AES

Presentation by:
V. Balasubramanian
SSN College of Engineering

Objectives

- Overview of the AES algorithm
- Internal structure of AES
  - Byte Substitution layer
  - Diffusion layer
  - Key Addition layer
  - Key schedule
- Decryption

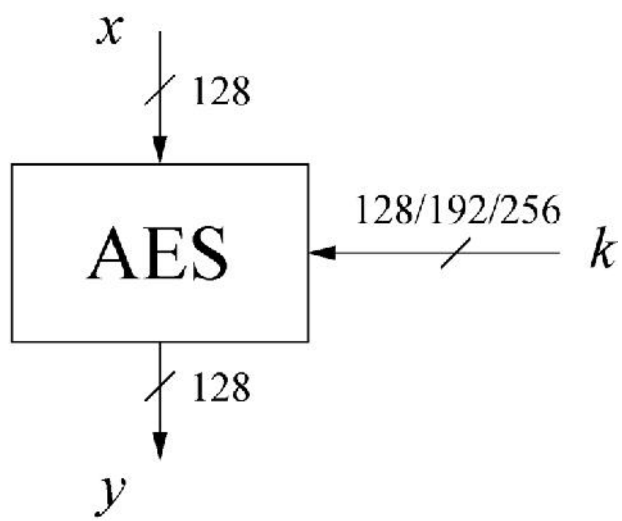
Introduction

- AES is the most widely used symmetric cipher today
- The algorithm for AES was chosen by the US *National Institute of Standards and Technology* (NIST) in a multi-year selection process
- The requirements for all AES candidate submissions were:
  - Block cipher with **128-bit block size**
  - **Three supported key lengths**: 128, 192 and 256 bit
  - Security relative to other submitted algorithms
  - **Efficiency** in software and hardware

Introduction

- The need for a new block cipher announced by NIST in January, 1997
- 15 candidates algorithms accepted in August, 1998
- 5 finalists announced in August, 1999:
  - *Mars* – IBM Corporation
  - *RC6* – *RSA* Laboratories
  - *Rijndael* – J. Daemen & V. Rijmen
  - *Serpent* – Eli Biham et al.
  - *Twofish* – B. Schneier et al.
- In October 2000, *Rijndael* was chosen as the AES
- AES was formally approved as a US federal standard in November 2001

AES: Overview

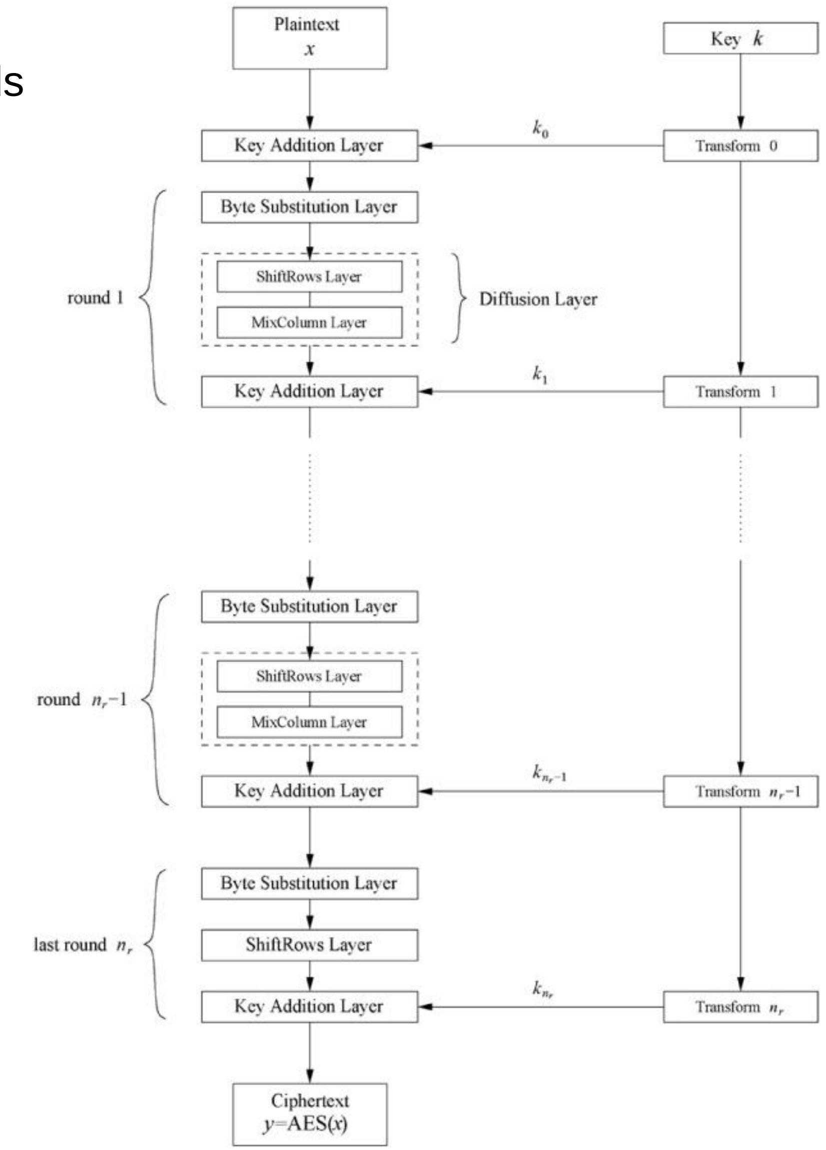


The number of rounds depends on the chosen key length:

Key length (bits)	Number of rounds
128	10
192	12
256	14

AES: Overview

- Iterated cipher with 10/12/14 rounds
- Each round consists of “Layers”



Internals

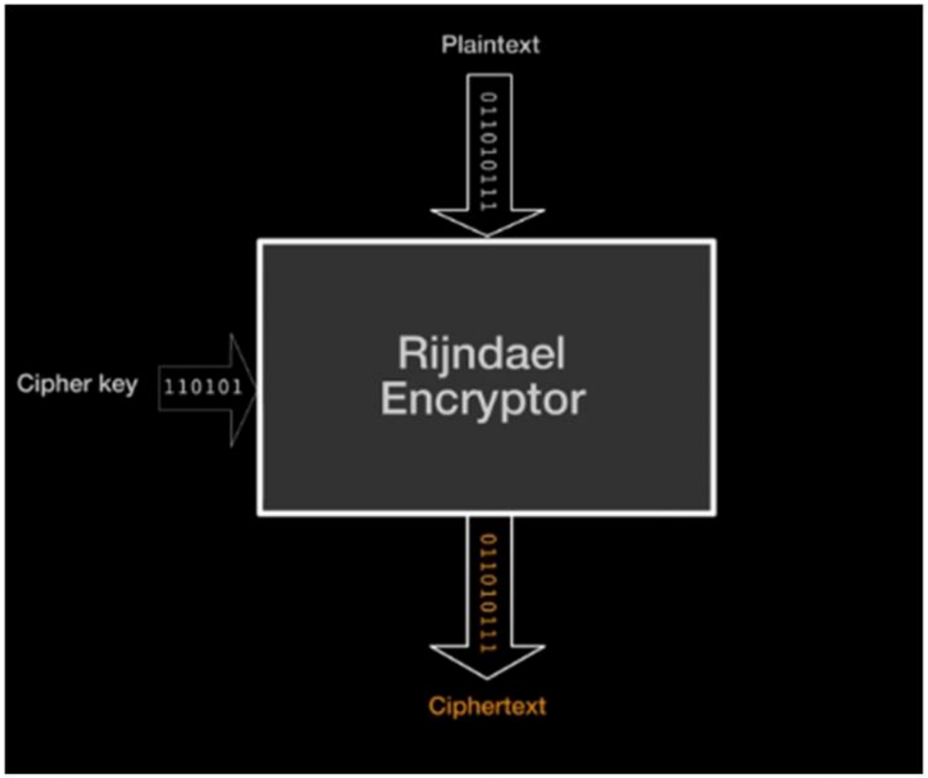
Key Addition layer A 128-bit round key, or subkey, which has been derived from the main key in the key schedule, is XORed to the state.

Byte Substitution layer (S-Box) Each element of the state is nonlinearly transformed using lookup tables with special mathematical properties. This introduces *confusion* to the data, i.e., it assures that changes in individual state bits propagate quickly across the data path.

Diffusion layer It provides *diffusion* over all state bits. It consists of two sublayers, both of which perform linear operations:

- The *ShiftRows* layer permutes the data on a byte level.
- The *MixColumn* layer is a matrix operation which combines (mixes) blocks of four bytes.

Similar to DES, the key schedule computes round keys, or subkeys, $(k_0, k_1, \ldots, k_{n_r})$ from the original AES key.



Plaintext
011010111
Rijndael
Encryptor
Cipher key
110101
011010111
Ciphertext

State

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

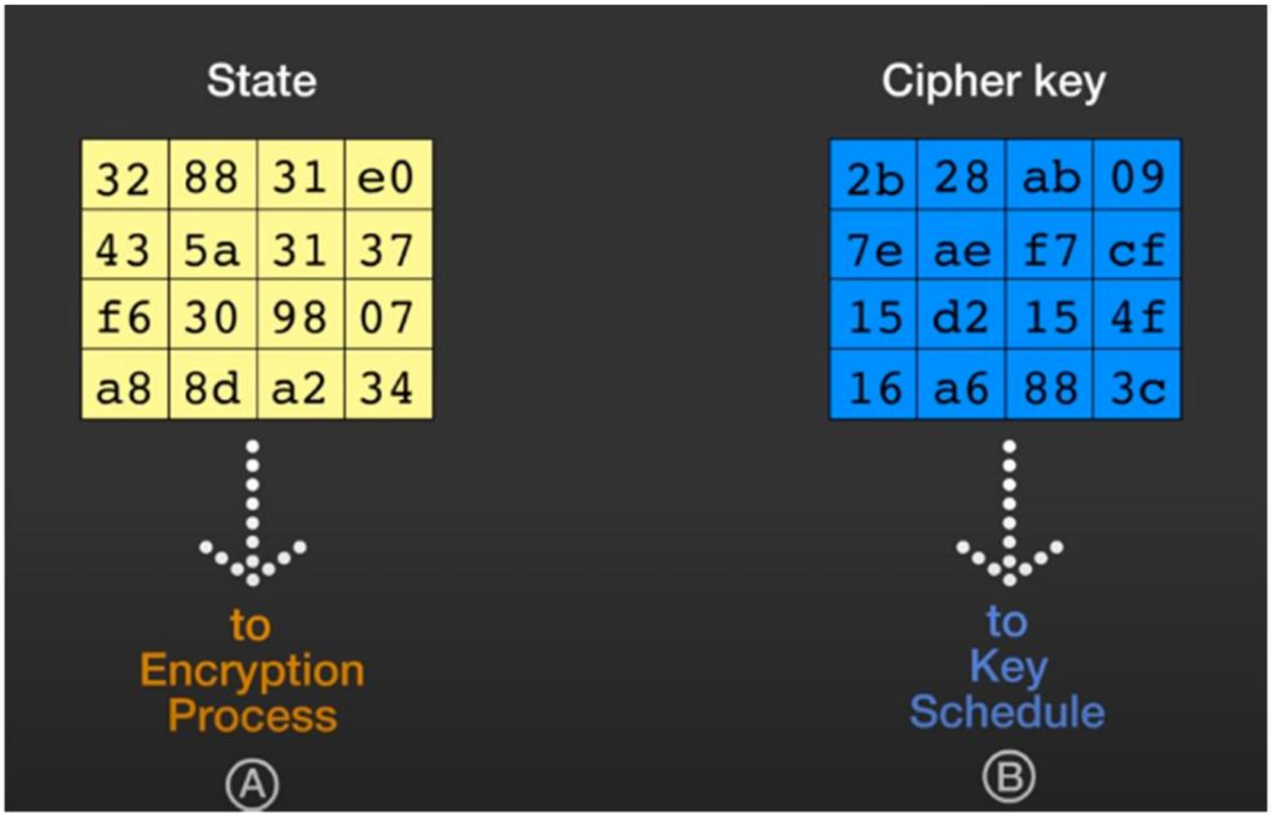
This is a block from the plaintext message to be encrypted.

Cipher key

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Hexadecimal notation (sample):

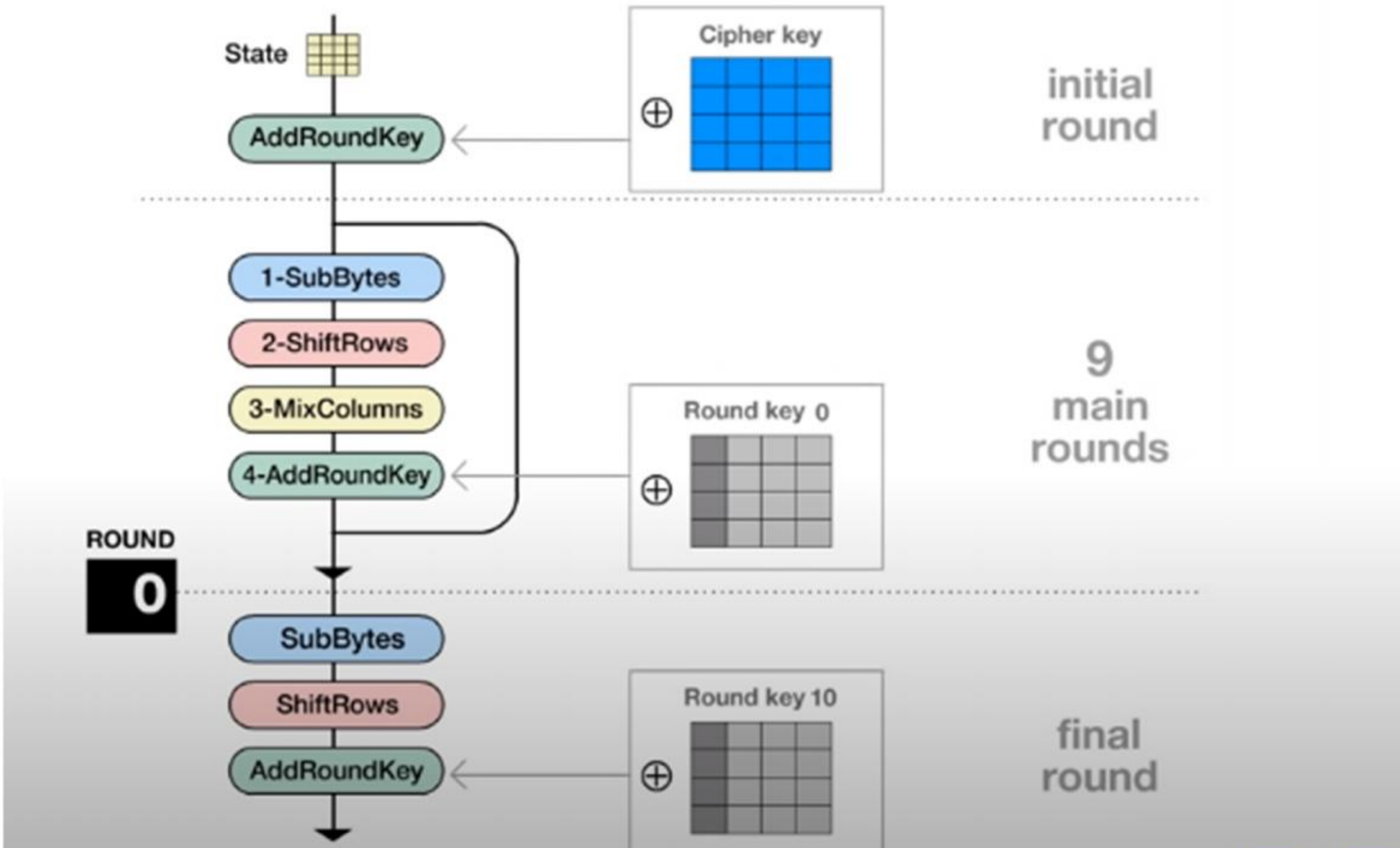
32 = $\underbrace{0011}_{\text{3hex}}\underbrace{0010}_{\text{2hex}}$ (1 byte)

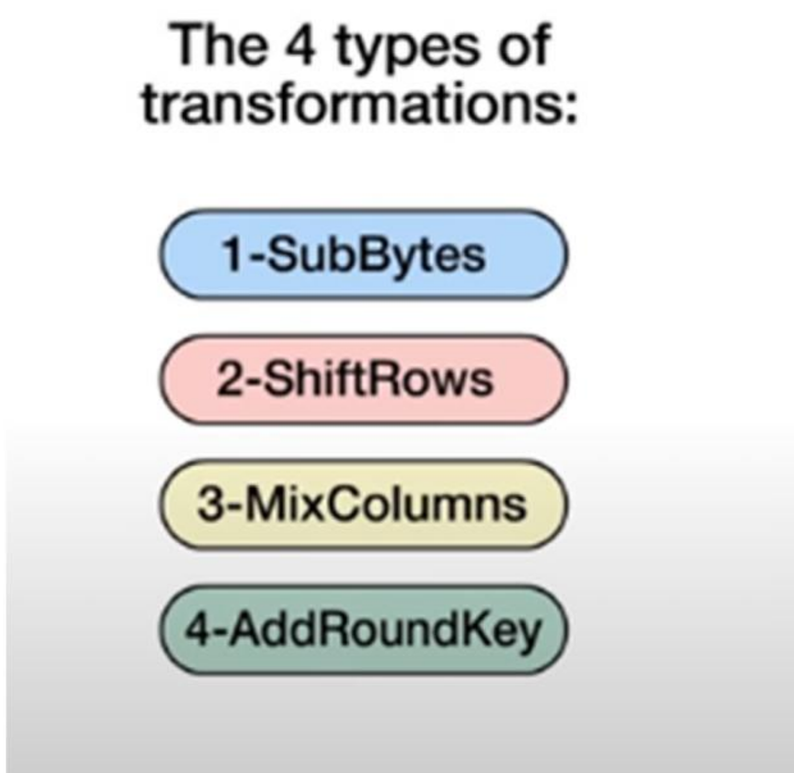

State
32 88 31 e0
43 5a 31 37
f6 30 98 07
a8 8d a2 34
to
Encryption
Process
A
Cipher key
2b 28 ab 09
7e ae f7 cf
15 d2 15 4f
16 a6 88 3c
to
Key
Schedule
B

Ⓐ

Encryption Process

(Performing the encryption of the given plaintext block using 4 different transformations in the initial round, the 9 main rounds and the final round)


State
AddRoundKey
Cipher key
initial round
1-SubBytes
2-ShiftRows
3-MixColumns
4-AddRoundKey
Round key 0
9 main rounds
ROUND
0
SubBytes
ShiftRows
AddRoundKey
Round key 10
final round


The 4 types of
transformations:
1-SubBytes
2-ShiftRows
3-MixColumns
4-AddRoundKey

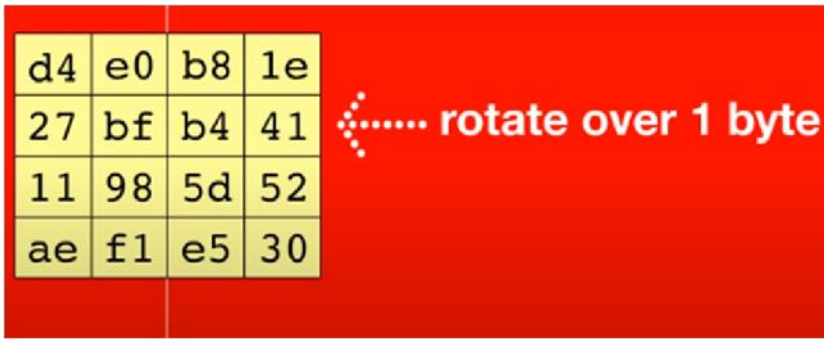
19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

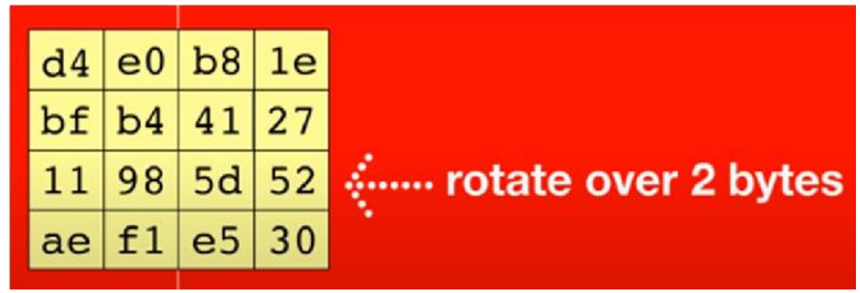
hex	y: 0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x: 0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	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	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

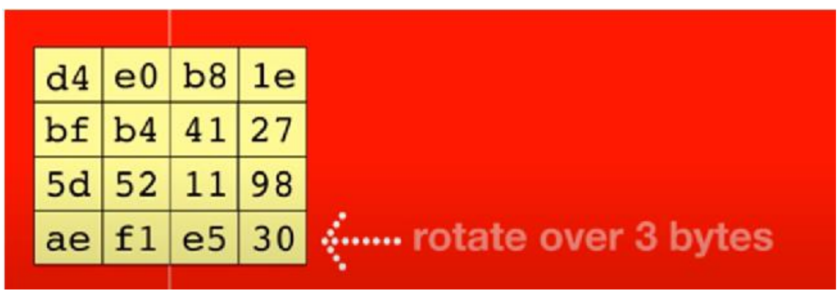
S-BOX byte substitution table

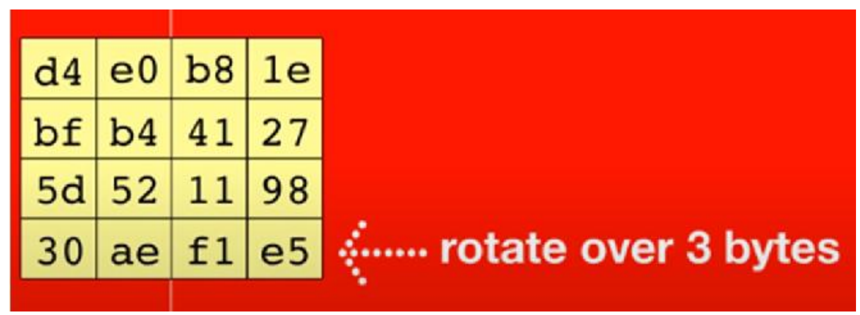
d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

Shift Rows


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


d4 e0 b8 1e
bf b4 41 27
11 98 5d 52
ae f1 e5 30
rotate over 2 bytes


d4 e0 b8 1e
bf b4 41 27
5d 52 11 98
ae f1 e5 30
rotate over 3 bytes


d4 e0 b8 1e
bf b4 41 27
5d 52 11 98
30 ae f1 e5
rotate over 3 bytes

Shift Rows

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

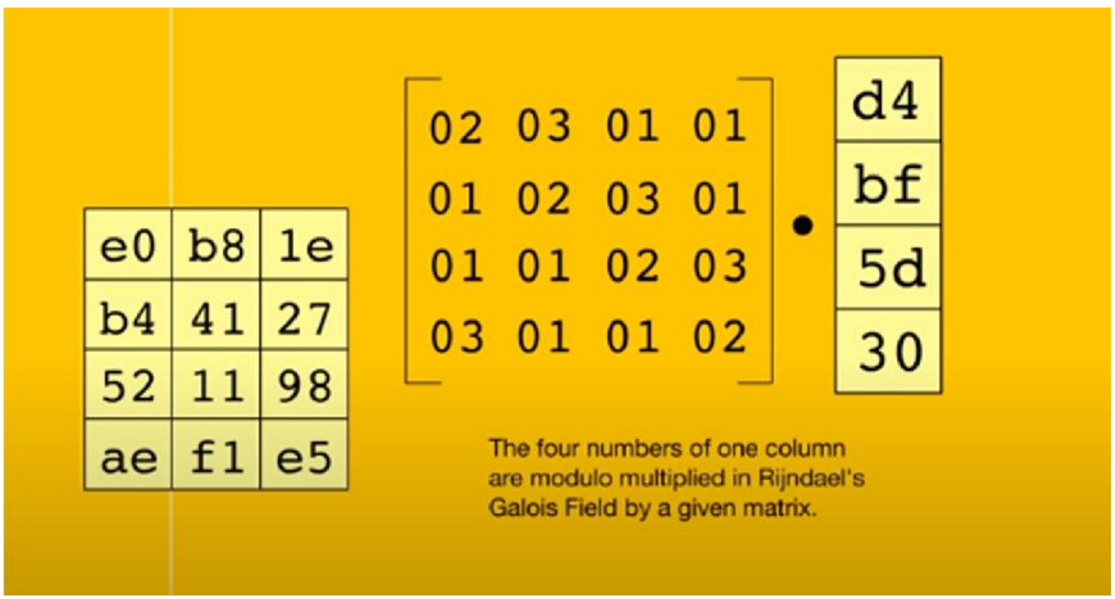
the output is the new state:

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	$\longleftarrow$ one position left shift
B_{10}	B_{14}	B_2	B_6	$\longleftarrow$ two positions left shift
B_{15}	B_3	B_7	B_{11}	$\longleftarrow$ three positions left shift

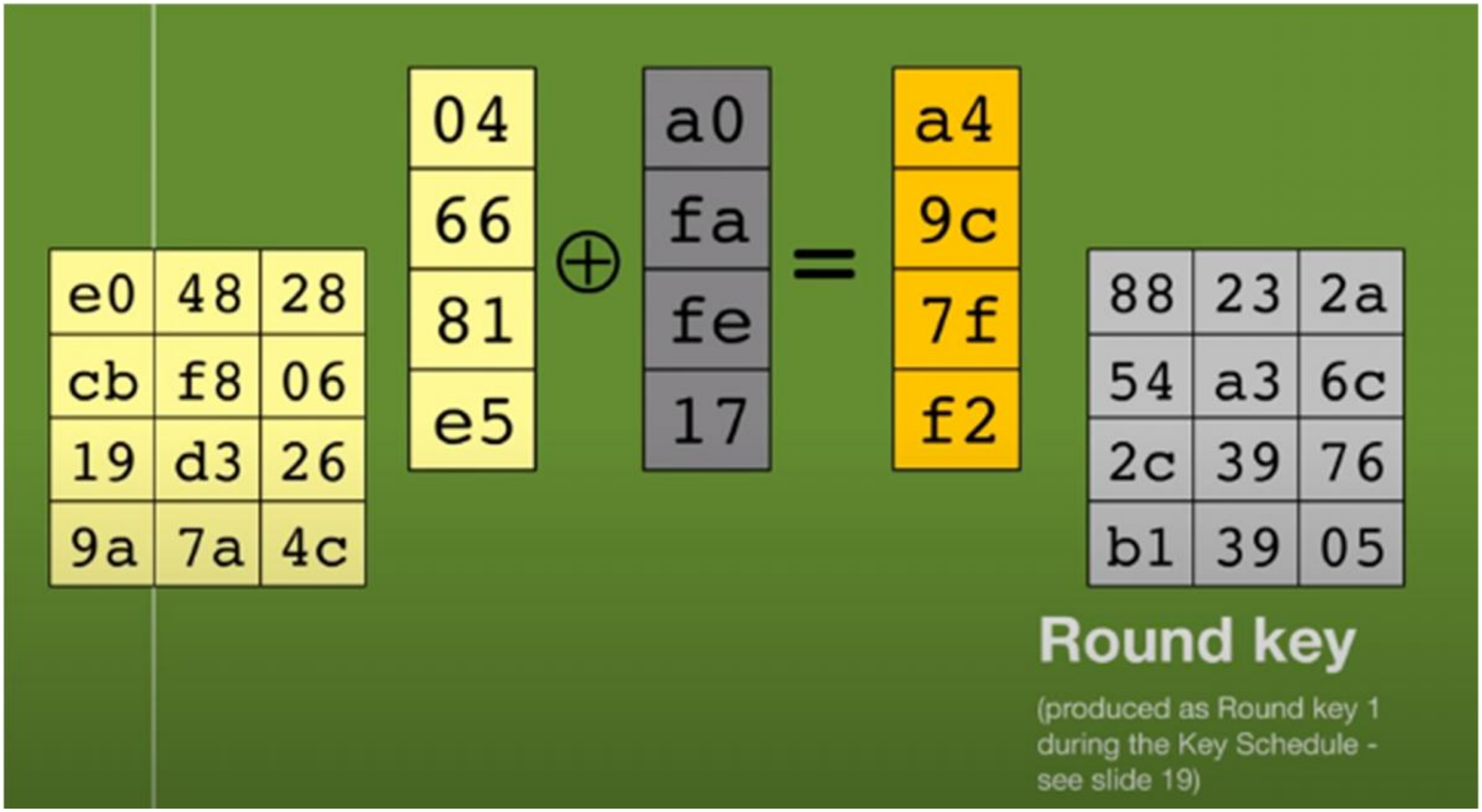
Mix Columns

$$\mathrm{MixColumn}(B) = C,$$

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}.$$



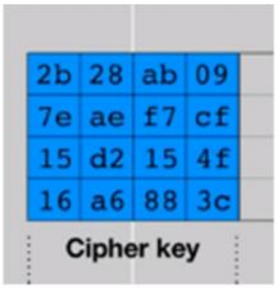
Add Round Key



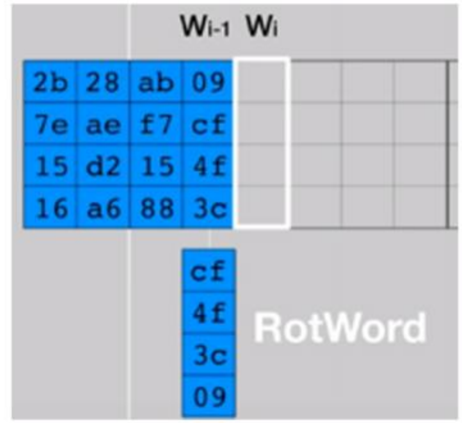
B

Key Schedule

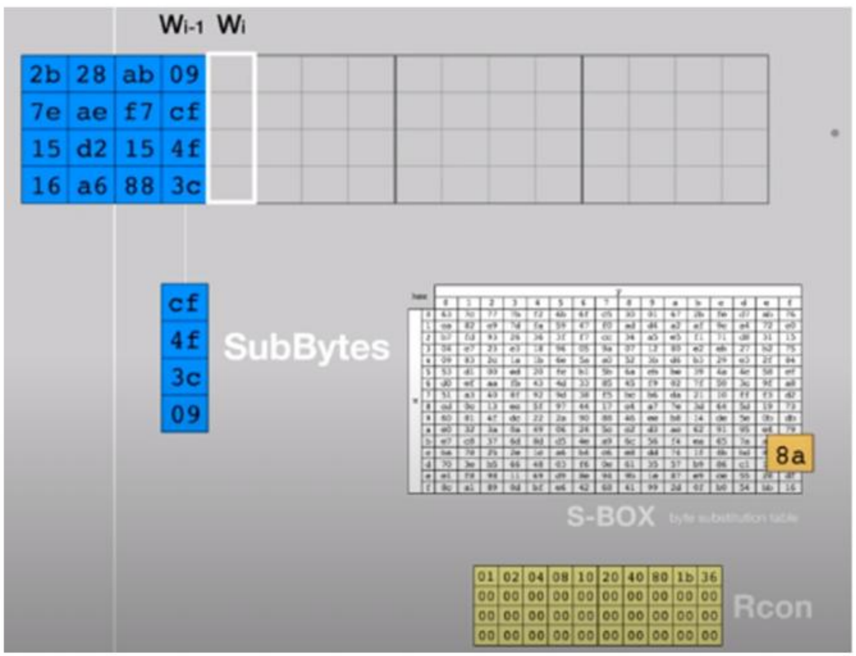
(Expansion of the given Cipher key into 11 partial keys, used in the initial round, the 9 main rounds and the final round)



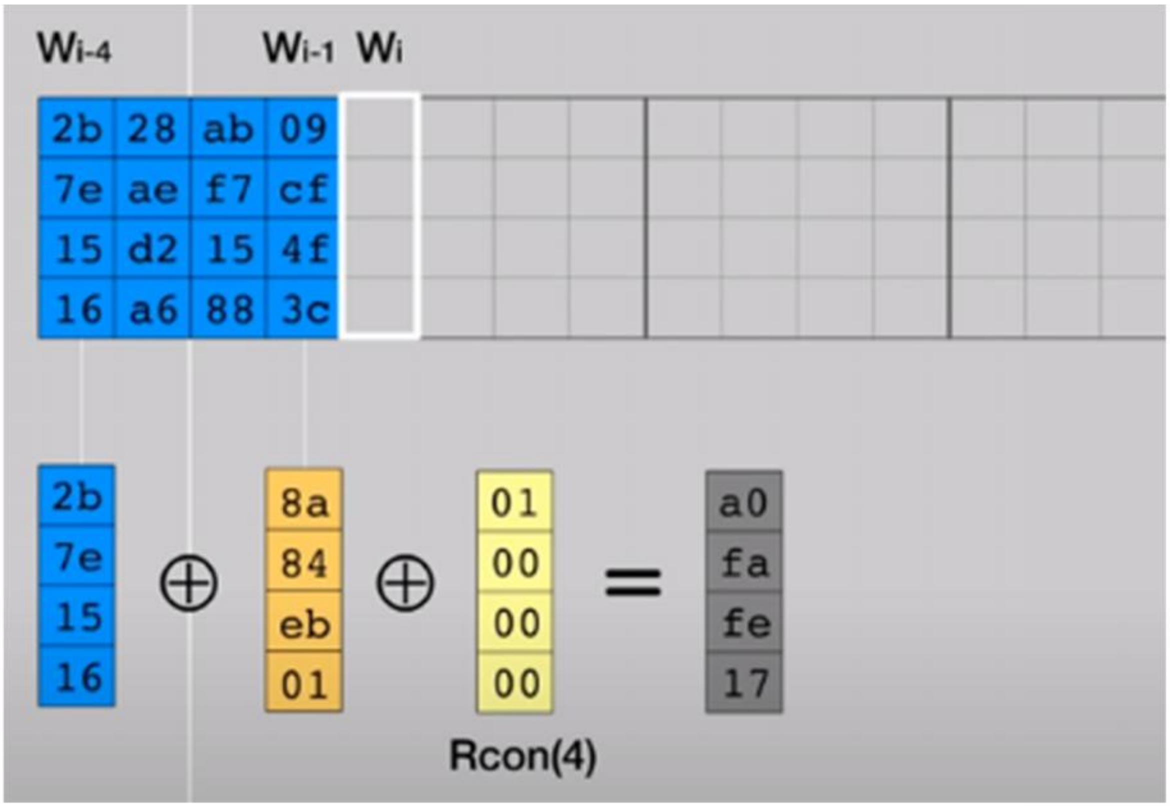
2b 28 ab 09
7e ae f7 cf
15 d2 15 4f
16 a6 88 3c
Cipher key



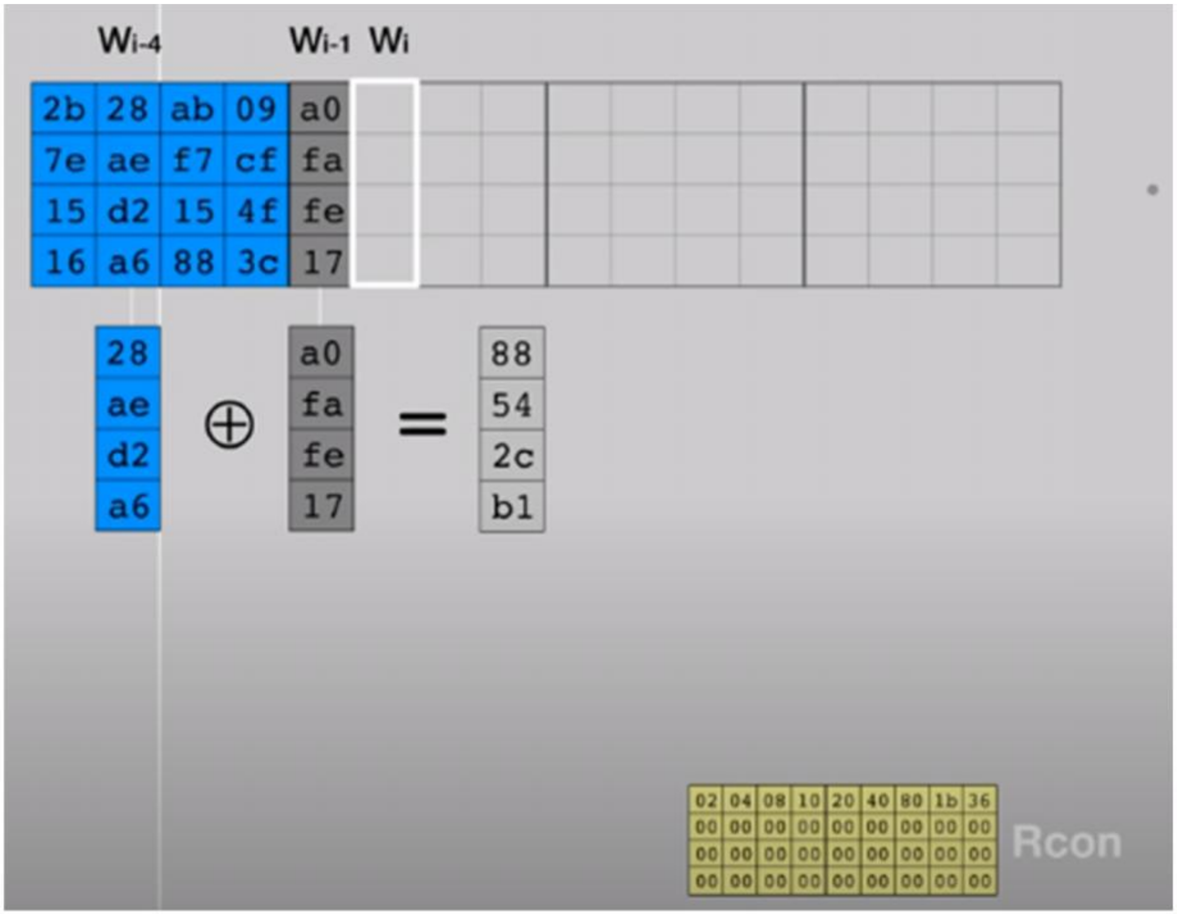
Wi-1 Wi
2b 28 ab 09
7e ae f7 cf
15 d2 15 4f
16 a6 88 3c
cf
4f
3c
09
RotWord



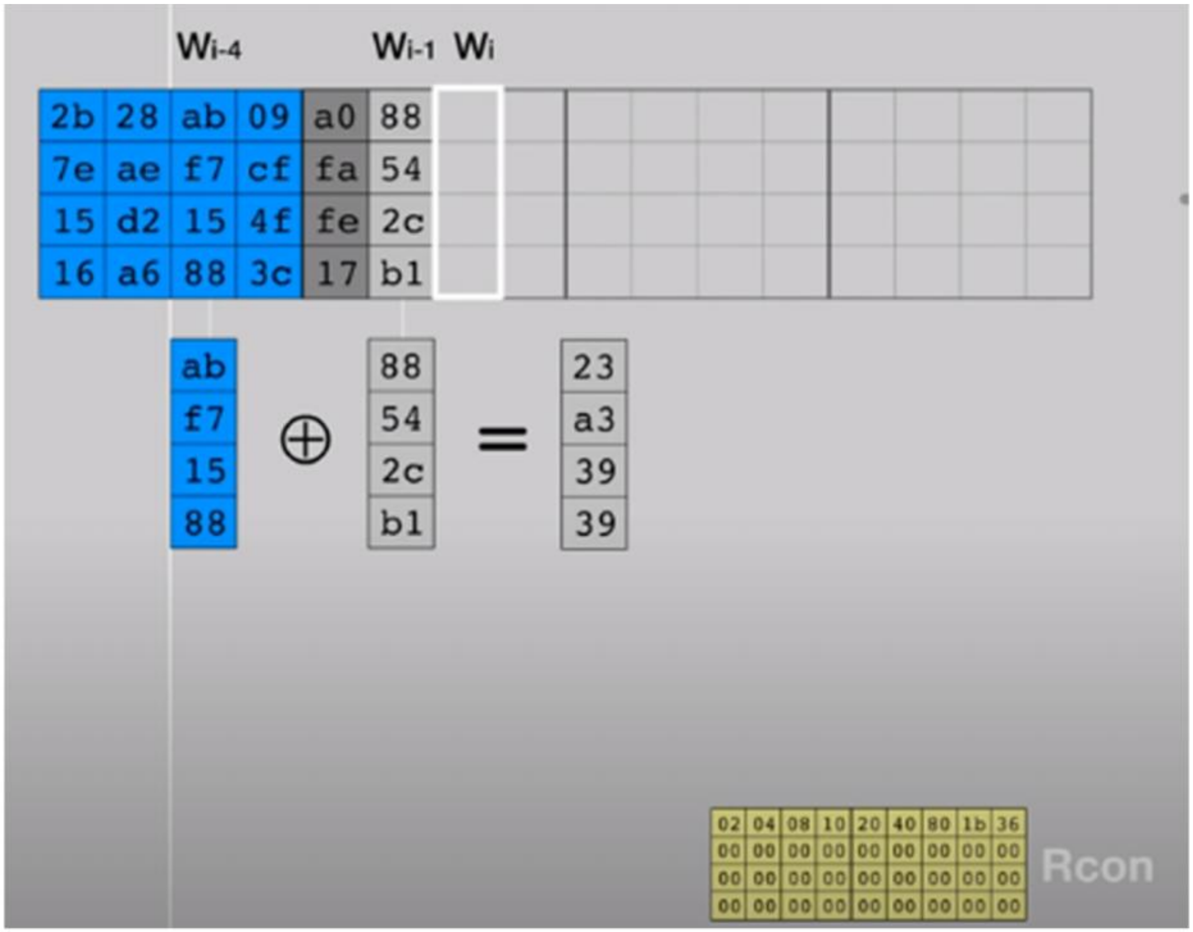
Wi-1 Wi
2b 28 ab 09
7e ae f7 cf
15 d2 15 4f
16 a6 88 3c
cf
4f
3c
09
SubBytes
8a
S-BOX
01 02 04 08 10 20 40 80 1b 36
00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00
Rcon

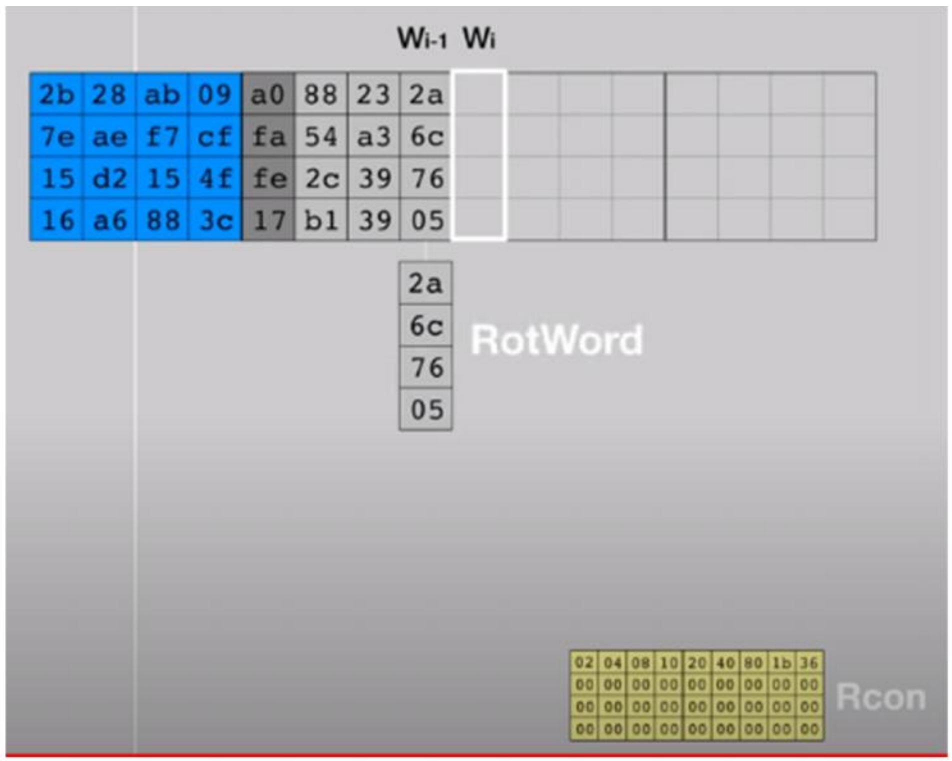


Wi-4
Wi-1 Wi
2b 28 ab 09
7e ae f7 cf
15 d2 15 4f
16 a6 88 3c
2b 7e 15 16
8a 84 eb 01
01 00 00 00
a0 fa fe 17
Rcon(4)



Wi-4
Wi-1
Wi
2b 28 ab 09 a0
7e ae f7 cf fa
15 d2 15 4f fe
16 a6 88 3c 17
28
ae
d2
a6
⊕
a0
fa
fe
17
=
88
54
2c
b1
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
Rcon


W_{i-4}
W_{i-1} W_i
2b 28 ab 09 a0 88
7e ae f7 cf fa 54
15 d2 15 4f fe 2c
16 a6 88 3c 17 b1
ab f7 15 88 ⊕ 88 54 2c b1 = 23 a3 39 39
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
Rcon


Wi-1 Wi
2b 28 ab 09 a0 88 23 2a
7e ae f7 cf fa 54 a3 6c
15 d2 15 4f fe 2c 39 76
16 a6 88 3c 17 b1 39 05
2a
6c
76
05
RotWord
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
Rcon

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Cipher key

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

Round key 1

f2	7a	59	73
c2	96	35	59
95	b9	80	f6
f2	43	7a	7f

Round key 2

3d	47	1e	6d
80	16	23	7a
47	fe	7e	88
7d	3e	44	3b

Round key 3

. . .

d0	c9	e1	b6
14	ee	3f	63
f9	25	0c	0c
a8	89	c8	a6

Round key 10

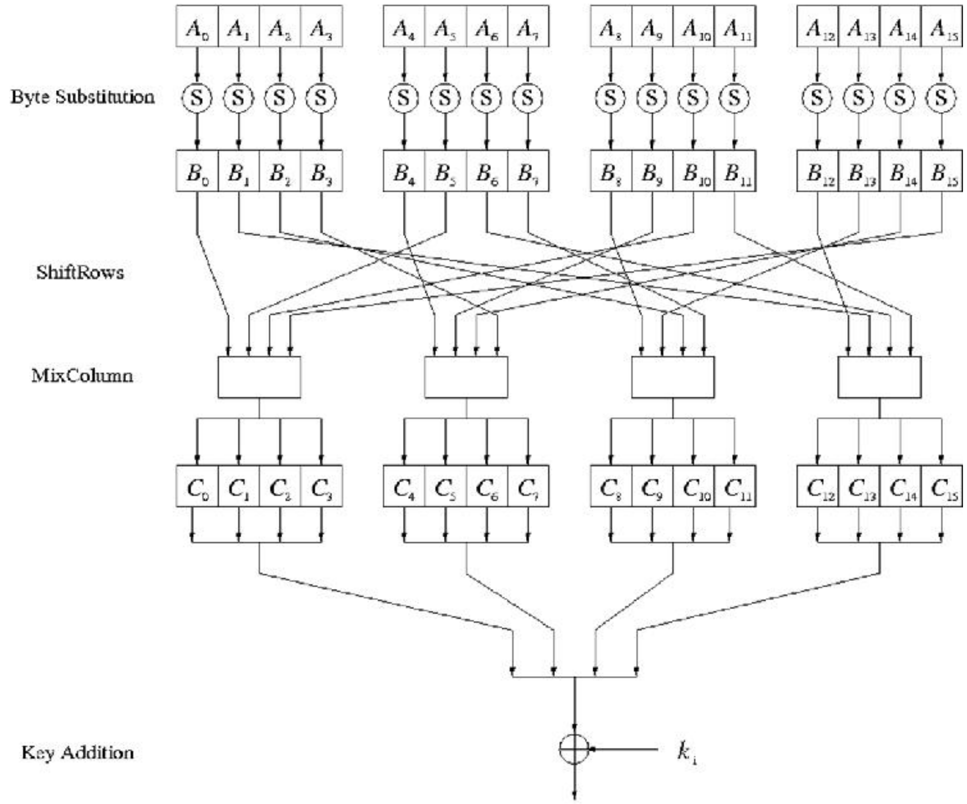
Internals

- AES is a byte-oriented cipher
- The state A (i.e., the 128-bit data path) can be arranged in a 4x4 matrix:

A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

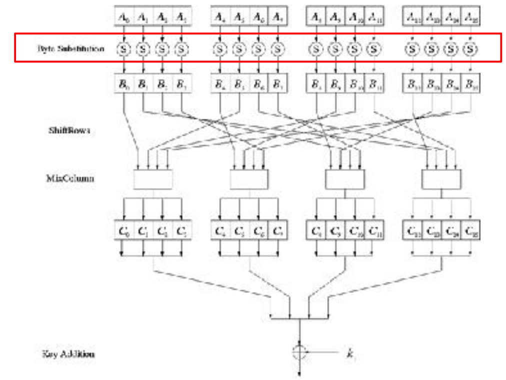
Internal Structure of AES

- Round function for rounds 1,2,…,n_{r-1}:



- Note: In the last round, the MixColumn tansformation is omitted

Byte Substitution Layer



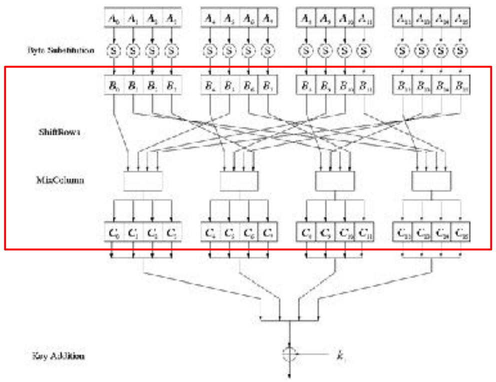
- The Byte Substitution layer consists of 16 **S-Boxes** with the following properties:

  The S-Boxes are

  - **identical**
  - the only **nonlinear** elements of AES, i.e.,
    ByteSub(A_i) + ByteSub(A_j) ≠ ByteSub(A_i + A_j), for i,j = 0,…,15
  - **bijective**, i.e., there exists a one-to-one mapping of input and output bytes
    $\Rightarrow$ S-Box can be uniquely reversed

- In software implementations, the S-Box is usually realized as a lookup table

Diffusion Layer



The Diffusion layer

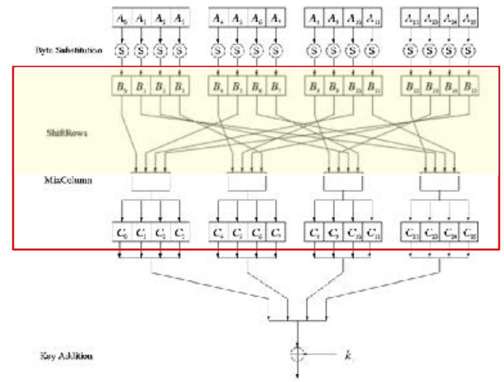
- provides diffusion over all input state bits

- consists of two sublayers:
  - **ShiftRows Sublayer**: Permutation of the data on a byte level
  - **MixColumn Sublayer**: Matrix operation which combines ("mixes") blocks of four bytes

- performs a linear operation on state matrices *A*, *B*, i.e.,

$$\mathrm{DIFF}(A) + \mathrm{DIFF}(B) = \mathrm{DIFF}(A + B)$$

ShiftRows Sublayer



- Rows of the state matrix are shifted cyclically:

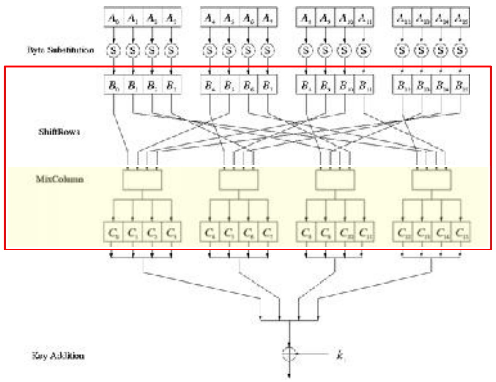
Input matrix

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	← one position left shift
B_{10}	B_{14}	B_2	B_6	← two positions left shift
B_{15}	B_3	B_7	B_{11}	← three positions left shift

MixColumn Sublayer



- Linear transformation which mixes each column of the state matrix

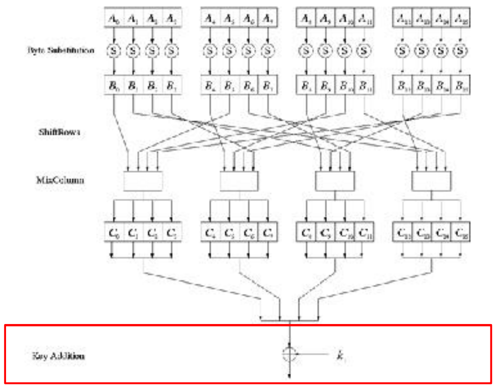
- Each 4-byte column is considered as a vector and multiplied by a fixed 4x4 matrix, e.g.,

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

where 01, 02 and 03 are given in hexadecimal notation

- All arithmetic is done in the Galois field $GF(2^8)$ (for more information see Chapter 4.3 in *Understanding Cryptography*)

Key Addition Layer



- Inputs:
  - 16-byte state matrix C
  - 16-byte subkey k_i
- Output: $C \oplus k_i$
- The subkeys are generated in the key schedule

Key Schedule

- Subkeys are derived recursively from the original 128/192/256-bit input key

- Each round has 1 subkey, plus 1 subkey at the beginning of AES

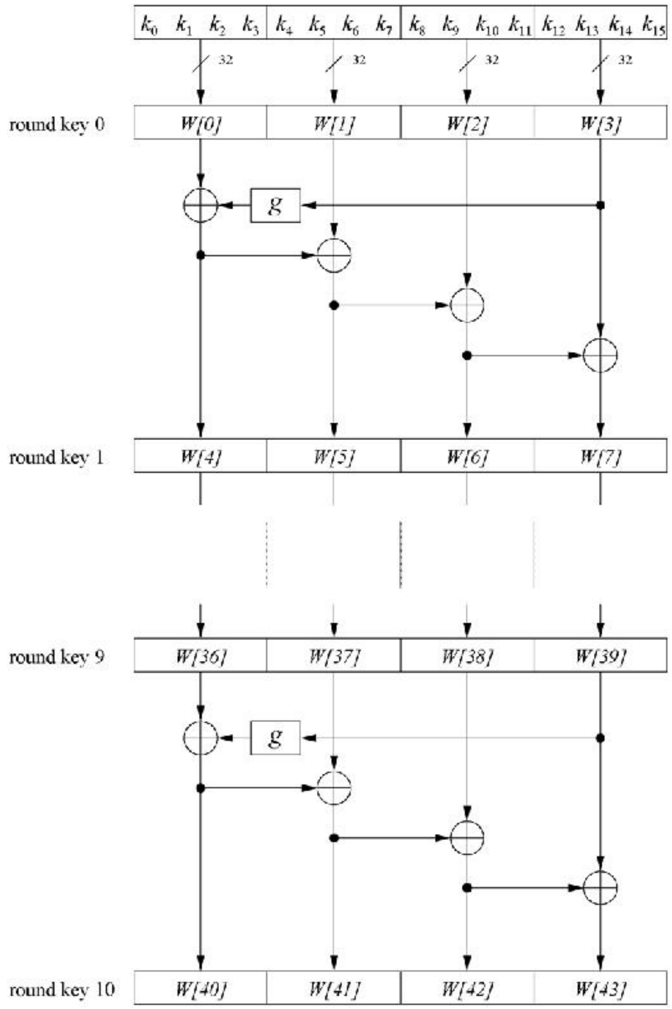
Key length (bits)	Number of subkeys
128	11
192	13
256	15

- Key whitening: Subkey is used both at the input and output of AES
  $\Rightarrow$ # subkeys = # rounds + 1

- There are different key schedules for the different key sizes

Key Schedule

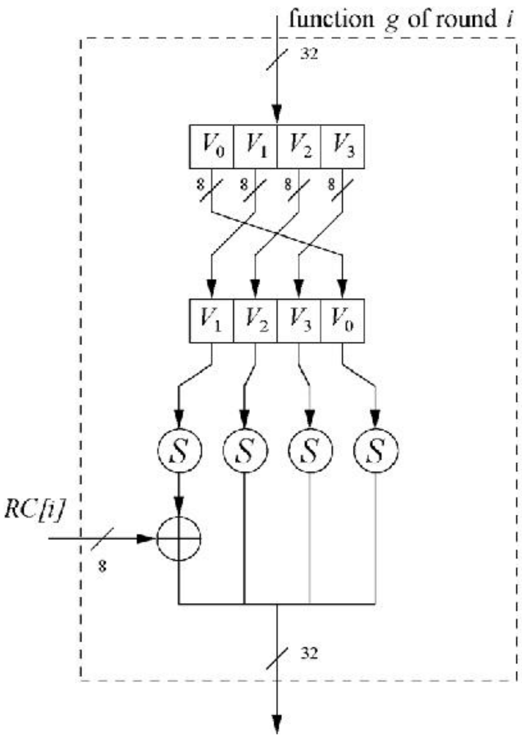
Example: Key schedule for 128-bit key AES



- Word-oriented: 1 word = 32 bits
- 11 subkeys are stored in *W[0]…W[3], W[4]…W[7], … , W[40]…W[43]*
- First subkey *W[0]…W[3]* is the original AES key

Key Schedule

- Function g rotates its four input bytes and performs a bytewise S-Box substitution $\Rightarrow$ nonlinearity



- The round coefficient RC is only added to the leftmost byte and varies from round to round:

$$RC[1] = x^0 = (00000001)_2$$
$$RC[2] = x^1 = (00000010)_2$$
$$RC[3] = x^2 = (00000100)_2$$
$$\ldots$$
$$RC[10] = x^9 = (00110110)_2$$

- x^i represents an element in a Galois field (again, cf. Chapter 4.3 of *Understanding Cryptography*)