AES

Длина блока и ключа 128 бит (16 байт)

Количество раундов – 10 (Для 128-битного ключа)

Шифруемое слово – AESTESTWORDCRYPT = 41 45 53 54 45 53 54 57 4F 52 44 43 52 59 50 54

Ключ – THISKEYFORAESALG = 54 48 49 53 4B 45 59 46 4F 52 41 45 53 41 4C 47

Nb = 128/32 = 4 (Количество колонок)

Nb = 128/32 = 4 (Количество слов)

Начальный блок состояния (S):

41	45	53	54
45	53	54	57
4F	52	44	43
52	59	50	54

Начальный ключ состояния, разделённый на слова (K):

54	48	49	53
4B	45	50	46
4F	52	41	45
53	41	4C	47

Генерация первого раундового ключа:

Sbox (16 X 16):

63, 7c, 77, 7b, f2, 6b, 6f, c5, 30, 01, 67, 2b, fe, d7, ab, 76,

ca, 82, c9, 7d, fa, 59, 47, f0, ad, d4, a2, af, 9c, a4, 72, c0,

b7, fd, 93, 26, 36, 3f, f7, cc, 34, a5, e5, f1, 71, d8, 31, 15,

04, c7, 23, c3, 18, 96, 05, 9a, 07, 12, 80, e2, eb, 27, b2, 75,

09, 83, 2c, 1a, 1b, 6e, 5a, a0, 52, 3b, d6, b3, 29, e3, 2f, 84,

53, d1, 00, ed, 20, fc, b1, 5b, 6a, cb, be, 39, 4a, 4c, 58, cf,

d0, ef, aa, fb, 43, 4d, 33, 85, 45, f9, 02, 7f, 50, 3c, 9f, a8,

51, a3, 40, 8f, 92, 9d, 38, f5, bc, b6, da, 21, 10, ff, f3, d2,

cd, 0c, 13, ec, 5f, 97, 44, 17, c4, a7, 7e, 3d, 64, 5d, 19, 73,

60, 81, 4f, dc, 22, 2a, 90, 88, 46, ee, b8, 14, de, 5e, 0b, db,

e0, 32, 3a, 0a, 49, 06, 24, 5c, c2, d3, ac, 62, 91, 95, e4, 79,

e7, c8, 37, 6d, 8d, d5, 4e, a9, 6c, 56, f4, ea, 65, 7a, ae, 08,

ba, 78, 25, 2e, 1c, a6, b4, c6, e8, dd, 74, 1f, 4b, bd, 8b, 8a,

70, 3e, b5, 66, 48, 03, f6, 0e, 61, 35, 57, b9, 86, c1, 1d, 9e,

e1, f8, 98, 11, 69, d9, 8e, 94, 9b, 1e, 87, e9, ce, 55, 28, df,

8c, a1, 89, 0d, bf, e6, 42, 68, 41, 99, 2d, 0f, b0, 54, bb, 16

Rcon (Массив раундовых констант):

0x01000000, 0x02000000, 0x04000000, 0x08000000, 0x10000000, 0x20000000, 0x40000000, 0x80000000, 0x1b000000, 0x36000000

Пример для байта 54

54 = 0101 0100 – Два полубайта

0101 – строка

0100 – столбец

Sbox(0101, 0100) = Sbox(5, 3) = 20

Полная подстановка:

54 48 49 53 → 20 52 3B ED

4B 45 50 46 → B3 6E 53 5A

4F 52 41 45 → 84 00 83 6E

53 41 4C 47 → ED 83 29 A0

Побитовый цикличный сдвиг влево <<

20 52 3B ED → 52 3B ED 20

B3 6E 53 5A → 6E 53 5A B3

84 00 83 6E → 00 83 6E 84

ED 83 29 A0 → 83 29 A0 ED

Снова пропускаем матрицу через Sbox:

52 3B ED 20 → 00 E2 55 B7

6E 53 5A B3 → 9F ED BE 6D

00 83 6E 84 → 63 EC 9F 5F

83 29 A0 ED → EC A5 E0 55

Первоначальный ключ XOR матрица, полученная в прошлом шаге:

54 48 49 53		00 E2 55 B7		54 DA 1C E4
4B 45 50 46		9F ED BE 6D		D4 68 1C 7B
4F 52 41 45	XOR	63 EC 9F 5F	=	2C BE 9E 7A
53 41 4C 47		EC A5 E0 55		BF F6 9C 12

Добавление константы Rcon[0] к каждому байту :

54 DA 1C E4		01 00 00 00		55 DA 1C E4
D4 68 1C 7B		01 00 00 00		D5 68 1C 7B
2C BE 9E 7A	XOR	01 00 00 00	=	2D BE 9E 7A
BF F6 9C 12		01 00 00 00		BE F6 9C 12

Ключ для 1 раунда шифрования:

55 DA 1C E4

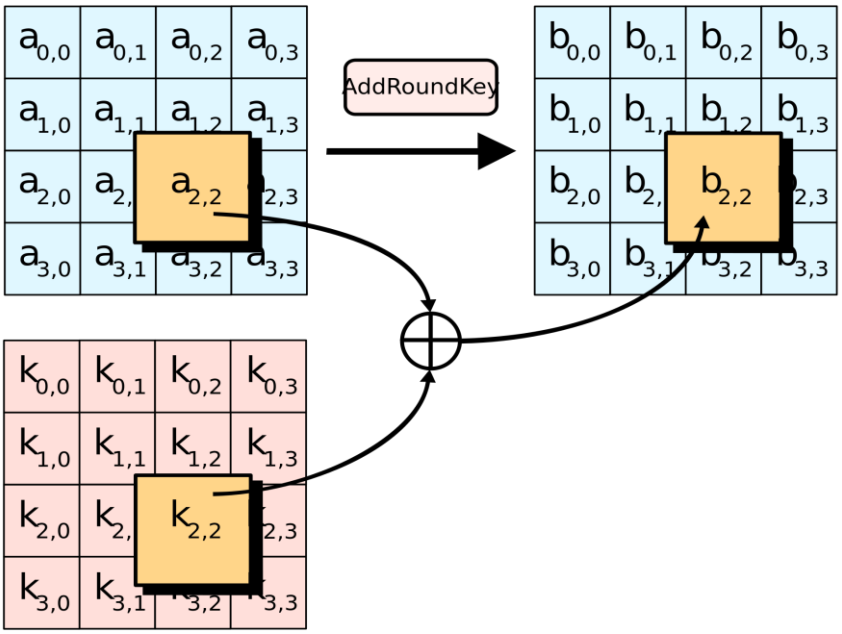
D5 68 1C 7B

2D BE 9E 7A

BE F6 9C 12

Шифрование

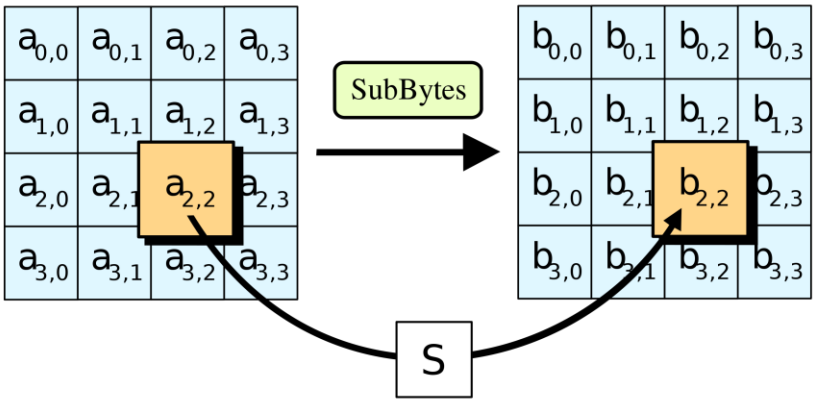
1. ***AddRoundKey***



S XOR K, где S это шифруемые данные, а K – ключ шифрования для раунда

41 45 53 54		55 DA 1C E4		14 9F 4F B0
45 53 54 57		D5 68 1C 7B		90 3B 48 2C
4F 52 44 43	XOR	2D BE 9E 7A	=	62 EC DA 39
52 59 50 54		BE F6 9C 12		EC 36 F3 46

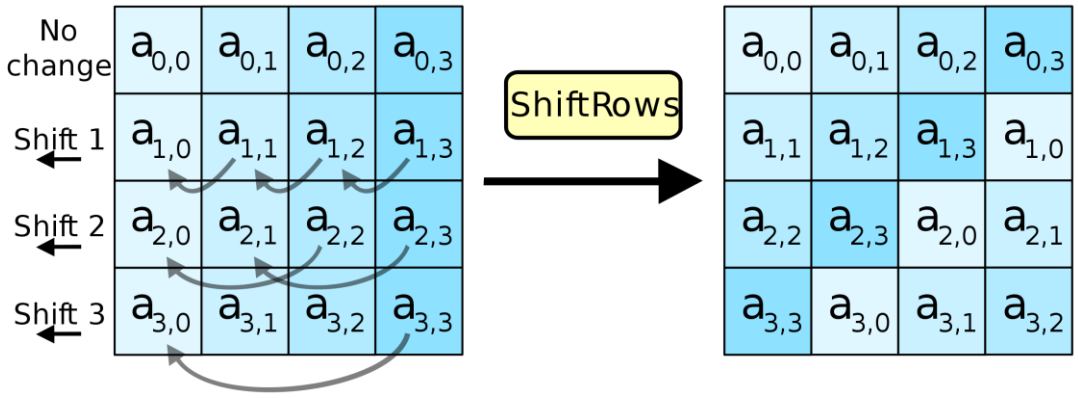
2. *SubBytes*



Замена байтов на значения из Sbox

14 9F 4F B0		FA D8 84 E7
90 3B 48 2C		60 E2 52 71
62 EC DA 39	->	AA CE 57 12
EC 36 F3 46		CE 05 0D 5A

3. *ShiftRows*



Побайтовый сдвиг.

Строка 1 остаётся без изменений

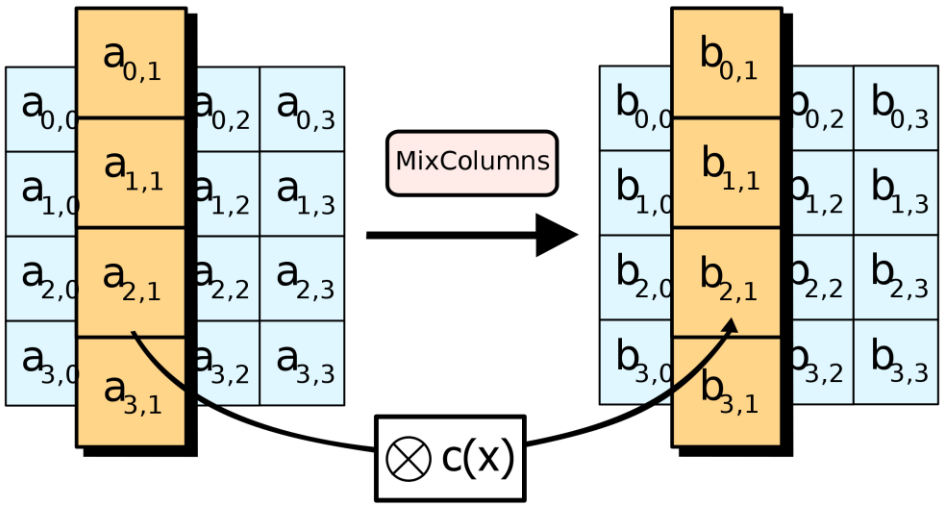
Строка 2 сдвигается на 1 байт

Строка 3 сдвигается на 2 байта

Строка 4 сдвигается на 3 байта

FA D8 84 E7		FA D8 84 E7
60 E2 52 71		E2 52 71 60
AA CE 57 12	<<	57 12 AA CE
CE 05 0D 5A		5A CE 05 0D

4. MixCoulmns



Fixed matrix	=	02 03 01 01
		01 02 03 01
		01 01 02 03
		03 01 01 02

Columns:	FA	D8	84	E7
	E2	52	71	60
	57	12	AA	CE
	5A	CE	05	0D

NewColumn[i]=FixedMatrix[i,0]·Column[0]⊕FixedMatrix[i,1]·Column[1]⊕FixedMatrix[i,2]·Column[2]⊕FixedMatrix[i,3]·Column[3]

[0] = (0x02·0xFA)⊕(0x03·0xE2)⊕(0x01·0x57)⊕(0x01·0x5A) = DF

[1] = (0x01·0xFA)⊕(0x02·0xE2)⊕(0x03·0x57)⊕(0x01·0x5A) = 86

[2] = (0x01·0xFA)⊕(0x01·0xE2)⊕(0x02·0x57)⊕(0x03·0x5A) = 58

[3] = (0x03·0xFA)⊕(0x01·0xE2)⊕(0x01·0x57)⊕(0x02·0x5A) = 14

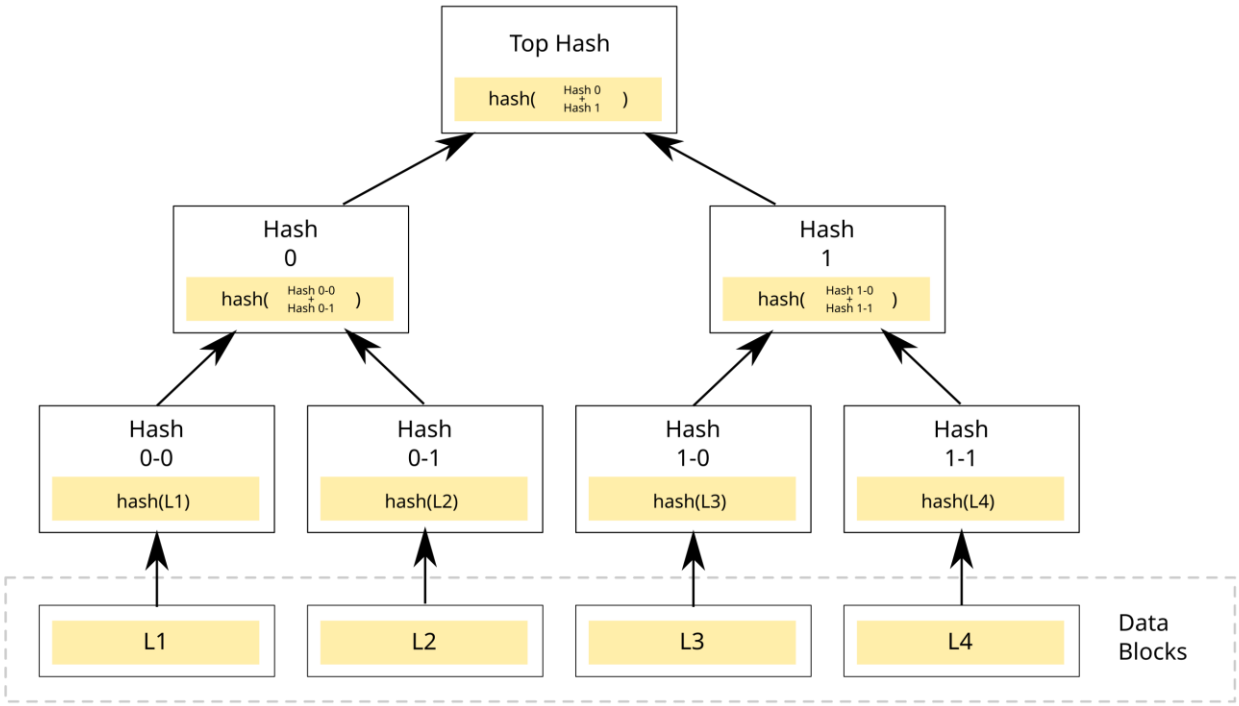
DF 81 2F B6

86 84 86 63

58 E7 B5 17

14 B4 46 86

Merkle



Генерация ключей

Пусть n=4, то есть дерево имеет 4 листа (максимум 4 подписи).

- Закрытые ключи
    - sk0="privatekey0"
    - sk1="privatekey1"
    - sk2="privatekey2"
    - sk3="privatekey3"
- Открытые ключи

Хэшируем каждый закрытый ключ для получения открытого ключа через алгоритм хеширования H

pk0=H(sk0) = "d2b6..."

pk1=H(sk1) = "f3a1..."

pk2=H(sk2) = "a4f9..."

pk3=H(sk3) = "e1c3..."

Построение дерева

Каждый публичный ключ является одним начальным узлом дерева

Вычисление промежуточных узлов происходит путём хеширования пары соседних узлов, например

L01 = H(pk0 || pk1)

Посредством хеширования двух промежуточных узлов получаем корневой узел root

Root = H(H01 ‖ H23)

Формирование подписи

Делаем подпись на основе одного из ключей skn

Находим соседний узел pkn+-1 для хеша ключа skn

Находим соседний родительский узел A

Формируем путь Proof = [pkn+-1, A]

Формируем подпись Signature = (skn, proof, n)

Проверка подписи

Хешируем skn и получаем публичный ключ pkn

Получаем родительский узел Hnn+-1 хешируя H(pkn || pkn+-1)

Хешируем полученный узел вместе с узлом A root’ = H(A || Hnn+-1)

Сравниваем root’ с публичным root