Федеральное государственное бюджетное образовательное учреждение высшего образования Национальный исследовательский университет "МЭИ"  
Кафедра Автоматики и вычислительной техники.

Курсовой проект по дисциплине «Защита данных»

Тема: Программная реализация криптоалгоритма 3-DES.

Студент Жилманов В.Ю.

Группа А-13-14

Преподаватель Хорев П.Б.

Москва 2017

Оглавление

[Введение 4](#_Toc501411154)

[Глава 1 Описание алгоритма 3-DES 4](#_Toc501411155)

[1.1 Входные данные 5](#_Toc501411156)

[1.2 Схема шифрования алгоритма DES 7](#_Toc501411157)

[1.2.1 Начальная перестановка 7](#_Toc501411158)

[1.2.2 Циклы шифрования 7](#_Toc501411159)

[1.2.3 Основная функция шифрования (функция Файестеля) 8](#_Toc501411160)

[1.2.4 Конечная перестановка 8](#_Toc501411161)

[1.2.5 Генерация ключей ki 9](#_Toc501411162)

[1.3 Схема расшифрования DES 11](#_Toc501411163)

[1.4 Режимы использования 3-DES 12](#_Toc501411164)

[Режим электронной кодовой книги (ECB - Electronic Code Book) 12](#_Toc501411165)

[Режим сцепления блоков (СВС - Cipher Block Chaining) 13](#_Toc501411166)

[Режим обратной связи по выходу (OFB - Output Feed Back) 15](#_Toc501411167)

[Глава 2. Результаты проектирования программы. 16](#_Toc501411168)

[2.1. Результаты проектирования графического интерфейса пользователя. 16](#_Toc501411169)

[2.1.1. Режимы работы программы. 16](#_Toc501411170)

[Глава 3. Результаты тестирования разработанной программы. 21](#_Toc501411171)

[Заключение. 27](#_Toc501411172)

[Список источников 27](#_Toc501411173)

# Введение

Цель работы – разработка приложения, позволяющего шифровать и расшифровывать по алгоритму 3-DES как выбранные произвольные файлы, так и вводимые текстовые сообщения с возможностью сохранения в файле результата, полученного шифротекста. Должна быть возможность шифрования, сохранения и рас шифрования текста, с помощью ключа длинны 168 бит, выводимого из парольной фразы с регулируемой минимальной длинной и сложностью.

В процессе выполнения работы были поставлены следующие задачи:

* Проектирование графического интерфейса программы
* Изучение и реализация алгоритма шифрования
* Реализация графического интерфейса и интеграция в него алгоритма шифрования
* Отладка графического интерфейса
* Отладка алгоритма шифрования

При выполнении данной работы все указанные задачи были решены.

# Глава 1 Описание алгоритма 3-DES

Алгоритм шифрования 3-DES (Triple data encryption standard) предназначен для шифрования сообщений произвольной длинны. 3-DES – симметричный блочный шифр, созданный Уитфилдом Диффи, Мартином Хеллманом и Уолтом Тачманном в 1978 году на основе алгоритма DES с целью устранения главного недостатка последнего — малой длины ключа (56 бит), который может быть взломан методом полного перебора ключа.

В процессе шифрование алгоритм шифрует исходный текст трижды применяя к нему алгоритм шифрование DES. Порядок применения может быть различным. Простой вариант 3DES можно представить так:

**DES(k3; DES(k2; DES(k1; M)))**, где **k1**, **k2**, **k3** – ключи для каждого DES-шага, **M** – входные данные, которые нужно шифровать.

Существует 3 типа алгоритма 3-DES:

* DES-EEE3: Шифруется три раза с тремя разными ключами (операции шифрование-шифрование-шифрование).
* DES-EDE3: 3DES операции шифровка-расшифровка-шифровка с тремя разными ключами.
* DES-EEE2 и DES-EDE2: Как и предыдущие, за исключением того, что на первом и третьем шаге используется одинаковый ключ.

В работе мною реализован разновидность 3- DES-EDE3.

## 1.1 Входные данные

Таблица 1.1. Входные данные

|  |  |
| --- | --- |
| T | исходное сообщение |
| k1, k2, k3 | ключи шифрования |
| IP | таблица для начальной перестановка |
| E | таблица для функции расширения |
| Si, i = 1,..,8 | таблицы для преобразования |
| P | таблица для перестановки в основной функции шифрования |
| IP-1 | Конечная перестановка |

Таблица 1.2 Значения параметра IP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 | 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Таблица 1.3 Значения параметра E

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |

Таблица 1.4 Значения параметра P

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

Таблица 1.5 Значения параметра Si

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***S1*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |
| ***S2*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |
| ***S3*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |
| ***S4*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |
| ***S5*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 15 | 13 | 0 | 14 | 9 |
| 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |
| ***S6*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |
| ***S7*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| 13 | 0 | 11 | 7 | 4 | 9 | 1 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |
| ***S8*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |
| 2 | 1 | 14 | 7 | 4 | 10 | 8 | 13 | 15 | 12 | 9 | 0 | 3 | 5 | 6 | 11 |

Таблица 1.6 Значение параметра IP-1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 | 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 | 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 | 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 | 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |

## 1.2 Схема шифрования алгоритма DES

Исходный текст для алгоритма блок **T** размера 64 бита. Процесс шифрования состоит из начальной перестановки, 16 циклов шифрования и конечной перестановки.

### 1.2.1 Начальная перестановка

Исходный текст T преобразуется с помощью начальной перестановки IP, которая определяется таблицей 1.2. Полученный после начальной перестановки 64-битный блок **IP(T)** участвует в 16 циклах преобразования Фейстеля.

Разбить IP(T) на две части **L0,** **R0**, где **L0**, **R0** - соответственно 32 старших битов и 32 младших битов блока **IP(T).**

### 1.2.2 Циклы шифрования

Пусть **Ti - 1**= **Li – 1 Ri – 1** результат (i-1) итерации, тогда результат i-ой итерации **Ti** = **Li Ri** определяется:

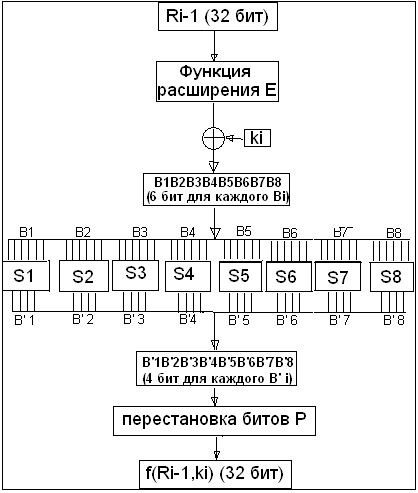


В 16-циклах преобразования Фейстеля функция f играет роль шифрования. Рассмотрим подробно функцию f.

### 1.2.3 Основная функция шифрования (функция Файестеля)

Аргументами функции **f** являются 32-битовый вектор **Ri** и 48-битовый ключ **ki**, который является результатом преобразования 56-битового исходного ключа шифра **k**.

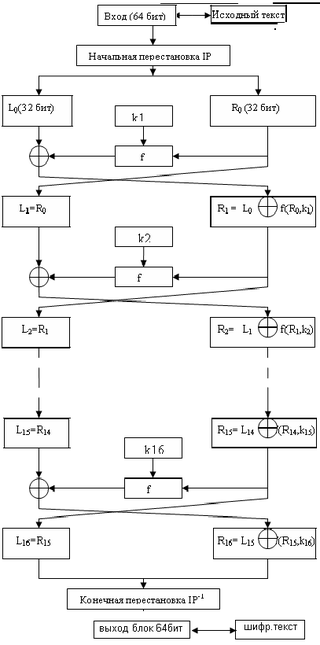
Для вычисления функции **f** последовательно используются функция расширения **E**, сложение по модулю 2 с ключом **ki,** преобразование **S**, состоящее из 8 преобразований **S**-блоков Si, = 1,..8, перестановка **P**.



(рисунок 1.1 Схема работы функции f)

### 1.2.4 Конечная перестановка

Конечная перестановка **IP-1** действует на **T16** и является обратной к первоначальной перестановке. Конечная перестановка определяется таблицей 1.6.



(Рисунок 1.2 Схема шифрование DES)

### 1.2.5 Генерация ключей ki

Ключи **ki** получаются из начального ключа **k** (56 байт) следующим образом. Добавляются биты в позиции 8, 16, 24, 32, 40, 48, 56, 64 ключа **k** таким образом, чтобы каждый байт содержал нечетное число единиц. Это используется для обнаружения ошибок при обмене и хранении ключей. Затем делают перестановку для расширенного ключа (кроме добавляемых битов 8, 16, 24, 32, 40, 48, 56, 64). Такая перестановка определена в таблице 1.7.

Таблица 1.7 Перестановка бит начального ключа

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Эта перестановка определяется двумя блоками **C0 D0** по 28 бит каждый. **Ci Di** i=1,2,3…получаются из **Ci -1 Di -1** одним или двумя левыми циклическими сдвигами согласно таблице 1.8.

Таблица 1.8 Циклические сдвиги ключей

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| Число сдвига | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

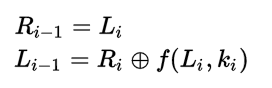
Ключ **ki**, i=1,…16 состоит из 48 бит, выбранных из битов вектора **Ci Di** согласно таблице 1.9.

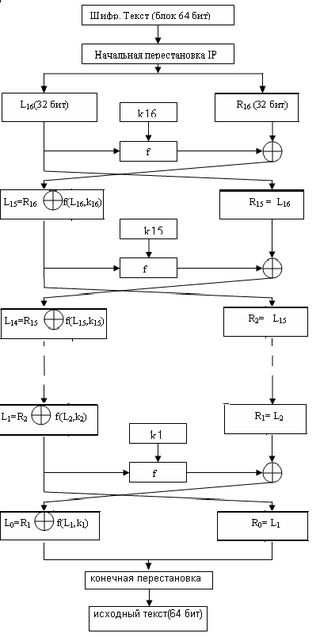
Таблица 1.9 Выбор бит для ключа ki

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | 17 | 11 | 24 | 1 | 5 | 3 | 28 | 15 | 6 | 21 | 10 | 23 | 19 | 12 | 4 |
| 26 | 8 | 16 | 7 | 27 | 20 | 13 | 2 | 41 | 52 | 31 | 37 | 47 | 55 | 30 | 40 |
| 51 | 45 | 33 | 48 | 44 | 49 | 39 | 56 | 34 | 53 | 46 | 42 | 50 | 36 | 29 | 32 |

## 1.3 Схема расшифрования DES

При расшифровании данных все действия выполняются в обратном порядке. В 16 циклах расшифрования, в отличие от шифрования c помощью прямого преобразования сетью Фейстеля, здесь используется обратное преобразование сетью Фейстеля.





(Рисунок 1.2 Схема расшифрования алгоритма DES)

## 1.4 Режимы использования 3-DES

Режим шифрования — метод применения блочного шифра (алгоритма), позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных. При этом для шифрования одного блока могут использоваться данные другого блока.

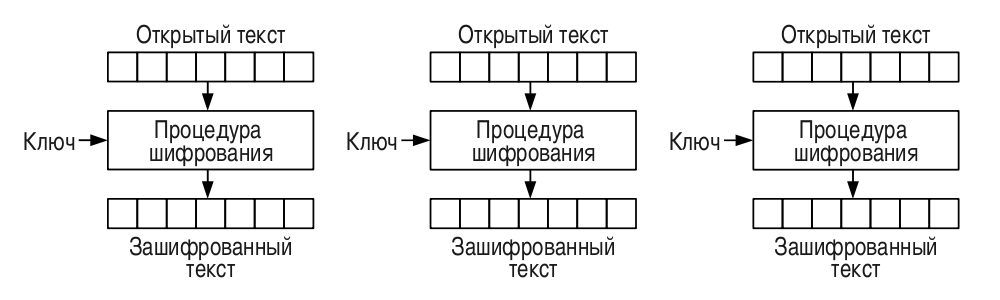
Обычно режимы шифрования используются для изменения процесса шифрования так, чтобы результат шифрования каждого блока был уникальным вне зависимости от шифруемых данных и не позволял сделать какие-либо выводы об их структуре. Это обусловлено, прежде всего, тем, что блочные шифры шифруют данные блоками фиксированного размера, и поэтому существует потенциальная возможность утечки информации о повторяющихся частях данных, шифруемых на одном и том же ключе.

Алгоритм 3-DES поддерживает основные режимы шифрования. В ходе работы были реализованы следующие режимы:

* Режим электронной кодовой книги (ECB - Electronic Code Book)
* Режим сцепления блоков (СВС - Cipher Block Chaining)
* Режим обратной связи по шифротексту (CFB - Cipher Feed Back)
* Режим обратной связи по выходу (OFB - Output Feed Back)

### Режим электронной кодовой книги (ECB - Electronic Code Book)

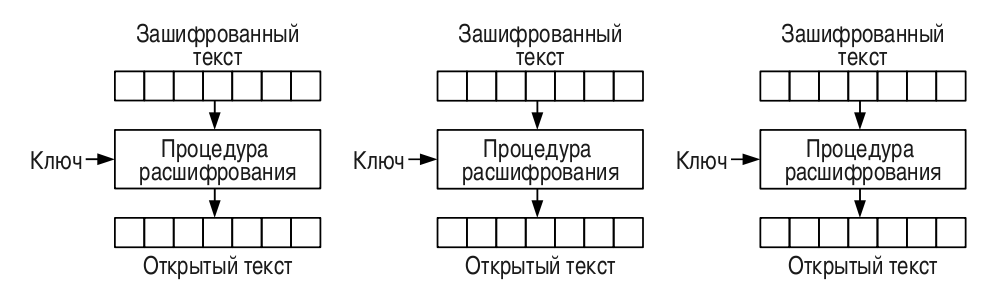
В ГОСТ 28147—89 этот режим называется режимом простой замены.



(Рисунок 1.3 Шифрование в режиме электронной кодовой книги)

Сообщение делится на блоки одинакового размера. В результате получается последовательность блоков **Pi** длинны n. Последний блок при необходимости дополняется до длины n.

Каждый блок **Pi** шифруется алгоритмом шифрования с использованием ключей **k1, k2, k3**.



(Рисунок 1.3 Расшифрование в режиме электронной кодовой книги)

### Режим сцепления блоков (СВС - Cipher Block Chaining)

Каждый блок открытого текста (кроме первого) побитово складывается по модулю 2 (операция XOR) с предыдущим результатом шифрования.

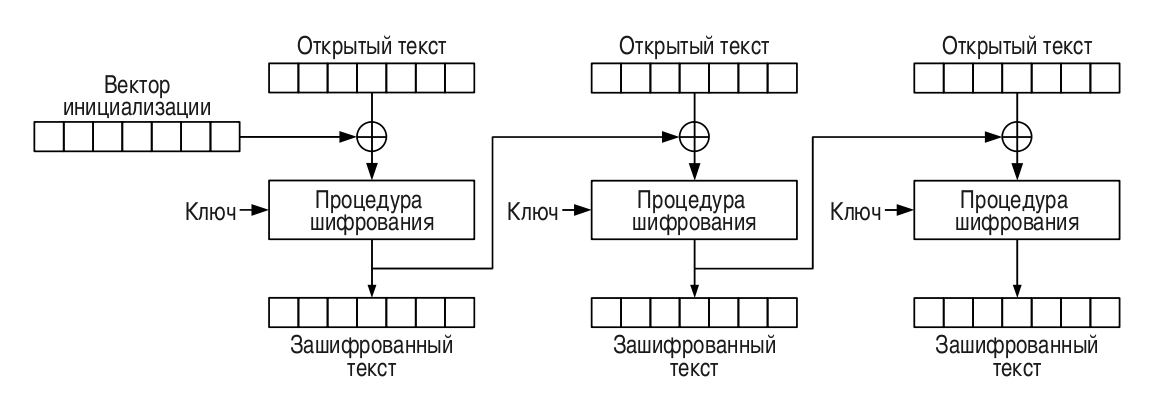
Шифрование может быть описано следующим образом:



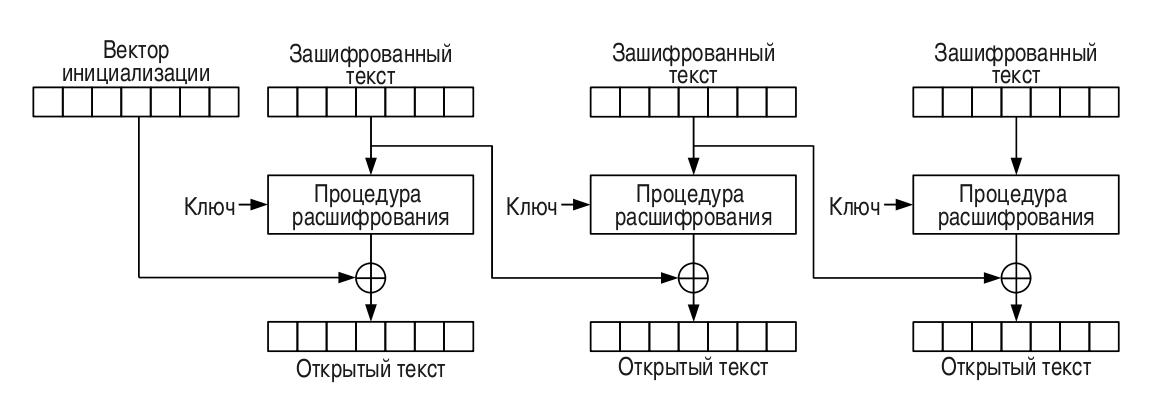
где i — номера блоков, IV — вектор инициализации (синхропосылка), **Ci**и **Pi** - блоки зашифрованного и открытого текстов соответственно, а **Ek** функция блочного шифрования.

Расшифрование:





(Рисунок 1.4 Шифрование в режиме сцепления блоков)



(Рисунок 1.5 Расшифрование в режиме сцепления блоков)

Режим обратной связи по шифротексту (CFB - Cipher Feed Back)

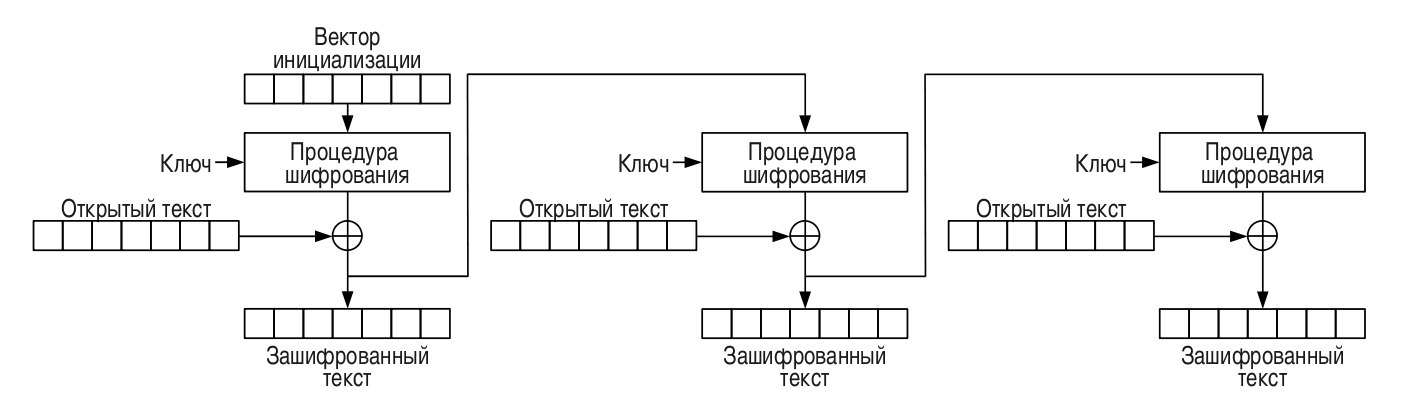
Для шифрования следующего блока открытого текста он складывается по модулю 2 с результатом шифрования предыдущего блока. Шифрование может быть описано следующим образом:



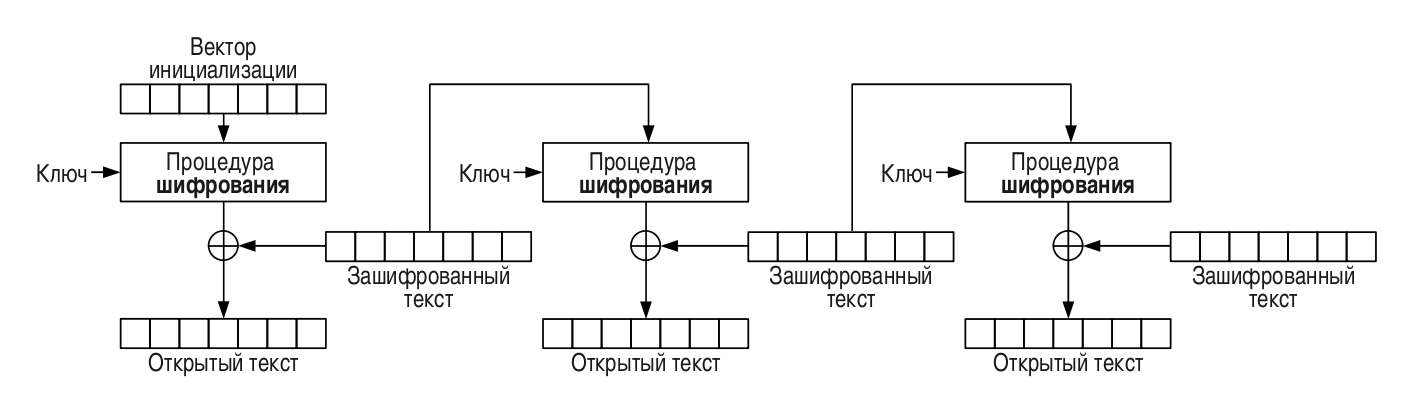
Расшифрование:



где **i** - номера блоков, **IV** - вектор инициализации (синхропосылка), **Ci**и **Pi** - блоки зашифрованного и открытого текстов соответственно, а **Ek** — функция блочного шифрования.



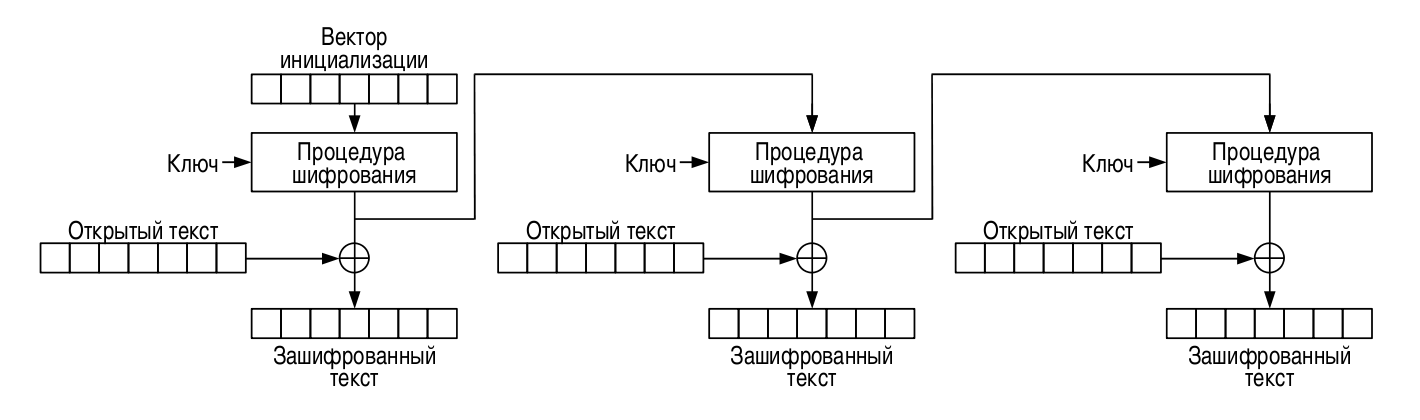
(Рисунок 1.6 Шифрование в режиме обратной связи по шифротексту)



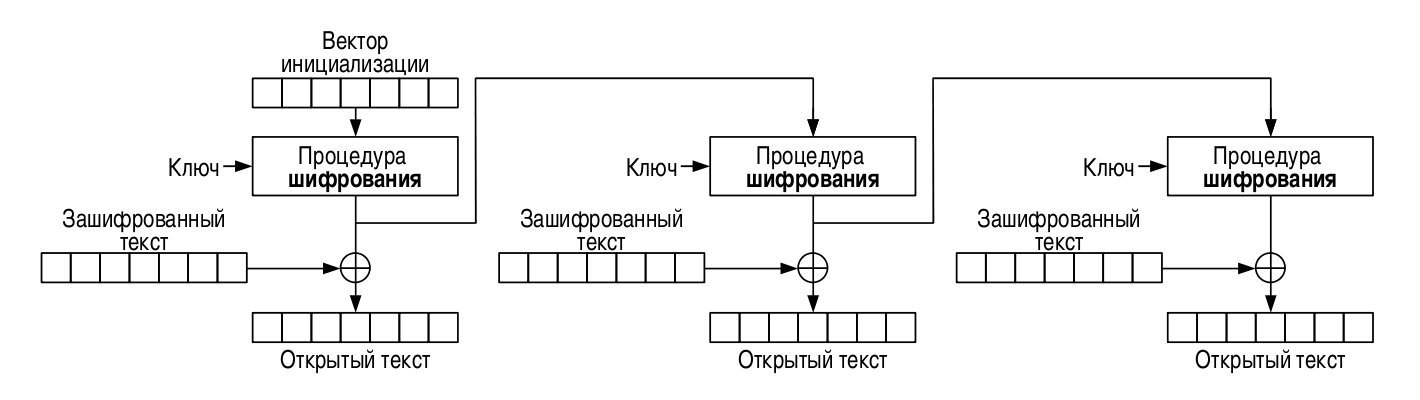
(Рисунок 1.7 Расшифрование в режиме обратной связи по шифротексту)

### Режим обратной связи по выходу (OFB - Output Feed Back)

Особенностью режима является то, что в качестве входных данных для алгоритма блочного шифрования не используется само сообщение. Вместо этого блочный шифр используется для генерации псевдослучайного потока байтов, который с помощью операции XOR складывается с блоками открытого текста. Подобная схема шифрования называется потоковым шифром (англ. stream cipher).



(Рисунок 1.8 Схема шифрования в режиме обратной связи по выходу)

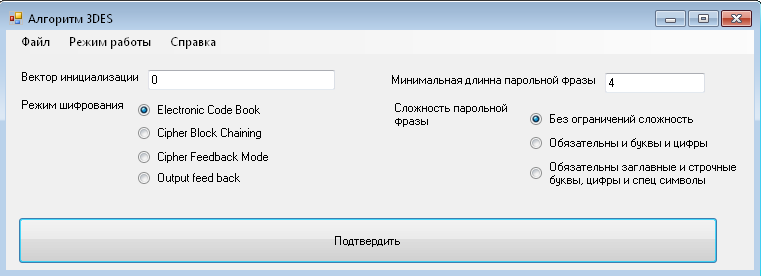


(Рисунок 1.9 Схема дешифрования в режиме обратной связи по выходу)

# Глава 2. Результаты проектирования программы.

## 2.1. Результаты проектирования графического интерфейса пользователя.

Главное окно представляет из себя окно выбора параметров для шифрования. В нём происходит выбор режима шифрования и задание вектора инициализации. Так же в главном окне пользователь задаёт минимальную длину парольной фразы и выбирает её сложность. Для сохранения выбранных параметров необходимо нажать кнопку «Подтвердить».

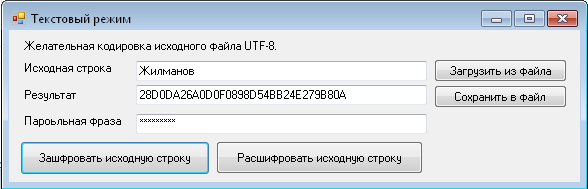


(Рисунок 2.1 Главное окно приложения)

### 2.1.1. Режимы работы программы.

Программа поддерживает следующие режимы работы:

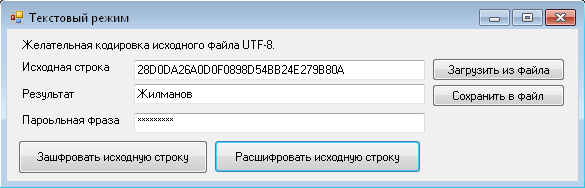
1) Ручной ввод сообщения для шифрования. В этом режиме исходный текст шифруется на ключах, полученных из парольной фразы, которая так же указывается на форме. После нажатия кнопки «Зашифровать исходную строку» полученный шифротекст выводится в виде строки в виде числа в 16ричной системе счисления.



(Рисунок 2.2 Работа с текстовым режимом)

Так же строку для шифрования можно загрузить из файла (если в файле несколько строк необходимо использовать другой режим работы). Полученный шифротекст можно так же сохранить в файл.

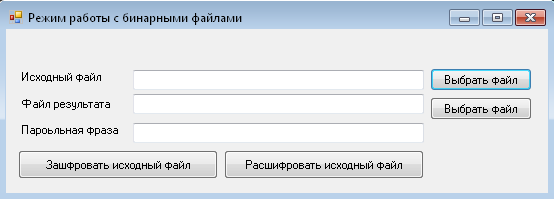
Что бы расшифровать текст необходимо ввести его в поле «Исходная строка» и нажать кнопку «Расшифровать исходную строку».



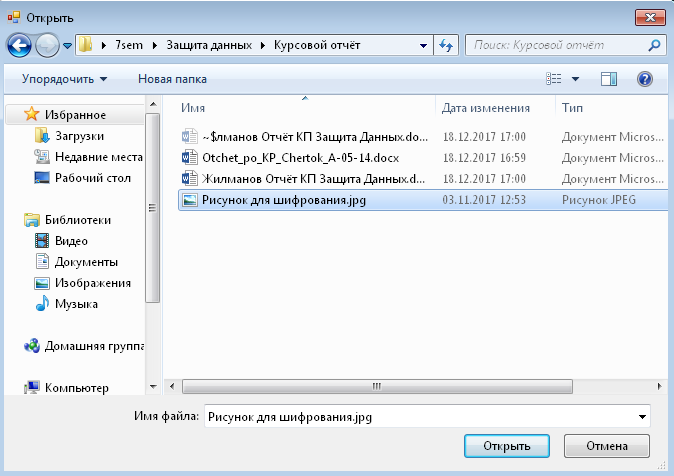
(Рисунок 2.3 Работа с текстовым режимом)

2) Режим работы с бинарными файлами.

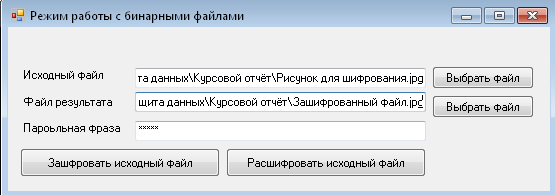
В этом режиме можно шифровать любые файлы. В строке исходный файл необходимо указать файл, который необходимо зашифровать. Выбрать его можно с помощью кнопки «Выбрать файл», после нажатия кнопки откроется диалог «Открыть» выбора файла. Затем необходимо выбрать файл, в который будет записан результат шифрования. Сделать это можно с помощью кнопки «Выбрать файла», расположенной напротив поля результат. После нажатия кнопки появится диалог «Сохранить» выбора файла. Если файл не создан, то он будет создан, если файл уже существует, то он будет перезаписан.



(Рисунок 2.4 Работа в бинарном режиме)



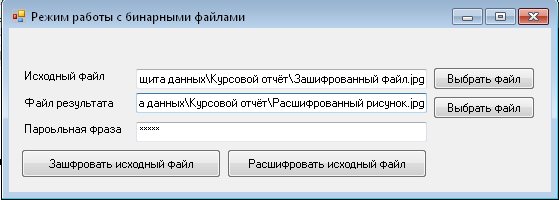
(Рисунок 2.5 Выбор файла для шифрования)



(Рисунок 2.6 Выбор файла для шифрования в бинарном режиме)

При нажатии кнопки «Зашифровать исходный файл» и подтверждения парольной фразы исходный файл будет зашифрован и сохранён в файл результата.

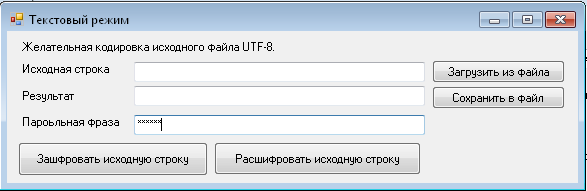
Что бы расшифровать файл необходимо в качестве исходного файла указать зашифрованный файл, а в качестве файла результата файл, в который необходимо записать результат.



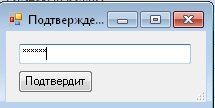
(Рисунок 2.6 Выбор файла для расшифрования в бинарном режиме)

#### 2.1.2. Работа с парольной фразой

Что бы зашифровать файл в любом из режимов необходимо в поле «Парольная фраза» ввести парольную фразу, которая будет удовлетворять требованиям длинны и сложности, выбранным на главной форме, а затем подтвердить её в открывшемся окне.

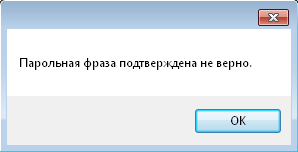


(Рисунок 2.7 Ввод парольной фразы)



(Рисунок 2.7 Подтверждение парольной фразы)

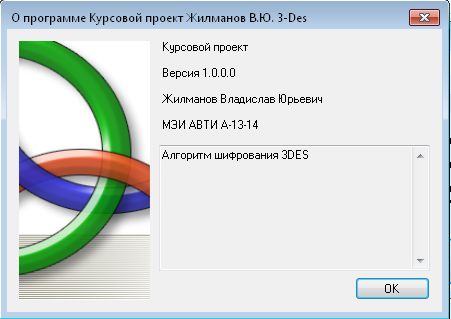
Если парольная фраза подтверждена не верно или окно было закрыто, то появится соответствующее сообщение и шифрование произведено не будет.



(Рисунок 2.7 Ошибка подтверждения парольной фразы)

#### 2.1.3. Окно справки

Содержит сведения о программе.



#### 2.1.4. Проектирование программы

Код ответственный за шифрование и расшифрования блоков по алгоритму 3-DES вынесен в отдельный класс TripleDES. Интерфейс этого класса представляют функции TripleDesIncode и TripleDesDecode.

Так же существует класс для работы с парольной фразой. Он проверяет сложность и длину парольной фразы, а так же преобразует её к трём первичным ключам для алгоритма 3-DES.

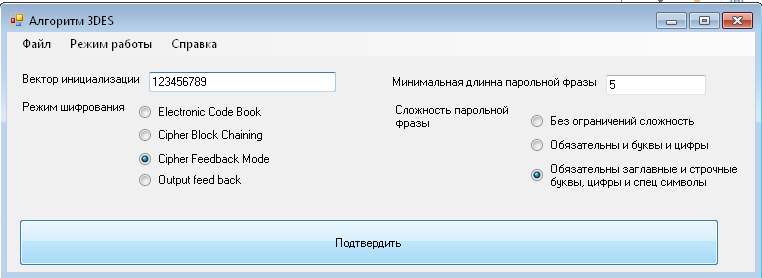
Класс ConvertBlocks преобразует строки текста в блоки длинны 64 бита для шифрования. А так же расшифрованные блики в текст.

# Глава 3. Результаты тестирования разработанной программы.

Программа разрабатывалась и отлаживалась на языке программирования C#, ориентированного на .NET Framework 4.5 в среде программирования Microsoft Visual Studio 2013.

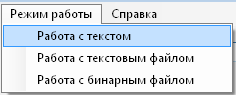
## 3.1 Пример 1

Протестируем программу в режиме Cipher Feedback Mode задав вектор инициализации 123456789. Требования к парольной фразе: длинна не менее 5 символов с обязательным наличием заглавных и строчных букв, цифр и спец символов.



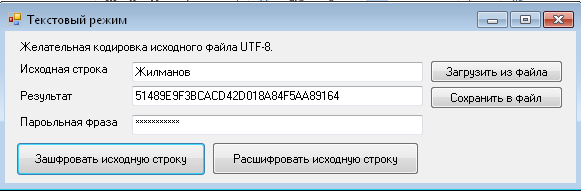
(Рисунок 3.1. Параметры для первого примера)

В качестве режима шифрования выберем режим работы с текстом.



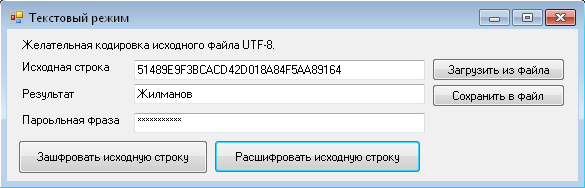
(Рисунок 3.2 Режим для первого примера)

В качестве парольной фразы укажем «Мэи А-13-14».



(Рисунок 3.3 Шифрование для первого примера)

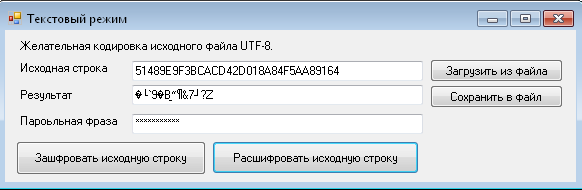
Так же если мы копируем зашифрованную строку и расшифруем её с помощью той же парольной фразы, то получим исходную строку.



(Рисунок 3.3 Расшифрование для первого примера)

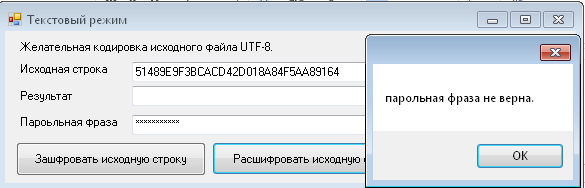
Если же мы укажем не правильную парольную фразу, то исходная строка не будет расшифрована. Укажем как парольную фразу «Мэи А-05-14»

Как видим строка была расшифрована, однако результат расшифрования не верный.



(Рисунок 3.4 Неправильное расшифрование для первого примера)

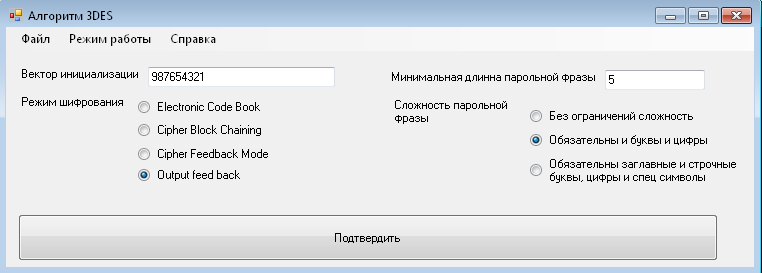
Так же может получиться так, что при расшифровки получатся символы, которые нельзя преобразовать в строку. Например такое получилось, если в качестве парольной фразы указать «Мэи А-13-15» Тогда программа выдаст такое сообщение:



(Рисунок 3.5 Некорректное расшифрование для первого примера)

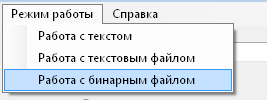
Пример 2.

Протестируем работу с бинарным файлом. Для тестирования выберем, например, режим шифрования Output feed back, вектор инициализации 987654321, а в качестве требований к парольной фразе мин. длину 5 и обязательное наличие букв и цифр.



(Рисунок 3.6. Параметры для второго примера)

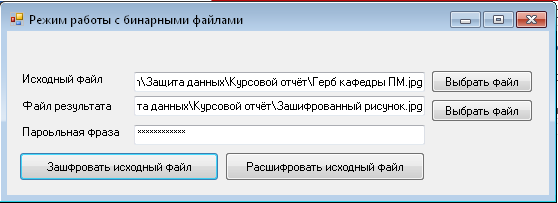
В качестве режима шифрования выберем режим работы с бинарным файлом.



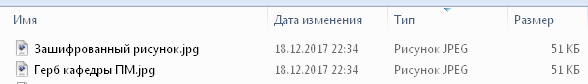
(Рисунок 3.7 Режим для второго примера)

Выберем в качестве файла для шифрование изображение в формате jpg. В качестве парольной фразы укажем «АВТИ-А-13-14».

Нажмём кнопку «Зашифровать», подождём немного времени, т.к. файл достаточно большой и после завершения шифрования зашифрованный файл появится в указанной папке.

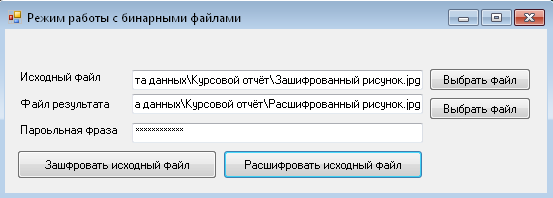


(Рисунок 3.8 Шифрование для второго примера)



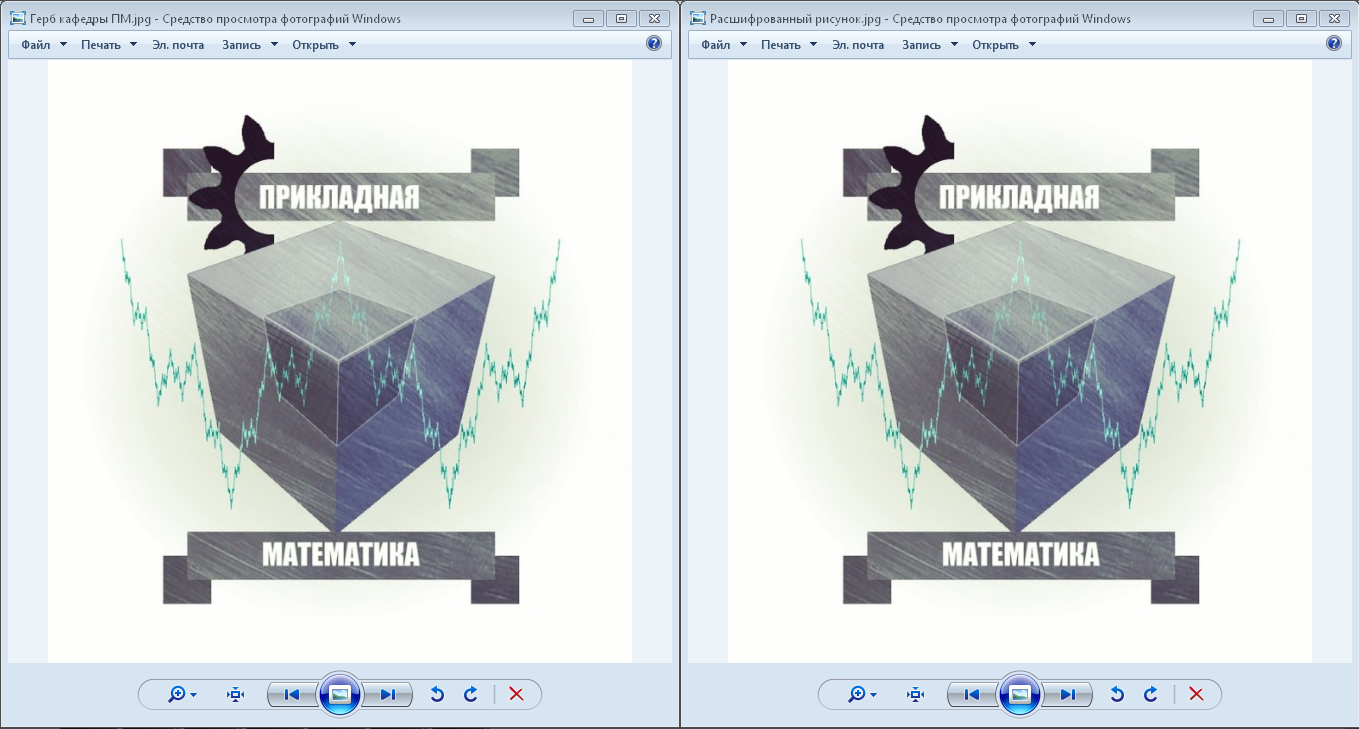
(Рисунок 3.9 Зашифрованный файл для второго примера)

Затем расшифруем полученный файл с помощью той же парольной фразы.



(Рисунок 3.10 Дешифрование для второго примера)

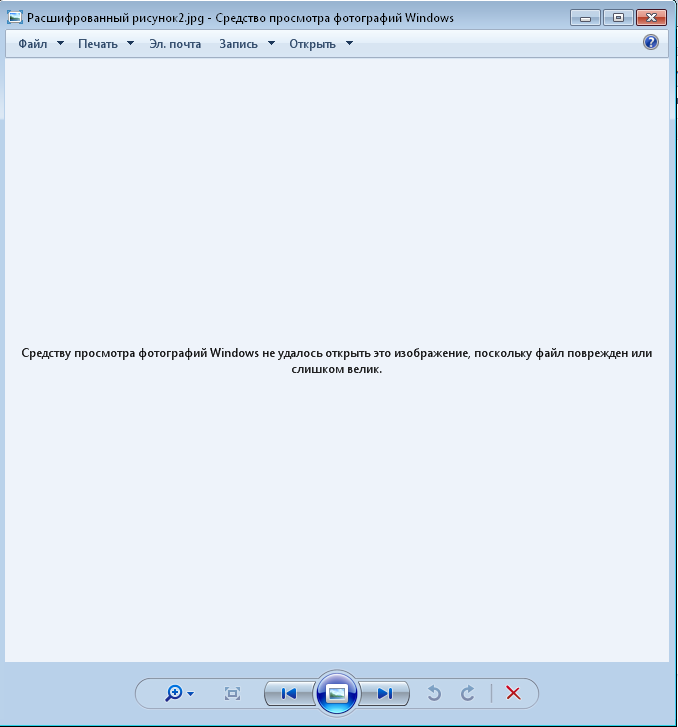
Откроем исходный файл и файл после расшифровки и сравним их.



(Рисунок 3.10 Сравнение файлов для второго примера)

Как видно файлы идентичны.

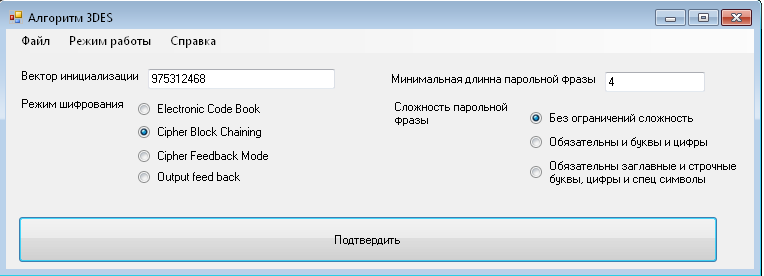
Если же не верно укажем парольную фразу или вектор инициализации, то файл всё равно будет расшифрован, однако открыть его не удастся. Например, укажем в качестве парольной фразы «АВТИ-А-14-14»



(Рисунок 3.12 Некорректная расшифровка для второго примера)

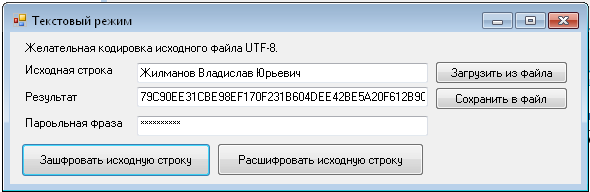
Пример 3.

Протестируем работу с текстом в режиме сцепления блоков шифротекста. Зададим вектор инициализации 975312468, а в качестве требований к парольной фразе мин. длину 4.



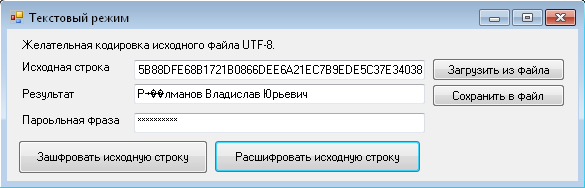
(Рисунок 3.13. Параметры для третьего примера)

В качестве парольной фразы зададим «Москва1996»



(Рисунок 3.14. Шифрование для третьего примера)

Сохраним полученный шифротекст. Теперь поменяем вектор инициализации на другой, например: 977312468.



(Рисунок 3.15. Расшифрование с неправильным для третьего примера)

Одна из особенность режима сцепления блоков шифротекста наличие механизма распространения ошибки: если при передаче произойдёт изменение одного бита шифротекста, данная ошибка распространится и на следующий блок. Однако на последующие блоки ошибка не распространится, поэтому режим CBC также называют самовосстанавливающимся.

Тестирование и отладка графического интерфейса пользователя позволили исключить выполнение пользователем некорректных в точки зрения логики программы действия, в том числе не предусмотренные возможности параллельного запуска алгоритма шифрования.

# Заключение.

В результате выполнения данной работы было разработано программное обеспечение, позволяющее шифровать и расшифровывать по алгоритму 3-DES как выбираемые произвольные файлы, так и вводимые текстовые сообщения с возможностью сохранения в файле полученного шифротекста. Введена возможность вывода ключей шифрования из парольной фразы с регулируемой минимальной длиной и сложностью.

При реализации программы был изучен алгоритм DES, TripleDes, режимы шифрования.

При реализации графического интерфейса были изучены основы платформы .Net.

# Список источников

1. Брюс Шнайер. Прикладная криптография. 2-3 издание. Протоколы, алгоритмы и исходные тексты на языке C.

2. Описание алгоритма DES. Интуит.

<https://www.intuit.ru/studies/courses/691/547/lecture/12377>

3. Описание алгоритма 3-DES в русской википедии

<https://ru.wikipedia.org/wiki/Triple_DES>