Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы защиты информации

ОТЧЁТ

по лабораторной работе

на тему

Ассиметричная криптография. Криптосистема Рабина

Выполнил

Студент гр. 053502

Юрьев В. А.

Проверил

Ассистент кафедры информатики

Лещенко Е. А.

Минск 2023

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc146017318)

[2 Теоретические сведения 4](#_Toc146017319)

[3 Блок-схема алгоритма 5](#_Toc146017320)

[4 Демонстрация работы 6](#_Toc146017321)

[5 Вывод 7](#_Toc146017322)

[Приложение А (обязательное) Код программы 8](#_Toc146017323)



1. **ПОСТАНОВКА ЗАДАЧИ**

Реализовать криптостойкое программное средство шифрования и дешифрования текстовых файлов при помощи Криптосистемы Рабина.

1. **ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Криптосистема Рабина используется для шифрования чисел. При шифровании и расшифровке используются публичный и секретный ключи. Секретный ключ представляет собой 2 больших взаимно простых числа p и q, которые соответствуют виду 4k + 3. Публичный ключ n является результатом произведения частей секретного ключа.

Для шифрования числа number необходимо вычислить:

c = number ^ 2 mod n

Для расшифровки необходимо вычислить:

1 mp = c(p + 1) / 4 mod p;

2 mq = c(q + 1) / 4 mod q.

Далее с помощью расширенного алгоритма Евклида вычисляются yp и yq, которые соответствуют равенству ypp + yqq = 1.

Затем вычисляются:

1 x1 = (yppmq + yqqmp) mod n;

2 x2 = (ypp(q - mq) + yqqmp) mod n;

3 x3 = (yppmq + yqq(p - mp)) mod n;

4 x4 = (ypp(q - mq) + yqq(p - mp)) mod n.

Одно из вычисленных значений является исходным числом.

1. **БЛОК-СХЕМА АЛГОРИТМА**

Схема алгоритма шифрования числа приведена на рисунке 3.1.

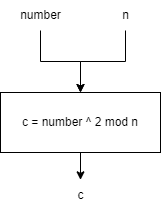


Рисунок 3.1 – Схема алгоритма шифрования числа

Схема алгоритма расшифрования числа приведена на рисунке 3.2.

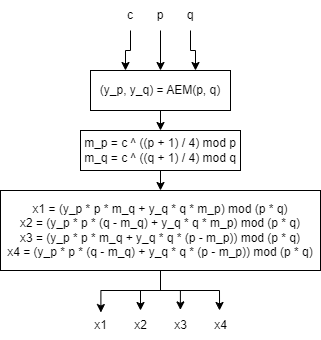


Рисунок 3.2 – Схема алгоритма расшифрования числа

1. **ДЕМОНСТРАЦИЯ РАБОТЫ**

Исходный текст считывается из файла input.txt текущей директории, после шифрования, результат записывается в файл encrypted\_message.txt. Пример работы программы показан на рисунке 4.1.

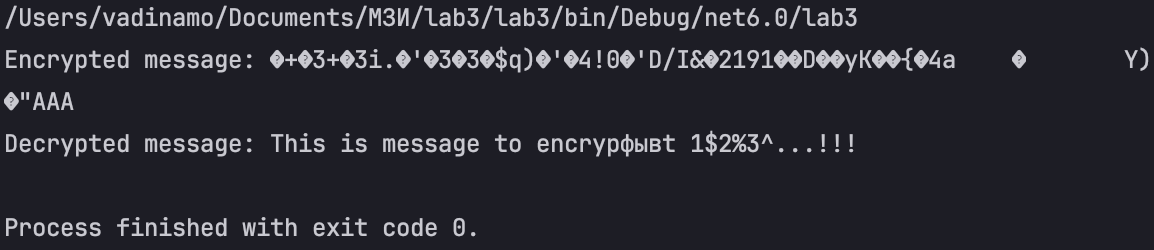


Рисунок 4.1 – Пример работы программы



**ВЫВОДЫ**

В ходе выполнения данной лабораторной работы был изучена криптосистема Рабина. Было разработано программное средство, которое шифрует и расшифровывает текстовые файлы алгоритмом криптосистемы Рабина. Недостатком данного алгоритма является 4 варианта исходного числа при расшифровке.

# **ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Код программы**

namespace lab3;

public static class Cryptographer

{

public static byte[] Encrypt(byte[] messageBytes, int n)

{

var result = new List<byte>();

foreach (var messageByte in messageBytes)

{

result.AddRange(BitConverter.GetBytes((ushort)(messageByte \* messageByte % n)));

}

return result.ToArray();

}

private static double ModularPow(double c, double k, int m)

{

var result = 1.0;

for (var i = 0; i < k; i++)

{

result = (result \* c) % m;

}

return result;

}

public static byte[] Decrypt(byte[] messageBytes, int p, int q)

{

var (yp, yq) = EuclidAlgorithm(p, q);

var result = new List<byte>();

for (var i = 0; i < messageBytes.Length / 2; i++)

{

var c = BitConverter.ToUInt16(messageBytes.Skip(i \* 2).Take(2).ToArray());

var variants = ChineseRemainderTheorem(c, p, q, yp, yq);

foreach (var variant in variants)

{

var literalBytes = BitConverter.GetBytes(variant);

if (literalBytes[1] == 0)

{

result.Add(literalBytes[0]);

}

}

}

return result.ToArray();

}

private static List<ushort> ChineseRemainderTheorem(ushort c, int p, int q, int yp, int yq)

{

var n = p \* q;

var mp = ModularPow(c, (p + 1) / 4.0, p);

var mq = ModularPow(c, (q + 1) / 4.0, q);

var r1 = (yp \* p \* mq + yq \* q \* mp) % n;

r1 += r1 < 0 ? n : 0;

var r2 = n - r1;

var r3 = (yp \* p \* mq - yq \* q \* mp) % n;

r3 += r3 < 0 ? n : 0;

var r4 = n - r3;

return new List<ushort>

{

(ushort)r1,

(ushort)r2,

(ushort)r3,

(ushort)r4

};

}

private static (int, int) EuclidAlgorithm(int p, int q)

{

int x0 = 1, x1 = 0, y0 = 0, y1 = 1;

while (q != 0)

{

var d = p / q;

var r = p % q;

p = q;

q = r;

var x = x0 - x1 \* d;

var y = y0 - y1 \* d;

x0 = x1;

y0 = y1;

x1 = x;

y1 = y;

}

return (x0, y0);

}

}