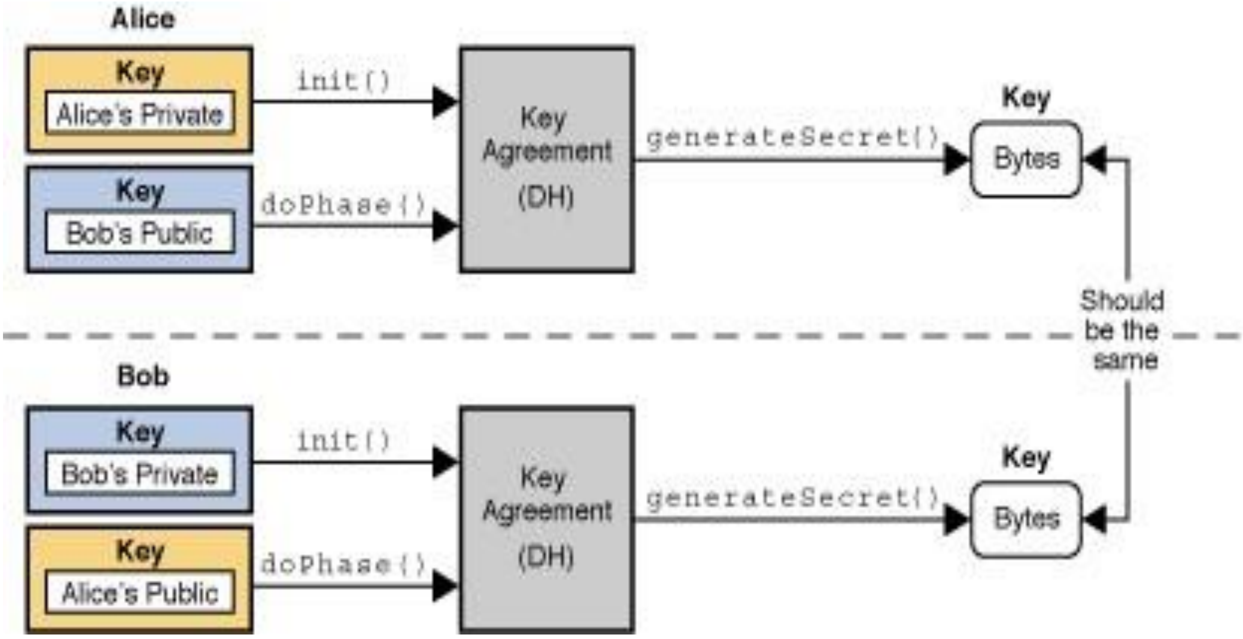
Java Libraries

1) Security



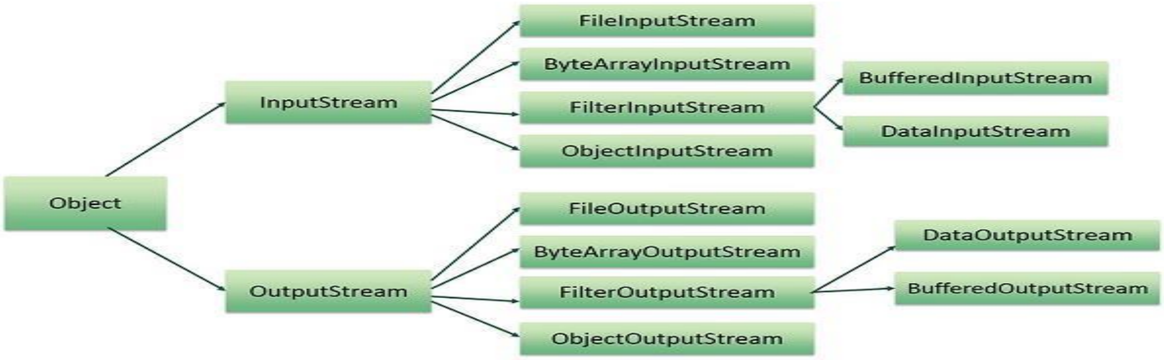
Java Security services have expanded and include a large set of application programming interfaces (APIs), tools, a number of security algorithm implementations, mechanisms, and protocols. This provides a comprehensive environment to develop secure applications and manage them accordingly. The span of the Java security API is extensive. The basis of developing a secure application lies in the Cryptographic and public key infrastructure (PKI) interfaces, multiple interoperable common algorithmic implementation, and other security services. There are interfaces for performing authentication and access control. This enables applications to guard against unauthorized access to protected resources.

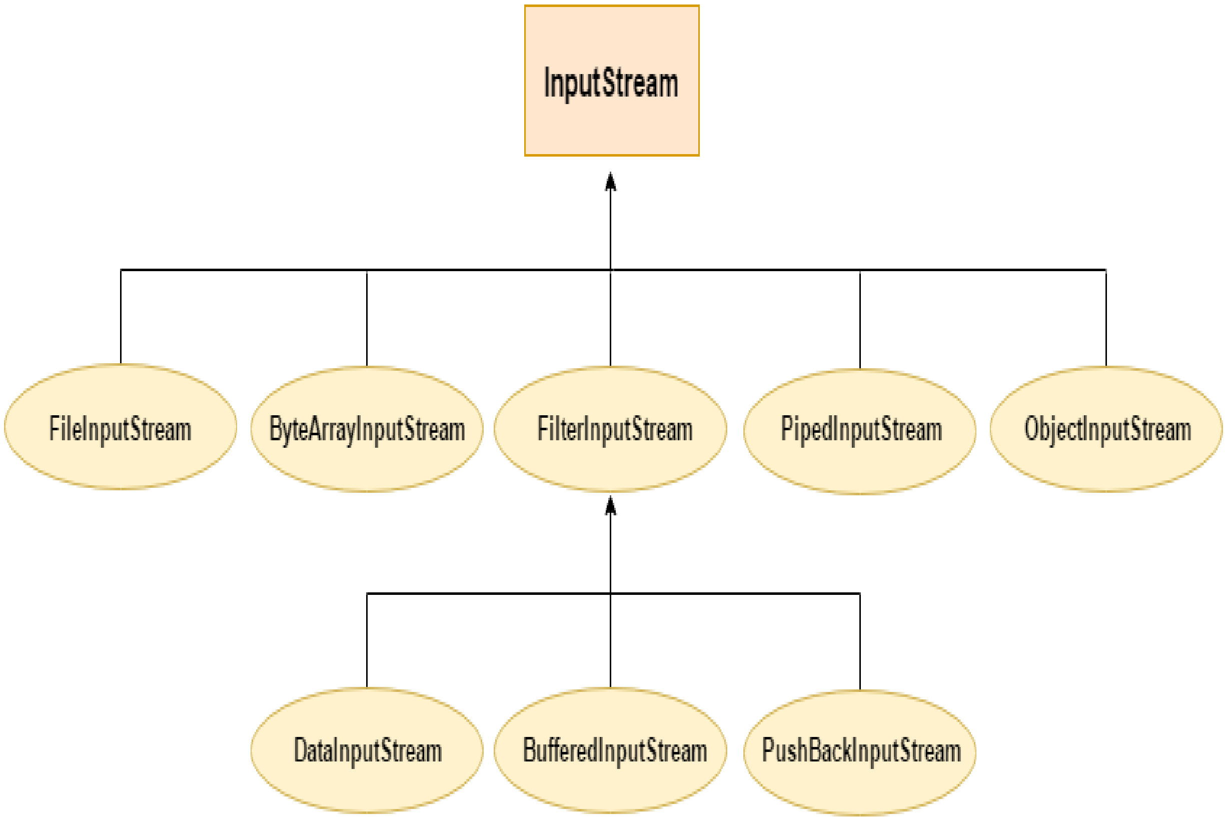
2) Crypto

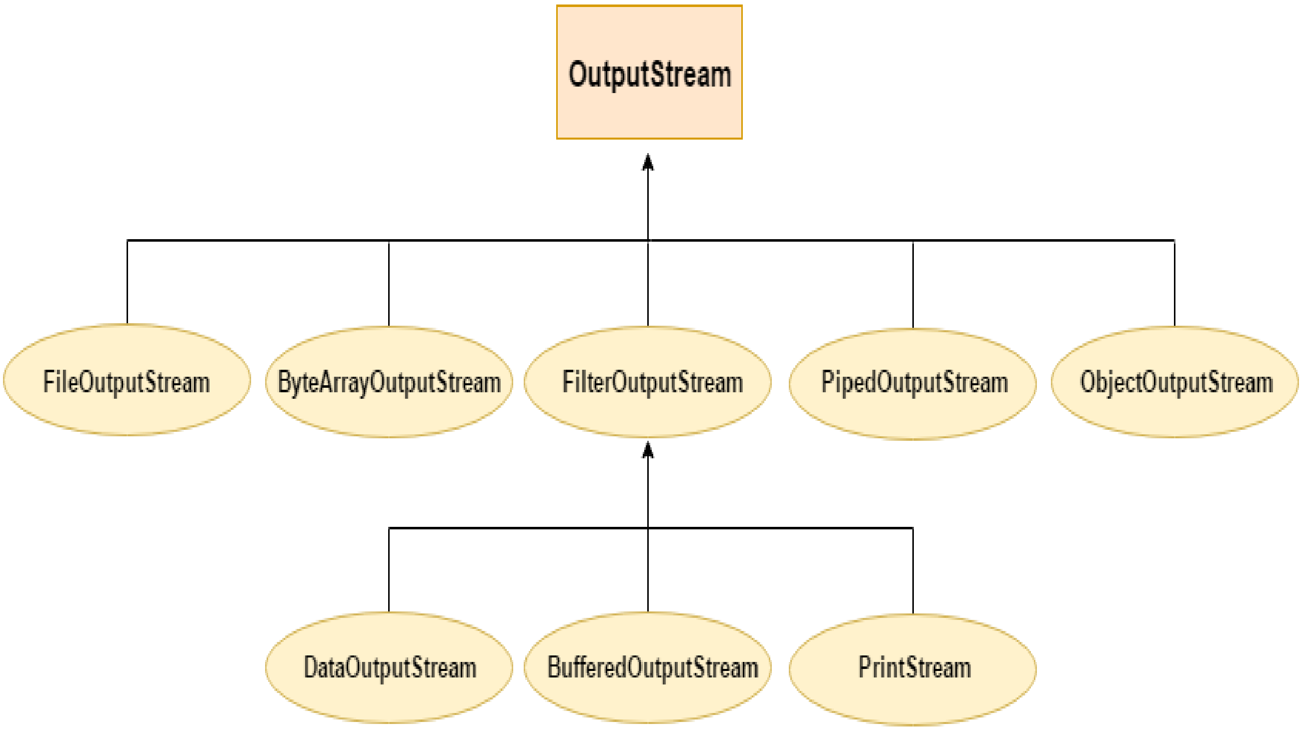


The Java Cryptography Architecture (JCA) is the name for the internal design of the Java cryptography API.JCA is structured around some central generalpurpose classes and interfaces. The real functionality behind these interfaces are provided by *providers*. Thus, you may use a Cipher class to encrypt and decrypt some data, but the concrete cipher implementation (encryption algorithm) depends on the concrete provider used. We can implement and plugin your own providers too, but you should be careful with that.
Implementing encryption correctly without security holes is hard! Unless you know that you are doing, you are probably better off using the built in Java provider, or use a well-established provider like Bouncy Castle.

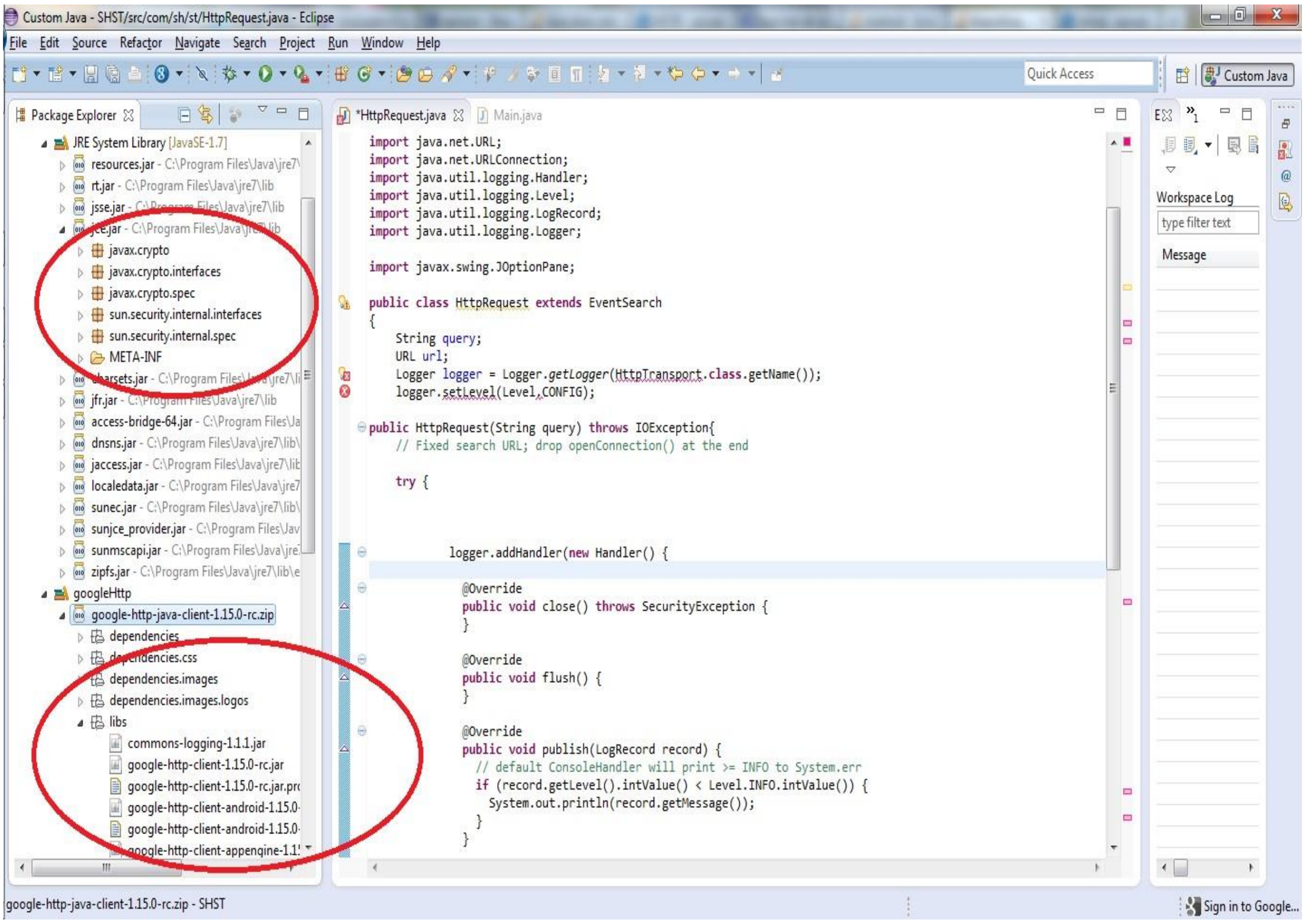
3) IO


Object
InputStream
OutputStream
FileInputStream
ByteArrayInputStream
FilterInputStream
ObjectInputStream
BufferedInputStream
DataInputStream
FileOutputStream
ByteArrayOutputStream
FilterOutputStream
ObjectOutputStream
DataOutputStream
BufferedOutputStream


InputStream
FileInputStream
ByteArrayInputStream
FilterInputStream
PipedInputStream
ObjectInputStream
DataInputStream
BufferedInputStream
PushBackInputStream



4) File

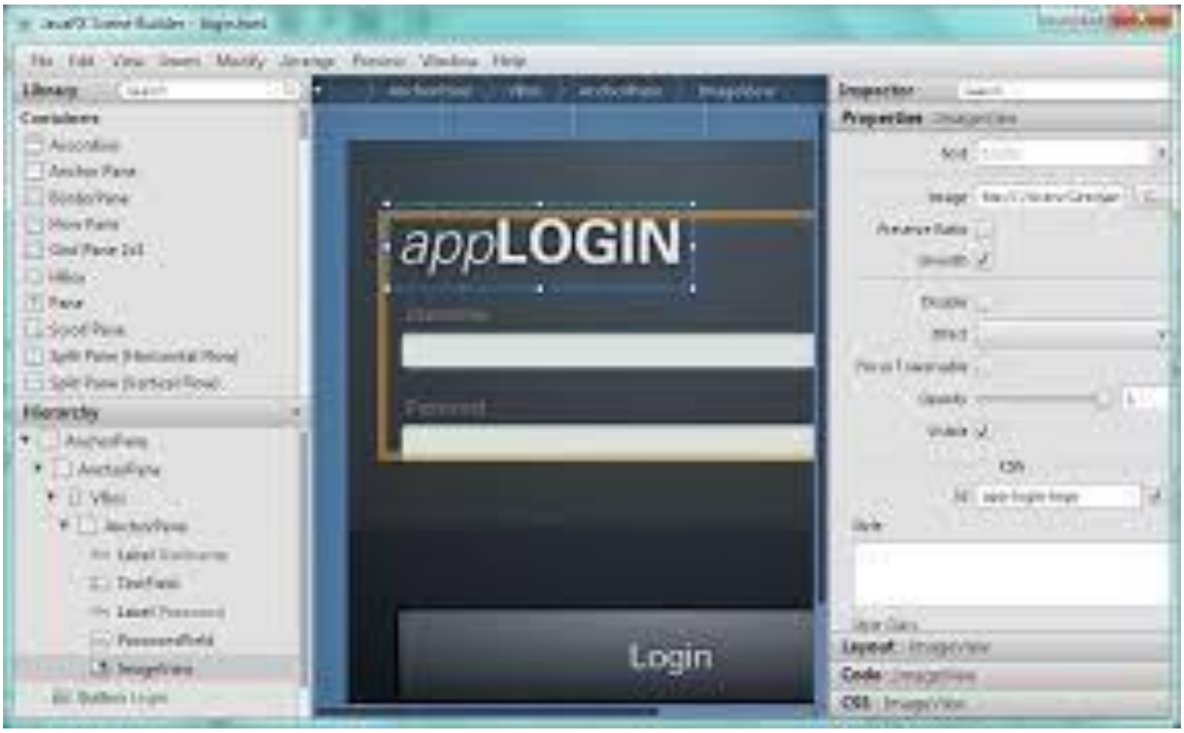

Custom Java - SHST/src/com/sh/st/HttpRequest.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Custom Java
Package Explorer
JRE System Library [JavaSE-1.7]
resources.jar - C:\Program Files\Java\jre7\
rt.jar - C:\Program Files\Java\jre7\lib
jsse.jar - C:\Program Files\Java\jre7\lib
jce.jar - C:\Program Files\Java\jre7\lib
javax.crypto
javax.crypto.interfaces
javax.crypto.spec
sun.security.internal.interfaces
sun.security.internal.spec
META-INF
charsets.jar - C:\Program Files\Java\jre7\li
jfr.jar - C:\Program Files\Java\jre7\lib
access-bridge-64.jar - C:\Program Files\Ja
dnsns.jar - C:\Program Files\Java\jre7\lib\
jaccess.jar - C:\Program Files\Java\jre7\lib
localedata.jar - C:\Program Files\Java\jre7
sunec.jar - C:\Program Files\Java\jre7\lib\
sunjce_provider.jar - C:\Program Files\Jav
sunmscapi.jar - C:\Program Files\Java\jre7
zipfs.jar - C:\Program Files\Java\jre7\lib\e
googleHttp
google-http-java-client-1.15.0-rc.zip
dependencies
dependencies.css
dependencies.images
dependencies.images.logos
libs
commons-logging-1.1.1.jar
google-http-client-1.15.0-rc.jar
google-http-client-1.15.0-rc.jar.pro
google-http-client-android-1.15.0-
google-http-client-android-1.15.0-
google-http-client-appengine-1.1!
*HttpRequest.java
Main.java
Workspace Log
type filter text
Message
google-http-java-client-1.15.0-rc.zip - SHST
Sign in to Google...

```java
import java.net.URL;
import java.net.URLConnection;
import java.util.logging.Handler;
import java.util.logging.Level;
import java.util.logging.LogRecord;
import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class HttpRequest extends EventSearch
{
    String query;
    URL url;
    Logger logger = Logger.getLogger(HttpTransport.class.getName());
    logger.setLevel(Level,CONFIG);

public HttpRequest(String query) throws IOException{
    // Fixed search URL; drop openConnection() at the end

    try {



            logger.addHandler(new Handler() {

             @Override
             public void close() throws SecurityException {
             }

             @Override
             public void flush() {
             }

             @Override
             public void publish(LogRecord record) {
               // default ConsoleHandler will print >= INFO to System.err
               if (record.getLevel().intValue() < Level.INFO.intValue()) {
                 System.out.println(record.getMessage());
               }
             }
```

The File class is Java’s representation of a file or directory path name. Because file and directory names have different formats on different platforms, a simple string is not adequate to name them. The File class contains several methods for working with the path name, deleting and renaming files, creating new directories, listing the contents of a directory, and determining several common attributes of files and directories.

- It is an abstract representation of file and directory pathnames.
- A pathname, whether abstract or in string form can be either absolute or relative. The parent of an abstract pathname may be obtained by invoking the getParent() method of this class.
- First of all, we should create the File class object by passing the filename or directory name to it. A file system may implement restrictions to certain operations on the actual file-system object, such as reading, writing, and executing. These restrictions are collectively known as access permissions.
- Instances of the File class are immutable; that is, once created, the abstract pathname represented by a File object will never change.

JavaFx



JavaFX is a GUI toolkit for Java (GUI is short for Graphical User Interface). JavaFX makes it easier to create desktop applications and games in Java. This JavaFX tutorial is a multi-page tutorial explaining the core features of JavaFX. See the menu in the left side of this page to see all the topics covered in this JavaFX tutorial.

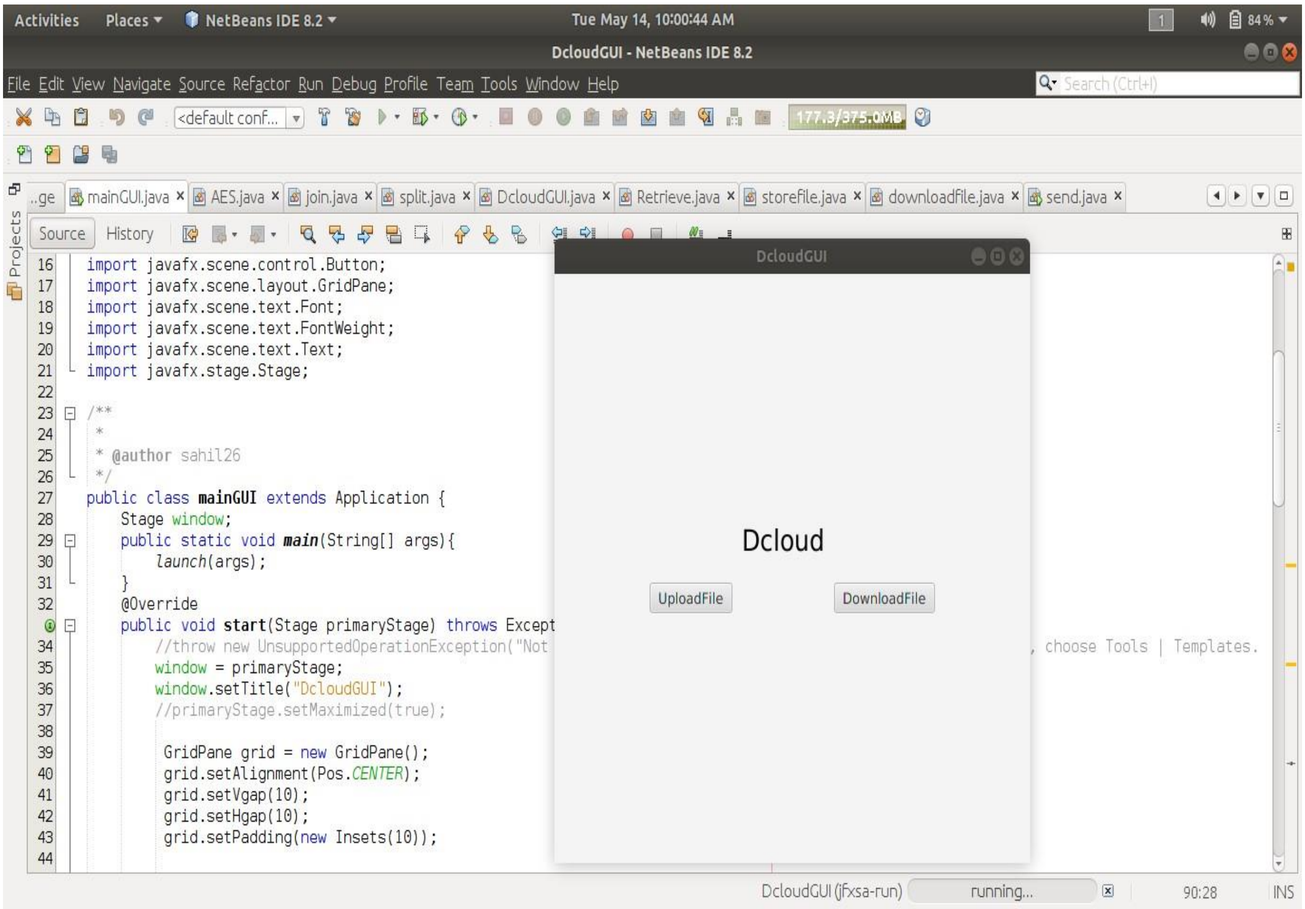
Some applications are just easier to create as standalone desktop applications than as web applications. For instance, applications that need to access the local disk of the computer it runs on, or which needs to communicate with many different remote systems, and sometimes using other protocols than HTTP (e.g. IAP or streaming protocols etc.). JavaFX is a good option in these cases. We at Nanosai are actually developing a desktop app using JavaFX for these exact reasons. See JavaFX use cases for more examples.

JavaFX has replaced Swing as the recommended GUI toolkit for Java. Furthermore, JavaFX is more consistent in its design than Swing, and has more features. It is more modern too, enabling you to design GUI using layout files (XML) and style them with CSS, just like we are used to with web applications. JavaFX also integrates 2D + 3D graphics, charts, audio, video, and embedded web applications into one coherent GUI toolkit.

The Gluon company has ported JavaFX so it can run on both Android and iOS.
See JavaFX on Mobile Devices for more information.

Application

1)



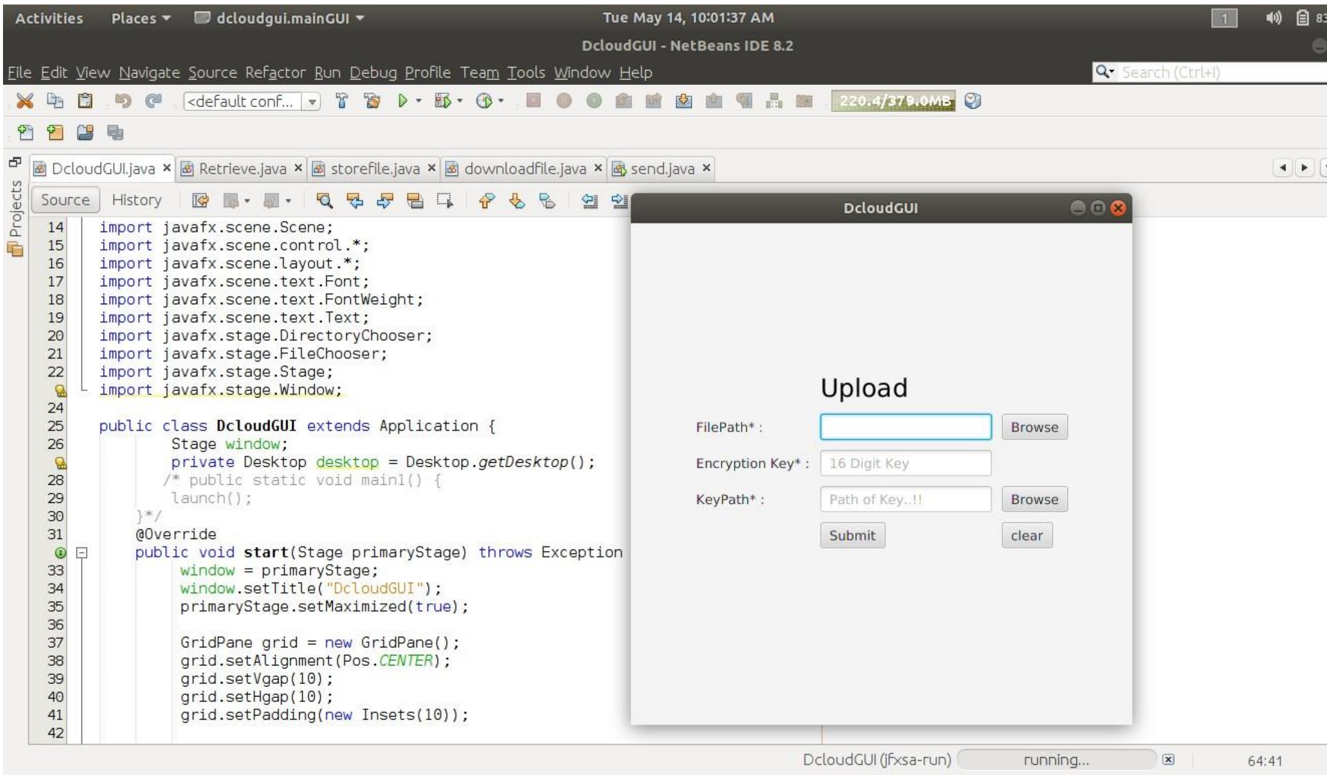
```java
import javafx.scene.control.Button;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.Stage;

/**
 *
 * @author sahil26
 */
public class mainGUI extends Application {
    Stage window;
    public static void main(String[] args){
        launch(args);
    }
    @Override
    public void start(Stage primaryStage) throws Except
        //throw new UnsupportedOperationException("Not                                      , choose Tools | Templates.
        window = primaryStage;
        window.setTitle("DcloudGUI");
        //primaryStage.setMaximized(true);

         GridPane grid = new GridPane();
         grid.setAlignment(Pos.CENTER);
         grid.setVgap(10);
         grid.setHgap(10);
         grid.setPadding(new Insets(10));

```

As the application is started two options are provided

Upload File: This is to upload a file that is to be safely stored on the decentralized cloud.

Download File: This is to download an already stored file by decrypting and reassembling it.

2)



```java
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.DirectoryChooser;
import javafx.stage.FileChooser;
import javafx.stage.Stage;
import javafx.stage.Window;

public class DcloudGUI extends Application {
        Stage window;
        private Desktop desktop = Desktop.getDesktop();
       /* public static void main1() {
        launch();
    }*/
    @Override
    public void start(Stage primaryStage) throws Exception
         window = primaryStage;
         window.setTitle("DcloudGUI");
         primaryStage.setMaximized(true);

         GridPane grid = new GridPane();
         grid.setAlignment(Pos.CENTER);
         grid.setVgap(10);
         grid.setHgap(10);
         grid.setPadding(new Insets(10));

```

Upload File

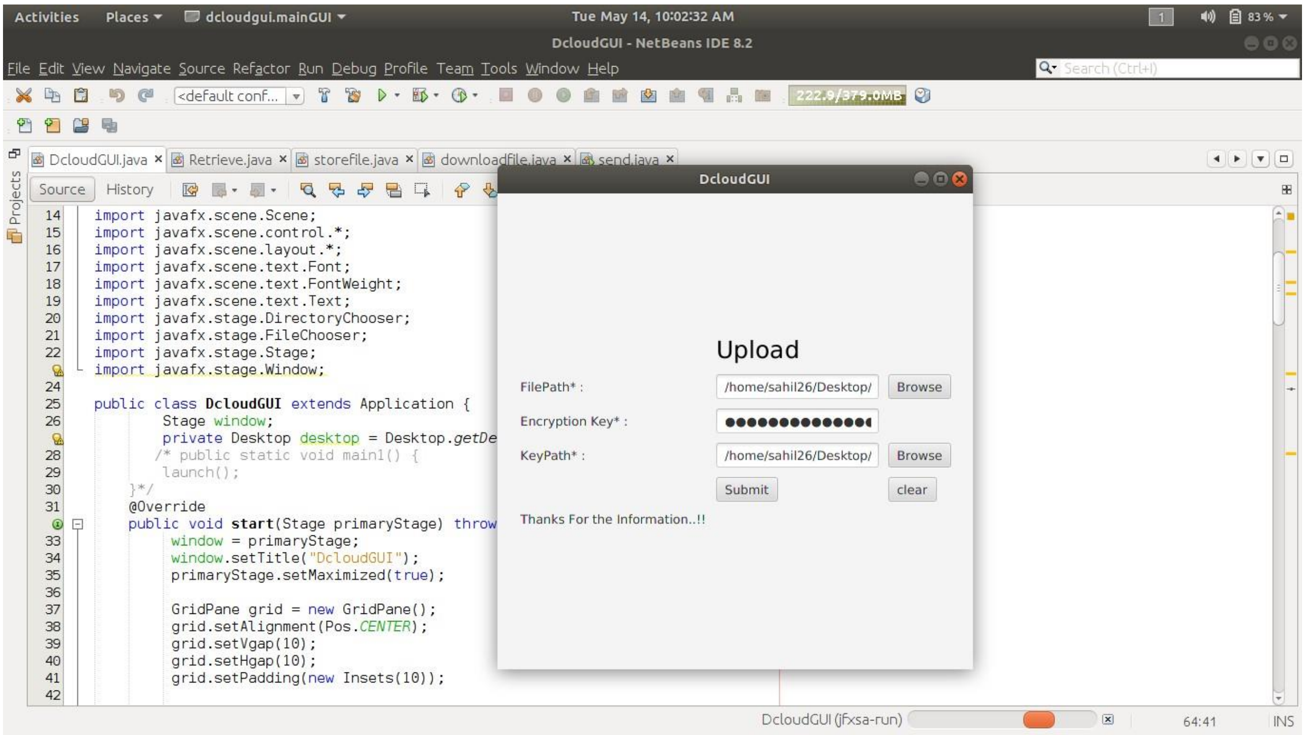
On selecting this option, a pop-up screen appears asking for following inputs:

File path: Here the path of the file that is to be stored is specified.

Encryption Key: The user has to enter a 16 bit password for the AES encryption.

KeyPath: This is the location where the returned key is to be stored.

3)



```
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.scene.text.FontWeight;
import javafx.scene.text.Text;
import javafx.stage.DirectoryChooser;
import javafx.stage.FileChooser;
import javafx.stage.Stage;
import javafx.stage.Window;

public class DcloudGUI extends Application {
        Stage window;
        private Desktop desktop = Desktop.getDe
       /* public static void main1() {
        launch();
    }*/
    @Override
    public void start(Stage primaryStage) throw
          window = primaryStage;
          window.setTitle("DcloudGUI");
          primaryStage.setMaximized(true);

          GridPane grid = new GridPane();
          grid.setAlignment(Pos.CENTER);
          grid.setVgap(10);
          grid.setHgap(10);
          grid.setPadding(new Insets(10));
```

Upload

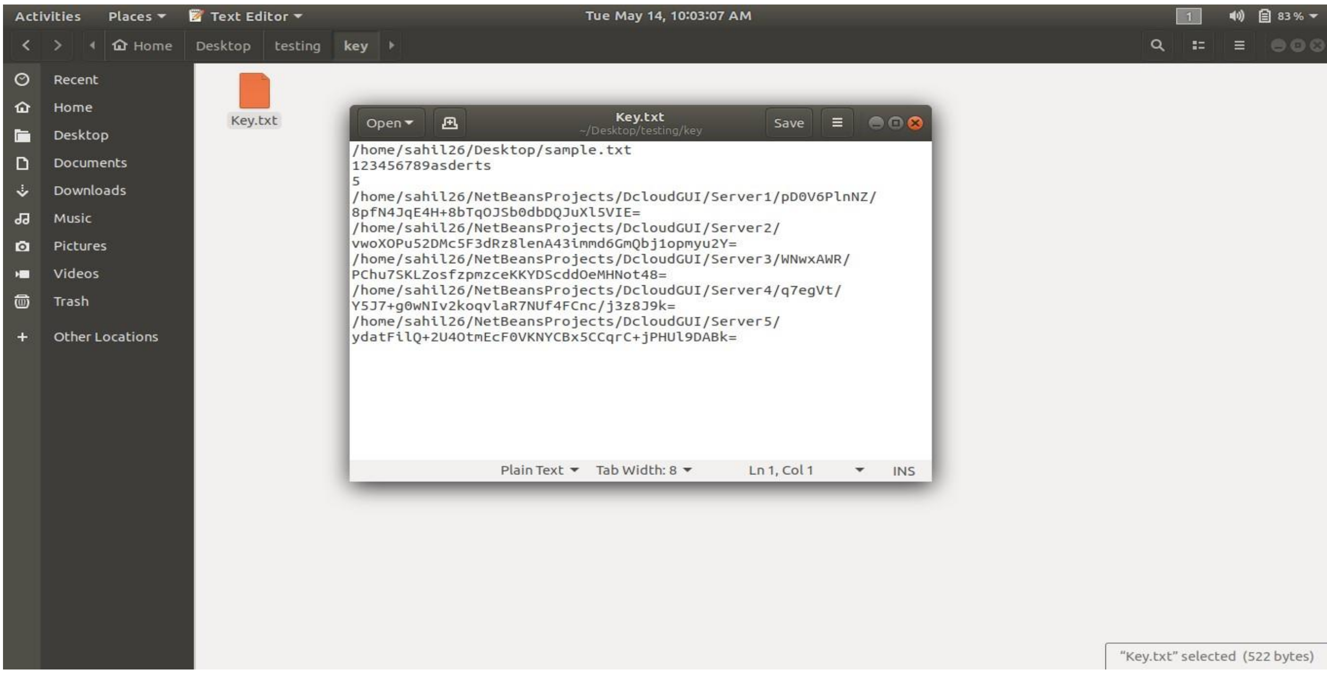
FilePath* : /home/sahil26/Desktop/

Encryption Key* : ●●●●●●●●●●●●●●

KeyPath* : /home/sahil26/Desktop/

Thanks For the Information..!!

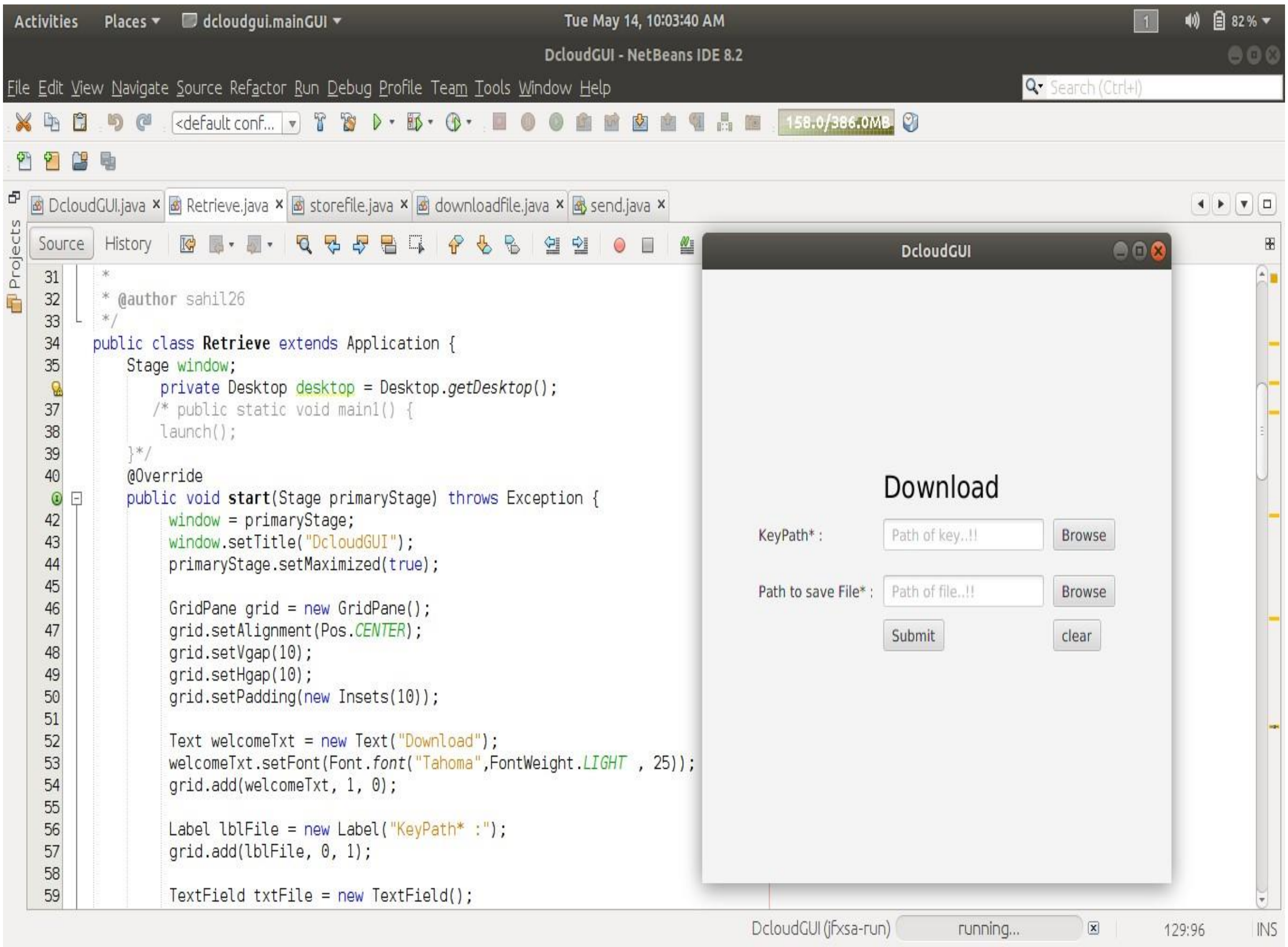
After filling all the mandatory fields click the submit button. **4)**



```
/home/sahil26/Desktop/sample.txt
123456789asderts
5
/home/sahil26/NetBeansProjects/DcloudGUI/Server1/pD0V6PlnNZ/
8pfN4JqE4H+8bTqOJSb0dbDQJuXl5VIE=
/home/sahil26/NetBeansProjects/DcloudGUI/Server2/
vwoXOPu52DMc5F3dRz8lenA43immd6GmQbj1opmyu2Y=
/home/sahil26/NetBeansProjects/DcloudGUI/Server3/WNwxAWR/
PChu7SKLZosfzpmzceKKYDScddOeMHNot48=
/home/sahil26/NetBeansProjects/DcloudGUI/Server4/q7egVt/
Y5J7+g0wNIv2koqvlaR7NUf4FCnc/j3z8J9k=
/home/sahil26/NetBeansProjects/DcloudGUI/Server5/
ydatFilQ+2U4OtmEcF0VKNYCBx5CCqrC+jPHUl9DABk=
```

This is how the key file that is returned by the AES appears.

5)



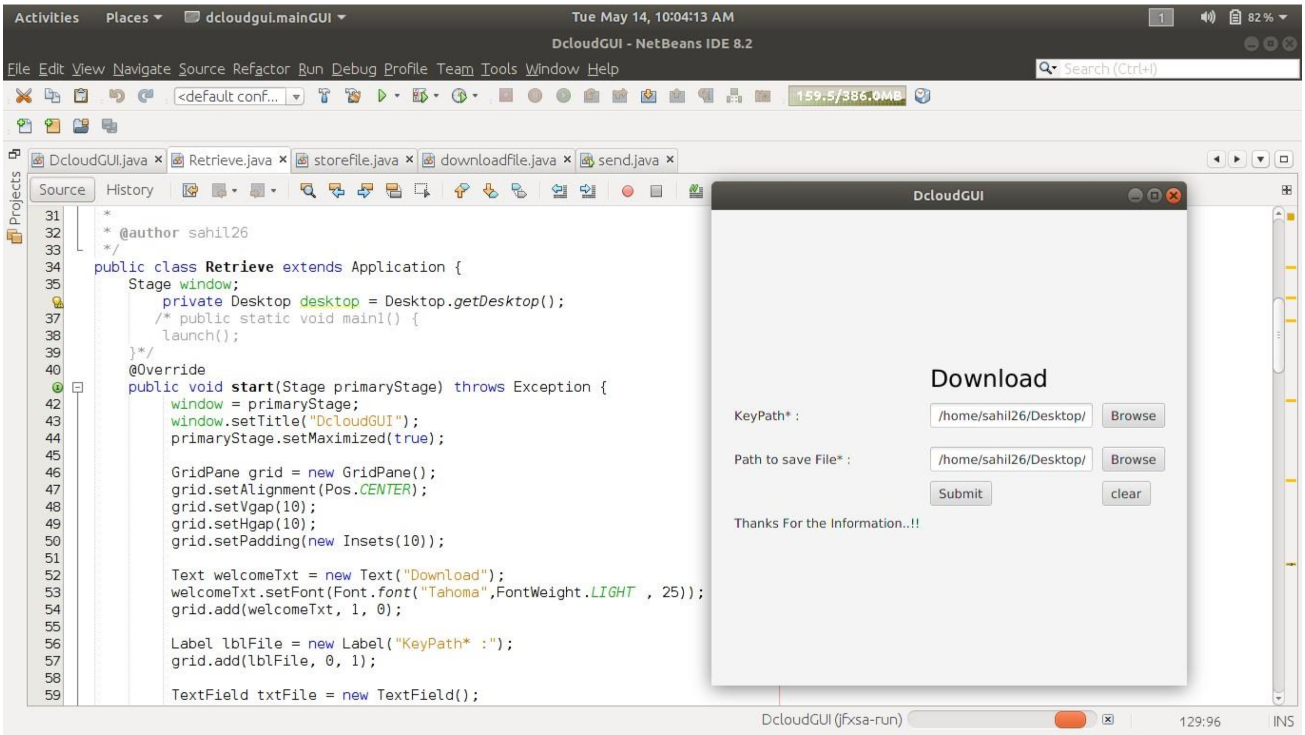
```
 *
 * @author sahil26
 */
public class Retrieve extends Application {
    Stage window;
        private Desktop desktop = Desktop.getDesktop();
       /* public static void main1() {
        launch();
    }*/
    @Override
    public void start(Stage primaryStage) throws Exception {
         window = primaryStage;
         window.setTitle("DcloudGUI");
         primaryStage.setMaximized(true);

         GridPane grid = new GridPane();
         grid.setAlignment(Pos.CENTER);
         grid.setVgap(10);
         grid.setHgap(10);
         grid.setPadding(new Insets(10));

         Text welcomeTxt = new Text("Download");
         welcomeTxt.setFont(Font.font("Tahoma",FontWeight.LIGHT , 25));
         grid.add(welcomeTxt, 1, 0);

         Label lblFile = new Label("KeyPath* :");
         grid.add(lblFile, 0, 1);

         TextField txtFile = new TextField();
```

While downloading the file that is safely stored following mandatory fields are to be input:

Key path: The path of the key file

Path to save the file: This is the location where the assembled decrypted file is stored.

6)



```
 *
 * @author sahil26
 */
public class Retrieve extends Application {
    Stage window;
        private Desktop desktop = Desktop.getDesktop();
       /* public static void main1() {
        launch();
    }*/
    @Override
    public void start(Stage primaryStage) throws Exception {
          window = primaryStage;
          window.setTitle("DcloudGUI");
          primaryStage.setMaximized(true);

          GridPane grid = new GridPane();
          grid.setAlignment(Pos.CENTER);
          grid.setVgap(10);
          grid.setHgap(10);
          grid.setPadding(new Insets(10));

          Text welcomeTxt = new Text("Download");
          welcomeTxt.setFont(Font.font("Tahoma",FontWeight.LIGHT , 25));
          grid.add(welcomeTxt, 1, 0);

          Label lblFile = new Label("KeyPath* :");
          grid.add(lblFile, 0, 1);

          TextField txtFile = new TextField();
```

DcloudGUI

Download

KeyPath* : /home/sahil26/Desktop/ Browse

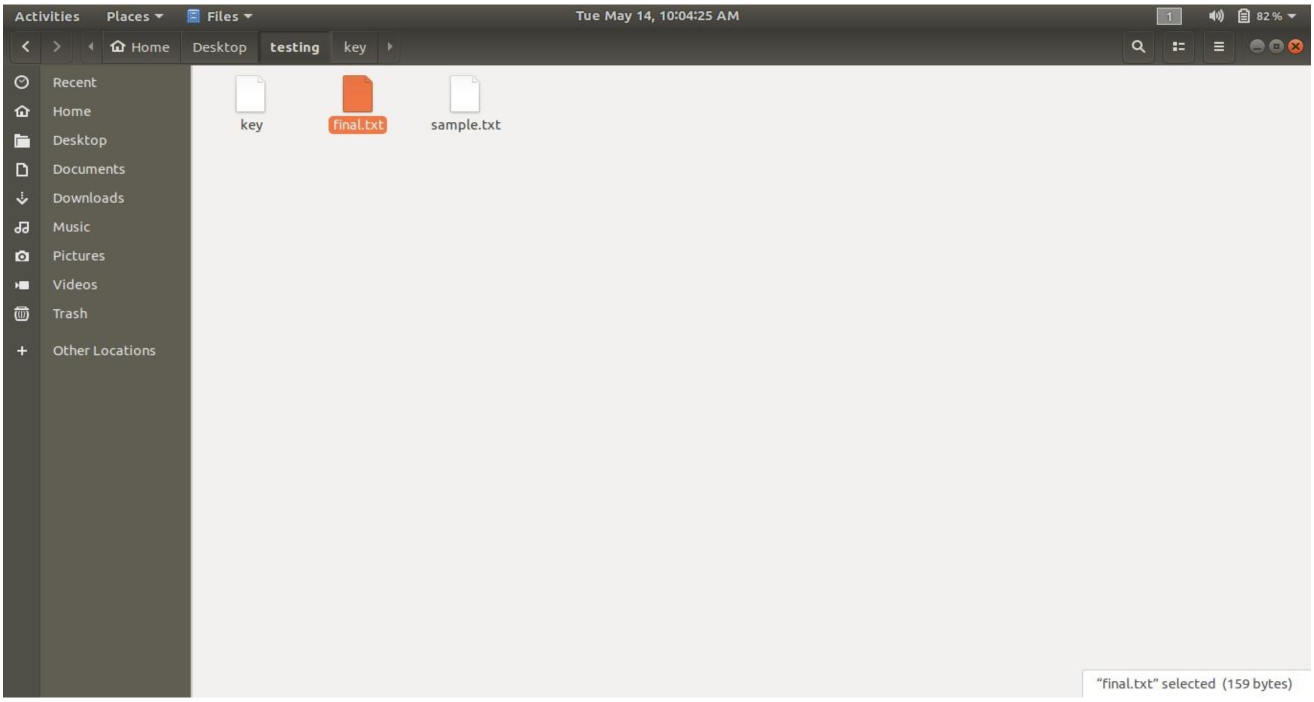
Path to save File* : /home/sahil26/Desktop/ Browse

Submit clear

Thanks For the Information..!!

After filling the necessary inputs, click the submit button.

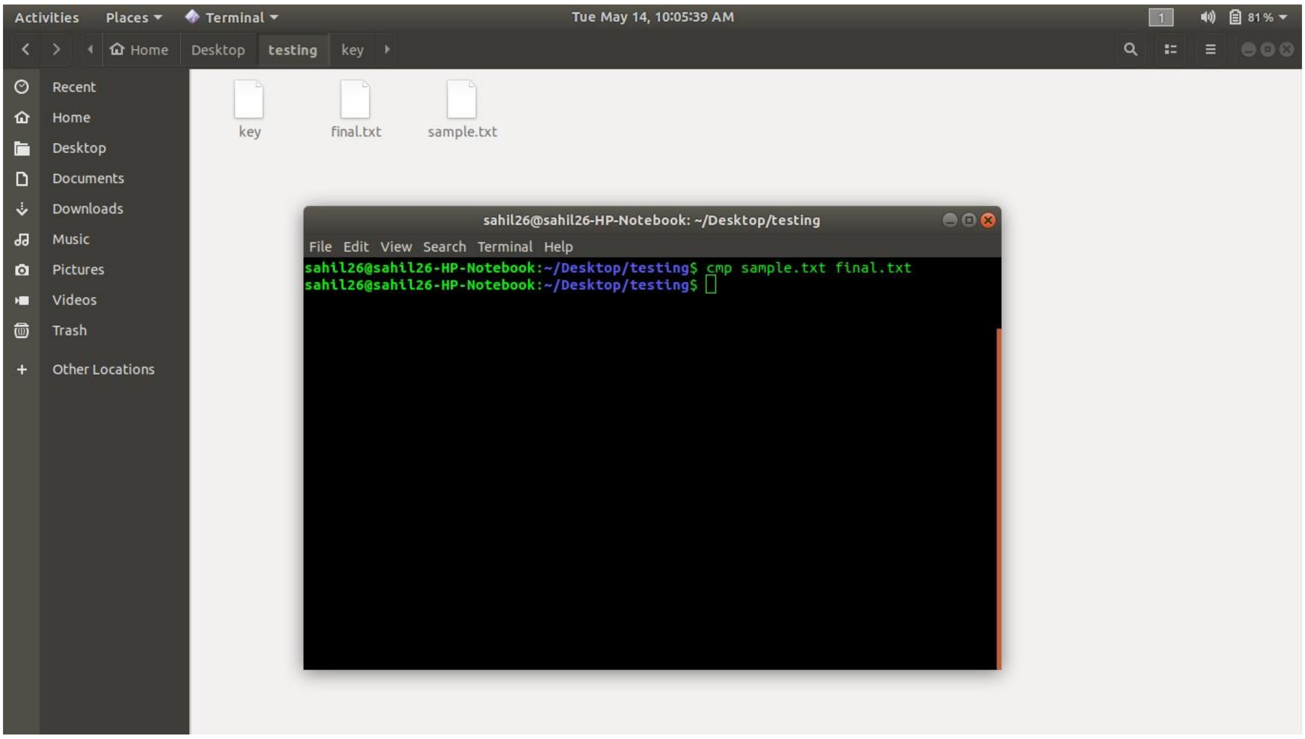
7)



“final.txt” selected (159 bytes)

This is how the file after retrieval appears.

8)



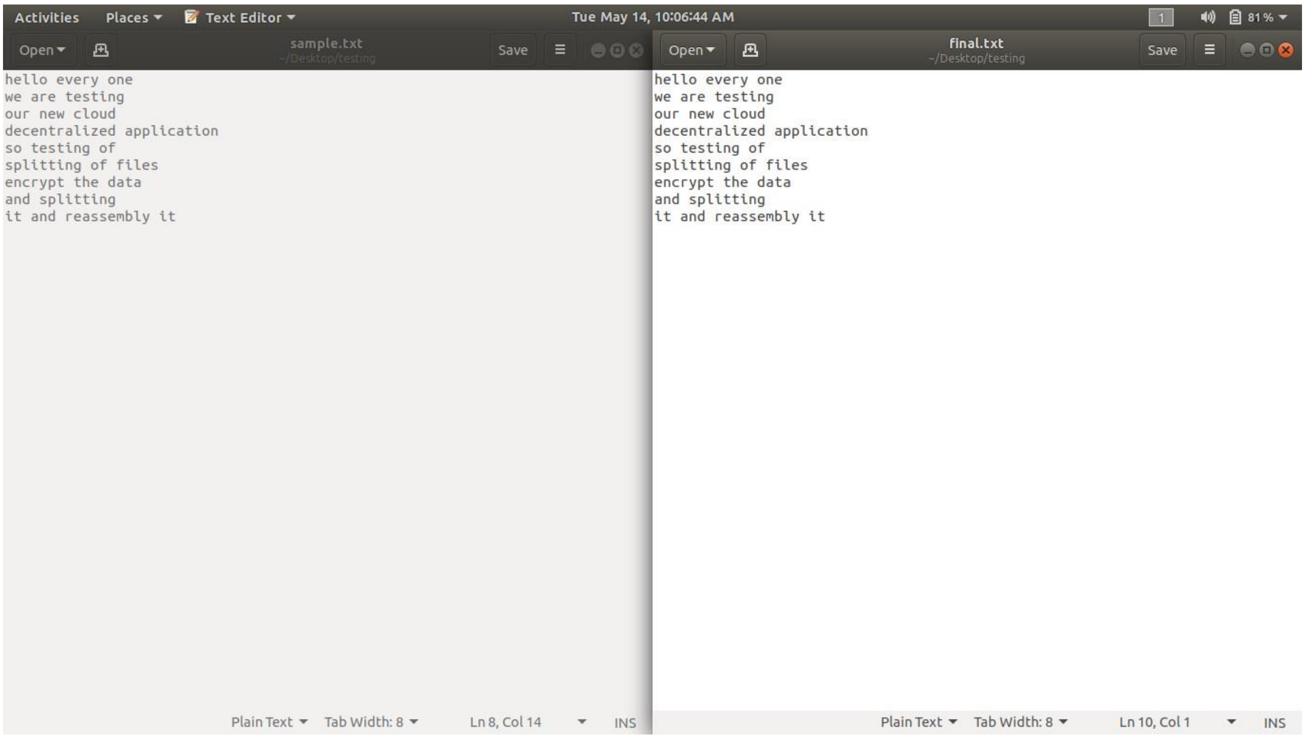
```
sahil26@sahil26-HP-Notebook:~/Desktop/testing$ cmp sample.txt final.txt
sahil26@sahil26-HP-Notebook:~/Desktop/testing$ 
```

On comparing the raw file to the file after retrieval we observer that there no changes in the file after saving it on the cloud.

cmp command in Linux/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not.

- When cmp is used for comparison between two files, it reports the location of the first mismatch to the screen if difference is found and if no difference is found i.e the files compared are identical.
- cmp displays no message and simply returns the prompt if the the files compared are identical.

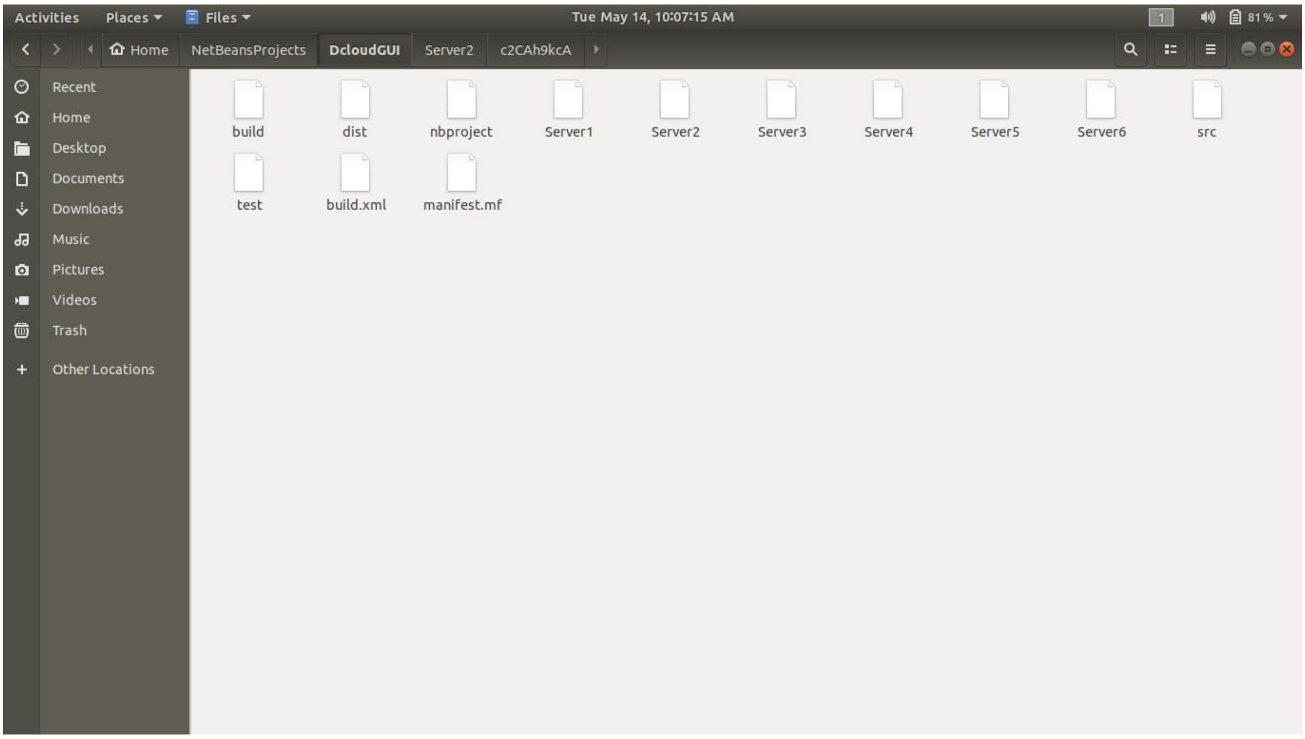
9)



```
hello every one
we are testing
our new cloud
decentralized application
so testing of
splitting of files
encrypt the data
and splitting
it and reassembly it
```

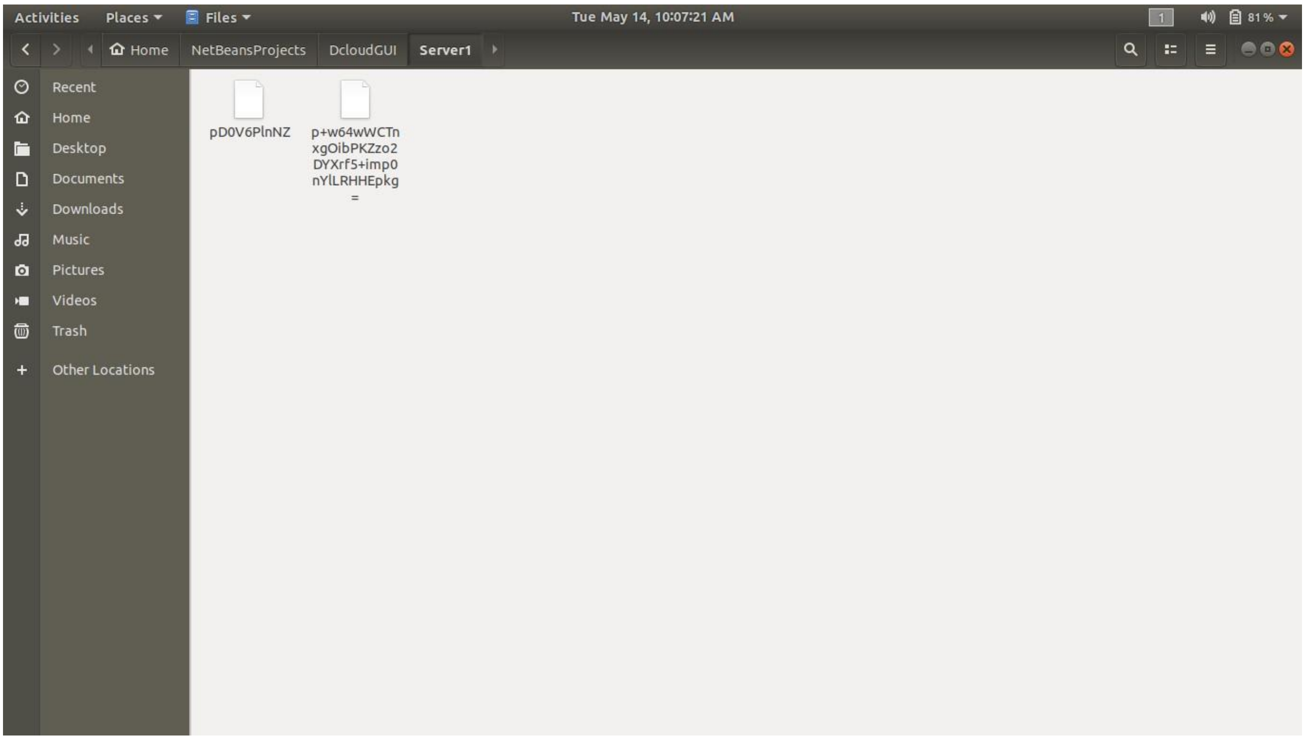
```
hello every one
we are testing
our new cloud
decentralized application
so testing of
splitting of files
encrypt the data
and splitting
it and reassembly it
```

10)



The original file split into multiple files.

11)



The key file after saving the original file.

Advantages

1.) Even if a cloud server is compromised, data still remains the safe.
2.) In case of failure of any cloud storage server. There is very loss of only a chunk of data which can also be retrieved because multiple copies of data are made.
3.) No need to apply expensive load balancing technique because data is already at multiple locations.
4.) Natural calamities or disaster have minimum to no effects on efficiency of data storage and retrieval system.
5.) Maximum bandwidth is achieved while uploading and downloading data due to multiple connections formations.
6.) Saves money and time to build gigantic data storage warehouses.

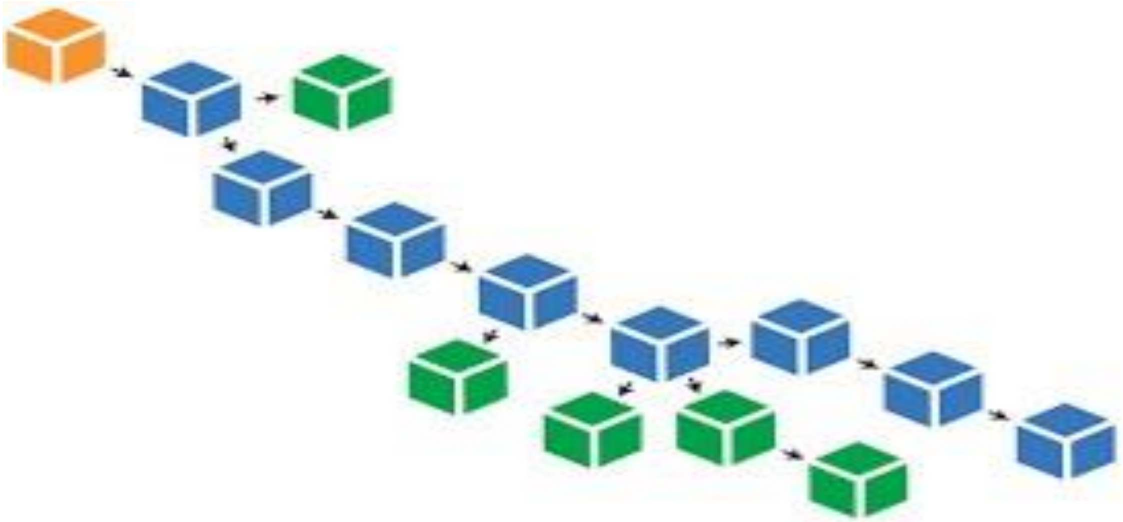
Future Scope

As a P2P Decentralized cloud

Using Block Chain Technology

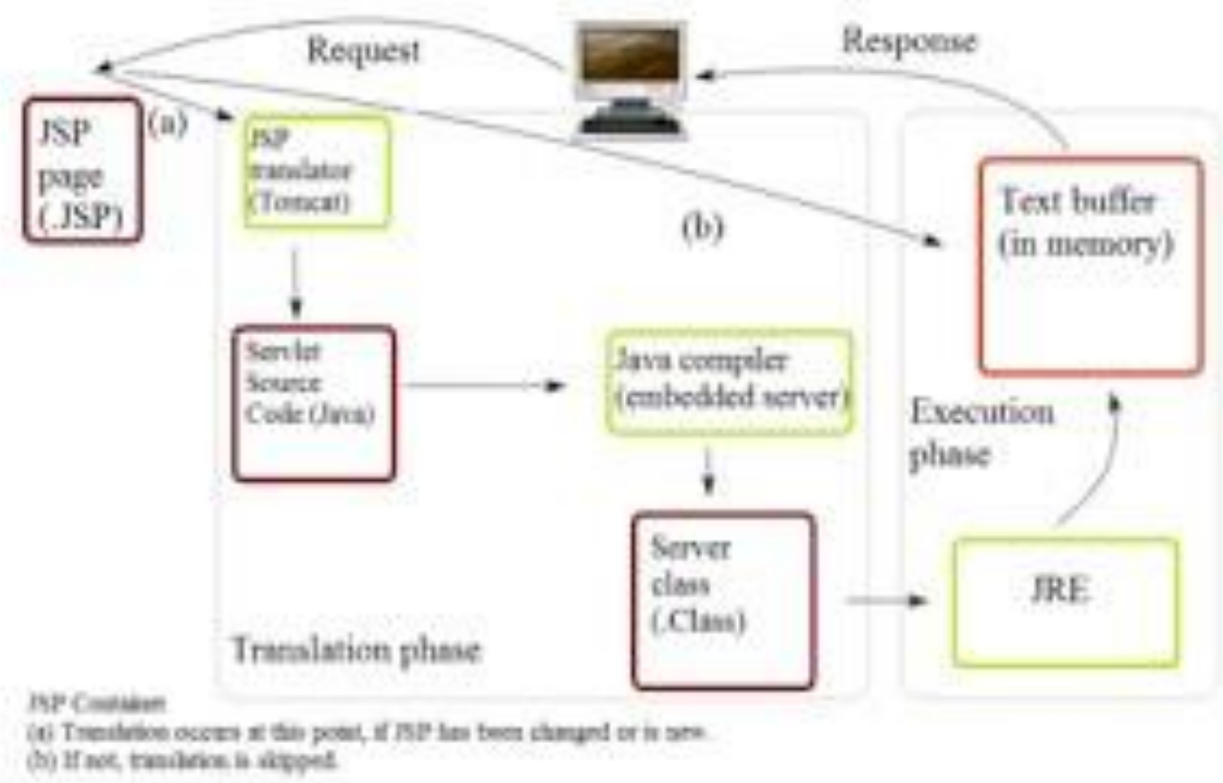
Blockchain technology is not a company, nor is it an app, but rather an entirely new way of documenting data on the internet. The technology can be used to develop blockchain applications, such as social networks, messengers, games, exchanges, storage platforms, voting systems, prediction markets, online shops and much more. In this sense, it is similar to the internet, which is why some have dubbed it “The Internet 3.0”.

The information recorded on a blockchain can take on any form, whether it be denoting a transfer of money, ownership, a transaction, someone's identity, an agreement between two parties, or even how much electricity a lightbulb has used. However, to do so requires a confirmation from several of devices, such a computer, on the network. Once an agreement, otherwise known as a consensus, is reached between these devices to store something on a blockchain it is unquestionably there, it cannot be disputed, removed or altered, without the knowledge and permission of those who made that record, as well as the wider community.



Rather than keeping information in one central point, as is done by traditional recording methods, multiple copies of the same data are stored in different locations and on different devices on the network, such as computers or printers. This is known as a P (P2P) network. This means that even if one point of storage is damaged or lost, multiple copies remain safe and secure elsewhere. Similarly, if one piece of information is changed without the agreement of the rightful owners, there are countless other examples in existence, where the information is true, making the false record obsolete.

Java Servlet


Request
Response
JSP page (.JSP)
(a)
JSP translator (Tomcat)
(b)
Text buffer (in memory)
Servlet Source Code (Java)
Java compiler (embedded server)
Execution phase
Server class (.Class)
JRE
Translation phase
JSP Container
(a) Translation occurs at this point, if JSP has been changed or is new.
(b) If not, translation is skipped.

A servlet is a Java Programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. It is also a web component that is deployed on the server to create a dynamic web page.

Before servlets, we had **CGI** i.e. **C**ommon **G**ateway **I**nterface. It is a standard way for a Web server to pass a user's request to an application program and receives the response to forward to the user. When the user requests a Web page, the server sends back the requested page. However, when a user fills out a form on a Web page and sends it in, it is processed by an application program. The Web server typically passes the form information to a small application program. This program processes the data and sends back a confirmation message. This process of passing data back and forth between the server and the application is called the common gateway interface (CGI). It is part of the Web's Hypertext Transfer Protocol.

Advantages:

1) Man in the middle attack becomes useless.

2) DOS and DDOS attacks become useless.

References

1.) https://ieeexplore.ieee.org/document/7813828

2.) https://hackernoon.com/research-on-decentralized-file-storage-and-sharing-on-the-blockchain-f3a224c4c85b

3.) https://searchsecurity.techtarget.com/definition/encryption

4.) https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard

5.) https://www.geeksforgeeks.org/hashing-data-structure/

6.) https://searchsqlserver.techtarget.com/definition/hashing

7.) https://www.geeksforgeeks.org/socket-programming-in-java/

8.) https://medium.com/bitfwd/what-is-decentralised-storage-ipfs-filecoin-sia-storj-swarm-5509e476995f