

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

**ЛАБОРАТОРНАЯ РАБОТА №5
по дисциплине «Криптография и защита информации»
Тема: Изучение шифра AES**

Студент гр. 6304

Преподаватель

Корытов П.В.

Племянников А.К.

Санкт-Петербург

2019

Цель работы

Цель работы: исследовать шифр AES, финалистов конкурса AES, атаку предсказанием дополнения и получить практические навыки работы с шифрами и атакой, в том числе и в программном продукте Cryptool 1 и 2.

1. Исследование преобразований AES

1.1. Формулировка задания

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 приложения- инспектора: Indiv.Procedures->Visualization...->AES->Rijndael Inspector.

1.2. Общее описание шифра

Используется не сеть Фейстеля, а Square-like структура.

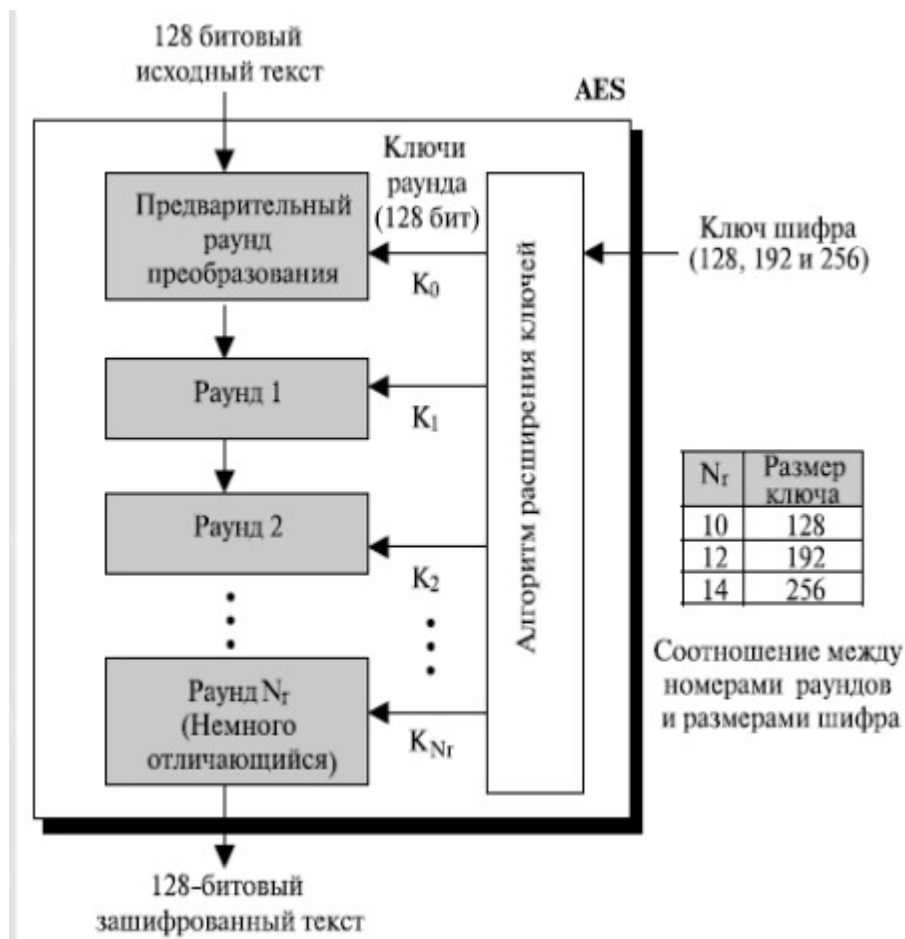


Рисунок 1. Структура шифра

- Операции проводятся над элементами поля Галуа $GF(2^8)$.
Т.е. байту соответствует многочлен $b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$
- Операция умножения выполняется по модулю неприводимого многочлена $x^8 + x^4 + x^3 + x + 1$

Размер блока — 16 байт, размер ключа — 16, 24 или 32 байт. Размер матрицы состояний — 4×4 блока

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

Рисунок 2. Матрица состояний

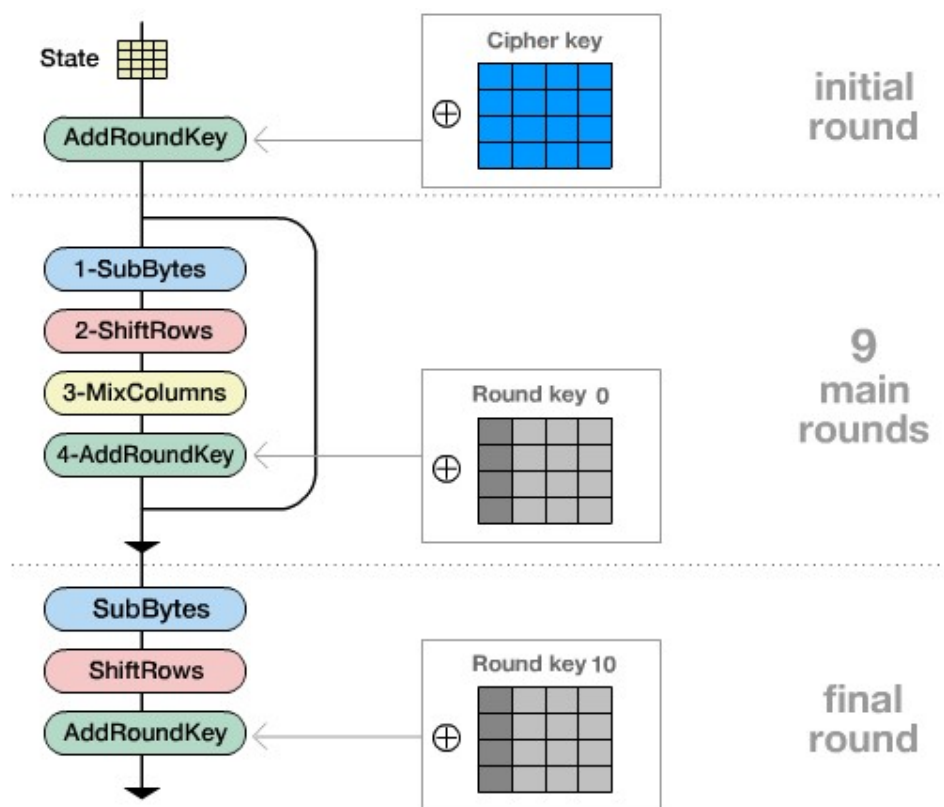


Рисунок 3. Операции шифра AES-192

1.3. Операции шифра

Нижесписанные операции выполняются для всех раундов, кроме MixColumns — она не выполняется для последнего раунда.

1.3.1. SubBytes

Производится нелинейная замена байтов с использованием таблицами Rijndael S-box.

Первая цифра в шестнадцатеричной записи ключа — строка, вторая — столбец. Например, “19” становится “d4”.

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

Рисунок 4. S-box

1.3.2. ShiftRows

Первая строка сдвигается на 1 байт, вторая — на 2, третья — на 3.

d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

(a) Вход ShiftRows

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

(b) Выход ShiftRows

1.3.3. MixColumns

Каждый столбец представляется как полином третьей степени. Он умножается в $GF(2^8)$ по модулю $x^4 + 1$ на многочлен $3x^3 + x^2 + x + 2$.

02	03	01	01	•	d4	=	04
01	02	03	01		bf		66
01	01	02	03		5d		81
03	01	01	02		30		e5

Рисунок 6. MixColumns

1.3.4. AddRoundKey

Сложение каждого столбца с раундовым ключом с помощью xor.

04		a0		a4
66	⊕	fa	=	9c
81		fe		7f
e5		17		f2

Рисунок 7. AddRoundKey

1.4. Генерация раундовых ключей

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Рисунок 8. Rcon

1. К столбцу матрицы ключа применяется побайтовый сдвиг на 1, SubBytes, его хог сложение с первым столбцом матрицы ключа и Rcon

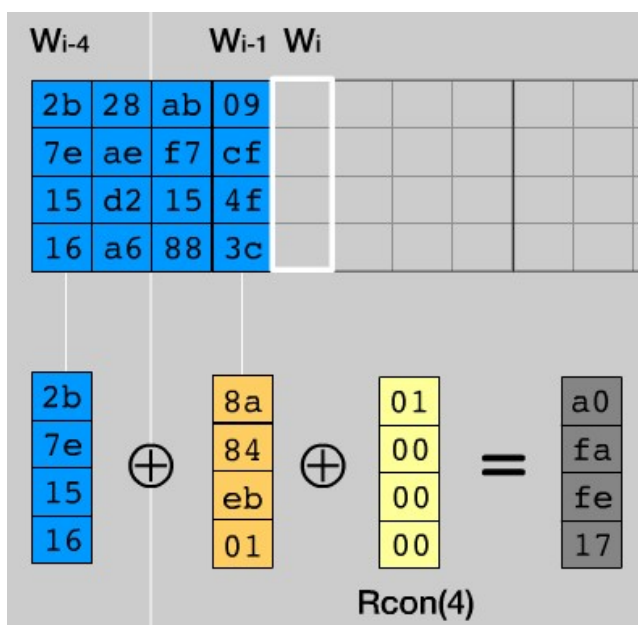


Рисунок 9. Первый столбец первого раундового ключа

2. Оставшиеся слова вычисляются хог'ом предыдущего слова со словом 4 позиции назад

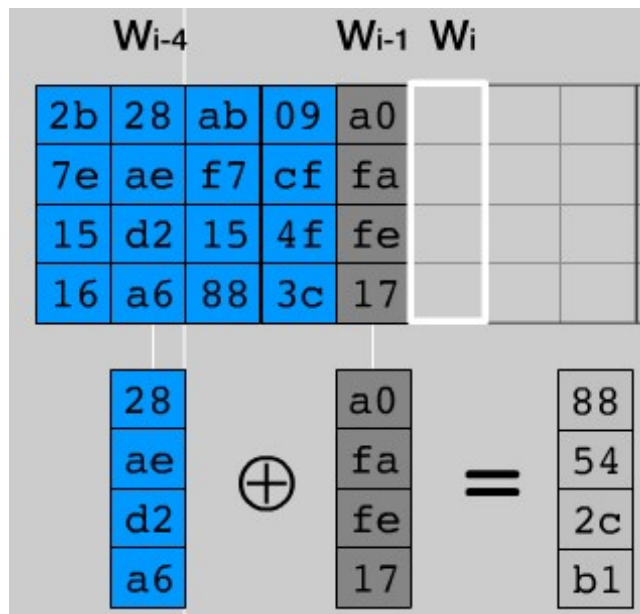


Рисунок 10. Второй столбец первого раундового ключа

3. Второй ключ вычисляется аналогичным образом по первому и т.д.

1.5. Ход работы

1. Изучены преобразования AES, заполнены разделы 1.2–1.4
2. Для одного раунда преобразования выполнены вручную.

Открытый текст — KORYTOV_PAVELLLL, ключ — 6304_VALERIEVICH

(а) Произведено преобразование ключа и текста в шестнадцатеричный формат:

- Текст — 4b4f5259544f565f504156454c4c4c4c
- Ключ — 363330345f56414c4552494556494348

Получившая начальные матрицы:

$$\text{Input} = \begin{bmatrix} 4b & 4f & 52 & 59 \\ 54 & 4f & 56 & 5f \\ 50 & 41 & 56 & 45 \\ 4c & 4c & 4c & 4c \end{bmatrix}; \text{Key} = \begin{bmatrix} 36 & 33 & 30 & 34 \\ 5f & 56 & 41 & 4c \\ 45 & 52 & 49 & 45 \\ 56 & 49 & 43 & 48 \end{bmatrix} \quad (1.1)$$

(b)

$$A_1 = AddRoundKey(Input) = Input \oplus Key = \begin{bmatrix} 7d & 7c & 62 & 6d \\ 0b & 19 & 17 & 13 \\ 15 & 13 & 1f & 00 \\ 1a & 05 & 0f & 04 \end{bmatrix} \quad (1.2)$$

(c)

$$B_1 = SubBytes(A_1) = \begin{bmatrix} ff & 10 & aa & 3c \\ 2b & d4 & f0 & 7d \\ 59 & 7d & c0 & 63 \\ a2 & 6b & 76 & f2 \end{bmatrix} \quad (1.3)$$

(d)

$$C_1 = ShiftRows(B_1) = \begin{bmatrix} ff & 10 & aa & 3c \\ d4 & f0 & 7d & 2b \\ c0 & 63 & 59 & 7d \\ f2 & a2 & 6b & 76 \end{bmatrix} \quad (1.4)$$

(e)

$$\begin{aligned} D_1 &= MixColumns(C_1) = \\ &= \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \bullet \left(\begin{bmatrix} ff \\ d4 \\ c0 \\ f2 \end{bmatrix}, \begin{bmatrix} 10 \\ f0 \\ 63 \\ a2 \end{bmatrix}, \begin{bmatrix} aa \\ 7d \\ 59 \\ 6b \end{bmatrix}, \begin{bmatrix} 3c \\ 2b \\ 7d \\ 76 \end{bmatrix} \right) = \\ &= \begin{bmatrix} b0 & ea & fa & 0e \\ e5 & ec & d0 & 9b \\ bd & db & d8 & 77 \\ f1 & fc & 17 & fe \end{bmatrix} \end{aligned} \quad (1.5)$$

(f) Раундовый ключ:

$$K_1 = \begin{bmatrix} 1e & 2d & 1d & 29 \\ 31 & 67 & 26 & 6a \\ 17 & 45 & 0c & 49 \\ 4e & 07 & 44 & 0c \end{bmatrix} \quad (1.6)$$

(g)

$$\begin{aligned} A_2 &= AddRoundKey(D_1) = \\ &= \begin{bmatrix} b0 & ea & fa & 0e \\ e5 & ec & d0 & 9b \\ bd & db & d8 & 77 \\ f1 & fc & 17 & fe \end{bmatrix} \oplus \begin{bmatrix} 1e & 2d & 1d & 29 \\ 31 & 67 & 26 & 6a \\ 17 & 45 & 0c & 49 \\ 4e & 07 & 44 & 0c \end{bmatrix} = \begin{bmatrix} ae & c7 & e7 & 27 \\ d4 & 8b & f6 & f1 \\ aa & 9e & d4 & 3e \\ bf & fb & 53 & f2 \end{bmatrix} \end{aligned} \quad (1.7)$$

3. Проверена корректность расчётов с помощью Rijndael Inspector
4. Проведены наблюдения в Rijndael Flow Visualization. Результаты на рис. 11

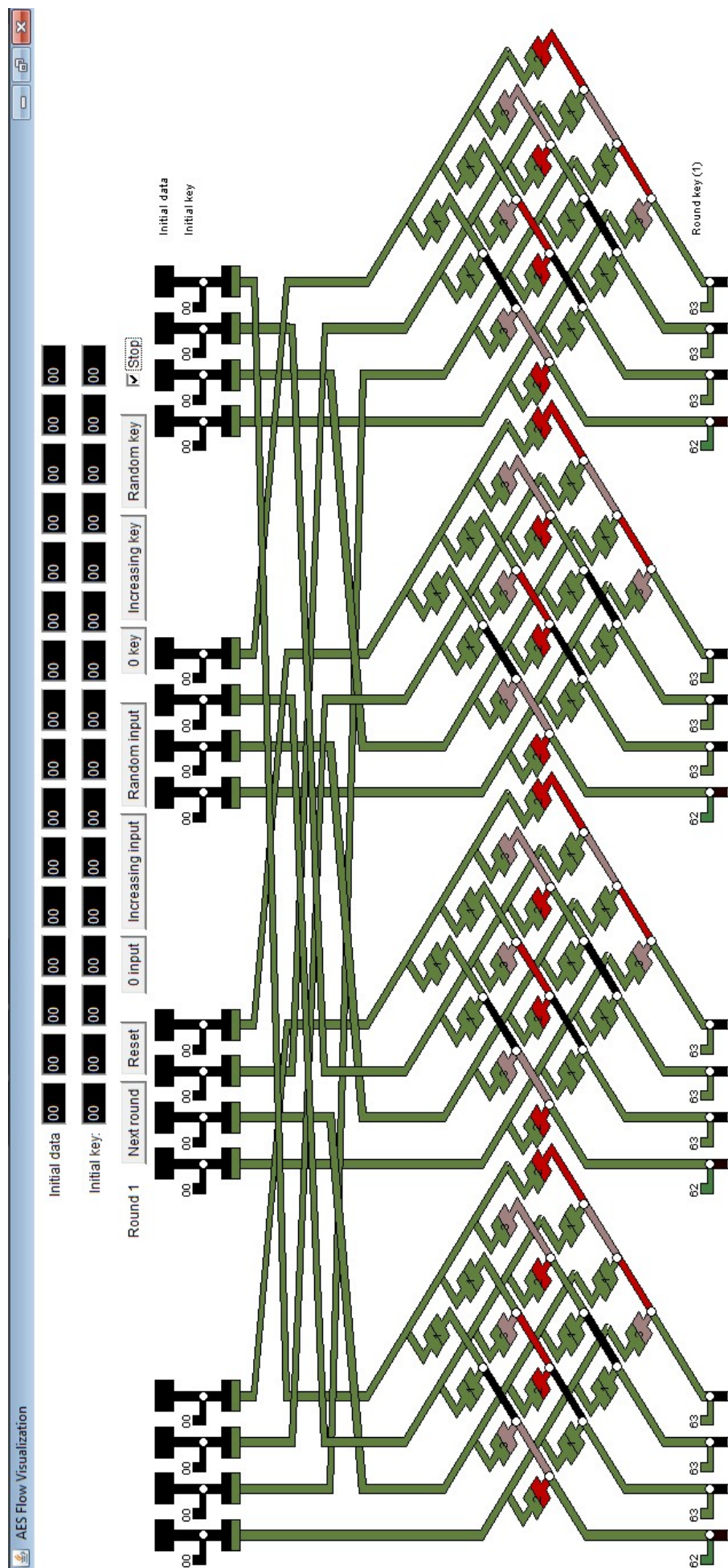


Рисунок 11. Потокная модель AES

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

2.1. Формулировка задания

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 оценить 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. Ход работы

1. Выбран текст на английском языке: Late in the planning of Caesar's assassination, there were two different opinions: one led by Brutus to kill only Caesar
2. Создан бинарный файл

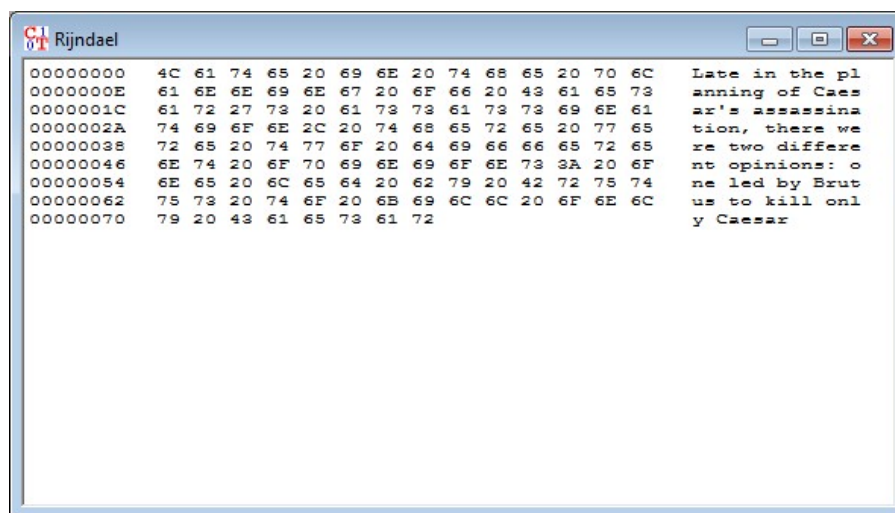


Рисунок 12. Бинарный файл

3. Произведено зашифрование файла всеми алгоритмами-финалистами конкурса AES

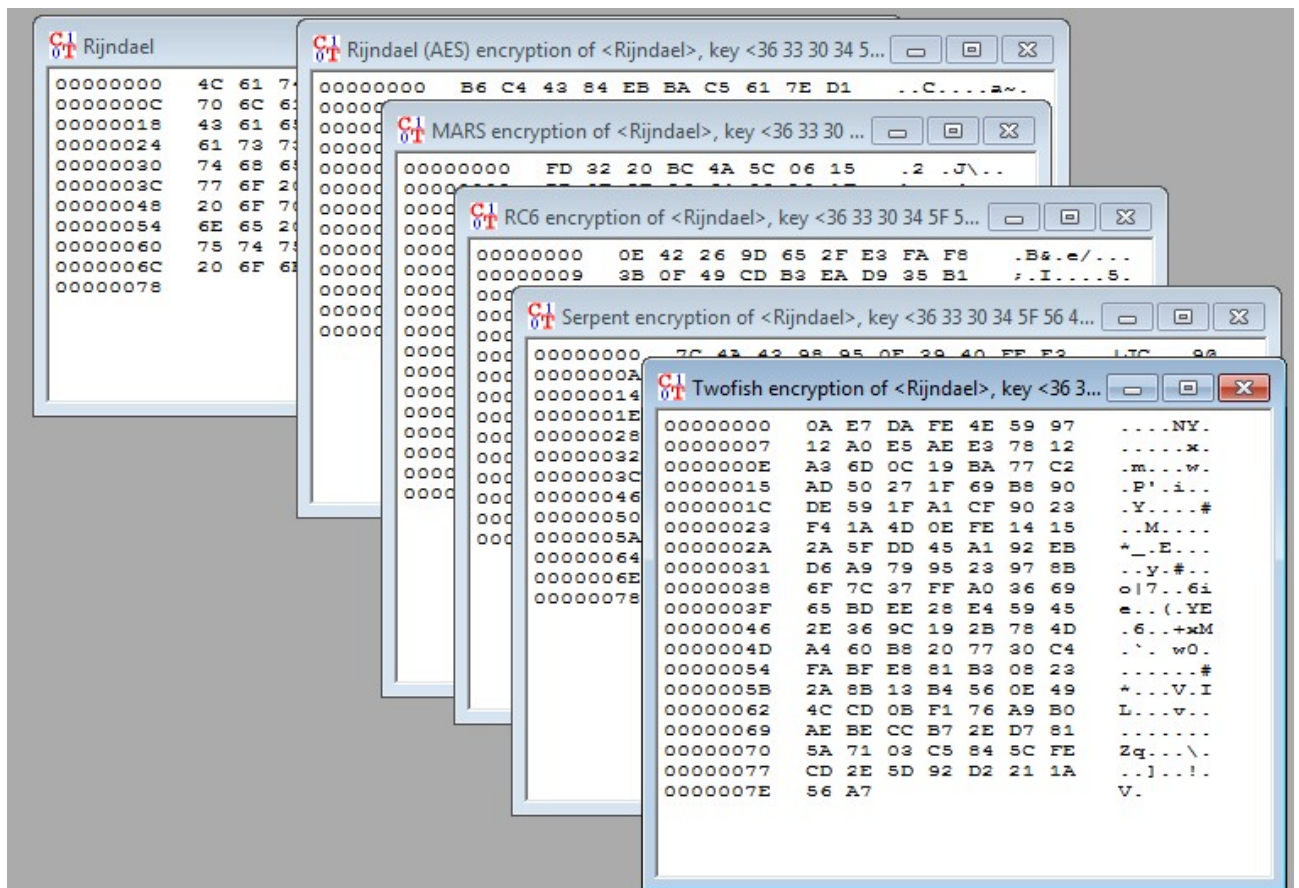


Рисунок 13. Шифрование текста

4. Зафиксирована энтропия исходного текста и полученных шифротекстов

Алгоритм	Энтропия
Исходный текст	4.10
Rijndael (AES)	6.59
MARS	6.50
RC6	6.45
Serpent	6.75
Twofish	6.49

5. Измерено время атаки грубой силой на все шифры

Алгоритм	Известно байт						
	2	4	6	8	10	12	14
Rijndael (AES)	$3 * 10^{20}$ лет	$4 * 10^{15}$ лет	$7 * 10^{10}$ лет	$1 * 10^6$ лет	16 лет	2 часа	1 сек
MARS	$4 * 10^{20}$ лет	$6 * 10^{15}$ лет	$1.11 * 10^{11}$ лет	$1.6 * 10^6$ лет	24 года	3 часа	1 сек
RC6	$3 * 10^{20}$ лет	$4.6 * 10^{15}$ лет	$7 * 10^{10}$ лет	$1.1 * 10^6$ лет	16 лет	2 часа	1 сек
Serpent	$8 * 10^{20}$ лет	$1.2 * 10^{16}$ лет	$1.9 * 10^{11}$ лет	$2.9 * 10^6$ лет	45 лет	5 часов	1 сек
Twofish	$4.8 * 10^{20}$ лет	$7.3 * 10^{15}$ лет	$1.11 * 10^{11}$ лет	$1.6 * 10^6$ лет	26 лет	3 часа	1 сек

3. Атака «грубой силы» на AES

3.1. Формулировка задания

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

3.2. Ход работы

1. Найден шаблон атаки в CrypTool 2
2. Выбран открытый текст: The US doesn't care about Taiwan or its people. The objective is to hold Taiwan, along with South Korea and Japan and form a containment barrier that fences China in from projecting its military out into the Pacific. If China is able to break through either Taiwan or Japan, then its navy can have a shot at challenging US dominance of the blue ocean. Hence the question is how much the US is willing to sacrifice along with Japanese and maybe South Korean forces in order to contain China within the East China sea.

I think all China needs to do in this situation, is build up its forces and wait. Nothing guarantees that South Korea and Japan

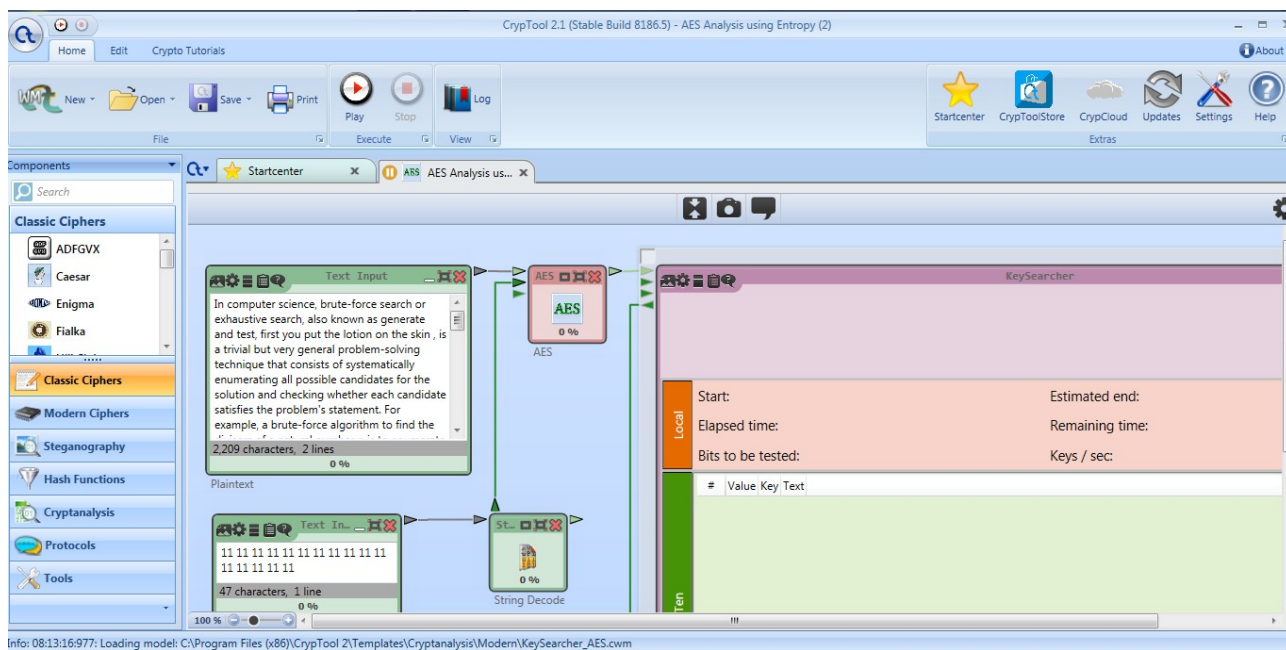


Рисунок 14. Шаблон атаки в CrypTool 2

will be on the same side in 20 years. And since China is at least ten times the size of Japan, I'm not sure Japan would be that enthusiastic about confronting them in 20 years time. That leaves the US. So China's big problem would be how to discourage further US action in its ``turf'' --- i.e. making the costs of intervention unacceptably high.

3. Проведена атака на шифр с использованием разного количества ядер

	1 ядро	2 ядра	4 ядра
14 байт	1 сек	1 сек	1 сек
12 байт	1.5 часа	47 мин	22 мин
10 байт	10 лет	5.5 лет	3 года

4. Сформировано сообщение в указанном формате. В качестве оценочной функции использовано выражение «DEAR SIRS». Оценка времени атаки:

	1 ядро	2 ядра	4 ядра
14 байт	1 сек	1 сек	1 сек
12 байт	21 мин	12 мин	7 мин
10 байт	3 года	1.4 года	310 дней

4. Атака предсказанием дополнения на шифр AES

4.1. Формулировка задания

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

4.2. Описание атаки

Используются правила дополнения неполных блоков — один байт дополняется как 01, два как 0202 и т.п.

Атака возможна в режиме CBC и работает через вычисление промежуточного состояния

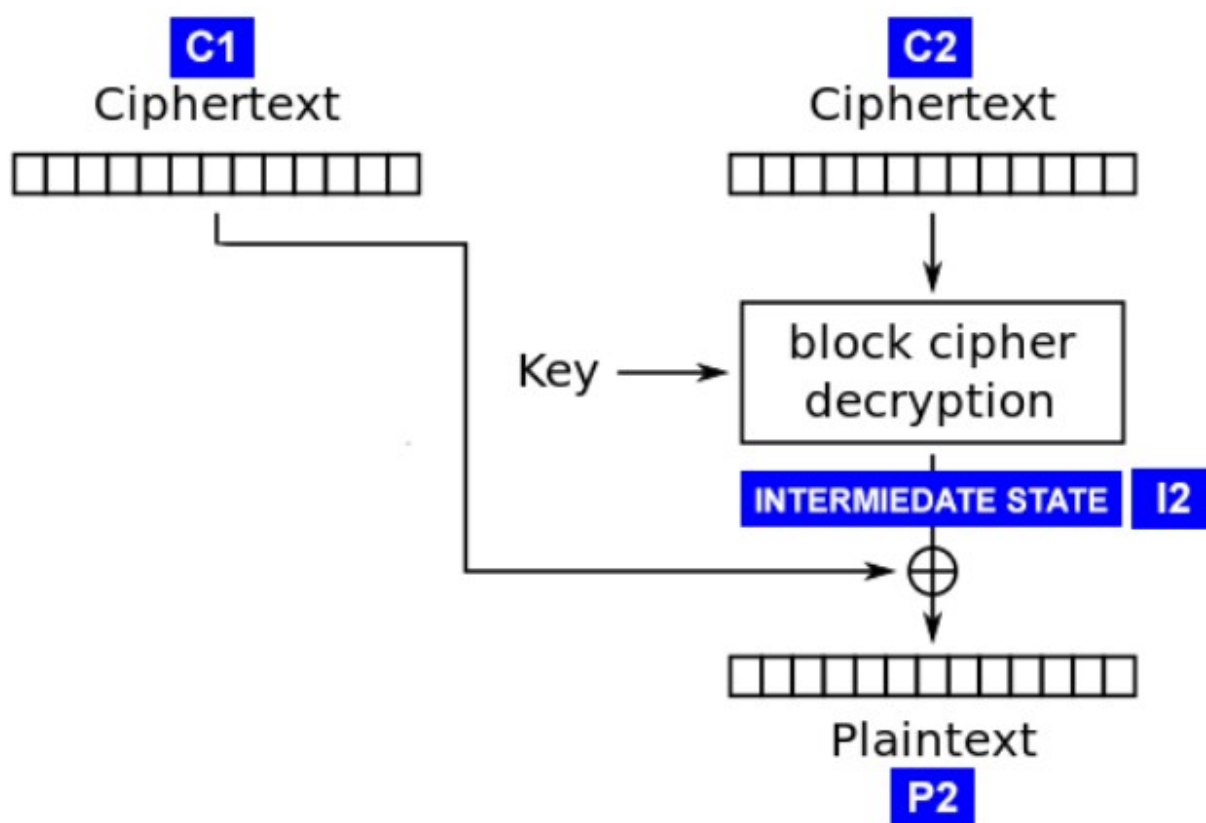


Рисунок 15. Промежуточное состояние CBC

Условие проведения атаки — когда сообщения отправляются серверу на расшифровку, тот возвращает ответ — корректно ли выполнено дополнение последнего блока.

Пусть размер блока — 16 байт, и известны блоки шифротекста C_1 и C_2 . Создается блок C'_1 , где $C'_1[1 \dots 15]$ — случайные байты, а $C'_1[16]$ перебирается от $0x00$ до $0xFF$

На каком-то этапе сервер ответит, что дополнение корректно и $P'_2 = 0x01$. Тогда

$$I_2 = C'_1 \oplus P'_2;$$

$$I_2[16] = C'_1[16] \oplus P'_2[16],$$

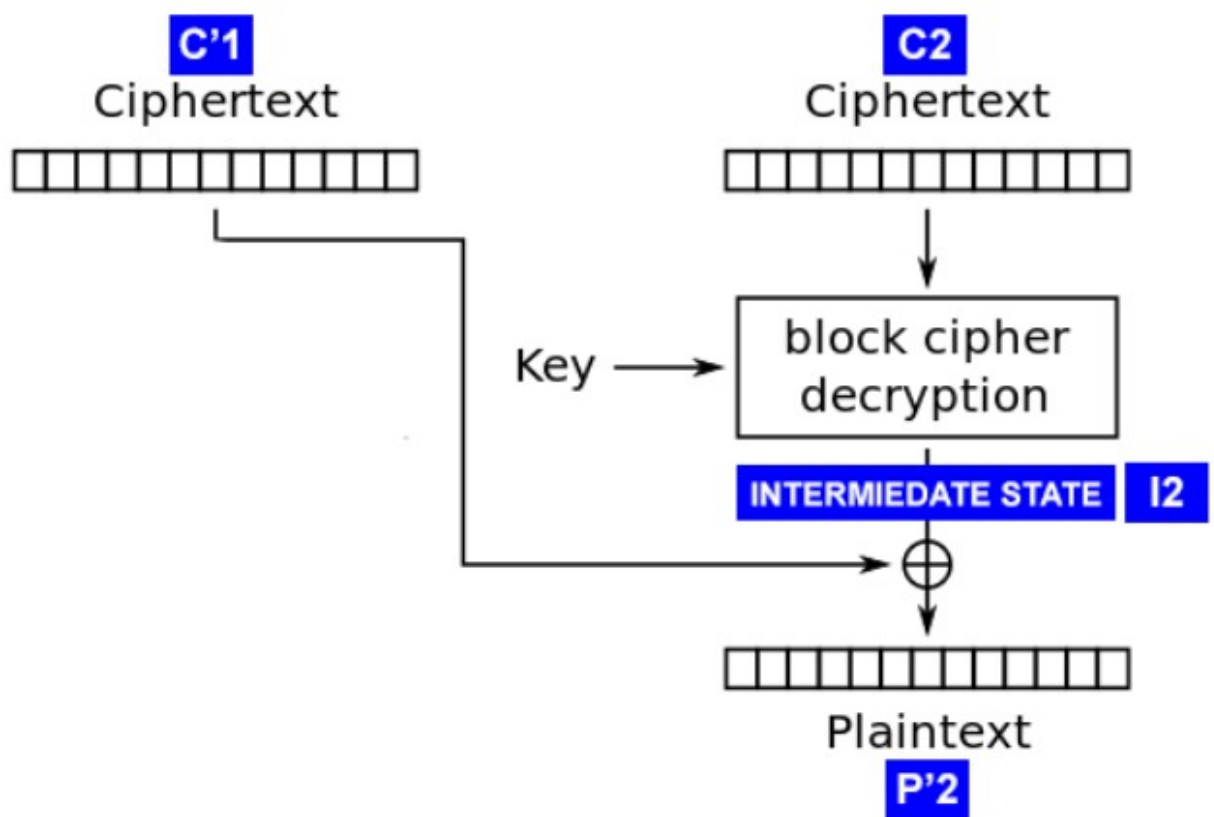


Рисунок 16. Вычисление промежуточного состояния

Теперь можно расшифровать настоящий $P_2[16]$:

$$P_2[16] = C_1[16] \oplus I_2[16].$$

Далее пытаемся получить дополнение 0202. $C'_1[1 \dots 14]$ заполняется случайно, $C'_1[15]$ итерируется, а $C'_1[16]$ устанавливается таким образом, чтобы $P_2[16] = 0x02$:

$$C'_1[16] = P'_2[16] \oplus I_2[16].$$

Когда сервер возвращает информацию о корректном дополнении, становится известен предпоследний байт промежуточного состояния:

$$I_2[15] = C'_1[15] \oplus P'_2[15]$$

И можно вычислить предпоследний байт открытого текста:

$$P_2[15] = C_1[15] \oplus I_2[15]$$

Аналогичным образом вычисляются все байты C_2 .

Описанный алгоритм можно применить к каждому блоку шифротекста, кроме первого. Первый блок зашифрован с помощью вектора инициализации, его всё же придется подобрать.

4.3. Ход работы

1. Найден шаблон в CrypTool 2

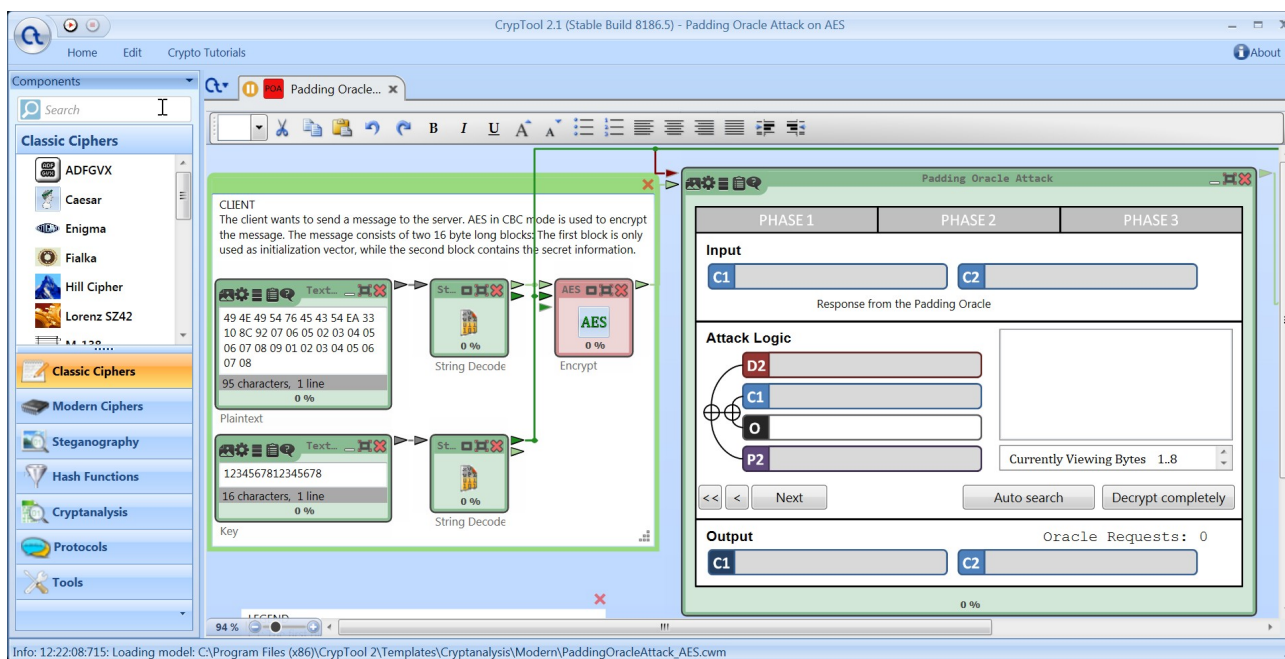


Рисунок 17. Шаблон атаки в CrypTool 2

2. В исходный блок текста внедрен текст KORYTOV_PAVEL:

49 4E 49 54 76 45 43 54 EA 33 10 8C 92 07 06 05 4b 4f 52 59 54 4f
56 5f 50 41 56 45 4c 03 03 03

3. Выполнена первая фаза атаки:

PHASE 1	PHASE 2	PHASE 3
Input <div> <div>C1</div> <div>F7 11 2E 66 B8 33 A4 3D</div> <div>C2</div> <div>86 94 D8 3C 46 95 DA 26</div> </div> <div> <div>VALID</div> <div>Response from the Padding Oracle</div> </div>		
Attack Logic <div> <div> <div>D2</div> <div>?? ?? ?? ?? ?? ?? ?? 00</div> </div> <div> <div>C1</div> <div>F7 11 2E 66 B8 33 A4 3D</div> </div> <div> <div>O</div> <div>00 00 00 00 00 00 00 00</div> </div> <div> <div>P2</div> <div>?? ?? ?? ?? ?? ?? ?? ??</div> </div> </div> <div> <div>Phase 1 finished! Valid padding found.</div> <div>Currently Viewing Bytes 9...16</div> </div> <div> <div><<</div> <div><</div> <div>Next</div> <div>Go to Phase 2</div> <div>Auto search</div> <div>Decrypt completely</div> </div>		
Output <div> <div>C1</div> <div>F7 11 2E 66 B8 33 A4 3D</div> <div>C2</div> <div>86 94 D8 3C 46 95 DA 26</div> </div> <div>Oracle Requests: 1</div>		

Рисунок 18. Первая фаза атаки

В первой фазе меняется последний байт C_1 , чтобы определить правильное дополнение

4. Выполнена вторая фаза атаки:

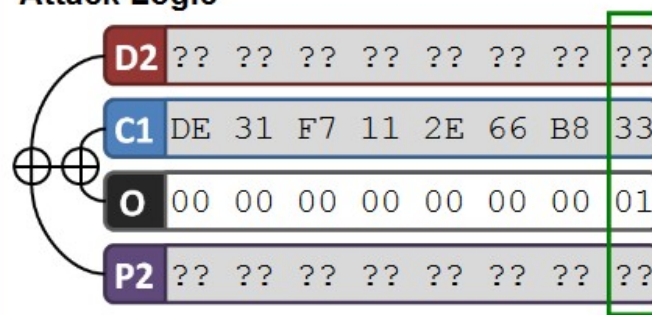
PHASE 1	PHASE 2	PHASE 3
Input <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> C1 DE 31 F7 11 2E 66 B8 33 </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> C2 50 07 86 94 D8 3C 46 95 </div> </div> <div style="margin-top: 5px; display: flex; align-items: center;"> <div style="background-color: #800000; color: white; padding: 2px 10px; font-weight: bold; font-size: 0.8em;">INVALID</div> <div style="margin-left: 10px; font-size: 0.8em;">Response from the Padding Oracle</div> </div>		
Attack Logic <div style="display: flex; align-items: flex-start; margin-top: 10px;"> <div style="flex: 1;">  <div style="margin-top: 10px;"> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> D2 ?? ?? ?? ?? ?? ?? ?? ?? </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> C1 DE 31 F7 11 2E 66 B8 33 </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> O 00 00 00 00 00 00 00 01 </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> P2 ?? ?? ?? ?? ?? ?? ?? ?? </div> </div> </div> <div style="flex: 1; padding-left: 10px;"> <div style="border: 1px solid #ff0000; padding: 10px; margin-bottom: 10px;"> Phase 2 finished! First padding byte found! Padding length:3 </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> Currently Viewing Bytes <div style="border: 1px solid #ccc; padding: 0 5px; font-size: 0.8em;">7...14</div> </div> </div> </div> <div style="margin-top: 10px; display: flex; justify-content: space-between; align-items: center;"> <div> << < Next Go to Phase 3 </div> <div> Auto search Decrypt completely </div> </div>		
Output <div style="float: right; font-size: 0.8em;">Oracle Requests: 15</div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> C1 DE 31 F7 11 2E 66 B8 32 </div> <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"> C2 50 07 86 94 D8 3C 46 95 </div> </div>		

Рисунок 19. Вторая фаза атаки

Меняется один байт правильно дополненного C_1 , начиная с начала. С того момента, как сервер начинает возвращать ошибку, начинается дополнение, т.к. с этого момента начинают изменяться байты дополнения.

На второй фазе определяется длина дополнения.

5. Выполнена третья фаза атаки:

PHASE 1	PHASE 2	PHASE 3
Input <div> <div>C1</div> <div>A4 07 6D D7 4E DE 31 F7</div> </div> <div> <div>C2</div> <div>13 F6 1E 80 F6 50 07 86</div> </div> <p>Response from the Padding Oracle</p>		
<div> Attack Logic <div> <div>D2</div> <div>EB 55 34 83 01 88 6E A7</div> </div> <div> <div>C1</div> <div>A4 07 6D D7 4E DE 31 F7</div> </div> <div> <div>O</div> <div>5F 42 49 44 5F 46 4F 40</div> </div> <div> <div>P2</div> <div>10 10 10 10 10 10 10 10</div> </div> </div> <div> <p>Message was decrypted! Click to see the original plaintext.</p> <p>Currently Viewing Bytes 2...9</p> </div> <div> <div><<</div> <div><</div> <div>Next</div> <div>Recover Plaintext</div> <div>Auto search</div> <div>Decrypt completely</div> </div>		
Output <div> <div>C1</div> <div>FB 45 24 93 11 98 7E B7</div> </div> <div> <div>C2</div> <div>13 F6 1E 80 F6 50 07 86</div> </div> <p>Oracle Requests: 1067</p>		

Рисунок 20. Третья фаза атаки

- Атака проведена успешно; блок шифротекста расшифрован корректно.

Выводы

AES — блочный симметричный шифр, размер блока — 128 бит. Шифр основан на подстановочно-перестановочной сети, а не на сети Фейстеля. Количество раундов:

- 10 для 128-битного ключа
- 12 для 192-битного ключа
- 14 для 256-битного ключа

Энтропия текста и сложность атаки грубой силы сопоставима с таковой для других алгоритмов-финалистов конкурса AES, немного лучше показал себя Serpent.

Как шифрование, так и дешифрование AES распаралелливаемо. Благодаря этому атака грубой силы с использованием большего количества ядер более эффективна.

Облегчить взлом может использование в качестве оценочной функции части выражения исходного текста (вместо энтропии) и знание части ключа.

Атака предсказанием дополнения в меньшей степени отражает недостатки шифра; она возможна только при его неправильном использовании.