Лабораторная работа № 3

по информационной безопасности

студента группы ИТ-32

Манукова Давида Альбертовича

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

**Стандарт симметричного шифрования AES**

Цель работы: реализовать процессы шифрования и дешифрования, используя алгоритм симметричного шифрования AES-128.

Содержание работы

1. Изучить теоретический материал.

2. Разработать приложение, выполняющее шифрование и дешифрование файла произвольного размера на основе алгоритма AES-128. Подробно рассмотреть действие всех цикловых преобразований, как при шифровании, так и дешифровании.

3. Сохранить в отчете экранные формы, демонстрирующие процесс шифрования и дешифрования информации, проанализировать полученные результаты.

Ход работы

1. Разработал программу, реализующую процесс шифрования/дешифрования.

using System;

using System.Collections.Generic;

using System.IO;

using System.Security.Cryptography;

namespace IS\_Lab\_3

{

class Programm

{

public static void Main()

{

try

{

Console.WriteLine("Введите текст:");

string original = Console.ReadLine();

using (Rijndael myRijndael = Rijndael.Create())

{

byte[] encrypted = EncryptStringToBytes(original, myRijndael.Key, myRijndael.IV);

Console.WriteLine("\nЗашифрованный текст:\n");

for (int i = 0; i < encrypted.Length; i+=8)

{

for (int k = i; k < i+8; k++)

{

Console.Write(encrypted[k] + "\t");

}

Console.WriteLine();

if ((i / 8 % 16) == 0 && i!=0)

{

Console.WriteLine('\n');

}

}

string roundtrip = DecryptStringFromBytes(encrypted, myRijndael.Key, myRijndael.IV);

Console.WriteLine("\nДешифрованный текст: {0}", roundtrip);

}

}

catch (Exception e)

{

Console.WriteLine("Error: {0}", e.Message);

}

}

static byte[] EncryptStringToBytes(string plainText, byte[] Key, byte[] IV)

{

if (plainText == null || plainText.Length <= 0)

throw new ArgumentNullException("plainText");

if (Key == null || Key.Length <= 0)

throw new ArgumentNullException("Key");

if (IV == null || IV.Length <= 0)

throw new ArgumentNullException("IV");

byte[] encrypted;

using (Rijndael rijAlg = Rijndael.Create())

{

rijAlg.KeySize = 128;

rijAlg.BlockSize = 128;

rijAlg.Key = Key;

rijAlg.IV = IV;

ICryptoTransform encryptor = rijAlg.CreateEncryptor(rijAlg.Key, rijAlg.IV);

using (MemoryStream msEncrypt = new MemoryStream())

{

using (CryptoStream csEncrypt = new CryptoStream(msEncrypt, encryptor, CryptoStreamMode.Write))

{

using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))

{

swEncrypt.Write(plainText);

}

encrypted = msEncrypt.ToArray();

}

}

}

return encrypted;

}

static string DecryptStringFromBytes(byte[] cipherText, byte[] Key, byte[] IV)

{

if (cipherText == null || cipherText.Length <= 0)

throw new ArgumentNullException("cipherText");

if (Key == null || Key.Length <= 0)

throw new ArgumentNullException("Key");

if (IV == null || IV.Length <= 0)

throw new ArgumentNullException("IV");

string plaintext = null;

using (Rijndael rijAlg = Rijndael.Create())

{

rijAlg.KeySize = 128;

rijAlg.BlockSize = 128;

rijAlg.Key = Key;

rijAlg.IV = IV;

ICryptoTransform decryptor = rijAlg.CreateDecryptor(rijAlg.Key, rijAlg.IV);

using (MemoryStream msDecrypt = new MemoryStream(cipherText))

{

using (CryptoStream csDecrypt = new CryptoStream(msDecrypt, decryptor, CryptoStreamMode.Read))

{

using (StreamReader srDecrypt = new StreamReader(csDecrypt))

{

plaintext = srDecrypt.ReadToEnd();

}

}

}

}

return plaintext;

}

}

}

1. Результаты работы(см)

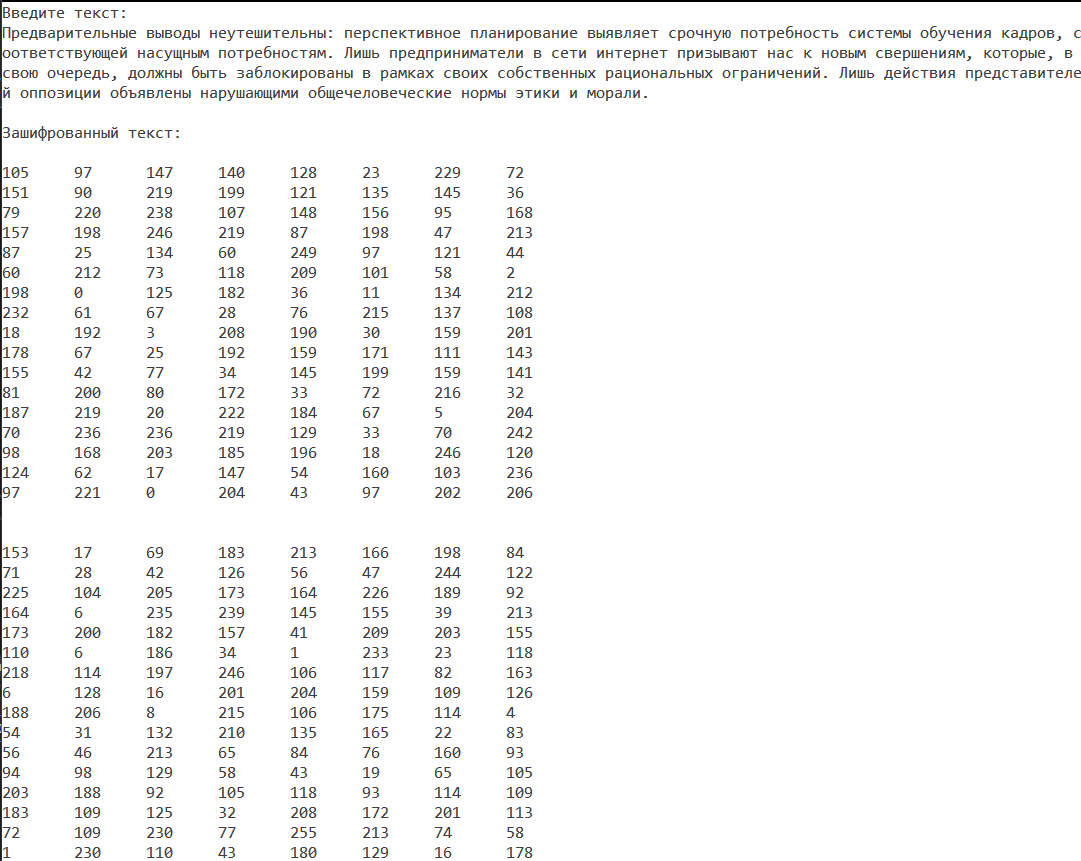


Рисунок 1 Результат работы

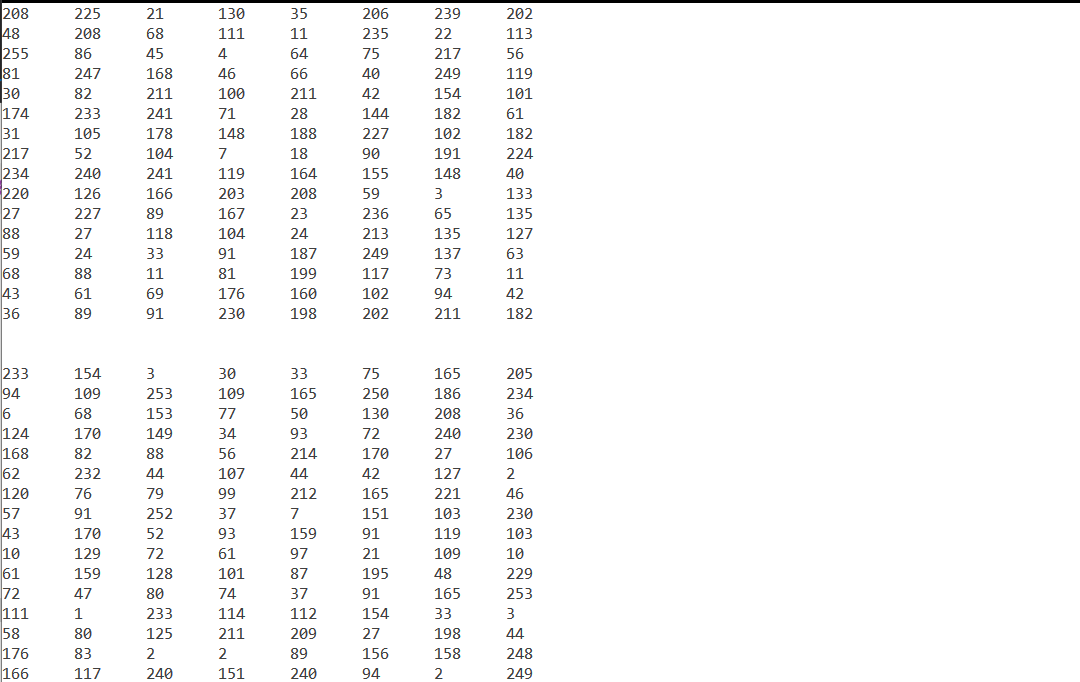


Рисунок 2 Результат работы

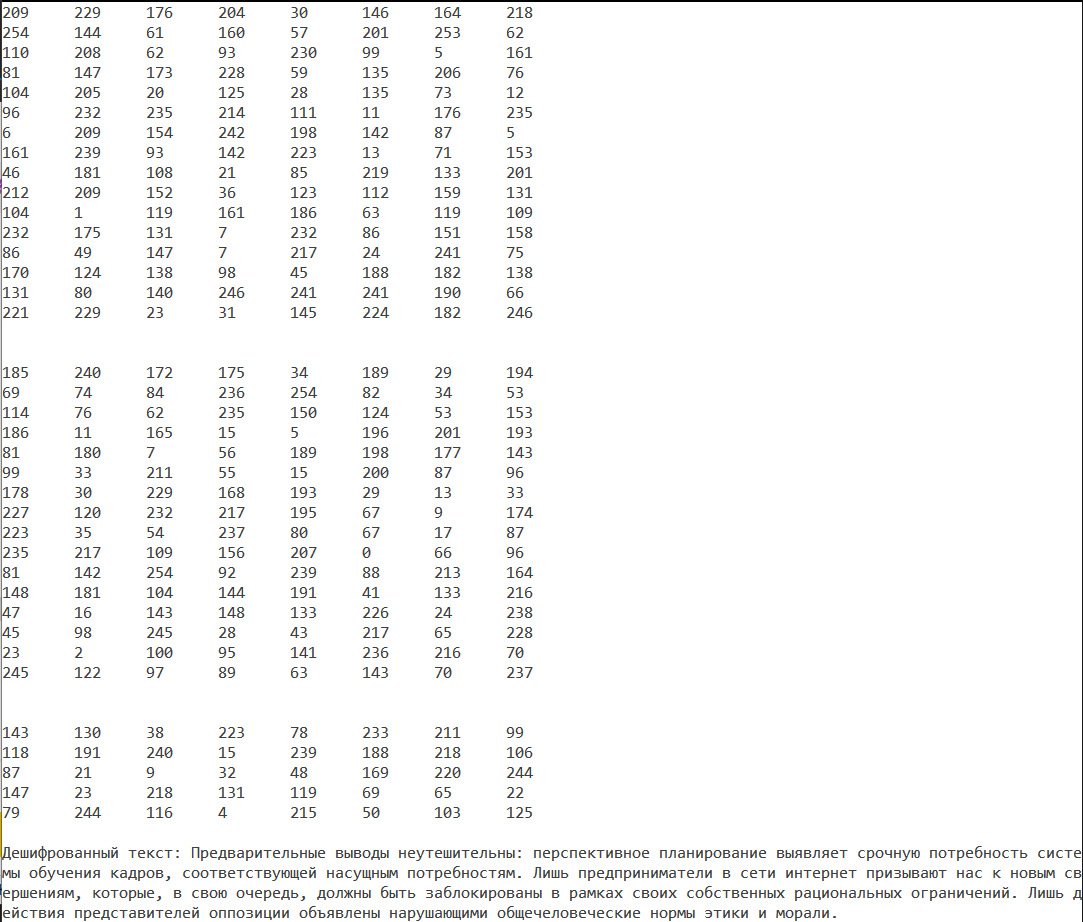


Рисунок 3 Результат работы

Вывод: в ходе выполнения лабораторной работы были изучены и получены практические навыки при работе со стандартом симметричного шифрования AES.