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

ОТЧЕТ

по лабораторной работе №5

“Изучение шифра AES”

Студент гр. 8382 Мирончик П.Д.

Преподаватель Племянников А.К.

Санкт-Петербург

2021

ЦЕЛЬ РАБОТЫ

Исследовать характеристики шифра AES и финалистов конкурса AES, а также изучить атаку предсказанием дополнения и получить практические навыки работы с шифрами и проведения атаки, в том числе с использованием приложения Cryptool 1 и 2.

1. ИССЛЕДОВАНИЕ ПРЕОБРАЗОВАНИЙ AES

1.1. Задание

1. Изучить преобразования шифра AES с помощью демонстрационного приложения из Cryptool 1: *Indiv.Procedures->Visualization…->AES->Rijndael Animation.*

2. Выполнить вручную преобразования для одного раунда и вычисление раундового ключа при следующих исходных данных:

a. Открытый текст – фамилия_имя (транслитерация латиницей)

b. Ключ – номер группы_отчество

3. Проверить полученные результаты с помощью приложения-инспектора: *Indiv.Procedures->Visualization…->AES->Rijndael Inspector.*

4. Провести наблюдения в потоковой модели шифра AES с помощью демонстрационного приложения из CrypTool 1 для 0-текста и 0-ключа: *Indiv.Procedures->Visualization…->AES->Rijndael Flow Visualisation*

1.2. Описание AES

Шифр AES (Rijndael) работает на основе перестановочно-подстановочной сети (SP-сеть). Обобщенная схема работы алгоритма представлена на рисунке 1.1.

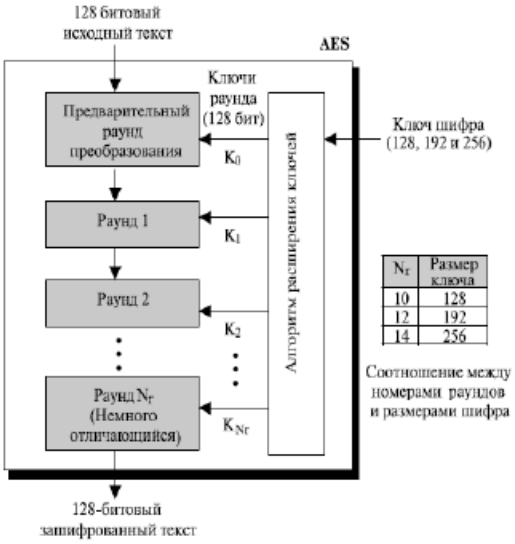

Рис.1.1 – Схема AES

В версии с наименьшей длиной ключа алгоритм AES получает на вход блок открытого текста размером 16 байт и 16 байт ключа. Значения блока записывается в столбцы матрицы состояний размером 4x4 байт.

Процедура расширения ключей ExpandKey создает последовательно (слово за словом) 128 битные раундовые ключи от единственного входного ключа шифра.

После того, как сформированы раундовые ключи, начинается раундовая обработка матрицы состояний. В каждом раунде алгоритма выполняются следующие преобразования, представленные на рисунке 1.2:

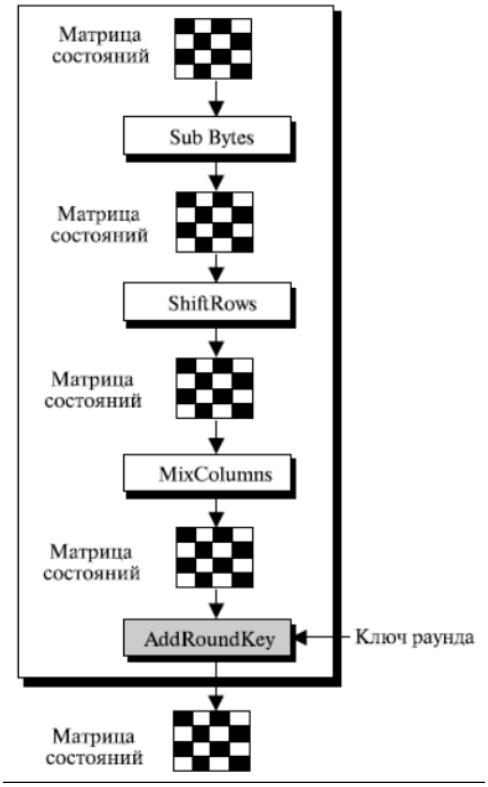

Рис.1.2 – Раундовые преобразования

1. Столбцы матрицы состояний складываются с ключом шифра операцией xor.

2. Полученная матрица состояний проходит через преобразование подстановки SubBytes.

3. Циклический сдвиг влево всех строк матрицы состояний выполняется преобразованием ShiftRows .

4. Смешивание столбцов матрицы состояний путем ее умножения на матрицу констант в конечном поле $GF(2^8)$ выполняет преобразовние MixColumn, а сложение полученных столбцов матрицы состояний с раундовым ключом операцией xor – преобразование AddRoundKey

5. Действия 2-4 повторяются в каждом раунде за исключение последнего.

6. Последний раунд не включает в себя смешивание столбцов.

Расшифровывание выполняется применением обратных операций и раундовых ключей в обратной последовательности.

Для генерации раундовых ключей будем использовать структуру, изображенную на рис. 1.3:

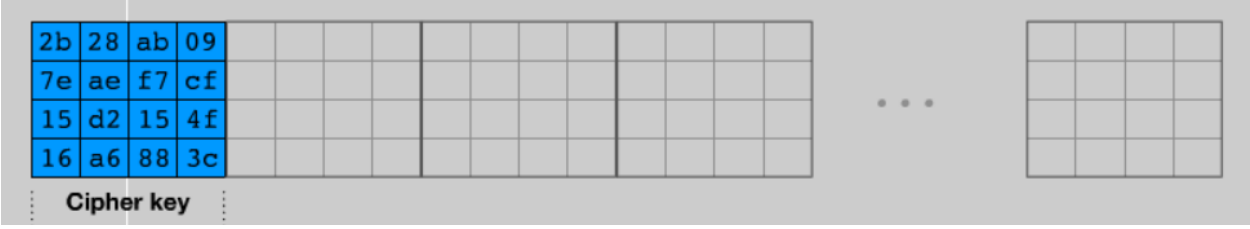

Рис.1.3 – Заполнение раундовых ключей

1. В первый блок заносим ключ шифрования.

2. Для первого слова W_i в следующем блоке выполняем следующие операции:

$$W_i = W_{i-1}$$
$$W_i = RotWord(W_i)$$
$$W_i = SubBytes(W_i)$$
$$W_i = W_i \oplus W_{i-4} \oplus Rcon_{\frac{i}{4}}$$

где *i* – номер слова во всей структуре (рис. 1.3), *RotWord* – функция, выполняющая сдвиг слова на один байт влево, *SubBytes* – функция, аналогичная описанной в п.2 раундовых преобразований, *Rcon* – массив из 10 константных слов.

3. Для второго, третьего и четвертого слова в том же блоке поочередно выполняем следующие операции:

$$W_i = W_{i-1} \oplus W_{i-4}$$

4. Пункты 2-3 выполняем поочередно для всех оставшихся блоков.

1.3. Расчет матрицы состояний и раундового ключа шифра для одного раунда.

Выполним преобразования для открытого текста mironchik_pavel и ключа 8382_denisovich.

Байтовое представление текста:

```
6d 6e 6b 76
69 63 5f 65
72 68 70 6c
```

```
6f 69 61 00
```

Байтовое представление ключа:

```
68 5f 69 69
33 64 73 63
38 65 6f 68
32 6e 76 00
```

Далее выполним расчет первого раундового ключа:

1. Заполнение первого слова:

$$W_i = W_{i-1} = (69, 63, 68, 00)$$

$$W_i = RotWord(W_i) = (63, 68, 00, 69)$$

$$W_i = SubBytes(W_i) = (fb, 45, 63, f9)$$

$$W_i = W_i \oplus W_{i-4} \oplus Rcon_{\frac{i}{4}} = (92, 76, 5b, cb)$$

2. Заполнение оставшихся трех слов:

$$(cb, 12, 3e, a5), (a4, 61, 51, d3), (cd, 02, 39, d3)$$

В результате ключ первого раунда:

```
92 cd a4 cd
76 12 61 02
5b 3e 51 39
cb a5 d3 d3
```

И выполним преобразование исходного текста до первого раунда включительно:

После SybBytes:

```
6b c7 77 c0
be c5 71 6f
d6 d7 c0 f2
4c c5 f0 63
```

После MixColumns:

```
21 b8 4c 65
c2 64 0d 35
90 9d fb c0
7e 49 b1 c9
```

После сложения с раундовым ключем:

```
b3 75 e8 a8
b4 76 6c 37
cb a3 aa f9
b5 ec 62 1a
```

1.4. Скриншоты приложения-инспектора, подтверждающие корректность расчетов.

Проверим вычисления пункта 1.3 с помощью CrypTool (рис. 1.4):

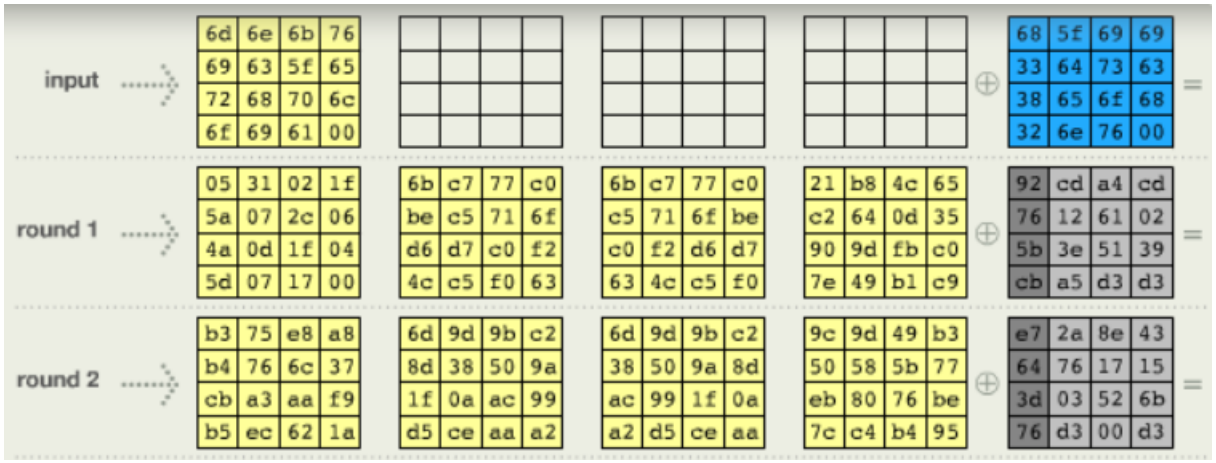

Рис. 1.4 – Визуализация AES в CrypTool

Видно, что вычисления для первого раунда сошлись.

2. ИССЛЕДОВАНИЕ ФИНАЛИСТОВ КОНКУРСА AES

2.1. Задание

1. Выбрать текст на английском языке (не более 120 знаков).

2. Создать бинарный файл с этим текстом, зашифровав и расшифровав его шифром AES на 0-м ключе.

3. С помощью Cryptool 1 зашифровать с ключом отличным от 0 текст с использованием шифров AES, MARS, RC6, Serpent и Twofish.

4. Приложением из Cryptool 1 вычислить энтропию исходного текста и шифротекстов, полученных в итоге. Зафиксировать результаты измерений в таблице.

5. Приложением из Cryptool 1 оцените время проведения атаки «грубой силы» всех шифров для одного и того же шифротекста в случаях, когда известно n-2, n-4, n-6,..., 2 байт секретного ключа. Зафиксировать результаты измерений в таблице.

2.1. Исходные данные для экспериментов.

Исходный текст:

```
The feeling came to him Suddenly, at random intervals, and now
it was coming more often than ever.
```

Секретный ключ:

```
38 33 38 32 30 5f 64 65 6e 69 73 6f 76 69 63 68
```

2.2. Исследование шифров

Зашифруем текст с использованием шифров AES, MARS, RC6, Serpent и Twofish и найдем для них энтропию (табл. 2.1) и оценим время проведения атаки “грубой силы” (табл. 2.2).

Табл. 2.1 – Исследование энтропии шифров

Шифр	Исходный текст	AES	MARS	RC6	Serpent	Twofish
Значение энтропии	3.97	6.43	6.35	6.45	6.27	6.30

Табл. 2.2 – Оценка времени атаки “грубой силы

	AES	MARS	RC6	Serpent	Twofish
14	0	0	0	0	0
12	1h	2h	1h	3h	2h
10	9.6y	17y	9.3y	27y	17y
8	6.3e05y	1.1e06y	6.1e05y	1.7e06y	1.1e06y
6	4.1e10y	7.1e10y	4.0e10y	1.1e11y	7.4e10y
4	2.7e15y	4.6e15y	2.6e15y	7.5e15y	4.8e15y
2	1.8e20y	3.1e20y	1.7e20y	4.9e20y	3.2e20y

Можно заметить, что все шифры показывают примерно одинаковые значения надежности как с точки зрения энтропии, так и с точки зрения атаки “грубой силы”. При этом в атаке “грубой силы” шифры Serpent, MARS и

Twofish показывают незначительно лучшие результаты в сравнении с AES и Serpent, что, однако, не играет решающей роли, т.к. атака в любом случае занимает непозволительно много времени.

3. АТАКА «ГРУБОЙ СИЛЫ» НА AES

3.1. Задание

1. Найти и запустить шаблон атаки в CrypTool 2: *AES Analysis using Entropy*.

2. Выбрать открытый текст (примерно 1000 знаков) и загрузить его в шаблон.

3. Провести атаку «грубой силы» когда известно n-2, n-4, n-6 байт секретного ключа, используя в качестве оценочной функции энтропию и задействовав 1 ядро процессора. Зафиксировать затраты времени.

4. Выполнить атаку повторно с средним и максимальным количеством процессорных ядер. Зафиксировать затраты времени.

5. Сформировать текст с произвольным сообщением в формате «DEAR SIRS message THANKS» и загрузить его в шаблон.

6. Провести атаку «грубой силы» когда известно n-2, n-4, n-6 байт секретного ключа, используя в качестве оценочной функции словосочетание DEAR SIRS задействовав 1 ядро процессора. Зафиксировать затраты времени.

7. Выполнить атаку повторно с средним и максимальным количеством процессорных ядер. Зафиксировать затраты времени.

3.2. Исходные данные

Исходный текст для энтропийной атаки:

```
Thank you, sir, said the voice, without interest, and the face
leaned forward for a moment. The face was wind-browned, cut by
lines of weariness and cynical resignation; the eyes were
intelligent. Eddie Willers walked on, wondering why he always
```

felt it at this time of day, this sense of dread without reason. No, he thought, not dread, there's nothing to fear: just an immense, diffused apprehension, with no source or object. He had become accustomed to the feeling, but he could find no explanation for it; yet the bum had spoken as if he knew that Eddie felt it, as if he thought that one should feel it, and more: as if he knew the reason.
Eddie Willers pulled his shoulders straight, in conscientious self-discipline. He had to stop this, he thought; he was beginning to imagine things. Had he always felt it? He was thirty-two years old. He tried to think back. No, he hadn't; but he could not remember when it had started. The feeling came to him Suddenly, at random intervals, and now it was coming more often than ever. It's the twilight, he thought; I hate the twilight.

Исходный текст для текстовой атаки:

```
DEAR SIRS Pavel, Andrey THANKS
```

Секретный ключ:

```
38 33 38 32 30 5f 64 65 6e 69 73 6f 76 69 63 68
```

3.3. Шаблон атаки в CrypTool 2

CrypTool 2 предлагает шаблон для атаки на зашифрованный текст (рис. 3.1), позволяющий задать функцию стоимости, известные части ключа, параметры нагрузки на процессор и т.д. На вход подается строка байт зашифрованного текста в шестнадцатеричном виде.

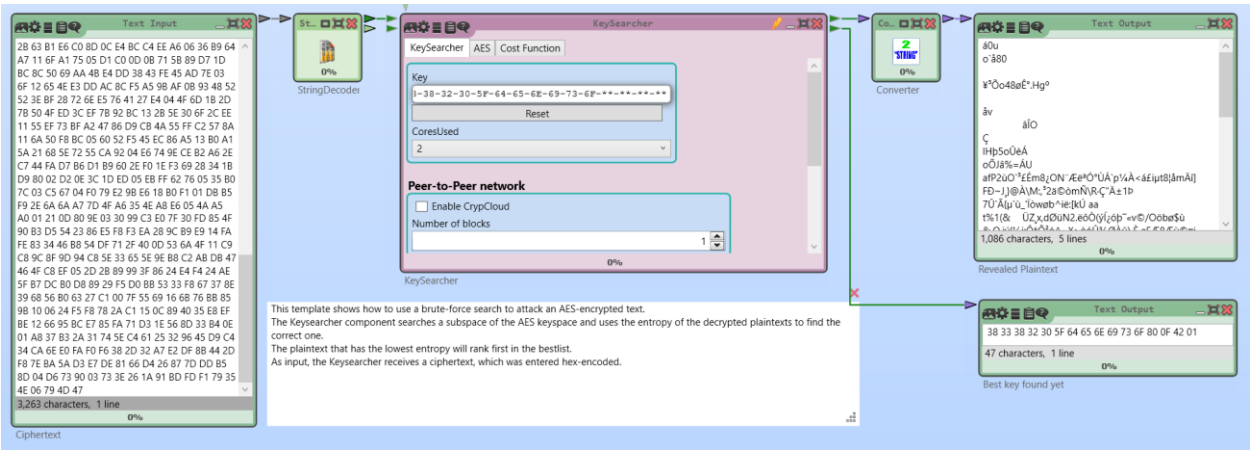

Рис. 3.1 – Шаблон AES атаки в CrypTool 2

3.4. Энтропийная атака

Была проведена атака на основе энтропийной функции, результаты которой записаны в таблице 3.1.

Табл. 3.1 – Результат энтропийной атаки

	1 ядро	3 ядра	6 ядер
10 байт	3552d	1250d	660d
12 байт	1h 10m	25m	14m
14 байт	1s	1s	1s

Результат энтропийной атаки на большом количестве ядер быстрее атаки грубой силы из CrypTool 1. Тем не менее, на одном ядре они примерно равны, поэтому можно предположить, что в CrypTool 1 используется этот же вариант атаки, но задействуется всегда одно ядро.

3.5. Текстовая атака

Была проведена текстовая атака на текст с использованием известной части сообщения, заданной в виде регулярного выражения:

```
DEAR SIRS .*
```

Результаты атаки представлены в таблице 3.2.

Таблица 3.2 – Результат текстовой атаки

	1 ядро	3 ядра	6 ядер
10 байт	1150d	425d	256d
12 байт	24m	9m	5m
14 байт	1s	1s	1s

Скорость текстовой атаки выше скорости энтропийной атаки, однако для при большом количестве неизвестных байт ключа это различие становится несущественным для дешифровщика в связи с большими временными затратами.

4. PADDING ORACLE ATTACK

4.1. Задание

1. Найти и запустить шаблон атаки в CrypTool 2: *Padding Oracle Attack on AES.*

2. Подготовьтесь к атаке теоретически:

a. Изучите комментарии к шаблону

b. Изучите публикацию

3. Внедрите во второй блок исходного текста коды символов своего имени.

4. Выполните 3 фазы атаки и сохраните итоговые скриншоты по окончанию каждой фазы.

5. Убедитесь, что атака удалась.

4.2. Исходные данные

Во втором блоке последние 5 байт были заменены на коды символов `pavel`:

```
70 61 76 65 6C
```

Секретный ключ:

```
83820_denisovich
```

4.3. Атака в CrypTool 2

В первой фазе атаки был найден такой последний байт первого блока, при котором получается верное дополнение (рис. 4.1).

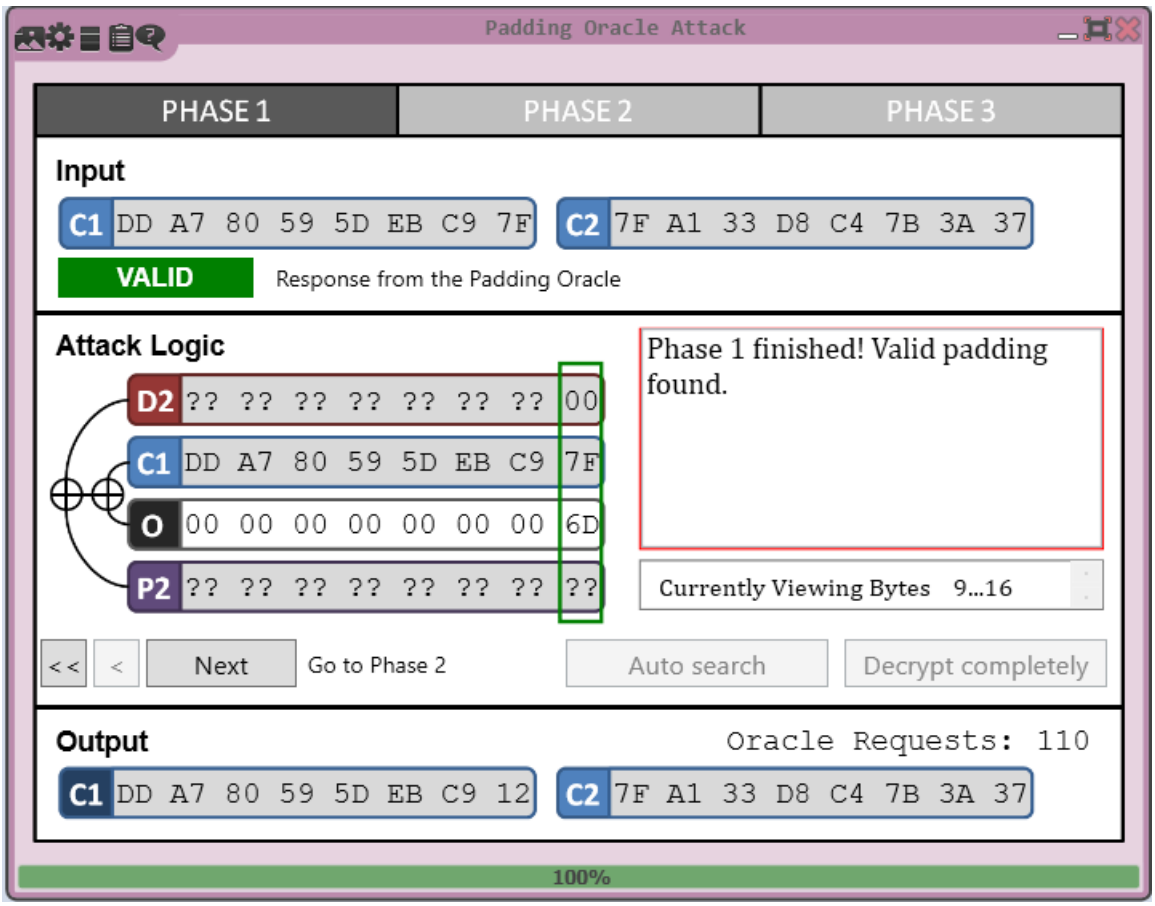

Рис. 4.1 – Результат первой фазы атаки

В данном случае таким байтом является $7F \oplus 6D = 12_{16}$.

Во второй фазе необходимо определить, какое значение байта дополнения было получено – конец текста P2 мог быть как *01*, так и *02 02*, так и *03 03 03* и т.д. Для этого поочередно изменим все байты, начиная с первого байта второго блока и заканчивая 15-м байтом второго блока: если при очередном изменении будет получена ошибка дополнения, то этот байт является частью дополнения, а значит известна длина дополнения, что позволяет получить значения байтов дополнения. В нашем случае длина дополнения равна 1 и последний байт текста P2 на конец первой фазы был равен *01*. Результат второй фазы представлен на рис. 4.2.

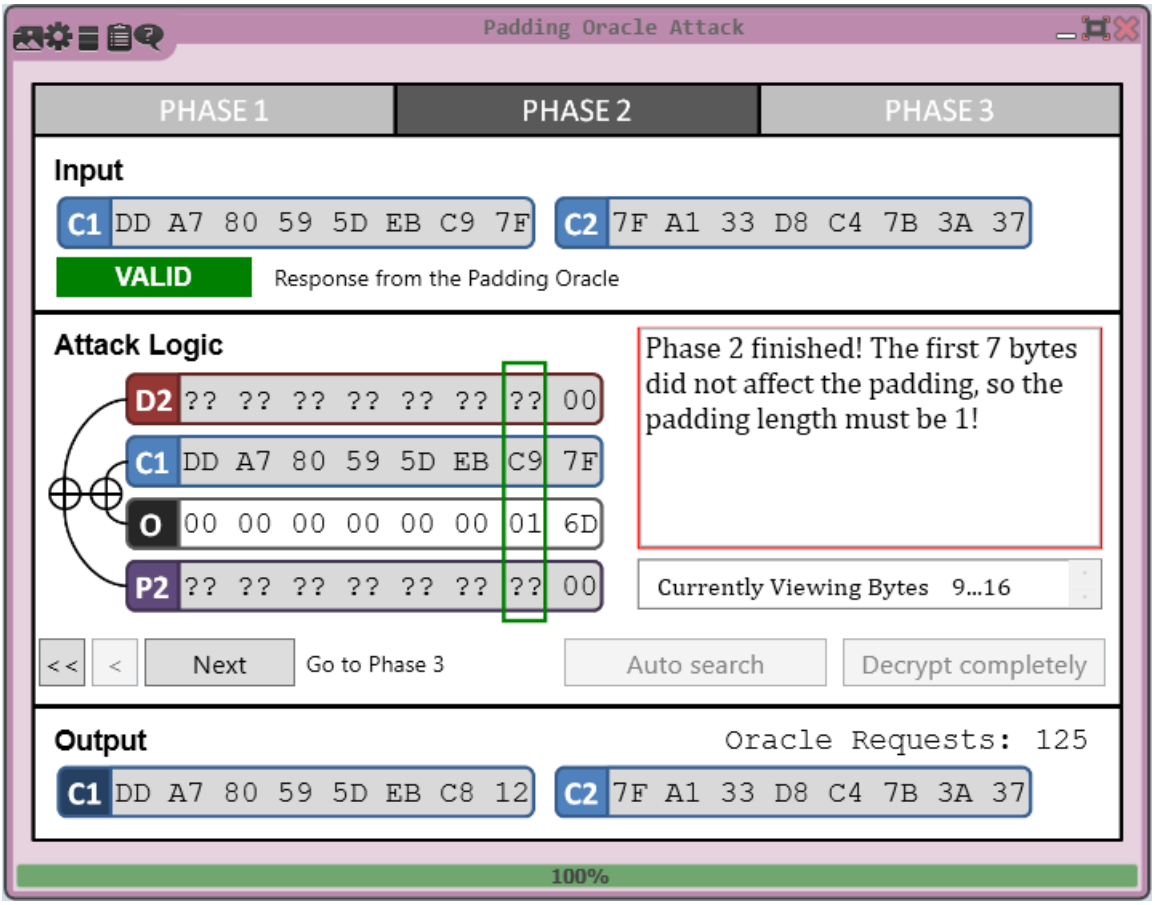

Рис. 4.2 – Результат второй фазы атаки

Исходя из полученной информации, можно восстановить последний байт блока D2: он будет равен $12 \oplus 01 = 13_{16}$, а значит последний байт исходного текста равен сложению 13 и последнего байта первого блока: $13 \oplus 7F = 6C$ (что и правда совпадает с исходным текстом).

Далее проведем аналогичный процесс для оставшихся байт второго блока зашифрованного текста. Отличие заключается лишь в том, что теперь нам не нужно выполнять вторую фазу атаки – нам гарантированно известны значения байт исходного текста, для которых известны соответствующие D2 значения. Результат дешифровки показан на рис. 4.3.

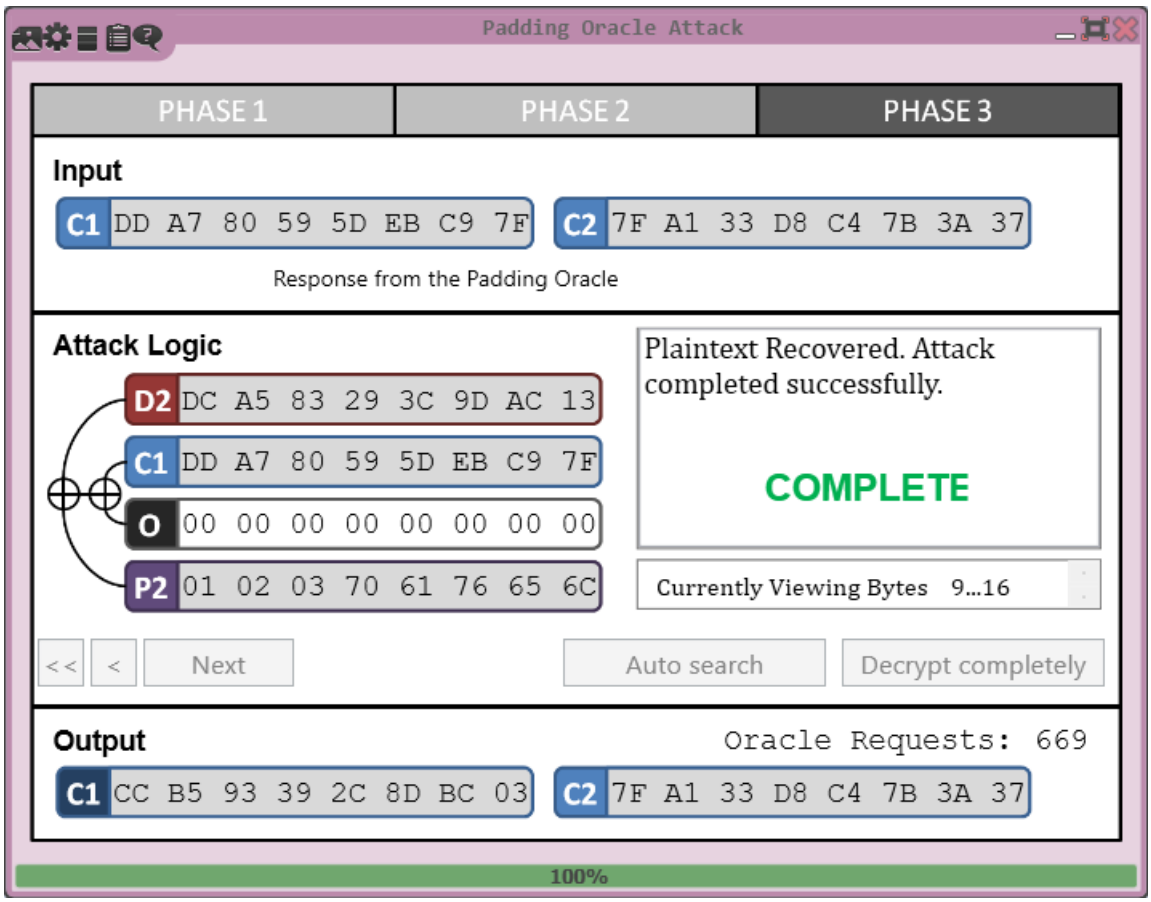

Рис. 4.3 – Результат третьей фазы атаки

Видно, что установленные 5 символов *pavel* в конце текста (как и остальные символы второго блока) были успешно дешифрованы, причем дешифровка потребовала 669 запросов к условному серверу.

5. ВЫВОДЫ

В ходе выполнения работы был разобран алгоритм работы шифра AES для 128-битного ключа. Описана реализация алгоритма, включая операции внутри раунда и процедуру генерации раундовых ключей.

В качестве проверки теоретического материала было выполнено ручное шифрование одного блока текста для одного раунда. Результат зашифрования был успешно проверен с использованием функций визуализации CrypTool 1.

Проведен сравнительный анализ шифров AES, MARS, RC6, Serpent и Twofish, основанный на проведении энтропийной атаки и атаке грубой силы.

Определено, что MARS, RC6, Serpent и Twofish не имеют решающего преимущества перед AES с использованием рассмотренных атак.

Далее на шифр AES были проведены текстовая и энтропийная атаки с использованием различного (1, 3 и 6) количества ядер процессора и известных байт ключа. На используемых исходных данных результаты атак показали сравнимую эффективность.

Также была рассмотрена Padding Oracle Attack на шифр AES. Изучен алгоритм проведения атаки, и проведена пробная атака средствами CrypTool 2. В результате атаки текст оказался успешно дешифрован и получено исходное значение.