1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт прикладной математики и механики
5. **Кафедра «Информационная безопасность компьютерных систем»**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

1. «**Математические примитивы криптографии**»
2. по дисциплине «Основы информационной безопасности»
3. Выполнил
4. студент гр. 4851001/20001 Козлов О. И

<*подпись*>

1. Преподаватель
2. профессор Калинин М. О.

<*подпись*>

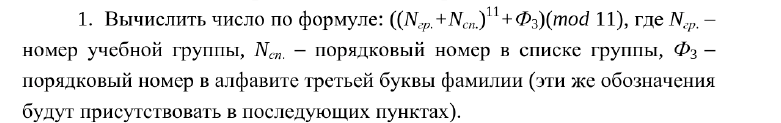
1. Санкт-Петербург
2. 2023

Оглавление:

1. [Математические вычисления и результаты вычислений 2](#_Toc127747392)
2. [Ответы на контрольные вопросы: 12](#_Toc127747393)
3. [Выводы: 14](#_Toc127747395)

[*Листинг программ (приложение 1) 15*](#_Toc127747396)

1. Математические вычисления и результаты вычислений



**Исходные данные:**

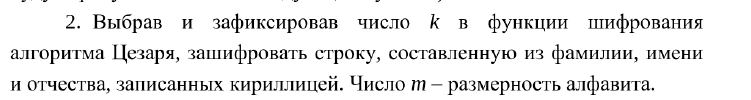
N*гр.* = 4851001

N*сп.* = 12

Ф*з* = 9

**Решение:**

((N*гр*.+ N*сп.*)11 + *Фз*) *(mod 11)* = ((4851001+12)11 + 9) (mod 11) = 35007155461393674171915354337783191831762272536474659532142390539229883037 mod 11 = 0

****

**Исходные данные:**

k = 3;

m = 33;

Строка = КОЗЛОВОЛЕГИЛЬИЧ

**Решение:**

К = 12 + 3(mod 33) = 15 => Н

О = 16 + 3(mod 33) = 19 => С

З = 9 + 3(mod 33) = 12 => К

Л = 13 + 3(mod 33) = 16 => О

В = 3 + 3(mod 33) = 6 => Е

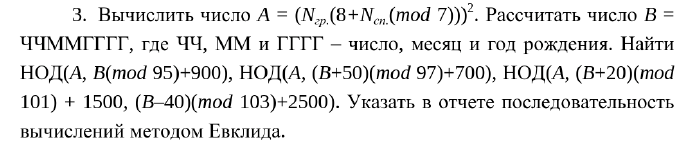
И = 10 + 3(mod 33) = 13 => Л

Ь = 30 + 3(mod 33) = 33 => Я

Ч = 25 + 3(mod 33) = 28 => Ъ

Г = 4 + 3(mod 33) = 7 => Ё

КОЗЛОВОЛЕГИЛЬИЧ => НСКОСЕСОЗЁЛОЯЛЪ



**Исходные данные:**

N*гр.* = 4851001

N*сп.* = 12

A **=** (Nгр. (8+Nсп. (mod 7)))2 = (4851001(8+12(mod 7)))2 = 3976943608638169

B = ЧЧММГГГГ=19062004

**Решение:**

1)НОД (A, B (mod 95) +900) = НОД (3976943608638169, 964) = 4

A = 3976943608638169, B = 964 = 3976943608638169 mod 964 = 829;

A = 964, B = 829 => 964 mod 829 = 135;

A = 829, B=135 => 829 mod 135 = 19;

A = 135; B = 19 => 135 mod 19 = 2;

A = 19; B = 2 => 19 mod 2 =1;

A = 2; B = 1 => 2 mod 1 = 0;

НОД = 1;

2) НОД (A, (B+50) (mod 97) +700) = НОД (3976943608638169, 702) =

A = 3976943608638169, B = 702 => 3976943608638169 mod 702 = 637;

A = 702, B = 637 => 702 mod 637 = 65;

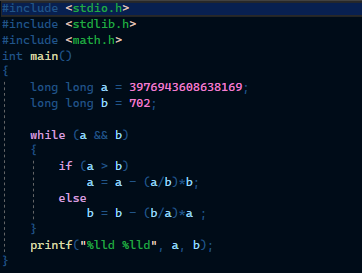
A = 637, B = 65 => 637 mod 65 = 52;

A = 65; B = 52 => 65 mod 52 = 13;

A = 52; B = 13 => 52 mod 13 = 0;

НОД = 13;

Программа для двух чисел:



3)НОД (A, (B+20) (mod 101) +1500), (B-40) (mod 103) + 2500)) =>

НОД (3976943608638169, 1592, 2563) = 1

A = 3976943608638169, B = 1592 => 3976943608638169 mod 1592 = 1201;

A = 1592; B = 1201 => 1592 mod 1201 = 391;

A = 1201; B = 391 => 1201 mod 391 = 28;

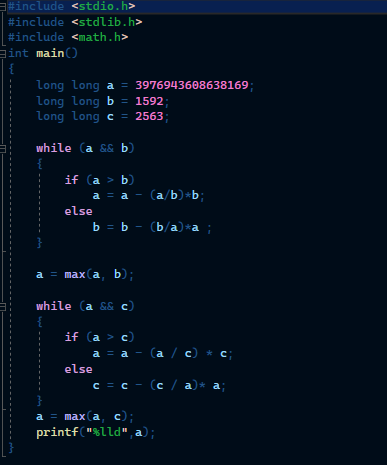
A = 391; B = 28 => 391 mod 28 = 27;

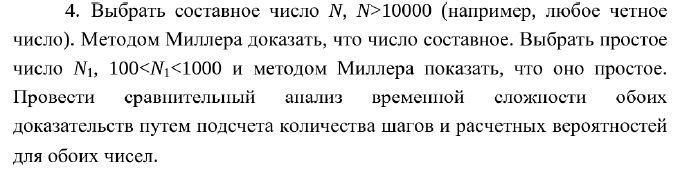
A = 28; B = 27 => 28 mod 27 = 1;

A = 27, B = 1 => 27 mod 1 = 0;

C = 2563, A =1 = > 2563 mod 1 = 0;

НОД = 1;





**Исходные данные:**

*Nсост* **=** 16825;

*Nпрост =* 167;

**Решение:**

1. Докажем, что *Nсост* **=** 16825 – составное методом Миллера;

N-1 *=* 2*s*t;

16825-1 = 23 \* 2103 => s = 3; t = 2103;

**Возьмём a = 5.** Тогда 16825/5 = 3365 => *Nсост* **=** 16825 – составное.

1. Докажем, что *Nпрост =* 167 – простое;

N-1 *=* 2*s*t;

167 – 1 = 21\*83 => s = 1; t = 83;

**Возьмем a = 3;**

167 не делится на 3 (167 mod 3 = 2);

at ≡ 1 (mod N*прост*), или существует целое k: 0≤k<s, такое, что

a2^(k-1)t = -1 (mod N*прост*)

355 1(mod 167)

**Возьмем a = 7;**

167 не делится на 7 (167 mod 7 = 6);

at ≡ 1 (mod N*прост*), или существует целое k: 0≤k<s, такое, что

a2^(k-1)t = -1 (mod N*прост*)

723 1(mod 167)

**Возьмем a = 11;**

167 не делится на 11 (167 mod 11 = 2);

at ≡ 1 (mod N*прост*), или существует целое k: 0≤k<s, такое, что

a2^(k-1)t = -1 (mod N*прост*)

1115 1(mod 167)

**Возьмем a = 34;**

167 не делится на 34 (167 mod 34 = 31);

at ≡ 1 (mod N*прост*), или существует целое k: 0≤k<s, такое, что

a2^(k-1)t = -1 (mod N*прост*)

344 1(mod 167)

**Возьмем a = 25;**

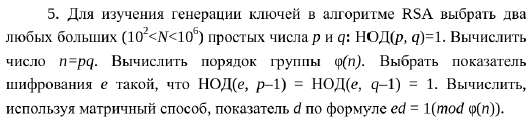
167 не делится на 25 (167 mod 25 = 17);

at ≡ 1 (mod N*прост*), или существует целое k: 0≤k<s, такое, что

a2^(k-1)t = -1 (mod N*прост*)

256 1(mod 167)

После k = 5 повторений по методу Миллера- N1 = 167 хороших чисел не было найдено. Тогда, вероятность, что 167 – составное число, не большей 1/1024 (1/4k).

****

**Исходные данные:**

* p = 383; q = 617(НОД (383, 617) = 1);
* n = pq = 383\*617 = 236311;
* φ(n) = (p-1) (q-1) = 382\*616 = 235312;
* Выберем e такой, что НОД (e, p-1) = НОД (e, q-1) = 1:
* Пусть e = 3, поскольку НОД (17, 383) = НОД (17, 617) = 1
* Вычислим d, используя матричные способ по формуле ed = 1(mod φ(n)):

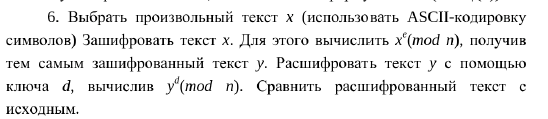
3d = 1(mod 235312) => d = 156875;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Шаг** | **большее** | **меньшее** | **частное** | **остаток** |
| 1 | 235312 | 3 | 78437 | 1 |
| 2 | 3 | 1 | 3 | 0 |
| 3 | 1 | 0 | - | - |

;

Проверим d =156875;

Значит открытый ключ (e, n) = (3, 236311), открытый ключ (d, n) = (156875,236311)



**Исходные данные:**

Пусть исходная строка x = “OLEG”;

e =3; n = 236311; d = 156875;

**Решение**

Тогда ASCII код для строки x будет:

O = 79

L = 76

E = 69

G = 71

Зашифруем ASCII код по формуле xe (mod n):

79 => 20417

76 => 202665

69 => 92198

71 => 121600

Дешифруем ASCII код по формуле yd (mod n):

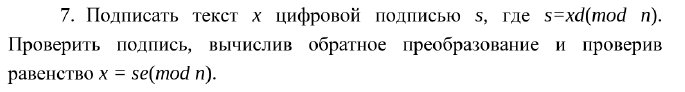
20417 => 79

202665 => 76

92198 => 69

121600 => 71

Следовательно, исходное шифрование и конечное совпадают



**Исходные данные:**

x = OLEG;

ASCII код строки x:

O = 79

L = 76

E = 69

G = 71

d = 156875;

n =236311;

e = 3;

**Решение задачи:**

s =xd(mod n)

79 => s = 135818

76 => s = 132496

69 => s = 92707

71 => s =169589

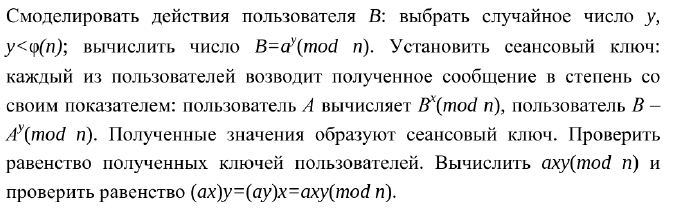
x=se (mod n)

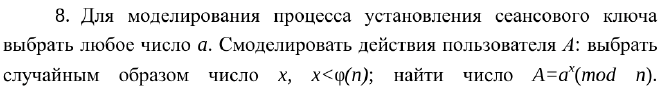
135818 => x = 79

132496 => x = 76

92707 => x = 69

169589 => x = 71

****

****

**Исходные данные:**

Пусть случайное числа a = 7; x =15; y = 6;

n = 236311;

φ(n) = 235312;

**Решение задачи:**

A=ax (mod n) =715 (mod 236311) = 27222;

B=ay (mod n) =76 (mod 236311) = 117649;

Проверка равенства сеансовых ключей:

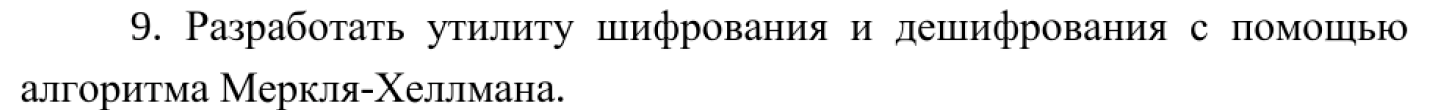
Bx (mod n) = 11764915 (mod 236311) = 43854

Ay (mod n) = 438546(mod 236311) = 43854

Проверить равенство:

axy (mod n) = 630

(ax)y = (ay)x =axy (mod n) = 630

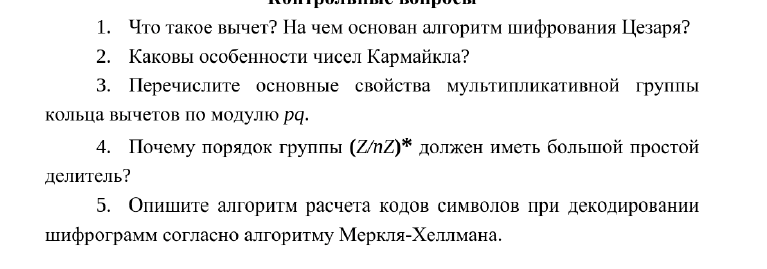
****

Приложение 1

**Проверка работы программы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Строка | Закрытый ключ | Открытый ключ | Шифрограмма | Дешифрованная строка | Дешифрованная табличная строка |
| OLEG | 11 20 42 78 155 313  624 1252 | 33 60 126 234 465 939 1872 1258 | 4594 1464 2257 4129 | 64 488 1585 2209 | 79 - O 76 - L 69 - E 71 - G |
| !IKIZI! | 19 38 68 135 275 548 1100 2194 | 38 76 136 270 550 1096 2200 1 | 137 627 2827 627 3096 627 137 | 2262 2507 3607 2507 1548 2507 2262 | 33 - ! 73 - I 75 - K 73 - I 90 - Z 73 - I 33 - ! |
| 714 | 4 9 15 39 78 148 304 605 | 32 72 120 312 624 1184 26 28 | 1670 460 1616 | 1111 659 202 | 55 - 7 49 - 1 52 - 4 |

2) Ответы на контрольные вопросы:

****

1. Число r называется вычетом числа a по модулю b, если разность (a-b)

делится на r без остатка (где a, b, r> 0 – целые числа).

Алгоритм шифрования Цезаря состоит из шифра замены, где каждый символ в открытом тексте заменяется символом, находящегося левее или правее данной символа в алфавите.

1. Числа Кармайлка – это составные числа, удовлетворяющие сравнению aN-1 = 1(mod N) для любого целого числа a с условием НОД(a,N) = 1 (a и N – взаимно простые).
2. Основные свойства мультипликативной группы кольца вычетов по модулю pq:

* Мультипликативная группа (Z/nZ)\* этого кольца является абелевой и циклической, состоит из ненулевых чисел, меньших n и взаимно простых с n;
* Остатки от деления образующей группы (Z/nZ)\* на p и q равны соответственно образующим мультипликативных групп полей Fp\* и Eq\*.
* Любой элемент кольца a € (Z/nz) может быть единственным образом представлен в виде a(mod p) и a(mod q) и обратно по китайской теореме об остатках
* Если a (mod p) = 0, то а принадлежит идеалу (p) и не является эл-том группы (Z/nZ) \*, при этом а образует в (Z/nZ) мультипликативную группу, изоморфную Fq\*;
* Единичным элементом в этой группе является эл-т, сравнимый с 1 по модулю q и с 0 по модулю p.

1. Порядок группы — её мощность, то есть количество элементов. Порядок группы равен значению функции Эйлера от n: #(Z/nZ)\*=φ(p)\*φ(q)=(p-1)(q-1).

Так как группа состоит из ненулевых чисел, взаимно простых и меньших n, n=pq, то взаимно простых с n чисел — (p-1)(q-1), так как из натуральных чисел меньше n исключаются все, кратные p и q.

1. На вход поступает шифрограмма . Вычисляется значение по формуле . Из полученного значения по убыванию вычитаются элементы закрытого ключа. Если элемент получилось вычесть из числа, тогда в битовом представлении исходного символа на месте по индексу элемента закрытого ключа должен стоять бит, равный единице. В результате двоичное число переводится в десятичное и по таблице ASCII переводится в символ.

3) Выводы:

В данной работе были изучены и отработаны базовые принципы криптографии, основы модульной арифметики, простейшие способы шифрования передаваемых сообщений, алгоритм RSA, ранцевая криптосистема Меркеля-Хеллмана.

4) Листинг программ (приложение 1)

Задание №9

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <time.h>

#include <malloc.h>

#include <math.h>

#include <string.h>

#include <locale.h>

#include <Windows.h>

int Evklid(int a, int b)

{

while (a && b)

{

if (a > b)

a = a - (a / b) \* b;

else

b = b - (b / a) \* a;

}

if (b > a) a = b;

if (a == 1)

return 1;

else return 0;

}

int Binary(int a, int openkey[])

{

if (a >= 0)

{

int shifor = 0;

for (int i = 7; i >= 0; i--)

{

if (a % 2 == 1)

shifor += openkey[i];

a /= 2;

}

return shifor;

}

else

{

a = abs(a)-1;

int shifor = 0;

for (int i = 7; i >= 0; i--)

{

if (a % 2 == 0)

shifor += openkey[i];

a /= 2;

}

return shifor;

}

}

int Desh(int closekey[], int itog, int m)

{

for (int i = 0; i < 256; i++)

{

int x = i, k = 0, sum = 0, bin[8] = { 0 }, j = 7;

for (int v = 7; v >= 0; v--) { bin[v] = x % 2; x /= 2; }

for (int v = 0; v < 8; v++) if (bin[v] == 1) sum += closekey[v];

if (sum == itog)

{

sum = 0;

int p = 0;

for (int v = 7; v >= 0; v--)

{

sum += bin[v] \* pow(2,p);

p++;

}

return sum;

}

}

}

void main()

{

setlocale(LC\_ALL, ".1251");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

char s[100];

scanf\_s("%100s", s, 99);

int length = strlen(s);

///

/// Генерация закрытого ключа

///

unsigned int closekey[8] = { 0 };

closekey[0] = rand() % 10;

int sum = closekey[0];

printf("Закрытый ключ: ");

for (int i = 0; i < 8; i++)

{

closekey[i] = rand() % 10 + sum + 1;

sum += closekey[i];

printf("%d ", closekey[i]);

}

printf("\n");

///

/// Получение чисел m и n

///

int m = sum + rand() % 10;

int n = rand() % 10 + 2;

while (Evklid(n, m) == 0)

{

n = rand() % 10 + 2;

}

///

/// Преобразование слова в ASCII код

///

int\* nums;

nums = (int\*)malloc(length \* sizeof(int));

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

{

nums[i] = (int)s[i];

}

///

/// Получение открытого ключа

///

unsigned int openkey[8] = { 0 };

printf("Открытый ключ: ");

for (int i = 0; i < 8; i++)

{

openkey[i] = (closekey[i] \* n) % m;

printf("%d ", openkey[i]);

}

printf("\n");

///

/// Получение шифрограммы

///

int\* itog;

itog = (int\*)malloc(length \* sizeof(int));

printf("Шифрограмма: ");

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

{

itog[i] = Binary(nums[i], openkey);

printf("%d ", itog[i]);

}

printf("\n");

///

/// Дешифрование

///

int no = 0;

while ((n \* no) % m != 1)

{

no++;

}

int\* deshifr;

deshifr = (int\*)malloc(length \* sizeof(int));

printf("Строка дешифрованая: ");

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

{

itog[i] = itog[i] \* no % m;

printf("%d ", itog[i]);

}

printf("\n");

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

{

deshifr[i] = Desh(closekey, itog[i], m);

char sym = deshifr[i];

printf("%d - %c ", deshifr[i], sym);

}

}