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

ОТЧЕТ

По лабораторной работе № 4
по дисциплине «Криптография и защита информации»
Тема: Исследование шифров DES, 3DES, Магма

Студент гр. 0303 _____ Болкунов В.О.

Преподаватель _____ Племянников А. К.

Санкт-Петербург
2023

Цель работы.

Цель работы: исследовать шифры DES, 3DES и Магма и получить практические навыки работы с ними, с использованием приложений Cryptool 1/2 и Литорея.

Порядок выполнения работы.

1. Изучить преобразования DES по шаблонной схеме DES Visualisation из CrypTool 2 с учетом рекомендаций Методического пособия (задание на с. 20)
2. Провести исследование DES в режимах работы ECB и CBC, используя CrypTool 1 и с учетом рекомендаций Методического пособия (задание на с. 22 - оценка трудоемкости атаки "грубой силы")
3. 3. Разработать схему в CrypTool 2 для экспериментального определения версии 3-DES.
4. Изучить преобразования шифра Магма с помощью приложения ЛИТОРЕЯ, с учетом рекомендаций Методического пособия (задание на с. 20)
5. Провести исследование шифра Магма в режимах работы простой замены и простой замены с зацеплением, используя приложение ЛИТОРЕЯ и с учетом рекомендаций Методического пособия (задание на с. 21 - шифрование изображения в разных режимах работы)

Выполнение работы.

1. Шифр DES

Были выбраны следующие исходные данные:

- Открытый текст M = “**bolkunov**”
- Ключ K = “**030304_o**”

Соответственно их байтовое представление:

- M_{16} = 62 6F 6C 6B 75 6E 6F 76
- K_{16} = 30 33 30 33 30 34 5F 6F

1.1 Ручные преобразования первого раунда

1.1.1. Раундовый ключ

Преобразуем ключ к бинарному виду:

K_2 = 00110000 00110011 00110000 00110011 00110000 00110100 01011111 01101111

Удаляем биты проверки (8, 16, 24, 32, 40, 48, 56, 64):

0011000~~0~~ 0011001~~1~~ 0011000~~0~~ 0011001~~1~~ 0011000~~0~~ 0011010~~0~~ 0101111~~1~~ 0110111~~1~~

← ← ←

Переставляем биты ключа в соответствии со следующей перестановкой

P_c = 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

P_d = 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

Получаем следующие половины преобразований ключа:

C = 00000000 11000000 10111111 0111

D = 11001010 11100000 11000000 1111

Циклически сдвигаем последовательности C, D на 1 бит влево, в соответствии с таблицей сдвигов DES:

C_{s1} = 00000001 10000001 01111110 1110

D_{s1} = 10010101 11000001 10000001 1111

Объединяем половины ключа с помощью P-блока сжатия:

P = 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

Итого ключ первого раунда:

K_1 = **00000000 00101100 11101100 01010100 01001001 11000111**

1.1.2. Начальное преобразование

Преобразуем блок сообщения к бинарному виду

$\mathbf{M_2}$ =

01100010 01101111 01101100 01101011 01110101 01101110 01101111 01110110

← ← ←

Выполним начальную перестановку блока в соответствии с таблицей IP DES

$\mathbf{P_{IP}}$ = 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

$\mathbf{M_{IP}}$ = 11111111 10010000 11110110 01011010 00000000 11111111 01101110 11101011

Получаем половины начального блока сообщения:

L_0 = **11111111 10010000 11110110 01011010**

R_0 = **00000000 11111111 01101110 11101011**

1.1.3. Раундовое преобразование

Выполним расширение блока R_0 в соответствии с перестановкой P-блока расширения:

P_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

R_{0e} = 10000000 00010111 11111110 10110101 11010111 01010110

Выполним побитовое XOR с раундовым ключом K_1:

$R_{0x} = R_{0e}$ ***xor*** $K_1 =$

100000 000011 101100 010010 111000 011001 111010 010001

Выполним преобразования S-блоков в соответствии с таблицами подстановок S-блоков.

1. R_{S1} = 100000 ; y = 2 ; x = 0 ; S_1 = 4 = **0100**
2. R_{S2} = 000011 ; y = 1 ; x = 1 ; S_1 = 13 = **1101**
3. R_{S3} = 101100 ; y = 2 ; x = 6 ; S_1 = 3 = **0011**
4. R_{S4} = 010010 ; y = 0 ; x = 9 ; S_1 = 2 = **0010**
5. R_{S5} = 111000 ; y = 2 ; x = 12 ; S_1 = 6 = **0110**
6. R_{S6} = 011001 ; y = 1 ; x = 12 ; S_1 = 0 = **0000**
7. R_{S7} = 111010 ; y = 2 ; x = 13 ; S_1 = 5 = **0101**
8. R_{S8} = 010001 ; y = 1 ; x = 8 ; S_1 = 12 = **1100**

Итого R_s = 01001101 00110010 01100000 01011100

Выполним преобразование прямого P-блока в соответствии с таблицей перестановок прямого P-блока

$P_s =$ 16 7 20 21 29 12 27 17 \
1 15 23 26 5 18 31 10 \
2 8 24 14 32 27 3 8 \
19 13 30 6 22 11 4 25

Итого ***f***(K_1, R_0) = 00001110 01011100 11000000 10110100

Для получения второго блока первого раунда сложим по модулю 2 результат раундовой функции ***f*** с L_0:

$R_1 = L_0$ ***xor f***(K_1, R_0) = **11110001 11001100 00110110 11101110**

$L_1 = R_0$ = **00000000 11111111 01101110 11101011**

1.2 Преобразования CrypTool

1.2.1 Раундовый ключ

В разделе PC1 выполняется начальная перестановка ключа (рис. 1)

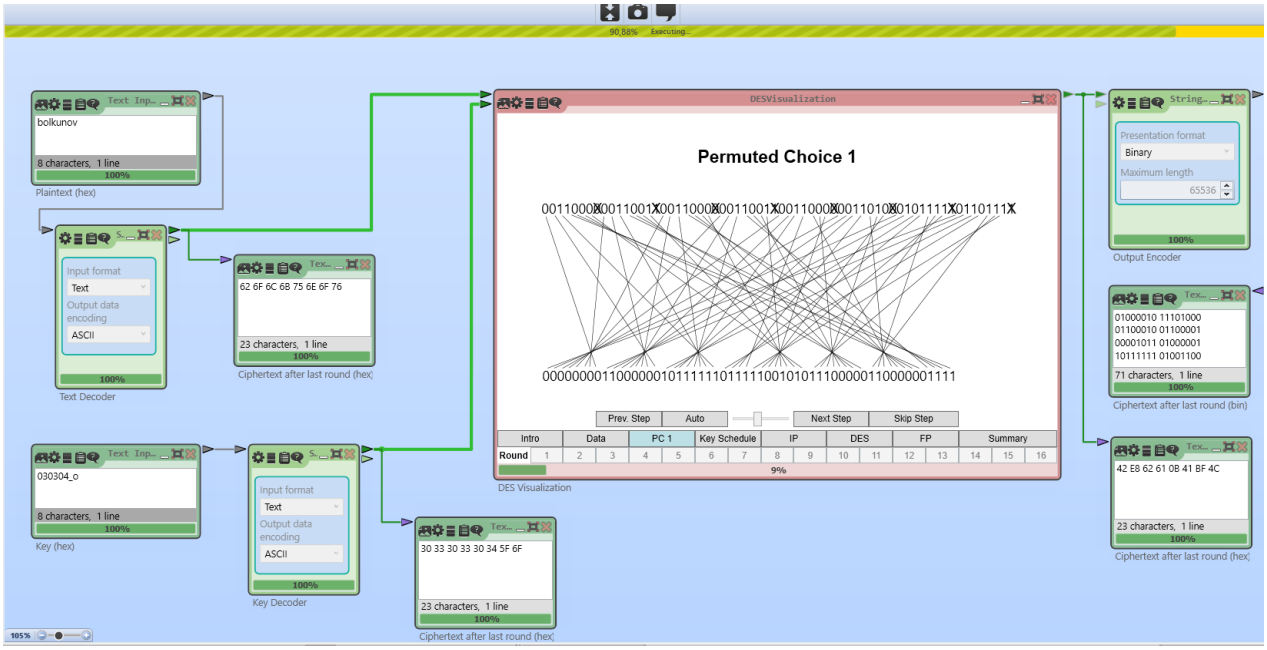

Рисунок 1: начальная перестановка ключа

Программа вычислила следующие половины ключа (рис 2):

$\mathbf{C_0}$ = 00000000 11000000 10111111 0111

$\mathbf{D_0}$ = 11001010 11100000 11000000 1111

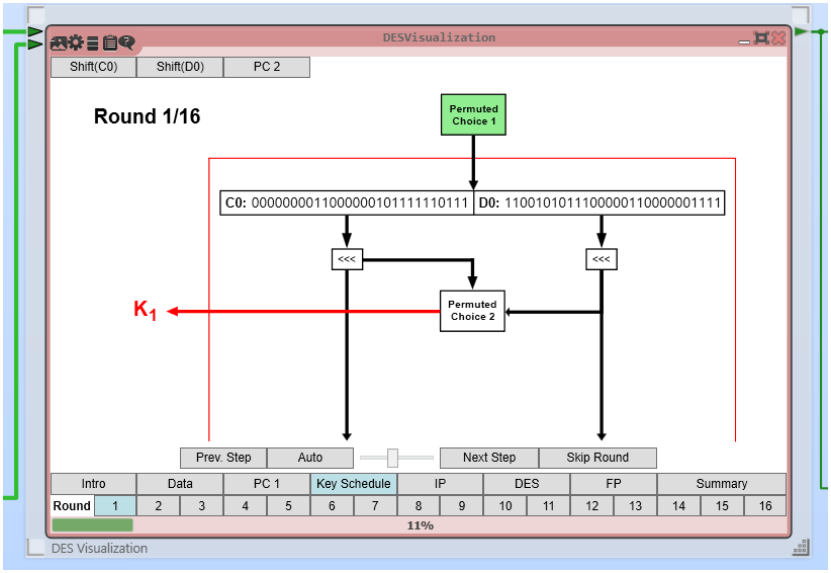

Рисунок 2: вычислены начальные половины ключа

Далее программа вычисляет циклические сдвиги для C_0 и D_0 (рис. 3-4), получая следующие значения:

$\mathbf{C_{s1}}$ = 00000001 10000001 01111110 1110

$\mathbf{D_{s1}}$ = 10010101 11000001 10000001 1111

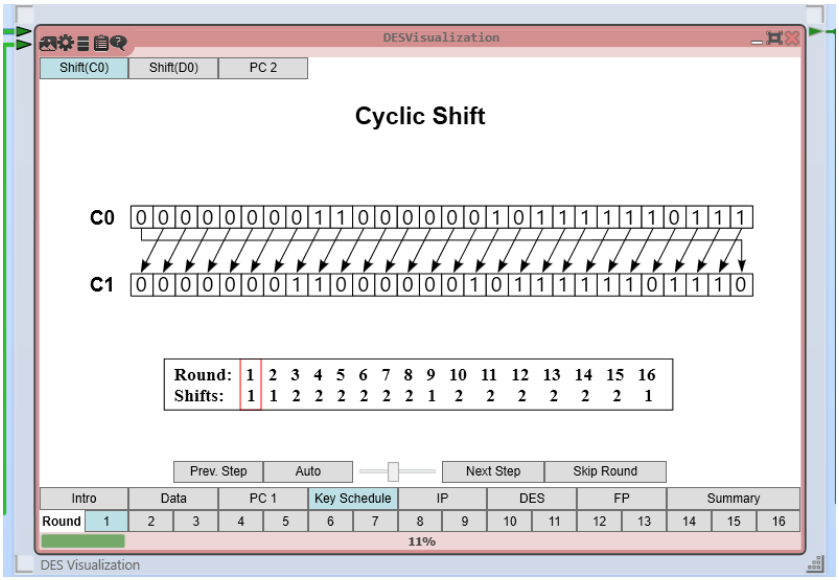

Рисунок 3: циклический сдвиг C_0

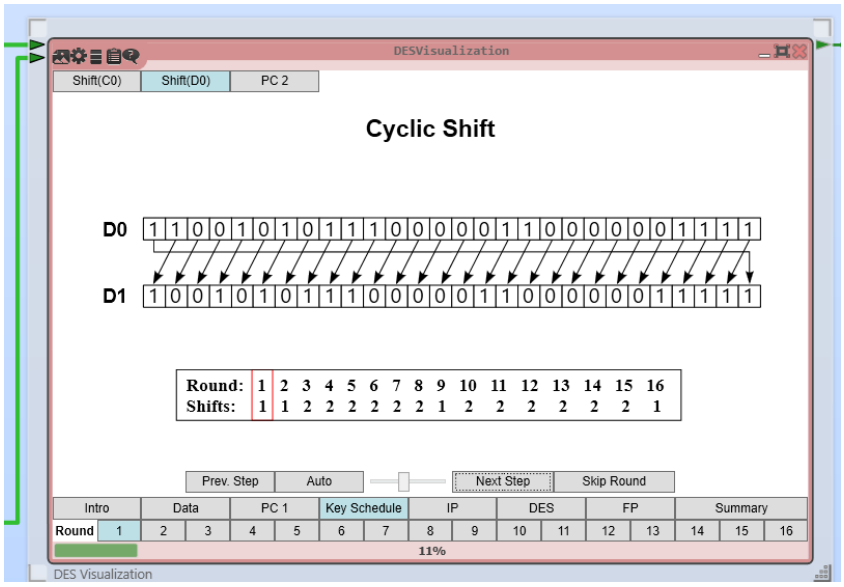

Рисунок 4: Циклический сдвиг D0

После чего программа применяет перестановку (P-блок сжатия), объединяя половины ключа (рис. 5)

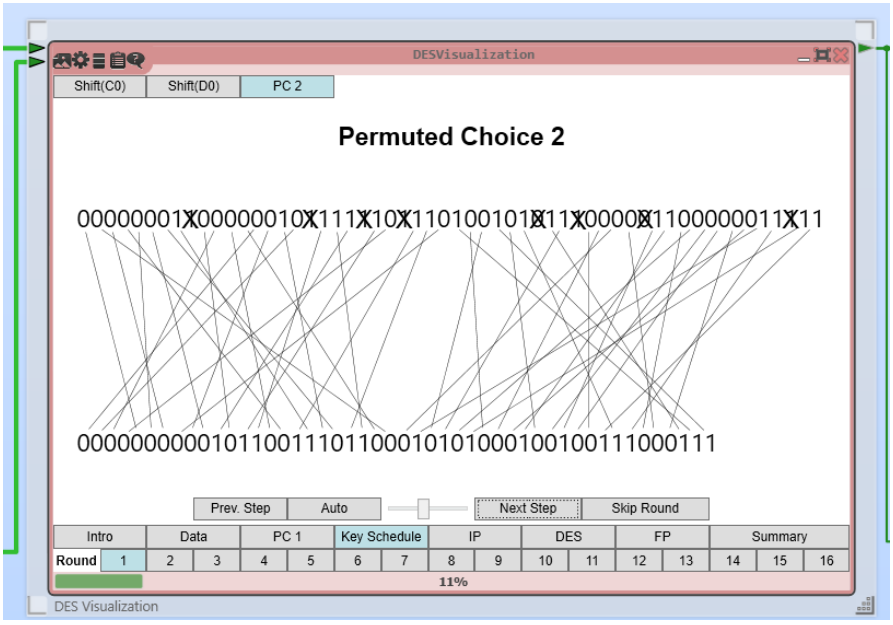

Рисунок 5: работа P-блока сжатия

Итого программа собирает ключ для первого раунда (рис. 6):

$\mathbf{K_1}$ = 00000000 00101100 11101100 01010100 01001001 11000111

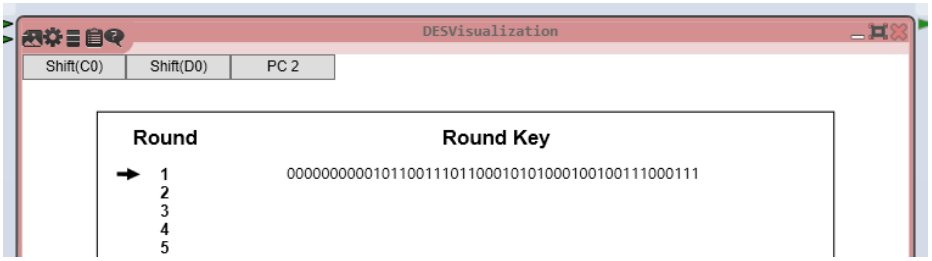

Рисунок 6: ключ первого раунда

1.2.2 Начальное преобразование

Программой было выполнено начальное преобразование блока сообщения (рис. 7-8).

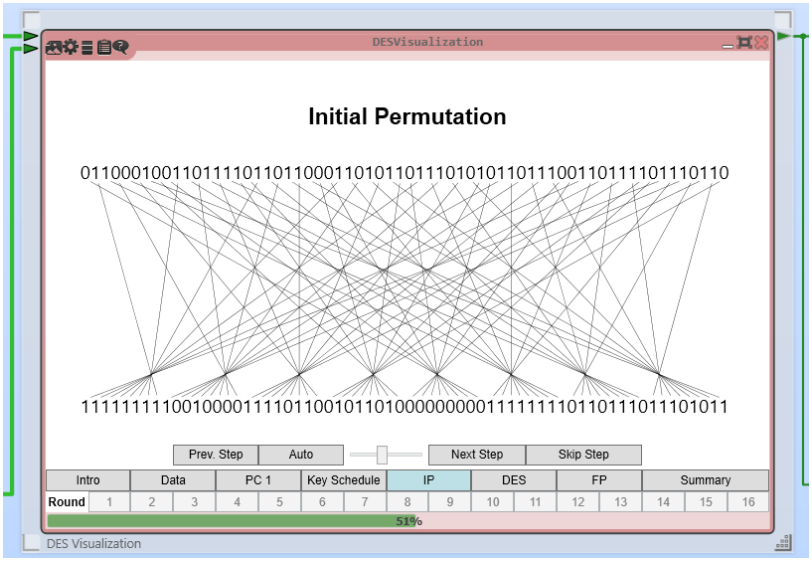

Рисунок 7: начальная перестановка (IP)

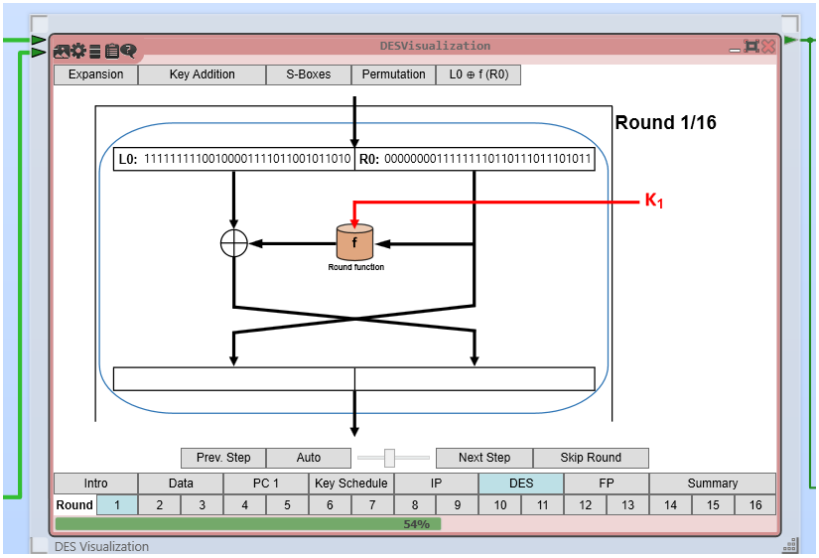

Рисунок 8: начальные половины блока сообщения

Полученные половины начального блока:

L_0 = **11111111 10010000 11110110 01011010**

R_0 = **00000000 11111111 01101110 11101011**

1.2.3 Раундовое преобразование

Программа выполнила начальное расширение блока R_0 (рис. 9)

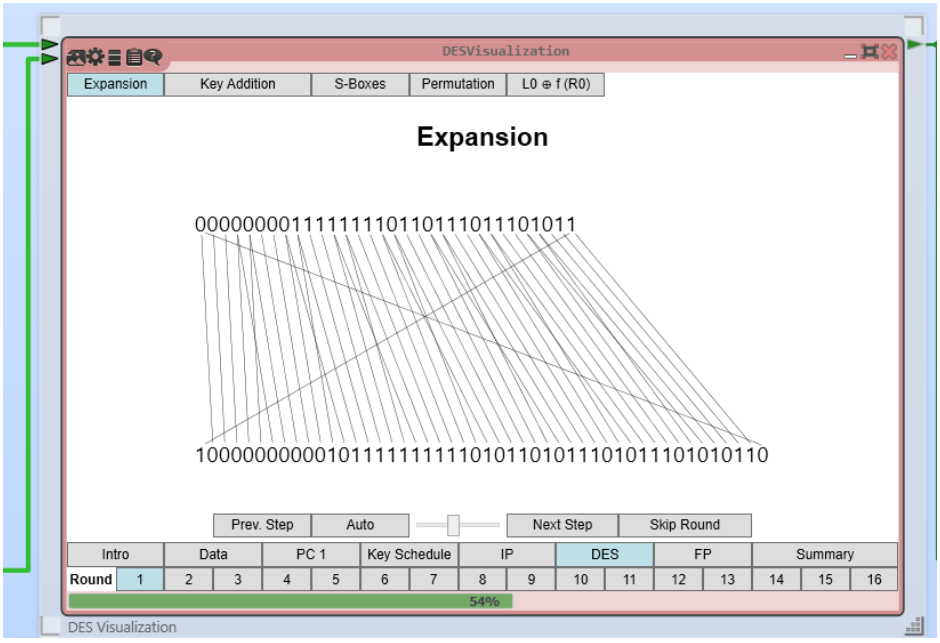

Рисунок 9: работа P-блока расширения

Получив: R_{0e} = **10000000 00010111 11111110 10110101 11010111 01010110**

Далее программа выполняет **xor** полученного 48-битного блока R_{0e} с раундовым ключом (рис. 10), получая на выходе:

R_{0x} = **100000 000011 101100 010010 111000 011001 111010 010001**

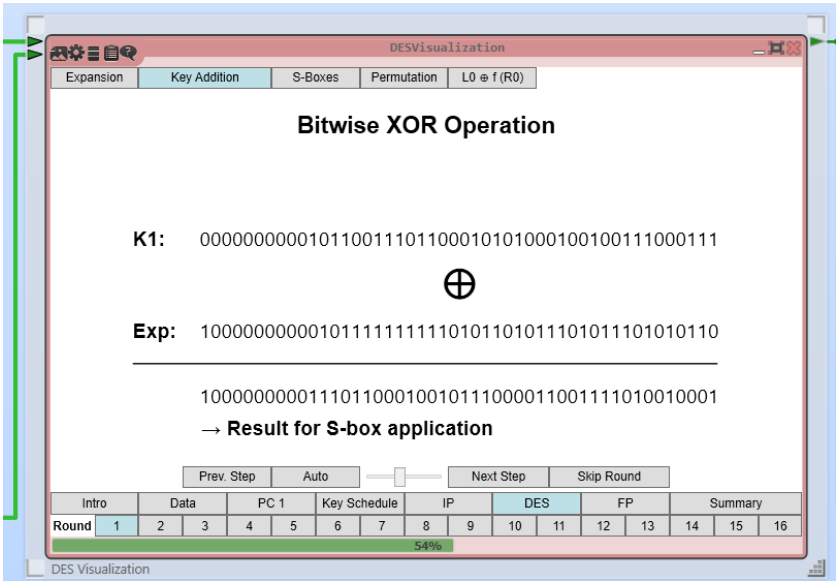

Рисунок 10: xor с раундовым ключом

Далее программа выполняет преобразования S-блоков (рис. 11)

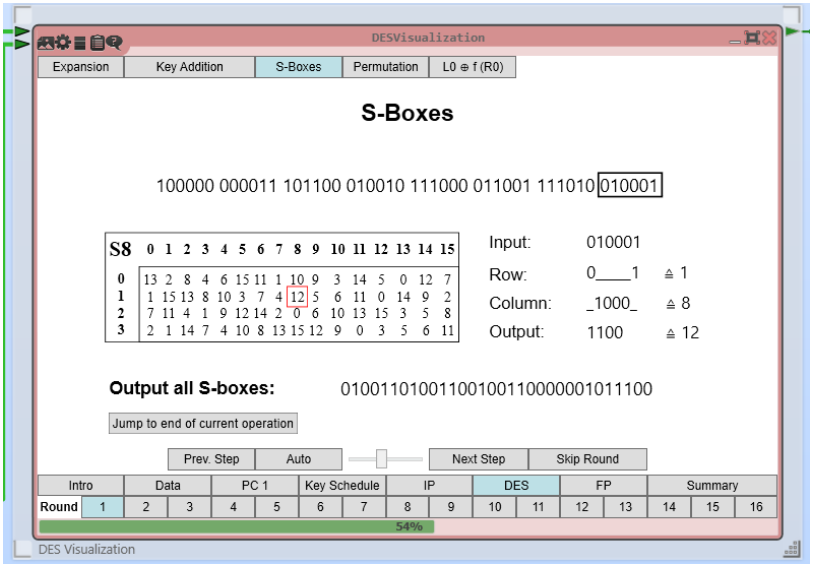

Рисунок 11: преобразования S-блоков

Получая R_s = **01001101 00110010 01100000 01011100**

После данной операции программа выполняет преобразование прямого P-блока (рис. 11)

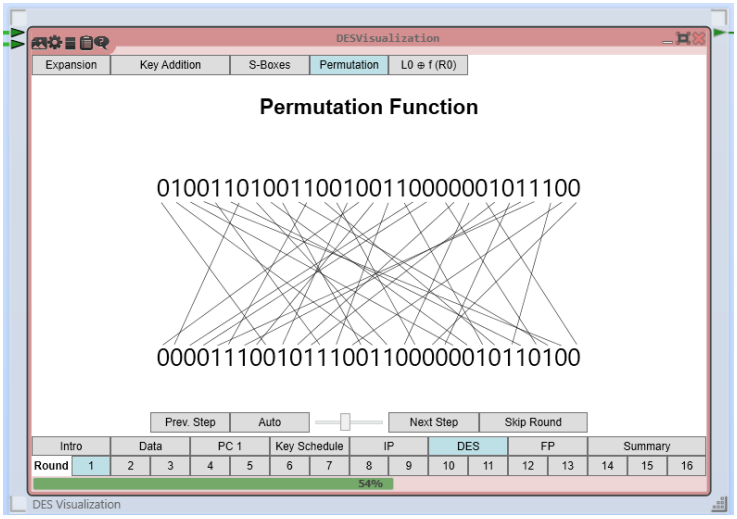

Рисунок 12: преобразование прямого P-блока

Получая значение $\boldsymbol{f}(K_1, R_0)$ = **00001110 01011100 11000000 10110100**

Для вычисления правой части блока первого раунда остаётся сложить по модулю 2 результат раундовой функции $\boldsymbol{f}$ с L_0 (рис. 13)

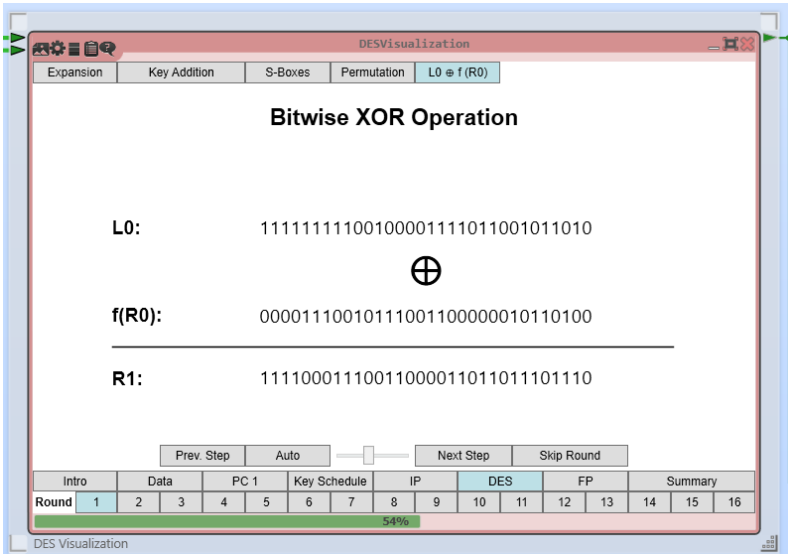

Рисунок 13: вычисление R1

Итого с помощью средств CrypTool 2 был проведён первый раунд шифрования блока DES (рис. 14):

R_1 = **11110001 11001100 00110110 11101110**

L_1 = **00000000 11111111 01101110 11101011**

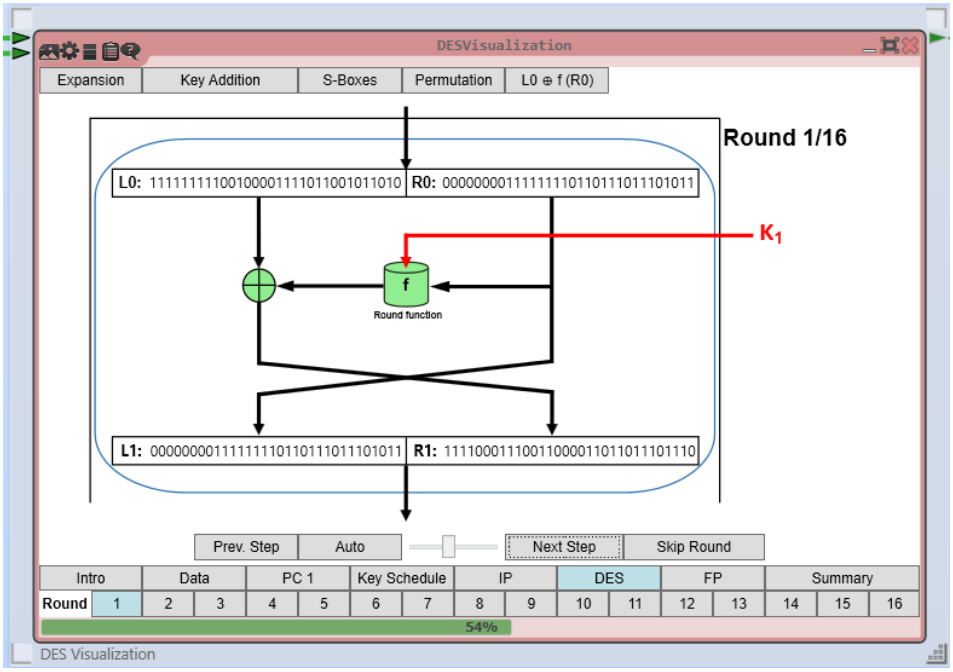

Рисунок 14: результат первого раунда

Результаты вычислений инструмента DES Visualizer полностью совпадают с полученными вручную результатами шифрования.

1.3 Обратное преобразование

Выполним обратное преобразование первого раунда, имея R_1 и L_1 и ключ первого раунда K_1

R_1 = 11110001 11001100 00110110 11101110

L_1 = 00000000 11111111 01101110 11101011

K_1 = 00000000 00101100 11101100 01010100 01001001 11000111

Очевидно, $R_0 = L_1$ и не требует вычислений.

Вычислим L_0. Для этого, зная значение блока R_0 найдём значение раундовой функции $\boldsymbol{f}(K_1, R_0)$ (п. 1.1.3.)

$\boldsymbol{f}(K_1, R_0)$ = **00001110 01011100 11000000 10110100**

Тогда: $R_1 = L_0$ ***xor*** $\boldsymbol{f}(K_1, R_0) \rightarrow L_0 = R_1$ ***xor*** $\boldsymbol{f}(K_1, R_0)$

Так как L_0 ***xor*** $\boldsymbol{f}(K_1, R_0)$ ***xor*** $\boldsymbol{f}(K_1, R_0) = L_0$ ***xor*** $0 = L_0$

Итого L_0 = **1111 1111 1001 0000 1111 0110 0101 1010**, что полностью совпадает с исходными данными.

2. Шифр DES: режимы работы

2.1. ECB

Было подготовлено и зашифровано с помощью средств CrypTool 1 достаточно большое текстовое сообщение в режиме ECB (рис. 15)

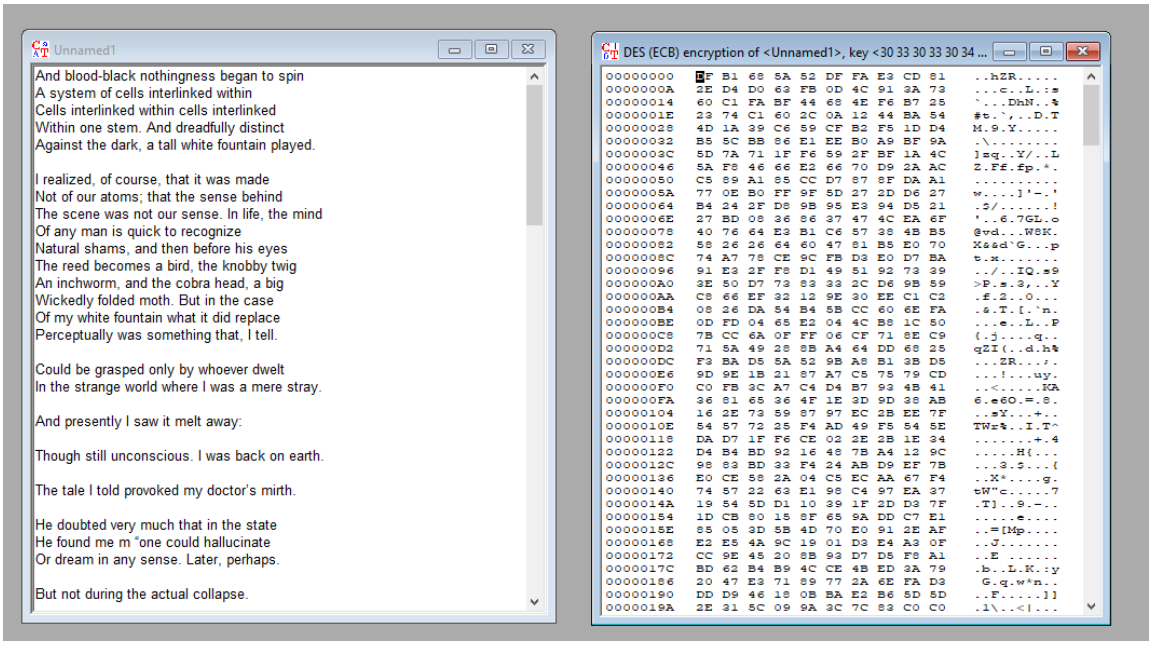

Рисунок 15: DES/ECB

После чего на данный шифр была проведена атака грубой силы, в таблице 1 представлены результаты оценки времени подбора ключа при разном количестве известных байтов ключа.

Таблица 1: результаты криптоанализа DES/ECB

N известных байтов	Оценка времени взлома
0	$\mathbf{1.6 * 10^4}$ ***лет***
1	$\mathbf{1.2 * 10^2}$ ***лет***
2	$\approx$ **360** ***дней***
3	**2.8** ***дней***
4	**33** ***минуты***
5	**15 секунд**
6	$\approx$ ***мгновенно***
7	$\approx$ ***мгновенно***

2.2. CBC

Аналогичным образом исходный текст был зашифрован в режиме CBC (рис. 16)

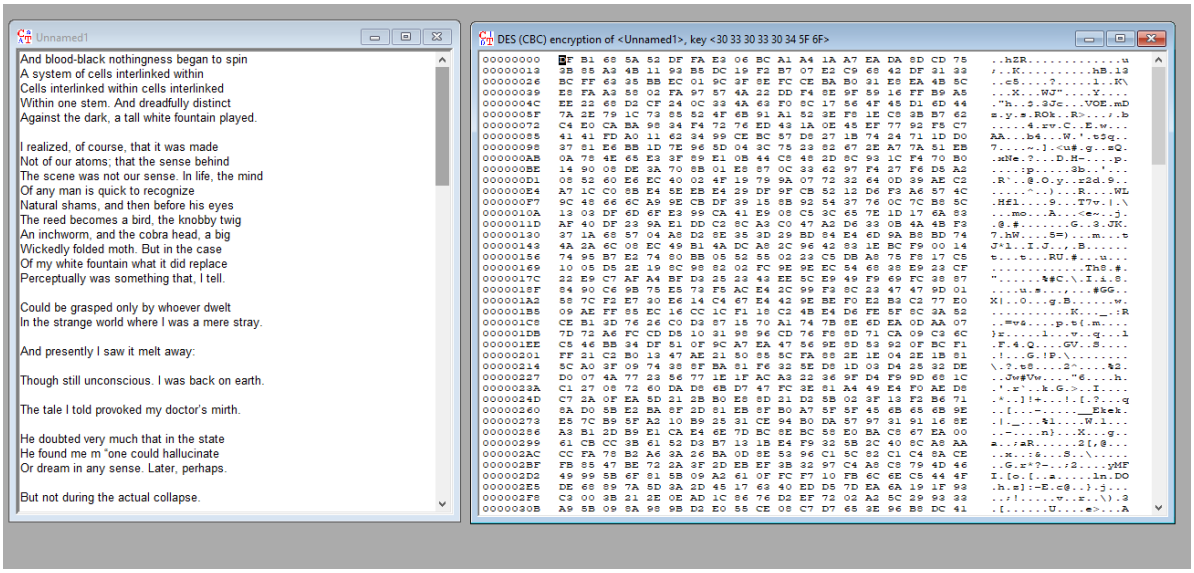

Рисунок 16: DES/CBC

После чего средствами CrypTool 1 был проведён криптоанализ для оценки времени подбора ключа при разных количествах известных байтов ключа, результаты представлены в таблице 2.

Таблица 2: результаты криптоанализа DES/CBC

N известных байтов	Оценка времени взлома
0	$\mathbf{2.6 * 10^4}$ ***лет***
1	$\mathbf{2 * 10^2}$ ***лет***
2	**1.6** лет
3	**4.6** ***дней***
4	**52** ***минуты***
5	**24 секунд**
6	***≈ мгновенно***
7	***≈ мгновенно***

Можно заметить, что в большинстве случаев атака на шифр DES в режиме CBC занимает больше времени, происходит это за счёт б**о**льшей энтропии (битовые последовательности более хаотичны).

3. Модификация 3-DES

Для определения версии шифра 3-DES используемого в среде CrypTool 2 были вручную построены следующие схемы шифрования 3-DES разными способами (EEE и EDE). Для 24-ёх и 16-ти байтного ключа схемы определения версии представлены соответственно на рисунках 17 и 18.

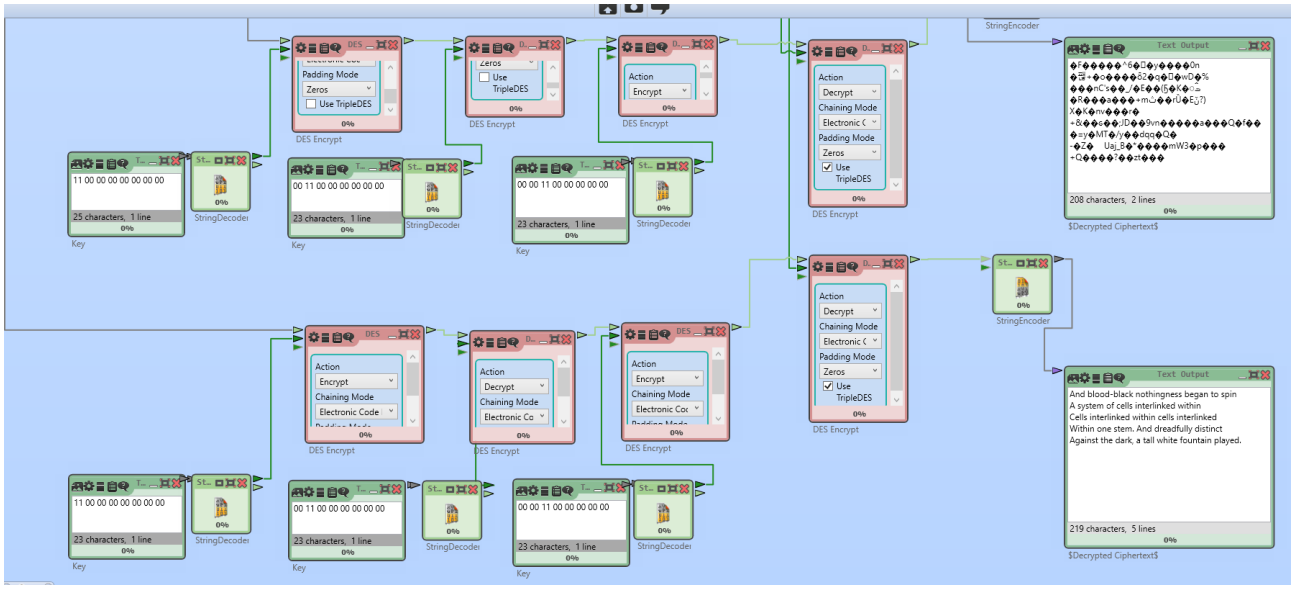

Рисунок 17: схема определения версии 3-DES для 24-ёх байтного ключа

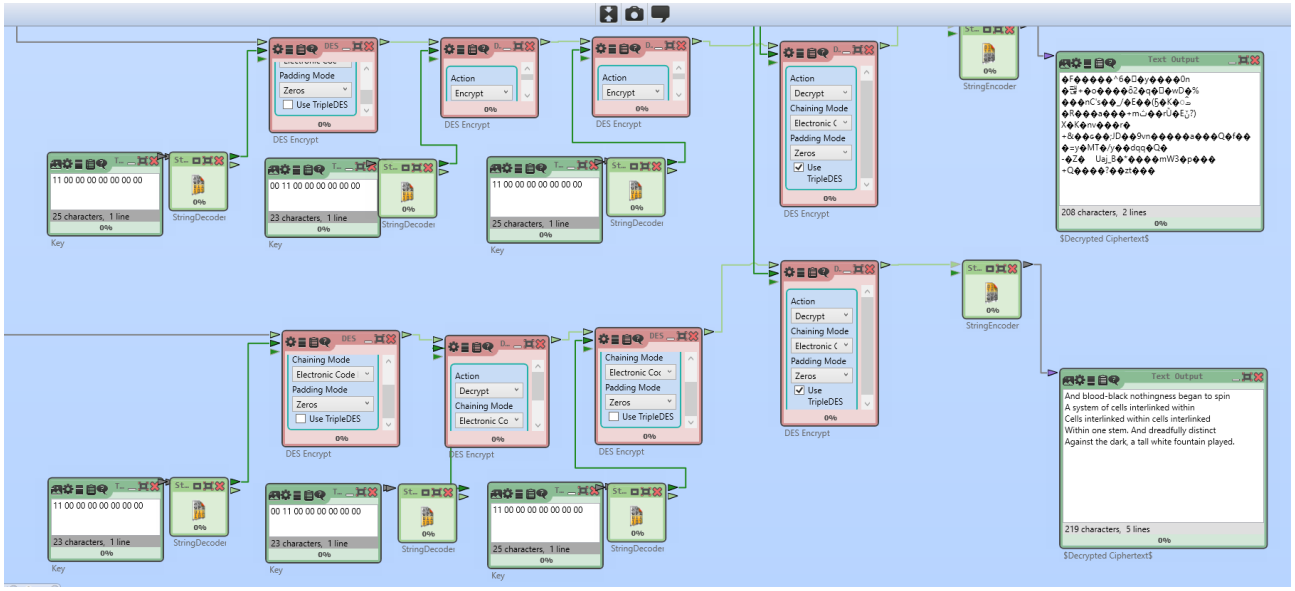

Рисунок 18: схема определения версии 3-DES для 16-ти байтного ключа

Как можно заметить сообщение зашифрованное DES-EDE3 и DES-EDE2 успешно расшифровалось стандартным инструментом расшифровки 3DES, что свидетельствует о том, что в CrypTool 2 используются данные версии шифра.

4. Шифр Магма

Были выбраны следующие исходные данные:

- Открытый текст M = “**Bolkunov**”
- Ключ K = “**030304O030304O030304O030304O0303**”

Соответственно их байтовое представление:

- M_{16} = 42 6F 6C 6B 75 6E 6F 76
- K_{16} = 30 33 30 33 30 34 4F \
  30 33 30 33 30 34 4F \
  30 33 30 33 30 34 4F \
  30 33 30 33 30 34 4F \
  30 33 30 33 30

4.1 Ручные преобразования первого раунда

Разбиваем ключ на 4-ёх-байтовые последовательности и берём первую в соответствии со схемой использования раундовых ключей; и получаем раундовый ключ первого раунда:

K_1 = **30 33 30 33**

Разобьём блок сообщения на две половины:

L_0 = 42 6F 6C 6B

R_0 = 75 6E 6F 76

Следующий шаг шифра: сложение R_0 с раундовым ключом:

$R_{0+} = R_0 + K_1 \bmod 2^{32}$ = **A5 A1 9F A9**

Далее осуществляем перестановки S-блоков в соответствии с таблицей S-блоков (рис. 19)

Номер S-блока	Значение															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	7	E	D	0	5	8	3	4	F	A	6	9	C	B	2
2	8	E	2	5	6	9	1	C	F	4	B	0	D	A	3	7
3	5	D	F	6	9	2	C	A	B	7	8	1	4	3	E	0
4	7	F	5	A	8	1	6	D	0	9	3	E	B	4	2	C
5	C	8	2	1	D	4	F	6	7	0	A	5	3	E	9	B
6	B	3	5	8	2	F	A	D	E	1	7	4	C	9	6	0
7	6	8	2	3	9	A	5	C	1	E	4	7	B	D	0	F
8	C	4	6	2	A	5	B	9	E	8	D	7	0	3	F	1

Рисунок 19: S-блоки магмы

1. A → A = 0001
2. 5 → 9 = 1101
3. A → 8 = 1100
4. 1 → F = 0111
5. 9 → 0 = 0100
6. F → 0 = 1110
7. A → 4 = 0001
8. 9 → 8 = 0010

Тогда R_{0s} = **A9 8F 00 48**

Следующим шагом осуществляем циклический сдвиг влево на 11 бит:

R_{0L} = 1010 1001 1000 1111 0000 0000 0100 1000 << 11

= 0111 1000 0000 0010 0100 0101 0100 1100

= **78 02 45 4C**

Чтобы получить R_1 остаётся только применить **xor** к полученному на предыдущем шаге R_{0L} с L_0 :

$R_1 = R_{0L}$ ***xor*** L_0 = 78 02 45 4C ***xor*** 42 6F 6C 6B = **3A 6D 29 27**

$L_1 = R_0$ = **75 6E 6F 76**

4.2 Преобразования LitoreR

В программу Литорея были введены индентичные исходные данные (рис. 20)

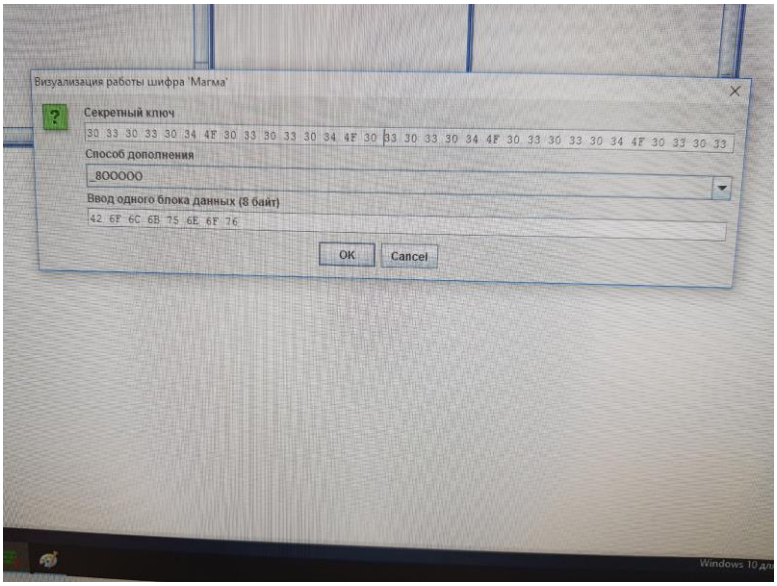

Рисунок 20: ввод исходных данных Магма

На рисунке 21 представлены результаты работы программы на первом раунде. Получены следующие данные:

Раундовый ключ K_1 = **30 33 30 33**

Субблоки: L_0 = 42 6F 6C 6B ; R_0 = **75 6E 6F 76**

Сложение по модулю 2: R_{0+} = **A5 A1 9F A9**

Подстановки s-блоков: R_{0s} = **A9 8F 00 48**

Циклический сдвиг: R_{0L} = **42 6F 6C 6B***

Сложение ***xor***: R_1 = **3A 6D 29 27**

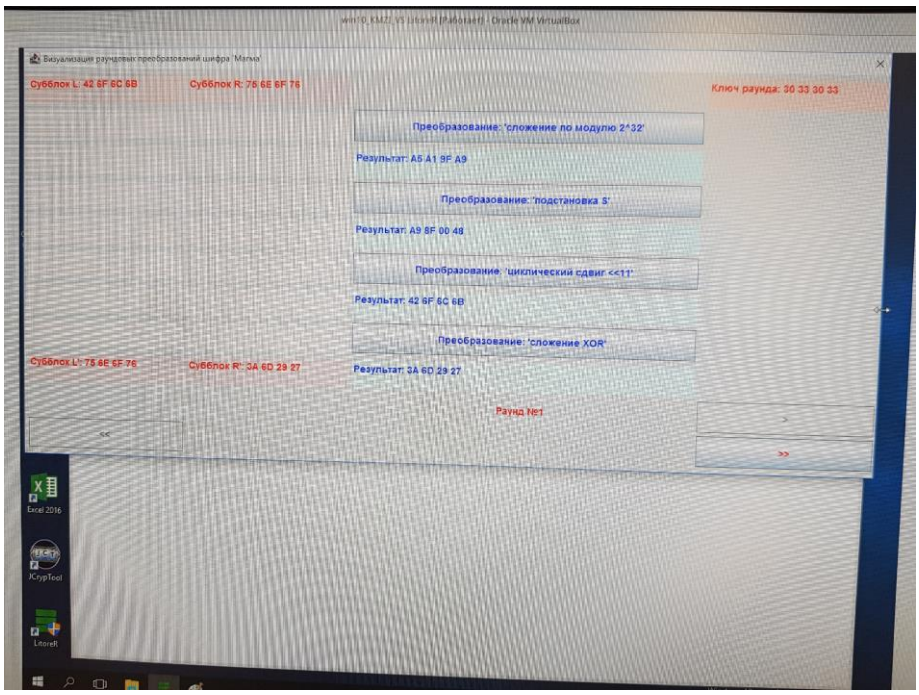

Рисунок 21: работа шифра Магма на первом раунде

Полученные значения совпадают* с вычисленными вручную.

4.3 Обратное преобразование

Выполним обратное преобразование первого раунда, имея R_1 и L_1 и ключ первого раунда K_1

R_1 = 3A 6D 29 27

L_1 = 75 6E 6F 76

K_1 = 30 33 30 33

Очевидно, $R_0 = L_1$ и не требует вычислений.

Вычислим L_0. Для этого выполним первые три шага шифрования зная R_0 (пункт 4.1)

Сложение по модулю 2: R_{0+} = **A5 A1 9F A9**

Подстановки s-блоков: R_{0s} = **A9 8F 00 48**

Циклический сдвиг: R_{0L} = **78 02 45 4C**

Выполним сложение R_1 **xor** с R_{0L} чтобы найти L_0

Так как $R_1 = R_{0L}$ ***xor*** $L_0 \rightarrow L_0$ ***xor*** R_{0L} ***xor*** $R_{0L} = L_0 = R_1$ ***xor*** R_{0L}

L_0 = 3A 6D 29 27 **xor** 78 02 45 4C = **42 6F 6C 6B**

Полученные расшифрованные субблоки полностью совпадают с исходными.

5. Шифр Магма: режимы работы

Было подготовлено следующее изображение, содержащие буквы и цифры в формате bmp (рис. 22):

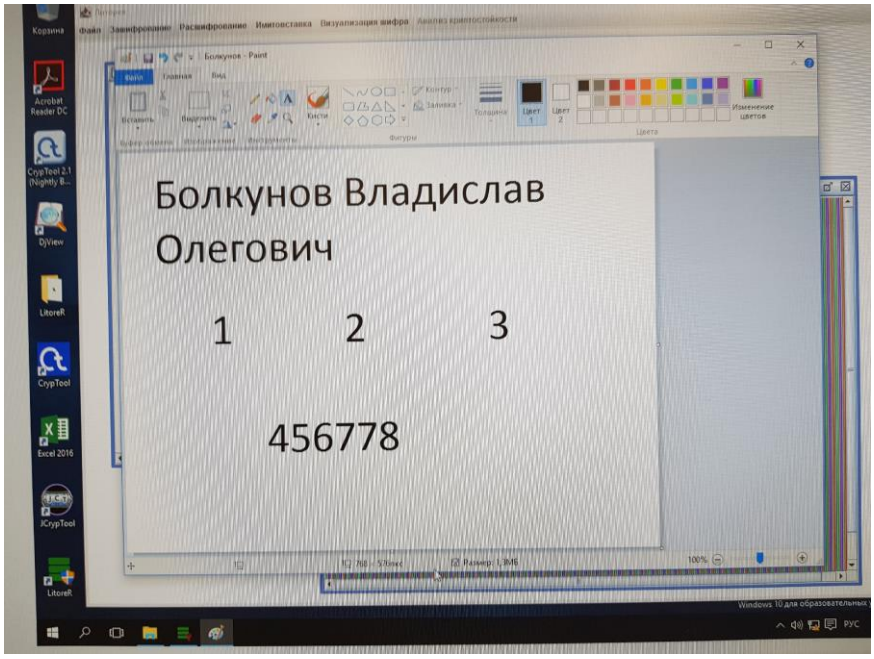

Рисунок 22: исходное изображение

5.1 Режим простой замены

Исходное изображение было зашифровано в программе Литорея с использованием режима простой замены (рис. 23)

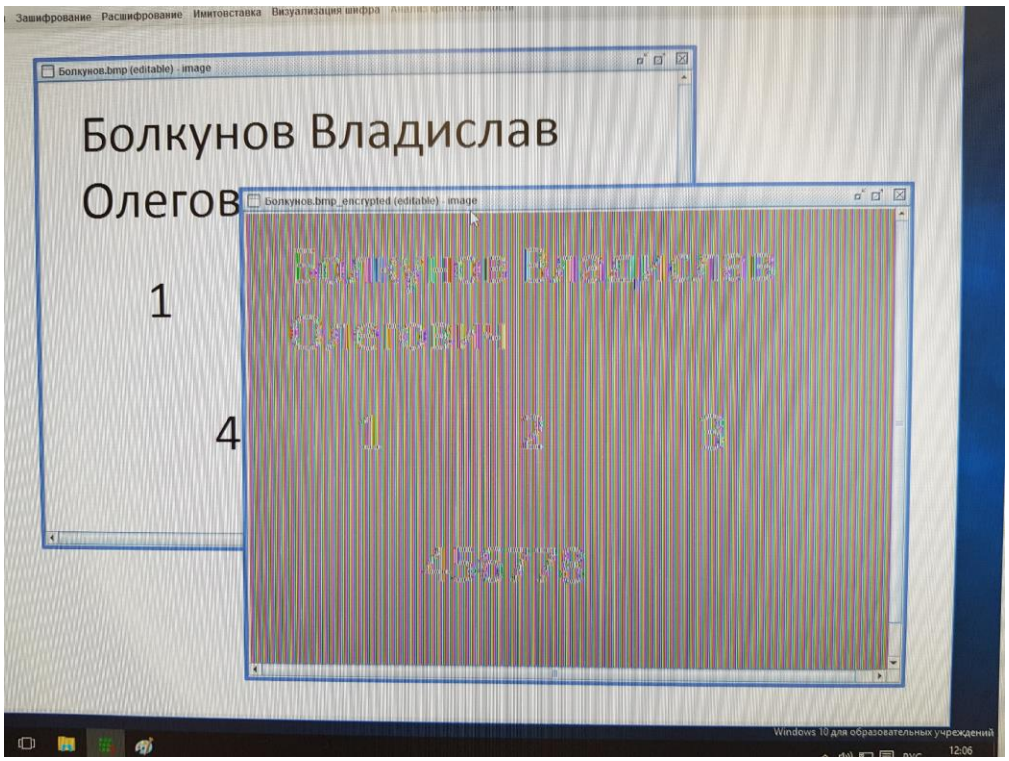

Рисунок 23: магма: режим простой замены

Как можно заметить текст на изображении различим.

Полученное изображение было сжато средствами CrypTool 1.

Результаты сжатия представлены на рисунке 24. Коэффициент сжатия составил 0.2 (20%)

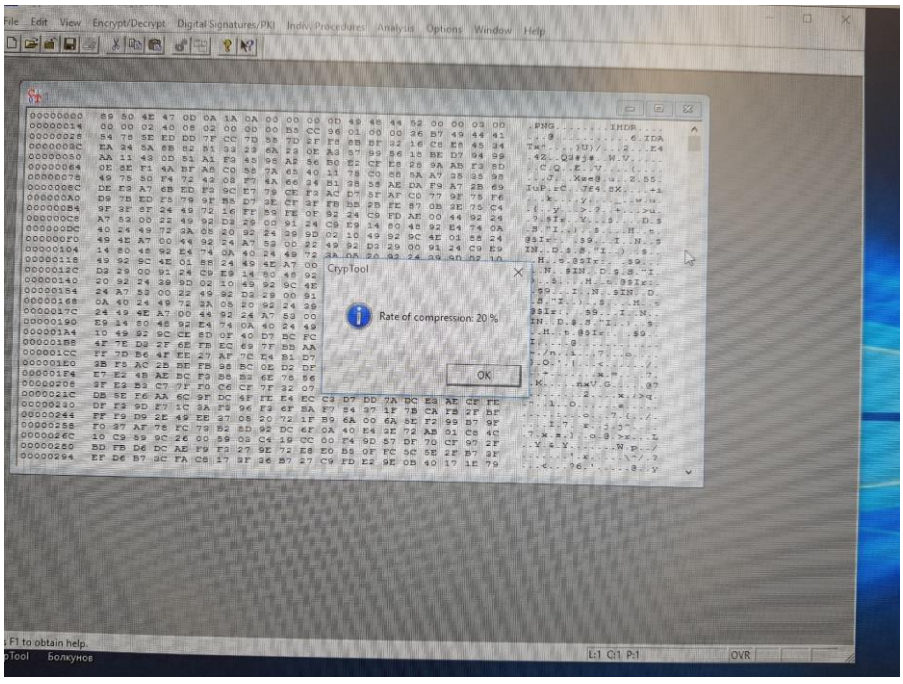

Рисунок 24: сжатие изображения зашифрованного в режиме простой замены

5.2 Режим простой замены с зацеплением

Идентичное изображение было зашифровано в режиме простой замены с зацеплением. Результат шифрования представлен на рисунке 25.

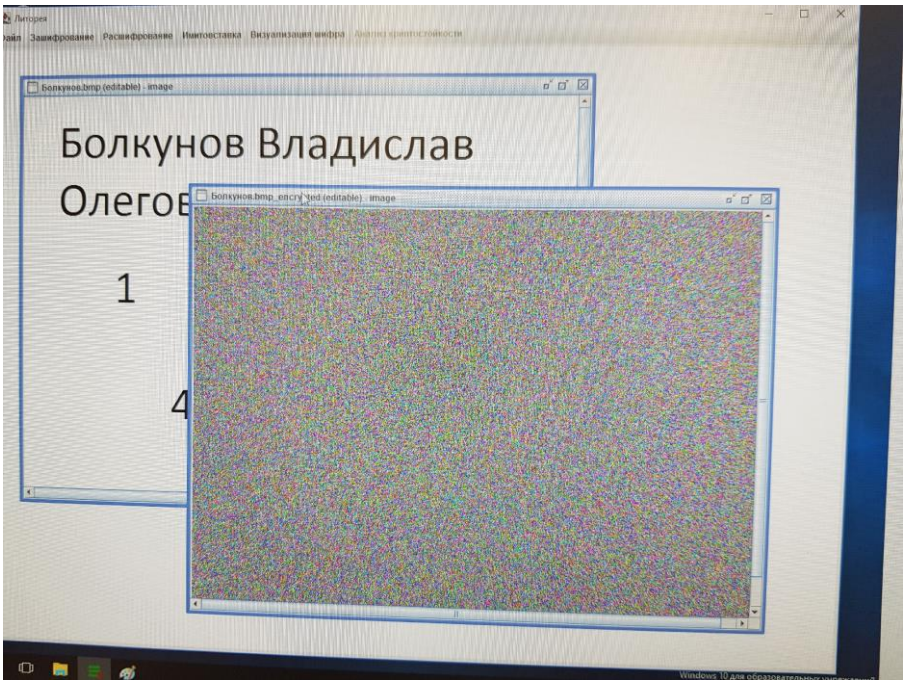

Рисунок 25: шифрование изображения в режиме простой замены с зацеплением

Полученное изображение было аналогично сжато средствами CrypTool 1 (рис. 26). В результате коэффициент сжатия составил 0 (0%), что говорит о крайне высокой энтропии зашифрованных данных.

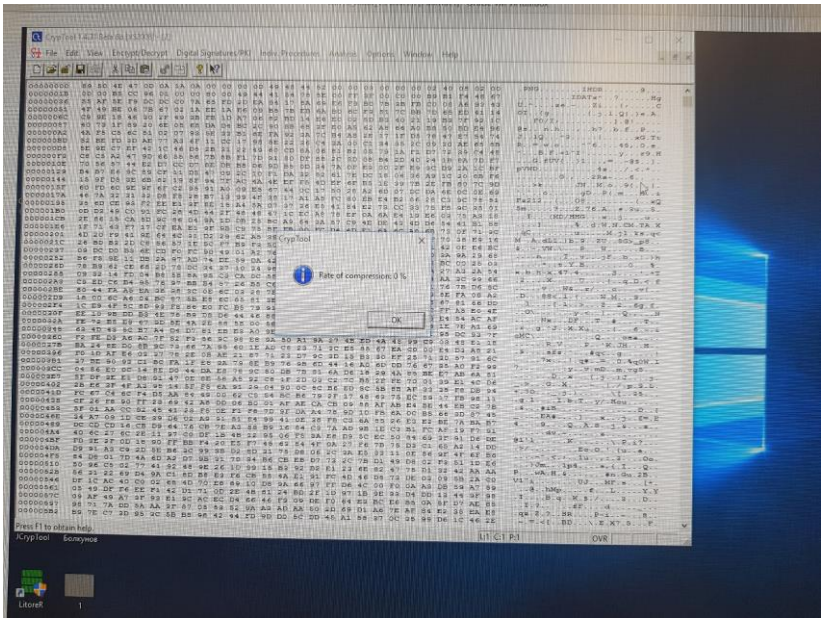

Рисунок 26: сжатие изображения зашифрованного в режиме простой замены с зацеплением

Выводы:

В ходе лабораторной работы были исследованы шифры DES, 3-DES и Магма, и режимы их работы: ECB / CBC и простая замена / простая замена с зацеплением соответственно для DES и Магма.

- Для шифра DES были изучены преобразования в среде CrypTool 2.

Результаты работы шифра на первом раунде были успешно сопоставлены с ручными вычислениями результатов первого раунда шифрования.

- На основе выбранного текста была изучена работа режимов шифра

DES: ECB и CBC в среде CrypTool. Для результатов шифрования была проведена оценка атаки грубой силы средствами программы CrypTool.
В среднем оценка времени атаки оказалась больше для режима CBC, что объясняется значительно более высокой энтропией по сравнению с режимом ECB.

- В среде CrypTool 2 была разработана схема для экспериментального

определения версии шифра 3-DES: вручную были созданы 4 версии данного шифра (EEE2, EDE2, EEE3 и EDE3); результаты их шифрования были сопоставлены со встроенным шифром 3-DES. По итогам эксперимента было выявлено что в данной программе используется версия EDE2 и EDE3 для 16-байтового и для 24-байтового ключа соответственно.

- Для шифра Магма были изучены преобразования в программе

Литорея. Результаты работы шифра на первом раунде были успешно сопоставлены с самостоятельными вычислениями значений первого раунда.

- В программе Литорея были исследованы режимы работы шифра магма. Было подготовлено изображение, содержащее символы, после чего зашифровано в режимах простой замены и простой замены с зацеплением. После шифрования изображения были сжаты средствами программы CrypTool. В результате степень сжатия файла, зашифрованного в режиме простой замены, оказалась значительно выше, чем для файла, зашифрованного в режиме простой замены с зацеплением, также в случае простой замены символы на изображении были достаточно хорошо различимы, в отличие от режима простой замены с зацеплением; что говорит о значительно более высокой энтропии в данных после шифрования в режиме простой замены с зацеплением по сравнению с режимом простой замены.

ПРИЛОЖЕНИЕ А. ПРИМЕЧАНИЯ.

* - в программе Литорея некорректно отображается результат циклического сдвига (вместо него показывается левый субблок L_0).