Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.01 – «Информатика и вычислительная техника»

Дисциплина: «Защита информации»

Профиль: «Автоматизированные системы обработки информации и управления»

Семестр 7

ОТЧЕТ

по лабораторной работе №3

Тема: «Несимметричные алгоритмы шифрования»

Выполнил: студент группы АСУ-19-1б

Шеретов М. А. \_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

Пермь, 2022

**ЦЕЛЬ РАБОТЫ**

Получить практические навыки по использованию несимметричных алгоритмов шифрования, на примере использования алгоритма Диффи-Хеллмана.

**ЗАДАНИЕ**

**Вариант №6**. Выполнить шифрование текста методом Диффи-Хеллмана, используя в качестве x и y простые числа с разрядностью не меньшей двадцати.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**Алгоритм Диффи-Хеллмана:**

Алгоритм назван по фамилиям его создателей Диффи (Diffie) и Хеллмана (Hellman).

Метод помогает обмениваться секретным ключом для симметричных криптосистем, но использует метод, очень похожий на асимметричный алгоритм RSA. Это не симметричный алгоритм, так как для шифрования и дешифрования используются различные ключи. Так же это не схема с открытым ключом, потому что ключи легко получаются один из другого, и ключ шифрования и ключ дешифрования должны храниться в секрете.

Определим круг его возможностей. Предположим, что двум абонентам необходимо провести конфиденциальную переписку, а в их распоряжении нет первоначально оговоренного секретного ключа. Однако, между ними существует канал, защищенный от модификации, то есть данные, передаваемые по нему, могут быть прослушаны, но не изменены (такие условия имеют место довольно часто). В этом случае две стороны могут создать одинаковый секретный ключ, ни разу не передав его по сети, по следующему алгоритму.

Предположим, что обоим абонентам известны некоторые два числа v и q. Они, впрочем, известны и всем остальным заинтересованным лицам. Например, они могут быть просто фиксированно «зашиты» в программное обеспечение. Далее один из партнеров P1 генерирует случайное или псевдослучайное простое число x и посылает другому участнику будущих обменов P2 значение A = qx mod n

По получении А партнер P2 генерирует случайное или псевдослучайное простое число у и посылает P2 вычисленное значение B = qy mod n

Партнер P1, получив В, вычисляет Kx = Bx mod n, а партнер P2 вычисляет Ky = Ay mod n. Алгоритм гарантирует, что числа Ky и Kx равны и могут быть использованы в качестве секретного ключа для шифрования. Ведь даже перехватив числа А и В, трудно вычислить Kx или Ky.

Например, по вычисленным Kx =Ky=K абоненты могут зашифровать сообщение M=123 по следующему алгоритму: к каждому символу сообщения M добавить K => сообщение С=234, при K=1. Соответственно алгоритмом расшифрования будет разность ключа K из каждого символа сообщения C.

Пример:

Пусть

n=3; q=5;

x=5; y=7;

тогда A = q^x mod n = 1, а B = q^y mod n = 2, то вычислив Kx = B^x mod n и Ky = A^y mod n получим Kx= Ky=1. Зашифруем приведенное выше сообщение M=123 по приведенному выше алгоритму => сообщение С=234, расшифровав сообщение C по обратному алгоритму получим сообщение M=123.

Необходимо еще раз отметить, что алгоритм Диффи-Хеллмана работает только на линиях связи, надежно защищенных от модификации. Если бы он был применим на любых открытых каналах, то давно снял бы проблему распространения ключей и, возможно, заменил собой всю асимметричную криптографию. Однако, в тех случаях, когда в канале возможна модификация данных, появляется очевидная возможность вклинивания в процесс генерации ключей «злоумышленника-посредника» по той же самой схеме, что и для асимметричной криптографии.

**ХОД РАБОТЫ**

На рисунке 1 представлена главная форма программы.

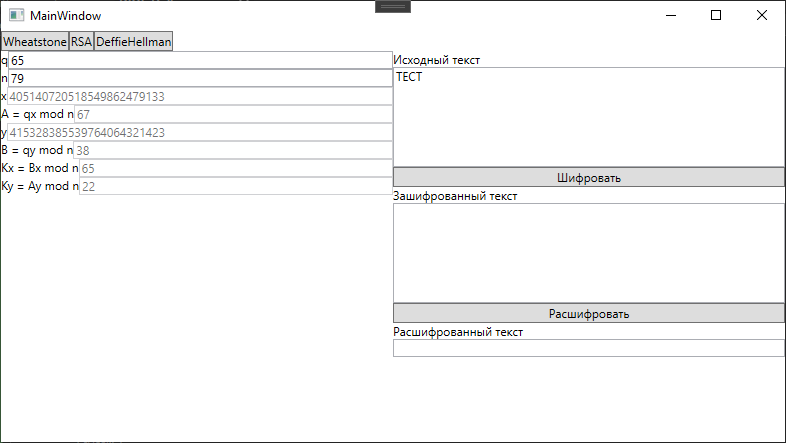


Рисунок 1 – Главная форма программы.

Пример работы программы представлен на рисунке 2.

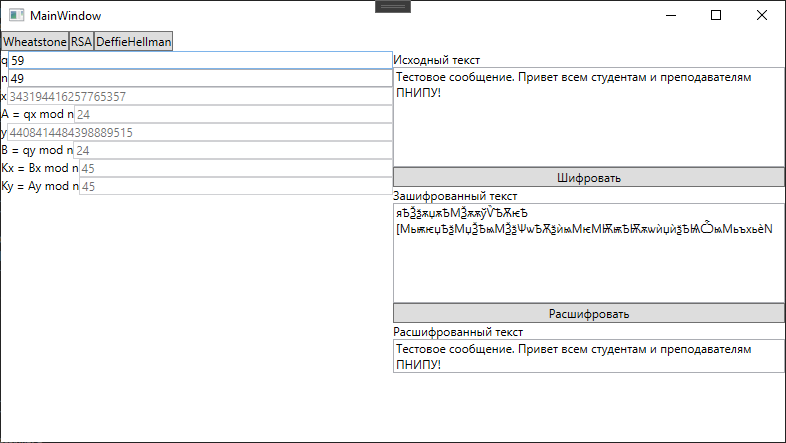


Рисунок 2 – Пример работы программы.

**ПРИЛОЖЕНИЕ А**

**Листинг класса Diffie\_Hellman**

public class Diffie\_Hellman

{

#region Properties

private BigInteger \_Q { get; set; }

public BigInteger Q { get => \_Q; set { \_Q = value; CalculateValues(); } }

private BigInteger \_N { get; set; }

public BigInteger N { get => \_N; set { \_N = value; CalculateValues(); } }

public BigInteger \_X { get; private set; }

public BigInteger \_A { get; private set; }

public BigInteger \_Y { get; private set; }

public BigInteger \_B { get; private set; }

public BigInteger \_Kx { get; private set; }

public BigInteger \_Ky { get; private set; }

#endregion

public Diffie\_Hellman(BigInteger q, BigInteger n)

{

\_Q = q;

\_N = n;

CalculateValues();

}

private void CalculateValues()

{

\_X = DiffieHellmanUtils.GenerateBigInt(80);

\_Y = DiffieHellmanUtils.GenerateBigInt(80);

\_A = BigInteger.ModPow(\_Q, \_X, \_N);

\_B = BigInteger.ModPow(\_Q, \_Y, \_N);

\_Kx = BigInteger.ModPow(\_B, \_X, \_N);

\_Ky = BigInteger.ModPow(\_B, \_Y, \_N);

}

public string Encrypt(string message)

{

string res = "";

foreach (var ch in message)

{

res += (char)(ch + \_Kx);

}

return res;

}

public string Decrypt(string message)

{

string res = "";

foreach (var ch in message)

{

res += (char)(ch - \_Ky);

}

return res;

}

}

public static class DiffieHellmanUtils

{

public static BigInteger Pow(this BigInteger a, BigInteger n)

{

var res = a;

for (int i = 0; i < n; i++)

{

res \*= a;

}

return res;

}

private static Random Gen = new Random();

public static BigInteger GenerateBigInt(int bits)

{

while (true)

{

var rng = new RNGCryptoServiceProvider();

byte[] bytes = new byte[bits / 8];

rng.GetBytes(bytes);

BigInteger p = new BigInteger(bytes);

if(!p.IsEven && !p.IsPowerOfTwo)

return BigInteger.Abs(p);

}

}

}