Министерство образования республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Индивидуальная работа №2

по дисциплине «Методы защиты информации»

Студент 4 курса

Группы № 493551

Рыбак Артем Владимирович

Минск, 2018

***Теоретические сведения***

Первые криптографические системы с открытым ключом или ассиметричные криптосистемы появились в конце 1970-х годов. От классических симметричных алгоритмов они отличаются тем, что для шифрования данных используется один ключ, обычно его называют открытый или публичный ключ, а для дешифрования ‑ другой, секретный или закрытый, ключ.

Диффи и Хеллман, которые впервые предложили и описали криптосистему с открытым ключом, выявляют следующие требования к криптосистемам:

1. Вычислительно легко создавать пару открытый ключ (Ko), закрытый ключ (Kc).

2. Вычислительно легко, имея открытый ключ и незашифрованное сообщение М, создать соответствующее зашифрованное сообщение: С= Еko[М].

3. Вычислительно легко дешифровать сообщение, используя закрытый ключ: М= DKc[C] = DKc [EKo[M]].

4. Вычислительно невозможно, зная открытый ключ Ko, определить закрытый ключ Kc.

5. Вычислительно невозможно, зная открытый ключ Ko и зашифрованное сообщение С, восстановить исходное сообщение М.

Можно добавить шестое требование, хотя оно не выполняется для всех алгоритмов с открытым ключом:

6. Шифрующие и дешифрующие функции могут применяться в любом порядке, т.е. М= ЕKo[DKc[M]] = DKc[EKo[M]].

Таким образом, данные, зашифрованные открытым ключом, можно расшифровать только секретным ключом. Следовательно, открытый ключ может распространяться через обычные коммуникационные сети и другие открытые каналы, что устраняет главный недостаток стандартных криптографических алгоритмов: необходимость использовать специальные каналы связи для распределения ключей.

***Криптосистема RSA***

В настоящее время лучшим и наиболее популярным криптографическим алгоритмом с открытым ключом считается ***RSA***, название которого получено по именам его создателей: ***Rivest***, ***Shamir*** и ***Adelman***.

Наиболее важной частью алгоритма ***RSA***, как и других алгоритмов с открытым ключом, является процесс создания пары открытый/секретный ключи. В ***RSA*** он состоит из следующих шагов:

1. Случайным образом выбираются два секретных простых числа ***p*** и ***q*** таких, что ***p ≈ q***.

2. Вычисляется их произведение ***r = p\*q***.

3. Вычисляется функция Эйлера для ***r***. Функция Эйлера для произвольного ***x*** представляет собой число взаимно простых с ***x*** чисел меньших, чем ***x***. Для простого ***x*** ***φ(x) = x‑1***, а для числа представляющего собой произведение двух простых чисел ***x= p\*q: φ(x) = (p‑1)\*(q‑1)***.

4. Выбирается целое значения открытой экспоненты ***e*** такой, что ***1<e<φ(r)*** и ***(e, φ(r)) = 1***.

5. Вычисляется значение секретной экспоненты ***d***, которая должна удовлетворять условию ***(e\*d) mod φ(r) = 1*** (т.е., ***d*** является мультипликативной инверсной по модулю ***φ(r)*** для ***e***).

Таким образом, ключом шифрования ***Ko***является пара значений ***(e, r)***, а ключом дешифрования ***Kc*** ‑ (***d, r***). Значение параметра ***r***, так же как и значение ***e*** являются общедоступной информацией, в то время как значения параметров ***p***, ***q*** и ***d*** хранятся в секрете.

Перед шифрованием исходное сообщения ***M*** необходимо разбить на блоки ***M = m1, m2, m3…***, где ***mi*** представляет собой число в диапазоне от ***0*** до ***r‑1***. Сам процесс шифрования открытым ключом ***Ko*** = (***e, r***) последовательности чисел ***mi*** происходит согласно формуле:

***ci = (mi e ) mod r***

где последовательность чисел ***ci*** представляет собой шифротекст.

Чтобы расшифровать эти данные секретным ключом ***Kc = (d, r),*** необходимо выполнить следующие вычисления:

***mi = (cid) mod r***

В результате будет получено множество чисел ***mi***, которые представляют собой исходный текст.

***Выполнение.***

***Листинг программы***

#include <conio.h> //Подключаемые файлы

#include <stdio.h>

#include <string.h>

#include <fstream.h>

char \*IshF = "c:\\Ish.txt"; //Файл с исходным текстом

char \*ShifrF = "c:\\Shifr.txt"; //Файл с зашифрованным текстом

char \*RasShifrF = "c:\\RasShifr.txt"; //Файл с расшифрованным текстом

unsigned long x1, Y1, p, q, e, r;

char \*simv = "ABCDHabcdefghijklmnopqrstuvwxyz .";

unsigned long mod(unsigned long a, unsigned long n)

{

unsigned long q;

q = a%n;

if (q < 0) q+=n;

return q;

}

void rashEvklid(unsigned long a, unsigned long n) //Расширенный метод Евклида

{

int i; //Вспомогательная переменная

unsigned long u[3], v[3], t[3], q; //Массивы для вычислений

u[0] = 0; u[1] = 1; u[2] = n;

v[0] = 1; v[1] = 0; v[2] = a;

t[0] = 0; t[1] = 0; t[2] = 0;

while (u[2] != 1)

{

q=u[2]/v[2];

for (i = 0; i<3; i++)

t[i] = u[i] - v[i]\*q;

for (i = 0; i<3; i++)

{

u[i] = v[i];

v[i] = t[i];

}

}

x1 = u[0]; //Сохранение вычислений

Y1 = u[1];

}

int bystrVozvStep(unsigned long a, unsigned long k, unsigned long n) //Быстрое возведение в степень

{

unsigned long b;

b = 1;

while (k != 0)

{

if (k%2 == 0)

{

k /= 2;

a = mod((a\*a), n);

}

else

{

k--;

b = mod((b\*a),n);

}

}

return b;

}

void shifrovanie(void) //Шифрование текста

{

char ch;

unsigned long i;

FILE \*f\_t, \*file2;

f\_t = fopen(IshF, "r");

file2 = fopen(ShifrF, "w");

while (fscanf(f\_t,"%c",&ch) != EOF)

for (i = 0; i < strlen(simv); i++)

if (simv[i] == ch)

fprintf(file2,"%d ", bystrVozvStep(i, e, r));

fclose(f\_t);

fclose(file2);

}

void rasshifruem(unsigned long d) //Расшифровывание текста

{

unsigned long j;

FILE \*fi,\*fo;

fi = fopen(ShifrF, "r");

fo = fopen(RasShifrF, "w");

while(fscanf(fi,"%d",&j) != EOF){

fprintf(fo,"%c",simv[bystrVozvStep(j, d, r)]);

}

fclose(fo);

fclose(fi);

}

void main(void)

{

unsigned long fr, d;

int i, j, k;

cout << " S H I F R O V A N I E\t R S A"<< endl;

cout << "Vvedite prostye chisla \n";

cout << "p = "; cin >> p;

if (p < 1){

p = 3;

printf("Ne vernoe znachenie p - budet ustanovleno po umolchaniju. p = %d", p);

}

cout << "q = "; cin >> q;

if (q < 1){

q = 11;

printf("Ne vernoe znachenie q - budet ustanovleno po umolchaniju. q = %d", q);

}

r = p \* q;

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

cout << "f(r) = " << fr <<endl;

e = 7;

cout << "Vzaimno prostoe e = " << e << endl;

rashEvklid(fr, e);

if (Y1 < 0)

Y1 += fr;

d = Y1;

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

cout << "Otkrytyj kljuch Ko (" << e << ", " << r << ")" << endl;

cout << "Sekretnyj kljuch Kc (" << d << ", " << r << ")" << endl;

cout << "\nFail s ishodnym tekstom: " << IshF << endl;

shifrovanie();

cout << "Fail s zashifrovannym tekstom: " << ShifrF << endl;

rasshifruem(d);

cout << "\n- - - - - - - - - - - - - - - - - - - - - - - - - - - -"<< endl;

cout << " R A S S H I F R O V K A\n"<< endl;

cout << "Fail s rasshifrovannym tekstom: " << RasShifrF << endl;

cout << "\n- - - - - - - - - - - - - - - - - - - - - - - - - - - -"<< endl;

cout << "\nPress any key to continue" << endl;

getch();

}

***Результат работы программы***

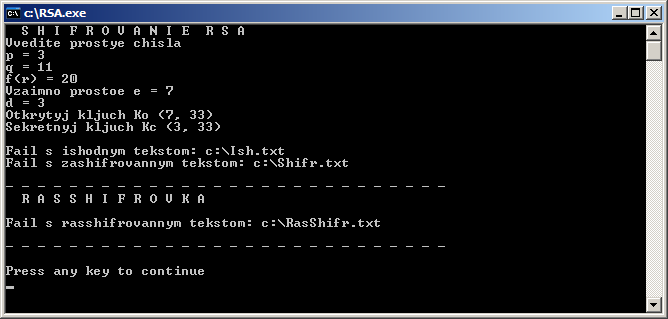


Рисунок 1 ‑ Работа программы

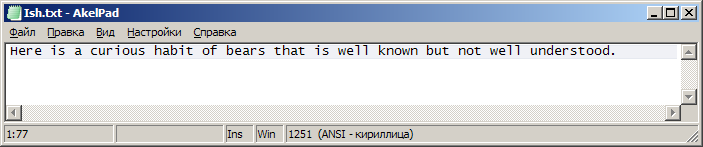


Рисунок 2 ‑ Исходный текст



Рисунок 3 ‑ Зашифрованный текст

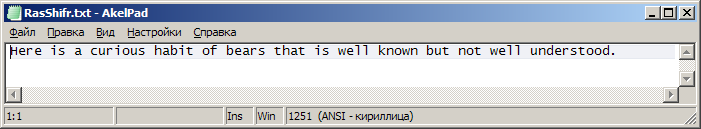


Рисунок 4 ‑ Расшифрованный текст