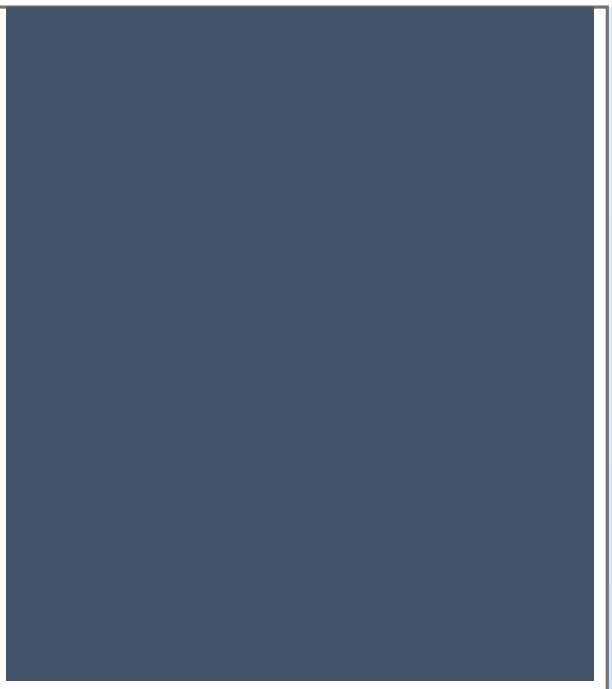


Programación de servicios y procesos

Act5.3. Confidencialidad e identidad

Francisco José García Cutillas | 2FPGS_DAM

Índice

Ejercicio 1

Confidencialidad e identidad

Programa una aplicación Java que, dado un fichero, lo cifre primeramente con la clave privada del emisor y, posteriormente, con la clave pública del destinatario.

Programa también la aplicación que descifre el fichero.

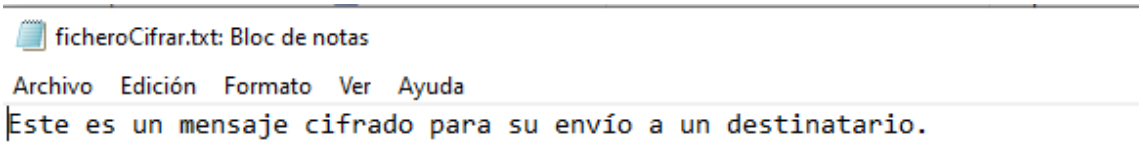
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */
package com.mycompany.psp_act5_3;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;
import javax.crypto.CipherOutputStream;

/**
 *
 * @author Fran
 */
public class PSP_Act5_3 {

    public static void main(String[] args) {

        try {

            //Generamos las claves privadas y públicas del emisor y el receptor
            KeyPair clavesEmisor = generarClaves();
            KeyPair clavesReceptor = generarClaves();

            //Ciframos el fichero con la clave privada del emisor
            cifrarPrivadaEmisor( rutaFichero: "ficheroCifrar.txt",  clavePrivadaEmisor: clavesEmisor.getPrivate());

            //Ciframos el fichero con la clave pública del receptor
            cifrarPublicaReceptor( rutaFichero: "ficheroCifrar.txt",  clavePublicaReceptor: clavesReceptor.getPublic());

            //Desencriptamos el fichero con la clave privada del receptor
            descifrar( ficheroCifrado: "ficheroCifradoPublicaReceptor.rsa",  nombreFicheroSinCifrar: "ficheroReceptorDescifrado.txt",
                    key: clavesReceptor.getPrivate());

        } catch (Exception ex) {

            System.out.println("Error: " + ex.getMessage());

        }

    }

    //Método para generar el par de claves pública y privada
    public static KeyPair generarClaves() throws NoSuchAlgorithmException {

        KeyPairGenerator generador = KeyPairGenerator.getInstance( algorithm: "RSA");
        generador.initialize( keysize: 2048);
        KeyPair claves = generador.generateKeyPair();

        return claves;

    }

```

```java
    //Método para cifrar un fichero con la clave privada del emisor. Recibe la ruta del fichero y la clave privada del emisor
    public static void cifrarPrivadaEmisor(String rutaFichero, PrivateKey clavePrivadaEmisor) throws Exception {

        //Cifrado con clave privada del emisor.
        Cipher cipher = Cipher.getInstance( transformation: "RSA");
        cipher.init( opmode: Cipher.ENCRYPT_MODE,  key: clavePrivadaEmisor);

        FileInputStream fis = new FileInputStream( name: rutaFichero);
        FileOutputStream fos = new FileOutputStream( name: "ficheroCifradoPrivadaEmisor.rsa");
        CipherOutputStream cos = new CipherOutputStream( os: fos,  c: cipher);

        //Leemos el fichero y lo cifamos
        byte[] buffer = new byte[1024];

        int bytesLeidos = fis.read( b: buffer);

        while (bytesLeidos != -1) {

            cos.write( b: buffer,  off: 0,  len: bytesLeidos);
            bytesLeidos = fis.read( b: buffer);

        }

        cos.close();
        fos.close();
        fis.close();

        System.out.println( x: "Fichero cifrado con clave privada del emisor.");

    }

    //Método para cifrar un fichero con la clave pública del receptor. Recibe la ruta del fichero y la clave pública del receptor
    public static void cifrarPublicaReceptor(String rutaFichero, PublicKey clavePublicaReceptor) throws Exception {

        //Cifrado con clave pública del receptor.
        Cipher cipher = Cipher.getInstance( transformation: "RSA");
        cipher.init( opmode: Cipher.ENCRYPT_MODE,  key: clavePublicaReceptor);

        FileInputStream fis = new FileInputStream( name: rutaFichero);
        FileOutputStream fos = new FileOutputStream( name: "ficheroCifradoPublicaReceptor.rsa");
        CipherOutputStream cos = new CipherOutputStream( os: fos,  c: cipher);

        //Leemos el fichero y lo cifamos
        byte[] buffer = new byte[1024];

        int bytesLeidos = fis.read( b: buffer);

        while (bytesLeidos != -1) {

            cos.write( b: buffer,  off: 0,  len: bytesLeidos);
            bytesLeidos = fis.read( b: buffer);

        }

        cos.close();
        fos.close();
        fis.close();

        System.out.println( x: "Fichero cifrado con clave pública del receptor.");

    }
```

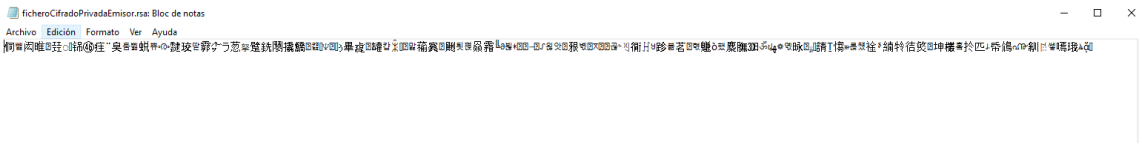
```java

    //Método para descifrar un fichero. Recibe la ruta del fichero cifrado, la del nuevo sin cifrar y la clave privada del receptor
    private static void descifrar(String ficheroCifrado, String nombreFicheroSinCifrar, PrivateKey key) throws Exception {

        //Descifrado con clave privada del receptor
        Cipher cipher = Cipher.getInstance( transformation: "RSA");
        cipher.init( opmode: Cipher.DECRYPT_MODE, key);

        FileInputStream fis = new FileInputStream( name: ficheroCifrado);
        FileOutputStream fos = new FileOutputStream( name: nombreFicheroSinCifrar);
        CipherInputStream cis = new CipherInputStream( is: fis,  c: cipher);

        //Leemos el fichero y lo desciframos
        byte[] buffer = new byte[1024];

        int bytesLeidos = cis.read( b: buffer);

        while (bytesLeidos != -1) {

            fos.write( b: buffer,  off: 0,  len: bytesLeidos);
            bytesLeidos = cis.read( b: buffer);

        }

        cis.close();
        fos.close();
        fis.close();

        System.out.println( x: "Fichero descifrado.");

    }

}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_Act5_3 ---
Fichero cifrado con clave privada del emisor.
Fichero cifrado con clave pública del receptor.
Fichero descifrado.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.595 s
Finished at: 2024-02-09T21:03:38+01:00
------------------------------------------------------------------------
```

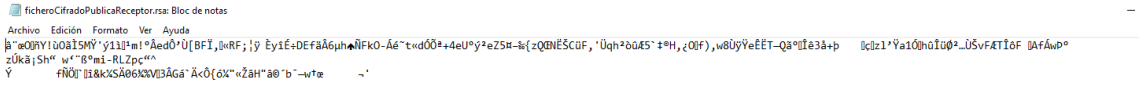
Fichero original



Fichero cifrado con clave privada del emisor



Fichero cifrado con clave pública del receptor



Fichero descifrado con clave privada del receptor