МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Информатики и систем управления»

ОТЧЕТ

по лабораторной работе

по дисциплине

Методы и средства защиты информации

ПРЕПОДАВАТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Капранов С.Н. (подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гореев А.Д. (подпись) (фамилия, и.,о.)

19-ИСТ-2

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2021

**Вариант № 7**

**Задание №8.1** Реализовать алгоритм шифрования данных «RSA».

**Алгоритм RSA**

Шаги алгоритма RSA

1. выбрать два больших простых числа p и q;
2. вычислить: n = p ⋅ q, m = (p – 1) ⋅ (q – 1);
3. выбрать случайное число d, взаимно простое с m;
4. определить такое число e, для которого является истинным выражение: (e ⋅ d) mod (m) = 1;
5. числа e и n – это открытый ключ, а числа d и n – это закрытый ключ;
6. На практике это означает следующее: открытым ключом зашифровывают сообщение, а закрытым – расшифровывают. Пара чисел закрытого ключа держится в секрете.
7. Разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа M(i);
8. Обычно блок берут равным одному символу и представляют этот символ в виде числа – его номера в алфавите или кода в таблице символов (например ASCII или Unicode).
9. Шифрование алгоритмом RSA производится по формуле: C(i) = (M(i)e) mod n;
10. Расшифровка сообщения производится с помощью формулы: M(i) = (C(i)d) mod n.

**Листинг программы**

#include <Windows.h>

#include <iostream>

#include <limits.h>

#include <cstdlib>

#include <string>

#include <ctime>

#include <cmath>

using namespace std;

typedef long long LL;

const LL MAX = SHRT\_MAX;

const LL MIN = SHRT\_MAX / 2 + 1;

LL p, q, n, e, d, f, mes;

//Поиск наибольшего общего делителя

int GCD(int a, int b)

{

if (b == 0)

{

return a;

}

return GCD(b, a % b);

}

LL modPow(LL x, LL e, LL n)

{

LL res = 1;

while (e>0)

{

if (e % 2 == 1)

{

res = (res \* x) % n;

}

e = e / 2;

x = (x \* x) % n;

}

return res;

}

//тест Миллера-Рабина

//Смотрим, является ли число простым или же составным

bool testMillerRabin(LL n)

{

LL a, s = 0, d = 1, x, temp = n - 1;

bool flag = false;

if (n % 2 == 0)

{

return false;

}

while (temp % 2 !=1)

{

s++;

temp /= 2;

}

for (int i = 0; i < 20; i++)

{

a = rand() % (n - 3) + 2;

x = modPow(a, d, n);

if (x == 1 || x == n - 1)

{

continue;

}

for (int r = 1; r < s; r++)

{

x = modPow(x, 2, n);

if (x == 1)

{

return false;

}

if (x == n - 1)

{

flag = true;

break;

}

}

if (flag)

{

continue;

}

return false;

}

return true;

}

//Генерация ключа

void keyGen()

{

do

{

p = rand() % MAX + MIN;

} while (!testMillerRabin(p));

do

{

q = rand() % MAX + MIN;

} while (!testMillerRabin(q));

n = p \* q;

f = (p - 1) \* (q - 1);

e = rand() % n;

cout << "p=" << p << "\n"

<< "q=" << q << "\n"

<< "N=" << n << "\n"

<< "f=" << f << "\n"

<< "e=" << e << "\n";

d = GCD(f, e);

cout << "d=" << d << endl;

}

//Алгорим RSA

string RSA(string& str)

{

keyGen();

string crypted;

string decrypted;

for (int i = 0; i < str.size(); i++)

{

mes = str[i];

crypted.push\_back(modPow(mes, e, n));

decrypted.push\_back(mes);

}

cout << "Зашифрованное сообщение\n" << crypted << endl;

cout << "Расшифрованное сообщение\n" << decrypted << endl;

return str;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

string message = "";

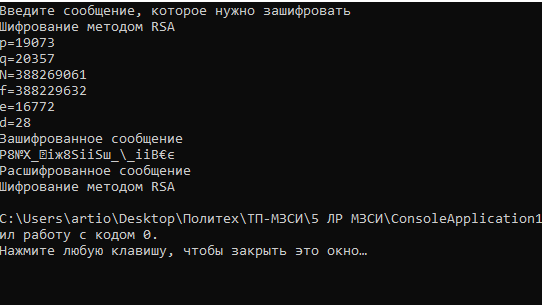
cout << "Введите сообщение, которое нужно зашифровать\n";

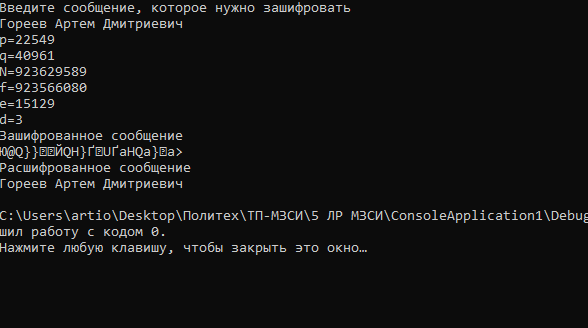
getline(cin, message);

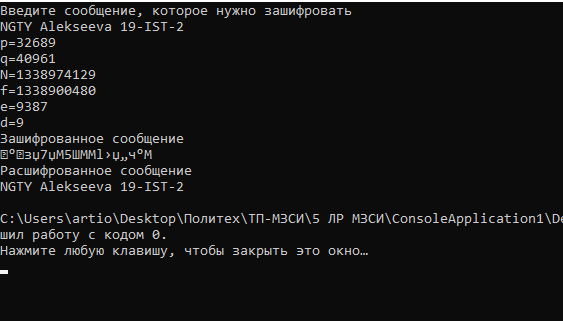
RSA(message)

}

**Примеры работы программы**

****

****

****