**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ИБ**

**отчет**

**по лабораторной работе №5**

**по дисциплине «Криптографические методы защиты информации»**

**Тема:** **Изучение шифра AES**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 5381 |  | Кобылянский А.В. |
| Преподаватель |  | Племянников А.К. |

Санкт-Петербург

2018

**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.**

Задание

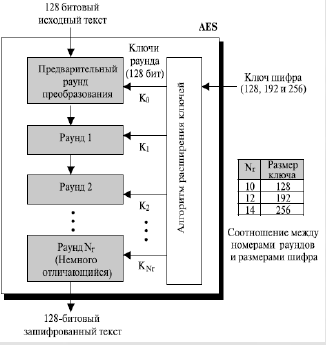
1. Изучить преобразования шифра AES с помощью демонстрационного приложения из Cryptool 1: *Indiv.Procedures->Visualization…->AES->Rijndael Animation.*
2. Выполнить вручную преобразования для одного раунда и вычисление раундового ключа при следующих исходных данных:
3. Открытый текст – фамилия\_имя (транслитерация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

Содержание раздела отчета

1. Формулировка задания.
2. Описание AES c примерами скриншотов из демоприложения.
3. Расчет матрицы состояний и раундового ключа шифра для одного раунда.
4. Скриншоты приложения-инспектора, подтверждающие корректность расчетов.
5. Выводы.

**1.2 Описание AES c примерами скриншотов из демоприложения.**

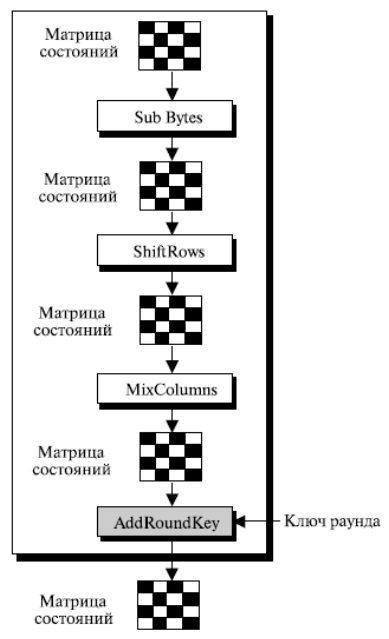
Шифр Rijndael (AES) работает на основе перестановочно-подстановочной сети (SP-сеть). Обобщенная схема работы алгоритма представлена на рисунке 5.1.

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:

Рисунок 5.1

1. Столбцы матрицы состояний складываются с ключом шифра операцией xor.
2. Полученный текст проходит через преобразование подстановки SubBytes.
3. Циклический сдвиг влево всех строк матрицы состояний преобразованием ShiftRows .
4. Смешивание столбцов матрицы состояний путем ее умножения на матрицу констант в конечном поле – преобразовние MixColumns Сложение полученных столбцов матрицы состояний с раундовым ключом операцией xor – преобразование AddRoundKey
5. Действия 2-5 повторяются в каждом из 10-ти раундов

*Рисунок 5.2*

1. Последний раунд не включает в себя смешивание столбцов.

Расшифровывание выполняется применением обратных операций в обратной последовательности.

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

Текст: *kobilianskyalexe*

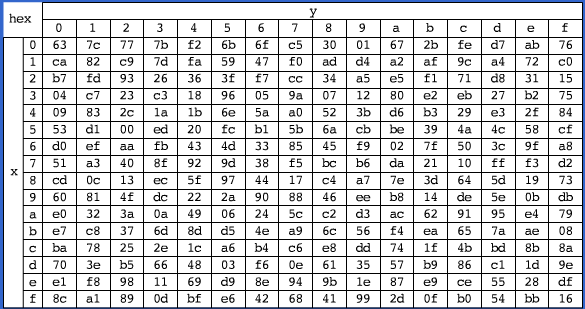
Ключ: 5381\_vasilievich

Матрицы открытого текста и ключа:

Первый раундовый ключ:

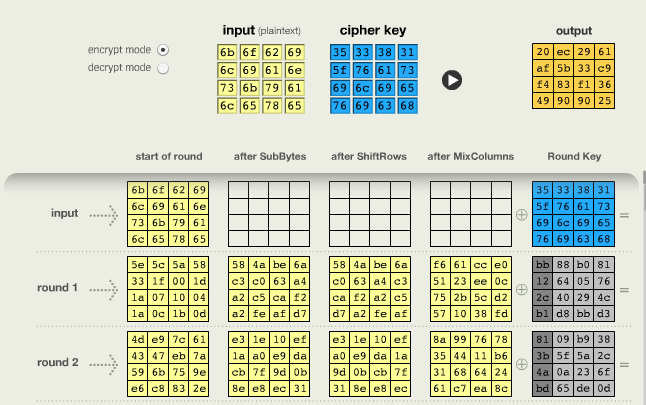
Первый раунд шифрования.

1. Сложение с ключом:
2. SubBytes



1. ShiftRows
2. MixColumns
3. AddRoundKey

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



**1.5 Выводы.**

Текста после первого раунда ручного шифрования совпал с текстом из приложения-инспектора.

Зашифровать текст вручную с помощью AES возможно, хотя это и займет очень много времени. Особенно сложно выполнить шаг MixColumns, где нужно перемножать матрицы в .

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

**2.1. Формулировка задания.**

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.

Содержание раздела отчета

1. Формулировка задания.
2. Исходные данные для экспериментов:
3. Исходный текст
4. Секретный ключ
5. Таблица с результатами качества зашифрования исследованными шифрами.
6. Таблица с результатами трудоемкости атаки «грубой силы» для исследованных шифров.
7. Выводы.
   1. **Исходные данные для экспериментов:**

**Исходный текст:**

It was early morning in the Crystal Empire, the sun just raising over the horizon as Princess Flurry Heart was in her bed

**Секретный ключ:**

11-11-11-11-11-11-11-11-11-11-11-11-11-11-11-11

* 1. **Таблица с результатами качества зашифрования исследованными шифрами.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Исходный | AES | MARS | RC6 | Serpent | Twofish |
| Энтропия | 4.25 | 6.59 | 6.69 | 6.57 | 6.52 | 6.62 |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Известно  байт | AES | MARS | RC6 | Serpent | Twofish |
| 2 | 3.8e20 | 5.1e20 | 3.1e20 | 1.1e21 | 7.2e20 |
| 4 | 5.9e15 | 7.8e15 | 4.8e15 | 1.7e16 | 1.1e16 |
| 6 | 8.9e10 | 1.2e11 | 7.3e10 | 2.6e11 | 1.7e11 |
| 8 | 1.4e6 | 1.8e6 | 1.1e6 | 3.9e6 | 2.6e6 |
| 10 | 21 | 21 | 17 | 60 | 39 |
| 12 | 2:45 h | 3:40 h | 2:16 h | 7:55 h | 5:10 h |
| 14 | 0 s | 0 s | 0 s | 1 s | 0 s |

* 1. **Выводы**

Энтропия максимальна у шифра Mars, минимальна у Serpent.

Дольше всего атака брутфорсом осуществляется для шифра Serpent, быстрее всего для RC6.

AES показал средние значения в обоих категориях.

Скорость подбора пароля в данном случае показывает не столько стойкость шифрования, столько быстродействие алгоритма. Это связано с тем, что чем дольше происходит расшифровка с помощью алгоритма, тем больше в итоге требуется время на подбор ключа.

Из всего выше сказанного можно заметить, что шифр Rijndael, который сейчас называется AES, образует шифротекст с высокой степенью случайности и при этом работает очень быстро. Указанные факторы были одними из основных на конкурсе AES, что и позволило алгоритму Rijndael победить.

1. **Атака «грубой силы» на AES**

**3.1. Формулировка задания.**

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 со 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 процессорных ядер. Зафиксировать затраты времени.

Содержание раздела отчета

1. Формулировка задания.
2. Исходные данные для экспериментов:
3. Исходный текст
4. Секретный ключ
5. Шаблон атаки «грубой силы» из Cryptool 2.
6. Таблица с результатами трудоемкости энтропийной атаки «грубой силы» для различных вариантов знаний о ключе и количестве задействованных процессорных ядер.
7. Таблица с результатами трудоемкости текстовой атаки «грубой силы» для различных вариантов знаний о ключе и количестве задействованных процессорных ядер.
8. Выводы.
   1. **Исходные данные для экспериментов:**

**Исходный текст 1:**

It was early morning in the Crystal Empire, the sun just raising over the horizon as Princess Flurry Heart was in her bed, with the covers over her, covering her entire body and even her head as she wanted to sleep in. Ever since she wanted more responsibilities, her parents, Princess Cadance and Prince Shining Armor had been sending her on lots of diplomatic missions, most of them have been easy going and she did have Moongleam to accompany her, but it was tiring just to travel back and forth.

As Flurry slept, a flugelhorn sounded right next to her as Flurry jumped up in fright, with her blankets still over her. Taking her blankets off, she saw a guard standing besides her bed.

"Your parents summon you to the throne room, my princess." The crystal guard announced.

"Yes, thank you very much…" Flurry rubbing one of her eyes as she was a bit annoyed having to be woken up in such a manner.

After the guard left and Flurry was able to make herself look presentable, since she had bedmane, she went to the throne room, where her parents were waiting for her.

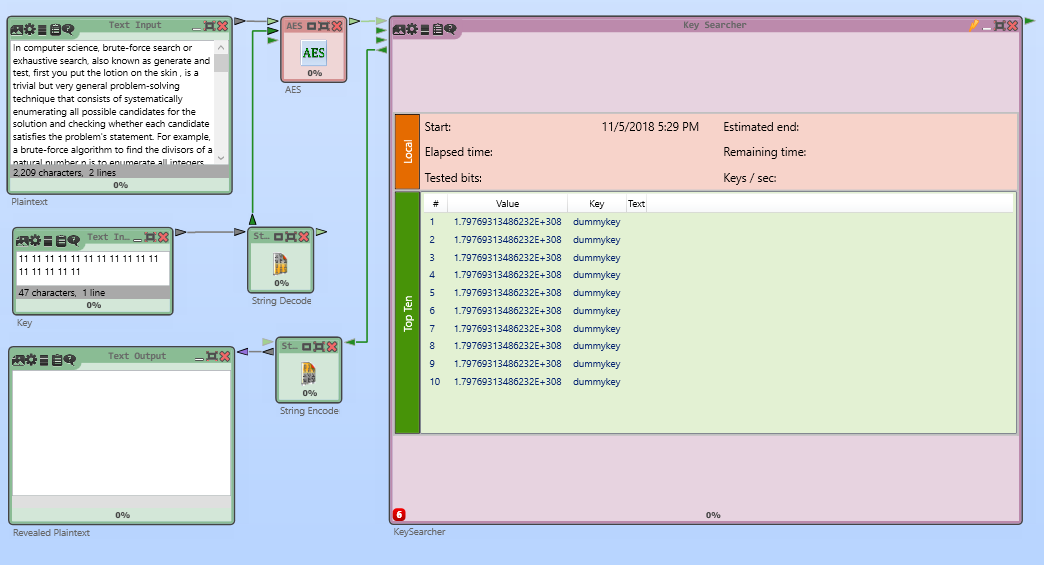
**Исходный текст 2:**

DEAR SIRS In computer science, brute-force search or exhaustive search, also known as generate and test, first you put the lotion on the skin , is a trivial but very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement. For example, a brute-force algorithm to find the divisors of a natural number n is to enumerate all integers from 1 to the square-root of n, and check whether each of them divides n without remainder. For another example, consider the popular eight queens puzzle, which asks to place eight queens on a standard chessboard so that no queen attacks any other. A brute-force approach would examine all possible arrangements of 8 pieces in the 64 squares, and, for each arrangement, check whether any queen attacks any other. Brute-force search is simple to implement, and will always find a solution if it exists. THANKS

**Секретный ключ:**

11-11-11-11-11-11-11-11-11-11-11-11-11-11-11-11

**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.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Количество, задействованных ядер** | **Известные байты ключа** | | |
| **10** | **12** | **14** |
| **Время атаки** | | |
| **1** | 24 года | 3 часа | 1 секунда |
| **2** | 13,5 лет | 1 час 40 минут | 1 секунда |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Количество, задействованных ядер** | **Известные байты ключа** | | |
| **10** | **12** | **14** |
| **Время атаки** | | |
| **1** | 5 лет | 41 минута | 1 секунда |
| **2** | 3 года | 33 минуты | 1 секунда |

**3.6 Выводы**

Атака со знанием части открытого текста осуществляется быстрее примерно в 4 раза, чем атака брутфорсом, т.е. не на порядки. Использование 2 ядер вместо одного уменьшает время перебора менее чем в 2 раза.

Взломать шифр зная меньше 10 байт ключа за разумное время все еще не реально.

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

**4.1 Формулировка задания.**

Задание

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

Содержание раздела отчета

1. Формулировка задания.
2. Исходные данные для экспериментов:
3. Исходный текст
4. Секретный ключ
5. Шаблон атаки «Padding Oracle Attack» из CrypTool 2.
6. Описание атаки «Padding Oracle Attack».
7. Результаты 3-х фазы атак в виде итоговых скриншотов ПО.
8. Выводы.

**4.2 Исходные данные для экспериментов:**

**a. Исходный текст**

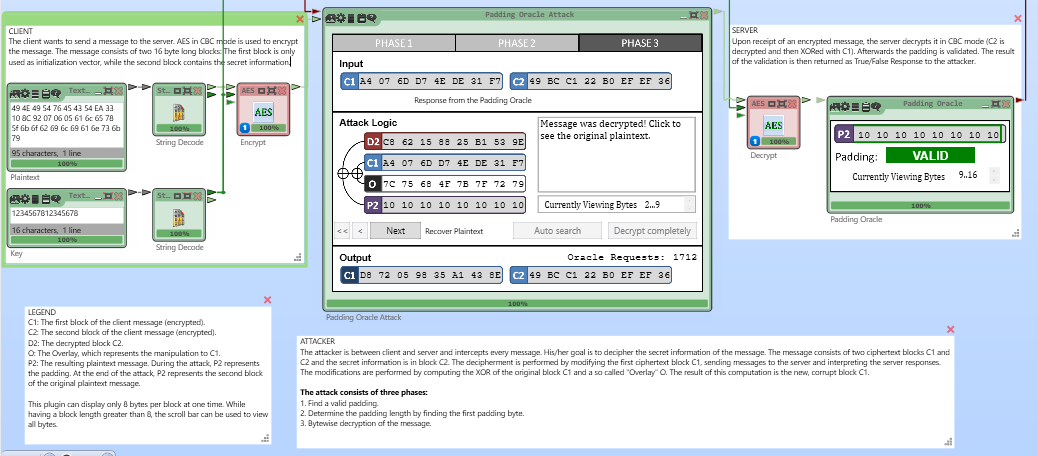
текст: alex\_kobiliansky

hex: 61 6c 65 78 5f 6b 6f 62 69 6c 69 61 6e 73 6b 79

**b. Секретный ключ**

1234567812345678

**4.3 Шаблон атаки «Padding Oracle Attack» из CrypTool 2.**



* 1. **Описание атаки «Padding Oracle Attack».**

При атаке Padding Oracle предполагается, что мы можем отправлять сообщения серверу на расшифровку, который может возвращать ответ, корректно ли выполнено дополнение последнего блока. Расшифровка сообщения происходит с последних блоков шифротекста.

Рассмотрим расшифровку блока .

1. Формируем *R* – все биты, кроме последнего, случайные значения. Перебираем байт от 0x00 до 0xFF, каждый раз посылая на сервер ]. Если при некотором сервер «одобряет», то . Схема первого этапа представлена на рисунке 5.3.

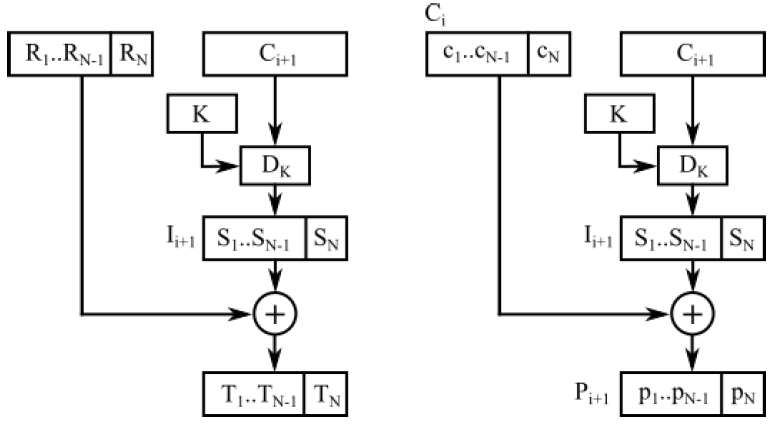


Рисунок 5.3

– *открытый текст, – шифротекст, – дополнение, – ключ, – функция расшифровки, – дополнение.*

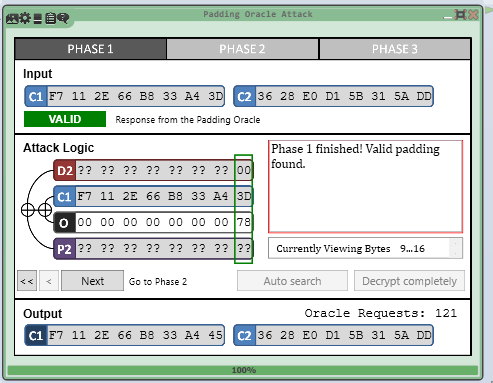
1. *Формируем R* – все биты, кроме двух последних, случайные значения. , чтобы . Перебираем байт от 0x00 до 0xFF, каждый раз посылая на сервер ]. Если при некотором сервер «одобряет», то .

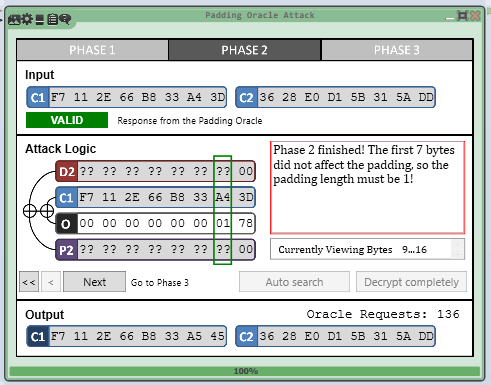
На третьем шаге пытаемся получить дополнение 030303, на четвертом – 04040404. После N шагов получаем полностью блок .

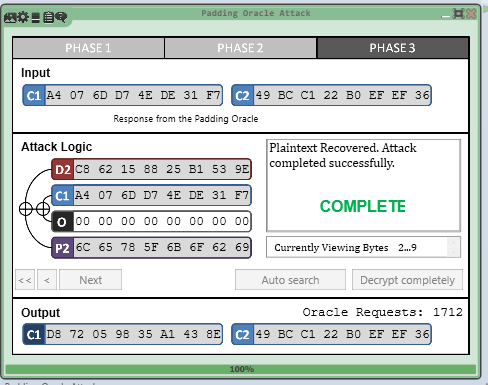
Более подробное описание атаки Padding Oracle можно найти в статье [4].

В CryoTool 2 атака предсказанием дополнения реализована в три фазы:

1. *Фаза 1.* Нахождение длины дополнения, т.е. последний байт
2. *Фаза 2*. Подбор дополнения
3. *Фаза 3*. Расшифровка текста
   1. **Результаты 3-х фазы атак в виде итоговых скриншотов ПО**







* 1. **Выводы**

Блочные CBC шифры сильно уязвимы к Padding Oracle Attack. Расшифровка перехваченного текста данной атакой осуществляется за линейное, от длины текста, время и занимает секунды на современном оборудовании.

**Вывод**

Rijndael основан на простых алгебраических операциях, образующих т.н. SP-сеть. В нём используются подстановки, перестановки, отбеливание и перемешивание значений блока.

Шифр AES является очень эффективным, так как образует шифротекст с высокой степенью случайности и работает очень быстро. Шифр до сих пор не потерял своей актуальности. К достоинствам алгоритма Rijndael можно отнести:

* Обеспечивает быстрое перемешивание информации, при этом за один раунд преобразованию подвергается весь входной блок
* Байт ориентированная структура удобна для реализации на восьмиразрядных микроконтроллерах
* Все раундовые преобразования – это операции в конечных полях, допускающие эффективную аппаратную и программную реализацию на различных платформах
* Преобразование раунда допускает параллельное выполнение, что является важным преимуществом для будущих процессоров и специализированной аппаратур

Шифр невозможно расшифровать буртфорсом за разумное время, зная менее 10 байт ключа, ни атакой по энтропии, ни атакой по тексту.

AES CBC уязвим к Padding Oracle Attack. Без ограничения на запросы к ораклу, текст расшифровывается за секунды.