МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

«**Севастопольский государственный университет**»

кафедра Информационных систем

Институт информационных технологий и управления в технических системах

Лисянский Александр Игоревич

курс 4 группа ИС/б-42-о

09.03.02 Информационные системы (уровень бакалавриата)

**ОТЧЁТ**

о лабораторном практикуме №4

ИЗУЧЕНИЕ АЛГОРИТМОВ RSA.

по дисциплине «Основы защиты информации»

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность) (подпись) (инициалы, фамилия)

Севастополь

2016

**Цель работы**: Освоить механизм шифрования и дешифрования данных в криптографической системе с открытыми ключами RSA.

**Криптосистема RSA**

RSA – криптографическая система открытого ключа, обеспечивающая такие механизмы защиты как шифрование и цифровая подпись (аутентификация – установление подлинности).

Алгоритм RSA работает следующим образом:

Пусть p и q - два больших различных простых числа, и пусть n = p\*q и e некоторое целое, взаимно простое с (p-1)\*(q-1).

Оба соответствующих пространства открытых текстов Mk и зашифрованных сообщений Ck суть Zn - множество неотрицательных целых, меньших n. Если подлинное сообщение окажется слишком длинным, чтобы принадлежать Zn, его необходимо разбить на части и зашифровать, используя режим шифрования со сцеплением блоков.

Соответствующая ключу k функция шифрования Ek: Mk -> Ck определяется как Ek(m) = me mod(n). Для того, чтобы полностью определить естественный алгоритм ее вычисления достаточно записать e и n в открытый справочник. Такая пара называется открытым ключом, который легко вычисляется с помощью личного ключа .

Ek является кандидатом на однонаправленную функцию с потайным ходом, и хотя существует эффективный алгоритм вычисления обратной ей функции Dk, мы не знаем, как получить его эффективно, задаваясь только естественным алгоритмом вычисления Ek (т.е. только для заданных n и e).

Эффективный алгоритм вычисления Dk легко получить, задав дополнительную секретную информацию p и q. С этой целью, используя обобщенные алгоритмы Евклида для нахождения наибольшего общего делителя, чтобы вычислить целое число d, такое что e\*d = 1 mod ф(n), где ф(n) = (p-1)\*(q-1). По известной теореме Эйлера m(e\*d) = m mod(n) для каждого целого числа m и, следовательно, (me) d mod(n) = m, при условии что 0 <= m < n, что обеспечивается, когда m принадлежит Mk.

Функция дешифрования Dk: Ck -> Mk в связи с этим определяется как Dk(c) = md mod(n), и эффективный алгоритм для модульного возведения в степень также может быть использован и для ее вычисления. Тогда каждый пользователь криптосистемы RSA должен раз и навсегда выбрать случайно подходящие целые числа p, q и e и вычислить с их помощью d. После чего он делает свой открытый ключ доступным в пользовательском справочнике, тогда как d сохраняет в секрете. Это дает возможность любому другому пользователю шифровать посылаемые ему сообщения, которые только он и может расшифровать. Для того чтобы эта идея была реализована практически, решающим является удовлетворение требование, чтобы генерация больших случайных простых чисел и вычисление d были легко осуществимы. Функция дешифрования Dk: Ck -> Mk в связи с этим определяется как Dk(c) = md mod(n), и эффективный алгоритм для модульного возведения в степень также может быть использован и для ее вычисления. Суммируя все сказанное, тогда каждый пользователь криптосистемы RSA должен раз и навсегда выбрать случайно подходящие целые числа p, q и e и вычислить с их помощью d. После чего он делает свой открытый ключ доступным в пользовательском справочнике, тогда как d сохраняет в секрете.

**Ход работы**: В соответствии с вариантом задания был разработан алгоритм шифрования сообщений пользователя несколькими методами.

В результате было принято решение о создании графического интерфейса программы с возможностью выбора вида шифрования сообщения.

Проверяем работу программы. Введем самое простое выражение для двух видов кодировки

Рисунок 1 – работа RSA алгоритма

В результате проверки было получено заключение, что программа работает правильно, кодировка удается.

Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private int[] gamma;

private int[] gamma2;

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

{

Random rand = new Random();

gamma = new int[WordTextBox.TextLength];

string temp = "";

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

{

gamma[i] = (ATrackBar.Value \* rand.Next() + CTrackBar.Value) % (int)Math.Pow(2, BTrackBar.Value);

temp += (char)((int)WordTextBox.Text[i] + (int)gamma[i]);

}

EncryptTextBox.Text = temp;

}

private void Form1\_Load(object sender, EventArgs e)

{

ALabel.Text = "Коэффициент A = " + ATrackBar.Value;

CLabel.Text = "Коэффициент C = " + CTrackBar.Value;

BLabel.Text = "Коэффициент B = " + BTrackBar.Value;

A\_Label.Text = "Коэффициент A = " + A\_TrackBar.Value;

C\_Label.Text = "Коэффициент C = " + C\_TrackBar.Value;

B\_Label.Text = "Коэффициент B = " + B\_TrackBar.Value;

}

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

{

string temp = "";

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

{

temp += (char)((int)EncryptTextBox.Text[i] - (int)gamma[i]);

}

DecryptTextBox.Text = temp;

}

private void Encrypt\_Button\_Click(object sender, EventArgs e)

{

gamma2 = new int[Word\_TextBox.TextLength];

string temp = "";

try

{

Convert.ToInt32(T\_TextBox.Text);

}

catch

{

MessageBox.Show("Начальные условия\n введены не верно");

return;

}

int ch = Convert.ToInt32(T\_TextBox.Text);

gamma2[0] = (A\_TrackBar.Value \* ch + C\_TrackBar.Value) % (int)Math.Pow(2, B\_TrackBar.Value);

temp += (char)((int)Word\_TextBox.Text[0] + (int)gamma2[0]);

for (int i = 1; i < gamma2.Length; i++)

{

ch = (int)Word\_TextBox.Text[i - 1];

ch = ch % 2;

gamma2[i] = (A\_TrackBar.Value \* ch + C\_TrackBar.Value) % (int)Math.Pow(2, B\_TrackBar.Value);

temp += (char)((int)Word\_TextBox.Text[i] + (int)gamma2[i]);

}

Encrypt\_TextBox.Text = temp;

}

private void Decrypt\_Button\_Click(object sender, EventArgs e)

{

string temp = "";

for (int i = 0; i < gamma2.Length; i++)

{

temp += (char)((int)Encrypt\_TextBox.Text[i] - (int)gamma2[i]);

}

Decrypt\_TextBox.Text = temp;

}

}

}

Вывод: В результате выполнения лабораторной работы были получены навыки по разработке алгоритмов кодирования методами гаммирования, на основе полученных алгоритмов разработаны функции кодирования, а так же написана и отлажена программа с полученными функциями. Результаты тестирования программы показали, что алгоритмы разработаны верно, программа работает правильно.