Лабораторная работа № 5

по информационной безопасности

студента группы ИТ-32

Манукова Давида Альбертовича

Выполнение: \_\_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_

**Алгоритм обмена ключами Диффи-Хеллмана**

Цель работы: изучить и реализовать на практике алгоритм обмена ключами Диффи-Хеллмана.

Содержание работы

Реализовать приложение, позволяющее выполнять следующие действия:

1. Изучить схему обмена ключами Диффи-Хелмана.

2. Реализовать подпрограмму, определяющую для заданного числа первые 100 первообразных корней, отображая при этом суммарное время, затраченное на их поиск. Число может задаваться десятеричной, шестнадцатеричной и двоичной формах.

3. Вручную для первых 5 полученных числовых значений привести доказательство, что они действительно являются первообразными корнями заданного числа n.

4. Реализовать подпрограмму, моделирующую обмен ключами между абонентами по схеме Диффи-Хеллмана. Программа должна получать большие простые числа XA, XB и n случайным образом с помощью алгоритма генерации простого числа, а также предоставлять пользователю возможность задавать их.

Ход работы

1. Разработал программу, реализующую процесс шифрования/дешифрования.

using System;

using System.IO;

using System.Security.Cryptography;

using System.Text;

namespace IS\_Lab\_5

{

class Alice

{

public static byte[] alicePublicKey;

public static void Main(string[] args)

{

using (ECDiffieHellmanCng alice = new ECDiffieHellmanCng())

{

alice.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;

alice.HashAlgorithm = CngAlgorithm.Sha256;

alicePublicKey = alice.PublicKey.ToByteArray();

Bob bob = new Bob();

CngKey bobKey = CngKey.Import(bob.bobPublicKey, CngKeyBlobFormat.EccPublicBlob);

byte[] aliceKey = alice.DeriveKeyMaterial(bobKey);

byte[] encryptedMessage = null;

byte[] iv = null;

string str = "Лишь сделанные на базе интернет-аналитики выводы представляют собой не что иное, как квинтэссенцию победы маркетинга над разумом и должны быть объективно рассмотрены соответствующими инстанциями. Прежде всего, высокое качество позиционных исследований обеспечивает широкому кругу (специалистов) участие в формировании системы обучения кадров, соответствующей насущным потребностям. Кстати, базовые сценарии поведения пользователей, инициированные исключительно синтетически, рассмотрены исключительно в разрезе маркетинговых и финансовых предпосылок.";

Send(aliceKey, str, out encryptedMessage, out iv);

bob.Receive(encryptedMessage, iv);

}

}

private static void Send(byte[] key, string secretMessage, out byte[] encryptedMessage, out byte[] iv)

{

using (Aes aes = new AesCryptoServiceProvider())

{

aes.Key = key;

iv = aes.IV;

// Encrypt the message

using (MemoryStream ciphertext = new MemoryStream())

using (CryptoStream cs = new CryptoStream(ciphertext, aes.CreateEncryptor(), CryptoStreamMode.Write))

{

byte[] plaintextMessage = Encoding.UTF8.GetBytes(secretMessage);

cs.Write(plaintextMessage, 0, plaintextMessage.Length);

cs.Close();

encryptedMessage = ciphertext.ToArray();

}

}

}

}

public class Bob

{

public byte[] bobPublicKey;

private byte[] bobKey;

public Bob()

{

using (ECDiffieHellmanCng bob = new ECDiffieHellmanCng())

{

bob.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;

bob.HashAlgorithm = CngAlgorithm.Sha256;

bobPublicKey = bob.PublicKey.ToByteArray();

bobKey = bob.DeriveKeyMaterial(CngKey.Import(Alice.alicePublicKey, CngKeyBlobFormat.EccPublicBlob));

}

}

public void Receive(byte[] encryptedMessage, byte[] iv)

{

using (Aes aes = new AesCryptoServiceProvider())

{

aes.Key = bobKey;

aes.IV = iv;

// Decrypt the message

using (MemoryStream plaintext = new MemoryStream())

{

using (CryptoStream cs = new CryptoStream(plaintext, aes.CreateDecryptor(), CryptoStreamMode.Write))

{

cs.Write(encryptedMessage, 0, encryptedMessage.Length);

cs.Close();

string message = Encoding.UTF8.GetString(plaintext.ToArray());

Console.WriteLine(message);

}

}

}

}

}

}

1. Результаты работы (см Рисунок 1)



Рисунок 1 Результат работы

Вывод: изучил и реализовал на практике алгоритм обмена ключами Диффи-Хеллмана.