Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных машин

Отчет

По лабораторной работе №2

«Криптосистема без передачи ключей»

|  |  |  |
| --- | --- | --- |
| Выполниил:  Ст. гр. 050501  Тускенис Д. С. |  | Проверил:  Кобяк И. П. |

Минск 2014

1. **ПОСТАНОВКА ЗАДАЧИ**

Абоненты A и B условились организовать между собой секретную от абонента C переписку. С этой целью они выбирают числа-ключи a и b, которые являются первыми секретными ключами участников связи. Для передачи сообщения должен быть выбран модуль: простое число p такое, чтобы значение хорошо разлагалось на достаточное число сомножителей. На основании закрытых ключей выбираются сопутствующие параметры α и β, которые являются вторыми секретными ключами.

Условия выбора ключей:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Цикл передачи сообщения состоит из следующих шагов:

1. Абонент A, используя первый секретный ключ, формирует сообщение :
2. Абонент B, получив сообщение , шифрует его своим первым ключом и формирует сообщение :
3. Абонент A своим вторым секретным ключом формирует сообщение :
4. Абонент B своим вторым секретным ключом формирует сообщение , которое и является исходным сообщением.

Зная значения или , а также модуль , необходимо определить все секретные ключи абонентов A и B.

1. **АЛГОРИТМ РЕШЕНИЯ**
2. Перебираются все пары простых чисел меньших , которые будут являться первыми секретными ключами ( и ).
3. Для каждой такой пары определяются вторые секретные ключи α и β исходя из соотношения (1).
4. Провести цикл передачи сообщения или с использованием ключей, полученных в предыдущих пунктах.
5. Если в результате выполнения пункта 3 конечное сообщение эквивалентно исходному ( или ), то данная комбинация ключей является одним из возможных решений.
6. **ЛИСТИНГ ПРОГРАММЫ**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Numerics;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

class Program

{

static int[] primeNumbers = new int[]

{ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53 };

const int p = 53;

const int a = 37;

const int b = 41;

static int μ = 17;

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.Unicode;

Console.WriteLine("a = " + a + "\tb = " + b + "\tp = " + p + "\tμ = " + μ);

int α = Key(a, p);

int β = Key(b, p);

Console.WriteLine("α = " + α + "\tβ = " + β);

int μ1 = (int)BigInteger.ModPow(μ, a, p);

int μ2 = (int)BigInteger.ModPow(μ1, b, p);

int μ3 = (int)BigInteger.ModPow(μ2, α, p);

int μ4 = (int)BigInteger.ModPow(μ3, β, p);

Console.WriteLine();

HashSet<int> equals = new HashSet<int>();

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

{

for (int j = 0; j < primeNumbers.Length; j++)

{

int \_a = primeNumbers[i];

int \_b = primeNumbers[j];

int \_α = Key(\_a, p);

int \_β = Key(\_b, p);

if (\_α != 0 && \_β != 0)

{

int r = (primeNumbers[i] \* primeNumbers[j]) % (p - 1);

if (equals.Add(r))

{

if (μ2 == Exchange(μ2, \_α, \_β, \_a, \_b))

{

Console.WriteLine("a = " + \_a + "\tb = " + \_b +

"\tα = " + \_α + "\tβ = " + \_β);

}

}

}

}

}

Console.WriteLine("\nPress any key...");

Console.ReadKey();

}

static int Exchange(int μ, int a, int b, int α, int β)

{

int μ1 = (int)BigInteger.ModPow(μ, a, p);

int μ2 = (int)BigInteger.ModPow(μ1, b, p);

int μ3 = (int)BigInteger.ModPow(μ2, α, p);

int μ4 = (int)BigInteger.ModPow(μ3, β, p);

return μ4;

}

static int Key(int a, int p)

{

for (int i = 1; (i < p - 1); i++)

{

if ((a \* i) % (p - 1) == 1)

{

return i;

}

}

return 0;

}

}

}

1. **РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ**

