

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Криптография и защита информации»
Тема: Изучение шифра DES

Студент гр. 8383

Ларин А.

Преподаватель

Племянников А.К.

Санкт-Петербург

2021

Цель работы

Исследовать шифры DES, 3DES, а также другие модификации шифра DES: DESX, DESL, DESXL и получить практические навыки работы с ними, в том числе с использованием приложения Cryptool 1 и 2.

1. Исследование преобразований DES

Задание

1. Изучить преобразования шифра демонстрационного приложения из Cryptool 1.
 - a. Indiv.Procedures-> Visualization...-> DES...
2. Выполнить вручную преобразования первых двух раундов и вычисление раундовых ключей при следующих исходных данных:
 - a. Открытый текст (не более 64 бит) – фамилия_имя (транслитерация латиницей)
 - b. Ключ (56 бит) – номер зачетной книжки и инициал отчества (всего 7 символов)
3. Выполнить вручную обратное преобразование зашифрованного сообщения
4. Убедиться в совпадении результатов

Выполнение

Описание DES

Стандарт шифрования данных (DES) — блочный симметричный шифр, разработанный Национальным Институтом Стандартов и Технологии (NIST – National Institute of Standards and Technology). Шифр DES основан на сети Фейстеля. DES шифрует информацию блоками по 64 бита с помощью 64-битного ключа шифрования.

Ручной расчет

Ключ

838310d

Дополнение до 64 бит

[0 0 1 1 1 0 0 0]

[0 0 0 1 1 0 0 0]

[1 1 0 0 1 1 1 0]

[0 0 0 0 0 1 1 0]

[0 0 1 1 0 0 1 0]

[1 0 0 0 1 0 0 0]

[1 1 0 0 0 0 0 0]

[1 1 0 0 1 0 0 0]

Ключ 56 бит

[1 1 1 0 0 1 0]

[0 1 1 0 0 0 1]

[0 0 0 0 0 1 0]

[0 0 1 0 0 0 1]

[0 0 0 1 1 1 0]

[0 0 0 0 0 1 1]

[0 0 1 0 1 0 0]

[1 1 1 0 0 1 1]

L

[1 1 1 0 0 1 0]

[0 1 1 0 0 0 1]

[0 0 0 0 0 1 0]

[0 0 1 0 0 0 1]

R

[0 0 0 1 1 1 0]

[0 0 0 0 0 1 1]

[0 0 1 0 1 0 0]

[1 1 1 0 0 1 1]

K1

[0 0 1 0 1 0]

[1 0 1 0 1 1]

[0 0 0 1 0 0]

[0 1 0 0 0 1]

[0 1 0 0 0 0]

[0 0 1 0 1 1]

[0 0 0 1 1 1]

[1 1 0 0 1 0]

K2

[0 0 1 0 0 1]

[1 0 0 0 0 0]

[0 0 1 1 1 1]

[1 0 0 0 0 1]

[0 1 0 0 1 0]

[1 0 0 0 0 0]

[1 1 0 0 1 1]

[0 0 0 1 1 0]

Текст latin_an

Биты

[0 1 1 0 1 1 0 0]

[0 1 1 0 0 0 0 1]
 [0 1 1 1 0 0 1 0]
 [0 1 1 0 1 0 0 1]
 [0 1 1 0 1 1 1 0]
 [0 1 0 1 1 1 1 1]
 [0 1 1 0 0 0 0 1]
 [0 1 1 0 1 1 1 0]

После перестановки

[1 1 1 1 1 1 1 1]
 [0 0 1 0 0 1 0 0]
 [1 0 1 1 0 0 0 1]
 [0 1 1 0 1 0 1 0]
 [0 0 0 0 0 0 0 0]
 [1 1 0 1 1 1 1 1]
 [1 0 1 1 1 0 0 1]
 [1 0 1 1 0 1 0 0]

R0

[0 0 0 0 0 0 0 0]
 [1 1 0 1 1 1 1 1]
 [1 0 1 1 1 0 0 1]
 [1 0 1 1 0 1 0 0]

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 22 23 24 25 26
 24 25 26 27 28 29
 28 29 30 31 32 1

ER0

[0 0 0 0 0 0 0 0]
 [0 0 0 1 0 1 1 0]
 [1 1 1 1 1 1 1 1]
 [1 1 0 1 1 1 1 1]
 [0 0 1 1 1 1 0 1]
 [1 0 1 0 1 0 0 0]

xor K

[0 0 1 0 1 0]
 [1 0 1 0 1 0]
 [0 1 1 1 1 1]
 [1 0 1 1 1 0]
 [1 0 0 1 1 1]
 [1 1 1 0 0 0]
 [1 1 0 0 0 1]

```

[0 1 1 0 1 0]
→ Si→
[0 0 1 0 1 0]
[1 0 1 0 1 0]
[0 1 1 1 1 1]
[1 0 1 1 1 0]
[1 0 0 1 1 1]
[1 1 1 0 0 0]
[1 1 0 0 0 1]
[0 1 1 0 1 0]
→ P
f
[1 0 1 0 0 1 1 0]
[1 0 0 0 0 1 0 0]
[1 0 1 1 0 0 1 0]
[1 1 0 1 0 0 1 1]
xor L0
R1
[0 1 0 1 1 0 0 1]
[1 0 1 0 0 0 0 0]
[0 0 0 0 0 0 1 1]
[1 0 1 1 1 0 0 1]
L1 = R0
...
L2
[0 1 0 1 1 0 0 1]
[1 0 1 0 0 0 0 0]
[0 0 0 0 0 0 1 1]
[1 0 1 1 1 0 0 1]
R2
[1 0 1 1 1 1 0 0]
[1 0 0 1 1 0 0 0]
[1 0 1 1 1 1 0 1]
[1 0 1 1 1 1 1 0]
→ P →
[0 1 0 0 1 1 0 1]
[0 0 0 0 0 1 1 0]
[1 0 0 0 1 0 1 0]
[1 1 1 0 1 0 1 1]
[1 1 1 0 1 0 1 1]
[1 0 0 1 1 0 1 1]
[0 1 0 0 0 0 0 0]
[1 0 1 1 1 0 1 1]
[77, 6, 138, 235, 235, 155, 64, 187]

```

Обратный расчет

[0 1 1 0 1 1 0 0]

[0 1 1 0 0 0 0 1]

[0 1 1 1 0 0 1 0]

[0 1 1 0 1 0 0 1]

[0 1 1 0 1 1 1 0]

[0 1 0 1 1 1 1 1]

[0 1 1 0 0 0 0 1]

[0 1 1 0 1 1 1 0]

Все верно

2. Исследование DES в режимах ECB и CBC

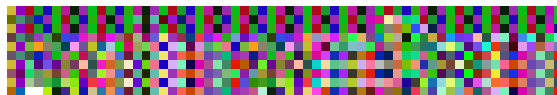
В режиме ECB шифра DES используется независимо для каждого 64-битного блока шифруемых данных.

В режиме CBC перед запуском DES для зашифрования каждого очередного блока открытого текста происходит побитовое XOR-сложение этого блока с блоком зашифрованного текста из предыдущего шага

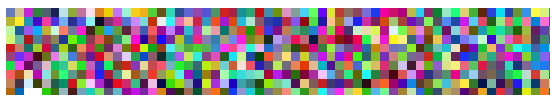
1. Была создана картинка с именем и фамилией larinanton



2. Картинка зашифрована при помощи DES в режиме ECB
Результат представлен ниже



3. Картинка зашифрована при помощи DES в режиме CBC
Результат представлен ниже



4. Картинки сжаты средствами Cryptool
Степени сжатия сведены в таблицу

Исходная	86%
ECB	51%
CBC	0%

5. Была взята часть текста поэмы Raven (1710 символов) и зашифрована DES ECB
6. Оценено время атаки грубой силой в зависимости от числа известных байт. Они сравнены относительно 4 байт.

Известные байты	Время ч.	Время отн 4 байт
-----------------	----------	------------------

4	.7	1
3	86,4	123,428571429
2	1e+4	1.8e+4
1	1e+7	1.7e+7
0	1.5e+8	1.5e+8

CBC

Известные байты	Время ч.
4	1
3	132
2	1.7e+3
1	2e+6
0	2e+8

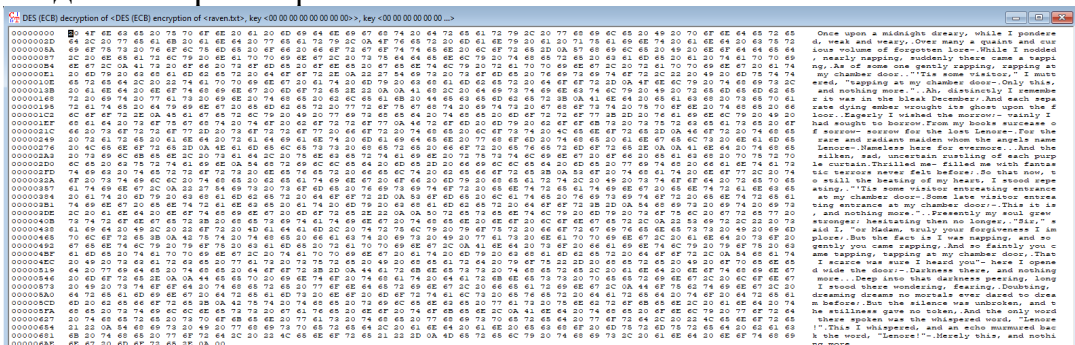
3. Исследование 3-DES

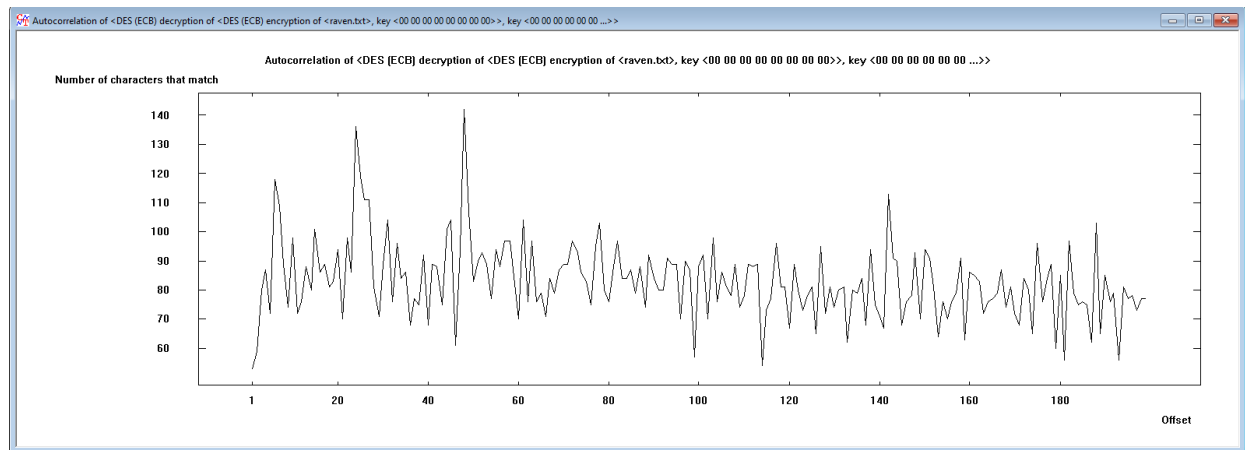
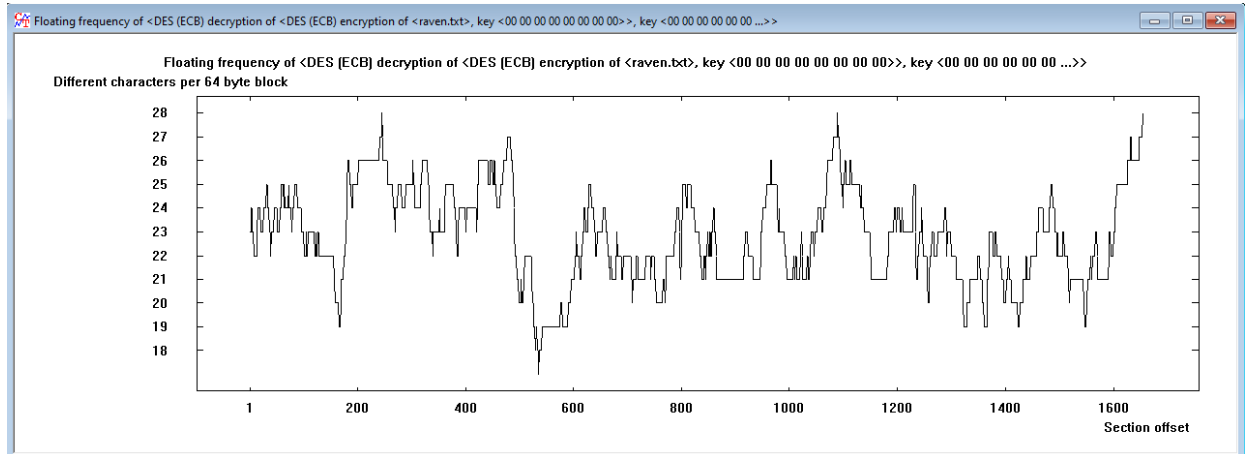
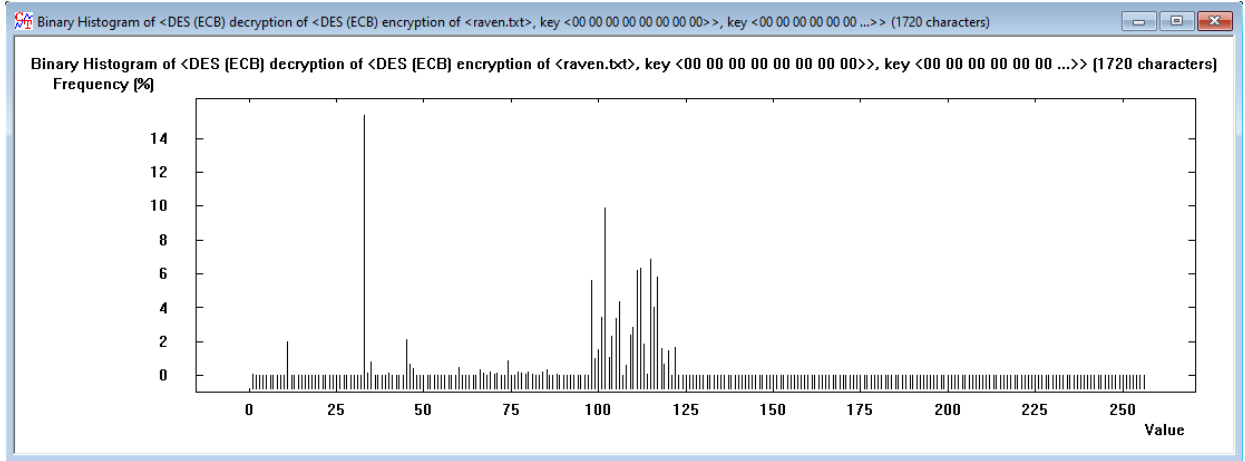
Задание

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 характеристику этого файла.
4. Зашифровать бинарный файл шифром 3-DES в режиме ECB.
5. Снять и сохранить частотную характеристику файла с шифровкой.
6. Зашифровать исходный бинарный файл 3-DES в режиме CBC с 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 СурфTool. Сохранить подтверждающие скриншоты.

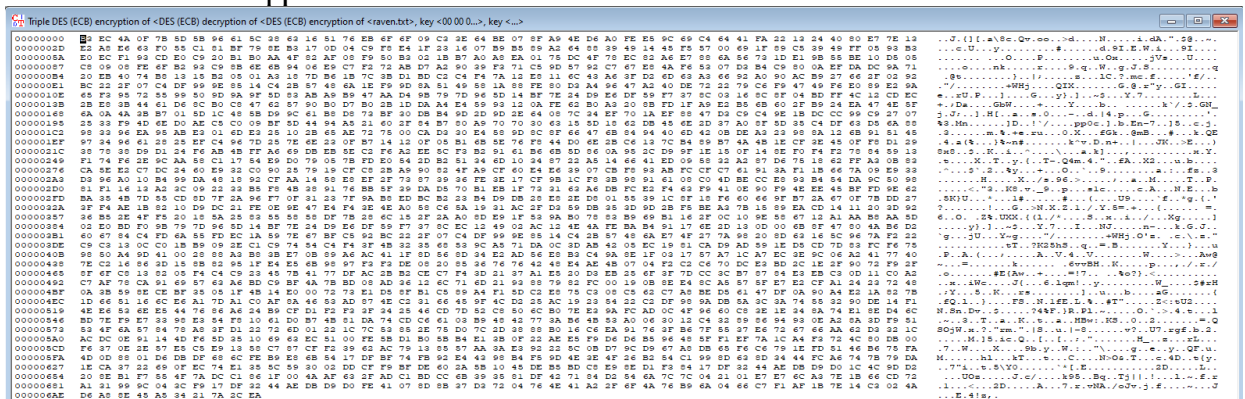
Выполнение

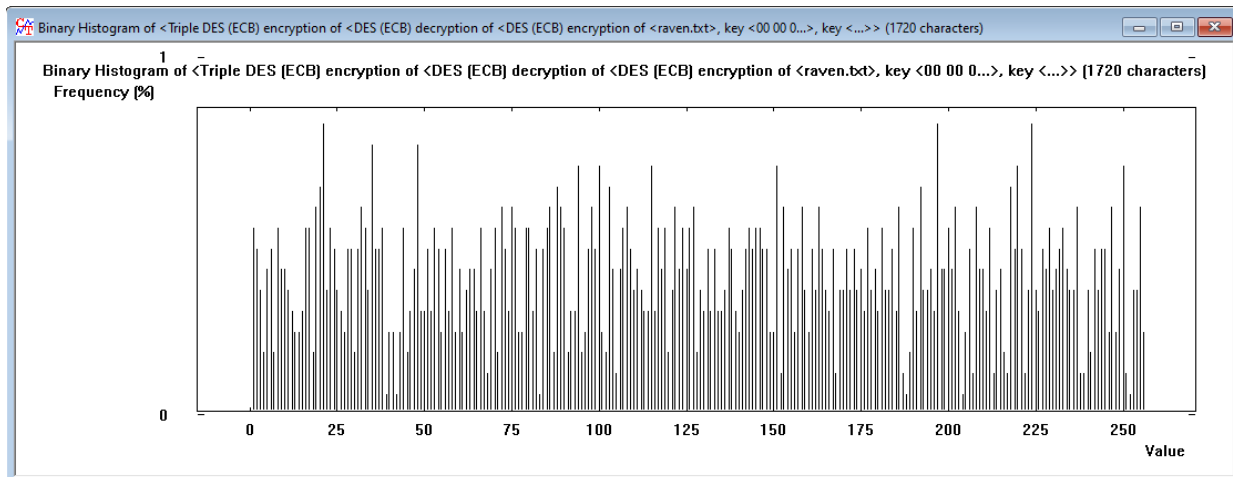
1. Выбраны первые пять параграфов из Raven, Edgar Allan Poe (1700 символов)
2. Создан бинарный файл с текстом

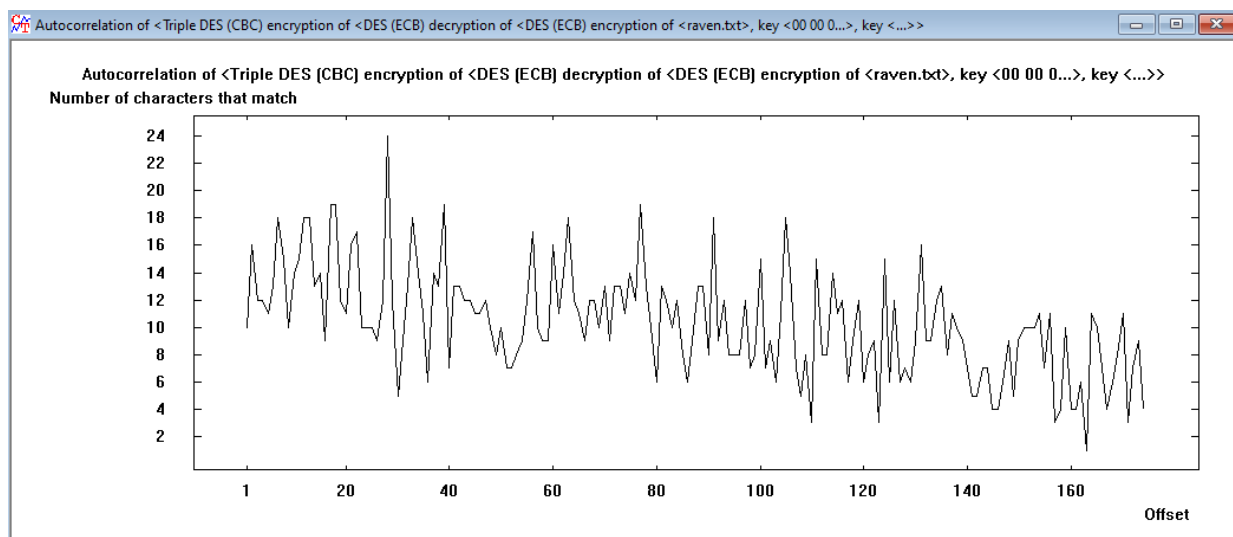
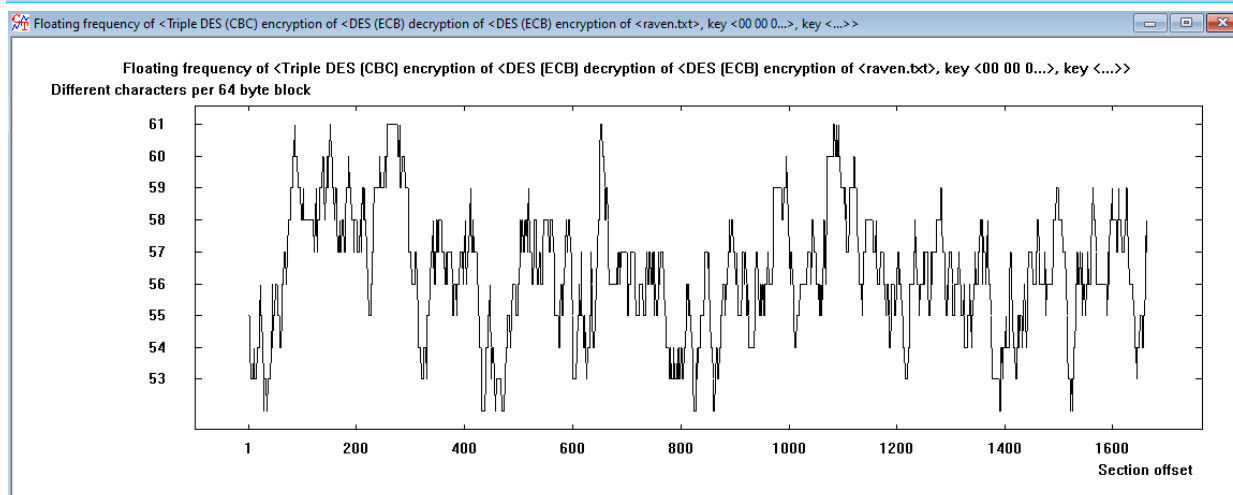
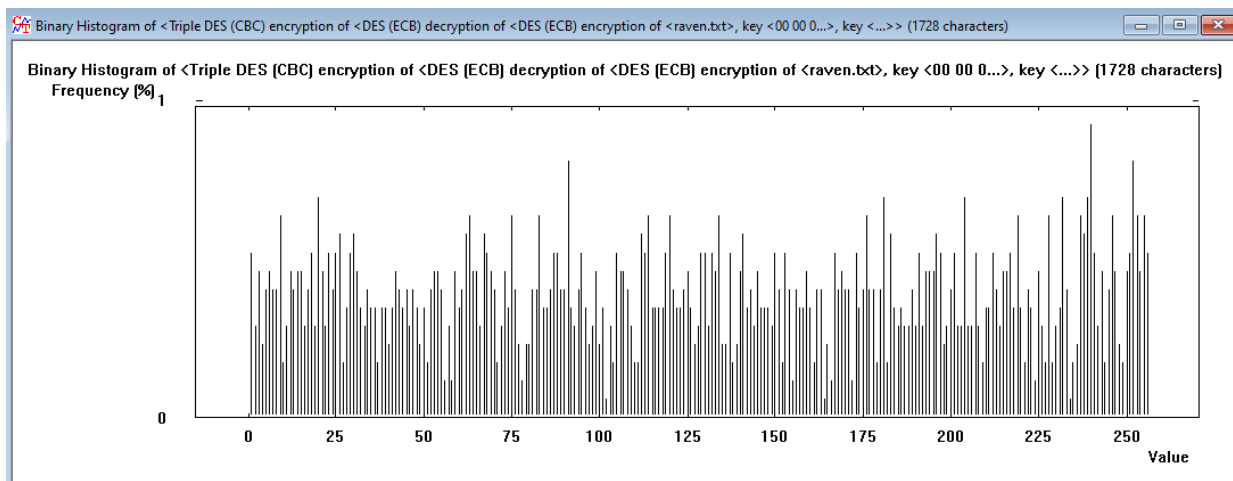




3. Текст зашифрован ECB с ключем DEADBEEFABCDEFABCDEF







5. 3-DES в CrypTool работает по схеме EDE2

DESL	9e+3
DESXL	3.2e+42

Выводы.

Был исследован шифр DES и его модификации: DES ECB, DES CBC, DESX, DESL, DESXL, 3DES (4 мод.).

1. Была изучена работа сетки Фейстеля и построенного на ней шифра DES, заключающегося на разделении информации на блоки по 64 бита. Проведена ручная зашифровка и расшифровка текста с фило из 8 символов (64 байта) ключем из 7. Результат расшифровки совпал с исходным текстом.

2. Были изучены режимы DES: ECB и CBC

В режиме ECB блоки по 64 бита зашифровываются независимо. Это видно на примере картинке: однотонным частям картинке соответствуют регулярные повторяющиеся паттерны в зашифрованной картинке

В режиме CBC зашифрованный блок хог-ится со следующим блоком, т.о. каждый блок зависит от всех предыдущих и повторяющихся паттернов не возникает. Это подтверждает несжимаемость результата при помощи zip.

Оценена сложность атаки на шифры грубой силой. Выяснено, что атака на CBC более затратна, чем ECB. Но оба шифра, без знания части ключа, не взламываются за разумное время.

3. Был исследована модификация 3-DES четырех видов(комбинации $2*2$):

1. EEE (трехкратное зашифрование)/EDE (за-, рас-, за- шифрование)

2. 3 — три различных ключа для шагов/ 2- один ключ для 1,3 шагов

Выяснено, что в стандартном инструменте СтурTool реализована схема EDE2 при помощи ручного воспроизведения трех этапов. Построены графики частотной и автокорреляционной характеристик

Исследовано время атаки грубой силой с зависимости от атаки грубой силой. Он возросло по сравнению с однократным DES, и так же превышает разумные пределы, что делает атаку грубой силой неэффективной

4. Исследованы модификации DESX, DESL, DESXL шифра DES

Модификация X заключается в xor-е значения аргумента и функции DES на константы, что увеличивает криптостойкость. При нулевых константах шифр сводится к простому DES.

Модификация L является оптимизацией шифра, путем исключения дополнительных шагов с перестановками, т. к. те не доавляют криптостойкости, и замены перестановки S с восемью чанками на один, но более стойкий.

Модификация XL является их комбинацией

Энтропия шифротекста для всех трех модификаций составила 7.88, по сравнению с 4.49 исходного текста.

5. При проверке времени атаки грубой силой обнаружено, что время атаки на модификации L на несколько порядков меньше, чем на остальные, а на XL меньше чем на X. Это говорит о том, что оптимизация L уменьшает криптостойкость шифра. Однако даже при нем без дополнительной информации о ключе атака займет количество лет в третьем порядке.