Звіт до ПЗ 6 Ремез

Завдання 1

using System;

using System.IO;

using System.Security.Cryptography;

using System.Text;

namespace Task6

{

class DesChipher

{

public static byte[] Encrypt(byte[] dataToEncrypt, byte[] key, byte[] iv)

{

using (var des = new DESCryptoServiceProvider())

{

des.Mode = CipherMode.CBC;

des.Padding = PaddingMode.PKCS7;

des.Key = key;

des.IV = iv;

using (var memoryStream = new MemoryStream())

{

var cryptoStream = new CryptoStream(memoryStream, des.CreateEncryptor(), CryptoStreamMode.Write);

cryptoStream.Write(dataToEncrypt, 0, dataToEncrypt.Length);

cryptoStream.FlushFinalBlock();

return memoryStream.ToArray();

}

}

}

public static byte[] Decrypt(byte[] dataToDecrypt, byte[] key, byte[] iv)

{

using (var des = new DESCryptoServiceProvider())

{

des.Mode = CipherMode.CBC;

des.Padding = PaddingMode.PKCS7;

des.Key = key;

des.IV = iv;

using (var memoryStream = new MemoryStream())

{

var cryptoStream = new CryptoStream(memoryStream, des.CreateDecryptor(), CryptoStreamMode.Write);

cryptoStream.Write(dataToDecrypt, 0, dataToDecrypt.Length);

cryptoStream.FlushFinalBlock();

return memoryStream.ToArray();

}

}

}

}

class TripledesChipher

{

public static byte[] Encrypt(byte[] dataToEncrypt, byte[] key, byte[] iv)

{

using (var des = new TripleDESCryptoServiceProvider())

{

des.Mode = CipherMode.CBC;

des.Padding = PaddingMode.PKCS7;

des.Key = key;

des.IV = iv;

using (var memoryStream = new MemoryStream())

{

var cryptoStream = new CryptoStream(memoryStream, des.CreateEncryptor(), CryptoStreamMode.Write);

cryptoStream.Write(dataToEncrypt, 0, dataToEncrypt.Length);

cryptoStream.FlushFinalBlock();

return memoryStream.ToArray();

}

}

}

public static byte[] Decrypt(byte[] dataToDecrypt, byte[] key, byte[] iv)

{

using (var des = new TripleDESCryptoServiceProvider())

{

des.Mode = CipherMode.CBC;

des.Padding = PaddingMode.PKCS7;

des.Key = key;

des.IV = iv;

using (var memoryStream = new MemoryStream())

{

var cryptoStream = new CryptoStream(memoryStream, des.CreateDecryptor(), CryptoStreamMode.Write);

cryptoStream.Write(dataToDecrypt, 0, dataToDecrypt.Length);

cryptoStream.FlushFinalBlock();

return memoryStream.ToArray();

}

}

}

}

class AesChipher

{

public static byte[] GenerateRandomNumber(int length)

{

using (var randomNumberGenerator = new RNGCryptoServiceProvider())

{

var randomNumber = new byte[length];

randomNumberGenerator.GetBytes(randomNumber);

return randomNumber;

}

}

public static byte[] Encrypt(byte[] dataToEncrypt, byte[] key, byte[] iv)

{

using (var aes = new AesCryptoServiceProvider())

{

aes.Mode = CipherMode.CBC;

aes.Padding = PaddingMode.PKCS7;

aes.Key = key;

aes.IV = iv;

using (var memoryStream = new MemoryStream())

{

var cryptoStream = new CryptoStream(memoryStream, aes.CreateEncryptor(), CryptoStreamMode.Write);

cryptoStream.Write(dataToEncrypt, 0, dataToEncrypt.Length);

cryptoStream.FlushFinalBlock();

return memoryStream.ToArray();

}

}

}

public static byte[] Decrypt(byte[] dataToDecrypt, byte[] key, byte[] iv)

{

using (var aes = new AesCryptoServiceProvider())

{

aes.Mode = CipherMode.CBC;

aes.Padding = PaddingMode.PKCS7;

aes.Key = key;

aes.IV = iv;

using (var memoryStream = new MemoryStream())

{

var cryptoStream = new CryptoStream(memoryStream, aes.CreateDecryptor(), CryptoStreamMode.Write);

cryptoStream.Write(dataToDecrypt, 0, dataToDecrypt.Length);

cryptoStream.FlushFinalBlock();

return memoryStream.ToArray();

}

}

}

}

class Program

{

static void Main(string[] args)

{

var key = AesChipher.GenerateRandomNumber(8);

var iv = AesChipher.GenerateRandomNumber(8);

const string originalDes = "Text to encrypt ";

var encrypted = DesChipher.Encrypt(Encoding.UTF8.GetBytes(originalDes), key, iv);

var decrypted = DesChipher.Decrypt(encrypted, key, iv);

var decryptedMessage = Encoding.UTF8.GetString(decrypted);

Console.WriteLine("----------------------");

Console.WriteLine("TripleDes Encryption in .NET");

Console.WriteLine();

Console.WriteLine("Original Text = " + originalDes);

Console.WriteLine("Encrypted Text = " +

Convert.ToBase64String(encrypted));

Console.WriteLine("Decrypted Text = " + decryptedMessage);

var key2 = AesChipher.GenerateRandomNumber(16);

var iv2 = AesChipher.GenerateRandomNumber(8);

//main message

const string originalTripleDes = "For encryption";

var encrypted2 = TripledesChipher.Encrypt(Encoding.UTF8.GetBytes(originalTripleDes), key2, iv2);

var decrypted2 = TripledesChipher.Decrypt(encrypted2, key2, iv2);

var decryptedMessage2 = Encoding.UTF8.GetString(decrypted2);

Console.WriteLine("----------------------");

Console.WriteLine("TripleDes Encryption in .NET");

Console.WriteLine();

Console.WriteLine("Main Text = " + originalTripleDes);

Console.WriteLine("Encrypted Text = " +

Convert.ToBase64String(encrypted2));

Console.WriteLine("Decrypted Text = " + decryptedMessage2);

var key3 = AesChipher.GenerateRandomNumber(32);

var iv3 = AesChipher.GenerateRandomNumber(16);

const string originalAES = "Text to encrypt ";

var encrypted3 = AesChipher.Encrypt(Encoding.UTF8.GetBytes(originalAES), key3, iv3);

var decrypted3 = AesChipher.Decrypt(encrypted3, key3, iv3);

var decryptedMessage3 = Encoding.UTF8.GetString(decrypted3);

Console.WriteLine("----------------------");

Console.WriteLine("AES Encryption in .NET");

Console.WriteLine();

Console.WriteLine("Original Text = " + originalAES);

Console.WriteLine("Encrypted Text = " +

Convert.ToBase64String(encrypted3));

Console.WriteLine("Decrypted Text = " + decryptedMessage3);

}

}

}

