Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**Raport**

Disciplina: Securitatea informațională.

Lucrarea de laborator nr. 2

**Tema:** Algoritmul RSA

A efectuat: st.gr. TI-194 Ceban Vitalie

A verificat: asis. univ. Răducanu Octavian

Chişinău 2022

**Scopul lucrării:**

- Studierea Algoritmului RSA

- Calculul cheilor

- Diferenţele între algoritmi simetrici şi asimetrici

- Realizarea unei aplicaţi ce utilizează RSA ca metodă de criptare

**Obiectivele lucrării:**

- Realizarea unei aplicaţi ce utilizează RSA ca metodă de criptare

import javax.crypto.Cipher;

import javax.crypto.IllegalBlockSizeException;

import javax.crypto.NoSuchPaddingException;

import javax.crypto.SecretKey;

import javax.crypto.BadPaddingException;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKeyFactory;

import javax.crypto.SealedObject;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.spec.PBEKeySpec;

import javax.crypto.spec.SecretKeySpec;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.Serializable;

import java.security.InvalidAlgorithmParameterException;

import java.security.InvalidKeyException;

import java.security.NoSuchAlgorithmException;

import java.security.SecureRandom;

import java.security.spec.InvalidKeySpecException;

import java.security.spec.KeySpec;

import java.util.Base64;

public class AESUtil {

public static String encrypt(String algorithm, String input, SecretKey key, IvParameterSpec iv)

throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidAlgorithmParameterException,

InvalidKeyException, BadPaddingException, IllegalBlockSizeException {

Cipher cipher = Cipher.getInstance(algorithm);

cipher.init(Cipher.ENCRYPT\_MODE, key, iv);

byte[] cipherText = cipher.doFinal(input.getBytes());

return Base64.getEncoder()

.encodeToString(cipherText);

}

public static String decrypt(String algorithm, String cipherText, SecretKey key, IvParameterSpec iv)

throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidAlgorithmParameterException,

InvalidKeyException, BadPaddingException, IllegalBlockSizeException {

Cipher cipher = Cipher.getInstance(algorithm);

cipher.init(Cipher.DECRYPT\_MODE, key, iv);

byte[] plainText = cipher.doFinal(Base64.getDecoder()

.decode(cipherText));

return new String(plainText);

}

public static SecretKey generateKey(int n) throws NoSuchAlgorithmException {

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

keyGenerator.init(n);

SecretKey key = keyGenerator.generateKey();

return key;

}

public static SecretKey getKeyFromPassword(String password, String salt)

throws NoSuchAlgorithmException, InvalidKeySpecException {

SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");

KeySpec spec = new PBEKeySpec(password.toCharArray(), salt.getBytes(), 65536, 256);

SecretKey secret = new SecretKeySpec(factory.generateSecret(spec)

.getEncoded(), "AES");

return secret;

}

public static IvParameterSpec generateIv() {

byte[] iv = new byte[16];

new SecureRandom().nextBytes(iv);

return new IvParameterSpec(iv);

}

public static void encryptFile(String algorithm, SecretKey key, IvParameterSpec iv,

File inputFile, File outputFile) throws IOException, NoSuchPaddingException,

NoSuchAlgorithmException, InvalidAlgorithmParameterException, InvalidKeyException,

BadPaddingException, IllegalBlockSizeException {

Cipher cipher = Cipher.getInstance(algorithm);

cipher.init(Cipher.ENCRYPT\_MODE, key, iv);

FileInputStream inputStream = new FileInputStream(inputFile);

FileOutputStream outputStream = new FileOutputStream(outputFile);

byte[] buffer = new byte[64];

int bytesRead;

while ((bytesRead = inputStream.read(buffer)) != -1) {

byte[] output = cipher.update(buffer, 0, bytesRead);

if (output != null) {

outputStream.write(output);

}

}

byte[] outputBytes = cipher.doFinal();

if (outputBytes != null) {

outputStream.write(outputBytes);

}

inputStream.close();

outputStream.close();

}

public static void decryptFile(String algorithm, SecretKey key, IvParameterSpec iv,

File encryptedFile, File decryptedFile) throws IOException, NoSuchPaddingException,

NoSuchAlgorithmException, InvalidAlgorithmParameterException, InvalidKeyException,

BadPaddingException, IllegalBlockSizeException {

Cipher cipher = Cipher.getInstance(algorithm);

cipher.init(Cipher.DECRYPT\_MODE, key, iv);

FileInputStream inputStream = new FileInputStream(encryptedFile);

FileOutputStream outputStream = new FileOutputStream(decryptedFile);

byte[] buffer = new byte[64];

int bytesRead;

while ((bytesRead = inputStream.read(buffer)) != -1) {

byte[] output = cipher.update(buffer, 0, bytesRead);

if (output != null) {

outputStream.write(output);

}

}

byte[] output = cipher.doFinal();

if (output != null) {

outputStream.write(output);

}

inputStream.close();

outputStream.close();

}

public static SealedObject encryptObject(String algorithm, Serializable object, SecretKey key,

IvParameterSpec iv) throws NoSuchPaddingException, NoSuchAlgorithmException,

InvalidAlgorithmParameterException, InvalidKeyException, IOException, IllegalBlockSizeException {

Cipher cipher = Cipher.getInstance(algorithm);

cipher.init(Cipher.ENCRYPT\_MODE, key, iv);

SealedObject sealedObject = new SealedObject(object, cipher);

return sealedObject;

}

public static Serializable decryptObject(String algorithm, SealedObject sealedObject, SecretKey key,

IvParameterSpec iv) throws NoSuchPaddingException, NoSuchAlgorithmException,

InvalidAlgorithmParameterException, InvalidKeyException, ClassNotFoundException,

BadPaddingException, IllegalBlockSizeException, IOException {

Cipher cipher = Cipher.getInstance(algorithm);

cipher.init(Cipher.DECRYPT\_MODE, key, iv);

Serializable unsealObject = (Serializable) sealedObject.getObject(cipher);

return unsealObject;

}

public static String encryptPasswordBased(String plainText, SecretKey key, IvParameterSpec iv)

throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidAlgorithmParameterException,

InvalidKeyException, BadPaddingException, IllegalBlockSizeException {

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

cipher.init(Cipher.ENCRYPT\_MODE, key, iv);

return Base64.getEncoder()

.encodeToString(cipher.doFinal(plainText.getBytes()));

}

public static String decryptPasswordBased(String cipherText, SecretKey key, IvParameterSpec iv)

throws NoSuchPaddingException, NoSuchAlgorithmException, InvalidAlgorithmParameterException,

InvalidKeyException, BadPaddingException, IllegalBlockSizeException {

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");

cipher.init(Cipher.DECRYPT\_MODE, key, iv);

return new String(cipher.doFinal(Base64.getDecoder()

.decode(cipherText)));

}

}

import java.io.Serializable;

import java.util.Objects;

public class Student implements Serializable {

private String name;

private int age;

public Student(String name, int age) {

this.name = name;

this.age = age;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

@Override

public boolean equals(Object o) {

if (this == o)

return true;

if (o == null || getClass() != o.getClass())

return false;

Student student = (Student) o;

return age == student.age && Objects.equals(name, student.name);

}

}