Міністерство освіти і науки України  
НТУУ «КПІ ім. І. Сікорського»

Кафедра цифрових технологій в енергетиці

Безпека інформаційних систем

Лабораторна робота № 5

«Алгоритм шифрування RSA»

Варіант №1

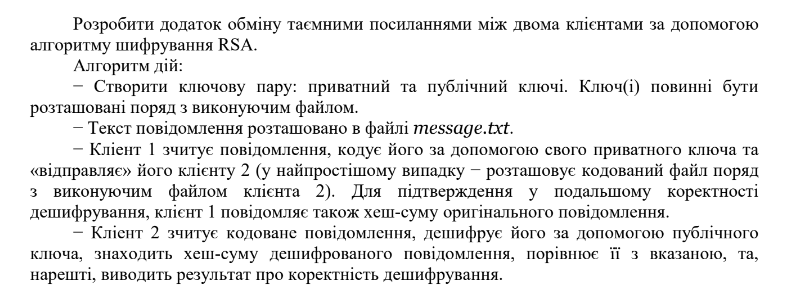
Виконав:  
студент 2-го курсу НН ІАТЕ  
групи ТР-15  
Руденко Владислав Ігорович

Перевірив:

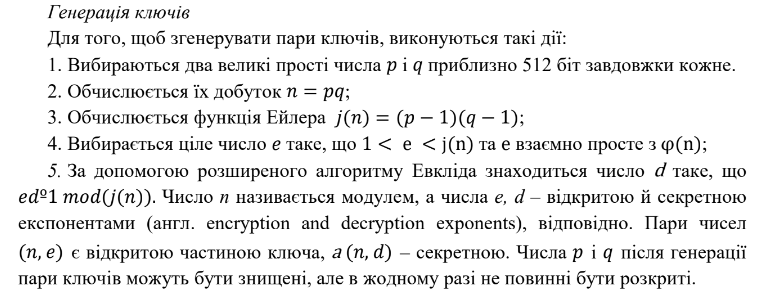
доц. Онисько А. І.

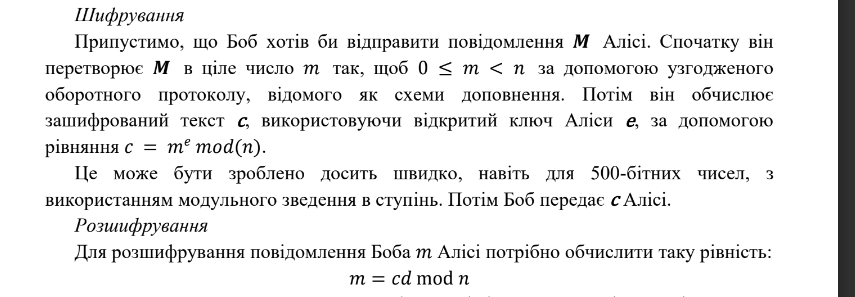
Київ – 2022

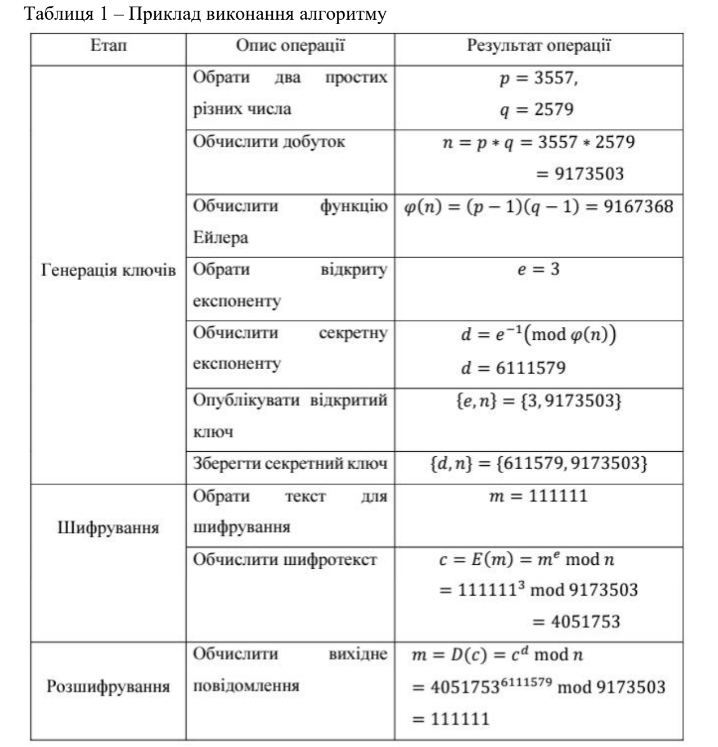
**Завдання**

****

**Теоретичні відомості**

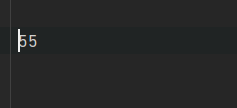
****

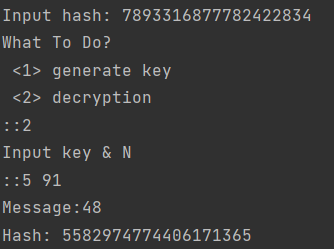
****

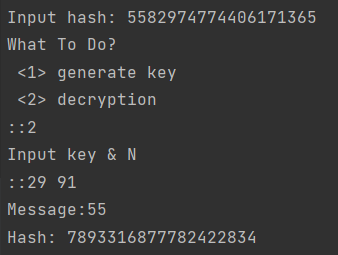


**Результати виконання**

****

****

****

****

**Висновок**

В ході виконання лабораторної роботи №5 здобуто навички асиметричного шифрування методом RES. Результатом роботи є алгоритм RES, який здатен шифрувати та дешифрувати деякий набір даних. В роботі представлено Відкриті та закриті ключі та приклад роботи програми з вхідним текстом в файлі

**Лістинг програми**

#include <fstream>  
#include <iomanip>  
#include <iostream>  
#include <string>  
  
using namespace std;  
  
class InputText  
{  
private:  
 string text;  
 string DIR = "message.txt";  
   
public:  
 string gettext(){return text;}  
 InputText()  
 {  
 ifstream File(DIR);  
 while(getline(File,text));  
   
 }   
};  
  
class SaveText  
{  
private:  
 string DIR = "message.txt";  
 ofstream File;  
public:  
 SaveText()  
 {  
 File.open(DIR);  
 File << endl;  
 }  
 void SaveTo(int temp)  
 {  
 hash<int> hash;  
 cout << "Message:" << temp << "\nHash: " << hash(temp) ;  
 File << temp;  
 }  
};  
  
  
class RSA  
{  
private:  
 InputText Text;  
 SaveText Save;  
  
 int Function(int a, int b)  
 {  
 if (b == 0)  
 return a;  
 else  
 return Function(b, a % b);  
 }  
 void key\_generation()  
 {  
 cout << "Input p & q keys in row\n::";  
 int p\_key,q\_key;  
 cin >> p\_key;  
 cin >> q\_key;  
   
 double n = p\_key\*q\_key;  
 int function = (p\_key-1)\*(q\_key-1);  
 int e = 2;  
 while (e < function)  
 {  
 if (Function(e, function) == 1)  
 break;  
 e++;  
 }  
 double d = (2\*function+1) / e;  
 cout << "Open key: " << e<<" : "<< fixed << setprecision(0) <<n << endl;  
 cout << "Private key: " << d<<" : "<< fixed << setprecision(0) <<n << endl;  
 }  
   
public:  
 void decryption(int (&key)[2])  
 {  
 long double temp = stod(Text.gettext());  
 Save.SaveTo(fmod(pow(temp,key[0]),key[1]));  
 }  
  
 RSA()  
 {  
 int choose;  
 hash<int> hash;  
 cout << "Input hash: " << hash(stoi(Text.gettext())) << endl;  
 cout << "What To Do?" << "\n <1> generate key\n <2> decryption" << endl <<"::";  
 cin >> choose;  
 switch (choose)  
 {  
 case 1:  
 key\_generation();  
 Save.SaveTo(stoi(Text.gettext()));  
 break;  
 case 2:  
 cout << "Input key & N\n::";  
 {  
 int key[2];  
 cin >> key[0] >> key[1];  
 decryption(key);  
 }  
 break;  
 default:  
 cout << "Invalid...";  
 break;  
 }  
 }  
};  
  
  
int main(int argc, char\* argv[])  
{  
 RSA rsa;  
 return 0;  
}

**Контрольні Запитання:**

1. **Чим відрізняються симетричний та асиметричний типи шифрування?**

Симетричне шифрування використовує один і той же ключ для шифрування та розшифрування повідомлення. Це швидкий та простий метод, але вимагає безпечного обміну спільним ключем між відправником і отримувачем.

Асиметричне шифрування використовує публічний ключ для шифрування та приватний ключ для розшифрування. Публічний ключ може бути розповсюджений відкрито, тоді як приватний ключ залишається таємним. Цей метод забезпечує більшу безпеку, але його обчислювальна складність вища, ніж у симетричного шифрування.

1. **Які переваги має симетричне шифрування? Наведіть приклади алгоритмів симетричного шифрування.**

Швидкість: Симетричне шифрування є швидшим у порівнянні з асиметричним. Операції шифрування та розшифрування можуть бути виконані значно швидше, оскільки використовується один ключ. Та простота Реалізації: Симетричні алгоритми шифрування зазвичай мають меншу складність і легше реалізовувати, що робить їх більш доступними для використання.

Прикладами алгоритмів можуть бути AES та DES, останній осучаснений до 3DES

1. **Які переваги та недоліки має асиметричне шифрування перед симетричним? Для чого доцільно використовувати асиметричне шифрування?**

Переваги: Безпека обміну, Цифровий підпис, Ключовий обмін.

Недоліки: Складність обчислень, Низька швидкість, Низька стійкість

Приклади використання Цифровий підпис, передача ключів

1. **Наведіть приклади алгоритмів асиметричного шифрування.**

RES,DSA

1. **Чим відрізняється відкритий ключ від секретного?**

Відрізняються тим, що відкритий ключ доступний для всіх і використовується для шифрування та перевірки підписів, тоді як секретний ключ є конфіденційним і використовується для розшифрування та підписування.

1. **З яких етапів складається алгоритм RSA?**

**1.**Генерація ключа

**2.**Шифрування

**3.**Передача Повідомлення

**4.**Розшифрування

**7. Які недоліки має алгоритм RSA?**

1.Складність обчислення

2.Розмір ключа

3. Вимога до безпечного зберігання приватного ключа

4.Продуктивність

1. **Як слід вибирати значення відкритого показника?**

Він має бути: Простим, Об’ємним, ефективним,стандартизованим, унікальним

1. **Як запобігти атаці по передбачуваному тексту?**

Використовувати стандартизовані протоколи шифрування та використовувати використовувати додаткову буферну інформацію на початку тексту

1. **Як можна створити труднощі щодо злому алгоритму RSA?**

1.Використовувати великий ключ.

2. Додатково захистити приватний ключ

3. Використовувати унікальні ключі

4.Використання надійних каналів для передачі ключів.