Шапка Денис

Лабораторна робота 2 Симетричне шифрування. Алгоритм AES

Дослідити принципи роботи симетричного шифрування на прикладі алгоритму AES.

Текст програми

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Security.Cryptography;
using System.IO;

namespace AES
{
    class Crypter
    {
        private System.Security.Cryptography.AesCryptoServiceProvider CryptKey;

        public Crypter(string key, string iv)
        {
            //создает объект шифрования который использует ключ (Key) и
инициализирует вектор (IV).
            CryptKey = new System.Security.Cryptography.AesCryptoServiceProvider();
            //Block size : задает размер блока в битах
            CryptKey.BlockSize = 128;
            //KeySize: задает размер ключа в битах
            CryptKey.KeySize = 128;
            //Key: задает симметричный ключ, который используется для шифрования и
дешифрования.
            CryptKey.Key =
System.Text.Encoding.UTF8.GetBytes(key).Take(16).ToArray();
            //IV : задает вектор инициализации (IV) симметричного алгоритма
            CryptKey.IV = System.Text.Encoding.UTF8.GetBytes(iv).Take(16).ToArray();
            //Padding: задает режим используемый в симметричном алгоритме
            CryptKey.Padding = System.Security.Cryptography.PaddingMode.PKCS7;
            //Mode: задает режим симметричного алгоритма
            CryptKey.Mode = System.Security.Cryptography.CipherMode.CBC;
        }

        //шифрование
        public byte[] Encrypt(byte[] dataToEncrypt)
        {
            //Создает симметричный AES объект
            var crypto = CryptKey.CreateEncryptor(CryptKey.Key, CryptKey.IV);
            //TransformFinalBlock – функция трансформации последнего блока или части
блока в поток.
            //Возвращает массив, который содержит преобразованные байты.
            byte[] encryptedData = crypto.TransformFinalBlock(dataToEncrypt, 0,
dataToEncrypt.Length);
            crypto.Dispose();
            return encryptedData;
        }

        //дешифрование
        public byte[] Decrypt(byte[] dataToDecrypt)
        {
            var crypto = CryptKey.CreateDecryptor(CryptKey.Key, CryptKey.IV);
```

```
            byte[] decryptedData = crypto.TransformFinalBlock(dataToDecrypt, 0,
dataToDecrypt.Length);
            crypto.Dispose();
            return decryptedData;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            var crypter = new Crypter("my key to encryption", "my vector of
crypting");
            var data = Encoding.ASCII.GetBytes("I've done AES-128 crypter!");
            var encrypted = crypter.Encrypt(data);
            var decrypted = crypter.Decrypt(encrypted);
            Console.WriteLine($"До шифрования     : {string.Join(" ", data)}");
            Console.WriteLine($"После шифрования  : {string.Join(" ", encrypted)}");
            Console.WriteLine($"После дешифрования: {string.Join(" ", decrypted)}");
            Console.ReadKey();
        }
    }
}
```