Міністерство освіти та науки

Прикарпатський національний університет

Фізико-технічний факультет

Кафедра комп’ютерної інженерії та електроніки

Лабораторна робота № 5

З курсу “Захист інформації у комп’ютерних мережах та системах”

Виконав студент групи КІ-41

Воробій Віталій

Івано-Франківськ 2020

5 варіант

Програмна реалізація мовою *Java:*

Реалізував клас, що містить ряд ефективних алгоритмів, що будуть корисні у подальшому.

package solution;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class RSAlgorithms {  
  
 */\*\*  
 \* Функція для генерування простих чисел  
 \*/* public static List<Integer> generatePrimeNumbers(int length) {  
 List<Integer> numbers = new ArrayList<>();  
 for (int n = 2; numbers.size() != length; n++) {  
 boolean isPrime = true;  
 for (Integer r: numbers) {  
 if (n % r == 0) {  
 isPrime = false;  
 break;  
 }  
 }  
 if (isPrime) {  
 numbers.add(n);  
 }  
 }  
 return numbers;  
 }  
  
 */\*\*  
 \* Рекурсивний алгоритм Евкліда  
 \*/* public static int greatestCommonDivider(int a, int b) {  
 if (b == 0) return a;  
 return *greatestCommonDivider*(b, a % b);  
 }  
  
 */\*\*  
 \* Дискретне піднесення до степеня по модулю  
 \*/* public static int discretePower(int x, int y, int p) {  
 int res = 1;  
 x = x % p;  
 if (x == 0) return 0;  
 while (y > 0)  
 {  
 if((y & 1)==1)  
 res = (res \* x) % p;  
 y = y >> 1;  
 x = (x \* x) % p;  
 }  
 return res;  
 }

}

Написав клас, що описує конфігурацію *RSA*

package solution;  
  
public class RSA {  
 private final int p;  
 private final int q;  
 private int n;  
 private int e;  
 private int d;  
  
 public RSA(int p, int q) {  
 this.p = p;  
 this.q = q;  
 calcKeys();  
 }  
  
 private void calcKeys() {  
 n = q \* p;  
 int phi = (p - 1) \* (q - 1);  
 for (int n = 2; n < phi; n++) {  
 if (RSAlgorithms.*greatestCommonDivider*(n, phi) == 1) {  
 e = n;  
 break;  
 }  
 }  
 for (int i = 1; ; i++) {  
 float v = (float) (1 + phi \* i) / e;  
 if (v - (int) v == 0) {  
 d = (int) v;  
 break;  
 }  
 }  
 }  
  
 public int getN() {  
 return n;  
 }  
  
 public int getE() {  
 return e;  
 }  
  
 public int getD() {  
 return d;  
 }  
}

Написав клас, що реалізує хешування повідомлення:

package solution;  
  
import java.util.List;  
import java.util.Random;  
  
public class RSAHasher {  
 private final int H\_0;  
 private final int n;  
  
 public RSAHasher(int n, int h\_0) {  
 this.n = n;  
 H\_0 = h\_0;  
 }  
  
 public RSAHasher(int n) {  
 this.n = n;  
 H\_0 = new Random().nextInt();  
 }  
  
 public int hash(List<Integer> M) {  
 int prevH = H\_0;  
 for (Integer num : M) {  
 prevH = RSAlgorithms.*discretePower*(  
 num + prevH,  
 2,  
 n  
 );  
 }  
 return prevH;  
 }  
  
}

Написав клас, для демонстрації роботи програми:

package solution;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Random;  
import java.util.Scanner;  
  
public class RSADemo {  
  
 private static List<Integer> pickRandomTwo(List<Integer> inputList) {  
 Random random = new Random();  
 final int N = inputList.size();  
 int a = Math.*abs*(random.nextInt()) % N;  
 int b = Math.*abs*(random.nextInt()) % N;  
 while (b == a) {  
 b = Math.*abs*(random.nextInt()) % N;  
 }  
 return List.*of*(inputList.get(a), inputList.get(b));  
 }  
  
 private static List<Integer> stringToIntList(String text) {  
 final int N = text.length();  
 List<Integer> integers = new ArrayList<>();  
 for (int i = 0; i < N; i++) {  
 integers.add((int)text.charAt(i));  
 }  
 return integers;  
 }  
  
 private static String intListToString(List<Integer> integers) {  
 final int N = integers.size();  
 StringBuilder builder = new StringBuilder();  
 for (int v : integers) {  
 builder.append((char) v);  
 }  
 return builder.toString();  
 }  
  
 public static void main(String[] args) {  
  
 *// Генеруємо 60 перших простих чисел і обираємо два довільних випадково* List<Integer> inputValues = *pickRandomTwo*(RSAlgorithms.*generatePrimeNumbers*(60));  
  
 final int p = inputValues.get(0);  
 final int q = inputValues.get(1);  
  
 *// Обєкт що описує RSA конфігурацію* RSA rsa = new RSA(p, q);  
  
 System.*out*.println("Параметри RSA: ");  
 System.*out*.println("p: " +p);  
 System.*out*.println("q: " +q);  
 System.*out*.println("n: " + rsa.getN());  
 System.*out*.println("e: " + rsa.getE());  
 System.*out*.println("d: " + rsa.getD());  
  
 *// Обєкт для знаходження хеш-коду* RSAHasher hasher = new RSAHasher(rsa.getN());  
  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.println("Введіть текст:");  
 while (scanner.hasNext()) {  
 String inputText = scanner.nextLine();  
 if ("stop".equals(inputText)) {  
 break;  
 }  
 List<Integer> inputList = *stringToIntList*(inputText);  
 final int m = hasher.hash(inputList);  
 final int S = RSAlgorithms.*discretePower*(m, rsa.getD(), rsa.getN());  
 final int mHashed = RSAlgorithms.*discretePower*(S, rsa.getE(), rsa.getN());  
 System.*out*.println("m = " + m);  
 System.*out*.println("S = " + S);  
 System.*out*.println("m' = " + mHashed);  
 System.*out*.println("Збігаються? " + ((m == mHashed) ? "Так" : "Ні"));  
 }  
  
 System.*out*.println("Програму завершено");  
 }  
  
}

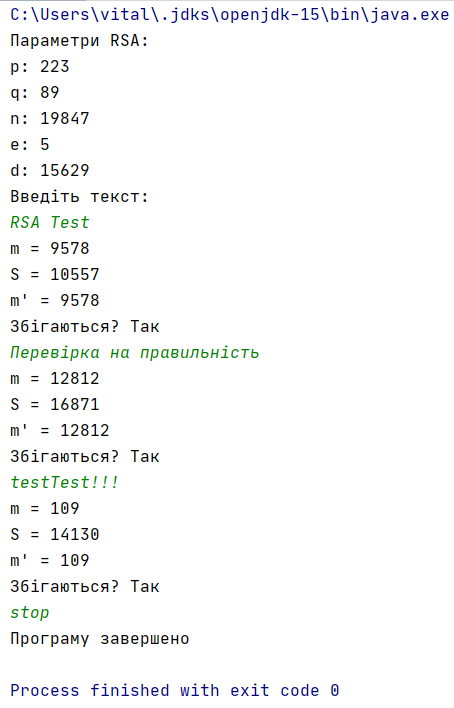


Рисунок 1. Демонстрація роботи програми

Висновок: на цій лабораторній роботі я засвоїв алгоритм RSA.