**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационной безопасности**

отчет

**по лабораторной работе №5**

**по дисциплине «Криптография и защита информации»**

Тема: Изучение шифра AES

|  |  |  |
| --- | --- | --- |
| Студент гр. 9381 |  | Колованов Р.А. |
| Преподаватель |  | Племянников А.К. |

Санкт-Петербург

2022

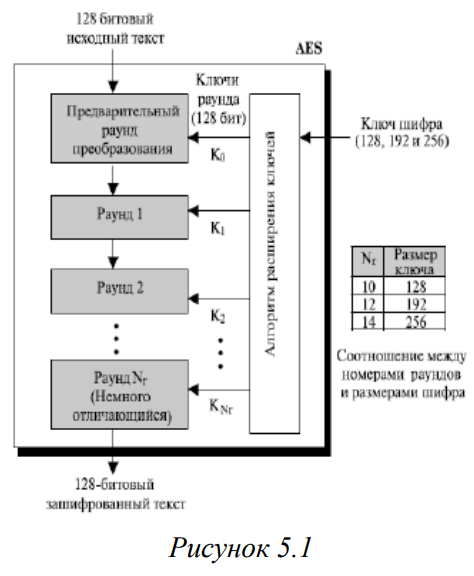
**Цель работы.**

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.

**Основные теоретические положения.**

***Преобразования шифра AES.***

Шифр AES (Rijndael) работает на основе перестановочно-подстановочной сети (SP-сеть). Обобщенная схема работы алгоритма представлена на рисунке 5.1.

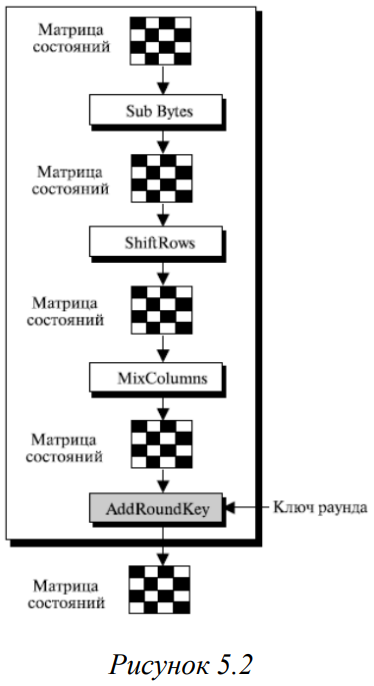
В версии с наименьшей длиной ключа алгоритм AES получает на вход блок открытого текста размером 16 байт и 16 байт ключа. Значения блока записывается в столбцы матрицы состояний размером 4х4 байт.

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 5.2:

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 (XOR 11B) на матрицу констант в конечном поле 𝐺𝐹(28) выполняет преобразование *MixColumn*, а сложение полученных столбцов матрицы состояний с раундовым ключом операцией XOR – преобразование *AddRoundKey*;

5. Действия 2-5 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.

***Финалисты конкурса AES.***

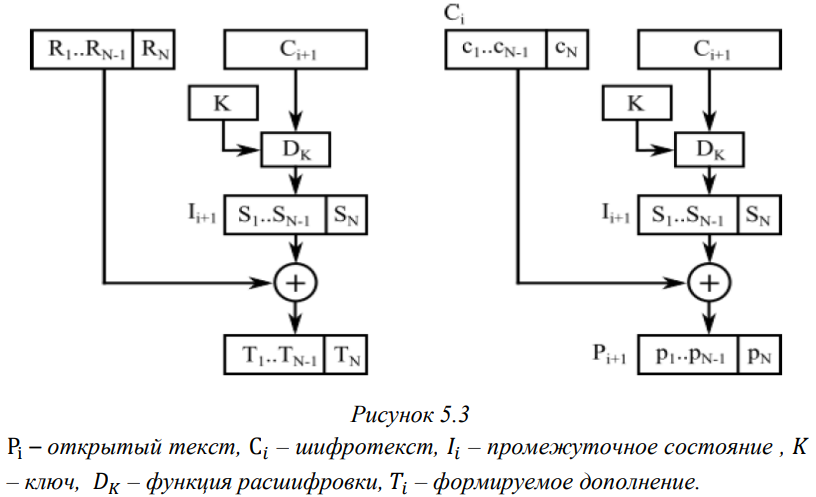
Победителем конкурса AES является алгоритм Rijndael (ставший AES), так как по всем характеристикам этот алгоритм не уступает остальным алгоритмам-финалистам. Остальные финалисты конкурса (Serpent, Twofish, MARS и RC6), практически равнозначны по совокупности характеристик, за исключением алгоритма MARS, имеющего существенно больше недостатков, в том числе алгоритм практически нереализуем в условиях ограниченных ресурсов.

***Атака предсказанием дополнения на шифр AES в режиме CBC.***

При проведении этой атаки предполагается, что нарушитель может модифицировать и отправлять зашифрованное сообщение серверу для расшифровки, а также распознавать ответы сервера о корректности дополнения последнего блока. Дешифровка сообщения нарушителем начинается с последнего блока шифротекста.

Рассмотрим расшифровку блока 𝐶i+1:

1. Формируем R – все биты, кроме последнего, случайные значения. Перебираем байт 𝑅n от 0x00 до 0xFF, каждый раз посылая на сервер [𝑅||𝐶i+1]. Если при некотором 𝑅n сервер «одобряет», то 𝑇n = 01, 𝑆n = 𝑅n ⊕ 0x01, 𝑝n = 𝑆n ⊕ 𝐶n. Схема первого этапа представлена на рисунке 5.3.



2. Формируем R – все биты, кроме двух последних, случайные значения. 𝑅n = 𝑆n ⊕ 0𝑥02, чтобы 𝑇n = 02. Перебираем байт 𝑅n−1 от 0x00 до 0xFF, каждый раз посылая на сервер [𝑅||𝐶i+1]. Если при некотором 𝑅n-2 сервер «одобряет», то 𝑇n-1 = 02, 𝑆n = 𝑅n-1 ⊕ 0x02, 𝑝n-1 = 𝑆n-1 ⊕ 𝐶n-1.

На третьем шаге пытаемся получить дополнение 030303, на четвертом – 04040404. После N шагов получаем полностью блок 𝑝i+1.

В CrypTool 2 атака предсказанием дополнения реализована в три фазы:

* Фаза 1. Нахождение длины дополнения, т.е. последний байт;
* Фаза 2. Подбор дополнения;
* Фаза 3. Расшифровка текста.

**Ход работы.**

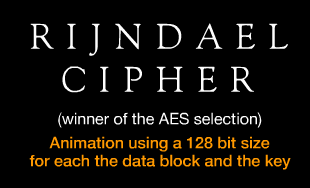
***Исследование преобразований AES.***

*Задание.*

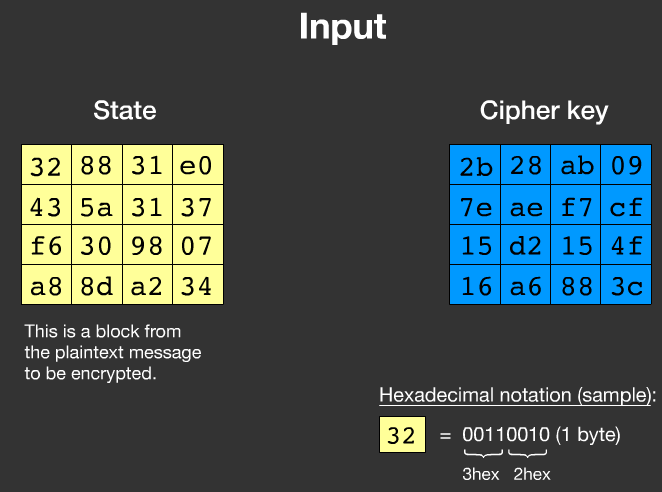
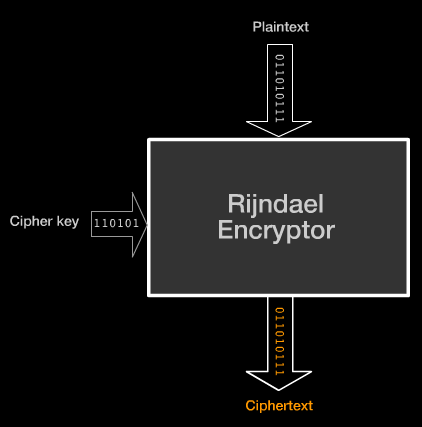
1. Изучить преобразования шифра AES с помощью демонстрационного приложения из Cryptool 1 (Indiv.Procedures -> Visualization -> AES -> Rijndael Animation);
2. Выполнить вручную преобразования для одного раунда и вычисление раундового ключа при следующих исходных данных:
   1. Открытый текст – фамилия\_имя (транслитерация латиницей);
   2. 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).

*Изучение преобразования шифра AES.*

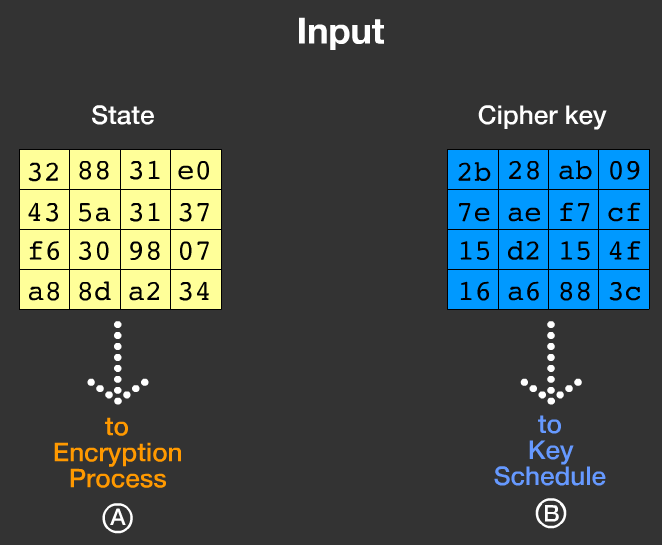
Изучим преобразования AES при помощи демонстрационного приложения из Cryptool 1. Как видно из первого кадра, демонстрироваться будет шифр AES c 128-битным блоком и ключом:



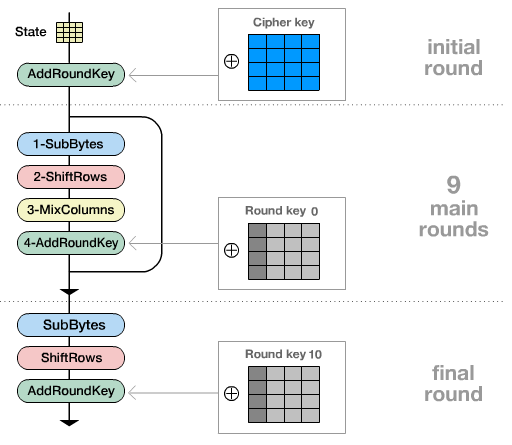
Далее показывается обобщенная схема шифрования. На вход шифру поступает 128-битный блок данных и 128-битный ключ, которые представлены в виде матриц 4 на 4:



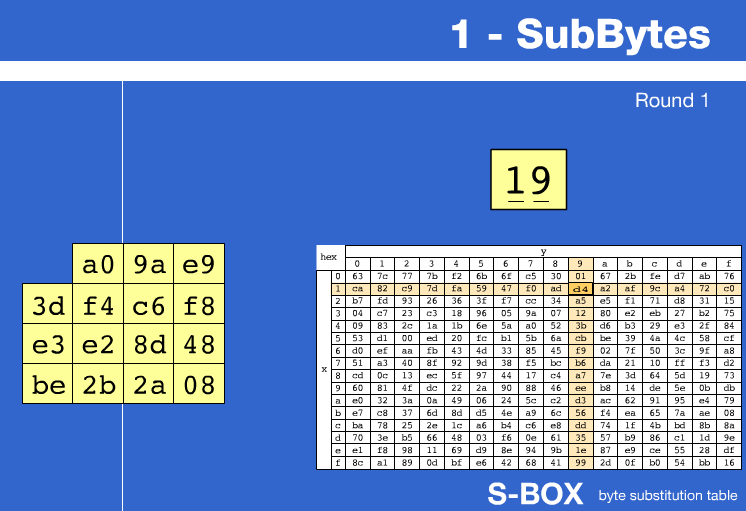
Матрица состояний подается на раундовые преобразования, а ключ – на процедуру генерации раундовых ключей:



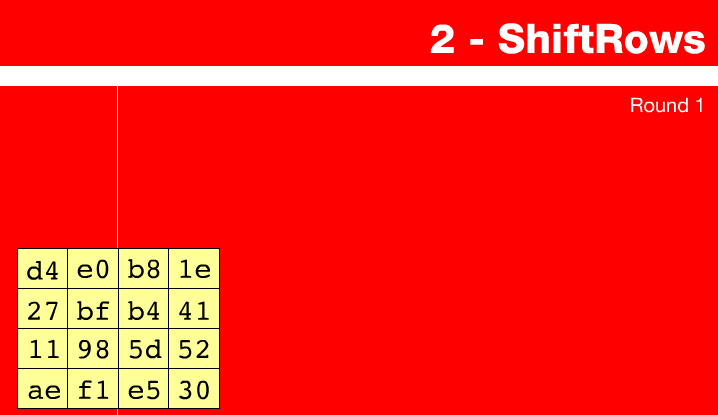
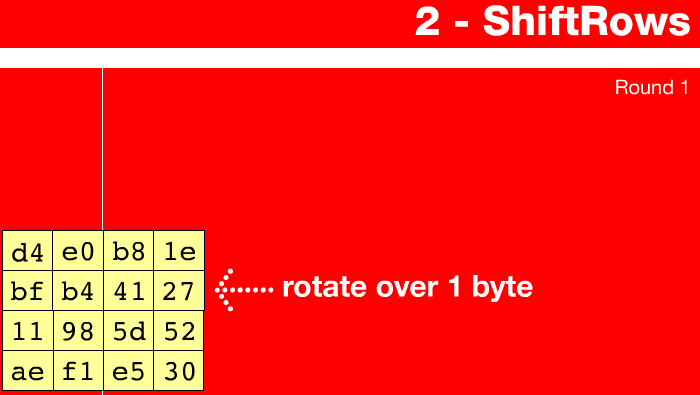
В процессе шифрования матрица состояний преобразуется при помощи 10 раундов. Раунды состоят из четырех последовательных преобразований: *SubBytes*, *ShiftRows*, *MixColumns* и *AddRoundKey*, при этом последний раунд не включает в себя *MixColumns*. Перед раундовыми преобразованиями матрицы данных на «initial round» производится преобразование *AddRoundKey*, только с исходным ключом. Следующие десять раундов используют раундовые ключи. Схема представлена на следующем рисунке:

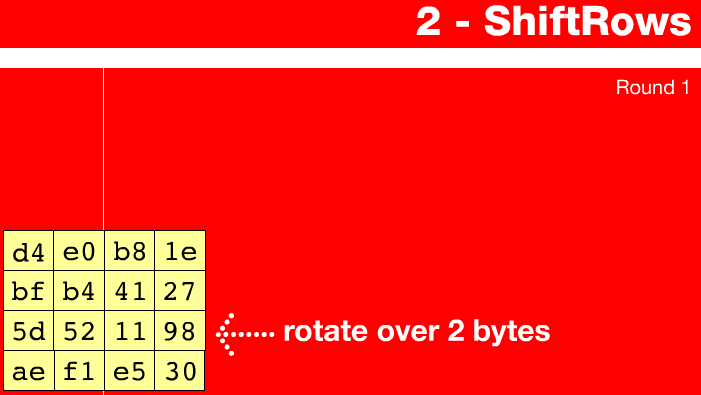
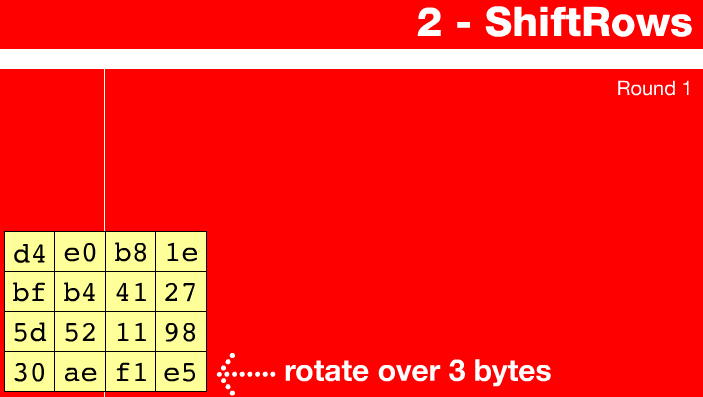


Далее рассматриваются раундовые преобразования. Первое преобразование – *SubBytes*. На нем осуществляется замена байтов матрицы состояний при помощи таблицы замены:

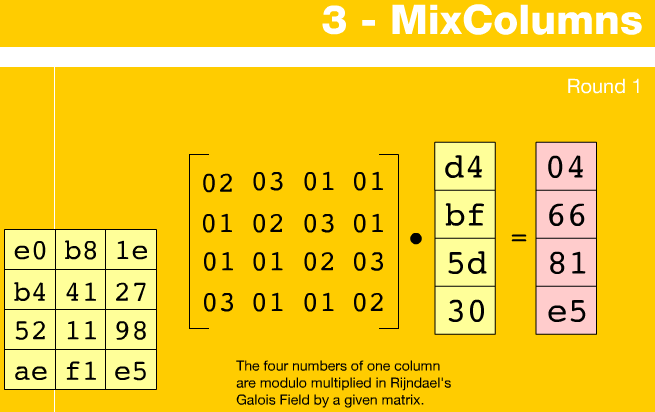


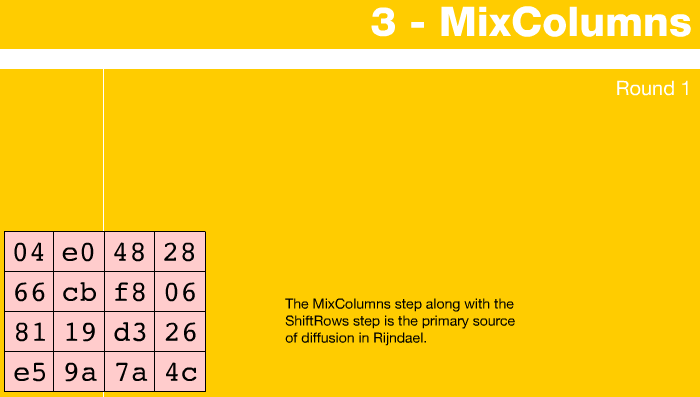
Второе преобразование – *ShiftRows*. На нем происходит побайтовый сдвиг строк матрицы состояний. Размер сдвига определяется индексом строки:

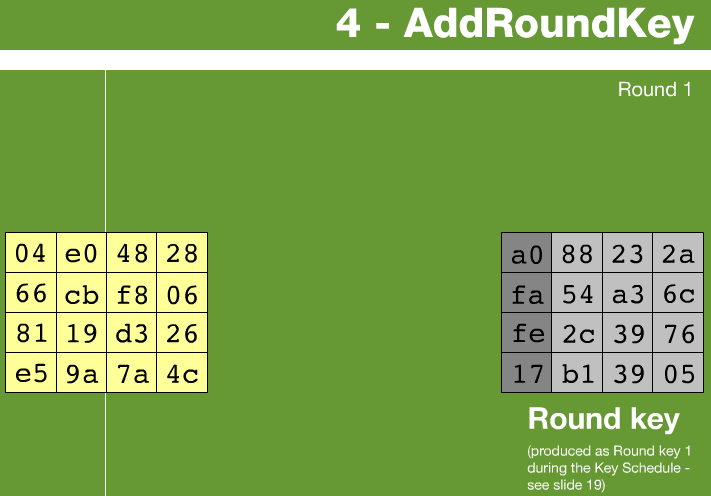
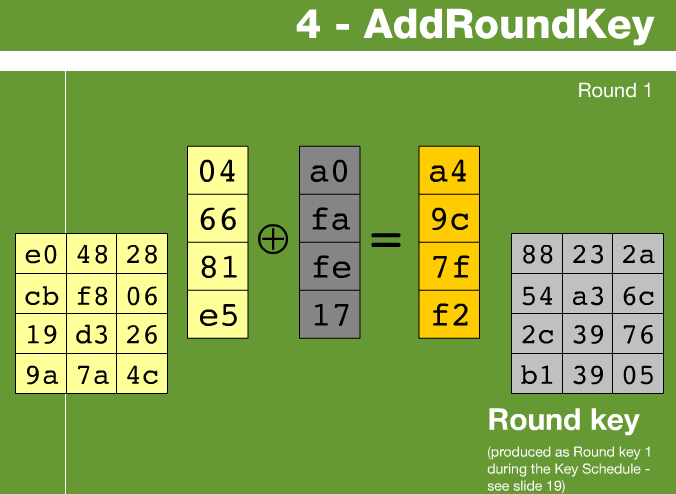
 

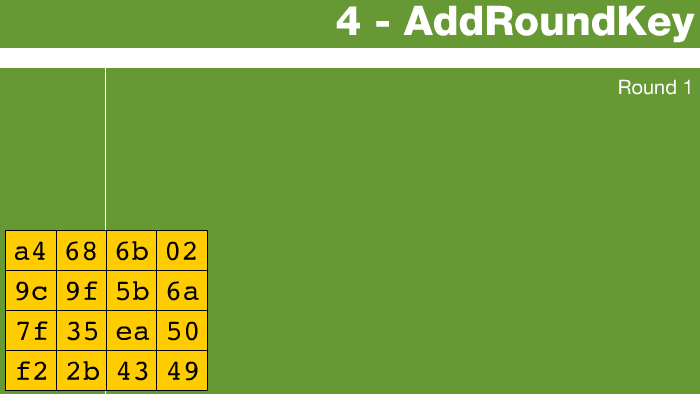
Третье преобразование – *MixColumns*. На нем происходит смешивание столбцов матрицы состояний путем ее умножения на матрицу констант в конечном поле 𝐺𝐹(28):

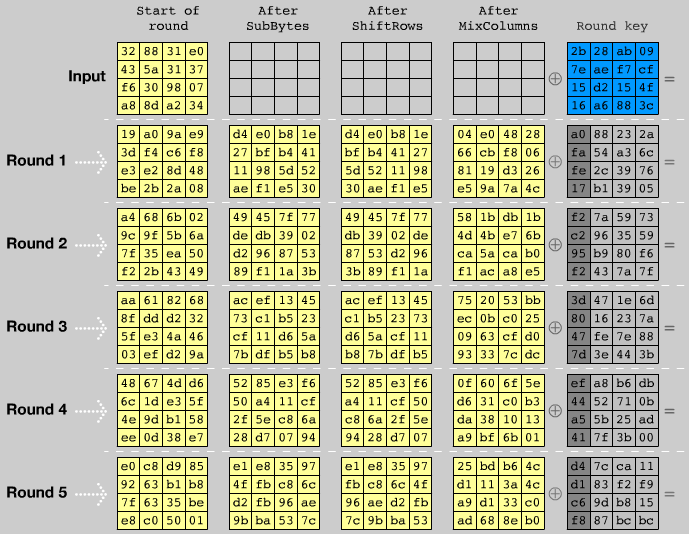
****

Четвертое преобразование – *AddRoundKey*. На нем происходит сложение по модулю 2 матрицы состояний с раундовым ключом:

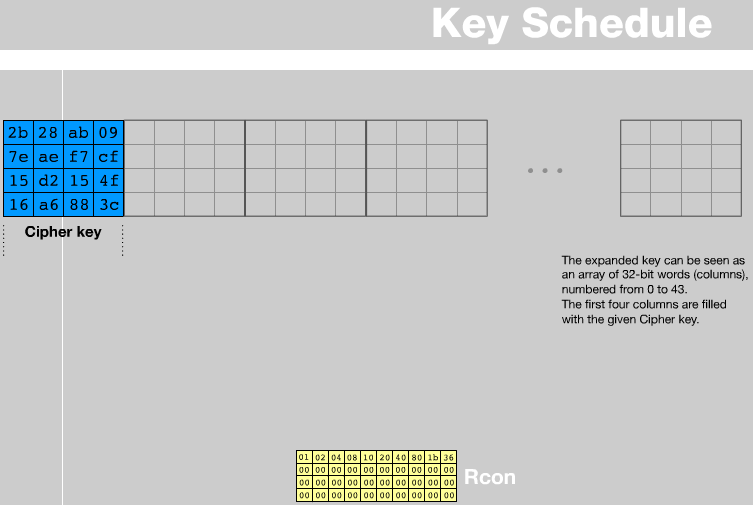
****

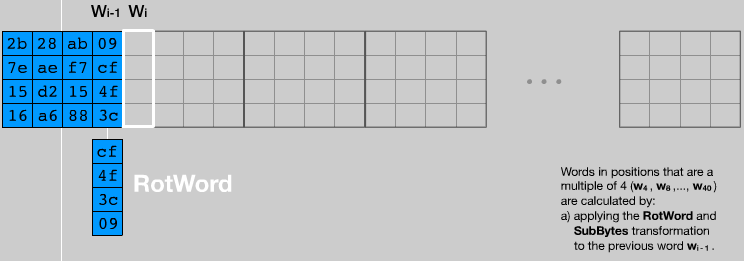
Результаты раундовых преобразований:

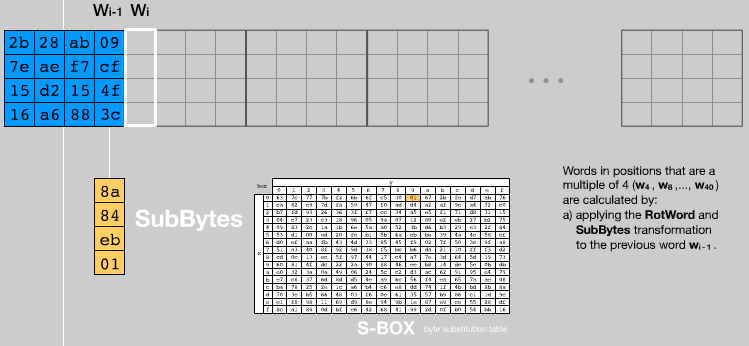


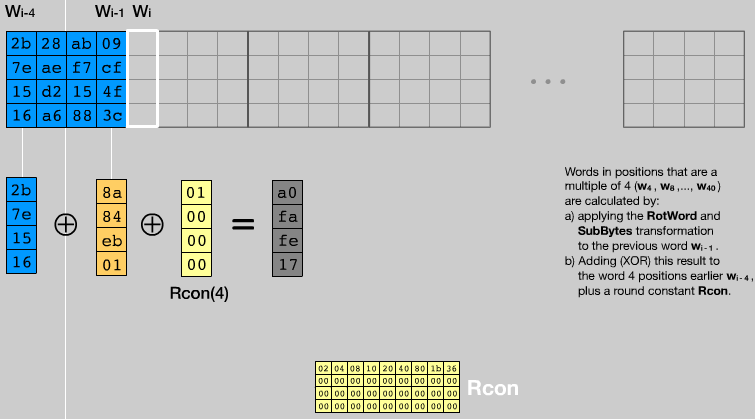


Далее рассматривается генерация 11 раундовых ключей. Для генерации следующего столбца ключа берется предыдущий, для него применяется преобразование *RotWord* (циклический сдвиг верх) и *SubBytes* (замена байтов по таблице замены), после чего результат складывается по модулю 2 со столбцом, стоящим на 4 позиции позади, и со соответствующим столбцом матрицы *Rcon*:

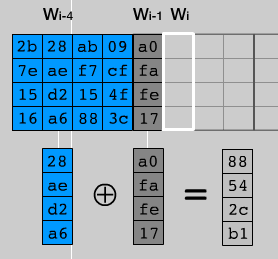
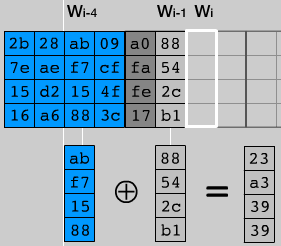


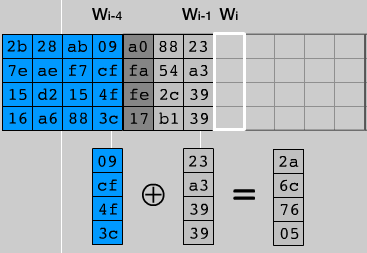
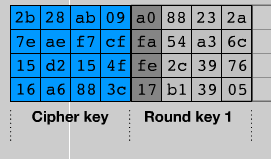
****

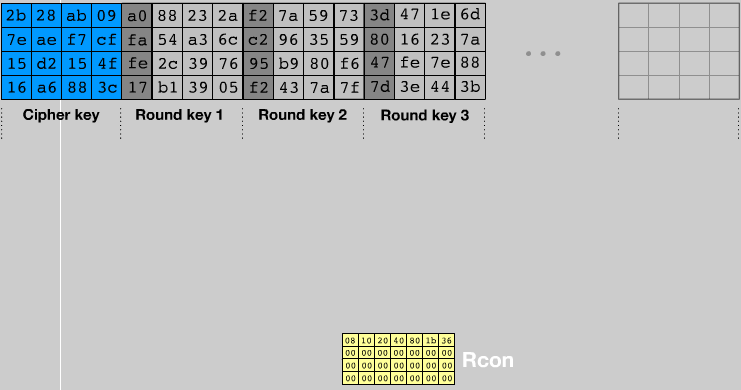
****

****

После генерации первого столбца, следующие три генерируются при помощи сложения по модулю два со столбцом, стоящим на 4 позиции позади. Для генерации пятого-восьмого столбцов и следующих четверок осуществляются аналогичные действия.

** **

**** ****

****

****

На этом демонстрация преобразований окончена.

*Ручное преобразование первого раунда шифра AES.*

Теперь выполним ручное преобразование первого раунда шифра AES и вычисление первого раундового ключа при следующих исходных данных:

А. Открытый текст – текст «KOLOVANOV\_RODION»;

Б. Ключ – текст «9381\_ALEKSEEVICH».

Для начала преобразуем исходные данные к бинарному виду:

|  |  |  |  |
| --- | --- | --- | --- |
| Буква | Код ASCII | Двоичный код | HEX код |
| K | 75 | 01001011 | 4B |
| O | 79 | 01001111 | 4F |
| L | 76 | 01001100 | 4C |
| O | 79 | 01001111 | 4F |
| V | 86 | 01010110 | 56 |
| A | 65 | 01000001 | 41 |
| N | 78 | 01001110 | 4E |
| O | 79 | 01001111 | 4F |
| V | 86 | 01010110 | 56 |
| \_ | 95 | 01011111 | 5F |
| R | 82 | 01010010 | 52 |
| O | 79 | 01001111 | 4F |
| D | 68 | 01000100 | 44 |
| I | 73 | 01001001 | 49 |
| O | 79 | 01001111 | 4F |
| N | 78 | 01001110 | 4E |

|  |  |  |  |
| --- | --- | --- | --- |
| Буква | Код ASCII | Двоичный код | HEX код |
| 9 | 57 | 00111001 | 39 |
| 3 | 51 | 00110011 | 33 |
| 8 | 56 | 00111000 | 38 |
| 1 | 49 | 00110001 | 31 |
| \_ | 95 | 01011111 | 5F |
| A | 65 | 01000001 | 41 |
| L | 76 | 01001100 | 4C |
| E | 69 | 01000101 | 45 |
| K | 75 | 01001011 | 4B |
| S | 83 | 01010011 | 53 |
| E | 69 | 01000101 | 45 |
| E | 69 | 01000101 | 45 |
| V | 86 | 01010110 | 56 |
| I | 73 | 01001001 | 49 |
| C | 67 | 01000011 | 43 |
| H | 72 | 01001000 | 48 |

Получаем входной 128-битный блок для шифрования и 128-битный ключ. Представим данные в виде матриц 4 на 4:

|  |  |  |  |
| --- | --- | --- | --- |
| 4B | 56 | 56 | 44 |
| 4F | 41 | 5F | 49 |
| 4C | 4E | 52 | 4F |
| 4F | 4F | 4F | 4E |

|  |  |  |  |
| --- | --- | --- | --- |
| 39 | 5F | 4B | 56 |
| 33 | 41 | 53 | 49 |
| 38 | 4C | 45 | 43 |
| 31 | 45 | 45 | 48 |

Матрица состояний (открытый текст) Ключ

Теперь найдем первый раундовый ключ:

|  |  |  |  |
| --- | --- | --- | --- |
| 39 | 5F | 4B | 56 |
| 33 | 41 | 53 | 49 |
| 38 | 4C | 45 | 43 |
| 31 | 45 | 45 | 48 |

|  |  |  |  |
| --- | --- | --- | --- |
| 03 | 5C | 17 | 41 |
| 29 | 68 | 3B | 72 |
| 6A | 26 | 63 | 20 |
| 80 | C5 | 80 | C8 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C[4] | RotWord | SubBytes | C[1] | Rcon[1] | C[5] | C[2] | C[6] | C[3] | C[7] | C[4] | C[8] |
| 56 | 49 | 3B | 39 | 01 | 03 | 5F | 5C | 4B | 17 | 56 | 41 |
| 49 | 43 | 1A | 33 | 00 | 29 | 41 | 68 | 53 | 3B | 49 | 72 |
| 43 | 48 | 52 | 38 | 00 | 6A | 4C | 26 | 45 | 63 | 43 | 20 |
| 48 | 56 | B1 | 31 | 00 | 80 | 45 | C5 | 45 | 80 | 48 | C8 |

Итого, первый раундовый ключ равен:

|  |  |  |  |
| --- | --- | --- | --- |
| 03 | 5C | 17 | 41 |
| 29 | 68 | 3B | 72 |
| 6A | 26 | 63 | 20 |
| 80 | C5 | 80 | C8 |

Далее выполним раундовое преобразование. Перед началом первого раунда необходимо сложить по модулю 2 матрицу состояний с исходным ключом:

|  |  |  |  |
| --- | --- | --- | --- |
| 4B | 56 | 56 | 44 |
| 4F | 41 | 5F | 49 |
| 4C | 4E | 52 | 4F |
| 4F | 4F | 4F | 4E |

|  |  |  |  |
| --- | --- | --- | --- |
| 39 | 5F | 4B | 56 |
| 33 | 41 | 53 | 49 |
| 38 | 4C | 45 | 43 |
| 31 | 45 | 45 | 48 |

|  |  |  |  |
| --- | --- | --- | --- |
| 72 | 09 | 1D | 12 |
| 7C | 00 | 0C | 00 |
| 74 | 02 | 17 | 0C |
| 7E | 0A | 0A | 06 |

МС Ключ Результат

Теперь можно начать первые раундовые преобразования. Выполним *SubBytes*:

|  |  |  |  |
| --- | --- | --- | --- |
| 72 | 09 | 1D | 12 |
| 7C | 00 | 0C | 00 |
| 74 | 02 | 17 | 0C |
| 7E | 0A | 0A | 06 |

|  |  |  |  |
| --- | --- | --- | --- |
| 40 | 01 | A4 | C9 |
| 10 | 63 | FE | 63 |
| 92 | 77 | F0 | FE |
| F3 | 67 | 67 | 6F |

МС Результат

Выполним *ShiftRows*:

|  |  |  |  |
| --- | --- | --- | --- |
| 40 | 01 | A4 | C9 |
| 10 | 63 | FE | 63 |
| 92 | 77 | F0 | FE |
| F3 | 67 | 67 | 6F |

|  |  |  |  |
| --- | --- | --- | --- |
| 40 | 01 | A4 | C9 |
| 63 | FE | 63 | 10 |
| F0 | FE | F0 | FE |
| 6F | F3 | 67 | 67 |

МС Результат

Выполним *MixColumns*:

|  |  |  |  |
| --- | --- | --- | --- |
| 40 | 01 | A4 | C9 |
| 63 | FE | 63 | 10 |
| F0 | FE | F0 | FE |
| 6F | F3 | 67 | 67 |

МС

|  |  |  |  |
| --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |
| 01 | 02 | 03 | 01 |
| 01 | 01 | 02 | 03 |
| 03 | 01 | 01 | 02 |

|  |
| --- |
| 40 |
| 63 |
| F0 |
| 6F |

|  |
| --- |
| BA |
| E2 |
| 69 |
| 8D |

|  |  |  |  |
| --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |
| 01 | 02 | 03 | 01 |
| 01 | 01 | 02 | 03 |
| 03 | 01 | 01 | 02 |

|  |
| --- |
| 01 |
| FE |
| FE |
| F3 |

|  |
| --- |
| 16 |
| 0C |
| 16 |
| FE |

|  |  |  |  |
| --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |
| 01 | 02 | 03 | 01 |
| 01 | 01 | 02 | 03 |
| 03 | 01 | 01 | 02 |

|  |
| --- |
| A4 |
| 63 |
| F0 |
| 67 |

|  |
| --- |
| 03 |
| A8 |
| 51 |
| C8 |

|  |  |  |  |
| --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |
| 01 | 02 | 03 | 01 |
| 01 | 01 | 02 | 03 |
| 03 | 01 | 01 | 02 |

|  |
| --- |
| C9 |
| 10 |
| FE |
| 67 |

|  |
| --- |
| A9 |
| 17 |
| 9E |
| E9 |

|  |  |  |  |
| --- | --- | --- | --- |
| BA | 16 | 03 | A9 |
| E2 | 0C | A8 | 17 |
| 69 | 16 | 51 | 9E |
| 8D | FE | C8 | E9 |

Результат

Выполним *AddRoundKey*:

|  |  |  |  |
| --- | --- | --- | --- |
| BA | 16 | 03 | A9 |
| E2 | 0C | A8 | 17 |
| 69 | 16 | 51 | 9E |
| 8D | FE | C8 | E9 |

|  |  |  |  |
| --- | --- | --- | --- |
| 03 | 5C | 17 | 41 |
| 29 | 68 | 3B | 72 |
| 6A | 26 | 63 | 20 |
| 80 | C5 | 80 | C8 |

+

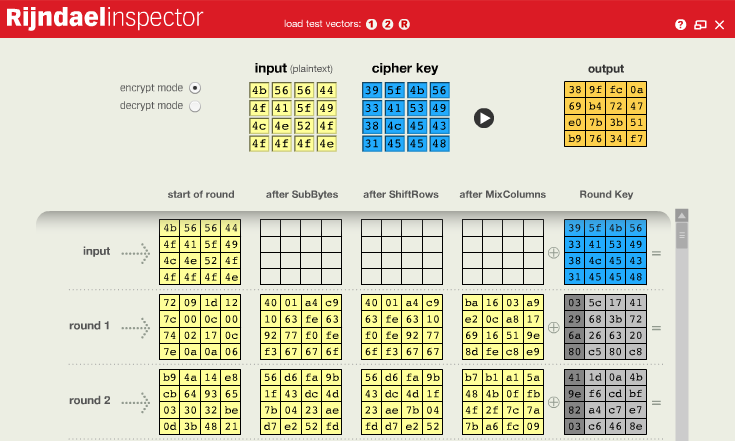
МС Первый раундовый ключ

Итого получаем:

|  |  |  |  |
| --- | --- | --- | --- |
| B9 | 4A | 14 | E8 |
| CB | 64 | 93 | 65 |
| 03 | 30 | 32 | BE |
| 0D | 3B | 48 | 21 |

МС

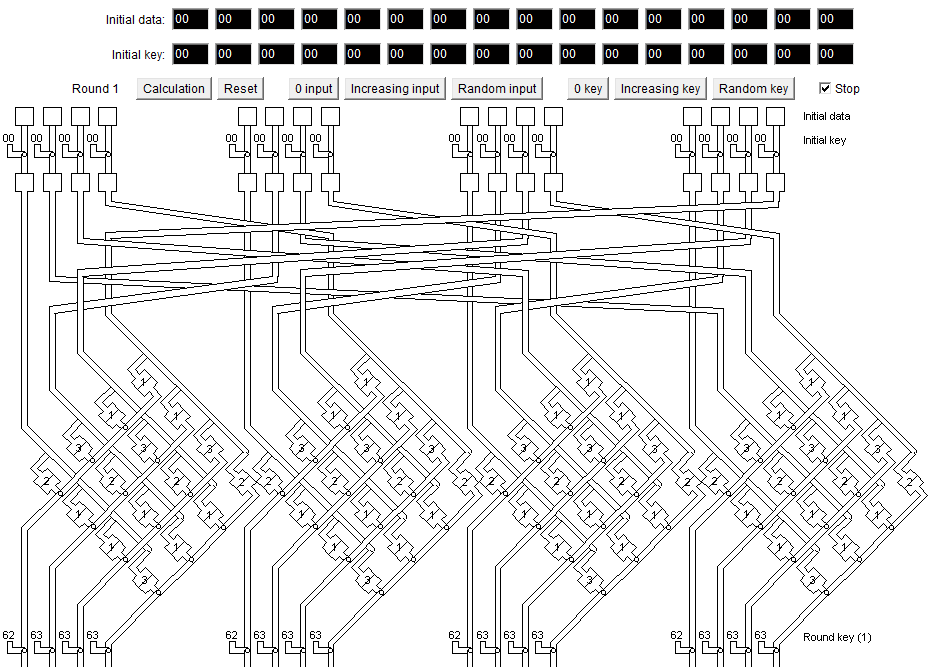
Проверим полученные результаты при помощи приложения инспектора из CrypTool 1:

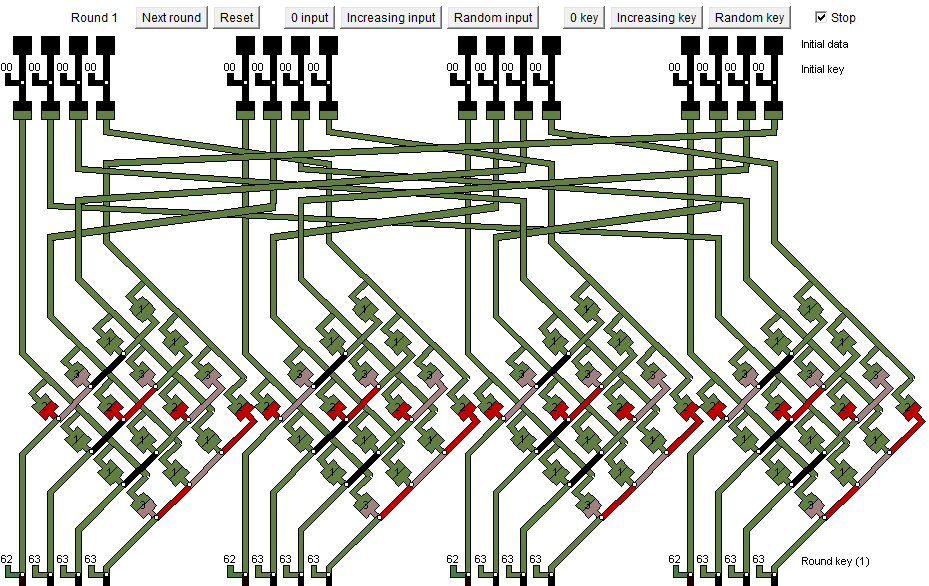


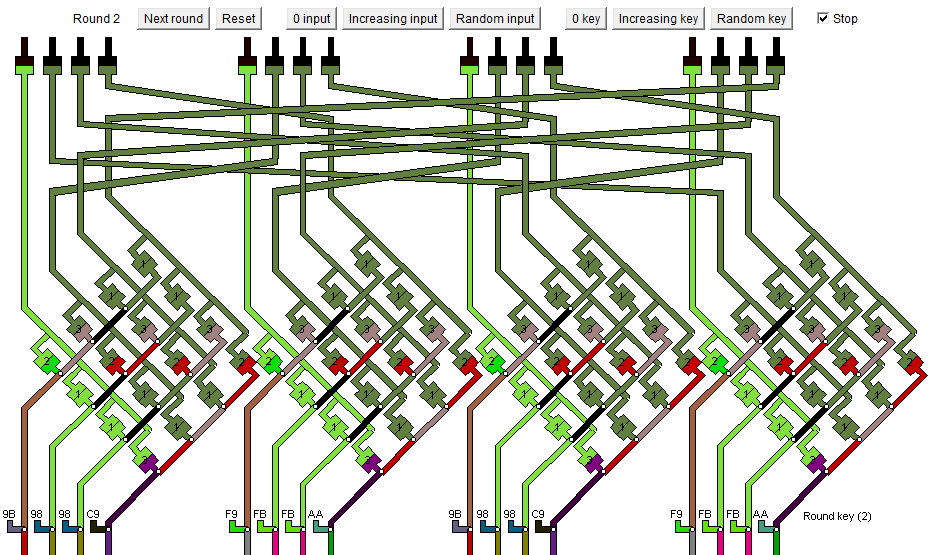
Как видно, полученные в ходе ручных преобразований промежуточные матрицы состояний совпадают.

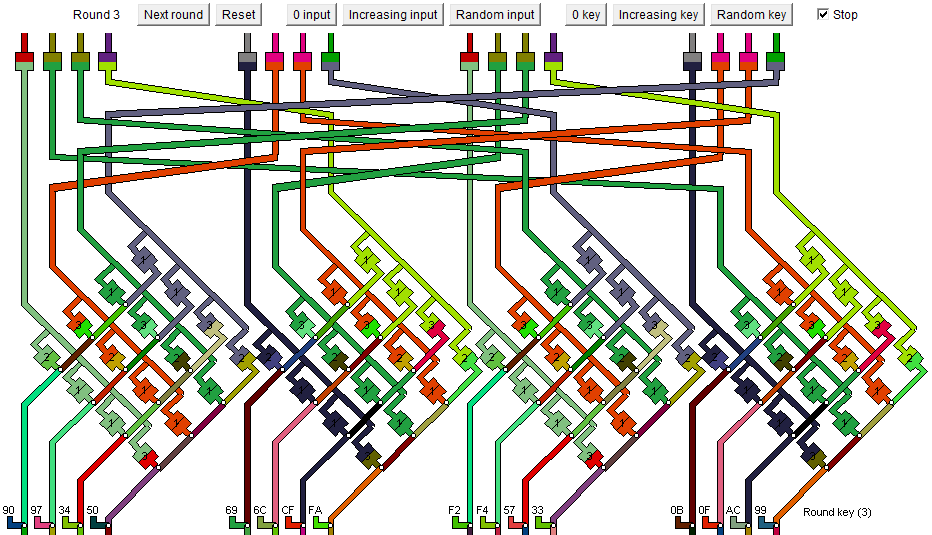
*Наблюдения в потоковой модели шифра AES.*

Проведем наблюдения в потоковой модели шифра AES при помощи демонстрационного приложения из CrypTool 1 для нулевого текста и нулевого ключа:

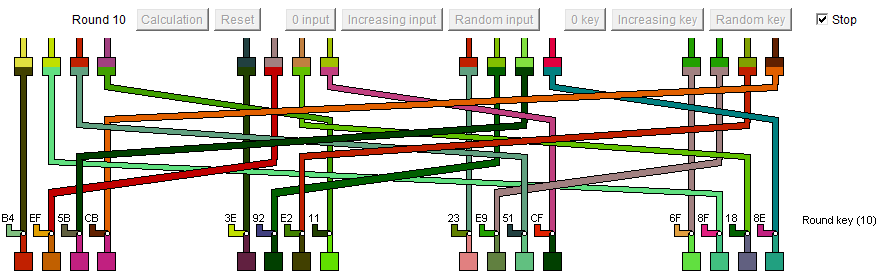








…



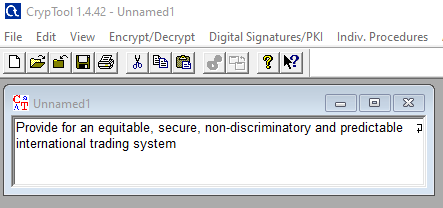
***Исследование финалистов конкурса AES (Rijndael, MARS, RC6, Serpent, Twofish).***

*Задание.*

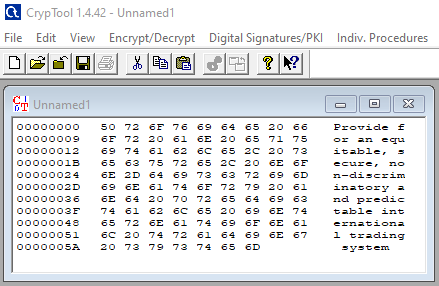
1. Выбрать текст на английском языке (не более 120 знаков);
2. Создать бинарный файл с этим текстом, зашифровав и расшифровав его шифром AES на 0-м ключе;
3. С помощью Cryptool 1 зашифровать c ключом отличным от 0 текст с использованием шифров AES, MARS, RC6, Serpent и Twofish;
4. Приложением из Cryptool 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.

*Энтропия исходного текста и шифротекстов, полученных при помощи шифров AES, MARS, RC6, Serpent и Twofish.*

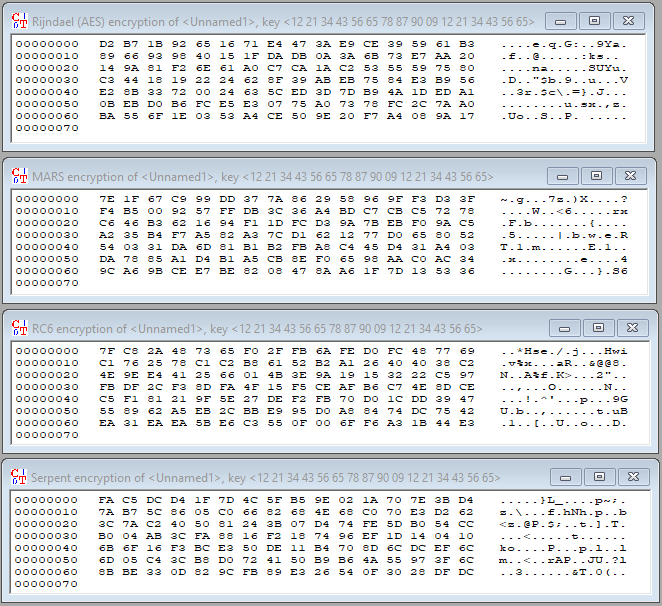
В качестве исходного текста был выбран следующий текст:

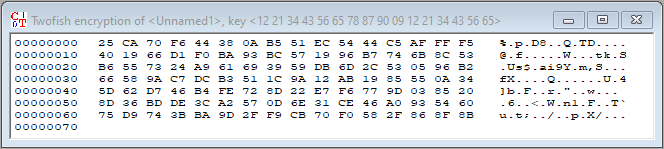


Бинарное представление выбранного текста:



Зашифруем c ключом «12 21 34 43 56 65 78 87 90 09 12 21 34 43 56 65» исходный текст с использованием шифров AES, MARS, RC6, Serpent и Twofish:





Вычислим энтропию для исходного текста и полученных шифротекстов при помощи инструментов CrypTool 1 и зафиксируем результаты в таблице:

|  |  |
| --- | --- |
| Текст | Значение энтропии |
| Исходный текст |  |
| Шифротекст AES |  |
| Шифротекст MARS |  |
| Шифротекст PC6 |  |
| Шифротекст Serpent |  |
| Шифротекст Twofish |  |

Как видно из результатов, для исходного текста значение энтропии самое низкое (4.15 из 8), а используемые значения байт охватывают лишь 24 из 256 возможных значений. Для шифров AES, MARS, PC6, Serpent и Twofish энтропия отличается не сильно: самая большая энтропия у шифротекстов MARS и PC6 (6.46 из 8), самая маленькая энтропия у шифротекста Serpent (6.27 из 8), у шифротекста Twofish – 6.37 из 8, у шифра AES – 6.41 из 8.

*Оценка времени проведения атаки «грубой силы» на шифры AES, MARS, RC6, Serpent и Twofish при различном количестве известных байт ключа.*

Оценим время проведения атаки «грубой силы» на шифры AES, MARS, RC6, Serpent и Twofish при 2, 4, 6, 8, 10, 12, 14 известных байт ключа с использованием CrypTool 1 и зафиксируем результаты в таблице:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество известных байт | Время проведения атаки «грубой силы» | | | | |
| AES | MARS | RC6 | Serpent | Twofish |
| 2 | ~ 2.4 \* 1020 лет | ~ 3.6 \* 1020 лет | ~ 2.4 \* 1020 лет | ~ 6.5 \* 1020 лет | ~ 4.8 \* 1020 лет |
| 4 | ~ 3.7 \* 1015 лет | ~ 5.5 \* 1015 лет | ~ 3.7 \* 1015 лет | ~ 1016 лет | ~7.4 \* 1015 лет |
| 6 | ~ 5.6 \* 1010 лет | ~ 8.4 \* 1010 лет | ~ 5.7 \* 1010 лет | ~ 1.6 \* 1011 лет | ~ 1.2 \* 1011 лет |
| 8 | ~ 8.5 \* 105 лет | ~ 1.3 \* 106 лет | ~ 8.6 \* 105 лет | ~ 2.4 \* 106 лет | ~ 1.7 \* 106 лет |
| 10 | ~ 13 лет | ~ 20 лет | ~ 13 лет | ~ 36 лет | ~ 27 лет |
| 12 | ~ 1 час 45 минут | ~ 2 часа 36 минут | ~ 1 час 45 минут | ~ 4 часа 50 минут | ~ 3 часа 32 минуты |
| 14 | ~ 0.1 секунд | ~ 0.1 секунд | ~ 0.1 секунд | ~ 0.3 секунд | ~ 0.1 секунд |

Как видно из результатов, самое большое время проведения атаки показал шифр Serpent – во всех рассматриваемых вариантах он показал наибольшее время атаки. Шифры AES и RC6 показали почти одинаковые результаты, при этом во всех рассматриваемых вариантах они показали наименьшее время атаки. Между ними в порядке увеличения времени проведения атаки «грубой силы» идут шифры MARS и Twofish.

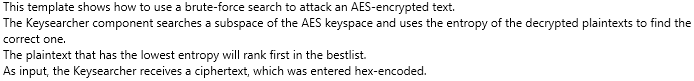
***Атака «грубой силы» на AES.***

*Задание.*

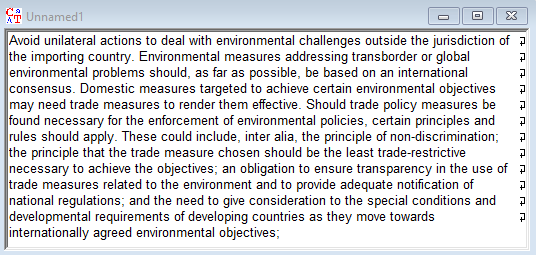
1. Найти и запустить шаблон атаки в CrypTool 2 (AES Analysis using Entropy);
2. Выбрать открытый текст (примерно 1000 знаков) и загрузить его в шаблон;
3. Провести атаку «грубой силы» когда известно n-2, n-4, n-6 байт секретного ключа, используя в качестве оценочной функции энтропию и задействовав 1 ядро процессора. Зафиксировать затраты времени;
4. Сформировать текст с произвольным сообщением в формате «DEAR SIRS message THANKS» и загрузить его в шаблон;
5. Провести атаку «грубой силы» когда известно n-2, n-4, n-6 байт секретного ключа, используя в качестве оценочной функции 34 словосочетание DEAR SIRS задействовав 1 ядро процессора. Зафиксировать затраты времени;
6. Выполнить атаку повторно с средним и максимальным количеством процессорных ядер. Зафиксировать затраты времени.

*Энтропийная атака «грубой силы» на шифр AES при различном количестве известных байт ключа.*

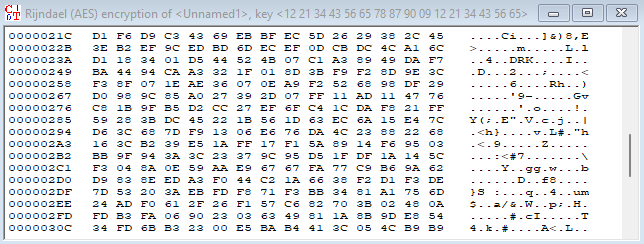
Для проведения атаки «грубой силы» будет использоваться шаблон «AES Analysis using Entropy» из CrypTool 2:



В качестве исходного текста был выбран следующий текст:



Исходный текст был зашифрован с ключом «12 21 34 43 56 65 78 87 90 09 12 21 34 43 56 65»:

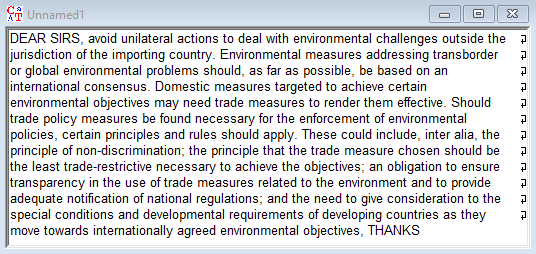


Зафиксируем затраты времени при проведении атаки «грубой силы», когда известно 14, 12, 10 байт секретного ключа, использующую в качестве оценочной функции энтропию и 1, 4 и 8 ядер процессора:

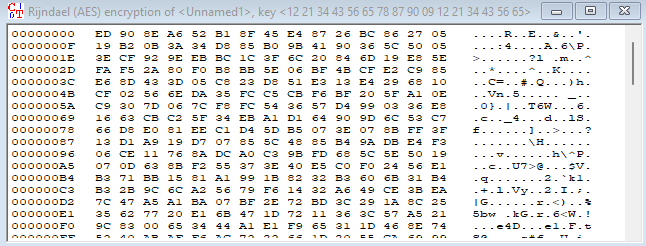
|  |  |  |  |
| --- | --- | --- | --- |
| Количество известных байт ключа | Затраты времени | | |
| 1 ядро | 4 ядра | 8 ядер |
| 10 | ~ 6220 дней | ~ 1777 дней | ~ 1036 дней |
| 12 | ~ 2 часа 16 минут | ~ 38 минут | ~ 23 минуты |
| 14 | ~ 1 секунда | ~ 1 секунда | ~ 1 секунда |

*Текстовая атака «грубой силы» на шифр AES при различном количестве известных байт ключа.*

В качестве исходного текста был выбран следующий текст:



Исходный текст был зашифрован с ключом «12 21 34 43 56 65 78 87 90 09 12 21 34 43 56 65»:



Зафиксируем затраты времени при проведении атаки «грубой силы», когда известно 14, 12, 10 байт секретного ключа, использующую в качестве оценочной функции словосочетание DEAR SIRS и 1, 4 и 8 ядер процессора:

|  |  |  |  |
| --- | --- | --- | --- |
| Количество известных байт ключа | Затраты времени | | |
| 1 ядро | 4 ядра | 8 ядер |
| 10 | ~ 3827 дней | ~ 1130 дней | ~ 721 день |
| 12 | ~ 1 час 24 минуты | ~ 23 минуты | ~14 минут |
| 14 | ~ 1 секунда | ~ 1 секунда | ~ 1 секунда |

Как видно из результатов, использование в качестве оценочной функции словосочетание DEAR SIRS из исходного текста значительно ускоряет поиск по сравнению с использованием в качестве оценочной функции энтропии (примерно в 1.6 раза быстрее). При увеличении количества используемых ядер процессора поиск также значительно ускоряется.

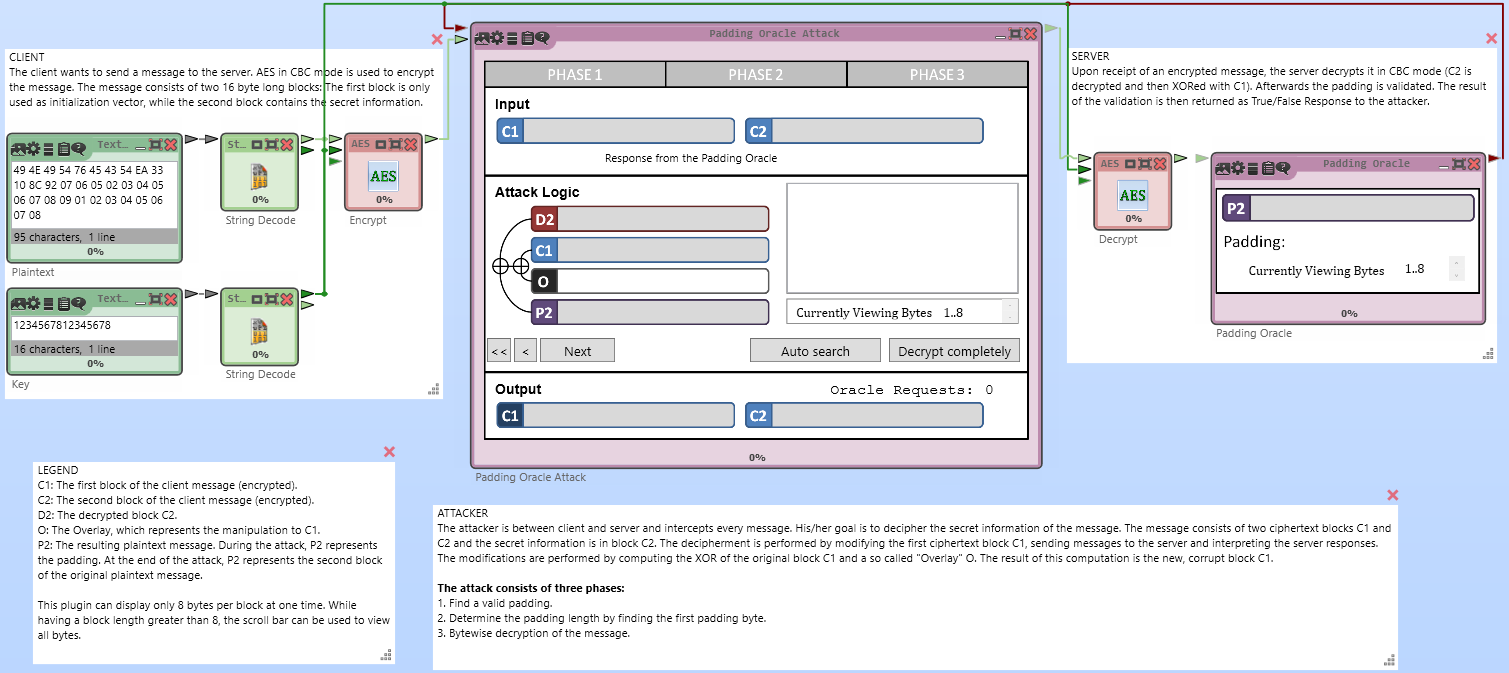
***Атака предсказанием дополнения на шифр AES в режиме CBC (Padding Oracle Attack).***

*Задание.*

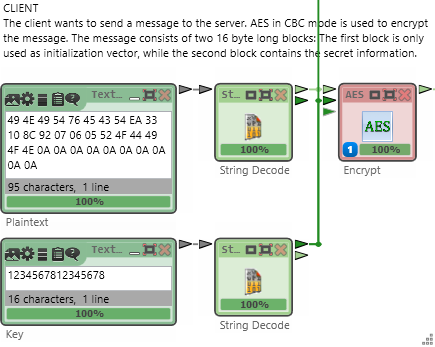
1. Найти и запустить шаблон атаки в CrypTool 2 (Padding Oracle Attack on AES);
2. Подготовьтесь к атаке теоретически:
   1. Изучите комментарии к шаблону;
   2. Изучите публикацию;
3. Внедрите во второй блок исходного текста коды символов своего имени;
4. Выполните 3 фазы атаки и сохраните итоговые скриншоты по окончанию каждой фазы;
5. Убедитесь, что атака удалась.

*Атака предсказанием дополнения на шифр AES в режиме CBC.*

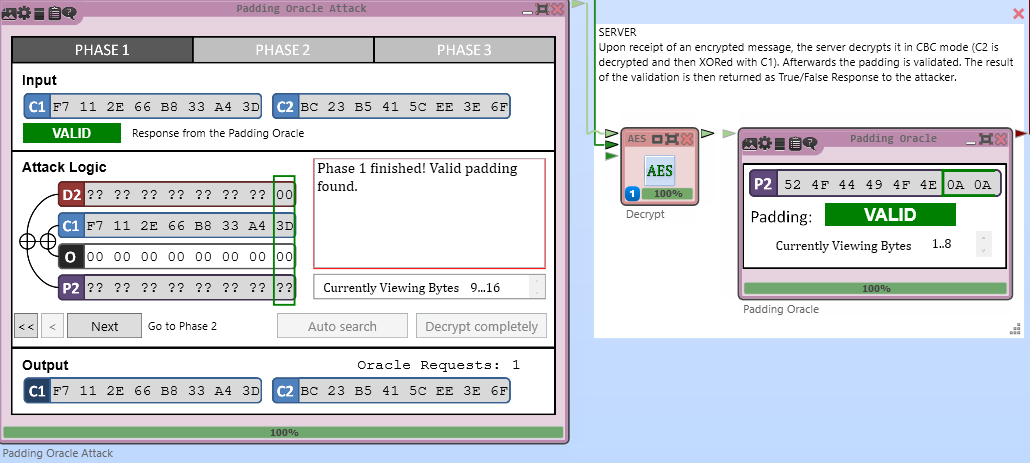
Для изучения и выполнения атаки был использован шаблон «Padding Oracle Attack on AES» приложения CrypTool 2:



Во второй 128-битный блок исходного текста был внедрен текст «RODION» (52 4F 44 49 4F 4E), после него был добавлен корректный padding в размере 10 байтов со значением 0x0A. В итоге второй блок выглядит следующим образом: «52 4F 44 49 4F 4E 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A».

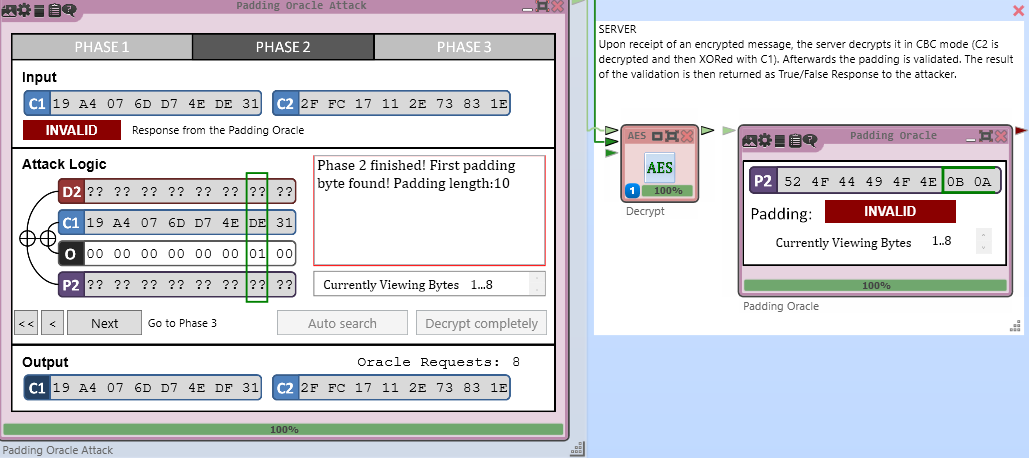


Выполним первую фазу атаки:



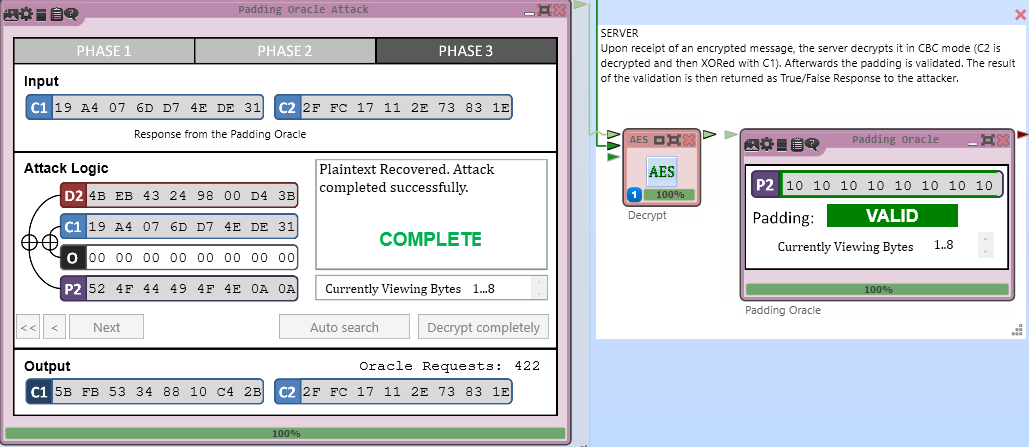
Поскольку изначально в исходном тексте установлен корректный padding, то передавая исходный шифротекст сервер сообщит о том, что padding исходного сообщения корректный.

Выполним вторую фазу:



На данном этапе мы определяем размер дополнения при помощи изменения байт последнего блока. Размер дополнения был определен корректно – 10.

Выполним третью фазу:



Зная дополнение, мы можем найти байты блока D2 по информации о корректности дополнения, перебирая значения для байтов C1. После того, как мы найдем D2, можно легко найти P2 (P2 = C1 XOR D2).

Как видно из рисунка, данные из блока P2 расшифровались корректно.

**Выводы.**

В ходе выполнения данной лабораторной работы был рассмотрен шифр AES, были исследованы характеристики шифра AES и финалистов конкурса AES (MARS, RC6, Serpent и Twofish), а также была изучена атака предсказанием дополнения.

1. Шифр AES и финалисты конкурса AES:
   1. При помощи демонстрационного примера была изучена работа шифра AES (Rijndael). Было определено, что AES является симметричным блочным шифром, использующим структуру «квадрат» и SP-сеть. Размер блока составляет 128 бит, размер ключа может составлять 128, 196 и 256 бит. В зависимости от размера ключа, количество раундов может составлять 10, 12 и 14 раундов соответственно. В основе выполнения операций над байтами используется поле Галуа. Каждый раунд состоит из 4 различных обратимых преобразований: слой подстановок, слой линейного перемешивания строк, слой линейного перемешивания столбцов, слой рандомизации. Для расшифрования операции производятся в обратном порядке;
   2. Было выполнено сравнение финалистов конкурса AES, а именно шифров AES, MARS, RC6, Serpent и Twofish, по значениям энтропии текста и времени проведения атаки «грубой» силы при различном количестве известных байт ключа. Значения энтропии для рассматриваемых шифров примерно равны между собой (AES – 6.41, MARS – 6.46, RC6 – 6.46, Serpent – 6.27, Twofish – 6.37). Рассматриваемые шифры показали значения времени проведения атаки «грубой силы» схожих порядков, наиболее криптостойким к атаке «грубой силы» оказался шифр Serpent, шифры AES и RC6 показали почти одинаковые результаты, при этом они показали наименьшее время атаки, между ними в порядке увеличения времени проведения атаки идут шифры MARS и Twofish.
2. Атака «грубой силы» и атака предсказанием дополнения на AES:
   1. Были проведены атаки «грубой силы» при различном количестве известных байт ключа и используемых ядер процессора, используя в качестве оценочной функции энтропию и знание части открытого текста. Было определено, что использование в качестве оценочной функции знание части открытого текста ускоряет проведение атаки примерно в 1.6 раз, а увеличение количества используемых процессов нелинейно уменьшает время проведения атаки;
   2. Была изучена и проведена атака предсказанием дополнения на шифр AES. В основе атаки лежит возможность перехвата и изменения блоков шифротекста, а также получения информации о корректности дополнения в последнем блоке шифротекста. В процессе проведения атаки потребовалось совершить 422 обращения к серверу, чтобы корректно расшифровать последний блок шифротекста.

Были получены практические навыки работы с рассматриваемыми шифрами с использованием приложения CrypTool 1. Были получены практические навыки осуществления атак «грубой силы» и атаки предсказанием дополнения на шифр AES с использованием приложения CrypTool 2.