**EndUserLogin.java**

import java.awt.Color;

import java.awt.Container;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.Socket;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

public class EndUserLogin implements ActionListener

{

JFrame jf;

Container c;

JLabel l1,l2,l3,a3;

JTextField t1;

JPasswordField t2;

JButton b1,b2,b3;

JComboBox jb,c1;

public Font f = new Font("Times new roman", Font.BOLD, 16);

@SuppressWarnings("deprecation")

EndUserLogin()

{

jf = new JFrame("UserLogin ::Contributory Broadcast Encryption with Efficient Encryption and Short Cipher texts");

c = jf.getContentPane();

c.setLayout(null);

c.setBackground(Color.pink);

l1 = new JLabel("Username");

l1.setFont(f);

l2 = new JLabel("Password");

l2.setFont(f);

t1 = new JTextField(15);

t2 = new JPasswordField(15);

t2.setEchoChar('\*');

b1 = new JButton("Login");

b2 = new JButton("Register");

b3 = new JButton("Reset");

a3 = new JLabel("Group");

a3.setFont(f);

a3.setBounds(70,170,110,35);

c.add(a3);

c1=new JComboBox();

c1.addItem("GROUP1");

c1.addItem("GROUP2");

c1.addItem("GROUP3");

c1.setBounds(175, 178, 130, 25);

c1.addActionListener(this);

c.add(c1);

l1.setBounds(75, 70, 110, 35);

l2.setBounds(75, 120, 110, 35);

t1.setBounds(175, 77, 130, 25);

t2.setBounds(175, 128, 130, 25);

b1.setBounds(80, 250, 80, 30);

b2.setBounds(170, 250, 100, 30);

b3.setBounds(280, 250, 80, 30);

b1.addActionListener(this);

b2.addActionListener(this);

c.add(l1);c.add(l2);//c.add(l3);

c.add(t1);c.add(t2);

c.add(b1);c.add(b2);c.add(b3);

jf.setBounds(550,220,400, 350);

jf.show();

}

public void actionPerformed(ActionEvent e)

{

Object o = e.getSource();

if(o == b1)

{

String name = t1.getText();

String pwd = t2.getText();

String ip=JOptionPane.showInputDialog("Enter the Signal Central Authority IP address");

try

{

Socket cn11 = new Socket(ip,1234);

DataOutputStream dos1 = new DataOutputStream(cn11.getOutputStream());

dos1.writeUTF(name);

dos1.writeUTF(pwd);

String grp=c1.getSelectedItem().toString();

dos1.writeUTF(grp);

DataInputStream din1 = new DataInputStream(cn11.getInputStream());

String status=din1.readUTF();

String sign=din1.readUTF();

if(status.equals("success"))

{

JOptionPane.showMessageDialog(null, "Login Success");

new EndUser(name,sign,grp);

}

else

{

JOptionPane.showMessageDialog(null, "You are not a Valid User");

}

System.out.println("Checking login");

}catch(Exception ee)

{

ee.printStackTrace();

}

}

if(o == b2)

{

Register user = new Register();

user.setSize(400, 440);

user.setVisible(true);

}

}

public static void main(String[] args) {

new EndUserLogin();

}

}

**GroupAuthority.java**

import java.awt.Color;

import java.awt.Container;

import java.awt.Font;

import java.awt.Menu;

import java.awt.MenuBar;

import java.awt.MenuItem;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedInputStream;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.PrintStream;

import java.math.BigInteger;

import java.net.ServerSocket;

import java.net.Socket;

import java.security.DigestInputStream;

import java.security.MessageDigest;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.Vector;

import javax.swing.BorderFactory;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.Timer;

import javax.swing.UIManager;

import javax.swing.border.Border;

public class GroupAuthority implements ActionListener {

JFrame jf;

Container c;

JLabel l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11, l12;

JPanel p1, p2;

JButton b2, b3, b4;

JScrollPane sp;

JTextArea ta;

JTextField t1;

MenuBar mbr;

Menu file;

MenuItem item2, users,csfile,exit;

Border b11, b22, b33;

JScrollPane pane;

String Scheme1, rank;

String f1 = "", f2 = "", f3 = "", f4 = "", f5 = "", f6 = "", f7 = "",f8 = "";

int count = 0;

Timer timer;

ImageIcon gol,one,two,three,four,five,six,seven,eight,arrow1,arrow2,arrow3,a1,a2,arrow4,arrow5,arrow6,arrow7,arrow8,arrow9,end,end1;

JLabel gol1,onel,twol,threel,fourl,fivel,sixl,sevenl,eightl,arrow1l,arrow2l,a1l,a2l,arrow3l,arrow4l,arrow5l,arrow6l,arrow7l,arrow8l,arrow9l,endl,end1l;

JLabel lab1,lab2,lab3,lab4,lab5,lab6,lab7,lab8,lab9,lab10,lab11;

public Font f = new Font("Times new roman", Font.BOLD, 14);

public Font font = new Font("Times new roman", Font.BOLD, 18);

JLabel mg1, mg2, mg3, mg4, mg5, mg6, mg7, mg8, mg9, mg10, g11, g22, g33,

g44, g55, g66, g77;

String keyWord = "ef50a0ef2c3e3a5fdf803ae9752c8c66";

GroupAuthority() {

jf = new JFrame("Group Authority :: Contributory Broadcast Encryption with Efficient Encryption and Short Cipher texts");

c = jf.getContentPane();

c.setLayout(null);

c.setBackground(Color.WHITE);

timer = new Timer(0, null);

one = new ImageIcon(this.getClass().getResource("main.png"));

onel=new JLabel();

onel.setIcon(one);

onel.setBounds(220, 0, 300, 200);

c.add(onel);

two = new ImageIcon(this.getClass().getResource("g1.png"));

twol=new JLabel();

twol.setIcon(two);

twol.setBounds(90, 200, 300, 200);

c.add(twol);

//

three = new ImageIcon(this.getClass().getResource("g2.png"));

threel=new JLabel();

threel.setIcon(three);

threel.setBounds(240, 210, 300, 200);

c.add(threel);

//

four = new ImageIcon(this.getClass().getResource("g3.png"));

fourl=new JLabel();

fourl.setIcon(four);

fourl.setBounds(390, 205, 300, 200);

c.add(fourl);

arrow1 = new ImageIcon(this.getClass().getResource("l1.png"));

arrow1l=new JLabel();

arrow1l.setIcon(arrow1);

arrow1l.setBounds(150, 100, 300, 200);

c.add(arrow1l);

//

arrow2 = new ImageIcon(this.getClass().getResource("l2.png"));

arrow2l=new JLabel();

arrow2l.setIcon(arrow2);

arrow2l.setBounds(240, 110, 300, 200);

c.add(arrow2l);

arrow3 = new ImageIcon(this.getClass().getResource("l3.png"));

arrow3l=new JLabel();

arrow3l.setIcon(arrow3);

arrow3l.setBounds(320, 100, 300, 200);

c.add(arrow3l);

//

lab1 = new JLabel("REGISTER");

lab1.setFont(f);

lab1.setBounds(370, 100, 300, 200);

lab1.setVisible(false);

c.add(lab1);

lab2 = new JLabel("LOGIN");

lab2.setFont(f);

lab2.setBounds(370, 100, 300, 200);

lab2.setVisible(false);

c.add(lab2);

lab3 = new JLabel("DOWNLOAD");

lab3.setFont(f);

lab3.setBounds(370, 100, 300, 200);

lab3.setVisible(false);

c.add(lab3);

lab4 = new JLabel("RE-REGISTER");

lab4.setFont(f);

lab4.setBounds(370, 100, 300, 200);

lab4.setVisible(false);

c.add(lab4);

//

lab5 = new JLabel("UPLOAD");

lab5.setFont(f);

lab5.setBounds(370, 100, 300, 200);

lab5.setVisible(false);

c.add(lab5);

lab6 = new JLabel("VERIFY");

lab6.setFont(f);

lab6.setBounds(370, 100, 300, 200);

lab6.setVisible(false);

c.add(lab6);

// b1 = new JButton("View Cloud Files");

// b1.setForeground(Color.BLACK);

//

// b1.setBounds(270, 370, 200, 30);

// b1.addActionListener(this);

ImageIcon banner = new ImageIcon(this.getClass().getResource("tree.png"));

JLabel title = new JLabel();

title.setIcon(banner);

title.setBounds(420, 50, 530, 275);

mbr = new MenuBar();

file = new Menu("View");

item2 = new MenuItem("View Registered Users");

users = new MenuItem("View Group Users");

csfile = new MenuItem("View Group Sign");

exit = new MenuItem("Exit");

item2.addActionListener(this);

users.addActionListener(this);

csfile.addActionListener(this);

exit.addActionListener(this);

file.add(item2);

file.add(users);

file.add(csfile);

file.add(exit);

mbr.add(file);

jf.setMenuBar(mbr);

// c.add(l2);

c.add(title);

// c.add(b1);

jf.setBounds(-10, 0, 910,480);

jf.show();

int[] ports = new int[] {5757,4093,4094,1234,2222,1333};

for (int i = 0; i < 6; i++) {

Thread t2 = new Thread(new PortListener(ports[i]));

t2.start();

}

}

public static void main(String args[])

{

// new CloudServer();

try {

UIManager

.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

} catch (Exception e1) {

e1.printStackTrace();

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new GroupAuthority();

}

});

}

public void actionPerformed(ActionEvent e) {

Object o = e.getSource();

if (o == item2) {

ViewRegisteredUsers v = new ViewRegisteredUsers();

v.setSize(650, 400);

v.setVisible(true);

}

if (o == users) {

GroupView v1 = new GroupView();

v1.setSize(200, 400);

v1.setVisible(true);

}

if (o == csfile) {

ViewGroupSign vv =new ViewGroupSign();

vv.setSize(670,400);

vv.setVisible(true);

}

if(o == exit)

{

System.exit(0);

}

}

class PortListener implements Runnable {

int port;

public PortListener(int port) {

this.port = port;

}

public void run() {

try{

if (this.port == 5757) {

try {

Socket s;

ServerSocket sc =new ServerSocket(5757);

Dbcon done=new Dbcon();

Connection connect = done.getConnection();

while(true)

{

s = sc.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

String name = din.readUTF();

String pass = din.readUTF();

String addr = din.readUTF();

String city = din.readUTF();

String contact =din.readUTF();

String group = din.readUTF();

String user = "Owner";

if(group.equalsIgnoreCase("GROUP1"))

{

String file="GroupSign";

String content1="group1";

PrintStream out1 = new PrintStream(new FileOutputStream("GroupSign\\"+file));

out1.print(content1);

out1.close();

MessageDigest md1 = MessageDigest.getInstance("SHA1");

FileInputStream in11 = new FileInputStream("GroupSign\\"+ file);

DigestInputStream dis21 = new DigestInputStream(in11, md1);

BufferedInputStream bd1 = new BufferedInputStream(dis21);

while (true) {

int b2 = bd1.read();

if (b2 == -1)

break;

}

BigInteger bi21 = new BigInteger(md1.digest());

String mac1 = bi21.toString(16);

System.out.println(mac1);

connect.createStatement().executeUpdate("insert into register values('"+name+"','"+pass+"','"+addr+"','"+city+"','"+contact+"','"+group+"','"+mac1+"','"+user+"')");

connect.createStatement().executeUpdate("insert into GroupDetails values('"+name+"','"+group+"','"+mac1+"')");

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

}

if(group.equalsIgnoreCase("GROUP2"))

{

String fname1="GroupSign";

String content2="group2";

PrintStream out2 = new PrintStream(new FileOutputStream("GroupSign\\"+fname1));

out2.print(content2);

out2.close();

MessageDigest md2 = MessageDigest.getInstance("SHA1");

FileInputStream in12 = new FileInputStream("GroupSign\\"+ fname1);

DigestInputStream dis22 = new DigestInputStream(in12, md2);

BufferedInputStream bd2 = new BufferedInputStream(dis22);

while (true) {

int b2 = bd2.read();

if (b2 == -1)

break;

}

BigInteger bi22 = new BigInteger(md2.digest());

String mac2 = bi22.toString(16);

System.out.println(mac2);

connect.createStatement().executeUpdate("insert into register values('"+name+"','"+pass+"','"+addr+"','"+city+"','"+contact+"','"+group+"','"+mac2+"','"+user+"')");

connect.createStatement().executeUpdate("insert into GroupDetails values('"+name+"','"+group+"','"+mac2+"')");

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

}

if(group.equalsIgnoreCase("GROUP3"))

{

String fname3="GroupSign";

String content3="group3";

PrintStream out3 = new PrintStream(new FileOutputStream("GroupSign\\"+fname3));

out3.print(content3);

out3.close();

MessageDigest md3 = MessageDigest.getInstance("SHA1");

FileInputStream in13 = new FileInputStream("GroupSign\\"+ fname3);

DigestInputStream dis23 = new DigestInputStream(in13, md3);

BufferedInputStream bd3 = new BufferedInputStream(dis23);

while (true) {

int b2 = bd3.read();

if (b2 == -1)

break;

}

BigInteger bi23 = new BigInteger(md3.digest());

String mac3 = bi23.toString(16);

System.out.println(mac3);

connect.createStatement().executeUpdate("insert into register values('"+name+"','"+pass+"','"+addr+"','"+city+"','"+contact+"','"+group+"','"+mac3+"','"+user+"')");

connect.createStatement().executeUpdate("insert into GroupDetails values('"+name+"','"+group+"','"+mac3+"')");

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

if (this.port == 4093) {

try {

Socket s;

ServerSocket sc =new ServerSocket(4093);

Dbcon done=new Dbcon();

Connection connect = done.getConnection();

while(true)

{

s = sc.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

String name = din.readUTF();

String pass = din.readUTF();

String addr = din.readUTF();

String city = din.readUTF();

String contact =din.readUTF();

String group = din.readUTF();

String user = "End User";

if(group.equalsIgnoreCase("GROUP1"))

{

String file="GroupSign";

String content1="group1";

PrintStream out1 = new PrintStream(new FileOutputStream("GroupSign\\"+file));

out1.print(content1);

out1.close();

MessageDigest md1 = MessageDigest.getInstance("SHA1");

FileInputStream in11 = new FileInputStream("GroupSign\\"+ file);

DigestInputStream dis21 = new DigestInputStream(in11, md1);

BufferedInputStream bd1 = new BufferedInputStream(dis21);

while (true) {

int b2 = bd1.read();

if (b2 == -1)

break;

}

BigInteger bi21 = new BigInteger(md1.digest());

String mac1 = bi21.toString(16);

System.out.println(mac1);

connect.createStatement().executeUpdate("insert into register values('"+name+"','"+pass+"','"+addr+"','"+city+"','"+contact+"','"+group+"','"+mac1+"','"+user+"')");

connect.createStatement().executeUpdate("insert into GroupDetails values('"+name+"','"+group+"','"+mac1+"')");

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

}

if(group.equalsIgnoreCase("GROUP2"))

{

String fname1="GroupSign";

String content2="group2";

PrintStream out2 = new PrintStream(new FileOutputStream("GroupSign\\"+fname1));

out2.print(content2);

out2.close();

MessageDigest md2 = MessageDigest.getInstance("SHA1");

FileInputStream in12 = new FileInputStream("GroupSign\\"+ fname1);

DigestInputStream dis22 = new DigestInputStream(in12, md2);

BufferedInputStream bd2 = new BufferedInputStream(dis22);

while (true) {

int b2 = bd2.read();

if (b2 == -1)

break;

}

BigInteger bi22 = new BigInteger(md2.digest());

String mac2 = bi22.toString(16);

System.out.println(mac2);

connect.createStatement().executeUpdate("insert into register values('"+name+"','"+pass+"','"+addr+"','"+city+"','"+contact+"','"+group+"','"+mac2+"','"+user+"')");

connect.createStatement().executeUpdate("insert into GroupDetails values('"+name+"','"+group+"','"+mac2+"')");

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

}

if(group.equalsIgnoreCase("GROUP3"))

{

String fname3="GroupSign";

String content3="group3";

PrintStream out3 = new PrintStream(new FileOutputStream("GroupSign\\"+fname3));

out3.print(content3);

out3.close();

MessageDigest md3 = MessageDigest.getInstance("SHA1");

FileInputStream in13 = new FileInputStream("GroupSign\\"+ fname3);

DigestInputStream dis23 = new DigestInputStream(in13, md3);

BufferedInputStream bd3 = new BufferedInputStream(dis23);

while (true) {

int b2 = bd3.read();

if (b2 == -1)

break;

}

BigInteger bi23 = new BigInteger(md3.digest());

String mac3 = bi23.toString(16);

System.out.println(mac3);

connect.createStatement().executeUpdate("insert into register values('"+name+"','"+pass+"','"+addr+"','"+city+"','"+contact+"','"+group+"','"+mac3+"','"+user+"')");

connect.createStatement().executeUpdate("insert into GroupDetails values('"+name+"','"+group+"','"+mac3+"')");

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

}

}

}

catch (Exception e) {

e.printStackTrace();

}

}

if (this.port == 4094) {

try {

Socket s;

ServerSocket sc =new ServerSocket(4094);

Dbcon done=new Dbcon();

Connection connect = done.getConnection();

while(true)

{

s = sc.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

String name = din.readUTF();

String pass = din.readUTF();

String grp = din.readUTF();

if(grp.equalsIgnoreCase("GROUP1"))

{

Thread.sleep(2000);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(300);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(2000);

arrow1l.setVisible(false);

Thread.sleep(300);

arrow1l.setVisible(true);

Thread.sleep(300);

arrow1l.setVisible(false);

Thread.sleep(300);

arrow1l.setVisible(true);

Thread.sleep(2000);

twol.setVisible(false);

Thread.sleep(300);

twol.setVisible(true);

Thread.sleep(300);

twol.setVisible(false);

Thread.sleep(300);

twol.setVisible(true);

}

if(grp.equalsIgnoreCase("GROUP2"))

{

Thread.sleep(2000);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(300);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(2000);

arrow2l.setVisible(false);

Thread.sleep(300);

arrow2l.setVisible(true);

Thread.sleep(300);

arrow2l.setVisible(false);

Thread.sleep(300);

arrow2l.setVisible(true);

Thread.sleep(2000);

threel.setVisible(false);

Thread.sleep(300);

threel.setVisible(true);

Thread.sleep(300);

threel.setVisible(false);

Thread.sleep(300);

threel.setVisible(true);

}

if(grp.equalsIgnoreCase("GROUP3"))

{

Thread.sleep(2000);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(300);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(2000);

arrow3l.setVisible(false);

Thread.sleep(300);

arrow3l.setVisible(true);

Thread.sleep(300);

arrow3l.setVisible(false);

Thread.sleep(300);

arrow3l.setVisible(true);

Thread.sleep(2000);

fourl.setVisible(false);

Thread.sleep(300);

fourl.setVisible(true);

Thread.sleep(300);

fourl.setVisible(false);

Thread.sleep(300);

fourl.setVisible(true);

}

String user="Owner";

ResultSet r1=connect.createStatement().executeQuery("select \* from register where name='"+name+"' and pass='"+pass+"' and grp='"+grp+"' and user='"+user+"'");

if(r1.next()==true)

{

String sig= r1.getString(7);

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

dout.writeUTF(sig);

}

else

{

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("no");

dout.writeUTF("no");

}

}

}

catch (Exception e) {

e.printStackTrace();

}

}

if (this.port == 1234) {

try {

Socket s;

ServerSocket sc =new ServerSocket(1234);

Dbcon done=new Dbcon();

Connection connect = done.getConnection();

while(true)

{

s = sc.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

String name = din.readUTF();

String pass = din.readUTF();

String group = din.readUTF();

if(group.equalsIgnoreCase("GROUP1"))

{

Thread.sleep(2000);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(300);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(2000);

arrow1l.setVisible(false);

Thread.sleep(300);

arrow1l.setVisible(true);

Thread.sleep(300);

arrow1l.setVisible(false);

Thread.sleep(300);

arrow1l.setVisible(true);

Thread.sleep(2000);

twol.setVisible(false);

Thread.sleep(300);

twol.setVisible(true);

Thread.sleep(300);

twol.setVisible(false);

Thread.sleep(300);

twol.setVisible(true);

}

if(group.equalsIgnoreCase("GROUP2"))

{

Thread.sleep(2000);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(300);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(2000);

arrow2l.setVisible(false);

Thread.sleep(300);

arrow2l.setVisible(true);

Thread.sleep(300);

arrow2l.setVisible(false);

Thread.sleep(300);

arrow2l.setVisible(true);

Thread.sleep(2000);

threel.setVisible(false);

Thread.sleep(300);

threel.setVisible(true);

Thread.sleep(300);

threel.setVisible(false);

Thread.sleep(300);

threel.setVisible(true);

}

if(group.equalsIgnoreCase("GROUP3"))

{

Thread.sleep(2000);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(300);

onel.setVisible(false);

Thread.sleep(300);

onel.setVisible(true);

Thread.sleep(2000);

arrow3l.setVisible(false);

Thread.sleep(300);

arrow3l.setVisible(true);

Thread.sleep(300);

arrow3l.setVisible(false);

Thread.sleep(300);

arrow3l.setVisible(true);

Thread.sleep(2000);

fourl.setVisible(false);

Thread.sleep(300);

fourl.setVisible(true);

Thread.sleep(300);

fourl.setVisible(false);

Thread.sleep(300);

fourl.setVisible(true);

}

String user="End User";

ResultSet r1=connect.createStatement().executeQuery("select \* from register where name='"+name+"' and pass='"+pass+"' and grp='"+group+"' and user='"+user+"'");

if(r1.next()==true)

{

String sig= r1.getString(7);

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

dout.writeUTF("success");

dout.writeUTF(sig);

}

else

{

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("no");

dout.writeUTF("no");

}

}

}

catch (Exception e) {

e.printStackTrace();

}

}

if (this.port == 2222) {

try {

Socket s;

ServerSocket sc =new ServerSocket(2222);

Dbcon done=new Dbcon();

Connection connect = done.getConnection();

while(true)

{

s = sc.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

String owner = din.readUTF();

String gsign = din.readUTF();

String user="Owner";

ResultSet r1=connect.createStatement().executeQuery("select \* from register where name='"+owner+"' and sign='"+gsign+"' and user='"+user+"'");

if(r1.next()==true)

{

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

dout.writeUTF("correct");

}

else

{

DataOutputStream dout =new DataOutputStream(s.getOutputStream());

dout.writeUTF("incorrect");

}

}

}

catch (Exception e) {

e.printStackTrace();

}

}

if (this.port == 1333)

{

try

{

ServerSocket server909 = new ServerSocket(1333);

Socket con777;

Dbcon db=new Dbcon();

Connection connect=db.getConnection();

while (true)

{

con777 = server909.accept();

ObjectInputStream in3=new ObjectInputStream(con777.getInputStream());

String cs=in3.readObject().toString();

Vector data = new Vector();

ResultSet r1=connect.createStatement().executeQuery("select user,grp,mac from groupdetails");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

// System.out.println(data);

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

}catch (Exception e) {

e.printStackTrace();

}

}

}catch (Exception e) {

// TODO: handle exception

}

}

void clear() throws InterruptedException {

// TODO Auto-generated method stub

Thread.sleep(2000);

lab1.setVisible(false);

lab2.setVisible(false);

lab3.setVisible(false);

lab4.setVisible(false);

lab5.setVisible(false);

lab6.setVisible(false);

}

}

}

**Storageserver.java**

import java.awt.Color;

import java.awt.Container;

import java.awt.Font;

import java.awt.Menu;

import java.awt.MenuBar;

import java.awt.MenuItem;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedInputStream;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.PrintStream;

import java.math.BigInteger;

import java.net.ServerSocket;

import java.net.Socket;

import java.security.DigestInputStream;

import java.security.MessageDigest;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.Random;

import java.util.Vector;

import javax.swing.BorderFactory;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.Timer;

import javax.swing.UIManager;

import javax.swing.border.Border;

public class StorageServer implements ActionListener {

JFrame jf;

Container c;

JLabel l1, l2, l3, l4, l5, l6, l7, l8, l9, l10, l11, l12;

JPanel p1, p2;

JButton b1, b2, b3, b4;

JScrollPane sp;

JTextArea ta;

JTextField t1;

MenuBar mbr;

Menu file;

MenuItem assign,item2, users,csfile,kdc,server,trans,exit;

Border b11, b22, b33;

JScrollPane pane;

String Scheme1, rank;

String f1 = "", f2 = "", f3 = "", f4 = "", f5 = "", f6 = "", f7 = "",f8 = "";

int count = 0;

Timer timer;

public Font f = new Font("Times new roman", Font.BOLD, 12);

public Font font = new Font("Times new roman", Font.BOLD, 18);

JLabel mg1, mg2, mg3, mg4, mg5, mg6, mg7, mg8, mg9, mg10, g11, g22, g33,

g44, g55, g66, g77;

String keyWord = "ef50a0ef2c3e3a5fdf803ae9752c8c66";

ImageIcon one,two,three,four,five,six,seven,eight,arrow1,arrow2,arrow3,arrow4,arrow5,arrow6,end,end1;

JLabel onel,twol,threel,fourl,fivel,sixl,sevenl,eightl,arrow1l,arrow2l,arrow3l,arrow4l,arrow5l,arrow6l,arrow7l,arrow8l,endl,end1l;

JLabel lab1,lab2,lab3,lab4,lab5,lab6,lab7,lab8,lab9,lab10,lab11;

StorageServer() {

jf = new JFrame("Storage Server :: Contributory Broadcast Encryption with Efficient Encryption and Short Cipher texts");

c = jf.getContentPane();

c.setLayout(null);

c.setBackground(Color.white);

timer = new Timer(0, null);

b1=new JButton("View All Files");

b1.setBounds(220,300,200,30);

b1.addActionListener(this);

one = new ImageIcon(this.getClass().getResource("StorageServer.jpg"));

onel=new JLabel();

onel.setIcon(one);

onel.setBounds(210, -20, 300, 200);

c.add(onel);

two = new ImageIcon(this.getClass().getResource("DataOwner.jpg"));

twol=new JLabel();

twol.setIcon(two);

twol.setBounds(90, 180, 300, 200);

c.add(twol);

//

three = new ImageIcon(this.getClass().getResource("EndUser.jpg"));

threel=new JLabel();

threel.setIcon(three);

threel.setBounds(370, 180, 300, 200);

c.add(threel);

arrow1 = new ImageIcon(this.getClass().getResource("arrow1.png"));

arrow1l=new JLabel();

arrow1l.setIcon(arrow1);

arrow1l.setBounds(225, 175, 300, 200);

c.add(arrow1l);

//

arrow2 = new ImageIcon(this.getClass().getResource("arrow2.png"));

arrow2l=new JLabel();

arrow2l.setIcon(arrow2);

arrow2l.setBounds(210, 190, 300, 200);

c.add(arrow2l);

arrow3 = new ImageIcon(this.getClass().getResource("arrow3.png"));

arrow3l=new JLabel();

arrow3l.setIcon(arrow3);

arrow3l.setBounds(135, 80, 300, 200);

c.add(arrow3l);

arrow4 = new ImageIcon(this.getClass().getResource("arrow4.png"));

arrow4l=new JLabel();

arrow4l.setIcon(arrow4);

arrow4l.setBounds(140, 100, 300, 200);

c.add(arrow4l);

//

arrow5 = new ImageIcon(this.getClass().getResource("arrow5.png"));

arrow5l=new JLabel();

arrow5l.setIcon(arrow5);

arrow5l.setBounds(315, 85, 300, 200);

c.add(arrow5l);

arrow6 = new ImageIcon(this.getClass().getResource("arrow6.png"));

arrow6l=new JLabel();

arrow6l.setIcon(arrow6);

arrow6l.setBounds(300, 80, 300, 200);

c.add(arrow6l);

lab1 = new JLabel("UPLOAD");

lab1.setFont(f);

lab1.setBounds(240, 100, 300, 200);

lab1.setVisible(false);

c.add(lab1);

lab2 = new JLabel("REQ-SK");

lab2.setFont(f);

lab2.setBounds(240, 100, 300, 200);

lab2.setVisible(false);

c.add(lab2);

lab3 = new JLabel("DOWNLOAD");

lab3.setFont(f);

lab3.setBounds(235, 100, 300, 200);

lab3.setVisible(false);

c.add(lab3);

mbr = new MenuBar();

// csfile = new MenuItem("Schedule Key Exchange");

kdc = new MenuItem("Give Access Permission");

file = new Menu("Storage Server Files");

assign= new MenuItem("View File Attackers");

item2 = new MenuItem("View Group Users");

users = new MenuItem("View Storage Server Files");

// server= new MenuItem("View Attackers");

// trans= new MenuItem("View Blocked Users");

exit = new MenuItem("Exit");

kdc.addActionListener(this);

assign.addActionListener(this);

item2.addActionListener(this);

users.addActionListener(this);

//// csfile.addActionListener(this);

// server.addActionListener(this);

// trans.addActionListener(this);

exit.addActionListener(this);

// file.add(csfile);

file.add(kdc);

file.add(assign);

file.add(item2);

file.add(users);

ImageIcon banner1 = new ImageIcon(this.getClass().getResource("group.jpg"));

JLabel title = new JLabel();

title.setIcon(banner1);

title.setBounds(560, 50, 530, 275);

//

// file.add(server);

// file.add(trans);

file.add(exit);

mbr.add(file);

jf.setMenuBar(mbr);

// c.add(l2);

c.add(title);

// c.add(b1);

jf.setBounds(0, 0,920 ,420);

jf.show();

int[] ports = new int[] {1111,4646,9988,1444,1555};

for (int i = 0; i < 5; i++) {

Thread t2 = new Thread(new PortListener(ports[i]));

t2.start();

}

}

public static void main(String args[])

{

// new CloudServer();

try {

UIManager

.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

} catch (Exception e1) {

e1.printStackTrace();

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new StorageServer();

}

});

}

public void actionPerformed(ActionEvent e) {

Object o = e.getSource();

Dbcon db=new Dbcon();

Connection con=db.getConnection();

if (o == assign) {

try

{

ViewAttackers v = new ViewAttackers();

v.setSize(680,320);

v.setVisible(true);

}catch (Exception e2) {

// TODO: handle exception

}

}

if (o == kdc) {

try

{

GivePrivilages v = new GivePrivilages();

v.setSize(350,350);

v.setVisible(true);

}catch(Exception es){System.out.println(es);}

}

if (o == item2) {

try

{

String dat="view";

Socket stm=new Socket("localhost",1333);

ObjectOutputStream dos3=new ObjectOutputStream(stm.getOutputStream());

dos3.writeObject(dat);

ObjectInputStream ins=new ObjectInputStream(stm.getInputStream());

Vector a1=(Vector) ins.readObject();

System.out.println(a1);

ViewGroupDetails v = new ViewGroupDetails(a1);

v.setSize(670,400);

v.setVisible(true);

}catch(Exception es){System.out.println(es);}

}

if (o == users) {

try

{

ViewStorageServerFiles v = new ViewStorageServerFiles();

v.setSize(680,320);

v.setVisible(true);

}catch(Exception es){System.out.println(es);}

}

if (o == csfile) {

try

{

}catch (Exception e4) {

// TODO: handle exception

}

}

if (o == server) {

//

// ViewAttackers v =new ViewAttackers();

// v.setSize(580, 400);

// v.setVisible(true);

}

if (o == trans) {

// ViewBlocked d = new ViewBlocked();

// d.setSize(580, 400);

// d.setVisible(true);

}

if(o == exit)

{

System.exit(0);

}

}

class PortListener implements Runnable {

int port;

public PortListener(int port) {

this.port = port;

}

public void run() {

try{

Class.forName("com.mysql.jdbc.Driver");

Dbcon db=new Dbcon();

Connection connect=db.getConnection();

Socket s;

if (this.port == 1111) {

try {

ServerSocket sc =new ServerSocket(1111);

while(true)

{

s = sc.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

String owner = din.readUTF();

String fname = din.readUTF();

String content = din.readUTF();

String sk = din.readUTF();

String grp = din.readUTF();

String grpsign = din.readUTF();

String dt = din.readUTF();

Thread.sleep(1000);

twol.setVisible(false);

Thread.sleep(500);

twol.setVisible(true);

Thread.sleep(500);

twol.setVisible(false);

Thread.sleep(500);

twol.setVisible(true);

Thread.sleep(1000);

arrow3 = new ImageIcon(this.getClass().getResource("arrow3g.png"));

arrow3l.setIcon(arrow3);

lab1.setVisible(true);

Thread.sleep(200);

lab1.setVisible(false);

Thread.sleep(200);

lab1.setVisible(true);

Thread.sleep(200);

lab1.setVisible(false);

Thread.sleep(200);

lab1.setVisible(true);

Thread.sleep(1000);

onel.setVisible(false);

Thread.sleep(500);

onel.setVisible(true);

Thread.sleep(500);

onel.setVisible(false);

Thread.sleep(500);

onel.setVisible(true);

Thread.sleep(1000);

arrow4 = new ImageIcon(this.getClass().getResource("arrow4g.png"));

arrow4l.setIcon(arrow4);

Thread.sleep(1000);

twol.setVisible(false);

Thread.sleep(500);

twol.setVisible(true);

Thread.sleep(500);

twol.setVisible(false);

Thread.sleep(500);

twol.setVisible(true);

clear();

Socket soc = new Socket("localhost",2222);

DataOutputStream dout = new DataOutputStream(soc.getOutputStream());

dout.writeUTF(owner);

dout.writeUTF(grpsign);

DataInputStream din1 = new DataInputStream(soc.getInputStream());

String msg = din1.readUTF();

if(msg.equalsIgnoreCase("correct"))

{

Statement stmt = connect.createStatement();

String acc="NO";

String query1 = "insert into CloudServer values('"+owner+"','"+fname+"','"+sk+"','"+grp+"','"+acc+"','"+acc+"','"+dt+"')";

stmt.executeUpdate(query1);

PrintStream out = null;

out = new PrintStream(new FileOutputStream("StorageServer\\"+fname));

out.print(content);

out.close();

DataOutputStream dout1 = new DataOutputStream(s.getOutputStream());

dout1.writeUTF("success");

}

if(msg.equalsIgnoreCase("incorrect"))

{

DataOutputStream dout1 = new DataOutputStream(s.getOutputStream());

dout1.writeUTF("unauth");

}

}

}catch (Exception e) {

e.printStackTrace();

}

}

if(this.port==4646)

{

String mac=null;

try

{

ServerSocket server1 = new ServerSocket(4646);

Socket con5 = null;

while (true)

{

con5=server1.accept();

DataInputStream dis = new DataInputStream(con5.getInputStream());

String file = dis.readUTF();

Thread.sleep(1000);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

Thread.sleep(500);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

Thread.sleep(1000);

arrow2 = new ImageIcon(this.getClass().getResource("arrow2g.png"));

arrow2l.setIcon(arrow2);

lab2.setVisible(true);

Thread.sleep(200);

lab2.setVisible(false);

Thread.sleep(200);

lab2.setVisible(true);

Thread.sleep(200);

lab2.setVisible(false);

Thread.sleep(200);

lab2.setVisible(true);

Thread.sleep(1000);

twol.setVisible(false);

Thread.sleep(500);

twol.setVisible(true);

Thread.sleep(500);

twol.setVisible(false);

Thread.sleep(500);

twol.setVisible(true);

Thread.sleep(1000);

arrow1 = new ImageIcon(this.getClass().getResource("arrow1g.png"));

arrow1l.setIcon(arrow1);

Thread.sleep(1000);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

Thread.sleep(500);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

clear();

ResultSet r1=connect.createStatement().executeQuery("select \* from CloudServer where fname='"+file+"'");

if(r1.next()==true)

{

String k= r1.getString(3);

DataOutputStream dos1 = new DataOutputStream(con5.getOutputStream());

dos1.writeUTF("key");

dos1.writeUTF(k);

}

if(r1.next()==false)

{

DataOutputStream dos1 = new DataOutputStream(con5.getOutputStream());

dos1.writeUTF("no");

dos1.writeUTF("no");

}

}

}catch(Exception es){System.out.println(es);}

}

if (this.port == 9988)

{

try

{

ServerSocket server909 = new ServerSocket(9988);

Socket con777;

while (true)

{

con777 = server909.accept();

DataInputStream dis = new DataInputStream(con777.getInputStream());

String user = dis.readUTF();

String file = dis.readUTF();

String sk = dis.readUTF();

Thread.sleep(1000);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

Thread.sleep(500);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

Thread.sleep(1000);

arrow6 = new ImageIcon(this.getClass().getResource("arrow6g.png"));

arrow6l.setIcon(arrow6);

lab3.setVisible(true);

Thread.sleep(200);

lab3.setVisible(false);

Thread.sleep(200);

lab3.setVisible(true);

Thread.sleep(200);

lab3.setVisible(false);

Thread.sleep(200);

lab3.setVisible(true);

Thread.sleep(1000);

onel.setVisible(false);

Thread.sleep(500);

onel.setVisible(true);

Thread.sleep(500);

onel.setVisible(false);

Thread.sleep(500);

onel.setVisible(true);

Thread.sleep(1000);

arrow5 = new ImageIcon(this.getClass().getResource("arrow5g.png"));

arrow5l.setIcon(arrow5);

Thread.sleep(1000);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

Thread.sleep(500);

threel.setVisible(false);

Thread.sleep(500);

threel.setVisible(true);

clear();

SimpleDateFormat sd = new SimpleDateFormat("dd/MM/yyyy");

Date d = new Date();

String dt = sd.format(d);

String permit="YES";

ResultSet r1=connect.createStatement().executeQuery("select \* from cloudserver where fname='"+file+"' and apermit='"+permit+"'");

if(r1.next()==true)

{

ResultSet r2=connect.createStatement().executeQuery("select \* from cloudserver where fname='"+file+"' and sk='"+sk+"'");

if(r2.next()==true)

{

FileInputStream fs = new FileInputStream("StorageServer\\"+file);

byte bs1[] = new byte[fs.available()];

fs.read(bs1);

String content = new String(bs1);

DataOutputStream dos1 = new DataOutputStream(con777.getOutputStream());

AES a = new AES();

dos1.writeUTF("success");

dos1.writeUTF(a.decrypt(content, keyWord));

}

if(r2.next()!=true)

{

connect.createStatement().executeUpdate("insert into Attacker values ('"+user+"','"+file+"','"+sk+"','"+dt+"')");

DataOutputStream dos1 = new DataOutputStream(con777.getOutputStream());

dos1.writeUTF("failure");

dos1.writeUTF("failure");

}

}

if(r1.next()!=true)

{

DataOutputStream dos1 = new DataOutputStream(con777.getOutputStream());

dos1.writeUTF("nopermit");

dos1.writeUTF("nopermit");

}

}

}catch (Exception e) {

}

}

if (this.port == 1444)

{

try

{

ServerSocket server909 = new ServerSocket(1444);

Socket con777;

while (true)

{

con777 = server909.accept();

ObjectInputStream in3=new ObjectInputStream(con777.getInputStream());

String fname=in3.readObject().toString();

String group=in3.readObject().toString();

Vector data = new Vector();

ResultSet r1=connect.createStatement().executeQuery("select owner,fname from cloudserver where fname like '%"+fname+"%' and grp='"+group+"'");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

}catch (Exception e) {

e.printStackTrace();

}

}

if (this.port == 1555)

{

try

{

ServerSocket server909 = new ServerSocket(1555);

Socket con777;

while (true)

{

con777 = server909.accept();

ObjectInputStream in3=new ObjectInputStream(con777.getInputStream());

String group=in3.readObject().toString();

if(group.equalsIgnoreCase("GROUP1"))

{

String[] dsname = {"GROUP2","GROUP3"};

String dataname = (String) JOptionPane.showInputDialog(null,"Select GROUP", "Group",

JOptionPane.QUESTION\_MESSAGE, null, dsname, dsname[0]);

if(dataname.equalsIgnoreCase("GROUP2"))

{

Vector data = new Vector();

String yes="YES";

ResultSet r1=connect.createStatement().executeQuery("select owner,fname from CloudServer where grp like '%"+dataname+"%' and spermit='"+yes+"'");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

// System.out.println(data);

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

if(dataname.equalsIgnoreCase("GROUP3"))

{

Vector data = new Vector();

String yes="YES";

ResultSet r1=connect.createStatement().executeQuery("select owner,fname from CloudServer where grp like '%"+dataname+"%' and spermit='"+yes+"'");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

// System.out.println(data);

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

}

if(group.equalsIgnoreCase("GROUP2"))

{

String[] dsname = {"GROUP1", "GROUP3"};

String dataname = (String) JOptionPane.showInputDialog(null,"Select GROUP", "Group",

JOptionPane.QUESTION\_MESSAGE, null, dsname, dsname[0]);

if(dataname.equalsIgnoreCase("GROUP1"))

{

Vector data = new Vector();

String yes="YES";

ResultSet r1=connect.createStatement().executeQuery("select owner,fname from CloudServer where grp like '%"+dataname+"%' and spermit='"+yes+"'");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

// System.out.println(data);

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

if(dataname.equalsIgnoreCase("GROUP3"))

{

Vector data = new Vector();

String yes="YES";

ResultSet r1=connect.createStatement().executeQuery("select owner,fname from CloudServer where grp like '%"+dataname+"%' and spermit='"+yes+"'");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

// System.out.println(data);

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

}

if(group.equalsIgnoreCase("GROUP3"))

{

String[] dsname = {"GROUP1", "GROUP2"};

String dataname = (String) JOptionPane.showInputDialog(null,"Select GROUP", "Group",

JOptionPane.QUESTION\_MESSAGE, null, dsname, dsname[0]);

if(dataname.equalsIgnoreCase("GROUP1"))

{

Vector data = new Vector();

String yes="YES";

ResultSet r1=connect.createStatement().executeQuery("select owner,fname from CloudServer where grp like '%"+dataname+"%' and spermit='"+yes+"'");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

// System.out.println(data);

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

if(dataname.equalsIgnoreCase("GROUP2"))

{

Vector data = new Vector();

String yes="YES";

ResultSet r1=connect.createStatement().executeQuery("select owner,fname from CloudServer where grp like '%"+dataname+"%' and spermit='"+yes+"'");

ResultSetMetaData rsm=r1.getMetaData();

int col=rsm.getColumnCount();

while(r1.next()==true)

{

Vector row = new Vector();

for(int i = 1; i <=col; i++){

row.addElement(r1.getObject(i));

}

data.addElement(row);

}

// System.out.println(data);

ObjectOutputStream dout = new ObjectOutputStream(con777.getOutputStream());

dout.writeObject(data);

}

}

}

}catch (Exception e) {

e.printStackTrace();

}

}

}catch (Exception e) {

// TODO: handle exception

}

}

}

void clear() throws InterruptedException {

// TODO Auto-generated method stub

Thread.sleep(2000);

arrow1 = new ImageIcon(this.getClass().getResource("arrow1.png"));

arrow1l.setIcon(arrow1);

//

arrow2 = new ImageIcon(this.getClass().getResource("arrow2.png"));

arrow2l.setIcon(arrow2);

arrow3 = new ImageIcon(this.getClass().getResource("arrow3.png"));

arrow3l.setIcon(arrow3);

arrow4 = new ImageIcon(this.getClass().getResource("arrow4.png"));

arrow4l.setIcon(arrow4);

//

arrow5 = new ImageIcon(this.getClass().getResource("arrow5.png"));

arrow5l.setIcon(arrow5);

arrow6 = new ImageIcon(this.getClass().getResource("arrow6.png"));

arrow6l.setIcon(arrow6);

lab1.setVisible(false);

lab2.setVisible(false);

lab3.setVisible(false);

}

}

**Ownerlogin.java**

import java.awt.Color;

import java.awt.Container;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.File;

import java.net.Socket;

import javax.swing.\*;

public class OwnerLogin extends JFrame implements ActionListener

{

Font f1 = new Font("Times new Roman",Font.BOLD,16);

Container c;

JButton j1,j2,j3;

JLabel a1,a2,a3;

JTextField t1;

JPasswordField p1;

File path;

String path1;

String content;

String username,password;

JComboBox c1;

OwnerLogin()

{

c = getContentPane();

c.setLayout(null);

c.setBackground(Color.pink);

setTitle("OwnerLogin::Contributory Broadcast Encryption with Efficient Encryption and Short Cipher texts");

a3 = new JLabel("Owner Login!!!!!!!!!!");

a3.setFont(f1);

a3.setBounds(100,20,200,100);

c.add(a3);

a1 = new JLabel("Username");

a1.setFont(f1);

a1.setBounds(50,100,80,30);

c.add(a1);

a2 = new JLabel("Password");

a2.setFont(f1);

a2.setBounds(50,150,80,30);

c.add(a2);

a3 = new JLabel("Group");

a3.setFont(f1);

a3.setBounds(50,200,80,30);

c.add(a3);

c1=new JComboBox();

c1.addItem("GROUP1");

c1.addItem("GROUP2");

c1.addItem("GROUP3");

c1.setBounds(150, 200, 100, 30);

c1.addActionListener(this);

c.add(c1);

t1 = new JTextField(20);

t1.setBounds(150,100,100,30);

c.add(t1);

p1 = new JPasswordField(20);

p1.setBounds(150,150,100,30);

c.add(p1);

j2 = new JButton("Login");

j2.setBounds(150,250,80,30);

j2.addActionListener(this);

j2.setFont(f1);

c.add(j2);

j3 = new JButton("Register");

j3.setBounds(20,250,110,30);

j3.addActionListener(this);

j3.setFont(f1);

c.add(j3);

j1 = new JButton("Reset");

j1.setBounds(250,250,80,30);

j1.addActionListener(this);

j1.setFont(f1);

c.add(j1);

setSize(380,350);

setVisible(true);

//c.setBackground(Color.WHITE);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==j2)

{

String name = t1.getText();

String pwd = p1.getText();

String grp=c1.getSelectedItem().toString();

String ip=JOptionPane.showInputDialog("Enter the Storage Server IP address");

try

{

Socket cn11 = new Socket(ip,4094);

DataOutputStream dos1 = new DataOutputStream(cn11.getOutputStream());

dos1.writeUTF(name);

dos1.writeUTF(pwd);

dos1.writeUTF(grp);

DataInputStream din1 = new DataInputStream(cn11.getInputStream());

String status=din1.readUTF();

String sign =din1.readUTF();

if(status.equals("success"))

{

JOptionPane.showMessageDialog(null, "Login Success");

new DataOwner(name,sign,grp);

}

else

{

JOptionPane.showMessageDialog(null, "You are not a Valid User");

}

System.out.println("Checking login");

}catch(Exception ee)

{

ee.printStackTrace();

}

}

if(e.getSource()==j1)

{

t1.setText(null);

p1.setText(null);

}

if(e.getSource()==j3)

{

new OwnerRegister();

}

}

public static void main(String[] args) {

//new OwnerLogin();

try {

UIManager

.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");

} catch (Exception e1) {

e1.printStackTrace();

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new OwnerLogin();

}

});

}

}