IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

Volume 10, Issue 5, May 2021

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.122

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 7.122|| A Monthly Peer Reviewed & Referred Journal |

|| Volume 10, Issue 5, May 2021 ||

DOI:10.15662/IJAREEIE.2021.1005039

Implementation of Control Privilege and Anonymity with Fully Anonymous Attribute-Based Encryption Based on Cloud Data Access

Prof.Poornanand Dubey, Prof.Nitesh Shukla

Department of Electronics & Communication Engineering, Global Nature Care Sanghthan's Group of Institutions, Jabalpur (M.P.), India

ABSTRACT: In a cipher text approach characteristic based encryption framework, a client'sprivate key is connected with an arrangement of qualities (portraying the client) and a scrambled cipher text will indicate an entrance strategy over traits. A client will have the capacity to decode if and just if his properties fulfill the figure content's arrangement. In this work, we show the primary development of a cipher text-approach characteristic based encryption plan having security evidence in view of a number theoretic presumption and supporting propelled access structures. Past CP-ABE frameworks could either bolster just exceptionally constrained access structures or had a proof of security just in the non specific gathering model. Our development can bolster access structures which can be spoken to by a limited size access tree with edge doors as its hubs. The bound on the extent of the entrance trees is picked at the season of the framework setup. Our security evidence depends on the standard Decisional Bilinear Daffier Hellman suspicion.

I. INTRODUCTION

Distributed computing is a progressive registering procedure, by which processing assets are given powerfully by means of Internet and the information stockpiling and calculation are outsourced to somebody or some gathering in a 'cloud'. It extraordinarily pulls in consideration and enthusiasm from both the scholarly world and industry because of the gainfulness, however it likewise has no less than three difficulties that must be taken care of before going to our genuine to the best of our insight. Most importantly, information privacy ought to be ensured. The information security is not just about the information substance. Since the most appealing part of the distributed computing is the calculation outsourcing, it is a long ways sufficiently past to simply lead an entrance control. More probable, clients need to control the benefits of information control over different clients or cloud servers. This is on the grounds that when delicate data or calculation is outsourced to the cloud servers or another client, which is out of clients' control much of the time, protection dangers would rise significantly on the grounds that the servers

may unlawfully investigate clients' information and access touchy data, or different clients may have the capacity to derive delicate data from the outsourced calculation. Accordingly, the entrance as well as the operation ought to be controlled. Also, individual data (characterized by every client's properties set) is at danger since one's personality is confirmed in light of his data with the end goal of access control (or benefit control in this paper). As individuals are turning out to be more worried about their character security nowadays, the personality protection likewise should be ensured before the cloud enters our life. Ideally, any power or server alone ought not know any customer's close to home data. To wrap things up, the distributed computing framework ought to be flexible on account of security rupture in which some a player in the framework is bargained by assailants. Different methods have been proposed to secure the information substance protection by means of access control. Character based encryption (IBE) was initially presented by Shamir, in which the sender of a message can indicate a personality to such an extent that lone a beneficiary with coordinating character can unscramble it. Couple of years after the fact, Fuzzy Identity-Based Encryption is proposed, which is otherwise called Attribute-Based Encryption (ABE). In such encryption plot, a character is seen as an arrangement of illustrative qualities, and decoding is conceivable if a descriptor's personality has a few covers with the one determined in the cipher text. Before long, more broad tree-based ABE plans,

II. JAVA CLOUD CODING

JTabbedPaneclass: TheJTabbedPanecontainer allows many panels to occupy the same area of the interface, and the user may select which to show by clicking on a tab.

Constructor

```
JTabbedPanetp = new JTabbedPane();
```

Adding tabs to the J Tabbed Pane

Add tabs to a tabbed pane by calling addTab and passing it a String title and aninstance of a class which should be called when we pressed a tab. That class should be a subclass of JPanel.
addTab(“String”,instance);

Example program:

```
}

}

class Count extends JPanel

{

Count()

{

JButton b1 = new JButton("India");

JButton b2 = new JButton("SriLanka");

JButton b3 = new JButton("Australia");

add(b1);

add(b2);

import javax.swing.*;

import java.awt.*;

public class TabbedPaneDemo extends JFrame

{

TabbedPaneDemo()

{

setLayout(new

FlowLayout(FlowLayout.LEFT)); setTitle("Tabbed Demo");
setVisible(true);

setSize(500,500);
                                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JTabbedPane pane = new JTabbedPane();

pane.addTab("Countries",new Count());
pane.addTab("Cities",newCit());
add(pane);
}
public static void main(String a[])
                                        add(b3);
}
}
class Cit extends JPanel
{
Cit()
{
JCheckBox cb1 = new JCheckBox("Hyderabad");
JCheckBox cb2 = new JCheckBox("Banglore");
JCheckBox cb3 = new JCheckBox("Pune");
                                        add(cb1);
                                        add(cb2);
                                        add(cb3);
}
}
```

2.1 JMenuBar, JMenu, JMenuItem

```
{

new TabbedPaneDemo();
```

A top-level window can have a menu bar connected with it. A menu bar shows a rundown of top-level menu decisions. Every decision is connected with a drop-down menu. This idea is executed in Java by the accompanying classes: JMenuBar, JMenu, and JMenuItem. By and large, a menu bar contains one or more JMenu articles. Each JMenu object contains a rundown of JMenuItem articles. Each JMenuItem object speaks to something that can be chosen by the client. To make a menu bar, first make an example of JMenuBar. This class just characterizes the default constructor. Next, make occasions of JMenu that will characterize the determinations showed on the bar. Taking after are the constructors for Menu:JMenu( )

JMenu(String optionName)

Here, option Name indicates the name of the menu choice. The principal structure makes a void menu. Singular menu things are of sort Menu Item. It characterizes these constructors : JMenuItem( )

JMenuItem(String itemName)

Here, itemName is the name shown in the menu.

2.2. Sample code

```
package privilege;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JButton;

import javax.swing.JPanel;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;
import javax.swing.UIManager;

import java.awt.BorderLayout;
import java.awt.Dimension;

import java.awt.Color;

import java.awt.Font;

import java.io.File;

import javax.swing.JFileChooser;

import java.io.FileInputStream;

import java.net.Socket;

import java.io.ObjectOutputStream;

import java.io.ObjectInputStream;
```

DOI:10.15662/IJAREEIE.2021.1005039

```
import javax.swing.JOptionPane;

public class DataOwner extends JFrame{

GradientPanel p1;

JPanel p2;

JLabel l1,l2;

JButton b1,b2,b3,b4;

Font f1;

Login login;

JFileChooser chooser;

public DataOwner(Login log){ super("Data Owner Modules"); login = log;

p1 = new GradientPanel(600,200);

p1.setLayout(null);

f1 = new Font("Courier New",Font.BOLD,14);

p2 = new JPanel();

p2.setBackground(new Color(204, 110, 155));

l1 = new

JLabel("<HTML><BODY><CENTER>CONTROL CLOUD DATA ACCESS PRIVILEGE AND ANONYMITY<BR/>WITH

FULLY ANONYMOUS ATTRIBUTE-BASED ENCRYPTION</CENTER></BODY></HTML>");

l1.setFont(new Font("Courier New",Font.BOLD,16));

p2.add(l1);
```

a. DataOwner.java

```
l2 = new JLabel("Data Owner Modules");
l2.setFont(new Font("Courier New",Font.PLAIN,20));
l2.setBounds(180,20,300,30);
p1.add(l2);
b4 = new JButton("Add Data User");
b4.setFont(f1);
b4.setBounds(180,60,240,30);
p1.add(b4);
b4.addActionListener(new ActionListener(){
public void actionPerformed(ActionEve
```

```
Object
req[] = {"upload",file.getName(),enc};
out.writeObject(req);
ObjectInputStream in = new
ObjectInputStream(socket.getInputStream());
Object
res[] = (Object[])in.readObject();
```

```
nt
ae){
                AddUser au = new AddUser();
                au.setVisible(true);
                au.setSize(500,600)
                ;
}
});
b1 = new JButton("Upload File");
b1.setFont(f1);
b1.setBounds(180,110,240,30);
p1.add(b1);
b1.addActionListener(new ActionListener(){
                    actionPerformed(ActionEve
public          void                          nt
ae){
                int    option =
chooser.showOpenDialog(DataOwner.this)
;
                if(option ==
chooser.APPROVE_OPTIO
N){
                    File         file =
chooser.getSelectedFil
e();
                    try{
FileInputStream fin = new
FileInputStream(file);
                                byte    b[] =
new
byte[fin.available()];
fin.read(b,0,b.length);
                                fin.close();
                                byte    enc[]
= AES.encrypt(b);
```

```
JOptionPane.showMessageDialog(DataOwner.this,res[
0].toString());
                    }catch(Exception e){
e.printStackTrace();
                    }
}
}
});
b3 = new JButton("Logout");
b3.setFont(f1);
b3.setBounds(180,160,240,30);
p1.add(b3);
b3.addActionListener(new ActionListener(){
                    actionPerformed(Action
public  void                          Event
ae){
setVisible(false);
login.clear();
login.setVisible(true);
}
});
chooser = new JFileChooser(new File("."));
getContentPane().add(p1,BorderLayout.CENTER);
getContentPane().add(p2,BorderLayout.NORTH);
}
}
```

b.CloudServer.java

```
Socket
package privilege;
socket = new Socket("localhost",3333);
import javax.swing.JFrame;
ObjectOutputStream
import javax.swing.JButton;
out = new ObjectOutputStream(socket.getOutputStream());
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
import javax.swing.UIManager;
import java.awt.BorderLayout;
import java.awt.Dimension;
```

```
table.setFont(f1);
table.setRowHeight(40);
jsp = new JScrollPane(table);
dtm.addColumn("Cloud  Server  Request  Processing
import java.awt.Color;
import java.awt.Font;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.JScrollPane;
import java.net.Socket;
import java.net.ServerSocket;
public class CloudServer extends JFrame{
Font f1;
JTable table;
DefaultTableModeldtm;
JScrollPanejsp;
ServerSocket server;

RequestHandler thread;

public void start(){
try{
```

```
Status");
jsp.getViewport().setBackground(Color.white);
getContentPane().add(jsp,BorderLayout.CENTER);
}
public static void main(String a[])throws Exception{
UIManager.setLookAndFeel("com.sun.java.swing.plaf.
nimbus.NimbusLookAndFeel");
CloudServercs = new CloudServer();
cs.setVisible(true);
cs.setSize(600,550);
new CloudThread(cs);
}
}
```

```
server = new ServerSocket(3333);

Object res[] = {"Cloud Server Started"};

dtm.addRow(res);

while(true){

                                                                    Socket socket = server.accept();

socket.setKeepAlive(true);

thread                              =           new

RequestHandler(socket,dtm);

                                    thread.start();

}

}catch(Exception e){

e.printStackTrace();

}

}

public CloudServer(){

super("Cloud Server Screen");

f1 = new Font("Courier New",Font.BOLD,13); dtm = new DefaultTableModel(){
public booleanisCellEditable(intr,int c){ return false;

}

};

table = new JTable(dtm);
```

III. TESTING

3 .1 EXECUTION AND TESTING

Execution is a standout amongst the most essential errands in venture is the stage in which one must be alerts since every one of the endeavors embraced amid the undertaking will be exceptionally intuitive. Execution is the most critical stage in accomplishing fruitful framework and giving the clients certainty that the new framework is workable and compelling. Every project is tried exclusively at the season of improvement utilizing the specimen information and has confirmed that these projects connect together in the route determined in the system particular. The PC framework and its surroundings are tried as per the general inclination of the client.

3.2 EXECUTION

The execution stage is less imaginative than framework plan. It is essentially worried with client preparing, and document transformation. The framework might require broad client preparing. The underlying parameters of the framework ought to be adjusts as a consequence of a programming. A straightforward working methodology is given so that the client can comprehend the distinctive capacities obviously and rapidly. The diverse reports can be gotten either on the inkjet or spot network printer, which is accessible at the transfer of the client. The proposed framework is anything but difficult to actualize. When all is said in done usage is utilized to mean the way toward changing over another or amended framework plan into an operational one.

3.3 FRAME WORK TESTING

Testing has turned into a fundamental part of any framework or venture particularly in the field of data innovation. The significance of testing is a technique for defending, in the event that one is prepared to move further, be it to be check on the off chance that one is skilled to with stand the rigors of a specific circumstance can't be underplayed and that is the reason testing before improvement is so basic. At the point when the product is created before it is given to client to client the product must be tried whether it is illuminating the reason for which it is created. This testing includes different sorts through which one can guarantee the product is solid. The project was tried coherently and example of execution of the system for an arrangement of information are rehashed. Subsequently the code was comprehensively checked for all conceivable right information and the results were additionally checked.

3.4 MODULE TESTING

To find blunders, every module is tried independently. This empowers us to recognize blunder and right it without influencing some other modules. At whatever point the system is not fulfilling the required capacity, it must be revised to get the required result. Along these lines every one of the modules are separately tried from base up beginning with the littlest and most reduced modules and continuing to the following level. Every module in the framework is tried independently. For instance the occupation characterization module is tried independently. This module is tried with various employment and its surmised execution time and the aftereffect of the test is contrasted and the outcomes that are arranged physically. The examination demonstrates that the outcomes proposed framework works productively than the current framework. Every module in the framework is tried independently. In this framework the asset arrangement and occupation booking modules are tried independently and their relating results are acquired which lessens the procedure holding up time.

3.5 MIX TESTING

After the module testing, the mix testing is connected. While connecting the modules there might be chance for blunders to happen, these mistakes are adjusted by utilizing this testing. In this framework all modules are associated and tried. The testing results are extremely right. In this manner the mapping of occupations with assets is done accurately by the framework**.**

3.6 ACKNOWLEDGMENT TESTING

At the point when that client fined no real issues with its precision, the framework passers through a last acknowledgment test. This test affirms that the framework needs the first objectives, goals and necessities built up amid examination without genuine execution which end wastage of time and cash acknowledgment tests on the shoulders of clients and administration, it is at last worthy and prepared for the operation.

3.7 TEST CASES:

Test Case Id	Test Case Name	Test Case Desc	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
Data Owner Login 01	Admin login	To verify the admin either authorized	If entered wrong details	Error in login	Displayed data owner Modules	High	High

		or not					
Add Data Users 02	Adding Data Users	Data Owner register the data users with different privileges	If leave any field empty when register	Error in registration	Registration process completed	Medium	Medium
Upload File 03	Uploading the file	To access the data, data owner uploading file	If file not uploaded	Users cannot access the files	After uploading we get message like file uploaded successfully	High	High
User Login 04	Login users	To verify the user either registered or not	If we enter wrong details of login	Error in login	It will display the user modules	High	High

Request Key 05	Request for Key	To access the files from cloud keep a request for keys	If we are not keep a request for key	File cannot encrypted or cannot decrypteD	Key received successfully	High	High

Encrypt 06	File Encryption	By using key we can encrypt the file	If we are not keep a request for key	File cannot encrypteD	File encrypted successfully	High	High
Decrypt 07	File Decryption	By using key we can decrypt the file	If we are not keep a request for key	File cannot decrypteD	File Decrypted successfully	High	High
Download 08	File Download	Based on privileges download the decrypted files	If we are not decrypt the file	We cannot download the files	File downloaded	High	High

IV. OUTPUTS OF PROJECT

FORMULATION

Fig.1 Cloud server:

Fig 2.Open the authority1 folder.click on run file

DOI:10.15662/IJAREEIE.2021.1005039

Fig 3.Open the authority2 folder.click on run file

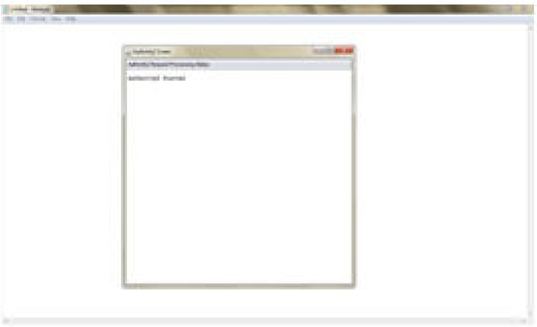


Fig 4 Open user_ownerfolder.click on run file Login screen

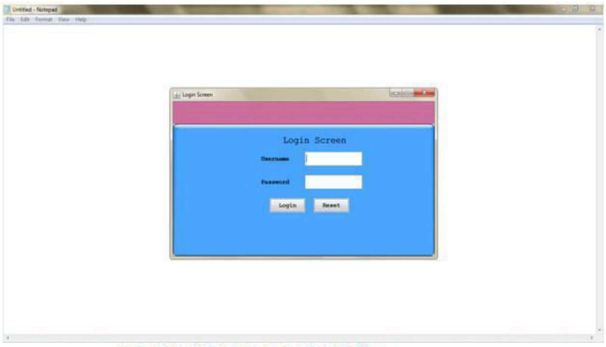
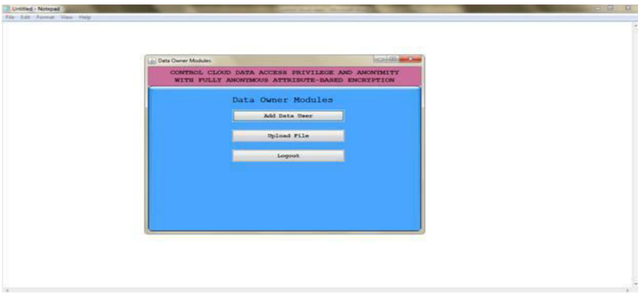


Fig 5 Login as admin.

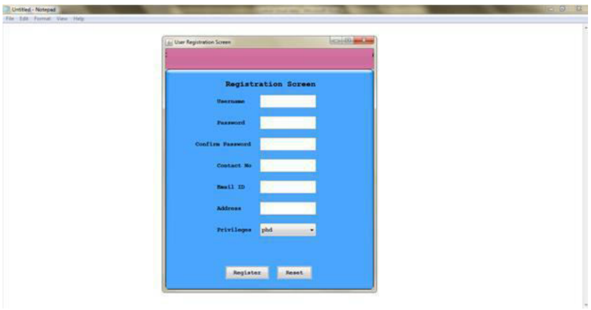
4.1 DATA OWNER MODULES SCREEN

Fig 6 Click on add data user(data owner adds the data users).


Data Owner Modules
CONTROL CLOUD DATA ACCESS PRIVILEGE AND ANONYMITY WITH FULLY ANONYMOUS ATTRIBUTE-BASED ENCRYPTION
Data Owner Modules
Add Data User
Upload File
Logout

DOI:10.15662/IJAREEIE.2021.1005039

Fig 7 User registration screen.



Registration Screen

Username

Password

Confirm Password

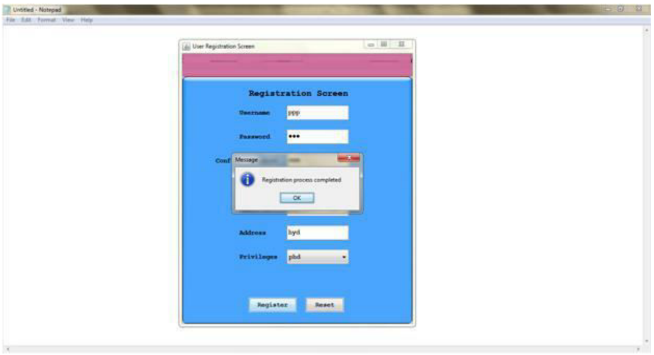
Contact No

Email ID

Address

Privileges phd

Fig 8 Register as phd(privilege)



Registration Screen

Username ppp

Password •••

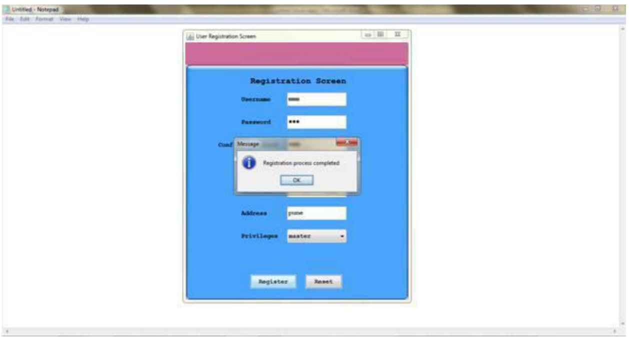
Conf

Registration process completed

Address hyd

Privileges phd

Fig 9 Register as master



Registration Screen

Username mmm

Password •••

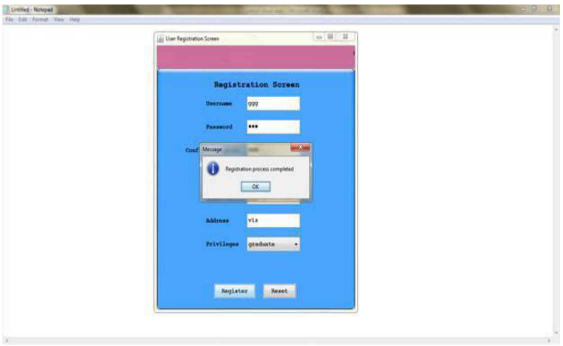
Conf

Registration process completed

Address pune

Privileges master

Fig 10 Register as graduate.



Registration Screen

Username ggg

Password •••

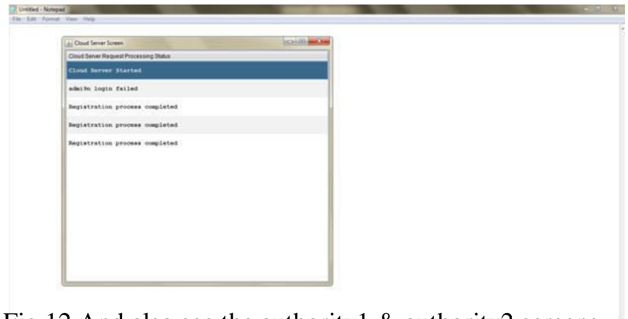
Conf

Registration process completed

Address viz

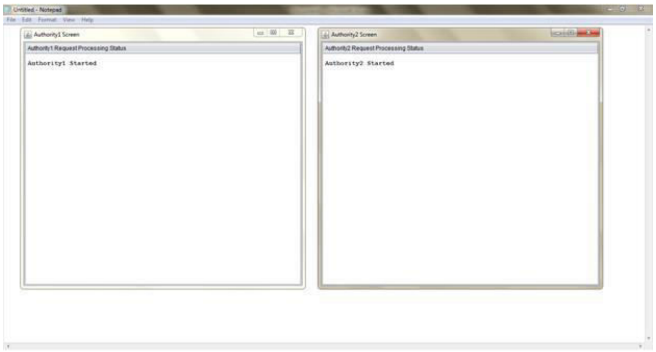
Privileges graduate

Fig 11 After complete the registration process see the cloud server.


Untitled - Notepad
File Edit Format View Help
Cloud Server Screen
Cloud Server Request Processing Status

```
Cloud Server Started
admi9n login failed
Registration process completed
Registration process completed
Registration process completed
```

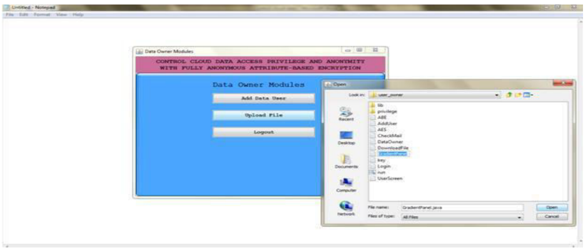
Fig 12 And also see the authority1 & authority2 screens


Untitled - Notepad
File Edit Format View Help
Authority1 Screen
Authority1 Request Processing Status
Authority2 Screen
Authority2 Request Processing Status

```
Authority1 Started
```

```
Authority2 Started
```

Fig 13 Next Click on upload file.



CONTROL CLOUD DATA ACCESS PRIVILEGE AND ANONYMITY WITH FULLY ANONYMOUS ATTRIBUTE-BASED ENCRYPTION

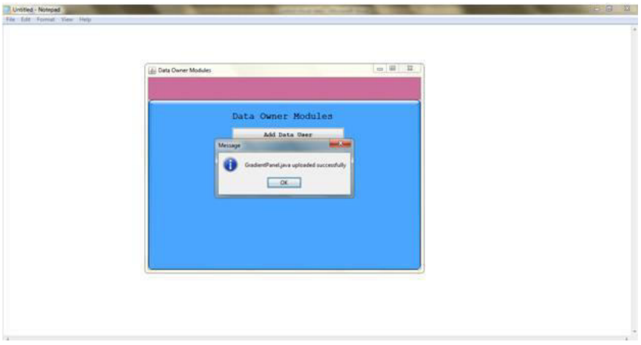
Data Owner Modules

Look in: user_owner

File name: GradientPanel.java

Files of type: All Files

Fig 14 File uploaded successfully.



Data Owner Modules

GradientPanel.java uploaded successfully

Fig 15 Click on logout.

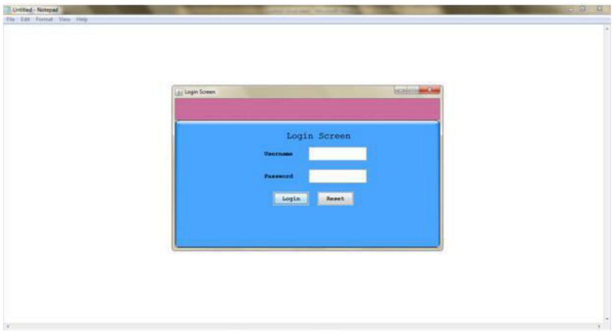
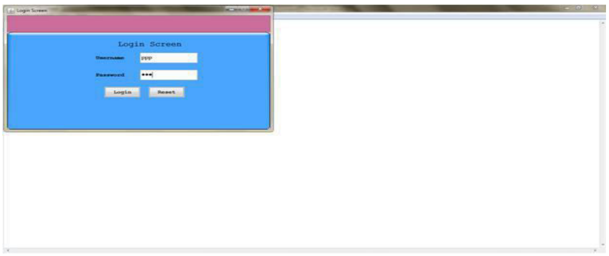


Fig 16 Login as ph d



Username ppp

Password ***

Fig 17 User screen

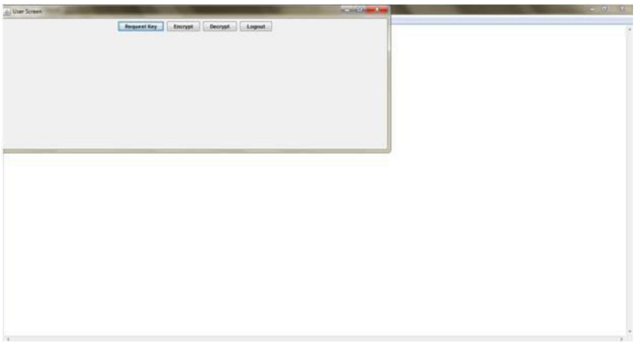
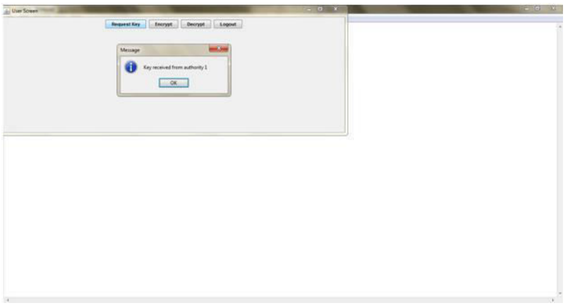
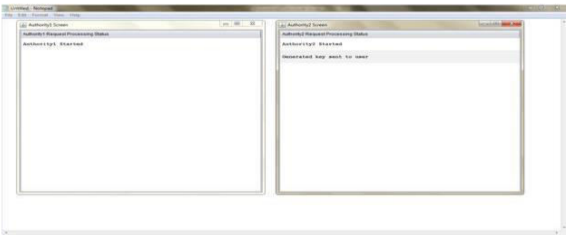


Fig 18 Click on request key utton.(privilege phd asks the request key to N-Authorities for uploaded file).



Key received from authority 1

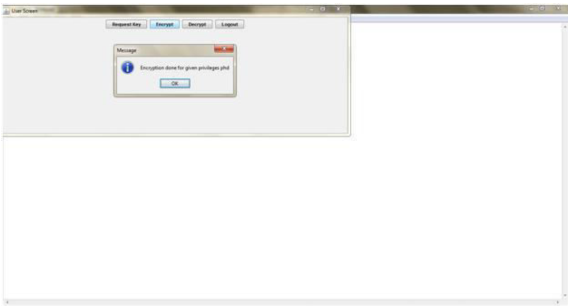
Fig 19 Authority1 generate the key for user, Authority2 send the generate key to user.


Untitled - Notepad
File Edit Format View Help
Authority1 Screen
Authority1 Request Processing Status
Authority2 Screen
Authority2 Request Processing Status

```
Authority1 Started
```

```
Authority2 Started
Generated key sent to user
```

Fig 20 Click on encryption button. the uploaded file converted in ciphertext format(see in clouddata folder).



Encryption done for given privileges phd

Fig 21 Click on decryption button. The encrypted file converted into plaintext format.

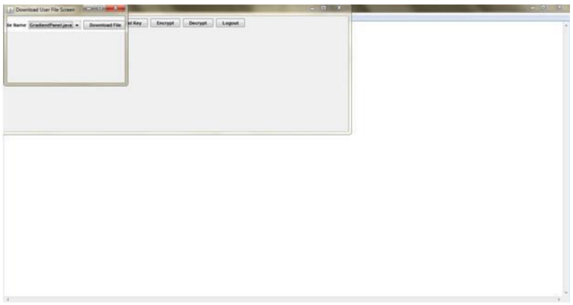
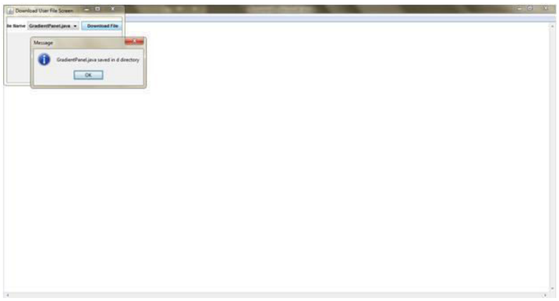


Fig 22 Click on downloadfile button.

Downloaded file stored into d directory.



GradientPanel.java saved in d directory

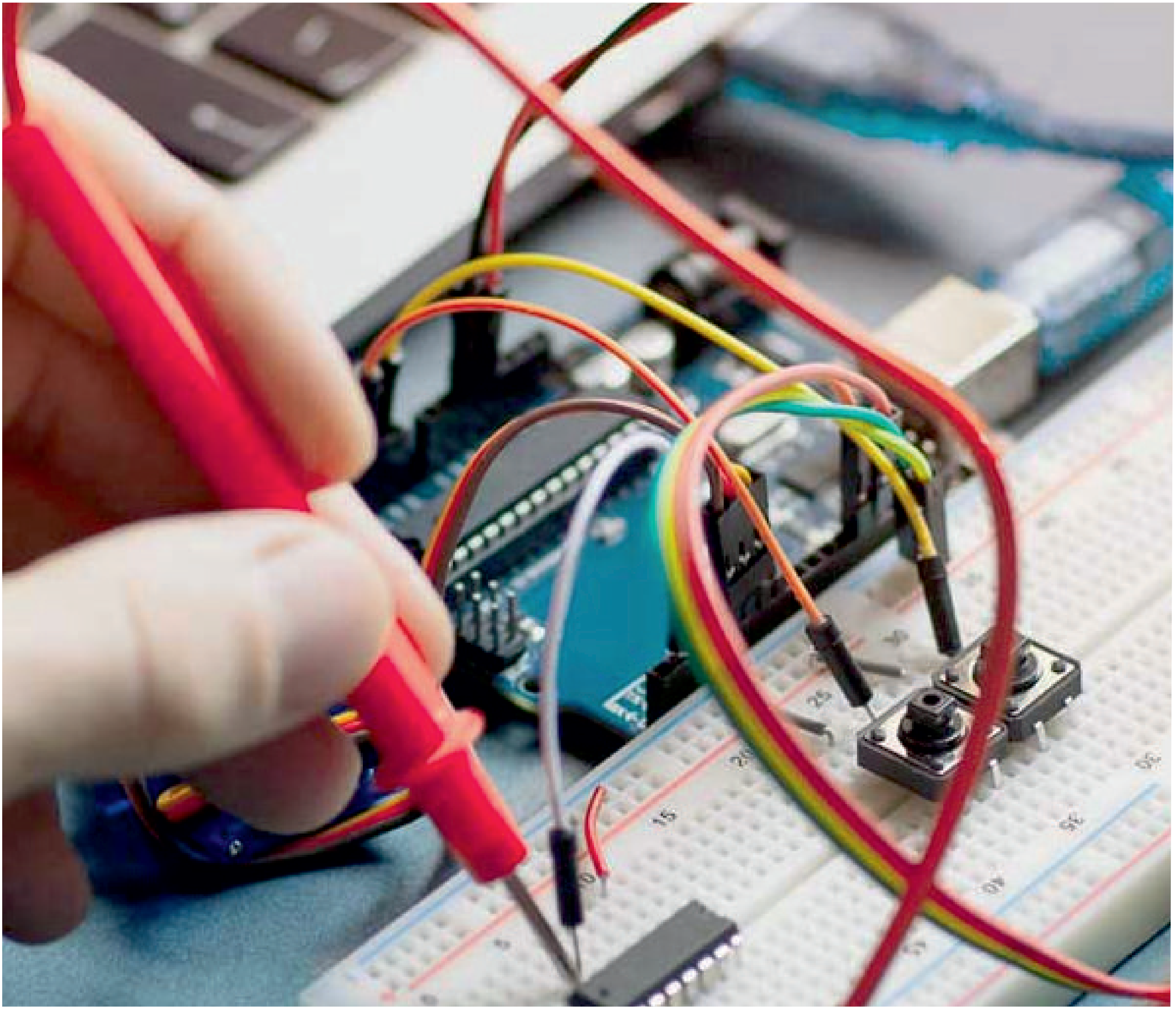
Click on logout.

V. CONCLUSION

This paper proposes a semi-mysterious trait based benefit control plan AnonyControl and a completely unknown characteristic based benefit control plan AnonyControl-F to address the client security issue in a distributed storage server. Utilizing various powers as a part of the distributed computing framework, our proposed plans accomplish fine-grained benefit control as well as character obscurity while leading benefit control in light of clients' personality data. All the more imperatively, our framework can endure up to N − 2 power bargain, which is profoundly ideal particularly in Internet-based distributed computing environment. We additionally led nitty gritty security and execution examination which demonstrates that Anony Control both secure and productive for distributed storage framework. The Anony Control-F straightforwardly acquires the security of the Anony Control and hence is comparably secure as it, yet additional correspondence overhead is caused amid the 1-out-of-n absent exchange.

REFERENCES

1. A. Shamir, “Identity-based cryptosystem and signature schemes,”in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 1985, pp. 47–53.
2. A. Sahai and B. Waters, “Fuzzy identity-basedencryption,”in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.
3. V. Goyal, O. Pandey, A. Sahai, and Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proc.13th CCS, 2006, pp. 89–98.
4. J. Bethencourt, A. Sahai, and B.Waters, “Ciphertext-policy attribute based encryption,” in Proc. IEEE SP, May 2007, pp. 321–334.
5. M. Chase, “Multi-authority attribute based encryption,” in Theory of Cryptography. Berlin, Germany: Springer-Verlag, 2007, pp. 515–534.
6. M. Chase and S. S. M. Chow,“Improving privacy and security in multi-authorityattribute-based encryption,” in Proc. 16th CCS, 2009,pp. 121–130.
7. H. Lin, Z. Cao, X. Liang, and J. Shao, “Secure threshold multi authority attribute based encryption without a central authority,” Inf. Sci., vol. 180, no. 13, pp. 2618–2632, 2010.
8. V. Božovi´c, D. Socek, R. Steinwandt, and V. I. Villányi, “Multi-authority attribute-based encryption with honest-but-curious central authority,” Int. J. Comput. Math., vol. 89, no. 3, pp. 268–283, 2012.
9. F. Li, Y. Rahulamathavan, M. Rajarajan, and R. C.-W. Phan, “Low complexity multi-authority attribute based encryption scheme for mobile cloud computing,” in Proc. IEEE 7th SOSE, Mar. 2013, pp. 573–577.
10. K. Yang, X. Jia, K. Ren, and B. Zhang, “DAC-MACS: Effective data access control for multi-authority cloud storage systems,” in Proc. IEEE INFOCOM, Apr. 2013, pp. 2895–2903.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor:
7.122

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

निस्केयर
NISCAIR

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

☎ 9940 572 462 6381 907 438 ✉ ijareeie@gmail.com

Scan to save the contact details

www.ijareeie.com