Subject: Re: RenRem protocol
Posted by snazy2000 on Wed, 15 Jul 2009 21:05:40 GMT
View Forum Message <> Reply to Message

i no im opening an old topic but ages ago i found this somewere

```
import java.io.*;
import java.net.*;

class JavaFDS {

byte[] message5 = new byte[20];

private String password;
private byte[] result;
private byte[] receiveData;
private String message;
private DatagramSocket clientSocket;
private InetAddress IPAdress;
private int port;
private DatagramPacket sendPacket,receivePacket;

 public void connectFDS(String password, int port) throws Exception
 {
  this.password = password;
  message = password;
  this.port = port;
  encrypt2();

  clientSocket = new DatagramSocket();
  IPAdress = InetAddress.getByName("loopback");
  receiveData = new byte[1024];

  sendPacket = new DatagramPacket(result,result.length,IPAdress,port);
  clientSocket.send(sendPacket);
  //receivePacket = new DatagramPacket(receiveData,receiveData.length);
  //clientSocket.receive(receivePacket);
  //decrypt2();
  //Connection1.sendMsg("PRIVMSG " + Connection1.chan + " " + message);
  //System.out.println(byteToInt(receiveData[1]));
  return;
 }
 public void sendMsg(String message) throws Exception
 {
  this.message = message;
  encrypt2();
  receiveData = new byte[1024];
  sendPacket = new DatagramPacket(result,result.length,IPAdress,port);
  clientSocket.send(sendPacket);
  //receivePacket = new DatagramPacket(receiveData,receiveData.length);
  //clientSocket.receive(receivePacket);
  //modifiedSentence = new String(receivePacket.getData());
  //System.out.println(byteToInt(receiveData[1]));
  //decrypt2();
  //decrypt2();
  //shutdown();
  //return this.message;
  return;
 }
 public void shutdown() throws Exception
 {
  clientSocket.close();
 }


// ****************
// Internal functions
// ****************


// Encrypt the variable "message" and stock the encryption into the variable "result"

private void decrypt2() throws Exception {
 int l=1023;
 while(byteToInt(receiveData[l])==0)
  l--;
 //System.out.println(l);
 while (l%4 != 0)
   l++;
 byte[] dmessage = new byte[l+1];
 for(int i=0;i<l+1;i++)
  dmessage[i] = receiveData[i];

 //System.out.println(dmessage[0]);

 byte ESI;
  byte[] ECX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x00};
  byte[] EDX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x00};
  byte[] EBX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x01};
  byte[] EAX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x00};

  String shortpass;
  if (password.length()>=8)
   shortpass = password.substring(0,8);
  else
   return;
  byte[] bpass = new byte[8];
  bpass = shortpass.getBytes();

 for(int i=4;i<l+1;i++)
 {
  EDX[3] = dmessage[i];
  mov(ECX,EAX);
  ECX[3] = (byte) (ECX[3] & (byte)0x07);
  ECX[0]=(byte)0x00;
  ECX[1]=(byte)0x00;
  ECX[2]=(byte)0x00;
  ESI = ECX[3];
  ECX[3] = bpass[byteToInt(ECX[3])];
  ECX[3] = (byte)(ECX[3] ^ EDX[3]);
  EDX[3] = ECX[3];
  bpass[(int)ESI] = ECX[3];
  EDX[3] = (byte)(EDX[3] + ~EAX[3] + (byte)0x01);
  EDX[3] = (byte)(EDX[3] + (byte)0x32);
  dmessage[i] = EDX[3];
  add(EAX,EBX);
 }

 for(int i=0;i<l+1;i++)
 {
  if(byteToInt(dmessage[i]) == 10)
   dmessage[i]=(byte)0x20;
   }

 byte[] dmessage2 = new byte[l+1-8];
 for(int i=0;i<l+1-8;i++)
  dmessage2[i] = dmessage[i+8];

 byte[] dmessage3 = new byte[l+1-11];
 for(int i=0;i<l+1-11;i++)
  dmessage3[i] = dmessage2[i];
 String tze = new String( dmessage3 , "Cp1252" );
 this.message=tze;
 //System.out.println(message);


}

private void encrypt2() throws Exception
 {
  int l = this.message.length();
  byte[] bmessage = new byte[l];
```

```
bmessage = this.message.getBytes();

String shortpass;
if (password.length()>=8)
 shortpass = password.substring(0,8);
else
 return;
byte[] bpass = new byte[8];
bpass = shortpass.getBytes();

l=l+9;
while (l%4 != 0)
 l++;
result = new byte[l];

// Initialisation

for(int i=0;i<l;i++)
{
 if(i<8)
 {
  result[i]=(byte)0x00;
 }
 else if(i>7 && i<8+this.message.length())
 {
  result[i]=bmessage[i-8];
 }
 else
 {
  result[i] = (byte)0x00;
 }
}

// Encryption

byte ESI;
byte[] ECX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x00};
byte[] EDX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x00};
byte[] EBX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x01};
byte[] EAX = {(byte)0x00,(byte)0x00,(byte)0x00,(byte)0x00};

for(int i=4;i<this.message.length()+9;i++)
{
 EAX[3] = result[i];
 mov(EDX,ECX);
 EDX[3] = (byte) (EDX[3] & (byte)0x07);
 EDX[0]=(byte)0x00;
 EDX[1]=(byte)0x00;
 EDX[2]=(byte)0x00;
 add(EAX,ECX);
 EAX[3] = (byte)(EAX[3] + ~(byte)0x32 + (byte)0x01);
 ESI = EDX[3];
 EDX[3] = bpass[(int)EDX[3]];
 EAX[3] = (byte)(EAX[3] ^ EDX[3]);
 result[i] = EAX[3];
 EDX[3] = (byte)(EDX[3] ^ EAX[3]);
 bpass[(int)ESI] = EDX[3];
 add(ECX,EBX);
}


int rrr;
if((this.message.length()+4+1)%4 == 0)
 rrr = (this.message.length()+4+1)/4;
 else
 rrr = (this.message.length()+4+1)/4+1;
//if((this.message.length())%4 != 0)
//rrr--;
for(int i=0;i<rrr;i++)
{
 ECX[0] = result[3];
 ECX[1] = result[2];
 ECX[2] = result[1];
 ECX[3] = result[0];

 mov(EAX,ECX);

 EAX[3] = (byte)((byte)(EAX[0] >> 7) & (byte)0x00 + 1);
 EAX[0] = (byte)0x00;
 EAX[1] = (byte)0x00;
 EAX[2] = (byte)0x00;

 shl(ECX);

 add(EAX,ECX);

 ECX[0] = result[4*i+7];
 ECX[1] = result[4*i+6];
 ECX[2] = result[4*i+5];
 ECX[3] = result[4*i+4];

 add(EAX,ECX);

 result[3] = EAX[0];
 result[2] = EAX[1];
 result[1] = EAX[2];
```

```
   result[0] = EAX[3];
  }
 }

// Convert a signed byte to an integer.

private int byteToInt(byte bIn){
if((bIn > 127) || (bIn < -128))
 return 0;
else
{
 if(bIn >= 0)
  return (int)bIn;
 else{
  return (-(-(int)bIn) & 0xff);
 }
}
}

// Replace the first registry by the second one.

private void mov(byte[] reg1, byte[] reg2)
{
 reg1[0] = reg2[0];
 reg1[1] = reg2[1];
 reg1[2] = reg2[2];
 reg1[3] = reg2[3];
}

// Add the second registry to the first one and stock the result into the first registry.

private void add(byte[] reg1, byte[] reg2)
{
 byte temp = (byte)0x00;
 byte temp2 = (byte)0x00;

 if(byteToInt(reg1[3])+byteToInt(reg2[3]) > 255)
  temp = (byte)0x01;
 reg1[3] = (byte)(reg1[3] + reg2[3]);

 if(byteToInt(reg1[2])+byteToInt(reg2[2])+temp > 255)
  temp2 = (byte)0x01;
 reg1[2] = (byte)(reg1[2] + reg2[2] + temp);

 if(byteToInt(reg1[1])+byteToInt(reg2[1])+temp2 > 255)
  temp = (byte)0x01;
 else
  temp = (byte)0x00;
 reg1[1] = (byte)(reg1[1] + reg2[1] +temp2);
 reg1[0] = (byte)(reg1[0] + reg2[0] +temp);
}

// Multiply the registry by 2.

private void shl(byte []reg)
{
 byte temp = (byte)0x00;
 byte temp2 = (byte)0x00;

 if((int)reg[3] < 0)
  temp = (byte)0x01;
 reg[3] = (byte)(reg[3] << 1);

 if((int)reg[2] < 0)
  temp2 = (byte)0x01;
 reg[2] = (byte)(reg[2] << 1);
 reg[2] = (byte)(reg[2] + temp);

 if((int)reg[1] < 0)
  temp = (byte)0x01;
 else
  temp= (byte)0x00;
 reg[1] = (byte)(reg[1] << 1);
 reg[1] = (byte)(reg[1] + temp2);
 reg[0] = (byte)(reg[0] << 1);
 reg[0] = (byte)(reg[0] + temp);
}

}
```

Dont no if that can help any 1