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.8382 _____ Нечепуренко Н.А.
Преподаватель _____ Племянников А.К.

Санкт-Петербург
2021

Цели работы.

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

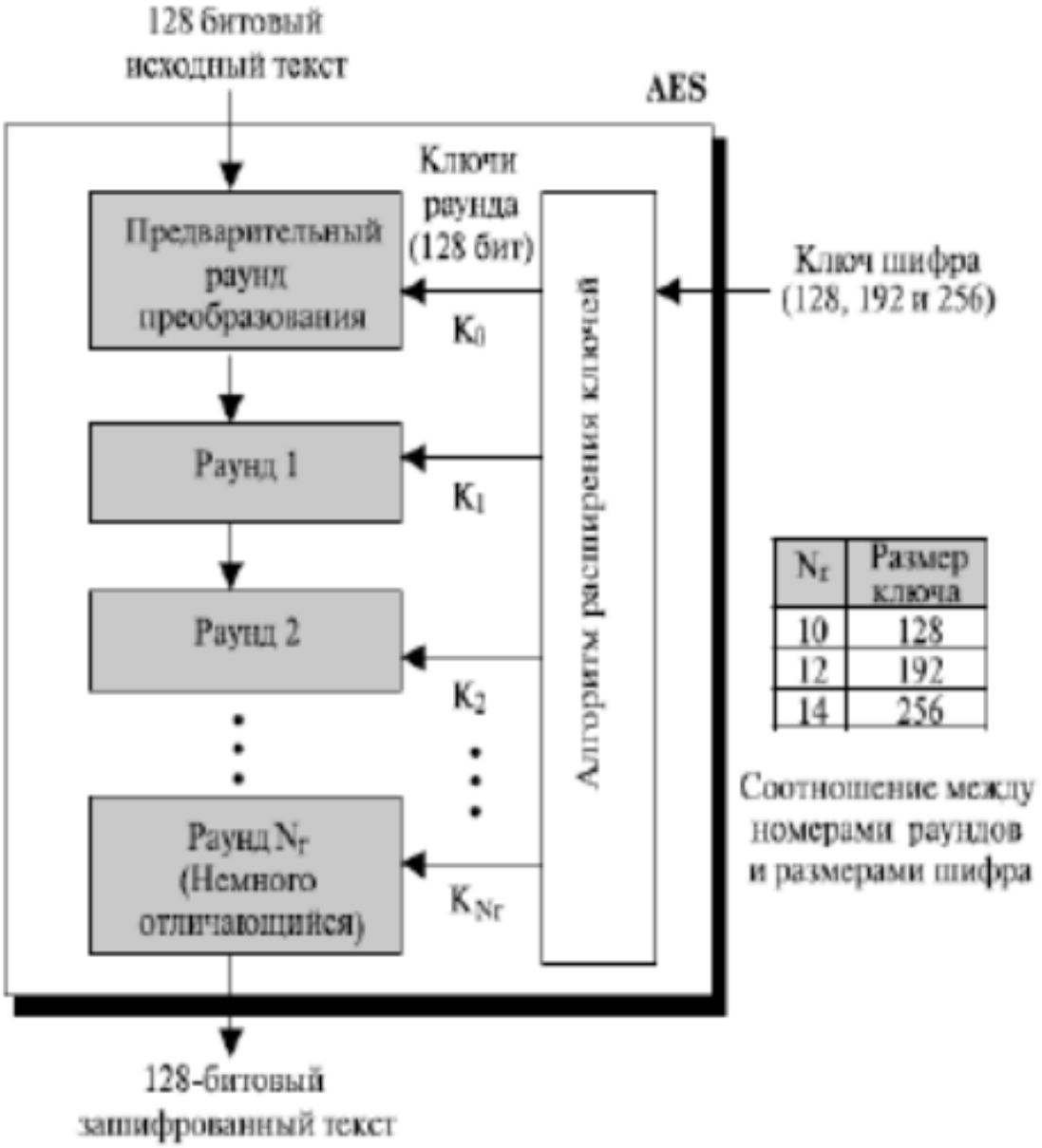
Исследование преобразований AES.

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:
    - Открытый текст – фамилия_имя (транслитерация латиницей)
    - Ключ – номер группы_отчество
3. Проверить полученные результаты с помощью приложенияинспектора: 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

Описание преобразований AES.

Шифр AES (Rijndael) работает на основе перестановочноподстановочной сети (SP-сеть). Обобщенная схема работы алгоритма представлена на рисунке 1.



N_r	Размер ключа
10	128
12	192
14	256

Соотношение между номерами раундов и размерами шифра

Рисунок 1 – Схема преобразований 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 2:

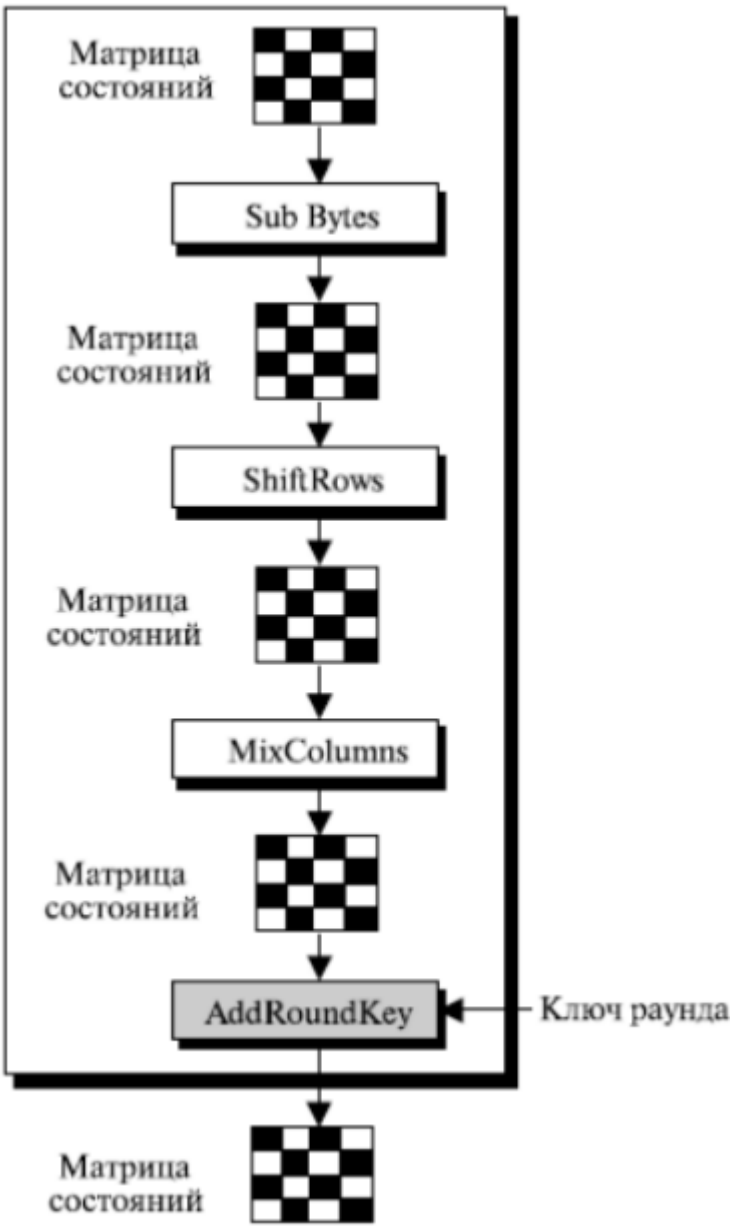


Рисунок 2 – Схема раунда AES

1. Столбцы матрицы состояний складываются с ключом шифра операцией xor.
2. Полученная матрица состояний проходит через преобразование подста-

новки 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 2-5 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.

Ручное преобразование для одного раунда AES.

Выполним ручной расчет для первого раунда AES.

В качестве открытого текста возьмем NECHEPURENKO_NIK, в качестве ключа – 8382_ALEKSANDROV. Для простоты и сообщение и ключ имеют длину в 128 бит.

В матричном виде открытый текст и ключ представляются следующим образом

$$\text{TEXT} = \begin{pmatrix} 4e & 45 & 45 & 5f \\ 45 & 50 & 4e & 4e \\ 43 & 55 & 46 & 49 \\ 48 & 52 & 4f & 46 \end{pmatrix}$$

$$\text{KEY} = \begin{pmatrix} 38 & 5f & 46 & 44 \\ 33 & 41 & 53 & 52 \\ 38 & 4c & 41 & 4f \\ 32 & 45 & 4e & 56 \end{pmatrix}$$

На 0 раунде посчитаем поэлементный xor матрицы и ключа.

$$\text{TEXT}_1 = \begin{pmatrix} 4e \oplus 38 & 45 \oplus 5f & 45 \oplus 46 & 5f \oplus 44 \\ 45 \oplus 33 & 50 \oplus 41 & 4e \oplus 53 & 4e \oplus 52 \\ 43 \oplus 38 & 55 \oplus 4c & 46 \oplus 41 & 49 \oplus 4f \\ 48 \oplus 32 & 52 \oplus 45 & 4f \oplus 4e & 46 \oplus 56 \end{pmatrix} = \begin{pmatrix} 76 & 1a & 03 & 1b \\ 76 & 11 & 1d & 1c \\ 7b & 19 & 07 & 09 \\ 7a & 17 & 01 & 19 \end{pmatrix}$$

Далее необходимо выполнить замену, согласно таблице на рисунке 3.

AES S-box

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Рисунок 3 – Rijndael S-box

Получаем

$$\text{TEXT}_{\text{SB}} = \begin{pmatrix} 38 & a2 & 7b & af \\ 38 & 82 & a4 & 9c \\ 21 & d4 & c5 & 01 \\ da & f0 & 7c & d4 \end{pmatrix}$$

Затем каждую строку $i \in 0..3$ сдвинем циклически влево на i.

$$\text{TEXT}_{\text{SR}} = \begin{pmatrix} 38 & a2 & 7b & af \\ 82 & a4 & 9c & 38 \\ c5 & 01 & 21 & d4 \\ d4 & da & f0 & 7c \end{pmatrix}$$

После этого выполним этап MixColumns, умножив полученную матрицу слева в $GF(2^8)$

$$\text{TEXT}_{\text{MC}} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 38 & a2 & 7b & af \\ 82 & a4 & 9c & 38 \\ c5 & 01 & 21 & d4 \\ d4 & da & f0 & 7c \end{pmatrix}$$

Умножение матриц в $GF(2^8)$ выглядит следующим образом, возьмем первую строку и первый столбец:

$$(2 \cdot 38) \oplus (3 \cdot 82) \oplus (1 \cdot c5) \oplus (1 \cdot d4)$$

Вычислим слагаемые отдельно, $+$ обозначает xor для простоты.

$$2 \cdot 38 = x(00111000) = x(x^3 + x^4 + x^5) = x^6 + x^5 + x^4 = 01110000 = 70$$

$$3 \cdot 82 = (x+1)(10000010) = (x+1)(x^7 + x) = x^8 + x^2 + x^7 + x$$

Заменим x^8 на $x^4 + x^3 + x + 1$, чтобы остаться в поле.

$$3 \cdot 82 = x^4 + x^3 + x + 1 + x^2 + x^7 + x = x^7 + x^4 + x^3 + x^2 + 1 = 10011101 = 9d$$

$$1 \cdot c5 = 10100101 = c5$$

$$1 \cdot d4 = 11010100 = d4$$

Теперь вычислим xor 4 слагаемых $70 \oplus 9d \oplus c5 \oplus d4 = fc$.

Элемент первой строки и первого столбца результирующей матрицы равен fc. Проводя аналогичные вычисления получаем

$$\text{TEXT}_{\text{MC}} = \begin{pmatrix} fc & 73 & 98 & a5 \\ a7 & 28 & cb & c4 \\ 4c & 71 & ae & a0 \\ bc & f7 & cb & fe \end{pmatrix}$$

Теперь необходимо вычислить ключ для первого раунда. Обозначим столбцы матрицы ключа как $\omega_0, \omega_1, \omega_2, \omega_3$.

Тогда столбцы матрицы ключа первого раунда могут быть выражены как

$$\omega_4 = \omega_0 \oplus (RC_1 \oplus SB(ROTL(\omega_3)))$$

$$\omega_5 = \omega_1 \oplus \omega_4$$

$$\omega_6 = \omega_2 \oplus \omega_5$$

$$\omega_7 = \omega_3 \oplus \omega_6$$

где $ROTL$ – циклический сдвиг влево на 1, SB – подстановка S-блоков, RC_i – раундовая константа, для первого раунда равна $\begin{pmatrix} 01 & 00 & 00 & 00 \end{pmatrix}$.

Вычислим ω_4.

$$ROTL(\omega_3) = ROTL(\begin{pmatrix} 44 & 52 & 4f & 56 \end{pmatrix}) = \begin{pmatrix} 52 & 4f & 56 & 44 \end{pmatrix}$$

$$SB(ROTL(\omega_3)) = \begin{pmatrix} 00 & 84 & b1 & 1b \end{pmatrix}$$

$$RC_1 \oplus SB(ROTL(\omega_3)) = \begin{pmatrix} 01 \oplus 00 & 00 \oplus 84 & 00 \oplus b1 & 00 \oplus 1b \end{pmatrix} = \begin{pmatrix} 01 & 84 & b1 & 1b \end{pmatrix}$$

$$\omega_4 = \omega_1 \oplus RC_1 \oplus SB(ROTL(\omega_3)) = \begin{pmatrix} 38 & 33 & 38 & 32 \end{pmatrix} \oplus \begin{pmatrix} 01 & 84 & b1 & 1b \end{pmatrix}$$

$$\omega_4 = \begin{pmatrix} 39 & b7 & 89 & 29 \end{pmatrix}$$

Остальные столбцы получаются простым xor'ом.

Ключ первого раунда имеет следующий вид:

$$\text{KEY}_1 = \begin{pmatrix} 39 & 66 & 20 & 64 \\ b7 & f6 & a5 & f7 \\ 89 & c5 & 84 & cb \\ 29 & 6c & 22 & 74 \end{pmatrix}$$

Таким образом, блок исходного текста после первого раунда вычисляется как

$$\text{TEXT}_2 = \text{TEXT}_{\text{MC}} \oplus \text{KEY}_1 = \begin{pmatrix} fc \oplus 39 & 73 \oplus 66 & 98 \oplus 20 & a5 \oplus 64 \\ a7 \oplus b7 & 28 \oplus f6 & cb \oplus a5 & c4 \oplus f7 \\ 4c \oplus 89 & 71 \oplus c5 & ae \oplus 84 & a0 \oplus cb \\ bc \oplus 29 & f7 \oplus 6c & cb \oplus 22 & fe \oplus 74 \end{pmatrix}$$

$$\text{TEXT}_2 = \begin{pmatrix} c5 & 15 & b8 & c1 \\ 10 & de & 6e & 33 \\ c5 & b4 & 2a & 6b \\ 95 & 9b & e9 & 8a \end{pmatrix}$$

Визуализация преобразований AES в Cryptool 1.

С помощью режима инспектора AES в Cryptool 1 проверим корректность расчетов из пункта выше.

Расчет первых трех раундов представлен на рисунке ниже.

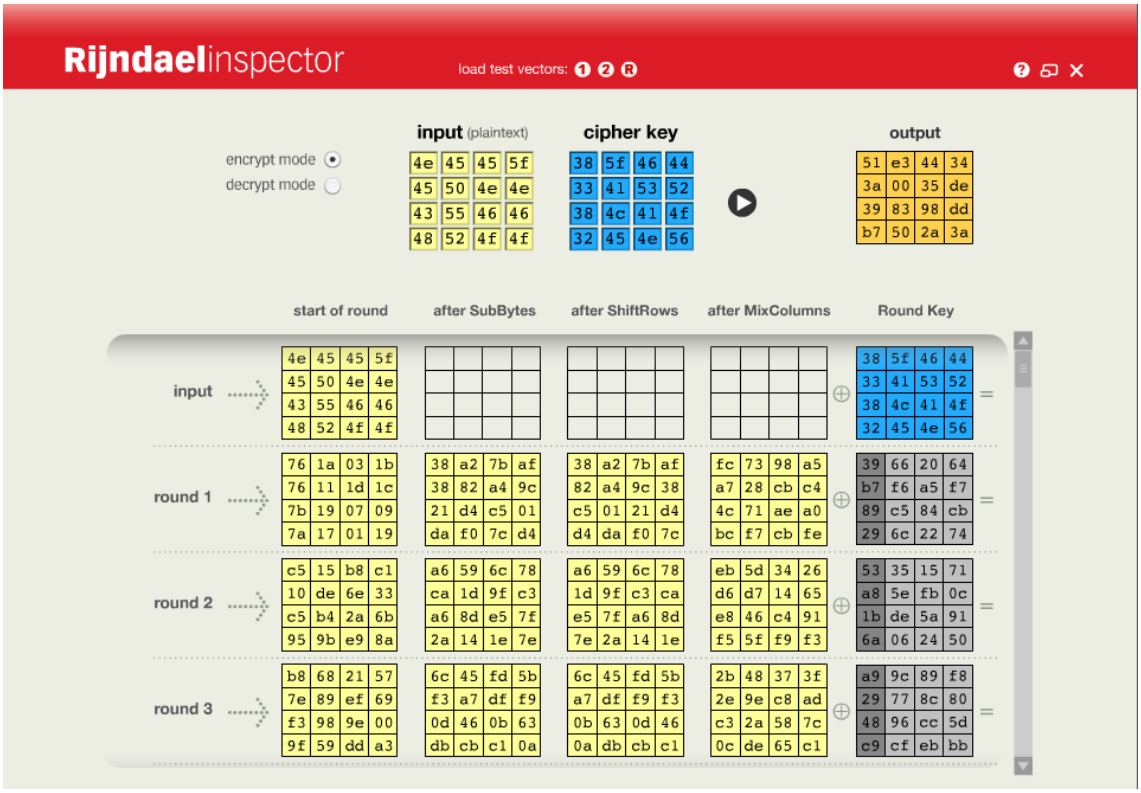


Рисунок 4 – Первые три раунда AES

Убеждаемся, что ручной расчет был корректным.

Расчет последних раундов представлен на рисунке 5.

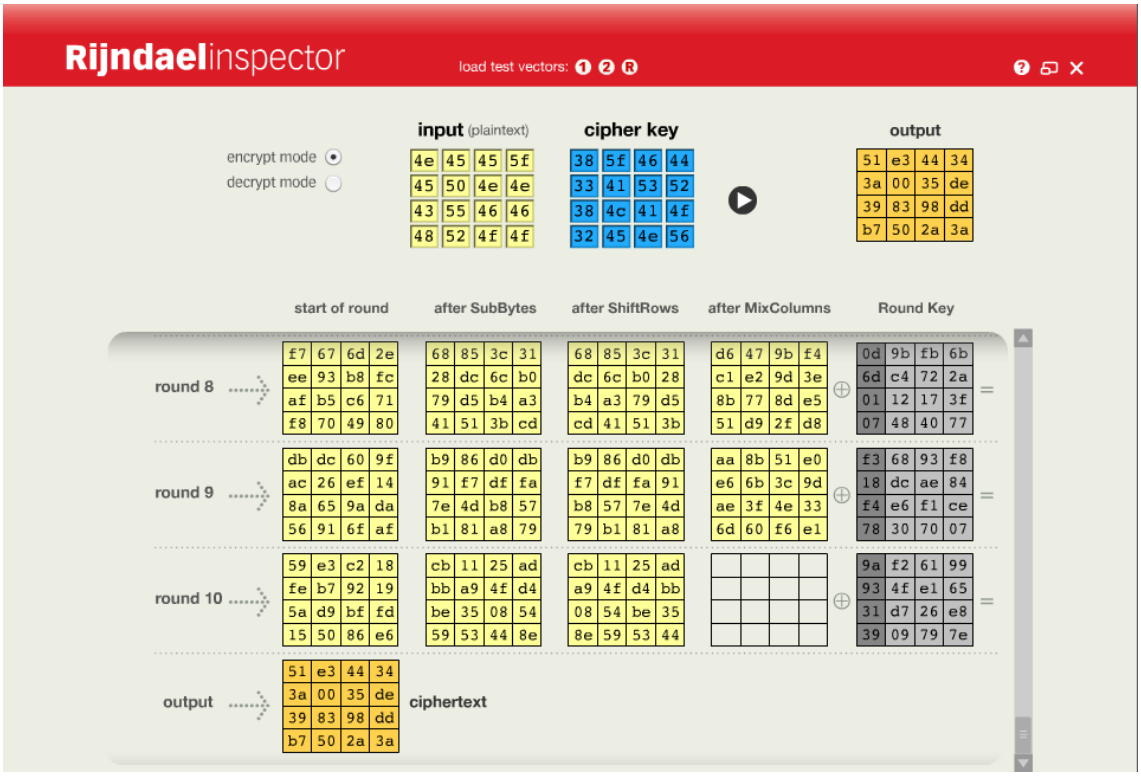


Рисунок 5 – Последние три раунда AES

Рассмотрим потоковую модель шифра AES с помощью демонстрационного приложения из CrypTool 1 для 0-текста и 0-ключа.

Визуализация 1, 5 и 10 раундов приведена на рисунках 6,7 и 8 соответственно.

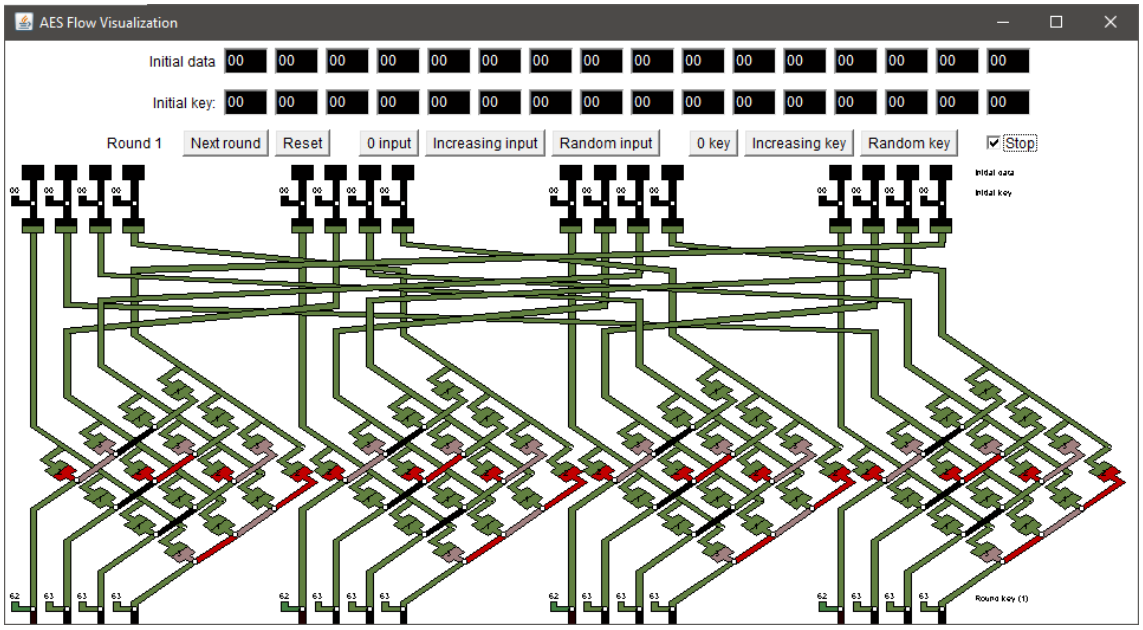


Рисунок 6 – Визуализация первого раунда AES

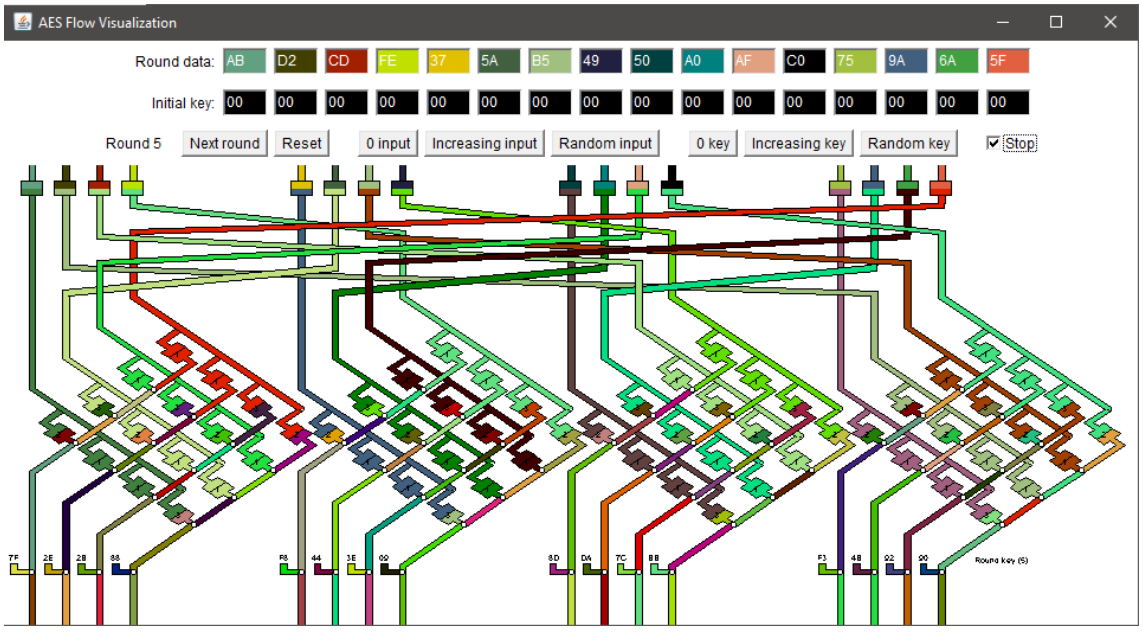


Рисунок 7 – Визуализация пятого раунда AES

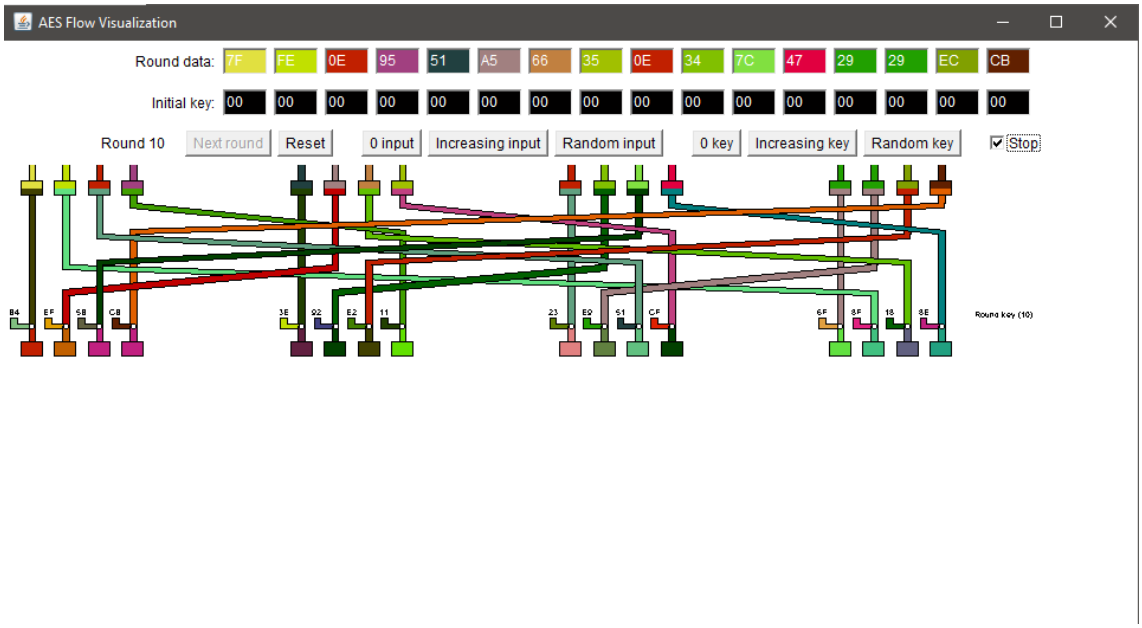


Рисунок 8 – Визуализация последнего раунда AES

Графически можно заметить насколько разнообразными стали цвета шифруемых байтов, что говорит о высокой степени энтропии выходных данных.

Вывод.

В данном разделе были изучены преобразования шифра AES Rijndael. Вручную был произведен один раунд шифрования открытого текста NECHEPURENKO с ключом 8382_ALEKSANDROV.

Полученные результаты были проверены с помощью инспектора AES в программе Cryptool 1. С помощью средства визуализации Cryptool 1 была изучена потоковая модель шифра AES.

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 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.

Сравнение финалистов конкурса AES

Сгенерируем текст-рыбу на 120 знаков. В качестве ключа возьмем 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 00.

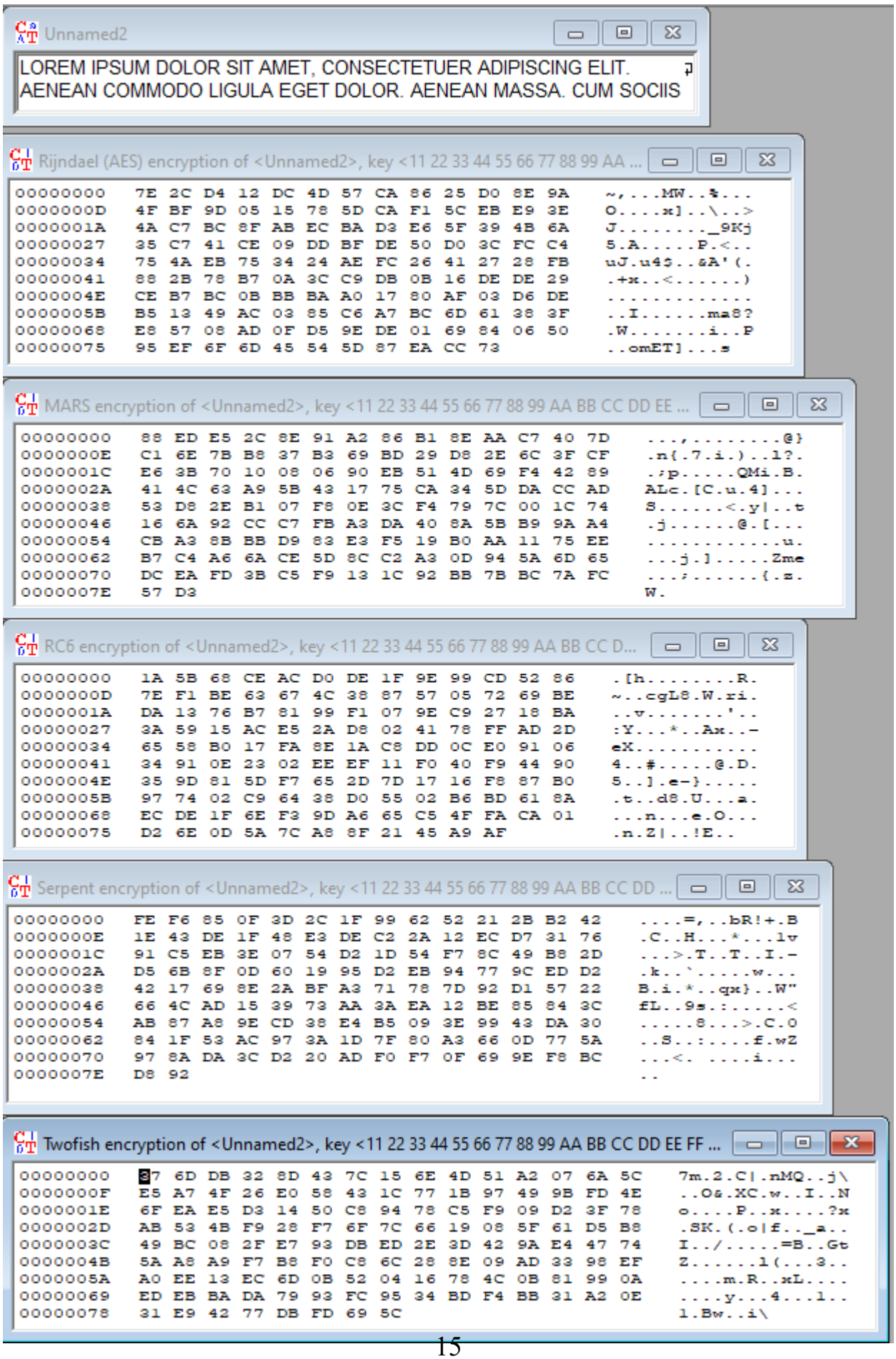


Рисунок 9 – Шифротексты Rijndael, MARS, RC6, Serpent и Twofish

Сравнение энтропии исходного текста и шифротекстов приведено в таблице 1.

Таблица 1 – Анализ энтропии Rijndael, MARS, RC6, Serpent и Twofish

-	Открытый текст	Rijndael	MARS	RC6	Serpent	Twofish
Значение энтропии	3.93	6.55	6.65	6.58	6.49	6.58

Наибольшее значение энтропии для заданного текста и ключа показал алгоритм MARS, наименьшее - Serpent.

С помощью Cryptool 1 выполним оценку времени проведения атаки грубой силы всех шифров. Результаты сравнения приведены в таблице 2.

Таблица 2 – Оценка времени атаки методом грубой силы для Rijndael, MARS, RC6, Serpent и Twofish

Известно байт	Rijndael	MARS	RC6	Serpent	Twofish
14	около секунды	около секунды	около секунды	около секунды	около секунды
8	900000 лет	21000000 лет	9800000 лет	29000000 лет	18000000 лет
4	2.7e15 лет	6e15 лет	4.4e15 лет	1.3e16 лет	7.7e15 лет
2	2.4e20 лет	4.4e20 лет	2.7e20 лет	5.3e20 лет	4.3e20 лет

Вывод.

По результатам исследования, наиболее криптостойким оказался шифр MARS, наименее криптостойким – Rijndael. Стоит оговорить, что оценка времени атаки грубой силы в Cryptool 1 достаточно условна и не может использоваться в качестве достоверного источника. Также необходимо упомянуть, что MARS – самый вычислительно сложный среди рассматриваемых алгоритмов, а Rijndael поддерживается современными процессорами на уровне инструкций.

Атака «грубой силы» на AES.

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 словосочетание DEAR SIRS задействовав 1 ядро процессора. Зафиксировать затраты времени.
7. Выполнить атаку повторно с средним и максимальным количеством про-

цессорных ядер. Зафиксировать затраты времени.

Атака «грубой силы» на AES в Cryptool 2.

Проведем атаку на шифротекст, полученный алгоритмом AES средствами Cryptool 2, рассматривая в качестве минимизируемой функции функцию энтропии. Выберем шаблон «AES Analysis using Entropy(2)», интерфейс программы представлен на рисунке ниже.

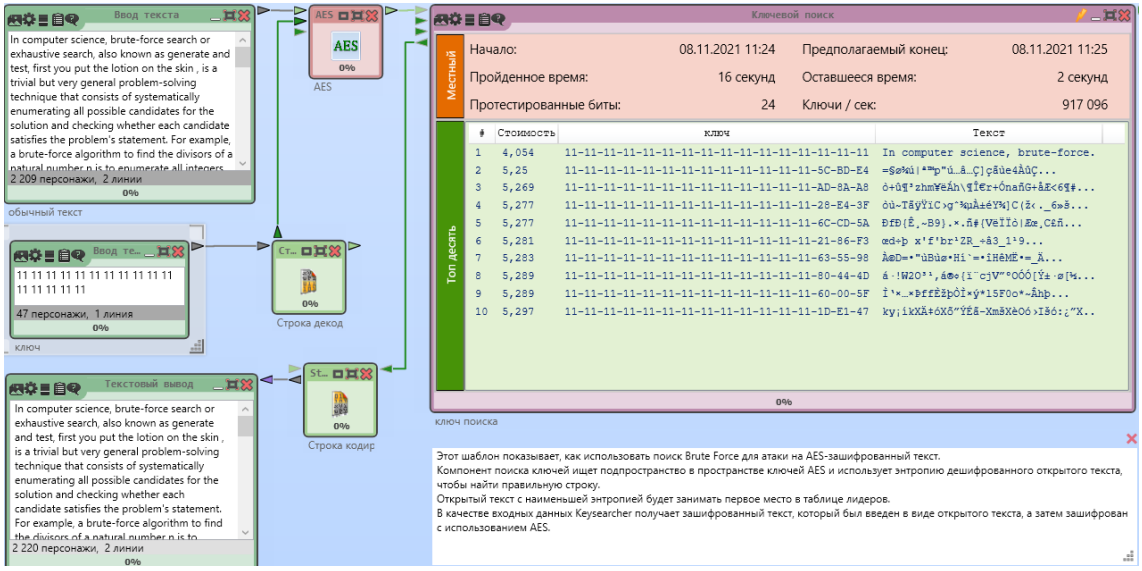


Рисунок 10 – Интерфейс шаблона «AES Analysis using Entropy(2)»

В качестве открытого текста возьмем рыбный текст в примерно 1000 символов. Сгенерированный текст приведен на рисунке 11.

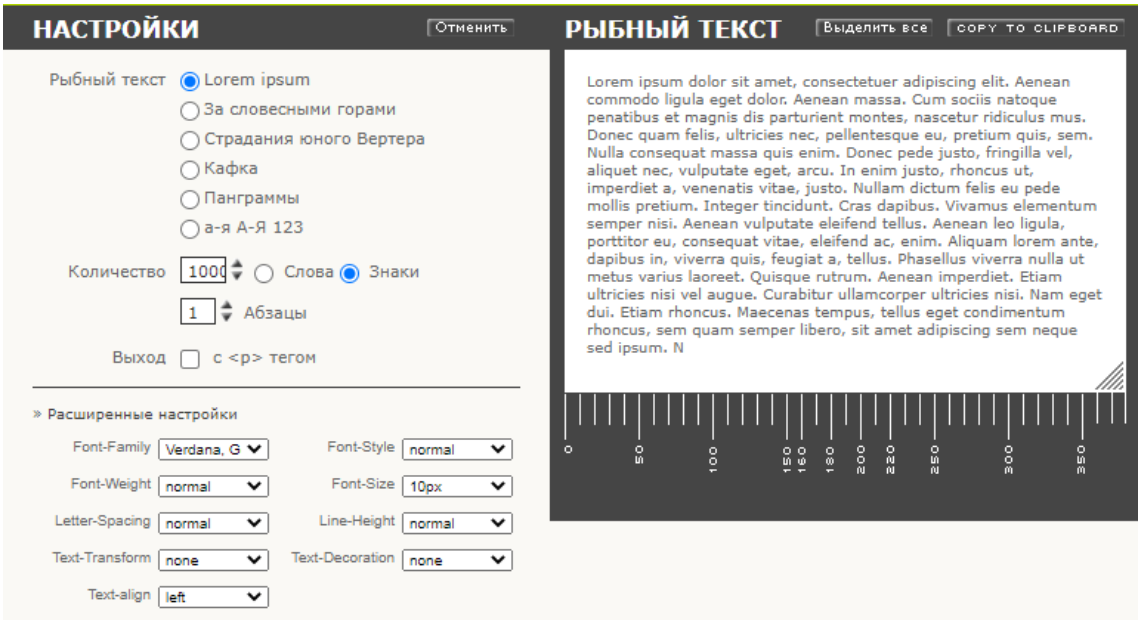


Рисунок 11 – Открытый текст для эксперимента

В качестве ключа будем использовать последовательность единиц.

Проведем атаку на шифротекст, результаты приведены в таблице 3.

Таблица 3 – Атака методом грубой силы в Cryptool 2

Известно байт	Время взлома, 1 ядро	Время взлома, 4 ядра	Время взлома, 8 ядер
14	2 с	2 с	2 с
12	2 ч 15 мин	1 ч 8 мин	35 мин
10	6220 д	2764 д	1650 д

Изменим открытый текст на другой, соответствующий шаблону «DEAR SIRS message THANKS». Новый открытый текст: DEAR SIRS It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using ’Content here, content here’, making it look like readable English THANKS.

Проведем аналогичные измерения. Результаты приведены в таблице 4.

Таблица 3 – Атака методом грубой силы в Cryptool 2 шаблонного текста

Известно байт	Время взлома, 1 ядро	Время взлома, 4 ядра	Время взлома, 8 ядер
14	2 с	2 с	2 с
12	1 ч 24 мин	23 мин	25 мин
10	4519 д	1715 д	1187 д

Вывод.

В данном разделе были изучены две разновидности атаки методом грубой силы на шифр AES. Первый вариант использовал в качестве целевой функции значение энтропии, второй – знание части исходного текста. При использовании энтропии время взлома уменьшается пропорционально логарифму количества ядер, для метода с использованием шаблонного текста зависимость времени от числа ядер более сложная, но было установлено, что использование 4 ядер достаточно, увеличение числа до 8 не дает значительного прироста в производительности.

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:
    - Изучите комментарии к шаблону
    - Изучите публикацию [4]

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(Padding Oracle Attack) в Cryptool 2.

Шаблон атаки Padding Oracle Attack в Cryptool 2 приведен на рисунке ниже.

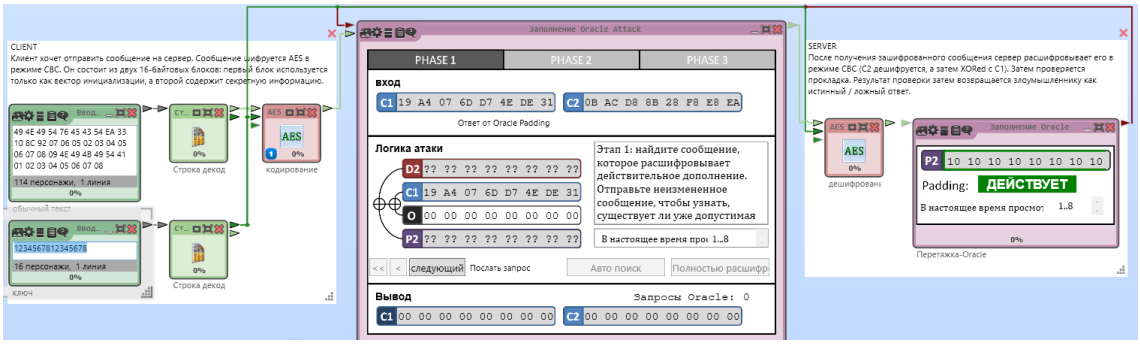


Рисунок 12 – Шаблон атаки Padding Oracle Attack в Cryptool 2

Внедрим имя NIKITA, вернее его шестнадцетеричное представление 4e 49 4b 49 54 41 в открытый текст и получим новый открытый текст: 49 4E 49 54 76 45 43 54 EA 33 10 8C 92 07 06 05 02 03 04 05 06 07 08 09 4E 49 4B 49 54 41 01 02 03 04 05 06 07 08. Ключ шифрования изменять не будем, примем его равным 1234567812345678 (16 символов = 128 бит).

Атака состоит из 3 фаз:

1. Нахождение длины дополнения
2. Подбор дополнения
3. Расшифровка текста

На рисунке 13 приведено состояние интерфейса атаки после первой фазы.

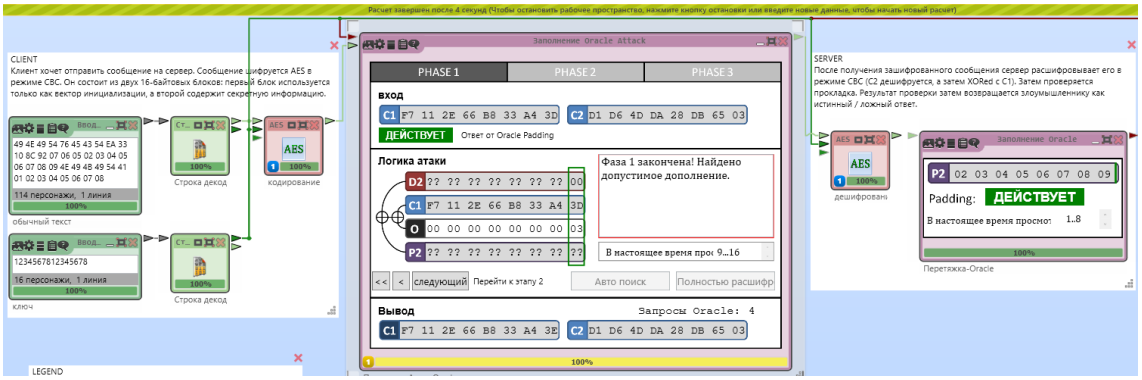


Рисунок 13 – Интерфейс атаки после первой фазы

На этом этапе было найдено допустимое дополнение.

На рисунке 14 приведено состояние интерфейса атаки после второй фазы.

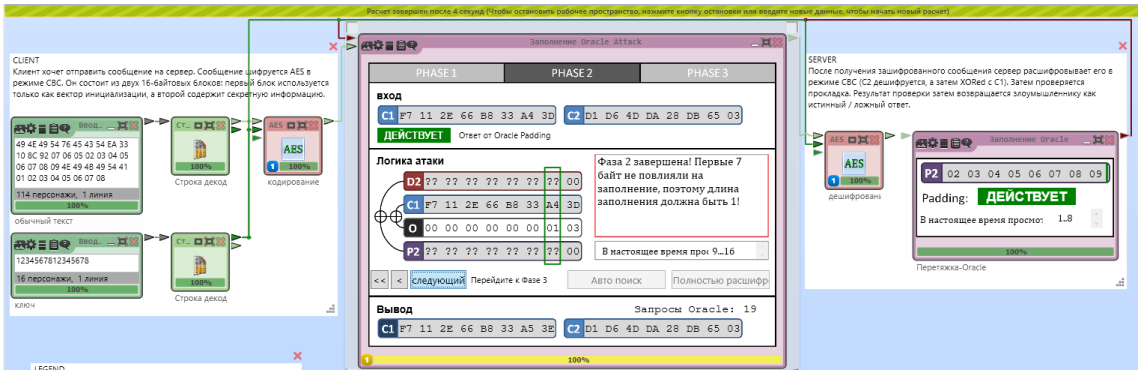


Рисунок 14 – Интерфейс атаки после второй фазы

В результате атаки имя NIKITA было расшифровано (см. рис. 15).

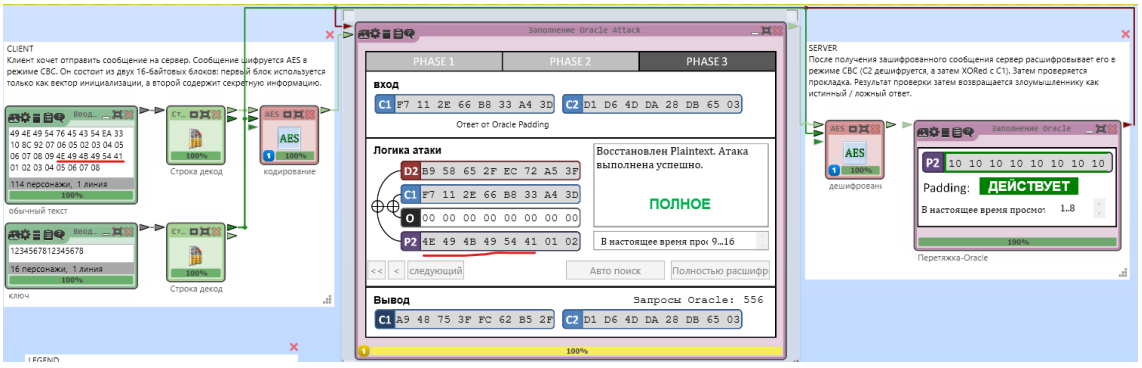


Рисунок 15 – Результат атаки Padding Oracle Attack

Вывод.

В данном разделе был изучен шаблон атаки предсказанием дополнения на шифр AES в режиме CBC(Padding Oracle Attack) в Cryptool 2. В открытый текст было внедрено сообщение с именем NIKITA. В результате атаки на шифротекст было расшифровано внедренное сообщение (см. рис. 15).

Заключение.

В данной работе был рассмотрен блочный симметричный шифр AES Rijndael, работающий на перестановочно-подстановочной сети (SP-сеть). Обратимость сети следует из обратимости линейных операций перестановки, обратимости xor и обратимости операций в поле Галуа $GF(2^8)$. Алгоритм шифрования применяется к блокам текста размером 128 бит, размер ключа может принимать значение 128, 192 или 256 битов. Примеры пошагового ручного расчета преобразований алгоритма приведены в разделе «Исследование преобразований AES».

С помощью алгоритма AES Rijndael был произведен один раунд шифрования открытого текста NECHEPURENKO_NIK с ключом 8382_ALEKSANDROV, расчеты были проверены средствами Cryptool 1.

Было произведено исследование скорости взлома шифра Rijndael и других финалистов конкурса AES (MARS, RC6, Serpent, Twofish) в зависимости от длины известной части ключа. По результатам исследования, наиболее криптостойким оказался шифр MARS, наименее криптостойким – Rijndael. Это исследование оперирует довольно грубыми оценками, полученными с помощью эвристических оценок в Cryptool 1. Также стоит отметить наличие аппаратной реализации инструкция Rijndael и вычислительную сложность алгоритма MARS.

Известно байт	Rijndael	MARS	RC6	Serpent	Twofish
14	около секунды	около секунды	около секунды	около секунды	около секунды
8	900000 лет	21000000 лет	9800000 лет	29000000 лет	18000000 лет

4	2.7e15 лет	6e15 лет	4.4e15 лет	1.3e16 лет	7.7e15 лет
2	2.4e20 лет	4.4e20 лет	2.7e20 лет	5.3e20 лет	4.3e20 лет

В разделе «Атака грубой силы на AES» были рассмотрены две эвристики: уменьшение энтропии текста и сопоставление с известной частью открытого текста. Эмпирически было установлено, что при использовании энтропии время взлома уменьшается пропорционально логарифму количества ядер, для метода с использованием шаблонного текста зависимость времени от числа ядер более сложная, но было установлено, что использование 4 ядер достаточно, увеличение числа до 8 не дает значительного прироста в производительности. Метод с использованием энтропии более универсальный, в большинстве случаев информация об открытом тексте оказывается недостаточной для проведения атаки с использованием шаблонного текста.

С помощью Cryptool 2 была успешно проведена атака предсказанием дополнения на шифр AES в режиме CBC(Padding Oracle Attack). Рассматриваемая атака основывается на том, что открытый текст дополняется до размера блока согласно стандарту PKCS7, и при использовании режима CBC выполняется следующее тождество:

$$I_2 = C_1 \oplus P_2 \Leftrightarrow P_2 = C_1 \oplus I_2$$

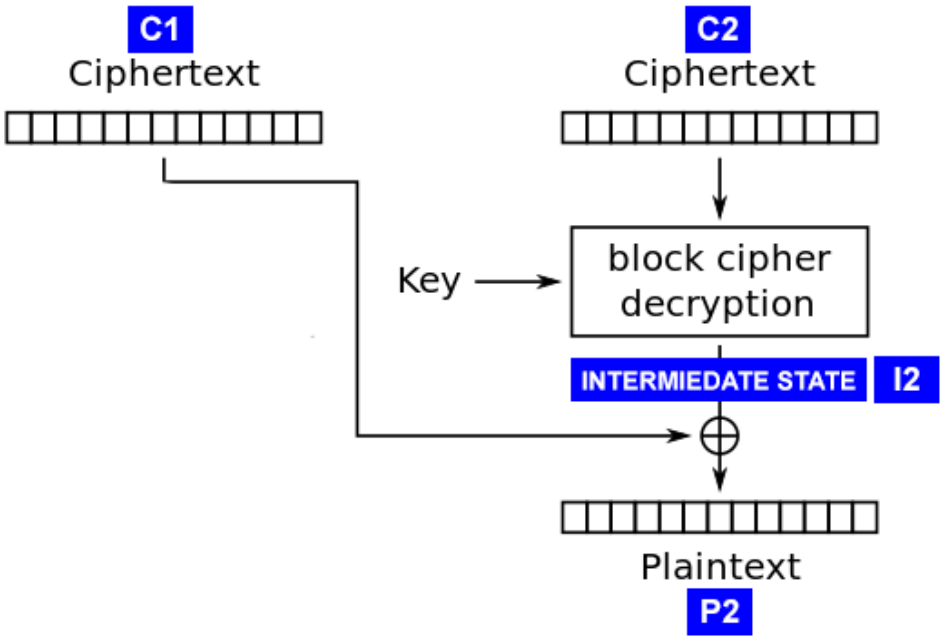


Рисунок 16 – Схема шифрования в режиме CBC