# МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

# «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра МО ЭВМ**

ЛАБОРАТОРНАЯ РАБОТА № 3

ПО ДИСЦИПЛИНЕ «КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ»

**ТЕМА: ИЗУЧЕНИЕ ШИФРА AES**

|  |  |
| --- | --- |
| Студент гр. 8304 | Сергеев А.Д. |
| Преподаватель | Племянников А.К. |

Санкт-Петербург 2021 г.

**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.

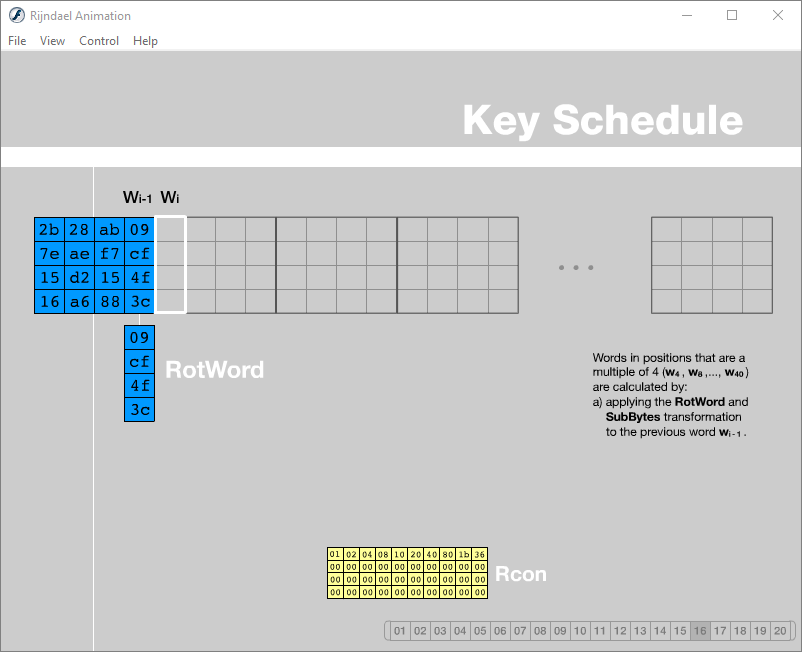
**Исследование преобразований AES.**

Задание:

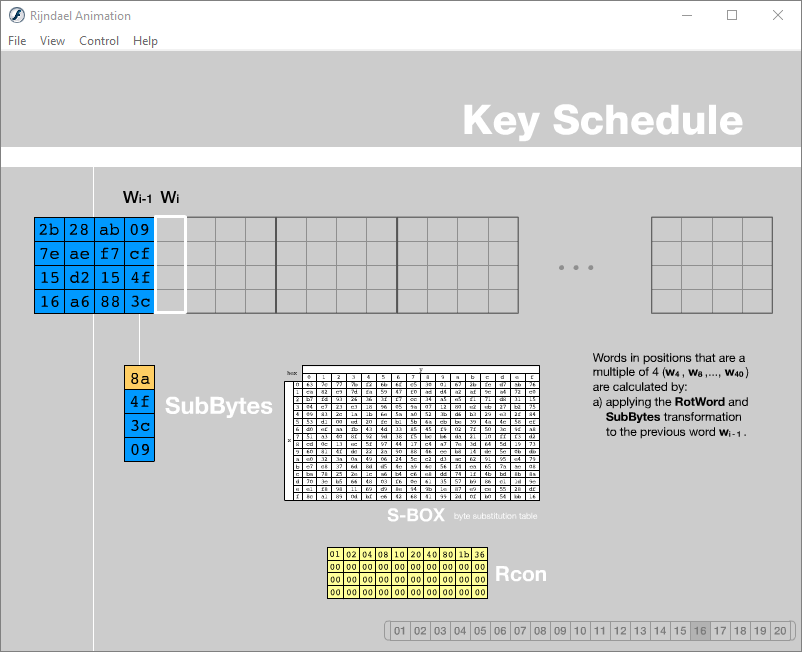
1. Изучить преобразования шифра AES с помощью демонстрационного приложения из Cryptool 1.
   1. *Indiv.Procedures* -> *Visualization…* -> *DES -> Rijndael Animation*
2. Выполнить вручную преобразования для одного раунда и вычисление раундового ключа при следующих исходных данных:
   1. Открытый текст – фамилия\_имя (транслитерация латиницей).
   2. Ключ (56 бит) – 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*.

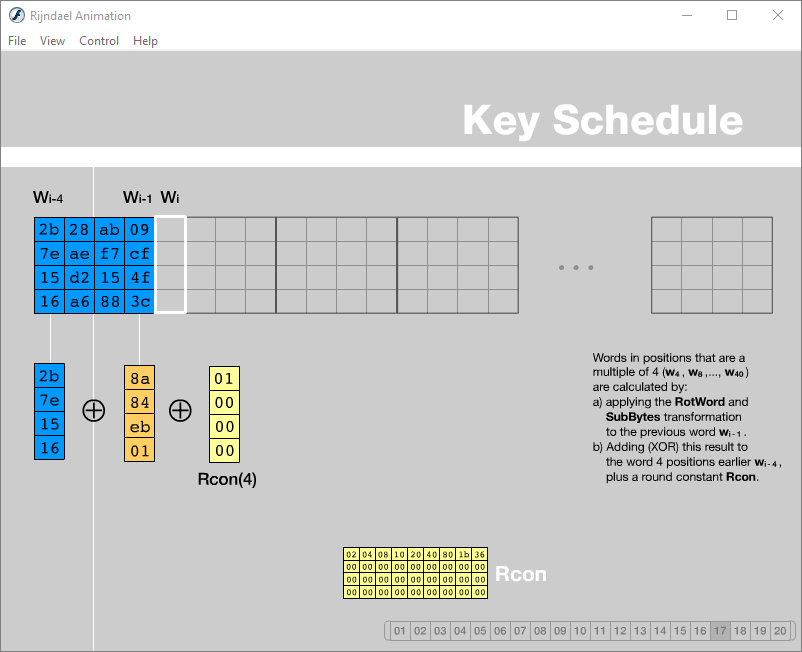
Описание AES c примерами скриншотов из демоприложения:

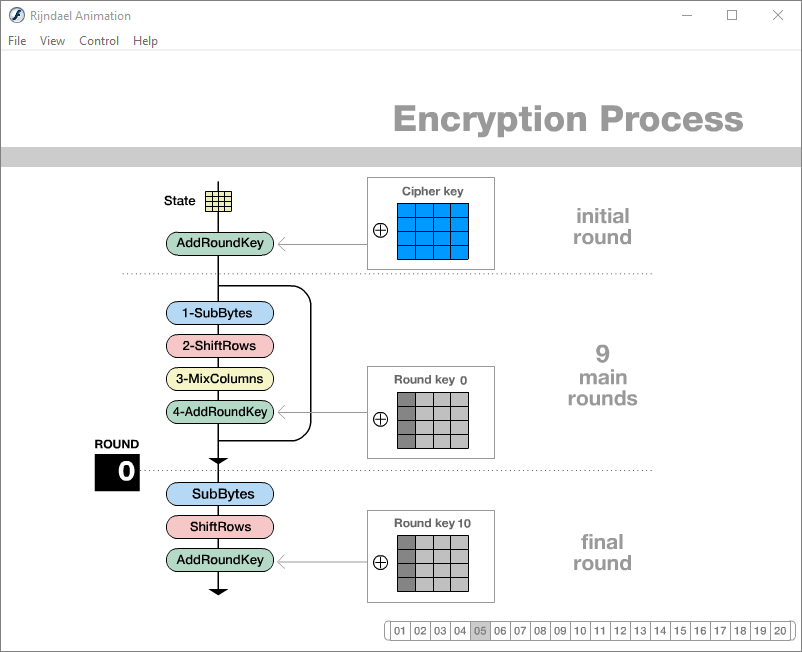
1. Разделение открытого текста на фрагменты длиной в 128 бита, выбор фрагмента. Дополнение последнего фрагмента до целого блока при помощи алгоритма дополнения. В случае, если последний блок занимает 128 бит – добавление дополнительного блока, целиком сформированного при помощи алгоритма дополнения.
2. Генерация раундовых ключей. Алгоритму передается ключ предыдущего раунда. Последние 4 байта предыдущего ключа смещаются на 1 позицию вперед:

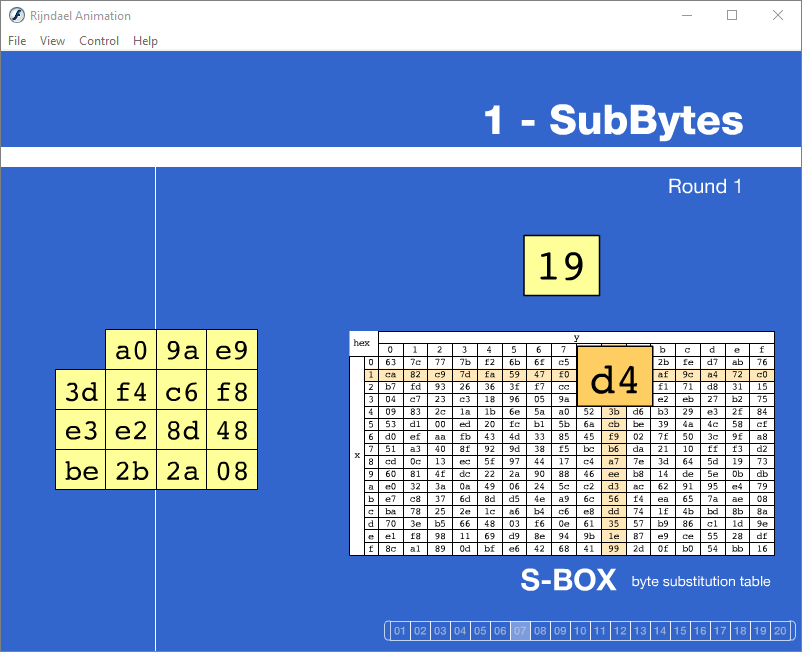


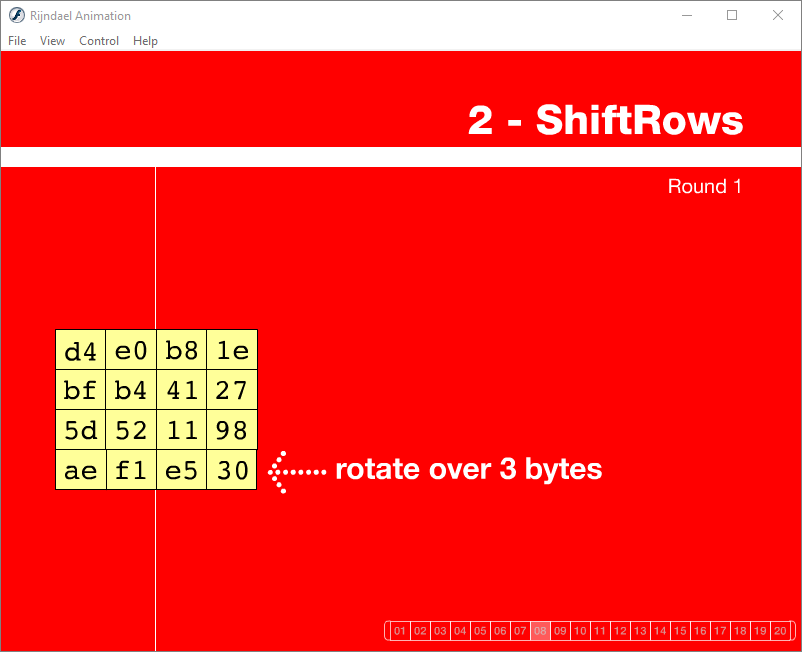
1. Полученные 4 байта замещаются в соответствии с матрицей перестановки:

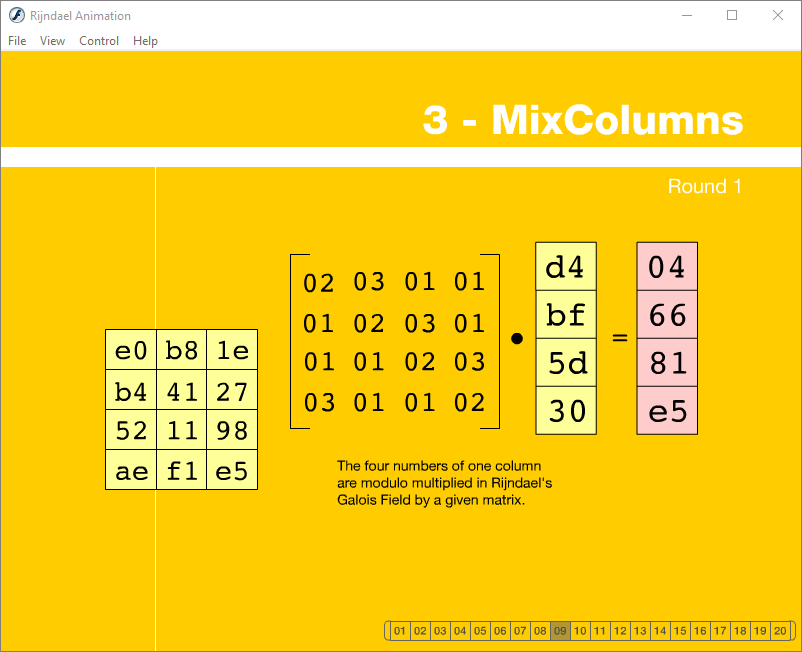


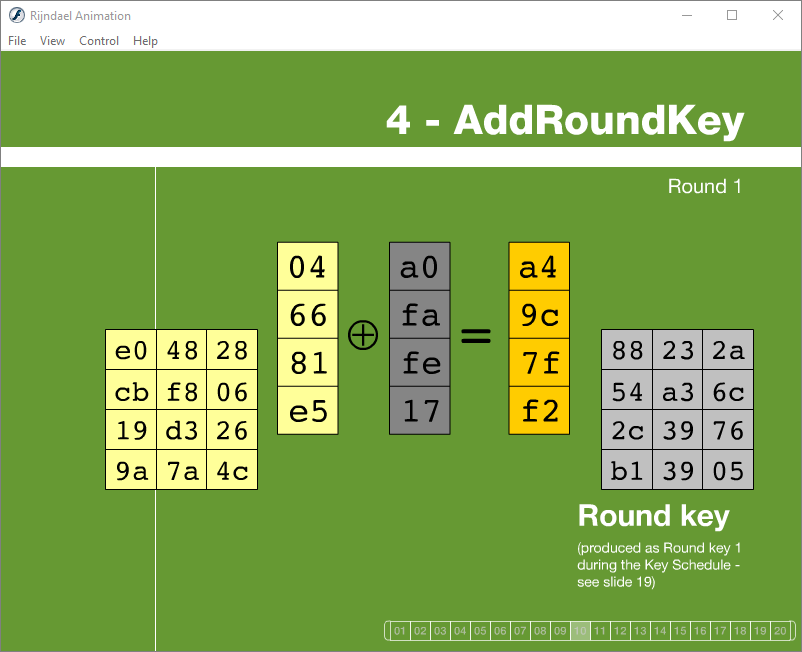
1. Над полученными 4 байтами и раундовой константой, представляющей из себя номер раунда \* 2^24 производится операция XOR:
2. Байты раундового ключа группами по 4 получаются при помощи применения операции XOR к предыдущим 4 байтам раундового ключа и предыдущим 4 байтам предыдущего ключа. В качестве изначальной группы выступают полученные в результате предыдущего действия 4 байта.
3. Вычисление матрицы: проведение операции XOR над ключом и открытам текстом.



1. Проведение операции SubBytes над матрицей, состоящей в замещении байтов матрицы в соответствии с матрицей перестановки:
2. Проведение операции ShiftRows над матрицей, состоящей в смещении байтов матрицы группами по 4 на количество позиций, равное номеру группы:



1. Проведение операции MixColumns над матрицей, состоящей в умножении матрицы на матрицу, состоящую из 4 раза повторенных чисел 2, 3, 1, 1, во время каждого повторения смещенных на количество позиций, равное номеру повторения:
2. Проведение операции AddRoundKey над матрицей, состоящей в проведении операции XOR над матрицей и соответствующему текущему раунду раундовом ключе:



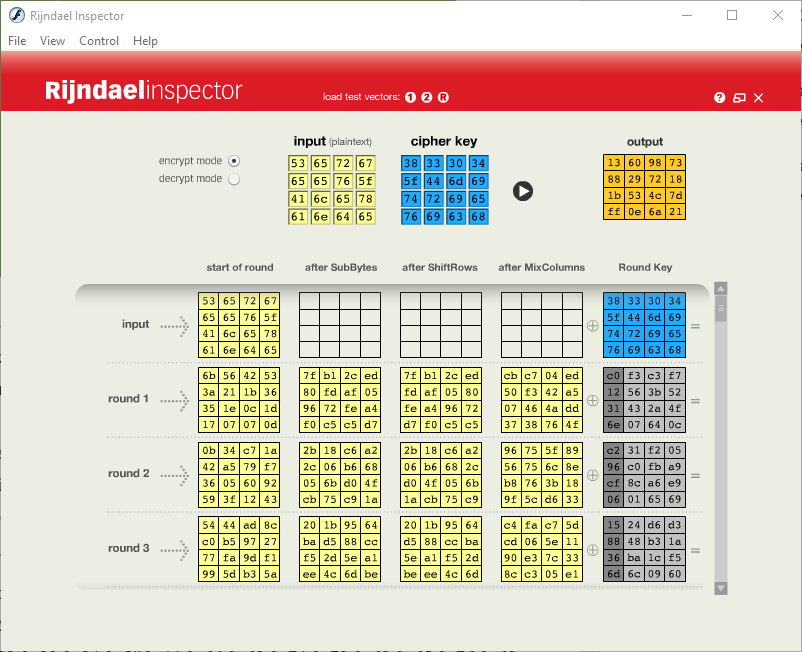
1. Повторение вышеописанных действий в течении 10 раундов, в последнем опуская проведение операции MixColumns.

Расчет матрицы состояний и раундового ключа шифра для одного раунда:

Исходные данные: открытый текст: «Sergeev\_Alexande», ключ: «8304\_Dmitrievich».

Для произведения более точных расчетов вычисление было автоматизировано при помощи специально написанного скрипта.

1. Исходный текст в 16 системе счисления: [0x53 0x65 0x72 0x67 0x65 0x65 0x76 0x5f 0x41 0x6c 0x65 0x78 0x61 0x6e 0x64 0x65], ключ: [0x38 0x33 0x30 0x34 0x5f 0x44 0x6d 0x69 0x74 0x72 0x69 0x65 0x76 0x69 0x63 0x68]
2. Вычисление раундового ключа (ShiftRows): [0x76 0x69 0x63 0x68] -> [0x69 0x63 0x68 0x76]
3. Вычисление раундового ключа (SubBytes): [0x69 0x63 0x68 0x76] -> [0xf9 0xfb 0x45 0x38]
4. Вычисление раундового ключа (AddRoundConst): [0xf9 0xfb 0x45 0x38] -> [0xf8 0xfb 0x45 0x38]
5. Раундовый ключ: [0xc0 0xc8 0x75 0xc 0x9f 0x8c 0x18 0x65 0xeb 0xfe 0x71 0x0 0xd3 0xcd 0x41 0x34]
6. Матрица (после изначального сложения с ключом): [0x6b 0x56 0x42 0x53 0x3a 0x21 0x1b 0x36 0x35 0x1e 0xc 0x1d 0x17 0x7 0x7 0xd]
7. Раундовое преобразование (SubBytes): [0x7f 0xb1 0x2c 0xed 0x80 0xfd 0xaf 0x5 0x96 0x72 0xfe 0xa4 0xf0 0xc5 0xc5 0xd7]
8. Раундовое преобразование (ShiftRows): [0x7f 0xfd 0xfe 0xd7 0x80 0x72 0xc5 0xed 0x96 0xc5 0x2c 0x5 0xf0 0xb1 0xaf 0xa4]
9. Раундовое преобразование (MixColumns): [0xcb 0x50 0x7 0x37 0xa5 0xdd 0x4f 0xed 0x4a 0x76 0x4 0x42 0x38 0xc7 0xf3 0x46]
10. Раундовое преобразование (AddRoundKey): [0xb 0x98 0x72 0x3b 0x3a 0x51 0x57 0x88 0xa1 0x88 0x75 0x42 0xeb 0xa 0xb2 0x72]
11. Результат: [0xb 0x98 0x72 0x3b 0x3a 0x51 0x57 0x88 0xa1 0x88 0x75 0x42 0xeb 0xa 0xb2 0x72]

Скриншоты приложения-инспектора, подтверждающие корректность расчетов:

**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.

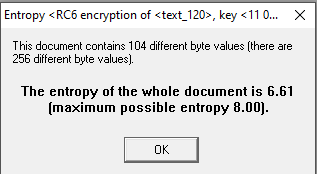
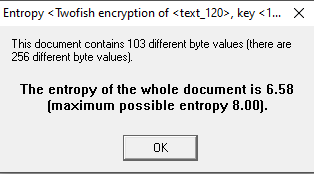
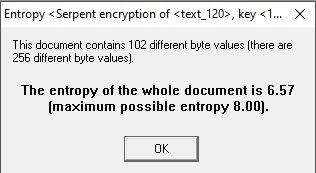
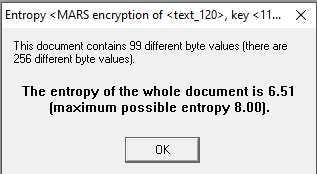
Исходные данные для экспериментов: открытый текст «Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.», ключ: «0f1a6b8d3673e38а34e8912ab459c3675».

Таблица с результатами качества зашифрования исследованными шифрами:

|  |  |
| --- | --- |
| **Шифр** | **Энтропия (max. 8.00)** |
| Открытый текст | 4.10 |
| AES | 6.53 |
| MARS | 6.55 |
| RC6 | 6.51 |
| Serpent | 6.52 |
| Twofish | 6.39 |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Количество известных байт ключа** | AES | MARS | RC6 | Serpent | Twofish |
| 2 | 9e+022 y | 1.2+023 y | 7.6e+022 y | 2.7e+023 y | 1.7e+023 y |
| 4 | 3.5e+020 y | 4.8e+020 y | 3e+020 y | 1e+021 y | 6.6e+020 y |
| 6 | 1.4e+018 y | 1.9e+018 y | 1.2e+018 y | 4e+018 y | 2.6e+018 y |
| 8 | 5.5e+015 y | 7.4e+015 y | 4.4e+015 y | 1.6e+016 y | 1e+016 y |
| 10 | 2.1e+013 y | 2.9e+013 y | 1.8e+013 y | 6.2e+013 y | 4e+013 y |
| 12 | 8.4e+010 y | 1.1e+011 y | 6.8e+010 y | 2.4e+011 y | 1.6e+011 y |
| 14 | 3.3e+008 y | 4.4e+008 y | 2.7e+008 y | 9.4e+008 y | 6e+008 y |
| 16 | 1.3e+006 y | 1e+006 y | 1.1e+006 y | 3.7e+006 y | 2.4e+006 y |
| 18 | 5e+003 y | 6.7e+003 y | 4e+003 y | 1.5e+004 y | 9.3e+003 y |
| 20 | 19 y | 26 y | 16 y | 56 y | 36 y |
| 22 | 27.2 d | 37.4 d | 22.9 d | 80.1 d | 51.4 d |
| 24 | 3 h | 3:25 h | 2:08 h | 7 h | 4 h |
| 26 | 30 s | 48 s | 30 s | 1:45 m | 1:05 m |
| 28 | 0 s | 0s | 0s | 1s | 0s |
| 30 | 0 s | 0s | 0s | 0s | 0s |
| 32 | 0 s | 0s | 0s | 0s | 0s |



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

Исходные данные для экспериментов: открытый текст: «Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer et turpis in orci accumsan elementum. Suspendisse potenti. Maecenas hendrerit libero vel orci accumsan ornare. Donec convallis molestie ultrices. Vivamus non lacus a est elementum pulvinar non sed eros. Quisque ut orci vel lorem venenatis semper. Morbi convallis sodales dolor, id vulputate lorem tempus viverra. Maecenas interdum ex a lorem pharetra, vitae maximus turpis imperdiet. Morbi luctus blandit justo a tempus. Morbi sed massa id lectus lobortis ullamcorper vitae a nisl. Integer finibus magna et nunc semper venenatis. Fusce sed lacinia lectus, luctus faucibus orci. Nulla nulla odio, bibendum vel nibh et, porta dictum tellus. Aenean scelerisque vulputate dui vitae sollicitudin. Donec tempus quam non ex pellentesque, sollicitudin euismod turpis suscipit. Fusce at odio molestie, feugiat ipsum sit amet, auctor augue. Cras ullamcorper diam orci, quis interdum tellus venenatis non. Cras ut lectus diam. Maecenas sapien vel.», ключ: «1111111111111111111111111123ab34».

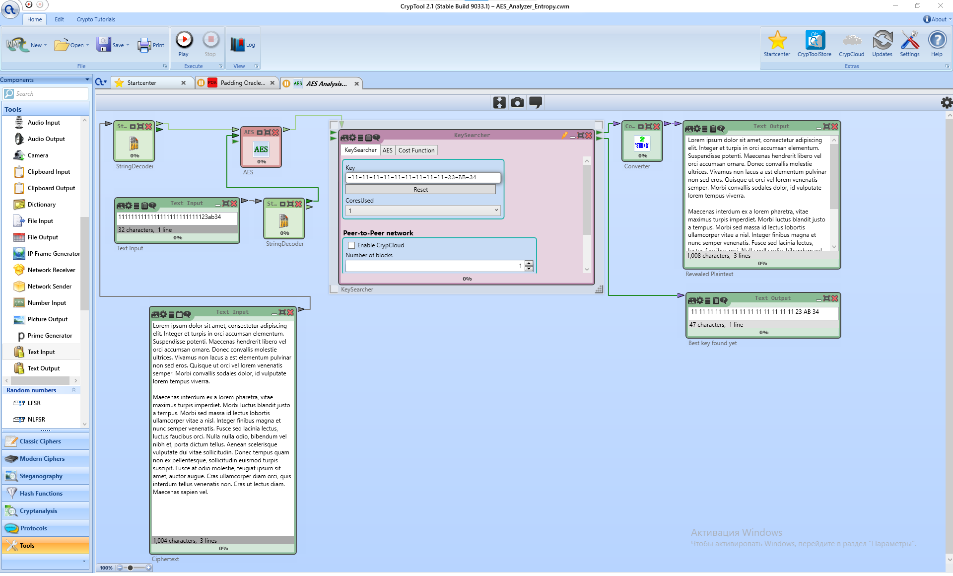
Шаблон атаки «грубой силой» из Cryptool 2:

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Количество известных байт ключа** | 1 ядро процессора | 2 ядра процессора | 3 ядра процессора | 4 ядра процессора |
| 30 | 1.91 s | 1.98 s | 1.91 s | 1.92 s |
| 28 | 2.31 s | 2.37 s | 2.43 s | 2.56 s |
| 26 | 38.50 s | 21.16s | 15.72 s | 13.68 s |

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

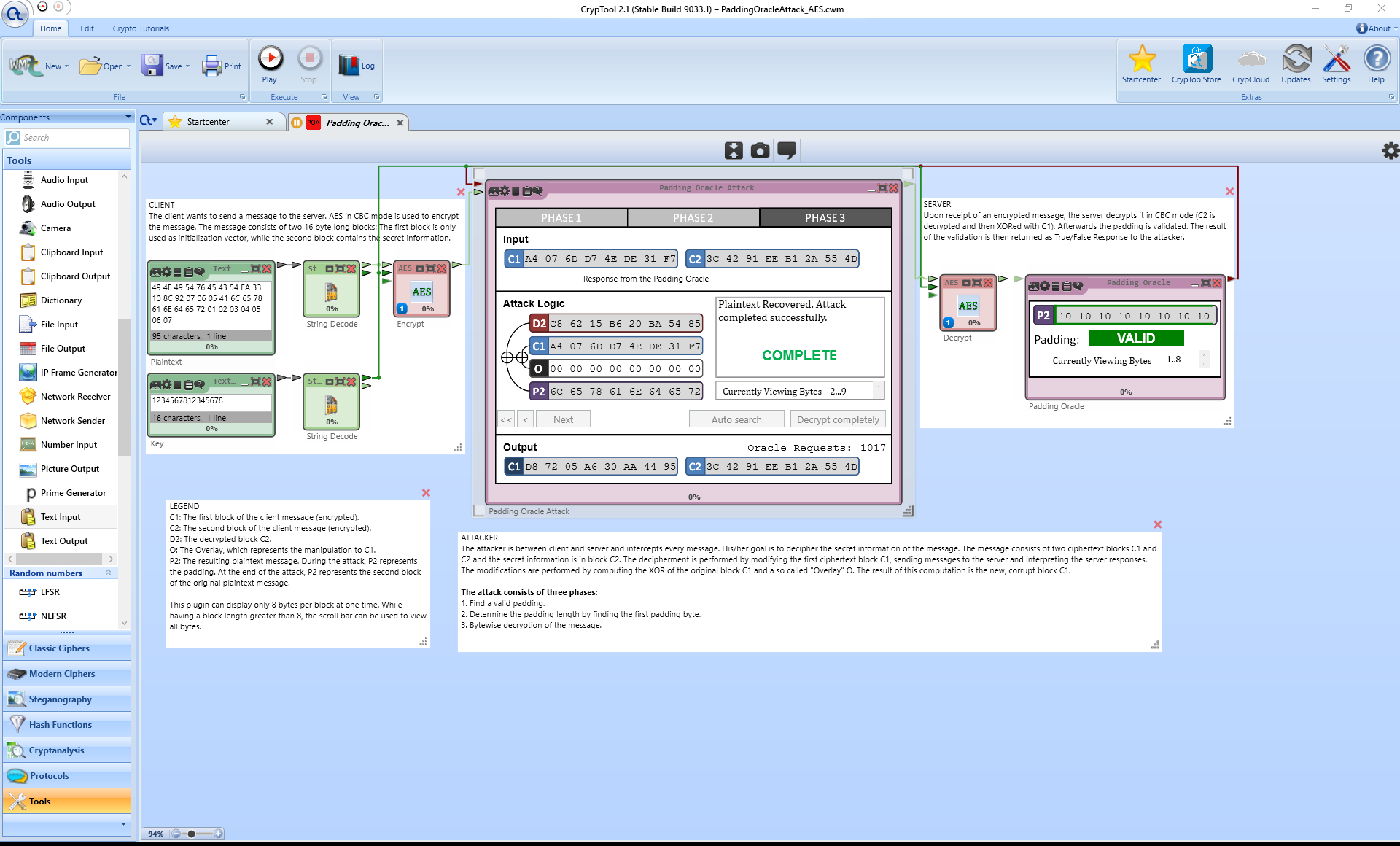
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Количество известных байт ключа** | 1 ядро процессора | 2 ядра процессора | 3 ядра процессора | 4 ядра процессора |
| 30 | 1.59 s | 1.45 s | 1.64 s | 1.65 s |
| 28 | 1.75 s | 1.66 s | 1.71 s | 1.85 s |
| 26 | 20.69 s | 12.67 s | 9.89 s | 8.86 s |

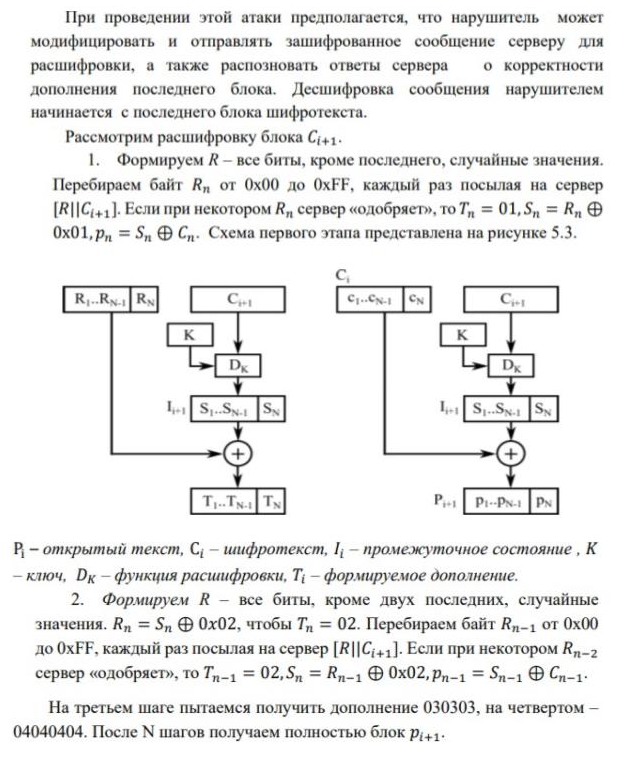
**Атака предсказанием дополнения на шифр AES в режиме CBC (Padding Oracle Attack).**

Задание:

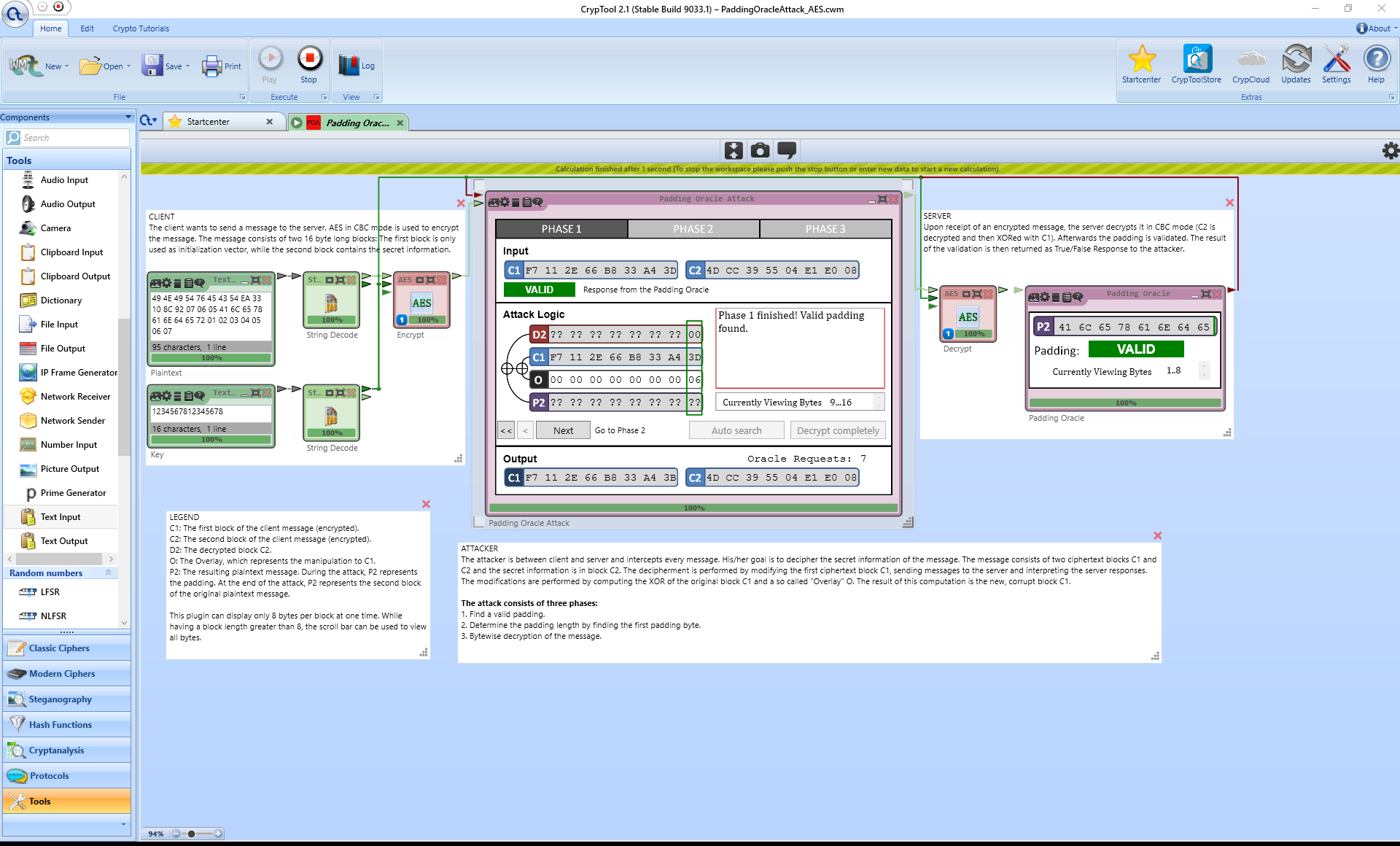
1. Найти и запустить шаблон атаки в CrypTool 2: *Padding Oracle Attack on AES*.
2. Подготовьтесь к атаке теоретически:
   1. Изучите комментарии к шаблону
   2. Изучите публикацию
3. Внедрите во второй блок исходного текста коды символов своего имени.
4. Выполните 3 фазы атаки и сохраните итоговые скриншоты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

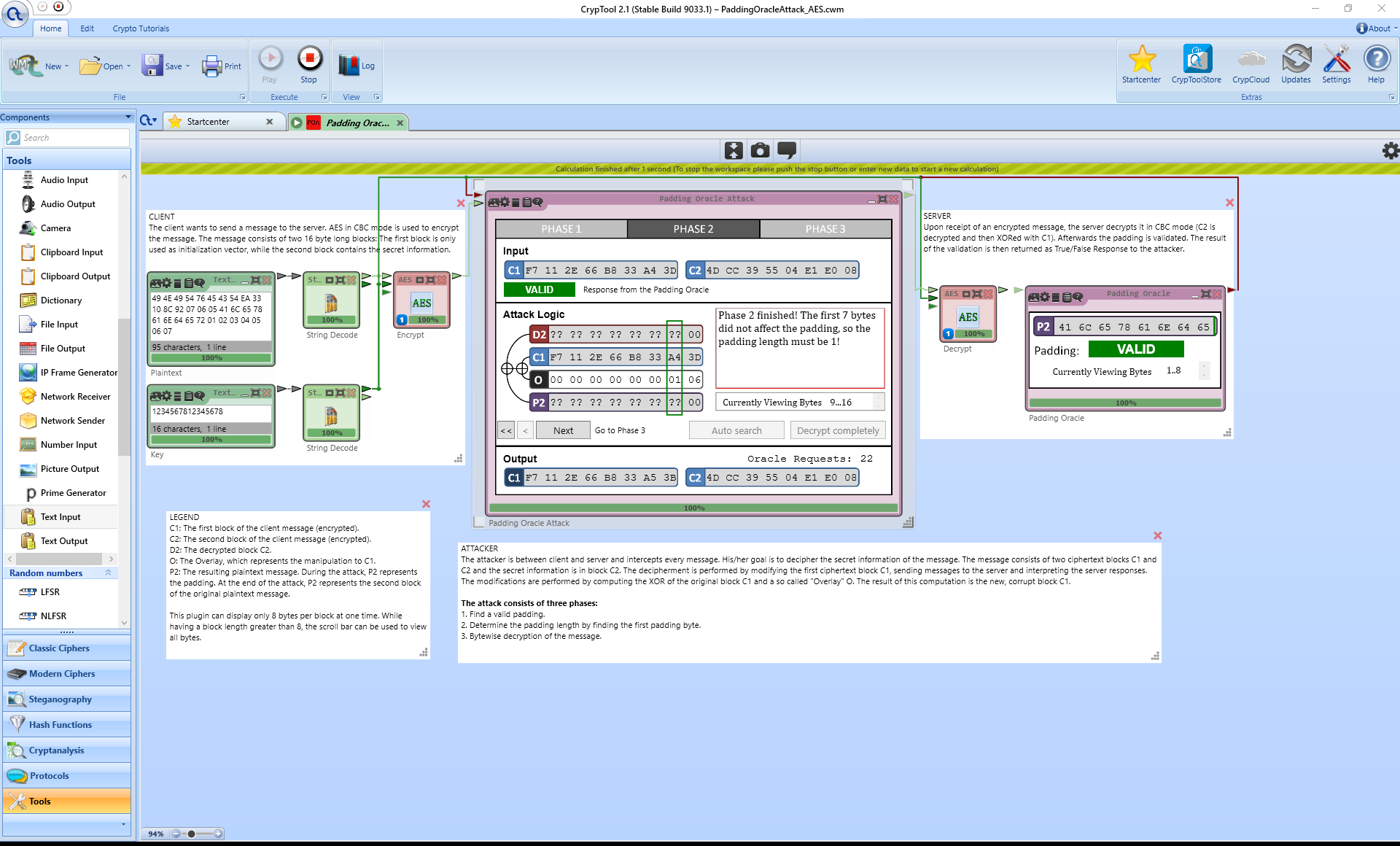
Исходные данные для экспериментов: открытый текст: [0x49 0x4e 0x49 0x54 0x76 0x45 0x43 0x54 oxea 0x33 0x10 0x8c 0x92 0x07 0x06 0x05 0x41 0x6c 0x65 0x78 0x61 0x6e 0x64 0x65 0x72 0x01 0x02 0x03 0x04 0x05 0x06 0x07], ключ: «1234567812345678».

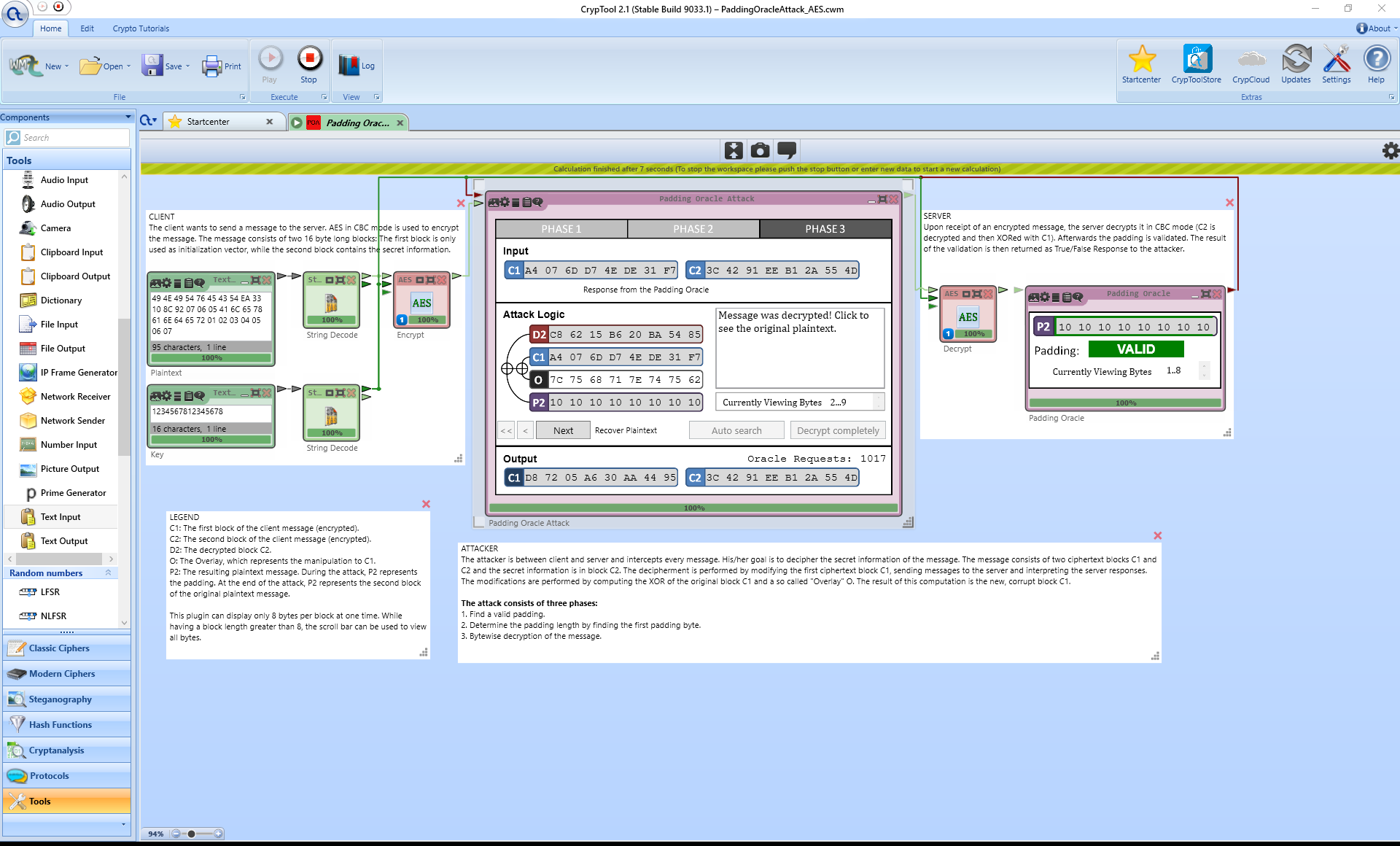
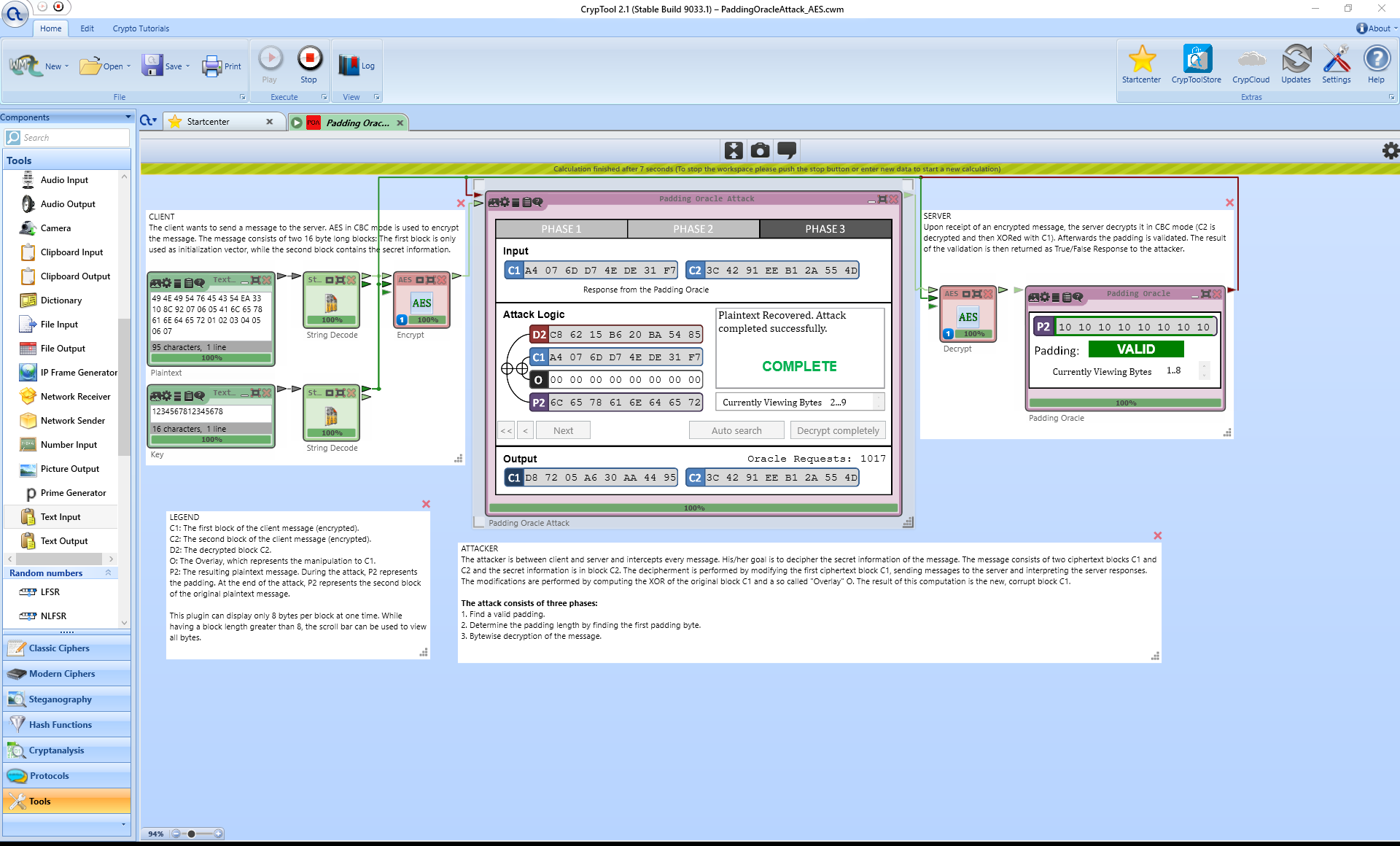
Шаблон атаки «Padding Oracle Attack» из CrypTool 2:

Описание атаки «Padding Oracle Attack»:

Результаты 3-х фаз атаки в виде итоговых скриншотов ПО:







**Выводы.**

1. AES (Rijndael) работает на основе перестановочно подстановочной сети (SP- сеть). Длина ключа может быть 128, 196, 256 бит. Длина блока шифрования 128 бит. Количество раундов зависит от длины ключа и составляет 10, 12, 14 для длины ключа 128, 196, 256 бит соответственно. Раунд шифрования состоит из 4 этапов:
   1. SubBytes – побайтовая подстановка в S-боксе с фиксированной таблицей замен;
   2. ShiftRows – побайтовый сдвиг строк матрицы State на различное количество байт;
   3. MixColumns – перемешивание байт в столбцах;
   4. AddRoundKey – сложение с раундовым ключом (операция XOR).

Перед выполнением раундов выполняется инициализирующий раунд, в ходе которого открытый текст складывается с ключом (операция XOR). Последний раунд отличается от предыдущих тем, что не задействует функцию MixColumns. При расшифровке осуществляются операции в указанном порядке InvShiftRows (обратная операции ShiftRows), InvSubBytes (обратная операции SubBytes), InvMixColumns (обратная процедуре MixColumns), AddRoundKey (является обратной сама к себе).

В математической основе шифра AES лежит поле Галуа GF(256) и опреации с ним связанные. Кажый байт представляется как многочлен вида:

Степень *x* меньше 8 (для того, чтобы многочлен можно было записать в один байт), а коэффициенты b7, b6 … b1, b0 принадлежат к {0,1}. Операция сложения реализовано как операция XOR. Умножение реализовано по модулю m(x), где m(x) = 𝑥8 + 𝑥4 + 𝑥3 + 𝑥 + 1 (11𝐵), т.е. (a(x) \* b(x)) mod m(x). Также для любого многочлена за исключением нулевого в поле Галуа существует обратный многочлен 𝑏(𝑥)−1 (обратный элемент: (𝑏(𝑥)−1 \* 𝑏(𝑥)) mod (m(x)) = 1).

1. Другие шифры-финалисты конкурса AES показывают сравнительно хорошие результаты при сравнении энтропии полученных шифротекстов. Сложность атаки грубой силой также примерно равна для всех шифров за исключением шифра Serpent, который показал практически в два раза лучшие результаты в сравнении со всеми остальными. Серьезных очевидных недостатков у шифров-финалистов в ходе лабораторной работы обнаружено не было.
2. Для каждого из трех вариантов ключей AES полный перебор требует 2128, 2192 или 2256 операций соответственно. Даже наименьшее из этих чисел свидетельствует, что атака с использованием перебора ключей не может быть применена на практике. Других эффективных атак на шифр AES до сих пор обнаружено не было.
3. Padding Oracle Attack - при проведении этой атаки предполагается, что нарушитель может модифицировать и отправлять зашифрованное сообщение серверу для расшифровки, а также распозновать ответы сервера о корректности дополнения последнего блока. Дешифровка сообщения нарушителем заключается в попытке узнать промежуточное значение алгоритма, полученное расшифровки блока текста, но до его сложения с предыдущим зашифрованным блоком (как того требует алгоритм CBC). В качестве предотвращения Padding Oracle Attack предлагается:
   1. Исправить уязвимость сервера.
   2. Использовать другой режим шифрования вместо CBC.
   3. Использовать другой протокол установки паддинга вместо PKCS.