**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИБ**

**отчет**

**по лабораторной работе №4**

**по дисциплине «Криптографические методы защиты информации»**

**Тема:** **Изучение шифра DES**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 5381 |  | Кобылянский А.В. |
| Преподаватель |  | Племянников А.К. |

Санкт-Петербург

2018

**Цель работы**

Исследовать шифры DES, 3DES, а также модификаций DESX, DESL, DESXL и получить практические навыки работы с ними, в том числе и в программном продукте Cryptool 1 и 2.

**1.** **Шифр DES**

**1.1 Формулировка задания и содержание этой части отчета.**

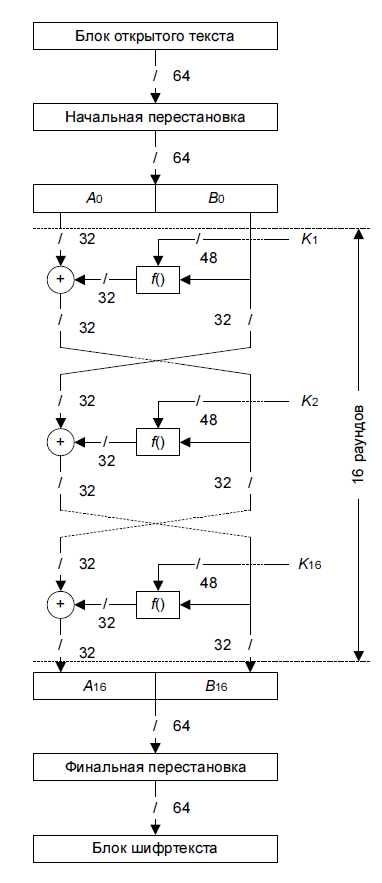
Задание

1. Изучить преобразования шифра DES с помощью демонстрационного приложения из Cryptool 1.
2. Indiv.Procedures-> Visualization…-> DES…
3. Выполнить вручную преобразования одного раунда и вычисление раундовых ключей при следующих исходных данных:
4. Открытый текст (не более 64 бит) – фамилия\_имя (транслитерация латиницей)
5. Ключ (56 бит) – номер зачетной книжки II инициал (всего 7 символов)
6. Выполнить вручную обратное преобразование зашифрованного сообщения
7. Убедиться в совпадении результатов

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.
2. Описание DES c примерами скриншотов из демоприложения.
3. Ручной расчет субблоков и раундовых ключей шифра для первых двух раундов.
4. Ручной расчет обратного преобразования шифровки.

**1.2 Описание DES c примерами скриншотов из демоприложения.**

Стандарт шифрования данных (DES) —блочный шифр с симметричными ключами, разработан Национальным Институтом Стандартов и Технологии (*NIST –National Instituteof Standards and Technology*).

Шифр DES основан на сети Фейстеля.

Алгоритм DES шифрует информацию блоками по 64 бита с помощью 64-битного ключа шифрования.  
 Шифрование выполняется следующим образом (рис. 4.1):

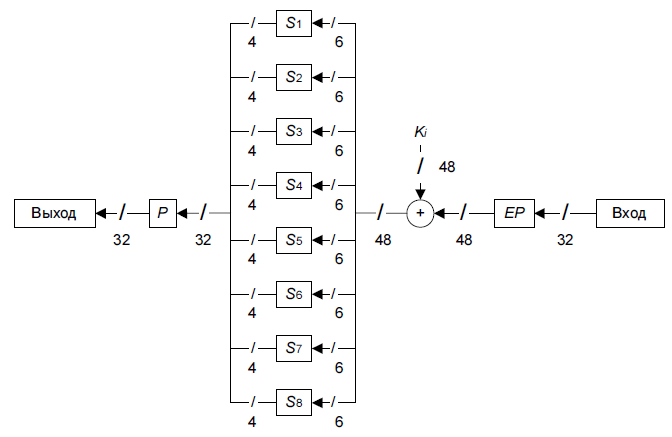
1. Над 64-битными блоками производится начальная перестановка согласно таблице.
2. Результат предыдущей операции делится на 2 субблока по 32 бита ( и ), над которыми производятся 16 раундов преобразований:

;,

Где: i- номер текущего раунда, - ключ раунда, - логическая операция xor.

Рисунок 4.1

Структура функции раунда *f()* представлена на рисунке 4.2. Этапы функции:

1. Расширяющая перестановка *EP*, которая преобразует входные 32 бита в 48 бит (рис. 4.3).
2. Полученные 48 бит складываются с операцией xor.

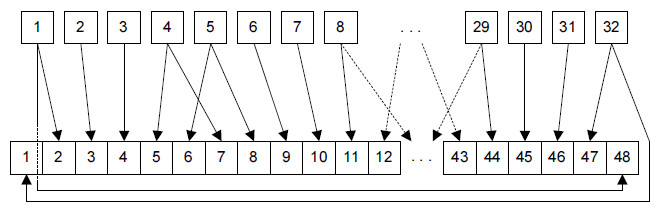
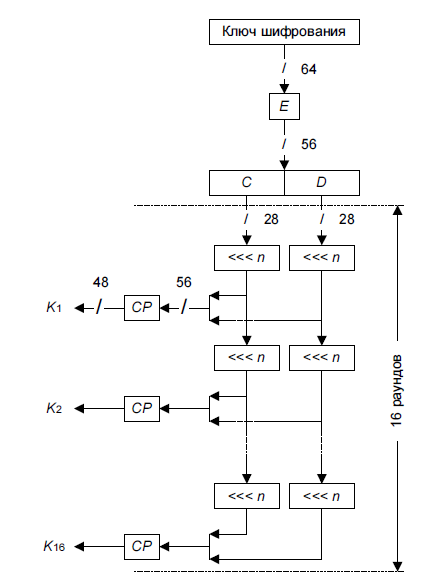


Рисунок 4.3

Рисунок 4.2

1. Результат сложения разбивается на 8 блоков по 6 битов. Каждый блок обрабатывается соответствующей таблицей замен.
2. Над полученными 32 битами, после выполнения замен, выполняется перестановка (на рисунке 4.2 обозначена *P*).

На последнем раунде алгоритма субблоки не меняются местами.

1. Полученные субблоки и образуют 64-битный блок, над которым производится перестановка*.* Результатом перестановки является шифротекст.

Процедура генерации раундовых ключей представлена на рисунке 4.4. Из 64-битного ключа шифрования используется только 56 бит, каждый 8-й бит исключается. На рисунке 4.4 операция сжатия ключа и перестановка обозначена как *E.*

Рисунок 4.4

После перестановки блок в 56 бит делится на два 28-битных блока (*C* и *D*). Затем выполняются 16 раундов преобразований:

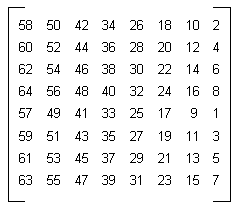
1. Текущие *C* и *D* циклически сдвигаются влево на определенное количество бит.
2. *C* и *D* объединяются в 56-битное значение, к которому применяется сжимающая перестановка*.* На выходе получаем 48-битный раундовый ключ.

Расшифровывание данных алгоритмом DES происходит при прохождении всех шагов алгоритма в обратном порядке.

**1.3. Ручной расчет субблоков и раундовых ключей шифра для первого раунда.**

a. Открытый текст (не более 64 бит) – фамилия\_имя (транслитерация латиницей) : **KOBILIAN**SKY

b. Ключ (56 бит) – номер зачетной книжки II инициал (всего 7 символов) : 538125A



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

Input Block

Initial Permutation matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

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

Permuted input

Вспомогательная таблица

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |  |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |  |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |  |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |  |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |  |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |  |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |  |

Permuted choice (PC1)

Ключ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 |  | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 |  | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 |  | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 |  | 0 | 0 | 1 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |

L часть ключа

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |

R часть таблицы

56-bit key

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |

L часть ключа после сдвига

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |

R часть ключа после сдвига

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |

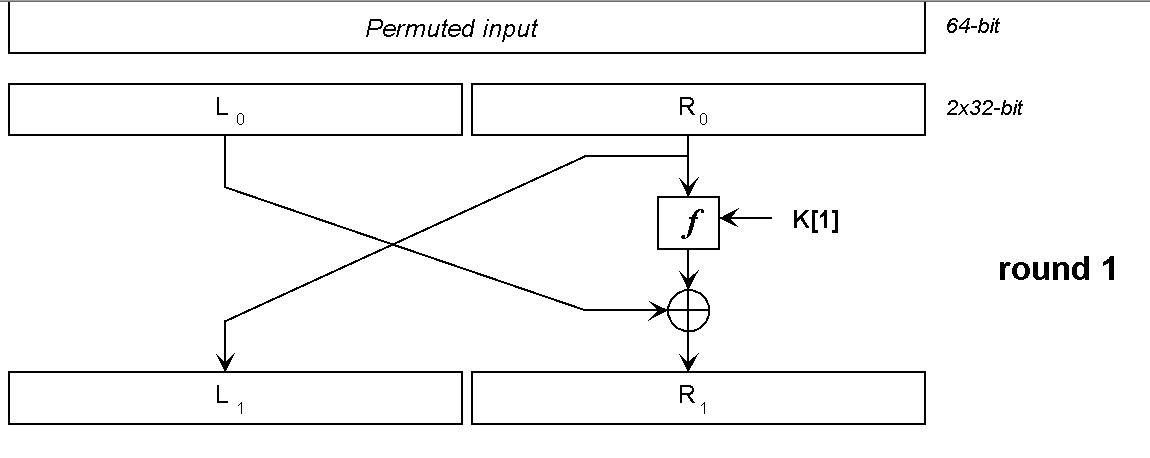
Ключ после объединения L и R

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

PC2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 |

Первый раундовый ключ



Алгоритм работы первого раунда

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

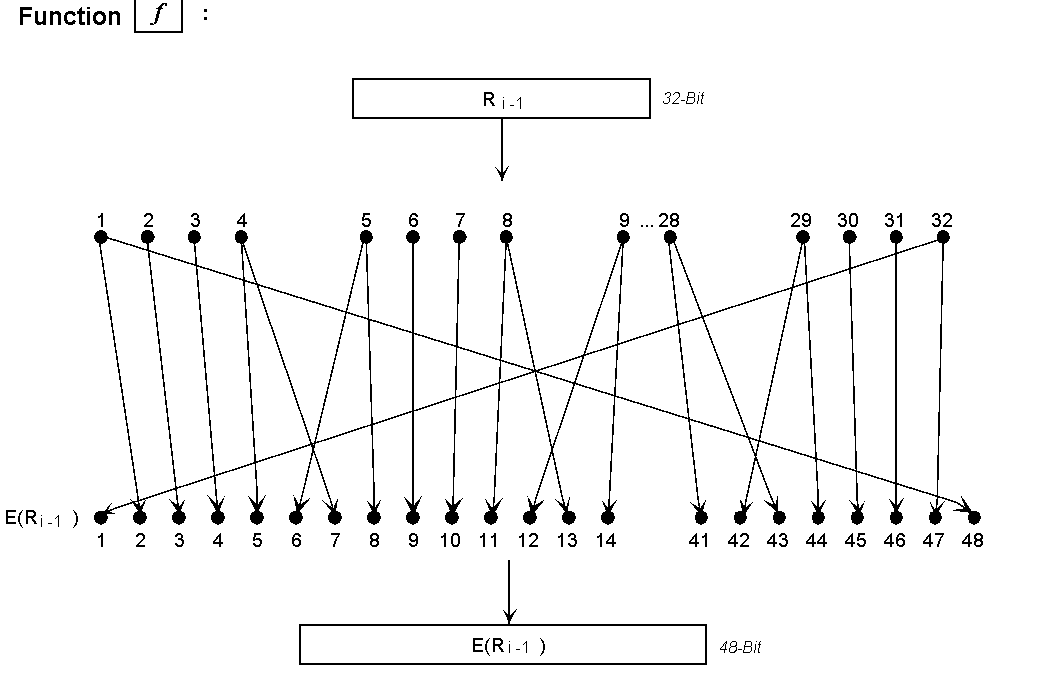
Permuted Input

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

L\_0 часть Permuted Input

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

R\_0 часть Permuted Input



Алгоритм функции f()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B[1] | 0 | 0 | 0 | 1 | 0 | 0 |
| B[2] | 0 | 1 | 0 | 0 | 0 | 0 |
| B[3] | 1 | 1 | 0 | 0 | 1 | 1 |
| B[4] | 0 | 0 | 1 | 1 | 0 | 0 |
| B[5] | 0 | 1 | 0 | 0 | 1 | 1 |
| B[6] | 0 | 0 | 0 | 1 | 0 | 1 |
| B[7] | 1 | 1 | 0 | 1 | 0 | 0 |
| B[8] | 1 | 1 | 0 | 1 | 0 | 1 |

Результат применения xor и разбиение результата на 8 блоков

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S1** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **0** | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| **1** | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| **2** | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| **3** | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

Первый S-box

Первый и последний бит B[n] образуют номер строки в соответствующем S-box’е. Для B[1] это значение равно 11 в двоичной системе, т.е. 3 в десятичной. Все остальные биты образуют номер столбца, для B[1] это значение равно 0010 в двоичной системе, т.е. 2 в десятичной. В итоге мы получаем значение на пересечении 3й строки и 2го столбца, т.е. 8 в десятичной системе или 1000 в двоичной. По аналогии для B[2] – B[8] получаем:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

Результат выполнения преобразований S-box'ов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Permutation table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

Результат работы f()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |

Результат xor f() и L\_0, т.е. R\_1

В итоге после первого раунда получаем сообщение:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

**1.4 Ручной расчет обратного преобразования шифровки.**

Пытаемся его расшифровать.

R(i-1)=L\_i

L(i-1)=R\_i xor f(L\_i,k\_i)

Т.к. мы прошли только один раунд, то i = 1, R\_0 = L\_1, а для получения L\_0 нам необходимо повторить алгоритм. Для начала посчитаем функцию f(L\_i, k\_i).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B[1] | 0 | 0 | 0 | 1 | 0 | 0 |
| B[2] | 0 | 1 | 0 | 0 | 0 | 0 |
| B[3] | 1 | 1 | 0 | 0 | 1 | 1 |
| B[4] | 0 | 0 | 1 | 1 | 0 | 0 |
| B[5] | 0 | 1 | 0 | 0 | 1 | 1 |
| B[6] | 0 | 0 | 0 | 1 | 0 | 1 |
| B[7] | 1 | 1 | 0 | 1 | 0 | 0 |
| B[8] | 1 | 1 | 0 | 1 | 0 | 1 |

Рисунок 1.Результат применения xor и разбиение результата на 8 блоков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

Рисунок 2.Результат выполнения преобразований S-box'ов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

Рисунок 3. Результат применения Permutation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

Рисунок 4. Применив xor и добавив к результату вторую часть матрицы, мы получили первоначальную (Рис. 9)

Осталось только изменить порядок символов согласно и сообщение будет окончательно расшифровано.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Рисунок 5. Final permutation (IP−1)

**2. Исследование DES в режимах ECB и CBC**

**2.1. Формулировка задания и содержание этой части отчета.**

Задание

1. Создать картинку со своими ФИО (формат bmp).
2. Зашифровать картинку шифром DES в режиме ECB.
3. Зашифровать картинку шифром DES в режиме CBC c тем же ключом.
4. Сохранить скриншоты картинок для отчета.
5. Сжать исходную и 2 зашифрованных картинки средствами CrypTool. Зафиксировать размеры полученных файлов в таблице.
6. Выбрать случайный текст на английском языке (не менее 1000 знаков) и зашифровать его DES в режиме ECB.
7. Для одного и того же шифротекста оцените время проведения атаки «грубой силы» в случаях, когда известно n-4, n-6, n-8,.., 2 байт секретного ключа. Зафиксировать результаты измерений в таблице.
8. Повторить подобные измерения для DES в режиме CBC.

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.
2. Основные параметры и обобщенная схема шифров (как в лекции).
3. Скриншоты исходного и шифрованных изображений в разных режимах работы шифров.
4. Таблица сравнений результатов сжатия исходного и шифрованных изображений.
5. Таблица зависимости оценки времени атаки грубой силы от размера известной части ключа.
   1. **Основные параметры и обобщенная схема шифров (как в лекции).**

Режим ECB шифра DES работает независимо с каждым 64-битным блоком шифруемых данных. Структура работы режима ECB представлена на рисунке 4.5.

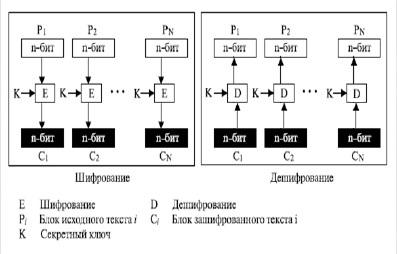
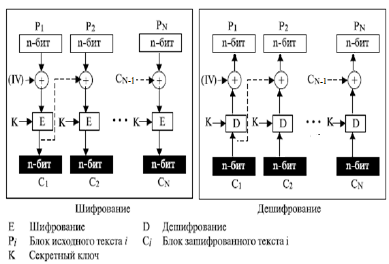
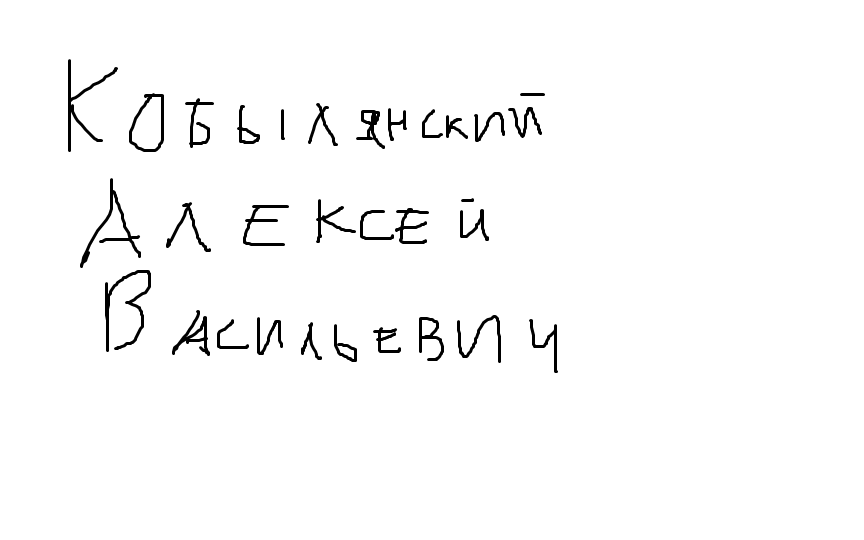
Режим CBC шифра DES перед запуском шифрования каждого очередного блока складывает его с предыдущим операцией xor. Структура работы режима CBC представлена на рисунке 4.6.

Рисунок 4.5

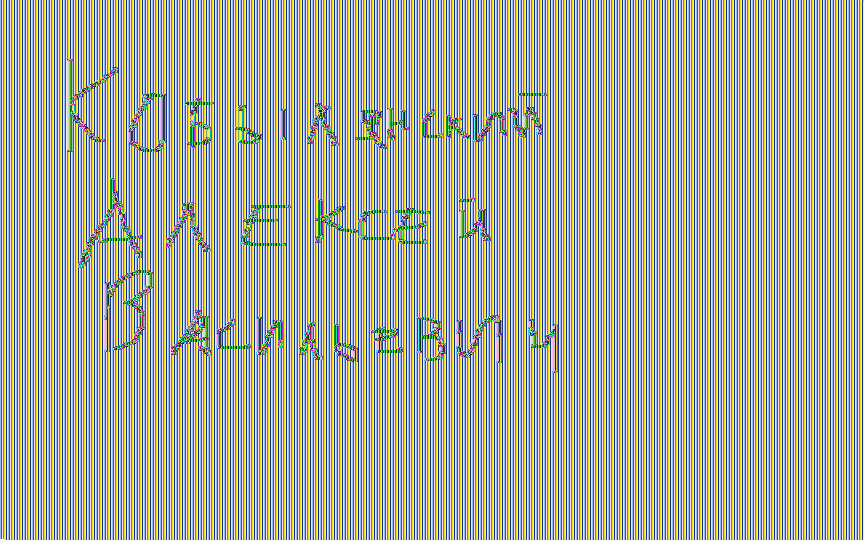
Рисунок 4.6

* 1. **Скриншоты исходного и шифрованных изображений в разных режимах работы шифров.**

Исходное

****

DES EBC



DES CBC



* 1. **Таблица сравнений результатов сжатия исходного и шифрованных изображений.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **% сжатия** | **Размер до сжатия, КБ** | **Размер после сжатия, КБ** |
| **Оригинал** | 99 | 1367 | 12 |
| **DES CBC** | 0 | 1368 |
| **DES ECB** | 97 | 33 |

**2.5 Таблица зависимости оценки времени атаки грубой силы от размера известной части ключа.**

Для ECB:

|  |  |
| --- | --- |
| Кол-во известных бит ключа | Время выполнения атаки «грубой силы» |
| 12 | 0 сек |
| 10 | 35 сек |
| 8 | 1 час 15 мин |
| 6 | 7.3 дней |
| 4 | 2.3 лет |
| 2 | 320 лет |

Для CBC:

|  |  |
| --- | --- |
| Кол-во известных бит ключа | Время выполнения атаки «грубой силы» |
| 12 | 0 сек |
| 10 | 60 сек |
| 8 | 2.5 час |
| 6 | 10 дней |
| 4 | 4 года |
| 2 | 440 лет |

1. **Исследование 3-DES**

**3.1. Формулировка задания и содержание этой части отчета.**

Задание

1. Выбрать случайный текст на английском языке (не менее 1000 знаков).
2. Создать бинарный файл с этим текстом, зашифровав и расшифровав его DES на 0-м ключе.
3. Снять и сохранить частотную и автокорреляционную характеристику этого файла.
4. Зашифровать бинарный файл шифром 3-DES в режиме ECB.
5. Снять и сохранить частотную и автокорреляционную характеристику файла с шифровкой.
6. Зашифровать исходный бинарный файл 3-DES в режиме CBC c тем же ключом.
7. Снять и сохранить частотную и автокорреляционную характеристику файла с шифровкой.
8. Определить экспериментальным путем по какой схеме работает реализация 3-DES в CrypTool. Сохранить подтверждающие скриншоты.

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.
2. Основные параметры и обобщенная схема шифра (как в лекции).
3. Скриншоты исходного и зашифрованного текста в бинарном представлении.
4. Скриншоты частотной и автокорреляционной характеристик исходного текста и шифровки.
5. Таблица зависимости времени атаки грубой силы от размера известной части ключа.
6. Схема реализации в CrypTool 1 и подтверждающие скриншоты.

**3.2 Основные параметры и обобщенная схема шифра (как в лекции).**

Шифр 3-DES (рисунок 4.7) состоит в трехкратном применении обычного DES. Существует 4 основные версии данного шифра:

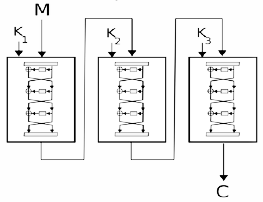
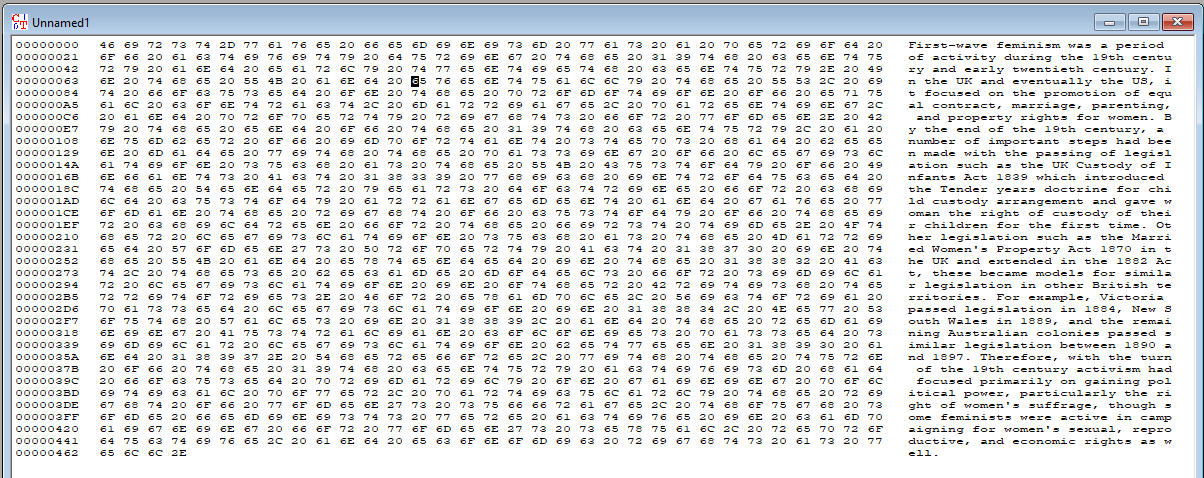
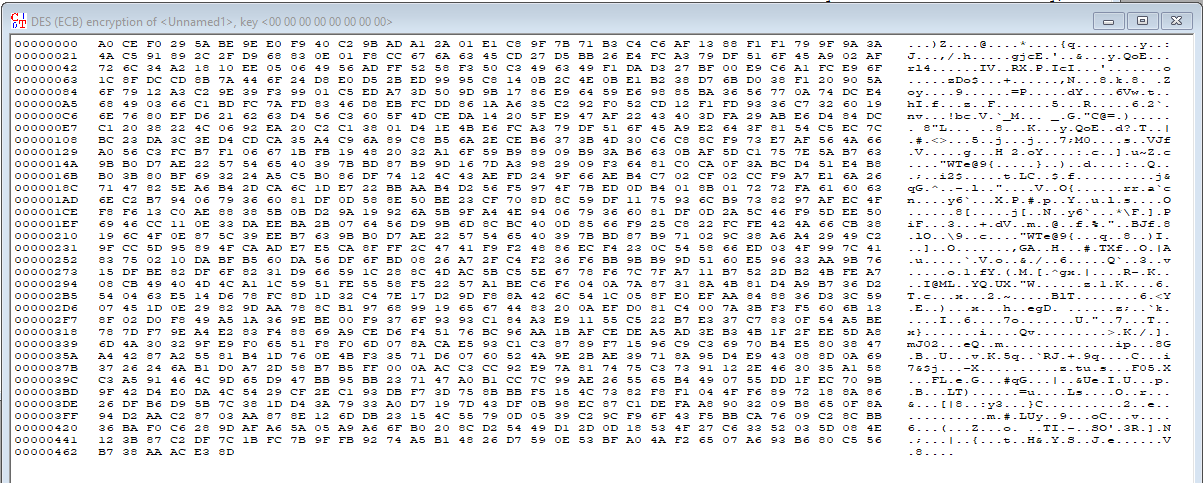
1. ****DES-EEE3 – шифрование происходит 3 раза независимыми ключами
2. DES-EDE3 – операции шифровка-расшифровка-шифровка с тремя разными ключами
3. DES-EEE2 – то же что и DES-EEE3, но на первом и последнем шаге одинаковый ключ
4. DES-EDE2 – то же что и DES-EDE3, но на первом и последнем шаге одинаковый ключ

Рисунок 4.7

На текущий момент самыми популярными разновидностями шифра являются DES-EDE3 и DES-EDE2.

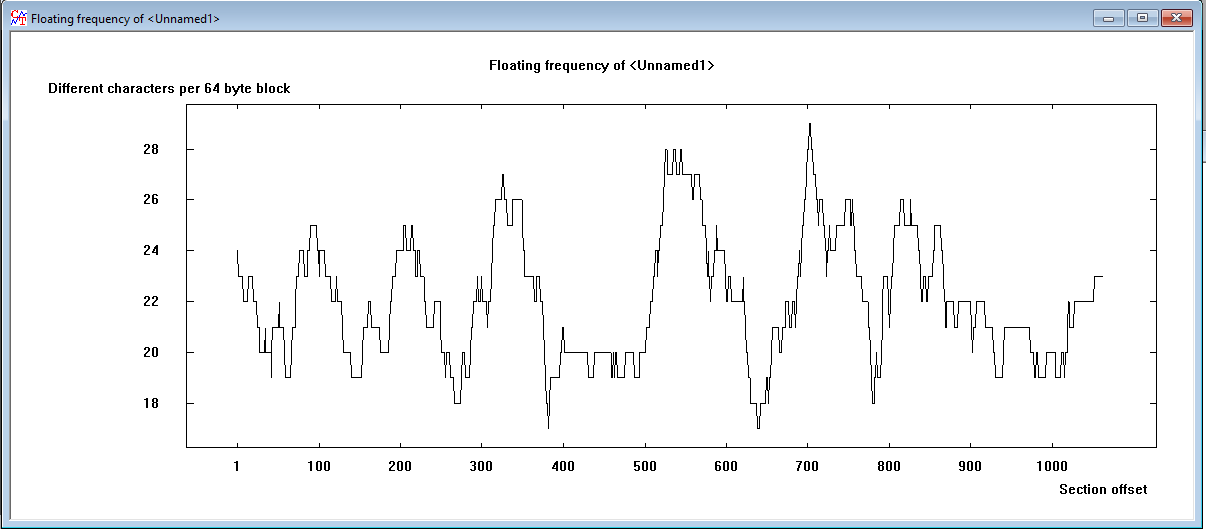
**3.3 Скриншоты исходного и зашифрованного текста в бинарном представлении.**

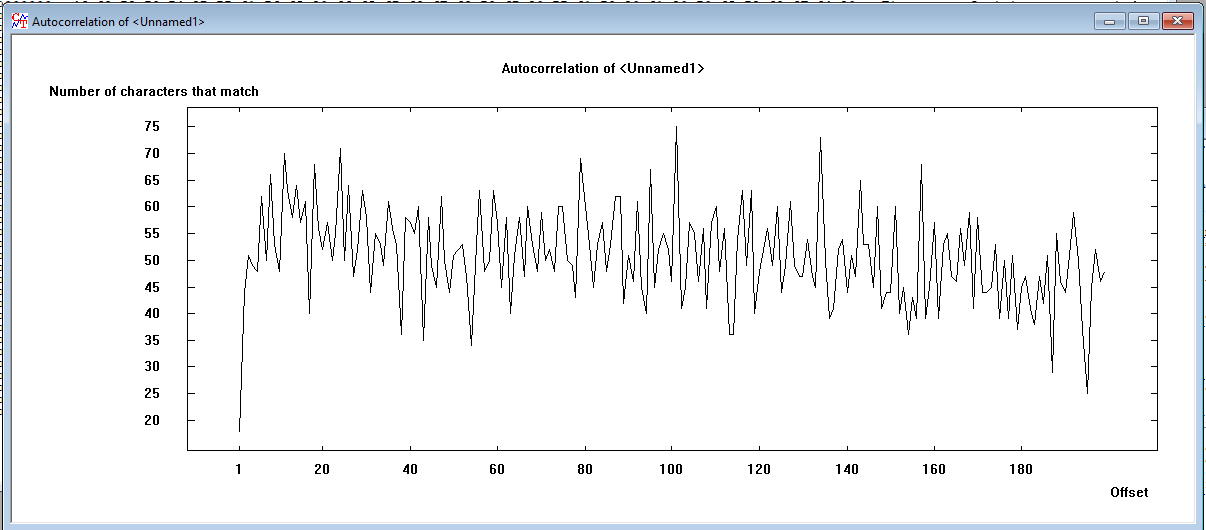
****

****

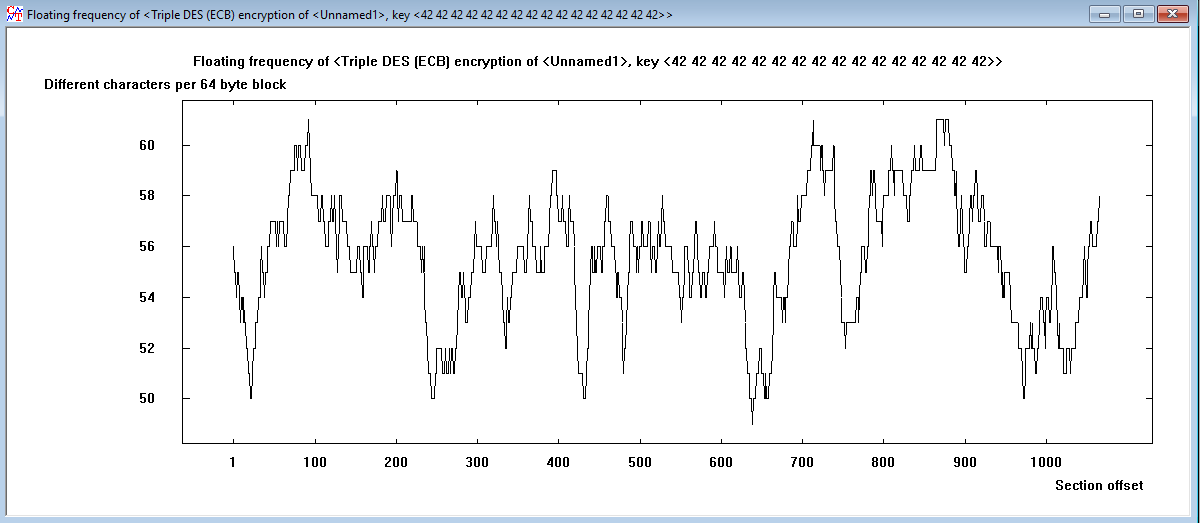
* 1. **Скриншоты частотной и автокорреляционной характеристик исходного текста и шифровки.**

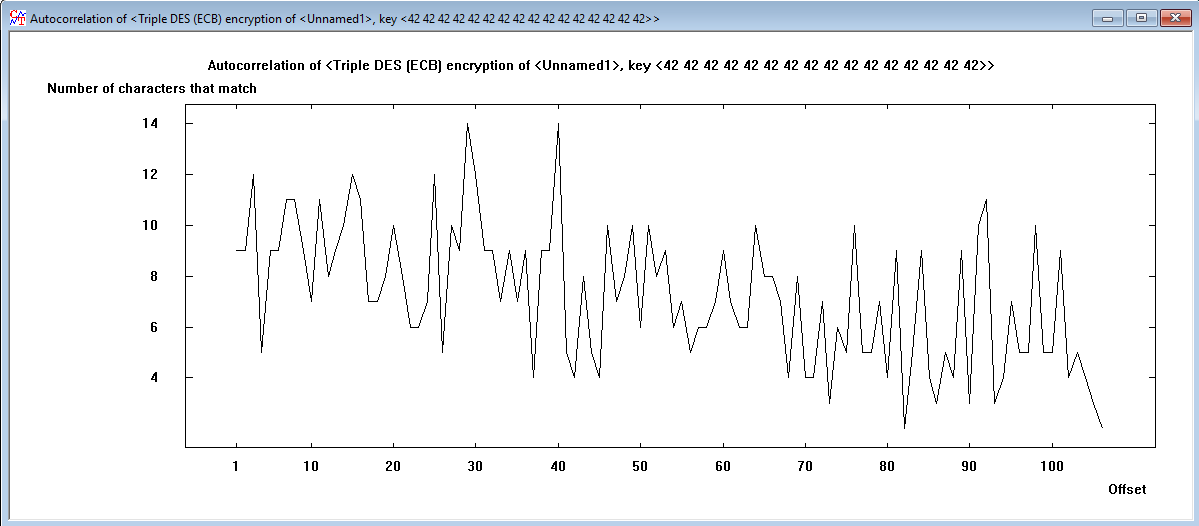
Исходный текст



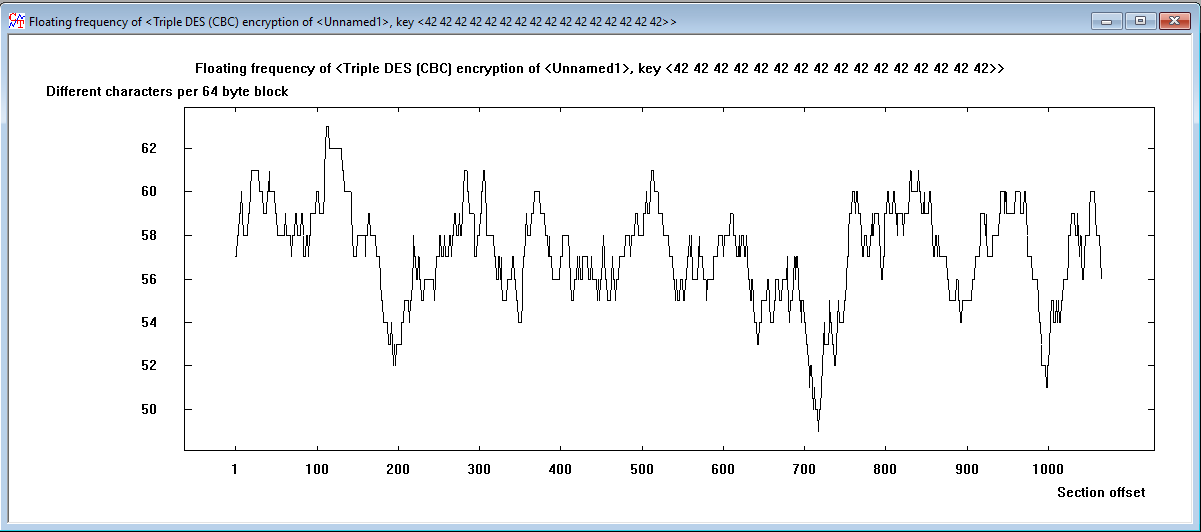


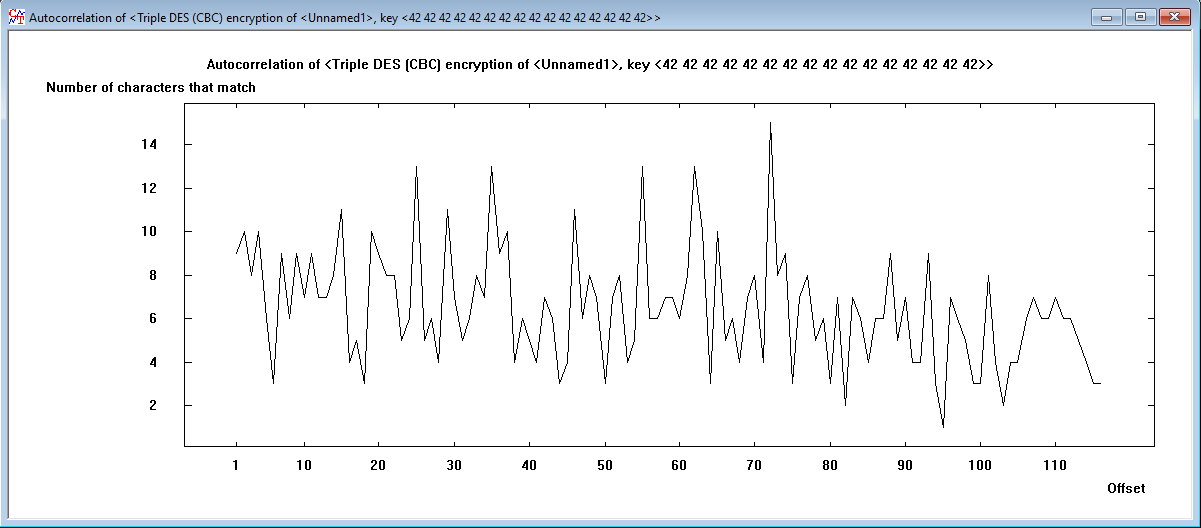
3DES ECB

****

****

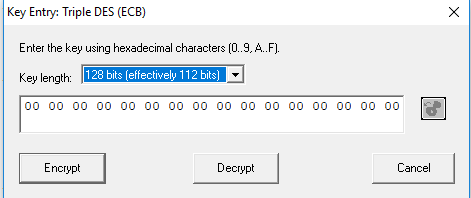
3DES CBC

****

****

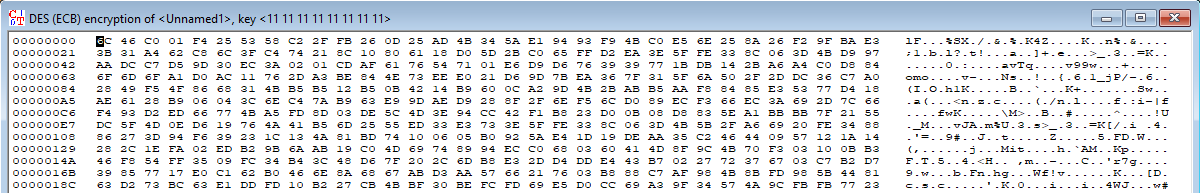
**3.5 Схема реализации в CrypTool 1 и подтверждающие скриншоты.**

В Cryptool 1 для 3DES требуется 128 байтовый ключ, т.е. два 64 байтовых ключа. Значит используется один из алгоритмов 3DES-\*\*\*2, где \* это E или D.

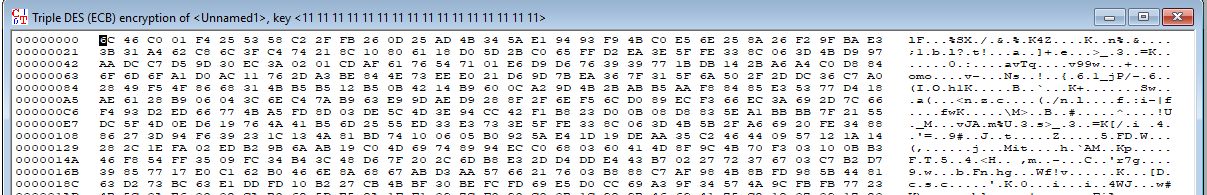


Зашифруем текст шифром DES ключом 11-11-11-11, а затем этот же текст зашифруем 3DES с ключом 11-11-11-11-11-11-11-11. Если это 3DES-EDE2, то ED дает исходный текст, а E дает одиночное шифрование DES(11-11-11-11) и результаты должны совпасть.

Результат DES(11-11-11-11)



Результат 3DES(11-11-11-11-11-11-11-11)



Результат совпал. В Cryptool 1 используется 3DES-EDE2.

1. **Исследование модификаций DESX, DESL, DESXL шифра DES**

**4.1. Формулировка задания и содержание этой части отчета.**

Задание

1. Выбрать случайный текст на английском языке (не менее 1000 знаков).
2. Создать бинарный файл с этим текстом, зашифровав и расшифровав его DES на 0-м ключе.
3. С помощью CrypTool зашифровать текст с использованием шифров DESX, DESL, DESXL.
4. Средствами CrypTool вычислить энтропию исходного текста и шифротекстов, полученных в итоге. Зафиксировать результаты измерений в таблице.
5. Средствами CrypTool оцените время проведения атаки «грубой силы» при полном отсутствии информации о секретном ключе

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.
2. Основные параметры и обобщенные схемы шифров (самостоятельно).
3. Таблица зависимости энтропии шифротекста от используемого шифра.
4. Таблица зависимости времени подбора ключа от используемого шифра.

**4.2 Основные параметры и обобщенные схемы шифров (самостоятельно).**

Алгоритм DESX использует на входе ключ длиной 184 бита, который делится на 3 56-битные части. Процесс шифрования происходит по следующей схеме:

Если , то данный алгоритм сводится к стандартному DES.

Алгоритм DESL является облегченной версией алгоритма DES. Данный алгоритм был создан в 2006 году для RFID-меток. Алгоритм предполагает отказ от входной и выходной перестановки блока текста, т.к. они не несут криптографической сложности, а также 8 S-блоков заменяется на 1, но более стойкий чем все 8 стандартных блока DES.

Алгоритм DESXL использует те же оптимизации что и DESL, но производит шифрование по алгоритму DESX.

**4.3 Таблица зависимости энтропии шифротекста от используемого шифра.**

|  |  |
| --- | --- |
|  | **Энтропия** |
| **Исходный текст** | 4.47 |
| **DESX** | 7.82 |
| **DESL** | 7.83 |
| **DESXL** | 7.83 |

**4.4 Таблица зависимости времени подбора ключа от используемого шифра.**

|  |  |
| --- | --- |
|  | **Оценка грубой силы (лет)** |
| **DESX** |  |
| **DESL** |  |
| **DESXL** |  |

**Вывод**

ECB:

* каждый блок шифруется/расшифровывается независимо от других блоков.
* постоянная скорость обработки блоков (скорость определяется эффективностью реализации шифра);
* возможно распараллеливание вычислений (так как блоки не связаны между собой).

CBC:

* постоянная скорость обработки блоков (скорость определяется эффективностью реализации шифра; время выполнения операции «xor» пренебрежимо мало);
* отсутствие статистических особенностей, характерных для режима ECB (поскольку каждый блок открытого текста «смешивается» с блоком шифротекста, полученным на предыдущем шаге шифрования);
* невозможность распараллеливания шифрования (поскольку для шифрования каждого i-го блока требуется блок, зашифрованный на предыдущем шаге (блоки связаны между собой)

Из модификаций DES были изучены 3-DES, DESX, DESL, DESXL.

Шифр 3DES строится на основе нескольких использований алгоритма DES. Самые стойкие – DES-EDE3 и DES-EDE2, опирающиеся на сочетании DES операций шифрования, расшифровки и шифрования.

Отличия:

DES-EDE3 - 3DES операции шифровка-расшифровка-шифровка с тремя разными ключами;

DES-EDE2 - на первом и третьем шаге используется одинаковый ключ

Шифр DESX - наиболее стойкая модификация шифра DES, обеспечивается ключом длиной 184 бита, который делится на 3 56-битные части.

Шифр DESL - отказ от входной и выходной перестановки, замены 8 S-блоков на 1 (но более стойкий).

Шифр DESX объединяет в себе оптимизацию DESL и стойкость DESX.

Наиболее стойким оказался DESX.