**Липецкий государственный технический университет**

Университетский колледж

Кафедра автоматизированных систем управления

Лабораторная работа №8

по «Защита информации»

Алгоритм шифрования RSA

Студент Щеглов Д.A

Группа T-ИС-20

Руководитель Седых Ю.И

Липецк 2022г.

Оглавление

[Цель работы 3](#_Toc98706477)

[Задание 4](#_Toc98706478)

# **Цель работы**

Изучить принцип работы асимметричных алгоритмов шифрования на примере алгоритма RSA.

# **Задание**

**RSA** (аббревиатура от фамилий создателей: Rivest, Shamir и Adleman) — один из самых популярных алгоритмов шифрования. Сначала приведем несколько определений:

mod — операция взятия остатка от деления.

Взаимно простыми называются такие числа, которые не имеют между собой ни одного общего делителя, кроме единицы.

Теперь опишем последовательность шагов алгоритма RSA:

* выбрать два больших простых числа p и q;
* вычислить: n = p ⋅ q, m = (p — 1) ⋅ (q — 1);
* выбрать случайное число d, взаимно простое с m;
* определить такое число e, для которого является истинным выражение: (e ⋅ d) mod (m) = 1;
* числа e и n — это открытый ключ, а числа d и n — это закрытый ключ;

На практике это означает следующее: открытым ключом зашифровывают сообщение, а закрытым — расшифровывают. Пара чисел закрытого ключа держится в секрете.

* разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа M(i);

Обычно блок берут равным одному символу и представляют этот символ в виду числа — его номера в алфавите или кода в таблице символов (например ASCII или Unicode).

* шифрование алгоритмом RSA производится по формуле: C(i) = (M(i)e) mod n;
* расшифровка сообщения производится с помощью формулы: M(i) = (C(i)d) mod n.

Программа «RSA»

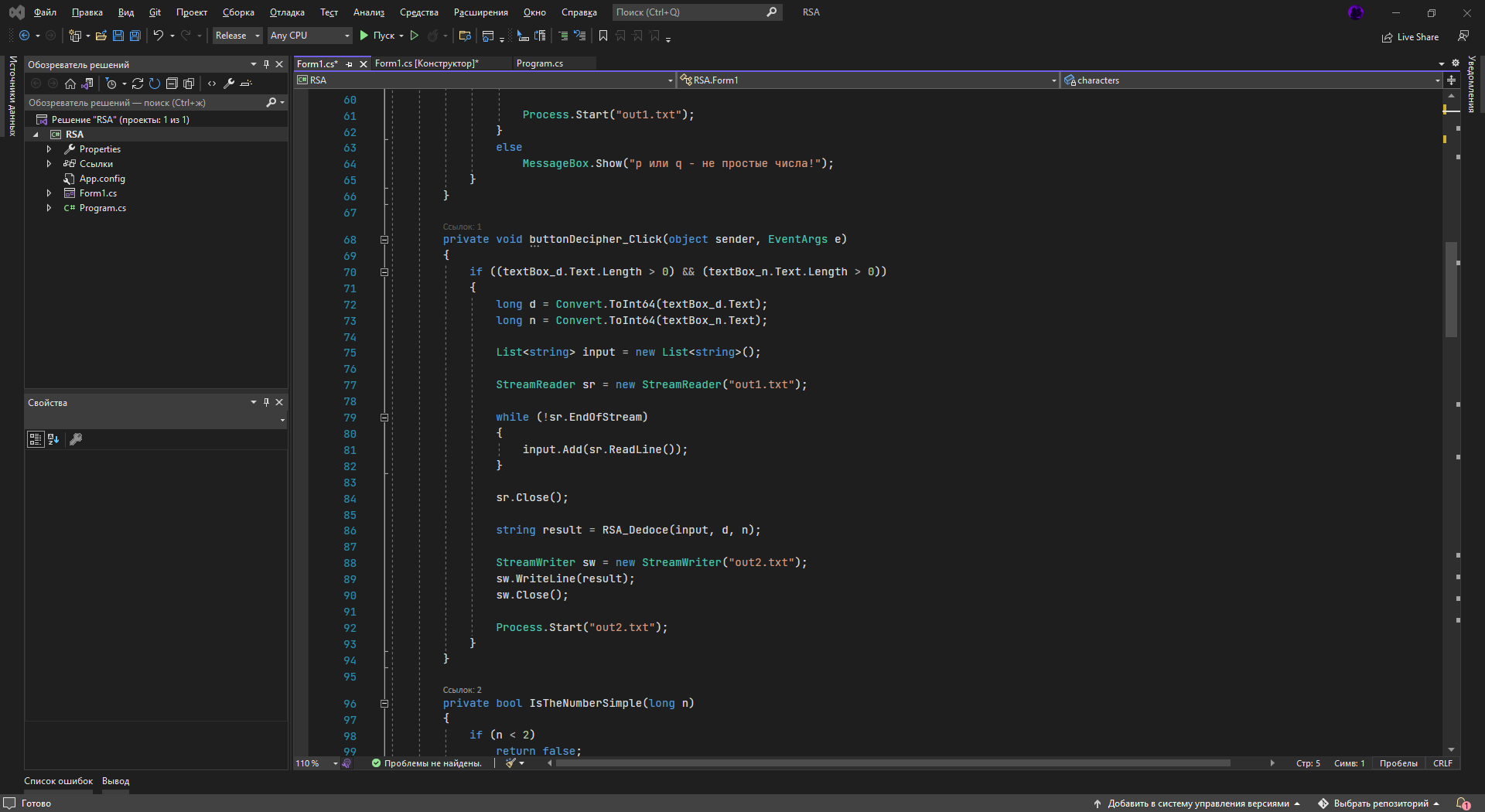


Рис.1

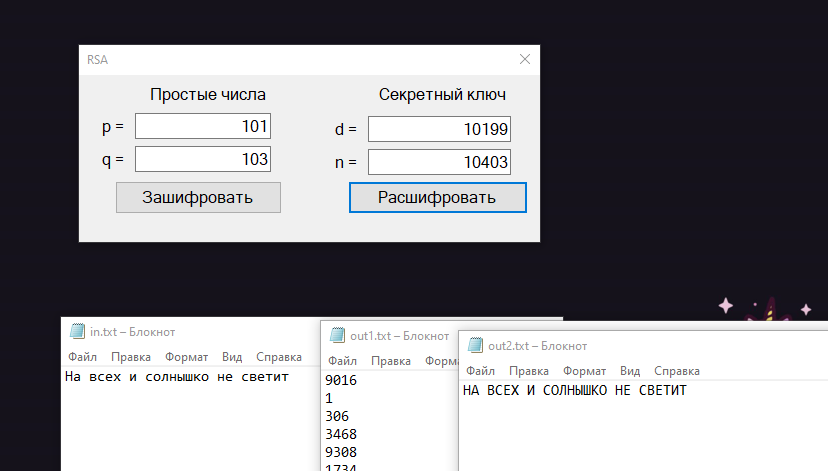


Рис.2

Программный код

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

using System.Windows.Forms;

using System.Numerics;

namespace RSA

{

public partial class Form1 : Form

{

char[] characters = new char[] { '#', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И',

'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С',

'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ь', 'Ы', 'Ъ',

'Э', 'Ю', 'Я', ' ', '1', '2', '3', '4', '5', '6', '7',

'8', '9', '0'

};

public Form1()

{

InitializeComponent();

}

private void buttonEncrypt\_Click(object sender, EventArgs e)

{

if ((textBox\_p.Text.Length > 0) && (textBox\_q.Text.Length > 0))

{

long p = Convert.ToInt64(textBox\_p.Text);

long q = Convert.ToInt64(textBox\_q.Text);

if (IsTheNumberSimple(p) && IsTheNumberSimple(q))

{

string s = "";

StreamReader sr = new StreamReader("in.txt");

while (!sr.EndOfStream)

{

s += sr.ReadLine();

}

sr.Close();

s = s.ToUpper();

long n = p \* q;

long m = (p - 1) \* (q - 1);

long d = Calculate\_d(m);

long e\_ = Calculate\_e(d, m);

List<string> result = RSA\_Endoce(s, e\_, n);

StreamWriter sw = new StreamWriter("out1.txt");

foreach (string item in result)

sw.WriteLine(item);

sw.Close();

textBox\_d.Text = d.ToString();

textBox\_n.Text = n.ToString();

Process.Start("out1.txt");

}

else

MessageBox.Show("p или q - не простые числа!");

}

}

private void buttonDecipher\_Click(object sender, EventArgs e)

{

if ((textBox\_d.Text.Length > 0) && (textBox\_n.Text.Length > 0))

{

long d = Convert.ToInt64(textBox\_d.Text);

long n = Convert.ToInt64(textBox\_n.Text);

List<string> input = new List<string>();

StreamReader sr = new StreamReader("out1.txt");

while (!sr.EndOfStream)

{

input.Add(sr.ReadLine());

}

sr.Close();

string result = RSA\_Dedoce(input, d, n);

StreamWriter sw = new StreamWriter("out2.txt");

sw.WriteLine(result);

sw.Close();

Process.Start("out2.txt");

}

}

private bool IsTheNumberSimple(long n)

{

if (n < 2)

return false;

if (n == 2)

return true;

for (long i = 2; i < n; i++)

if (n % i == 0)

return false;

return true;

}

//зашифровать

private List<string> RSA\_Endoce(string s, long e, long n)

{

List<string> result = new List<string>();

BigInteger bi;

for (int i = 0; i < s.Length; i++)

{

int index = Array.IndexOf(characters, s[i]);

bi = new BigInteger(index);

bi = BigInteger.Pow(bi, (int)e);

BigInteger n\_ = new BigInteger((int)n);

bi = bi % n\_;

result.Add(bi.ToString());

}

return result;

}

//расшифровать

private string RSA\_Dedoce(List<string> input, long d, long n)

{

string result = "";

BigInteger bi;

foreach (string item in input)

{

bi = new BigInteger(Convert.ToDouble(item));

bi = BigInteger.Pow(bi, (int)d);

BigInteger n\_ = new BigInteger((int)n);

bi = bi % n\_;

int index = Convert.ToInt32(bi.ToString());

result += characters[index].ToString();

}

return result;

}

//вычисление параметра d. d должно быть взаимно простым с m

private long Calculate\_d(long m)

{

long d = m - 1;

for (long i = 2; i <= m; i++)

if ((m % i == 0) && (d % i == 0))

{

d--;

i = 1;

}

return d;

}

//вычисление параметра e

private long Calculate\_e(long d, long m)

{

long e = 10;

while (true)

{

if ((e \* d) % m == 1)

break;

else

e++;

}

return e;

}

private void textBox\_p\_KeyPress(object sender, KeyPressEventArgs e)

{

char number = e.KeyChar;

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && number != 8 && (e.KeyChar <= 39 || e.KeyChar >= 46) && number != 47 && number != 61)

{

e.Handled = true;

}

}

private void textBox\_q\_KeyPress(object sender, KeyPressEventArgs e)

{

char number = e.KeyChar;

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && number != 8 && (e.KeyChar <= 39 || e.KeyChar >= 46) && number != 47 && number != 61)

{

e.Handled = true;

}

}

private void textBox\_d\_KeyPress(object sender, KeyPressEventArgs e)

{

char number = e.KeyChar;

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && number != 8 && (e.KeyChar <= 39 || e.KeyChar >= 46) && number != 47 && number != 61)

{

e.Handled = true;

}

}

private void textBox\_n\_KeyPress(object sender, KeyPressEventArgs e)

{

char number = e.KeyChar;

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && number != 8 && (e.KeyChar <= 39 || e.KeyChar >= 46) && number != 47 && number != 61)

{

e.Handled = true;

}

}

}

}