**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИБ**

отчет

**по лабораторной работе №5**

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

**Тема:** **Изучение шифра AES**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8383 |  | Сырцова Е.А. |
| 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.

**Выполнение работы**

# 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:
3. Открытый текст – фамилия\_имя (транслитерация латиницей)
4. Ключ – номер группы\_отчество
5. Проверить полученные результаты с помощью приложения-инспектора: *Indiv.Procedures->Visualization…->AES->Rijndael Inspector.*
6. 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. **Описание AES с примерами скриншотов из демоприложенияи**

Шифр AES (Rijndael) работает на основе перестановочно-подстановочной сети (SP-сеть). Обобщенная схема работы алгоритма представлена на рис. 1.

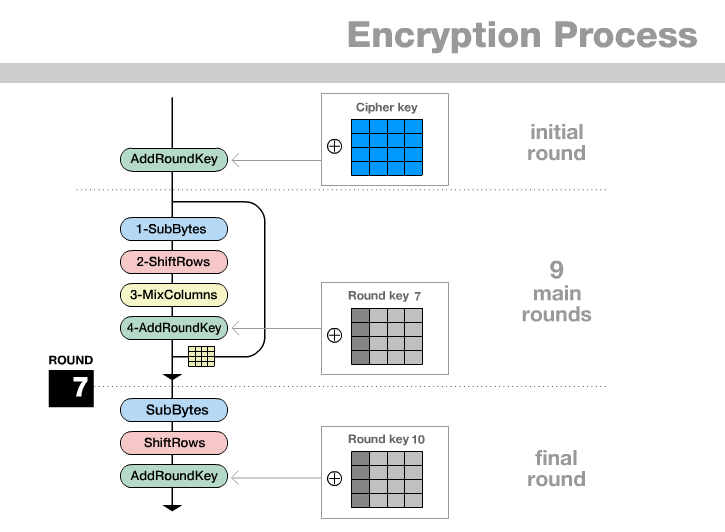


Рисунок 1 – Спецификация AES

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 4х4 байт. Примеры матриц блока открытого текста и ключа представлены на рис. 2.

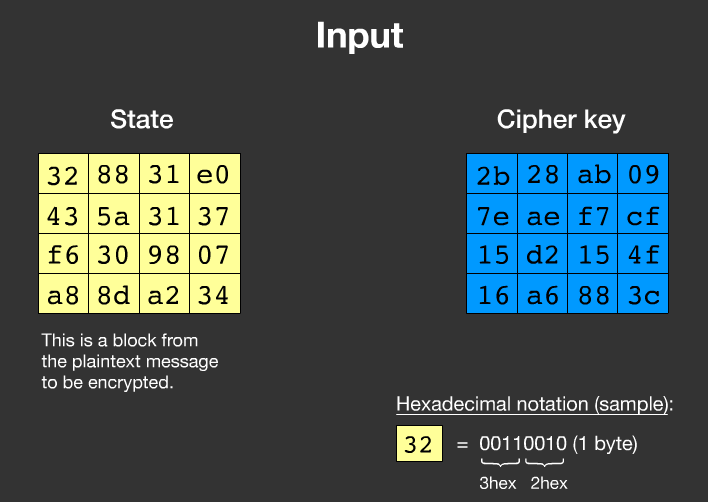
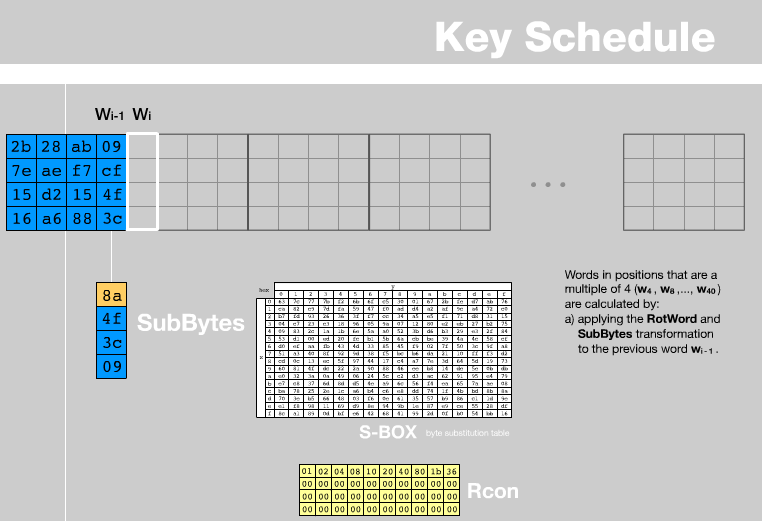
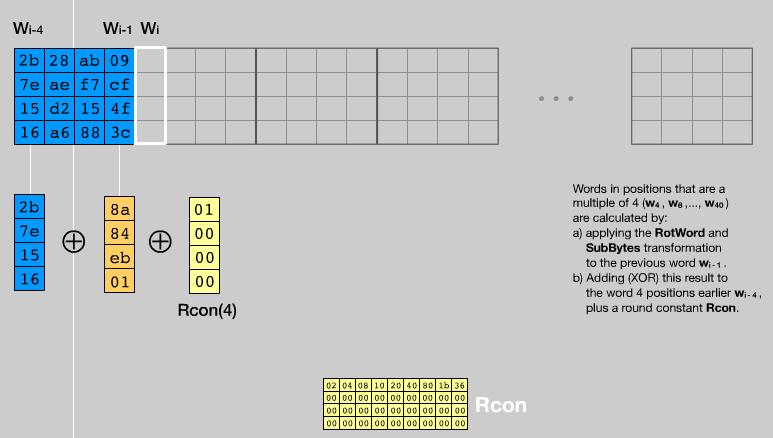
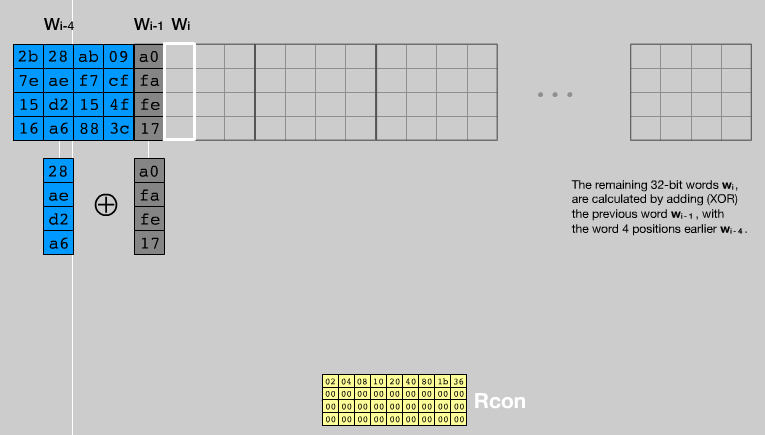


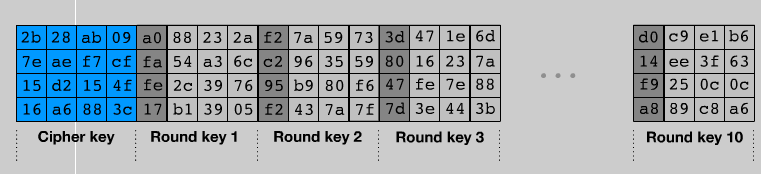
Рисунок 2 – Блок открытого текста и ключ

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



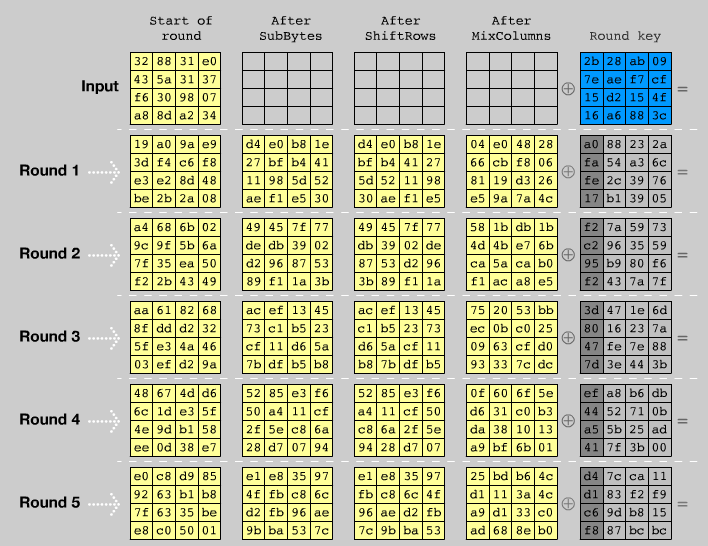




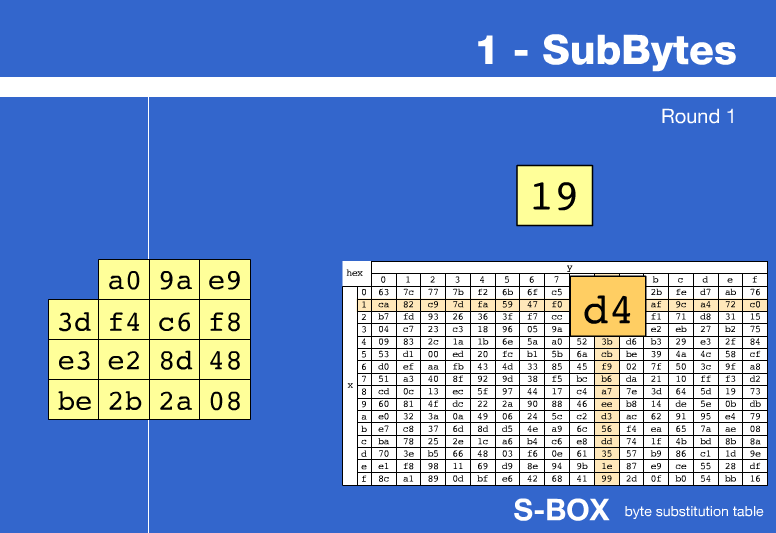


После того, как сформированы раундовые ключи, начинается раундовая обработка матрицы состояний. В каждом раунде алгоритма выполняются следующие преобразования:

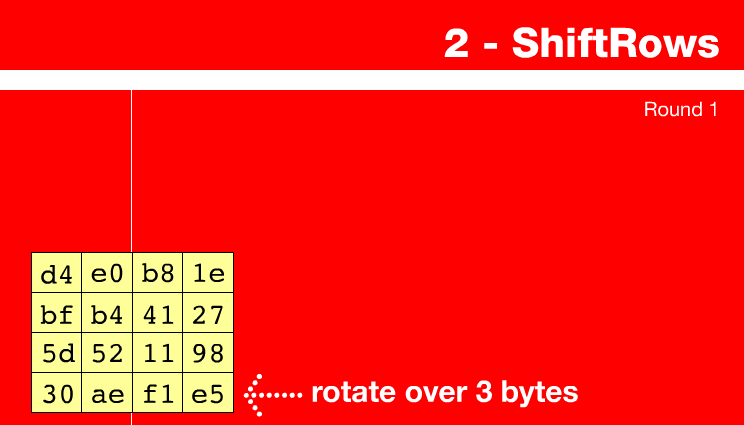
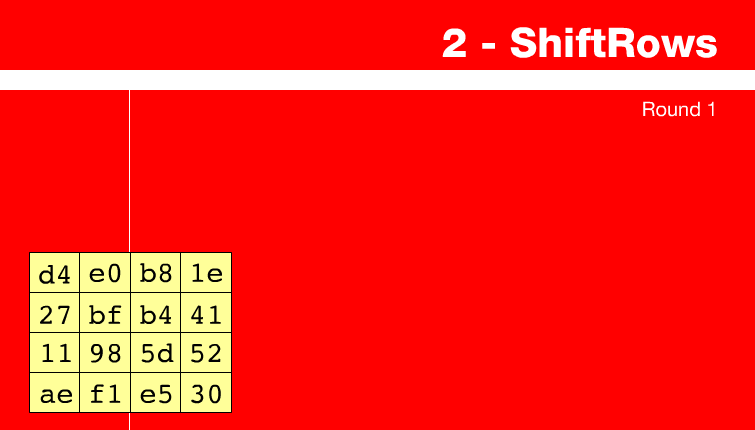
1. Столбцы матрицы состояний складываются с ключом шифра операцией xor.



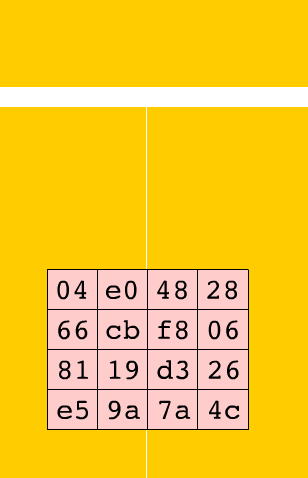
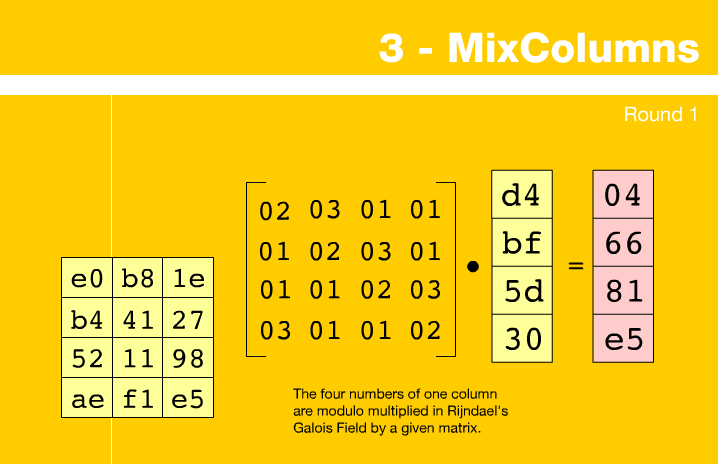
1. Полученная матрица состояний проходит через преобразование подстановки SubBytes.

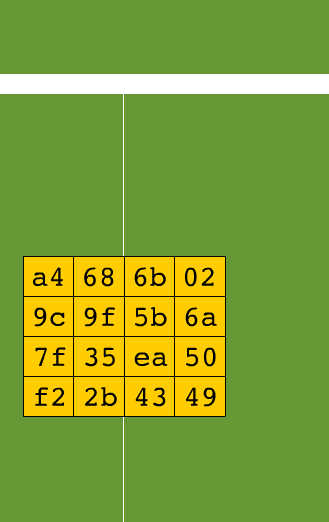
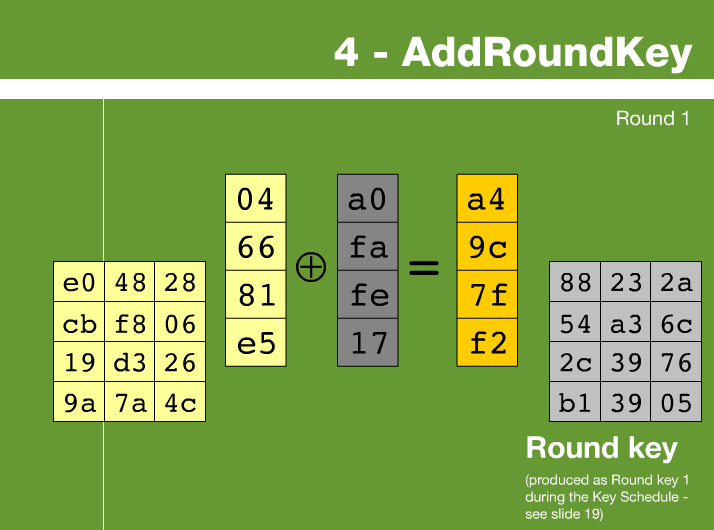


1. Циклический сдвиг влево всех строк матрицы состояний выполняется преобразованием ShiftRows.



1. Смешивание столбцов матрицы состояний путем ее умножения на матрицу констант в конечном поле 𝐺𝐹(28) выполняет преобразовние MixColumn, а сложение полученных столбцов матрицы состояний с раундовым ключом операцией xor – преобразование AddRoundKey





1. Действия 2-5 повторяются в каждом раунде за исключение последнего
2. Последний раунд не включает в себя смешивание столбцов. Расшифровывание выполняется применением обратных операций и раундовых ключей в обратной последовательности.



* 1. **Расчет матрицы состояний и раундового ключа шифра для одного раунда**

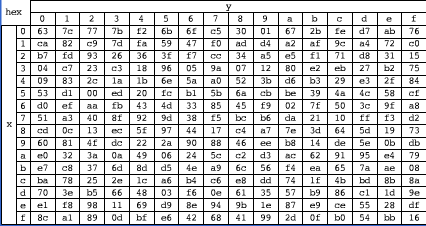
Зашифруем вручную открытый текст, содержащий фамилию\_имя (транслитерация латиницей) - «SyrtsovaEkaterina», с ключом номер группы\_отчество «8383Andreevna».

Представление открытого текста и ключа в матричном виде:

State Key

Применим к столбцам матрицы состояний и ключу шифра операцию XOR и получим следующую матрицу состояний:

К полученному тексту применим преобразование подстановки SubBytes с помощью таблицы замен S-BOX:



Произведем циклический сдвиг влево всех строк матрицы состояний на количество байт равное номеру строки преобразованием ShiftRows:

Применим преобразование MixColumns – смешаем столбцы матрицы состояний путем ее умножения на матрицу констант в конечном поле.

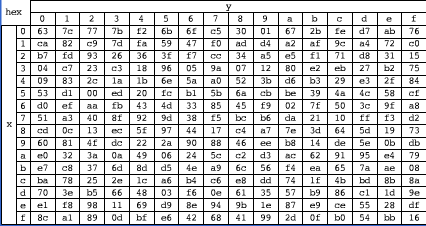
Столбец  по модулю , столбец – как полином с коэффициентами над GF (28).

Рассмотрим расчет первого столбца:

Для первого байта в итоговом первом столбце:

Рассчитав аналогично другие байты для первого столбца и оставшиеся три столбца, получим матрицу состояний после применения MixColumns:

Рассчитаем первый раундовый ключ. Для расчета первого слова раундового ключа применим сначала трансформацию SubBytes и RotWord:



Затем применим XOR к слову, стоящему на 4 позиции раньше, и к раундовой константе Rcon(1):

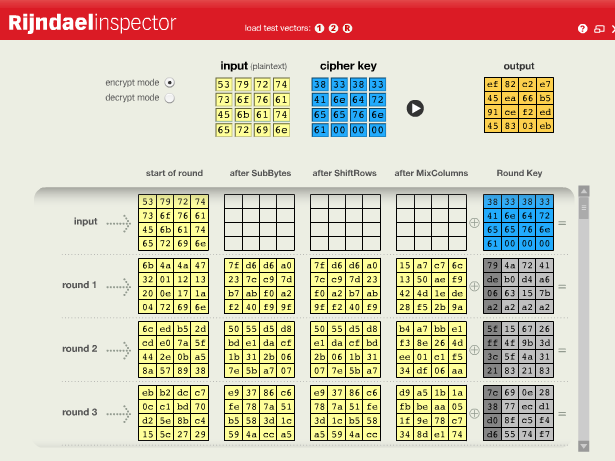
Для расчета второго слова раундового ключа применим XOR к предыдущему слову и слову, стоящему на 4 позиции ранее:

Рассчитав аналогично третье и четвертое слово, получим матрицу первого раундового ключа:

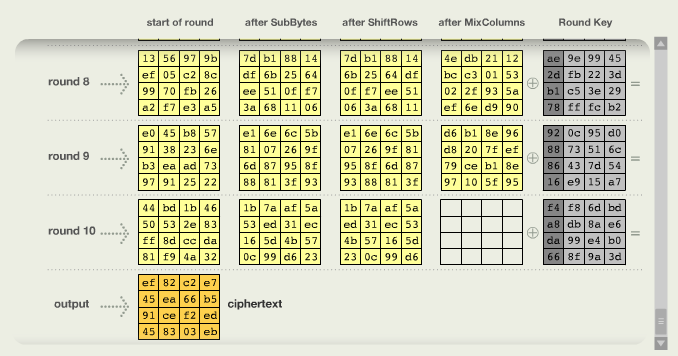
Теперь применим XOR к текущей матрице состояний и первому раундовому ключу, завершив тем самым первый раунд алгоритма AES:

* 1. **Скриншоты приложения-инспектора, подтверждающие корректность расчетов**

Зашифруем текст из предыдущего пункта в приложении CrypTool1 Rijndael inspector. Полученный результат для первого раунда шифрования совпал с результатом, полученным при зашифровке вручную.







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

**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

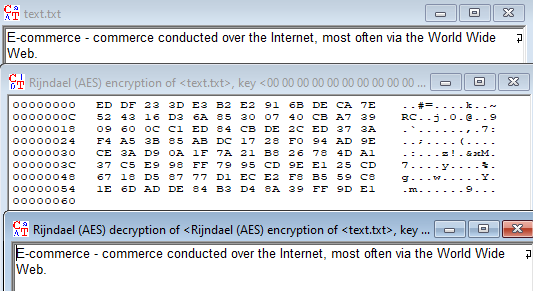
3. С помощью Cryptool 1 зашифровать c ключом отличным от 0 текст с использованием шифров AES, MARS, RC6, Serpent и Twofish

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

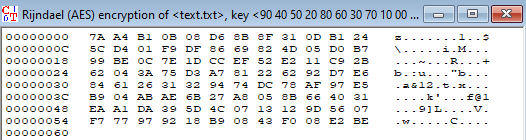
5. Приложением из Cryptool 1 оцените время проведения атаки «грубой силы» всех шифров для одного и того же шифротекста в случаях, когда известно n-2, n-4, n-6, ..., 2 байт секретного ключа. Зафиксировать результаты измерений в таблице.

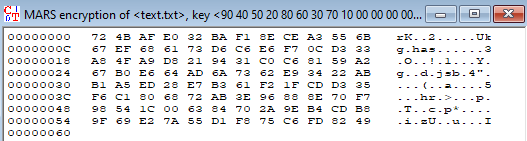
**2.2 Исходные данные для экспериментов**

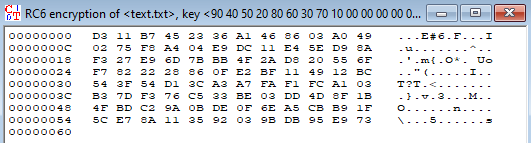
Создадим бинарный файл с произвольно выбранным текстом размером не более 120 символов, зашифруем и расшифруем этот текст шифром AES на 0-м ключе.

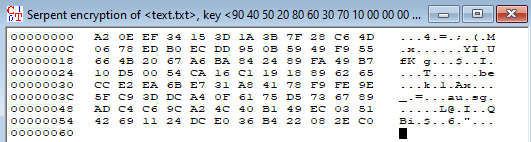


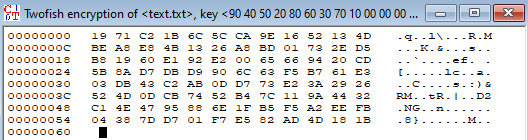
Теперь зашифруем этот текст c ключом «90 40 50 20 80 60 30 70 10 00 00 00 00 00 00 00» с использованием шифров AES, MARS, RC6, Serpent и Twofish.











**2.3 Таблица с результатами качества зашифрования исследованными шифрами**

С помощью приложения из Cryptool 1 вычислим энтропию исходного текста и полученных шифротекстов. Результаты измерений зафиксированы в табл. 1.

Таблица 1 – Результаты измерений энтропии

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Исходный текст | AES | MARS | RC6 | Serpent | Twofish |
| Энтропия | 3,72 | 6,25 | 6,26 | 6,20 | 6,27 | 6,16 |

**2.4 Таблица с результатами трудоемкости атаки «грубой силы» для исследованных шифров**

Теперь для этого шифротекста оценим время проведения атаки «грубой силы» в случаях, когда известно n-2, n-4, n-6, n-8, …, 2 байт секретного ключа. Результаты оценки для использованных шифров зафиксированы в табл. 2-6.

Таблица 2 – Оценка времени проведения атаки «грубой силы» на шифр AES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Число известных байт | | | | | | |
| 14 | 12 | 10 | 8 | 6 | 4 | 2 |
| Оценка времени проведения атаки «грубой силы» | 0-0,5 сек. | 1,5 часа | 11 лет | 7,2\*105 лет | 4,8\*1010 лет | 3,1\*1015 года | 2\*1020 года |

Таблица 3 – Оценка времени проведения атаки «грубой силы» на шифр MARS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Число известных байт | | | | | | |
| 14 | 12 | 10 | 8 | 6 | 4 | 2 |
| Оценка времени проведения атаки «грубой силы» | 0-0,5 сек. | 2 часа 22 минут | 18 лет | 1,2\*106 года | 7,7\*1010 лет | 5\*1015 года | 3,3\*1020 года |

Таблица 4 – Оценка времени проведения атаки «грубой силы» на шифр RC6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Число известных байт | | | | | | |
| 14 | 12 | 10 | 8 | 6 | 4 | 2 |
| Оценка времени проведения атаки «грубой силы» | 0-0,5 сек. | 1,5 часа | 11 лет | 7,3\*105 лет | 4,8\*1010 лет | 3,2\*1015 года | 2,1\*1020 года |

Таблица 5 – Оценка времени проведения атаки «грубой силы» на шифр Serpent

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Число известных байт | | | | | | |
| 14 | 12 | 10 | 8 | 6 | 4 | 2 |
| Оценка времени проведения атаки «грубой силы» | 0-0,5 сек. | 4 часа 20 минут | 33 года | 2,1 \*106 года | 1,4\*1011 года | 9,3\*1015 лет | 6\*1020 лет |

Таблица 6 – Оценка времени проведения атаки «грубой силы» на шифр Twofish

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Число известных байт | | | | | | |
| 14 | 12 | 10 | 8 | 6 | 4 | 2 |
| Оценка времени проведения атаки «грубой силы» | 0-0,5 сек. | 2 часа 42 минуты | 20 лет | 1,4\*106 года | 1,1\*1011 лет | 5,7\*1015 лет | 3,7\*1020 года |

Как видно из таблиц, чем больше размер известной части ключа, тем меньше времени требуется на атаку методом грубой силы. По результатам атаки на разные шифры, можно сделать вывод, что в случае шифра Serpent времени на атаку требуется больше, чем для всех остальных шифров, следовательно, шифр Serpent более устойчив.

1. **Атака «грубой силы» на AES**

**3.1 Задание**

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

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 34 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**

**3.3 Шаблон атаки «грубой силы» из Cryptool 2**

**3.4 Таблица с результатами трудоемкости энтропийной атаки «грубой силы» для различных вариантов знаний о ключе и количестве задействованных процессорных ядер**

**3.5 Таблица с результатами трудоемкости текстовой атаки «грубой силы» для различных вариантов знаний о ключе и количестве задействованных процессорных ядер**