МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Чувашский государственный университет имени И.Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Лабораторная работа №7

По дисциплине “Защита информации” на тему:

«Работа алгоритма DES в режиме OFB»

Выполнил: ст. гр. ИВТ-11-21

Еремеев С.А.

Проверил: канд. техн. наук. Ковалев С.В.

Чебоксары 2025

**Теоретическая часть**

# Цель работы

Целью работы является знакомство с особенностями работы алгоритма DES в режиме OFB.

# Задание на работу

В данной лабораторной работе необходимо программно реализовать алгоритмы шифрования и дешифрования DES в режиме OFB. 64-битовый начальный вектор C*0* следует сгенерировать, используя датчик ПСЧ, разработанный в лабораторной работе №3.

# Основные сведения

Наряду с режимами ECB и CBC, алгоритм DES используется еще в двух режимах с так называемой обратной связью.

Обобщенная структурная схема использования алгоритма DES в режиме обратной связи по выходу (OFB, Output FeedBack) с *k*-битовыми блоками приведена на рис. 7.

В отличие от режимов ECB и CBC, оперирующих 64-битовыми блоками, режимы CFB и OFB оперируют *k*-битовыми блоками (1<*k*<63). Это, в частности, позволяет шифровать текстовые данные посимвольно (для кода EBCDIC достаточно положить *k*=8, для кода ASCII - *k*=7).

Шифрование в режиме OFB происходит во многом аналогично шифрованию в режиме CFB. Отличие режима OFB от режима CFB заключается в том, что *k*-битовый вектор обратной связи поступает в *k* правых битов входного блока не после гаммирования, а до него.

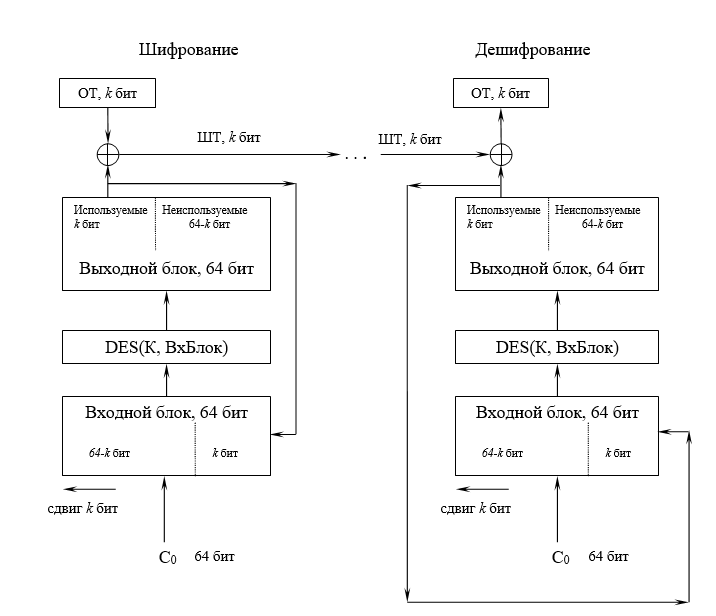


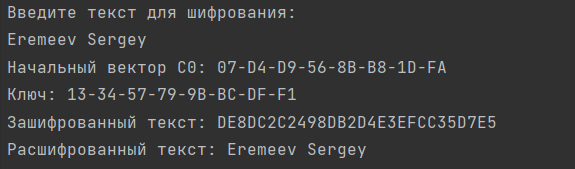
Рисунок 1 – Шифрование DES в режиме OFB

**Практическая часть**

# Текст программы

using System**;**using System**.**Security**.**Cryptography**;**using System**.**Text**;**class Program  
**{** *// Значения для генерации случайных чисел* static int A = 25**,** C = 37**,** T0 = 7**,** B = 256**;** static void Main**(**string**[]** args**)  
 {** Console**.**WriteLine**(**"Введите текст для шифрования:"**);** string text = Console**.**ReadLine**();** *// Генерируем начальный вектор C0* byte**[]** C0 = GenerateC0**(**8**);** Console**.**WriteLine**(**"Начальный вектор C0 (в HEX): " + BitConverter**.**ToString**(**C0**));** *// Ключ для DES* byte**[]** key = new byte**[**8**] {** 0x13**,** 0x34**,** 0x57**,** 0x79**,** 0x9B**,** 0xBC**,** 0xDF**,** 0xF1 **};** Console**.**WriteLine**(**"Ключ (в HEX): " + BitConverter**.**ToString**(**key**));** int k = 8**;** *// Шифруем текст* string encryptedText = EncryptOFB**(**text**,** key**,** C0**,** k**);** Console**.**WriteLine**(**"Зашифрованный текст: " + encryptedText**);** *// Дешифруем текст* string decryptedText = DecryptOFB**(**encryptedText**,** key**,** C0**,** k**);** Console**.**WriteLine**(**"Расшифрованный текст: " + decryptedText**);  
 }** *// Метод для генерации начального вектора C0 с помощью ПСЧ из Лаб. №3* static byte**[]** GenerateC0**(**int length**)  
 {** byte**[]** C0 = new byte**[**length**];** int**[]** gamma = new int**[**length**];** gamma**[**0**]** = T0**;** for **(**int i = 1**;** i < length**;** i++**)  
 {** gamma**[**i**]** = **(**A \* gamma**[**i - 1**]** + C**)** % B**;  
 }** for **(**int i = 0**;** i < length**;** i++**)  
 {** C0**[**i**]** = **(**byte**)**gamma**[**i**];  
 }** return C0**;  
 }** *// Метод для шифрования в режиме OFB* static string EncryptOFB**(**string text**,** byte**[]** key**,** byte**[]** C0**,** int k**)  
 {** byte**[]** inputBlock = **(**byte**[])**C0**.**Clone**();** byte**[]** textBytes = Encoding**.**ASCII**.**GetBytes**(**text**);** byte**[]** encryptedBytes = new byte**[**textBytes**.**Length**];** int kBytes = k / 8**;** *// Создаём объект DES для шифрования* using **(**DES des = DES**.**Create**())  
 {** des**.**Key = key**;** des**.**Mode = CipherMode**.ECB;** des**.**Padding = PaddingMode**.None;** ICryptoTransform encryptor = des**.**CreateEncryptor**();** for **(**int i = 0**;** i < textBytes**.**Length**;** i += kBytes**)  
 {** byte**[]** outputBlock = new byte**[**8**];** encryptor**.**TransformBlock**(**inputBlock**,** 0**,** 8**,** outputBlock**,** 0**);** byte keystream = outputBlock**[**0**];** for **(**int j = 0**;** j < kBytes && **(**i + j**)** < textBytes**.**Length**;** j++**)  
 {** encryptedBytes**[**i + j**]** = **(**byte**)(**textBytes**[**i + j**]** ^ keystream**);  
 }** inputBlock = **(**byte**[])**outputBlock**.**Clone**();  
 }  
 }** return BitConverter**.**ToString**(**encryptedBytes**).**Replace**(**"-"**,** ""**);  
 }** *// Метод для дешифрования в режиме OFB* static string DecryptOFB**(**string encryptedText**,** byte**[]** key**,** byte**[]** C0**,** int k**)  
 {** byte**[]** encryptedBytes = new byte**[**encryptedText**.**Length / 2**];** for **(**int i = 0**;** i < encryptedBytes**.**Length**;** i++**)  
 {** encryptedBytes**[**i**]** = Convert**.**ToByte**(**encryptedText**.**Substring**(**i \* 2**,** 2**),** 16**);  
 }** byte**[]** inputBlock = **(**byte**[])**C0**.**Clone**();** byte**[]** decryptedBytes = new byte**[**encryptedBytes**.**Length**];** int kBytes = k / 8**;** using **(**DES des = DES**.**Create**())  
 {** des**.**Key = key**;** des**.**Mode = CipherMode**.ECB;** des**.**Padding = PaddingMode**.None;** ICryptoTransform encryptor = des**.**CreateEncryptor**();** for **(**int i = 0**;** i < encryptedBytes**.**Length**;** i += kBytes**)  
 {** byte**[]** outputBlock = new byte**[**8**];** encryptor**.**TransformBlock**(**inputBlock**,** 0**,** 8**,** outputBlock**,** 0**);** byte keystream = outputBlock**[**0**];** *// Первый байт — это наш keystream* for **(**int j = 0**;** j < kBytes && **(**i + j**)** < encryptedBytes**.**Length**;** j++**)  
 {** decryptedBytes**[**i + j**]** = **(**byte**)(**encryptedBytes**[**i + j**]** ^ keystream**);  
 }** inputBlock = **(**byte**[])**outputBlock**.**Clone**();  
 }  
 }** return Encoding**.**ASCII**.**GetString**(**decryptedBytes**);  
 }  
}**

# Пример открытого текста и соответствующей ему шифрограммы

Рисунок 2 – Пример 1

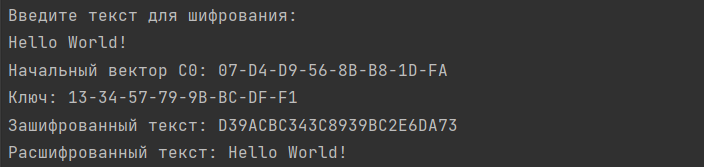
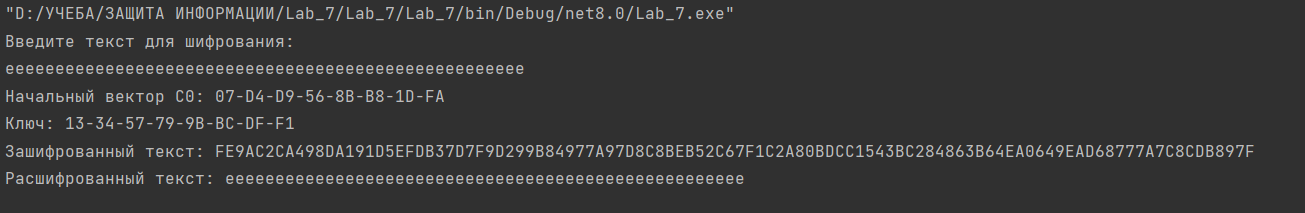


Рисунок 3 – Пример 2

 Рисунок 4 – Пример 3

# Вывод

В ходе данной лабораторной работы был рассмотрен и реализован на языке высокого программирования C# классический криптографический алгоритм DES в режиме OFB.