**Name – Snehal Trigune**

**Roll No – B1821024**

**Batch - A2**

**Assignment No 2**

**Title –**

**Program –**

**Server side –**

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Base64;

import javax.crypto.Cipher;

import javax.crypto.SecretKey;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.spec.SecretKeySpec;

public class AES\_SERVER {

public static void main(String[] args) {

final int PORT=8088;

try{

ServerSocket serverSocket = new ServerSocket(PORT);

System.out.println("Started server on port "+PORT+"\nWaiting for client connection");

Socket server = serverSocket.accept();

System.out.println("Just connected to " + server.getRemoteSocketAddress());

// Accepts the data from client

DataInputStream in = new DataInputStream(server.getInputStream());

DataOutputStream out = new DataOutputStream(server.getOutputStream());

String encodedkey = in.readUTF();

byte[] decodedKey = Base64.getDecoder().decode(encodedkey);

// rebuild key using SecretKeySpec

SecretKey key = new SecretKeySpec(decodedKey, 0, decodedKey.length, "AES");

byte[]IV=new byte[16];

int noBytes=in.read(IV);

String cipherText=in.readUTF();

System.out.println("Symmetric Key :"+encodedkey);

System.out.println("Received CipherText: "+cipherText);

String decryptedText=decrypt(Base64.getDecoder().decode(cipherText),key,IV);//todo

System.out.println("Decrypted text: "+decryptedText);

}

catch(Exception e){

}

}

public static String decrypt (byte[] cipherText, SecretKey key,byte[] IV) throws Exception

{

//Get Cipher Instance

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

//Create SecretKeySpec

SecretKeySpec keySpec = new SecretKeySpec(key.getEncoded(), "AES");

//Create IvParameterSpec

IvParameterSpec ivSpec = new IvParameterSpec(IV);

//Initialize Cipher for DECRYPT\_MODE

cipher.init(Cipher.DECRYPT\_MODE, keySpec, ivSpec);

//Perform Decryption

byte[] decryptedText = cipher.doFinal(cipherText);

return new String(decryptedText);

}

}

**Client side –**

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.Socket;

import java.net.UnknownHostException;

import java.security.NoSuchAlgorithmException;

import java.security.SecureRandom;

import java.util.Base64;

import javax.crypto.\*;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.spec.SecretKeySpec;

public class AES {

public final static String SERVER="127.0.0.1";

public static final int PORT=8088;

public static void main(String[] args) throws NoSuchAlgorithmException {

String plainText="Information and Cyber Security";

try {

Socket soc=new Socket(SERVER, PORT);

System.out.println("Just connected to " + soc.getRemoteSocketAddress());

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

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

//GENERATING KEYS

KeyGenerator keyGenerator=KeyGenerator.getInstance("AES");

keyGenerator.init(128);

SecretKey key=keyGenerator.generateKey();

// Generating IV.(INitializaion vector)/We are using CBC

byte[] IV = new byte[16];

SecureRandom random = new SecureRandom();

random.nextBytes(IV);

byte[] cipherText = encrypt(plainText.getBytes(),key, IV);

System.out.println("Plain Text: :"+plainText);

System.out.println("Encrypted Text : "+Base64.getEncoder().encodeToString(cipherText) );

String cipherString=Base64.getEncoder().encodeToString(cipherText);

System.out.println("Symmetric Key "+Base64.getEncoder().encodeToString(key.getEncoded()));

out.writeUTF(Base64.getEncoder().encodeToString(key.getEncoded())); //key

out.write(IV);

out.writeUTF(cipherString);

} catch (UnknownHostException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static byte[] encrypt(byte[] plaintext, SecretKey key, byte[] IV) throws Exception{

//Get Cipher Instance

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

//Create SecretKeySpec

SecretKeySpec keySpec = new SecretKeySpec(key.getEncoded(), "AES");

//Create IvParameterSpec

IvParameterSpec ivSpec = new IvParameterSpec(IV);

//Initialize Cipher for ENCRYPT\_MODE

cipher.init(Cipher.ENCRYPT\_MODE, keySpec,ivSpec );

//Perform Encryption

byte[] cipherText = cipher.doFinal(plaintext);

return cipherText;

}

public static String decrypt (byte[] cipherText, SecretKey key,byte[] IV) throws Exception

{

//Get Cipher Instance

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

//Create SecretKeySpec

SecretKeySpec keySpec = new SecretKeySpec(key.getEncoded(), "AES");

//Create IvParameterSpec

IvParameterSpec ivSpec = new IvParameterSpec(IV);

//Initialize Cipher for DECRYPT\_MODE

cipher.init(Cipher.DECRYPT\_MODE, keySpec, ivSpec);

//Perform Decryption

byte[] decryptedText = cipher.doFinal(cipherText);

return new String(decryptedText);

}

}

**Output –**

Just connected to /127.0.0.1:8088

Plain Text: :Information and Cyber Security

Encrypted Text : YFoNp9h+/5LVtaJfCvGAKBmplQCUsEuH+YL3gs0nUbs=

Symmetric Key SYGZo5fASQe62gV6gAnrVg==