Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ивановский государственный энергетический университет имени В.И. Ленина»

Кафедра программного обеспечения компьютерных систем

Отчет по лабораторной работе №5

Ассиметричные криптосистемы

Криптосистема Рабина

Выполнила студентка гр. 4-41 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Митюшина А.А.

Проверил

к.т.н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнатьев Е.Б.

Иваново, 2019

**Задание**

Написать программу, реализующую криптосистему Рабина.

**Листинг**

class Rabin

{

//работает//static BigInteger p = 199, q = 139, OpenyKey = p \* q;

static BigInteger p = 3004919, q = 20979403, OpenyKey = p \* q; // не работает(

public static void EncryptionBigText()

{

string textPath = @".\text\example.txt";

FileStream fstream = new FileStream(textPath, FileMode.Open);

StreamWriter streamWriterM = new StreamWriter(@".\text\listofM.txt");

StreamWriter streamWriter = new StreamWriter(@".\text\code.txt");

//определение длины сообщения М

//должно выполнятся условие M<C

//Чтобы оно было всегда меньше, то я беру блок на один бит меньше

byte[] array = OpenyKey.ToByteArray();

int maxLenghtM = array.Length - 1;

//массив для записи в него сообщения М

byte[] buf = new byte [maxLenghtM];

BigInteger M, C;

while (fstream.Length - fstream.Position > maxLenghtM)

{

fstream.Read(buf, 0, maxLenghtM);

M = new BigInteger(buf);

streamWriterM.Write(M); streamWriterM.Write(" ");

C = BigInteger.ModPow(M, 2, OpenyKey);

streamWriter.Write(C.ToString() + " ");

}

buf = new byte[fstream.Length - fstream.Position];

fstream.Read(buf, 0, Convert.ToInt32(fstream.Length - fstream.Position));

M = new BigInteger(buf);

streamWriterM.Write(M);

C = BigInteger.ModPow(M, 2, OpenyKey);

streamWriter.Write(C.ToString());

streamWriter.Close();

streamWriterM.Close();

}

public static void DecryptionBigText()

{

StreamReader streamReader = new StreamReader(@".\text\code.txt");

StreamWriter streamWriter = new StreamWriter(@".\text\decode.txt");

string c = "";

BigInteger C;

int sr=0;

while (sr != -1)

{

c = "";

sr = streamReader.Read();

while (sr != 32 && sr != -1)

{

c = c + Convert.ToChar(sr);

sr = streamReader.Read();

}

C = new BigInteger();

C = BigInteger.Parse(c);

BigInteger m1, m2, m3, m4, a, b, M1, M2, M3, M4;

m1 = BigInteger.ModPow(C, (p + 1) / 4, p);

m2 = p - m1;

m3 = BigInteger.ModPow(C, (q + 1) / 4, q);

m4 = q - m3;

a = q \* BigInteger.ModPow(q, p - 2, p);

b = p \*BigInteger.ModPow(p, q - 2, q);

M1 = (BigInteger.Multiply(a, m1) + BigInteger.Multiply(b, m3)) % OpenyKey;

M2 = (BigInteger.Multiply(a, m1) + BigInteger.Multiply(b, m4)) % OpenyKey;

M3 = (BigInteger.Multiply(a, m2) + BigInteger.Multiply(b, m3)) % OpenyKey;

M4 = (BigInteger.Multiply(a, m2) + BigInteger.Multiply(b, m4)) % OpenyKey;

streamWriter.Write("M1="); streamWriter.Write(M1); streamWriter.Write(" M2="); streamWriter.Write(M2); streamWriter.Write(" M3=");

streamWriter.Write(M3); streamWriter.Write(" M4="); streamWriter.Write(M4); streamWriter.WriteLine();

}

streamWriter.Close();

}

static BigInteger subtraction (BigInteger minuend, BigInteger subtrahend, BigInteger mod)

{

BigInteger result = minuend - subtrahend;

return result < 0 ? minuend + (mod -subtrahend) : result;

}

}

class Program

{

static void Main(string[] args)

{

try

{

Rabin.EncryptionBigText();

Rabin.DecryptionBigText();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

}

**Результат работы программы**

Первый файл – набор значений для кодирования M. Второй файл – закодированный текст. Третий – варианты раскодирования.

