Cryptography and Network Security (BTIT13502)

Module 05 : **Advanced Encryption Standard**

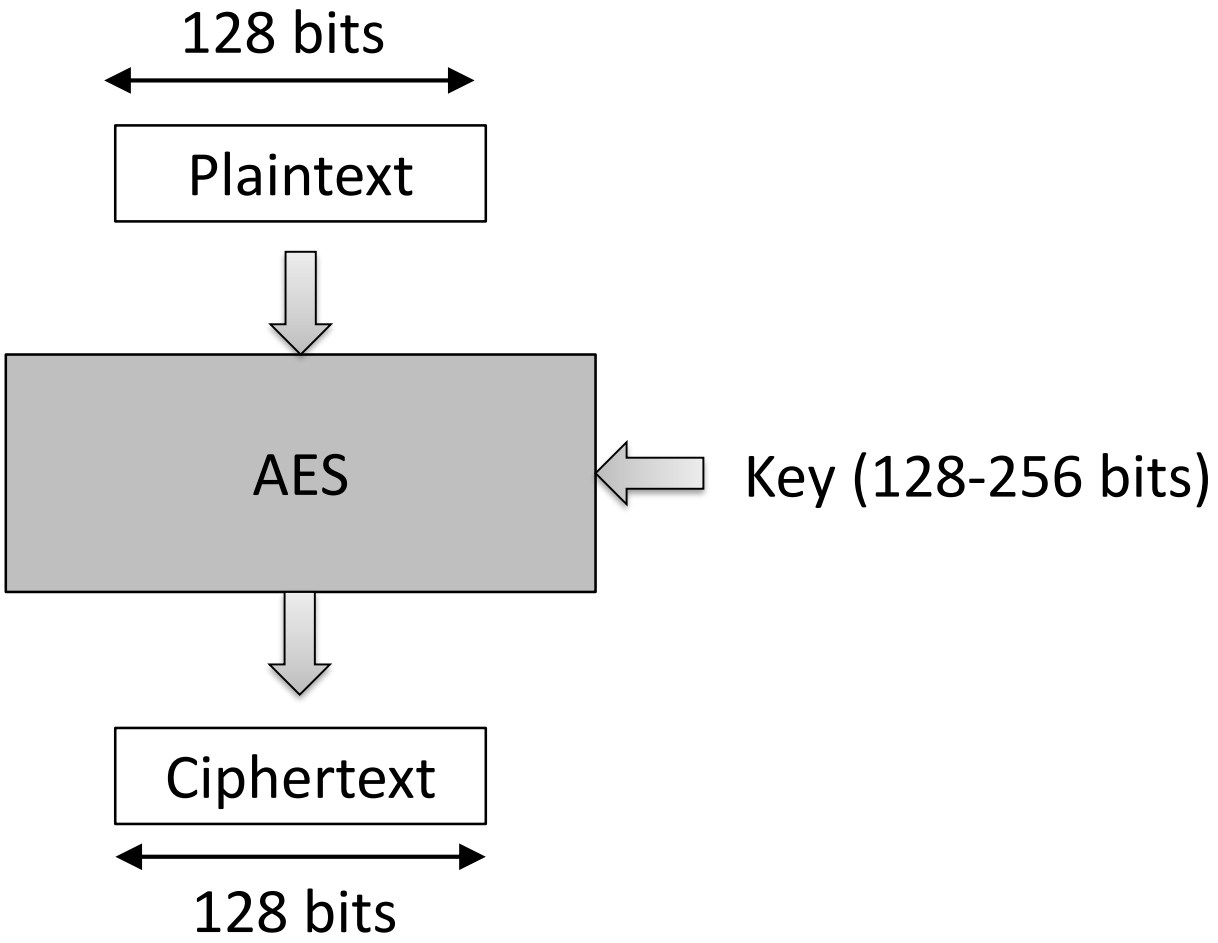
AES (Advanced Encryption Standard)

- The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits.

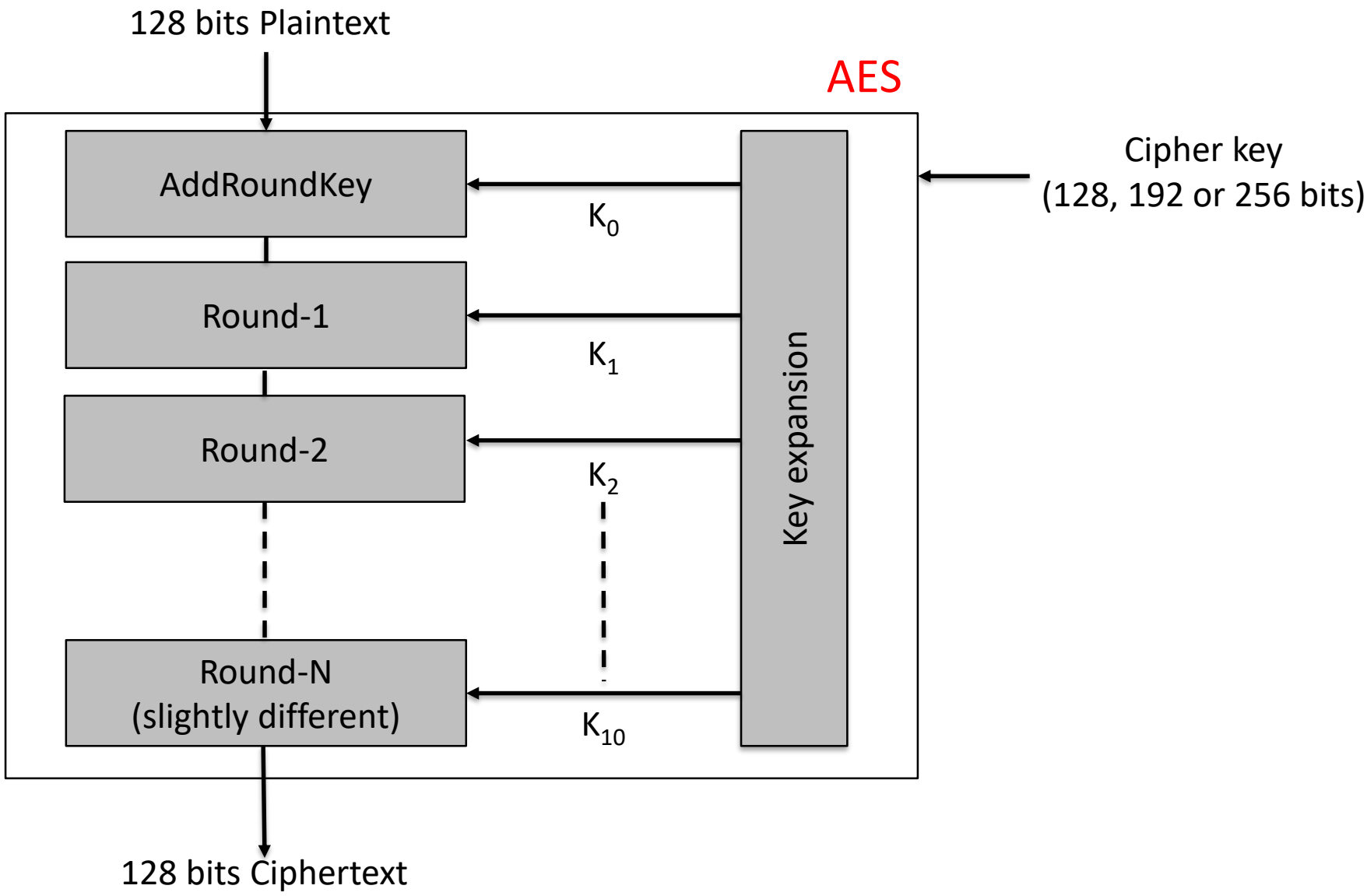
Key size (words/ bytes/ bits)	4/16/128	6/24/192	8/32/256
Block size (words/ bytes/ bits)	4/16/128	4/16/128	4/16/128
Round key size (words/ bytes/ bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14

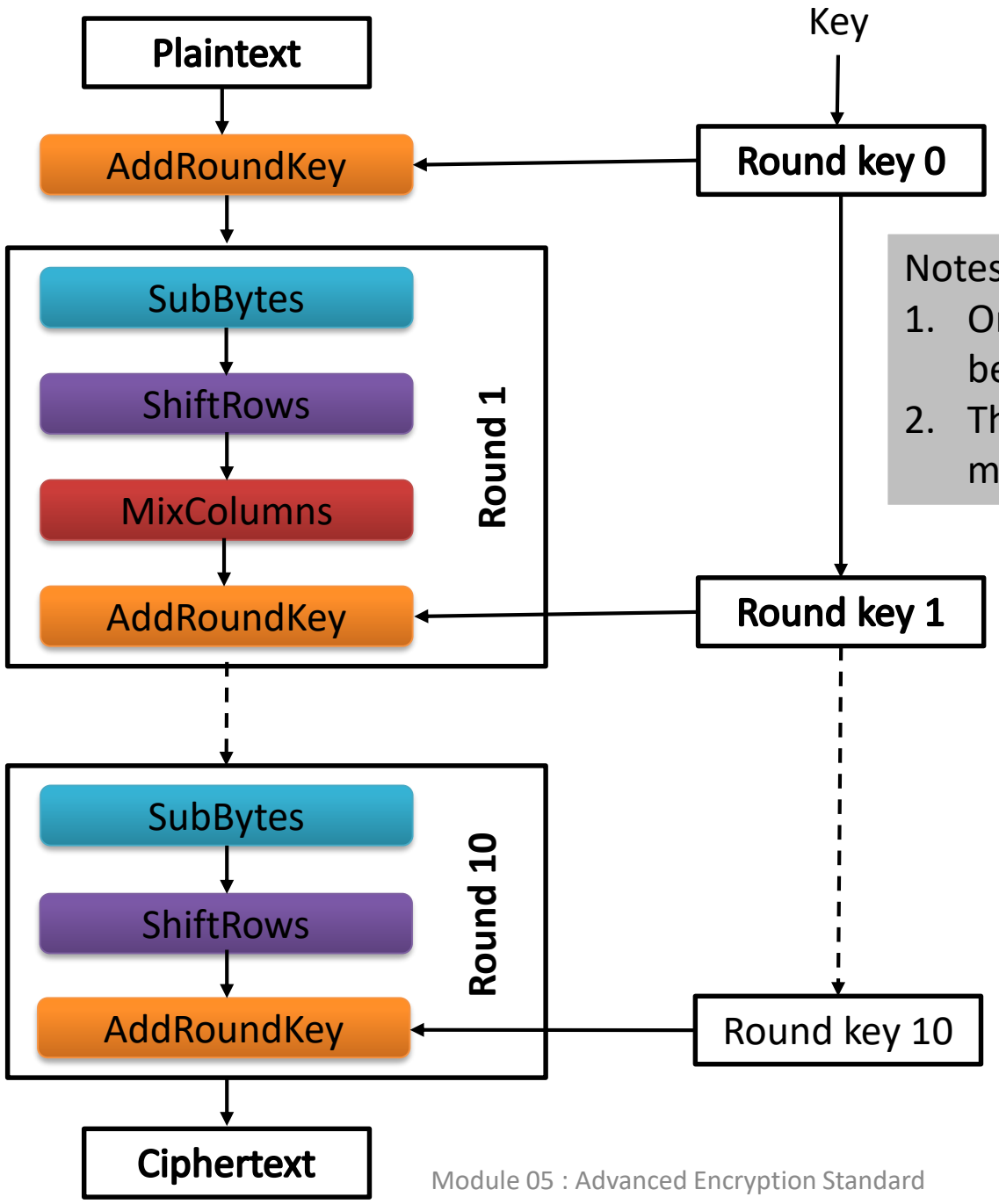
- AES designed to have characteristics
  1. Resistance against all known attacks
  2. Speed and code compactness on a wide range of platforms
  3. Design simplicity

AES (Advanced Encryption Standard)



AES (Advanced Encryption Standard)

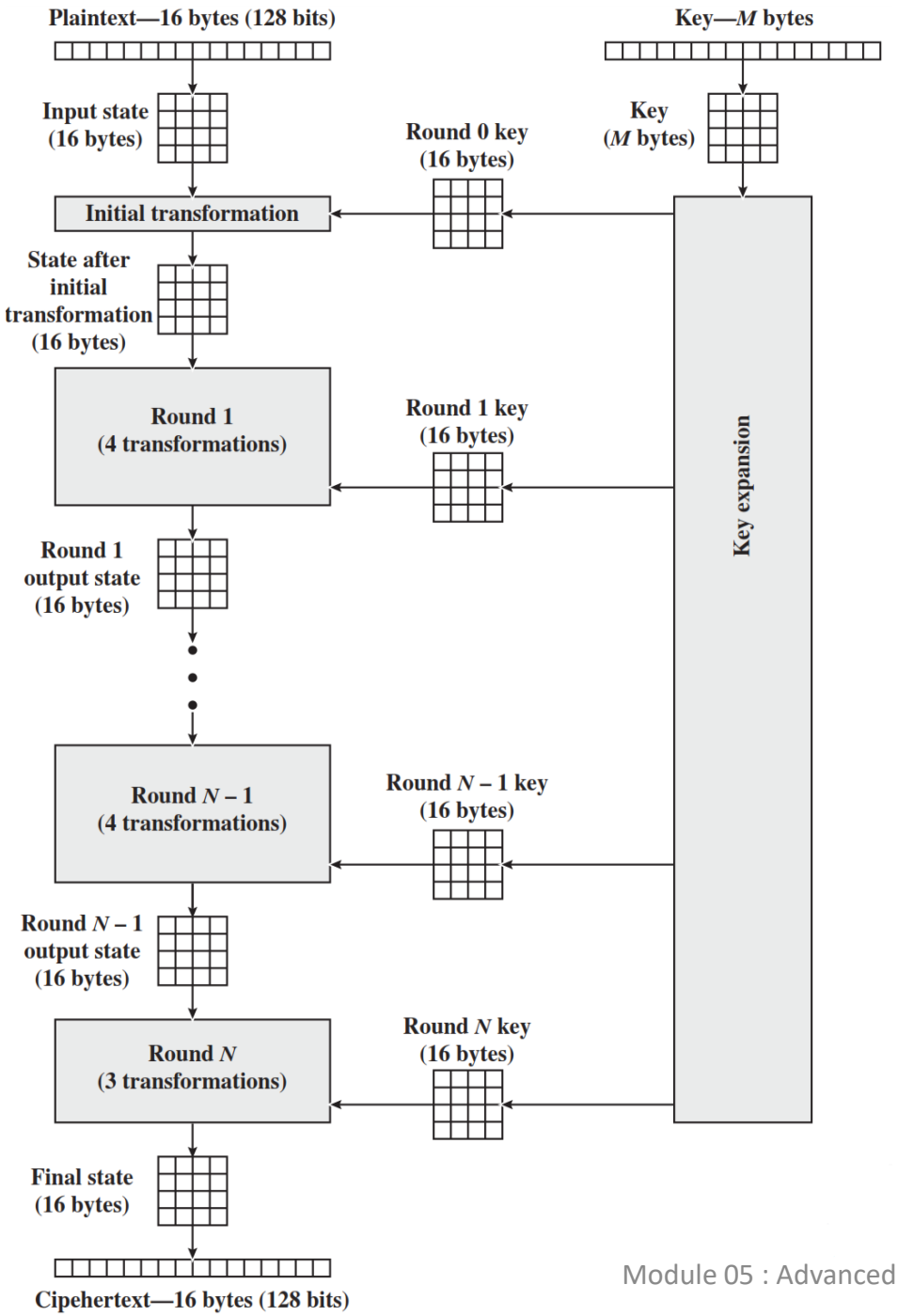




Notes:

1. One AddRoundKey is applied before the First round.
2. The third transformation is missing in the last round

AES Structure



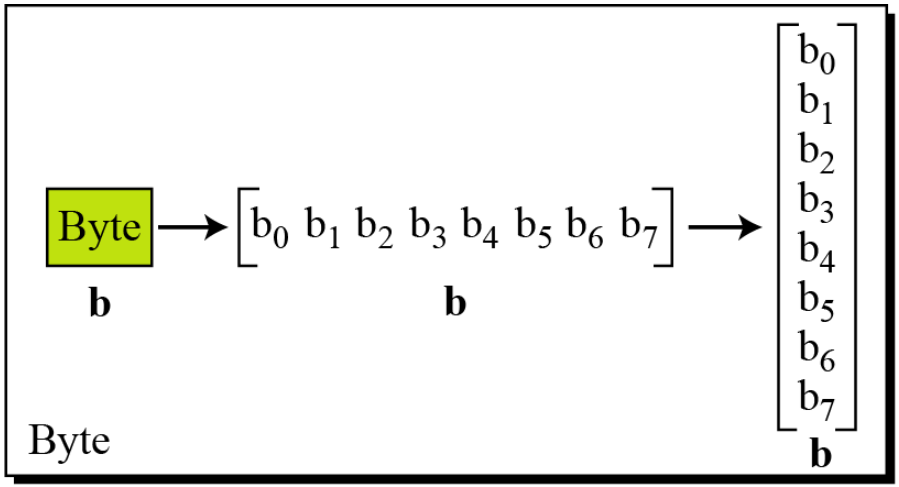
Initialization

1. Expand 16-byte key to get the actual **key block** to be used.
2. Initialize 16-byte plaintext block called as **state**.
3. XOR the **state** with the **key block**.

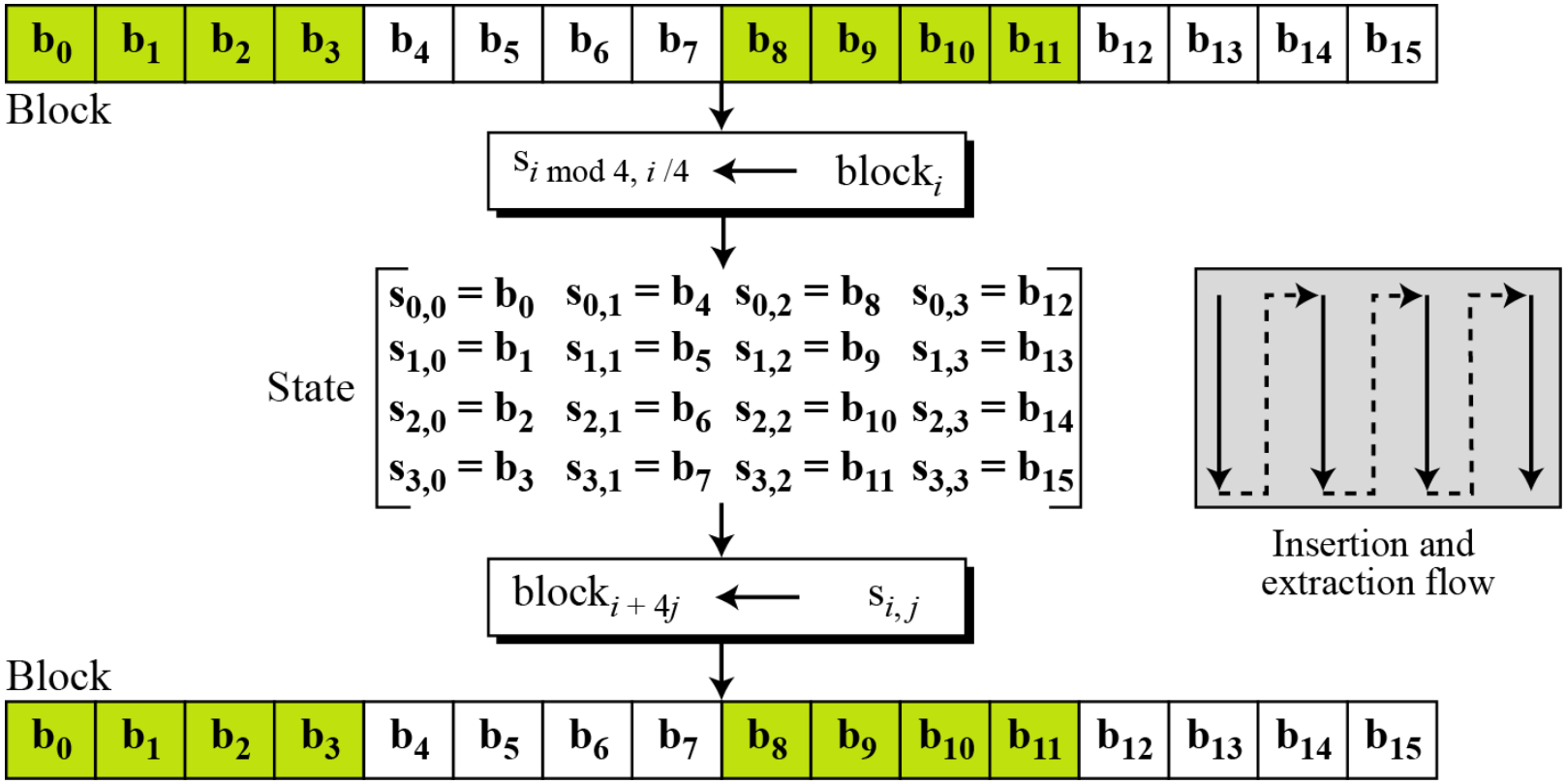
For each round

1. Apply S-box
2. Rotate rows of state
3. Mix columns
4. Add Round key: XOR the state with key block.

Data Units in AES



Block to State & State to Block



Plain Text to State

Hexadecimal	00	04	12	14	12	04	12	00	0C	00	13	11	08	23	19	19

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$$ State

AES Structure

- The first N-1 rounds consist of four distinct transformation functions.

SubBytes	• The 16 input bytes are substituted using an **S-box**
ShiftRows	• Each of the four rows of the matrix is shifted to the left
MixColumns	• Each column of four bytes is now transformed using a special mathematical function.
AddRoundKey	• The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key.

AES structure

State:

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

Cipher key:

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

Initial transformation(AddRoundKey)

AddRoundKey: input state ⊕ Cipher key

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

⊕

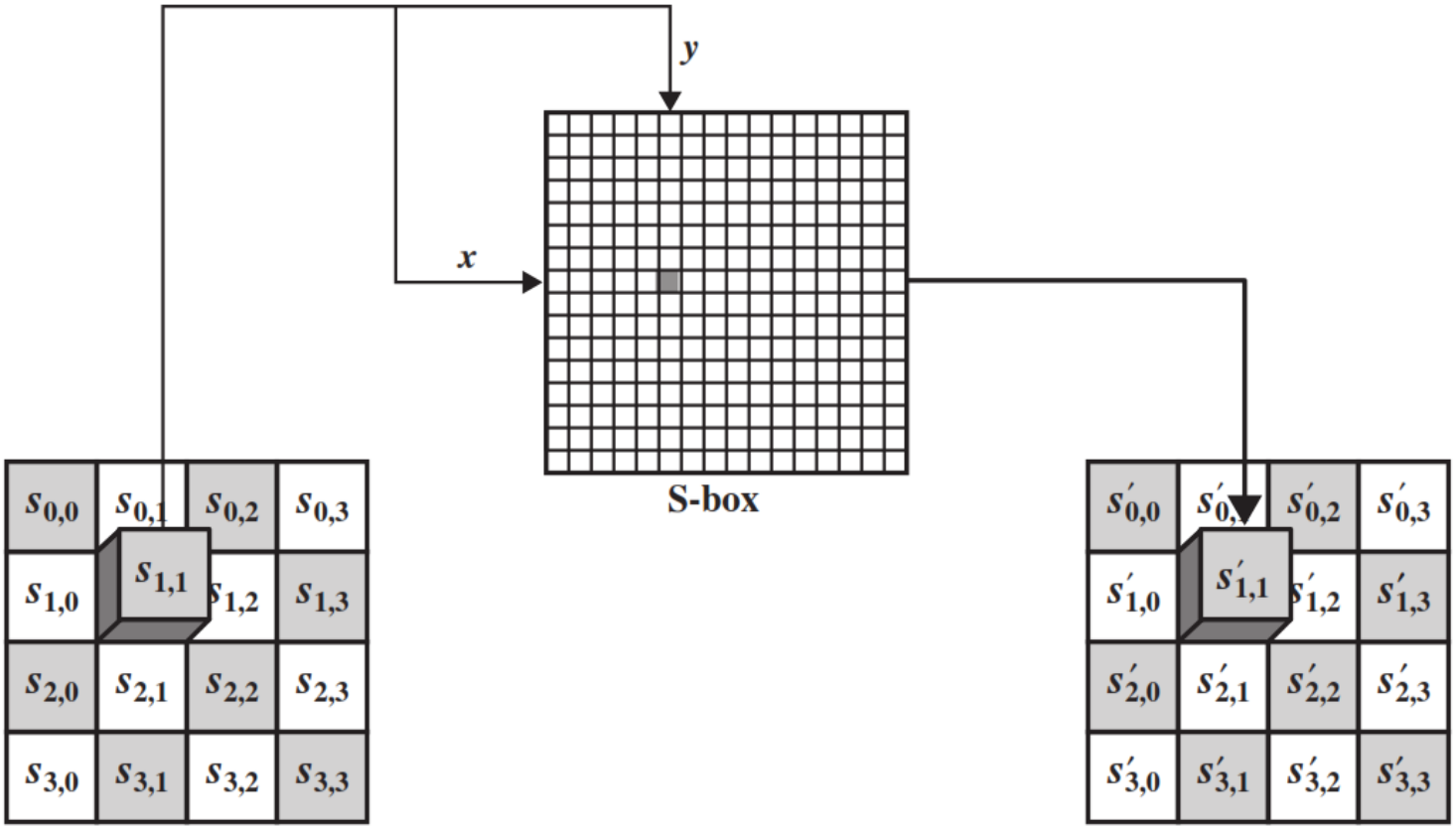
2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

=

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

SubByte Transformation

- The forward substitute byte transformation, called **SubBytes**, is a simple table lookup



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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

Input: 19

Output: D4

Row Column

SubByte output

Input for SubByte

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

Output of SubByte

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

ShiftRows

- The first row of State is not altered.
- For the second row, a 1-byte circular left shift is performed.
- For the third row, a 2-byte circular left shift is performed.
- For the fourth row, a 3-byte circular left shift is performed.

d4	e0	b8	le	← No rotation
27	bf	b4	41	← Rotate 1 byte
11	98	5d	52	← Rotate 2 bytes
ae	f1	e5	30	← Rotate 3 bytes

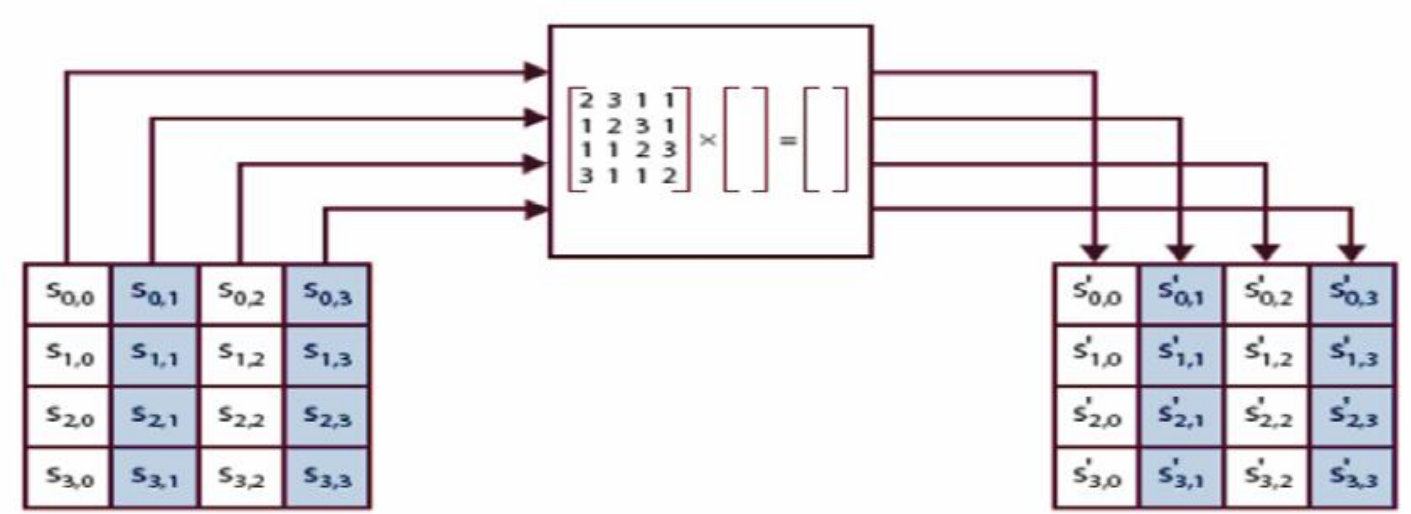
Input for ShiftRows

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

Output of ShiftRows

MixColumns

- Each byte of a column is mapped into a new value that is a function of all four bytes in that column.
- Constant matrices used by MixColumns.



$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

MixColumns

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

$$\bullet \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} =$$

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \bullet$$

d4
bf
5d
30

$$=$$

04
66
81
e5

AddRoundKey

- In the forward add round key transformation, the 128 bits of State are bitwise XORed with the 128 bits of the round key.

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

$\oplus$

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

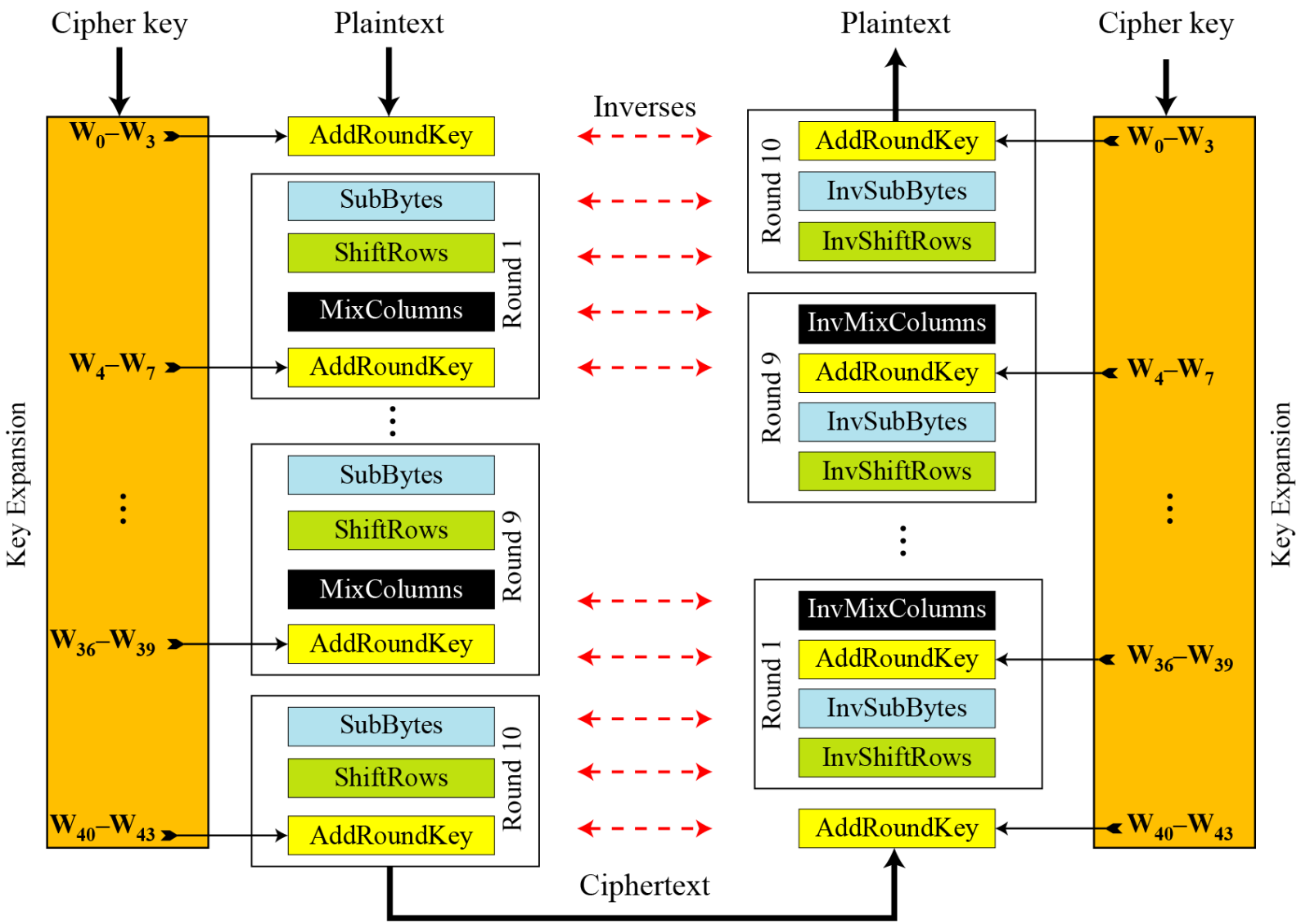
=

A4	68	6b	02
9c	9f	5b	6a
7f	35	Ea	50
F2	2b	43	49

04		a0	A4
66	$\oplus$	fa	9c
81		fe	7f
e5		17	F2

=

AES Overall Structure

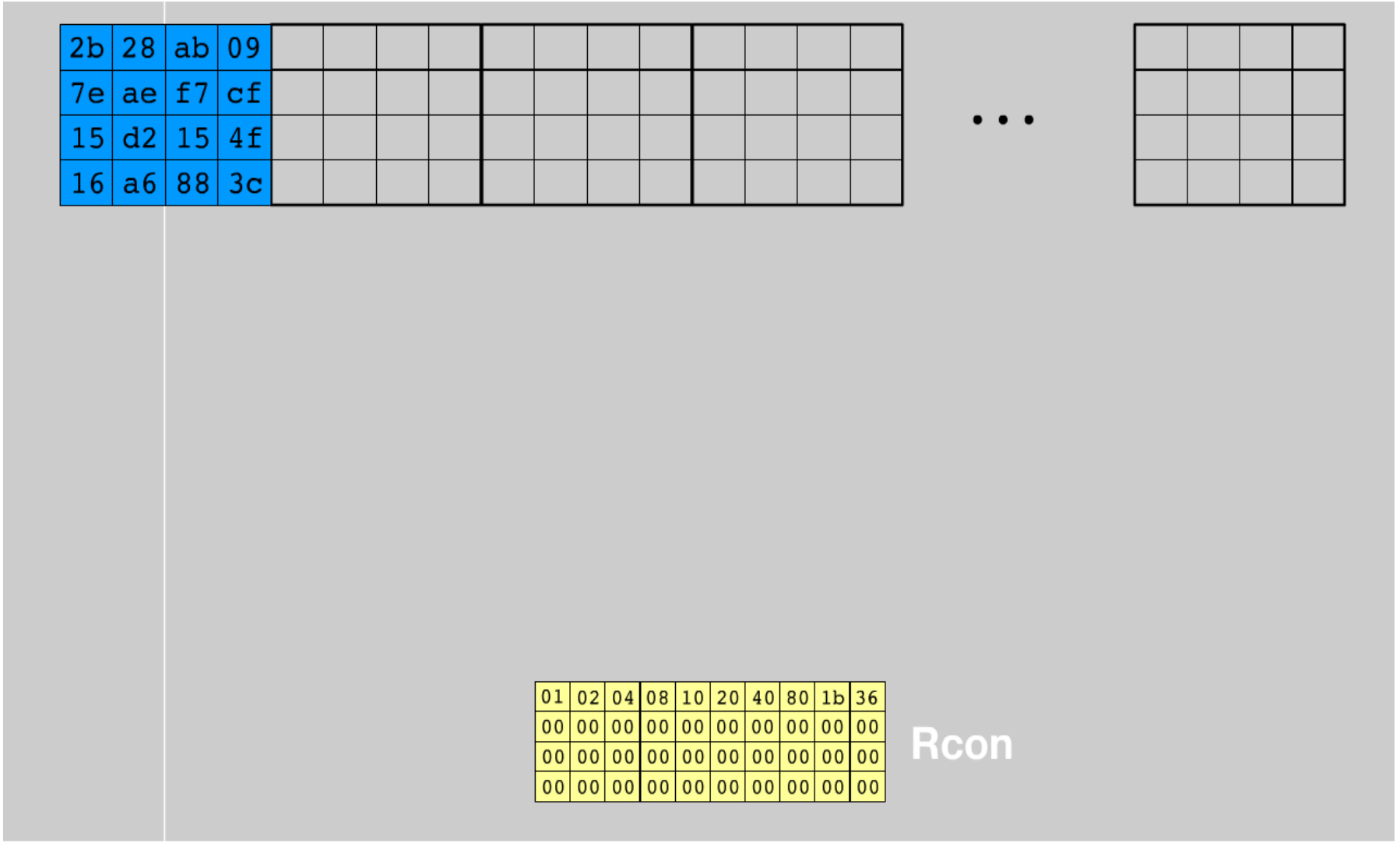


AES key expansion

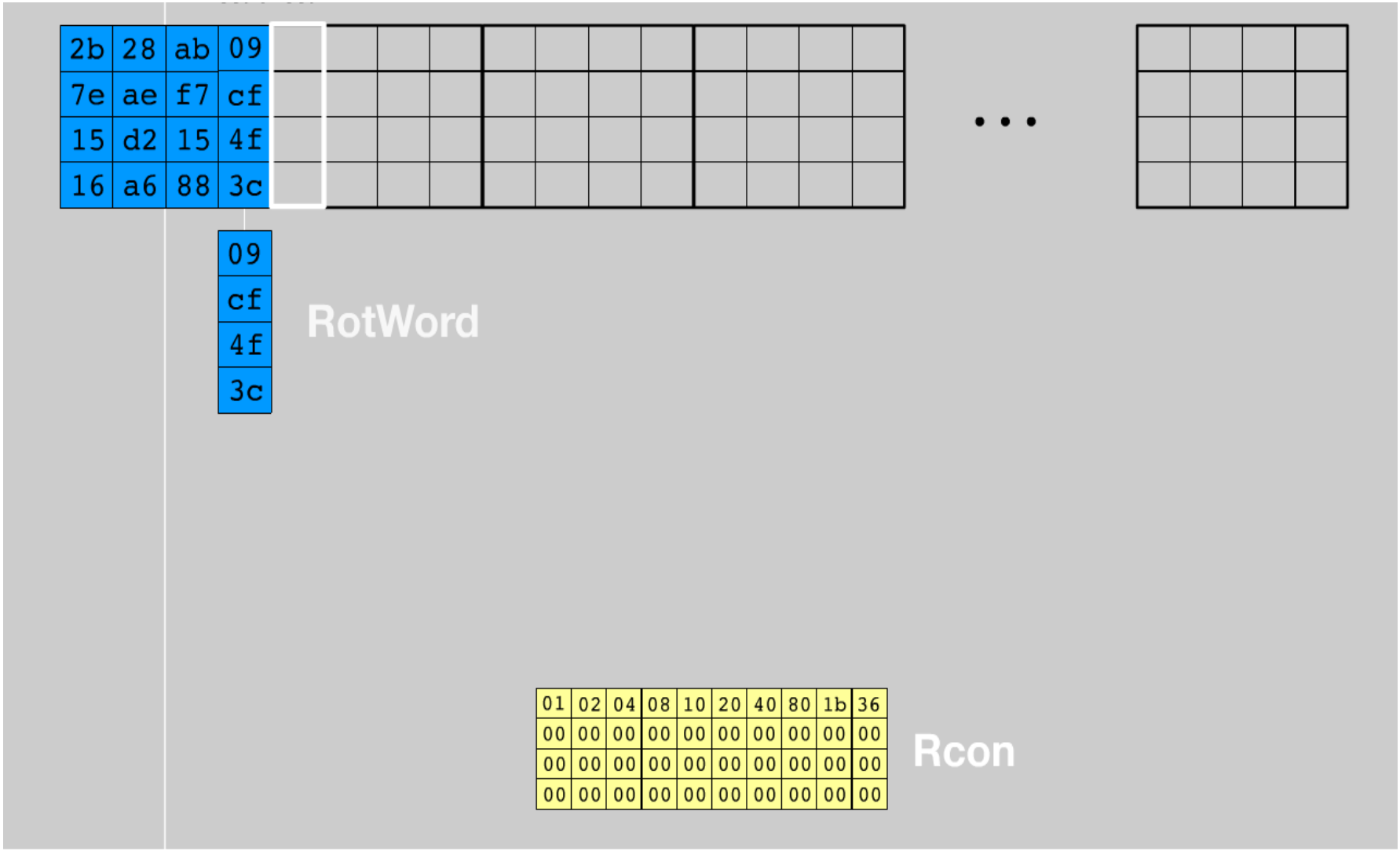
Words for each round

Round	Words			
Pre-round	W_0	W_1	W_2	W_3
Round 1	W_4	W_5	W_6	W_7
Round 2	W_8	W_9	W_{10}	W_{11}
. . .		. . .		
Round N	W_{40}	W_{41}	W_{42}	W_{43}

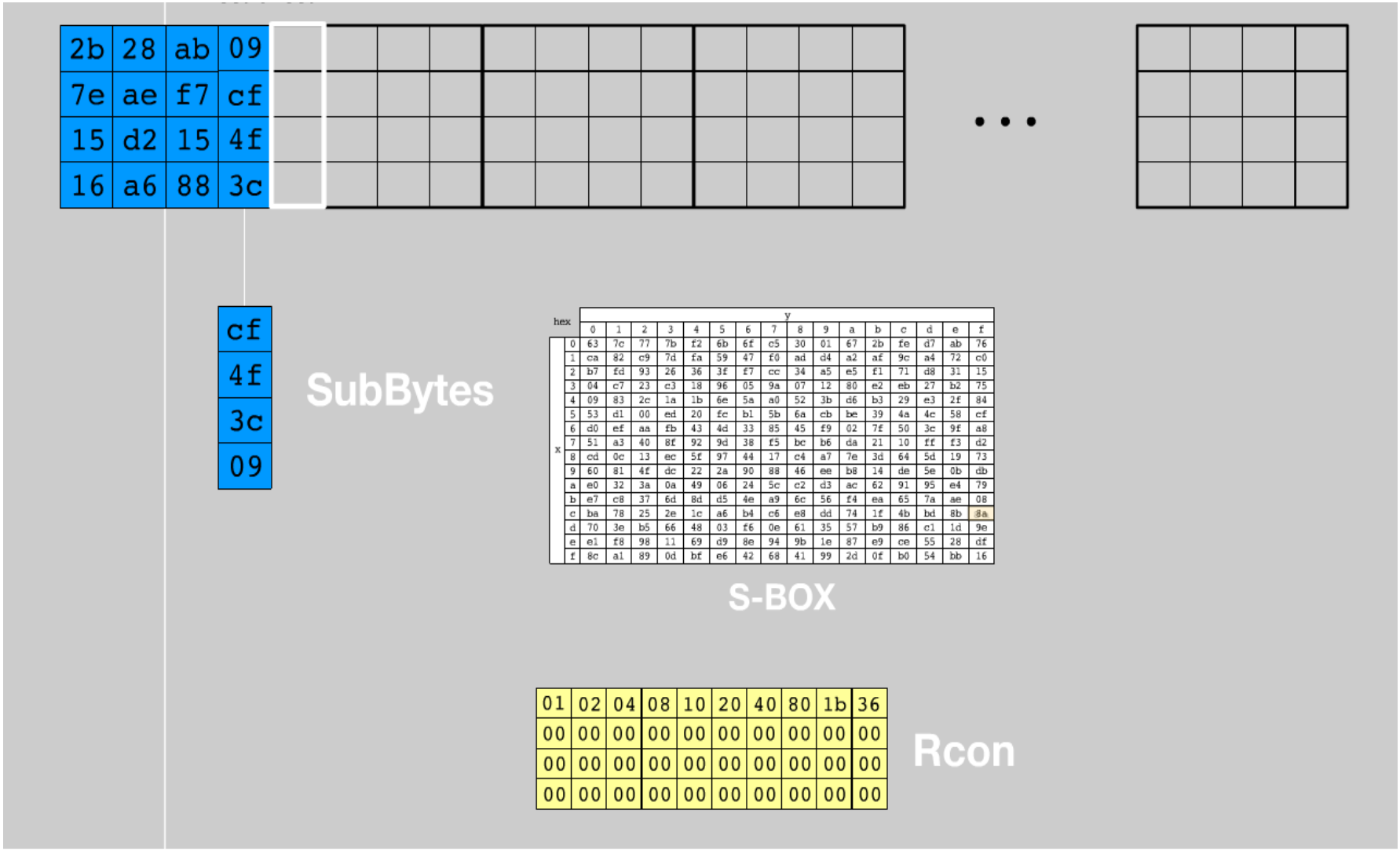
AES Key schedule generation (contd)



AES Key schedule generation (contd)

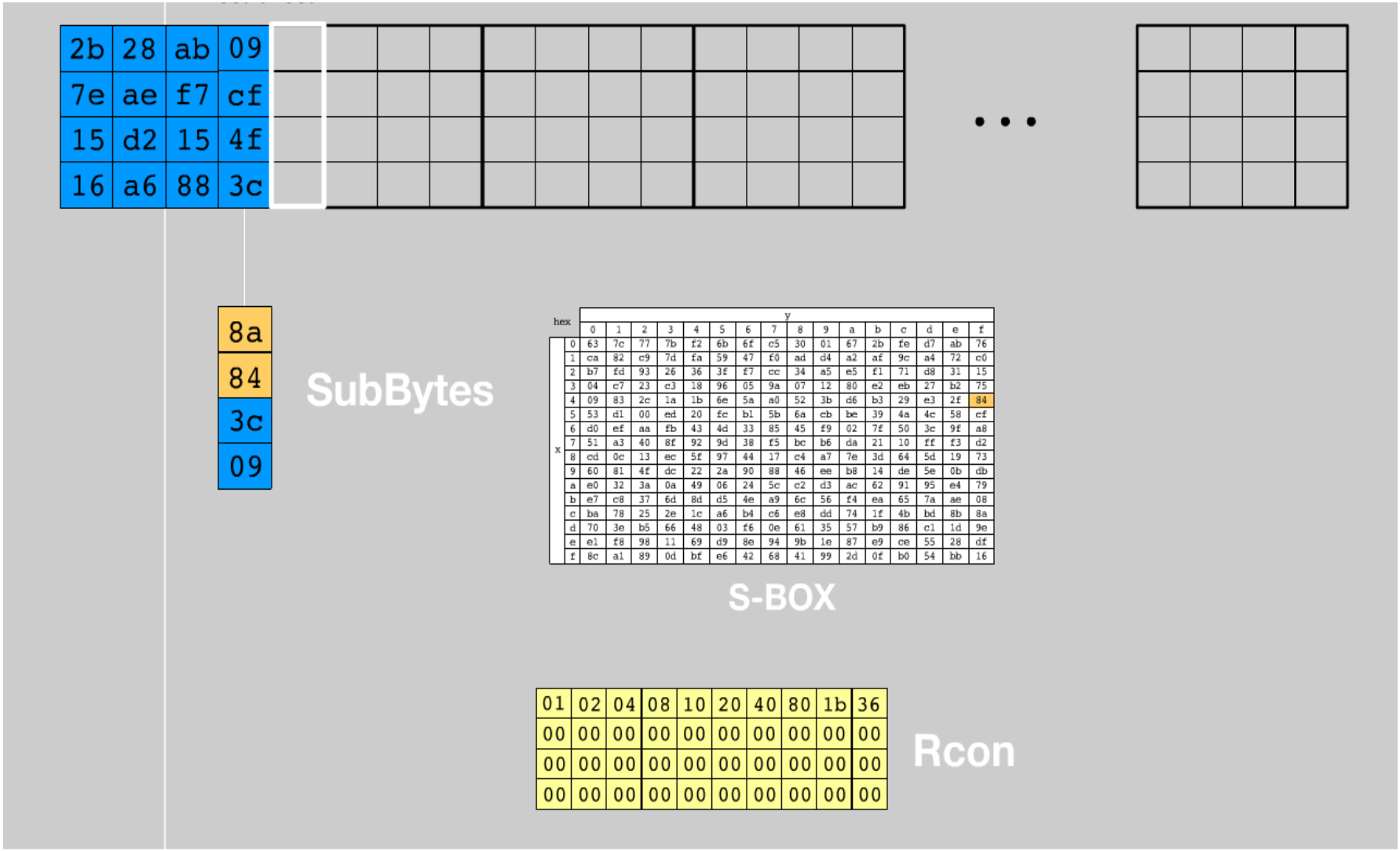


AES Key schedule generation (contd)

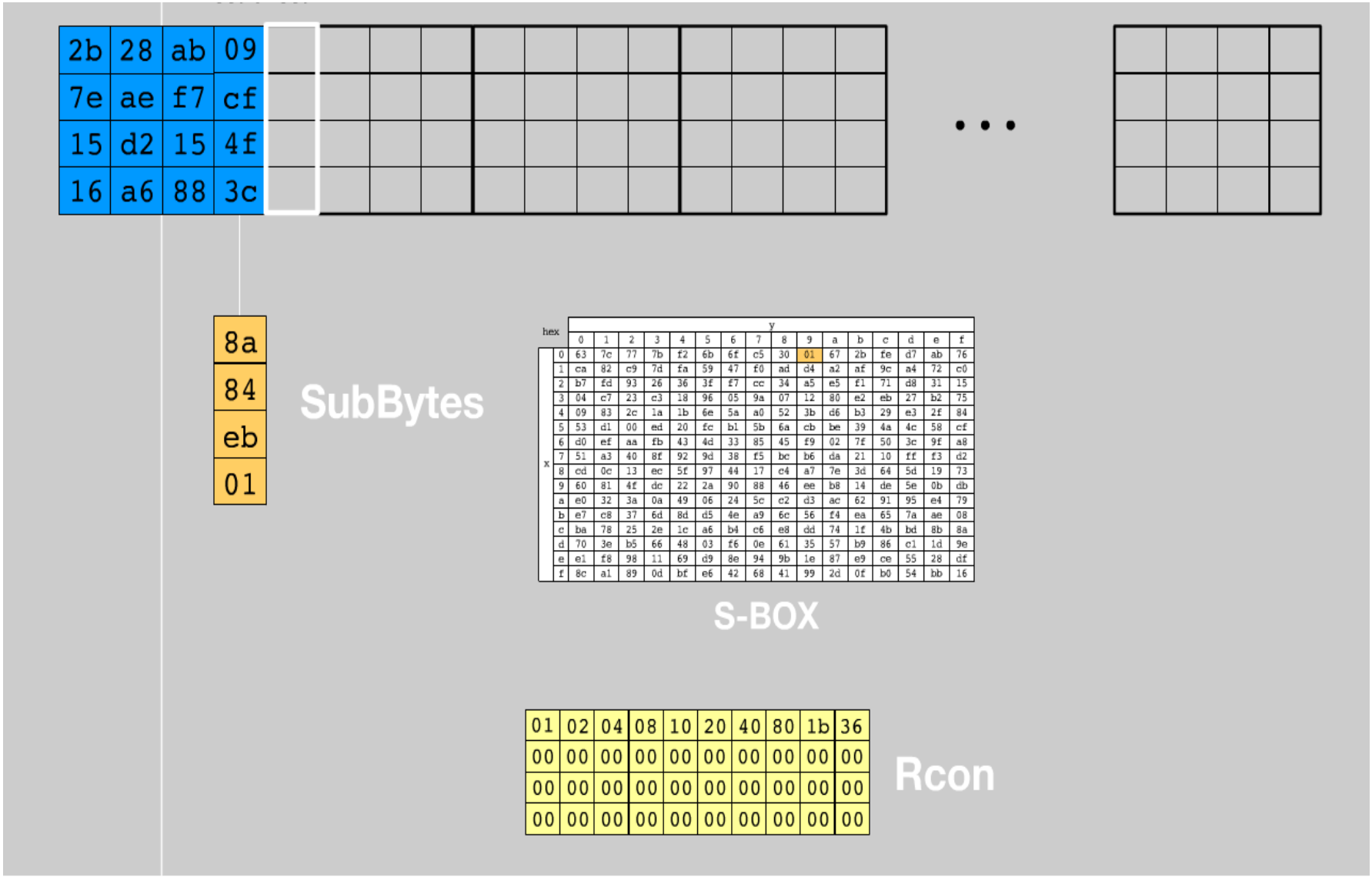


hex	y															
x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f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

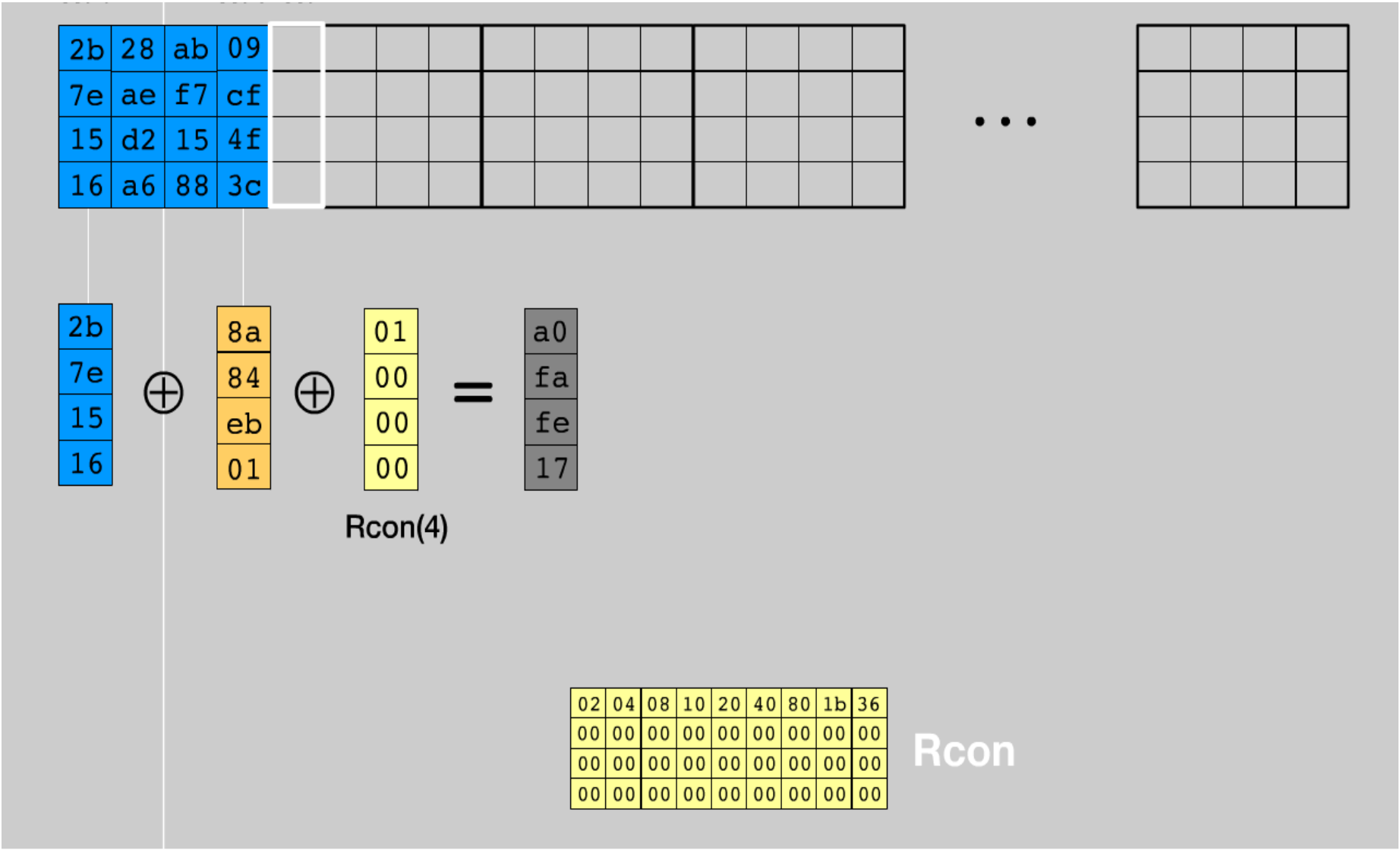
AES Key schedule generation (contd)



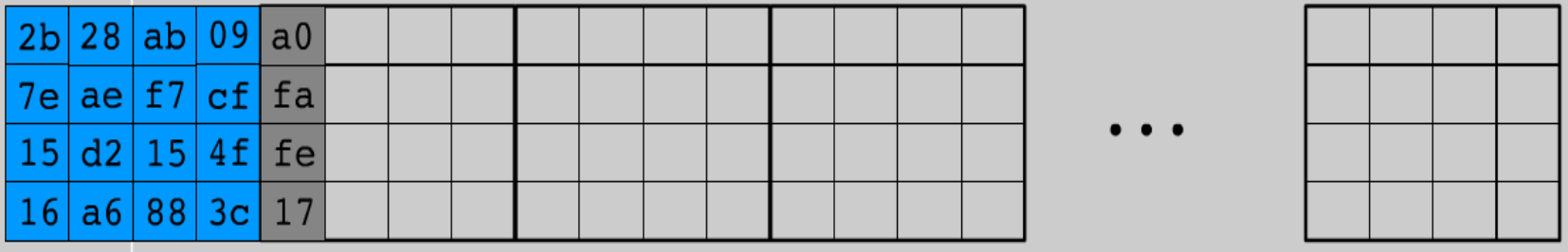
AES Key schedule generation (contd)

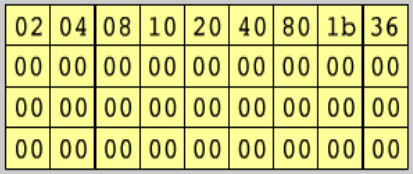


AES Key schedule generation (contd)



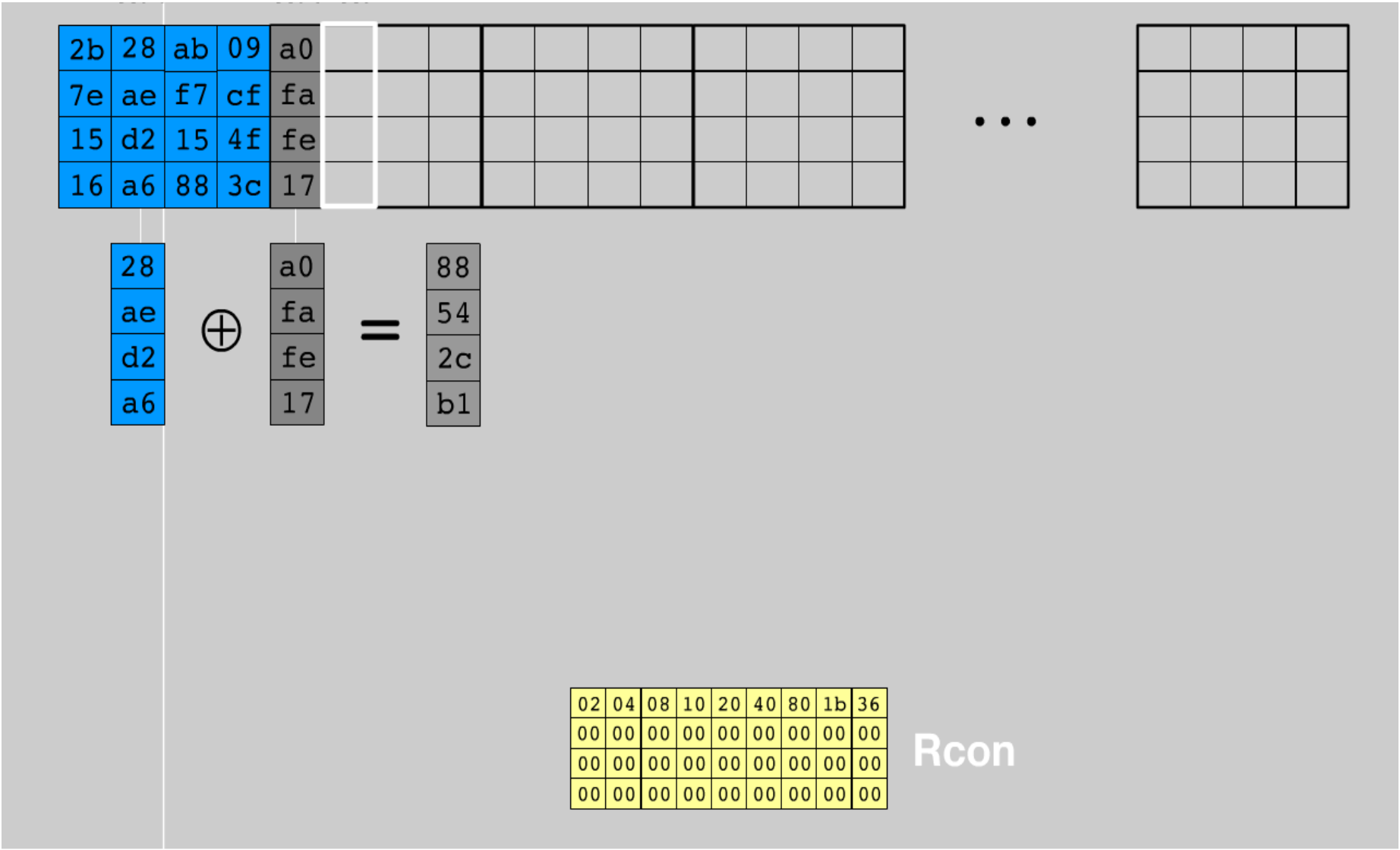
AES Key schedule generation (contd)



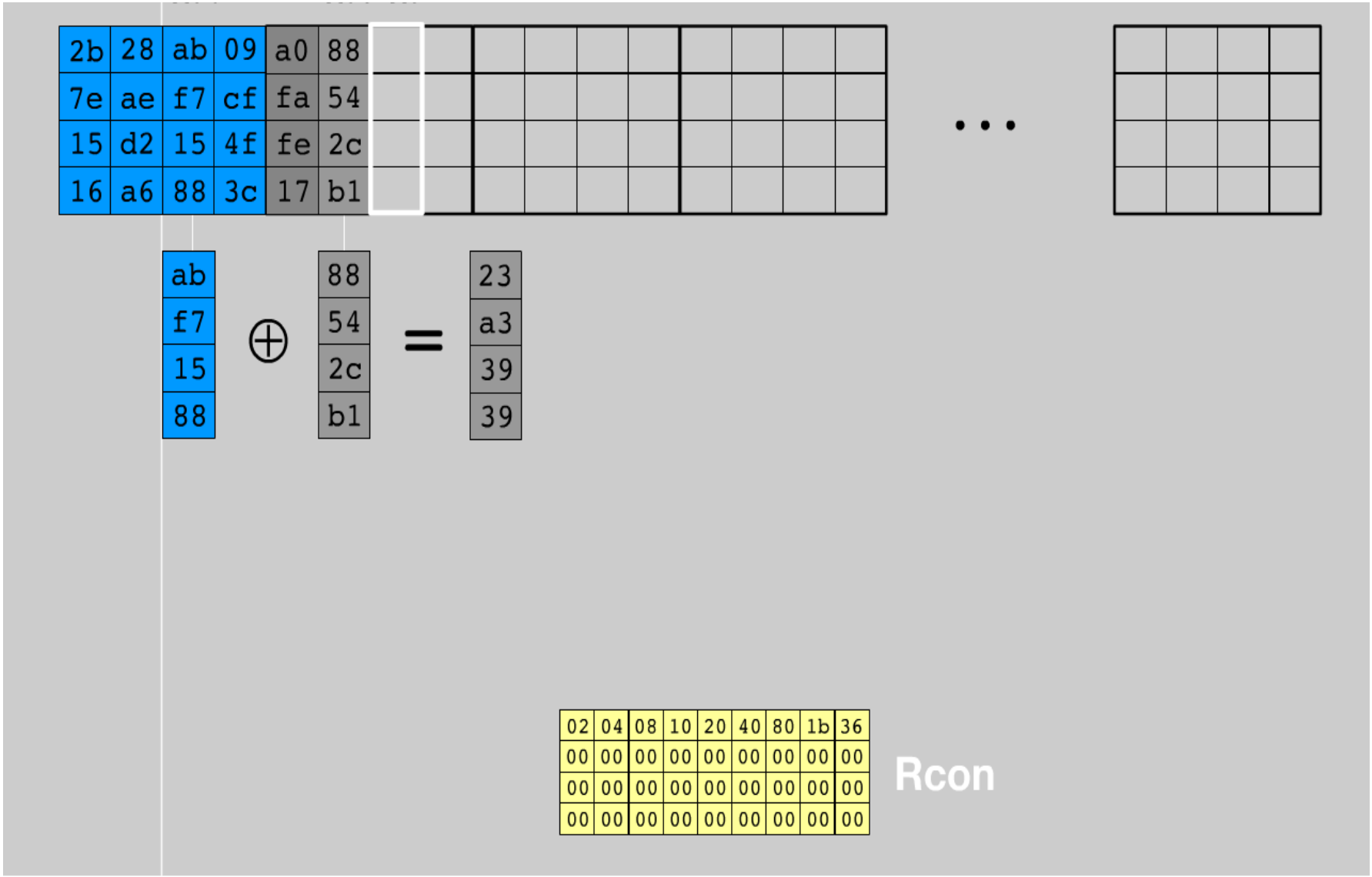


Rcon

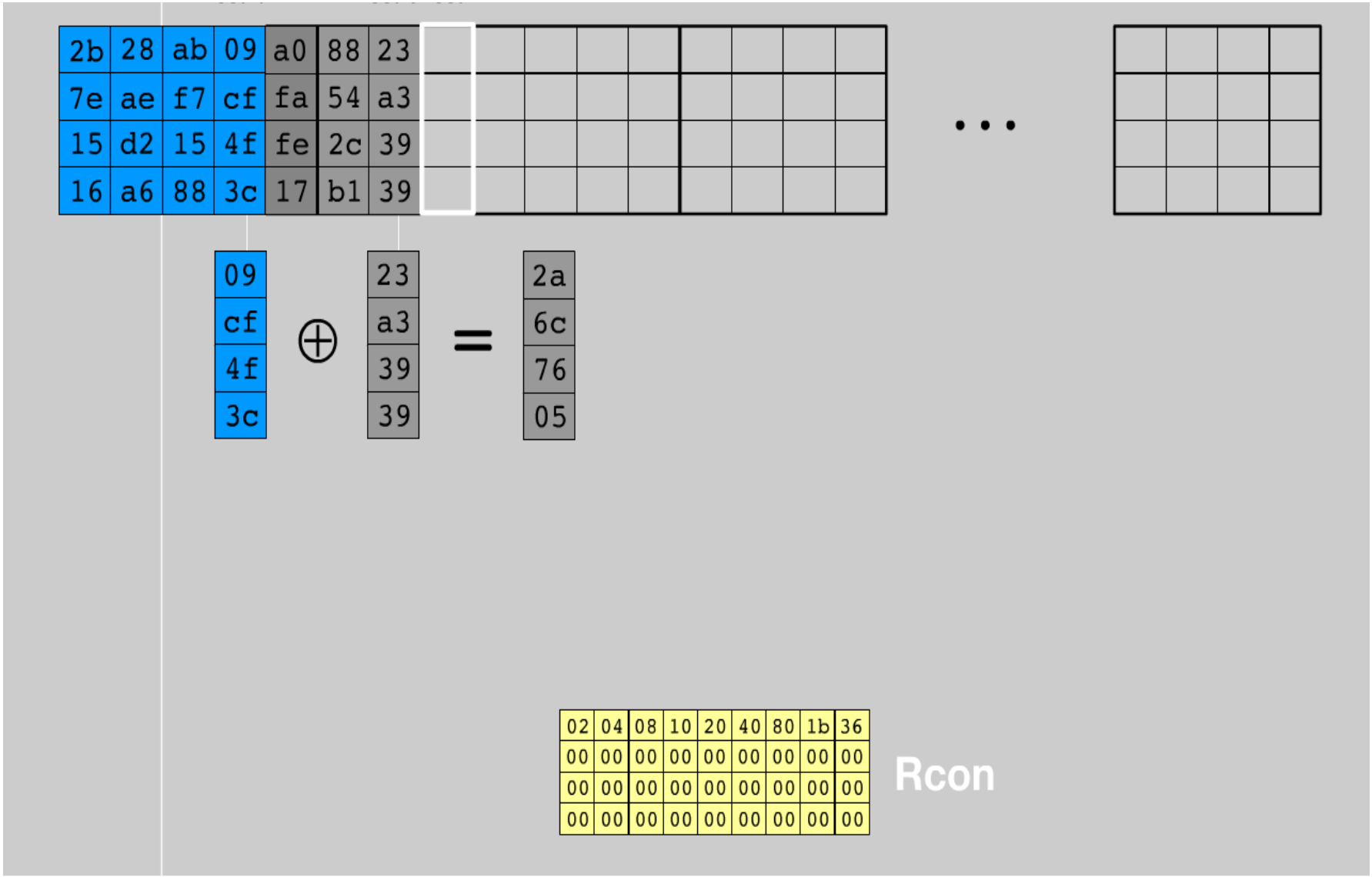
AES Key schedule generation (contd)



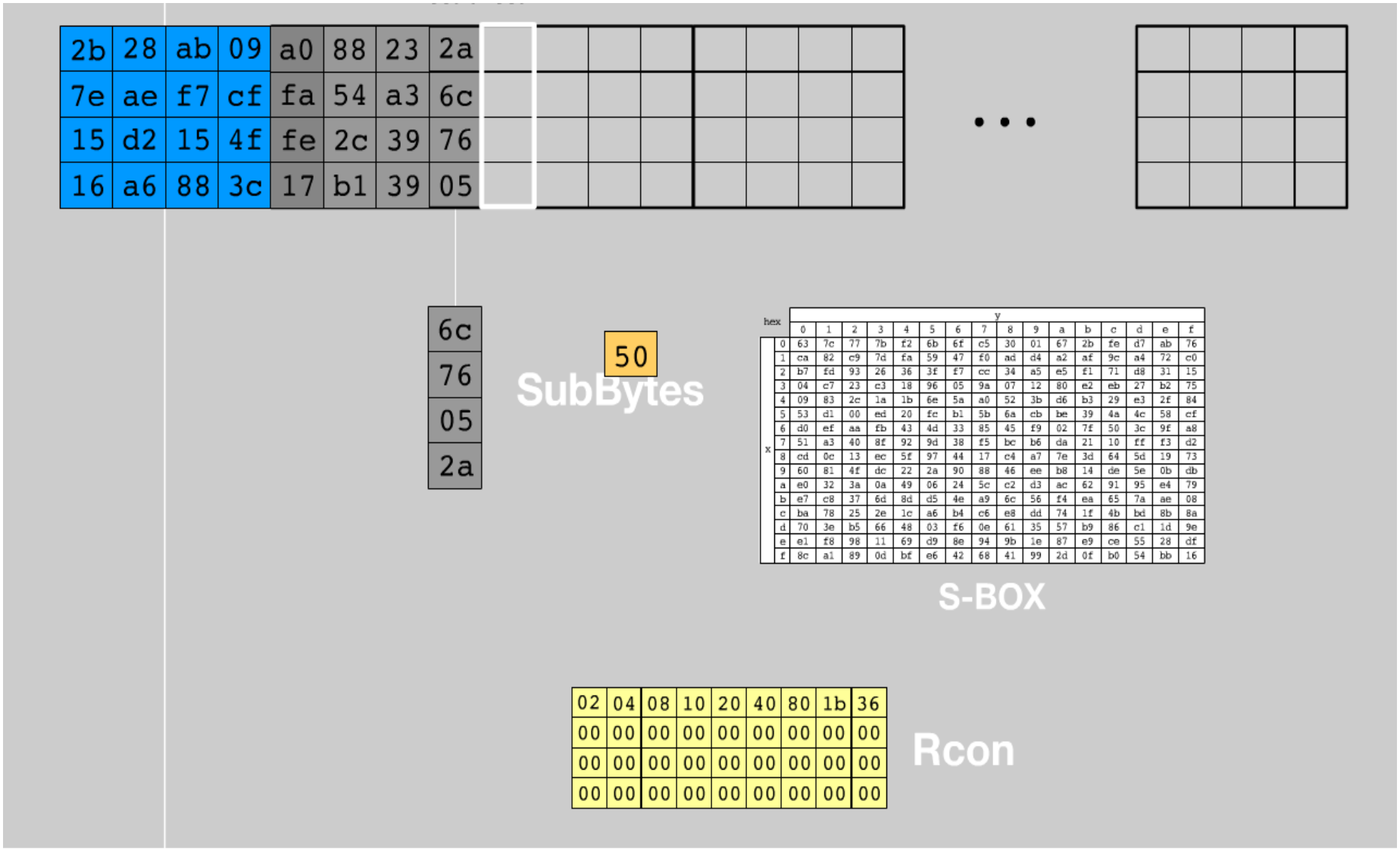
AES Key schedule generation (contd)



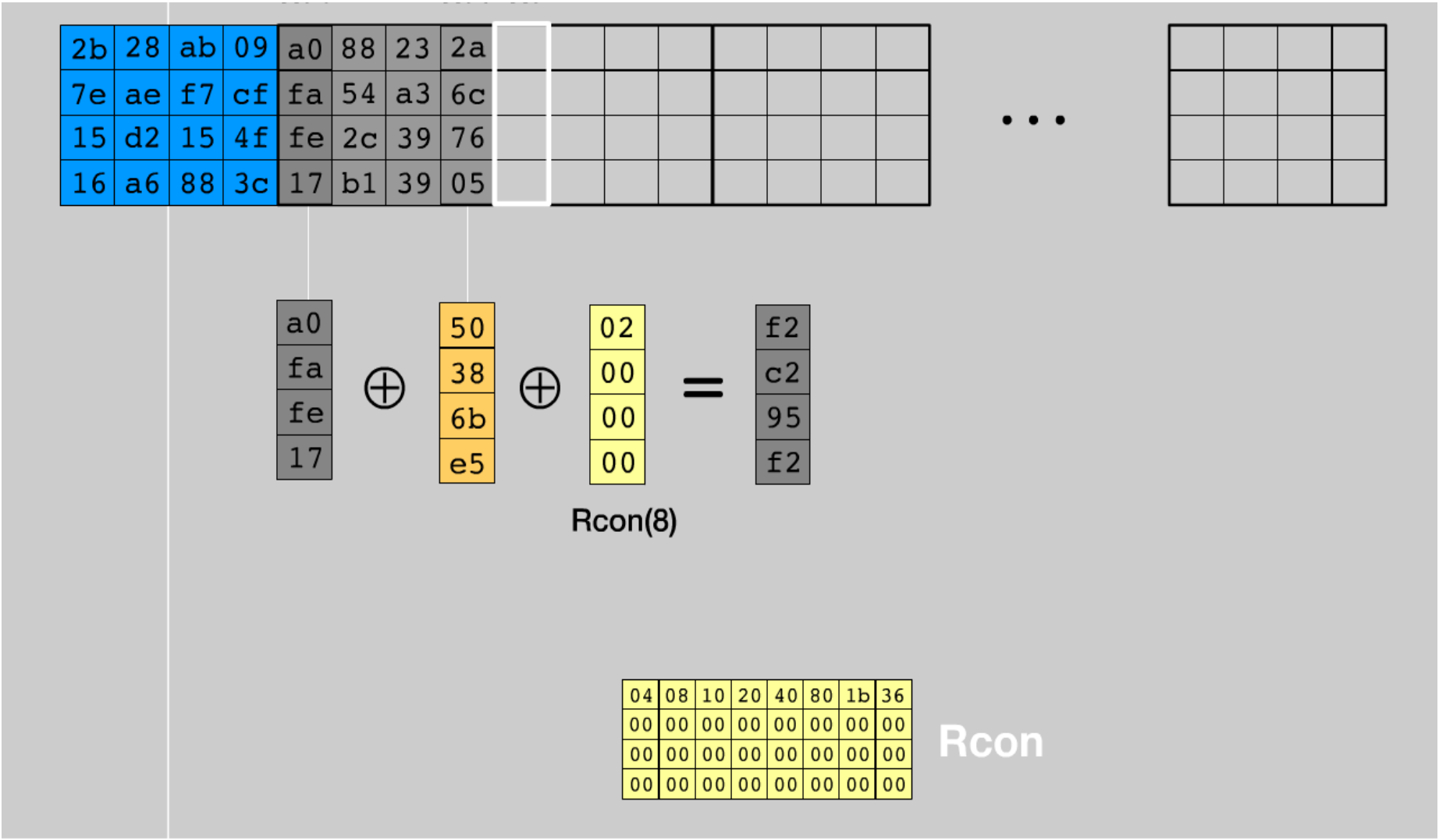
AES Key schedule generation (contd)



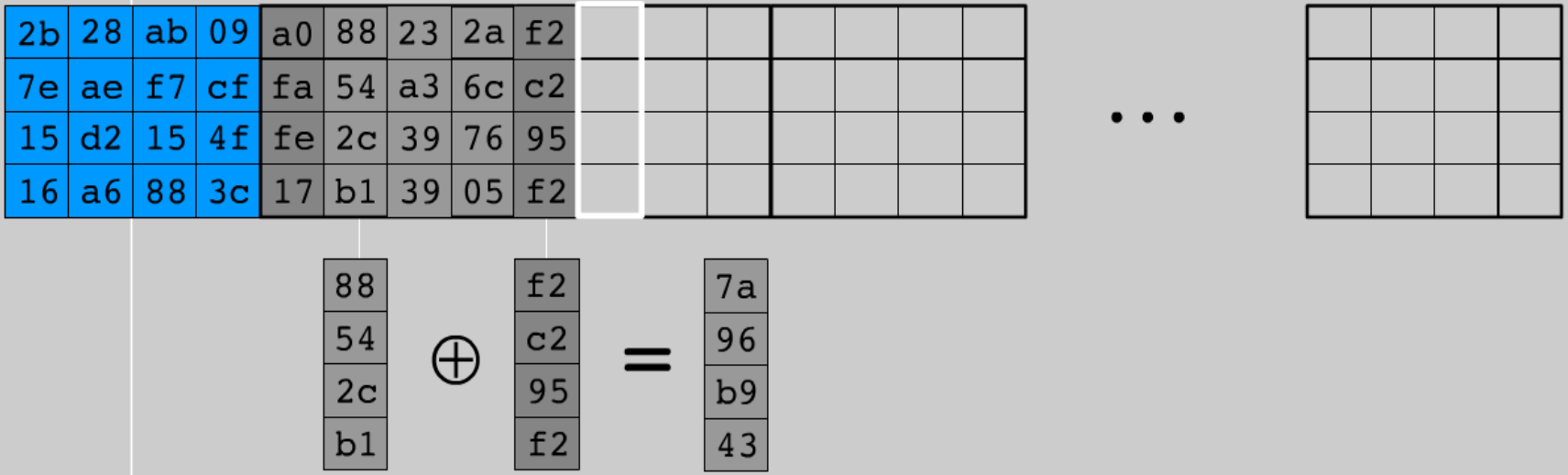
AES Key schedule generation (contd)



AES Key schedule generation (contd)



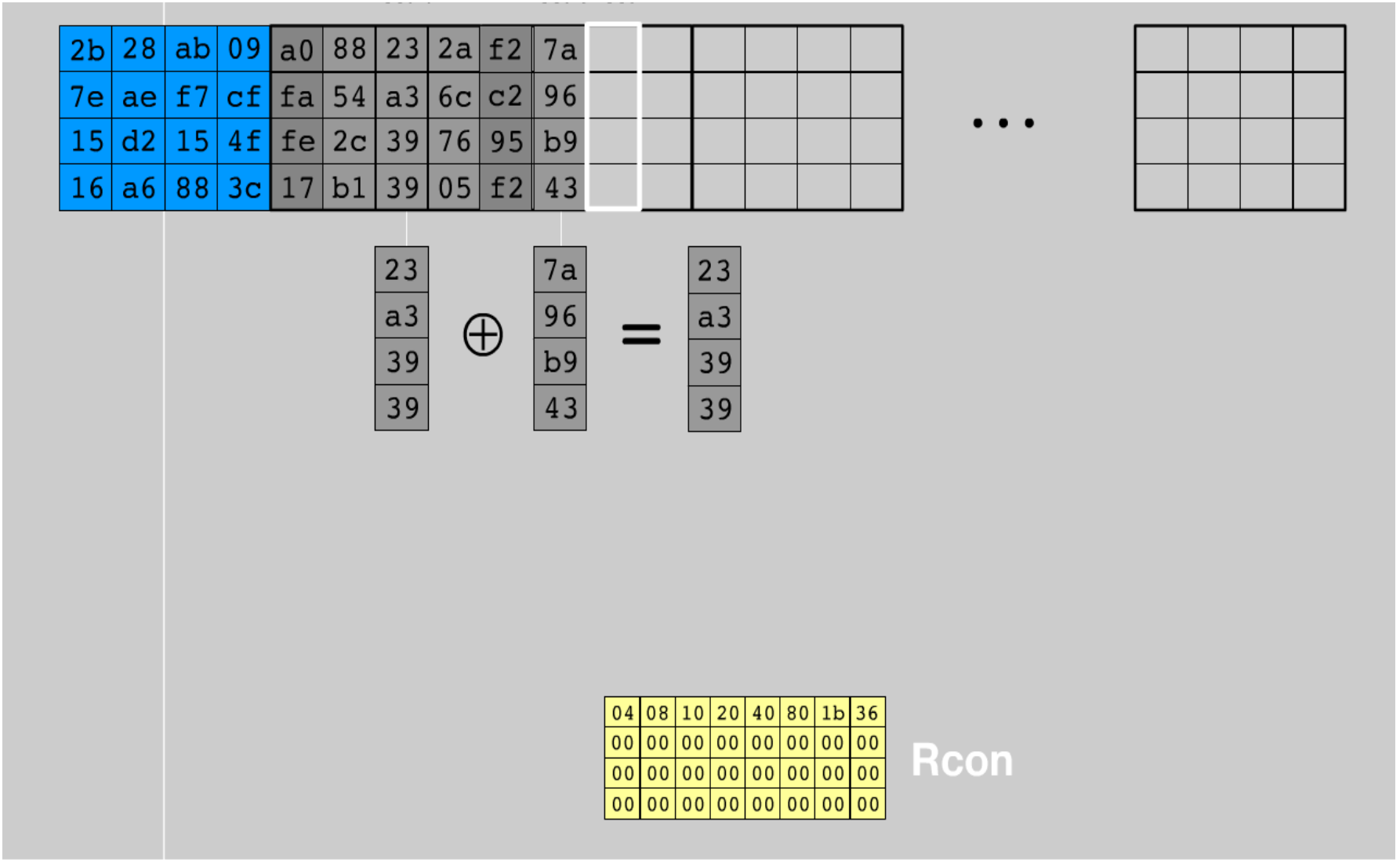
AES Key schedule generation (contd)



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

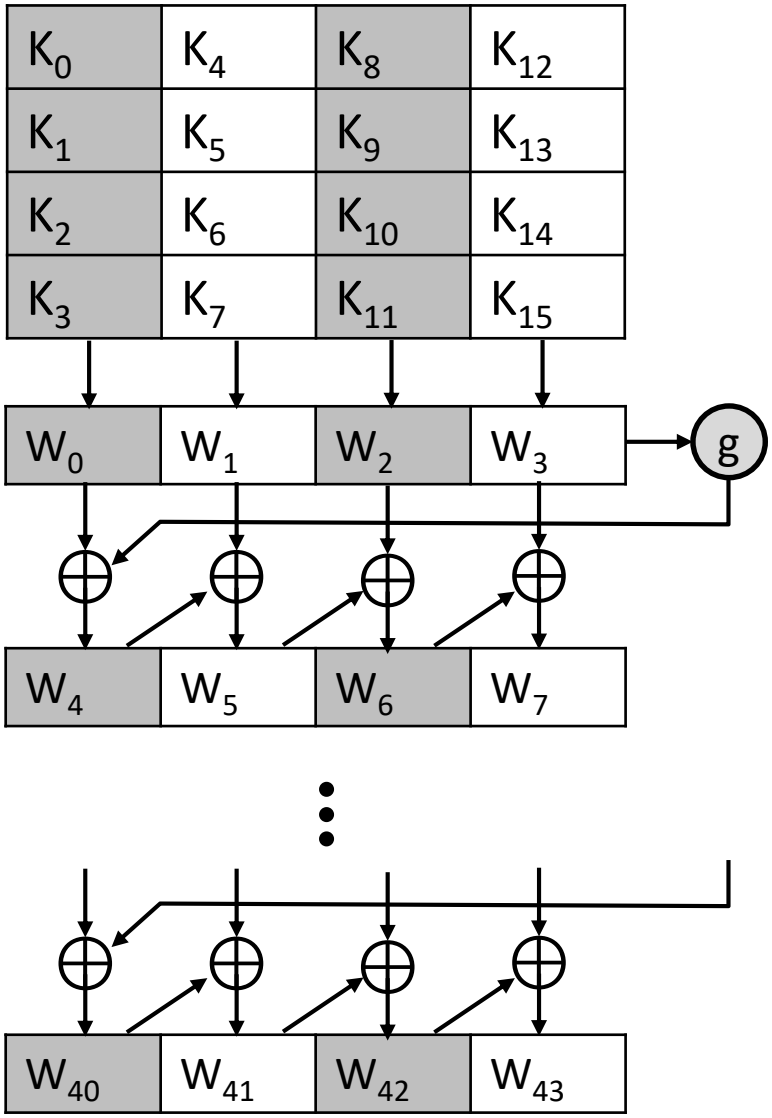
Rcon

AES Key schedule generation (contd)



AES Key schedule generation (contd)

AES key expansion

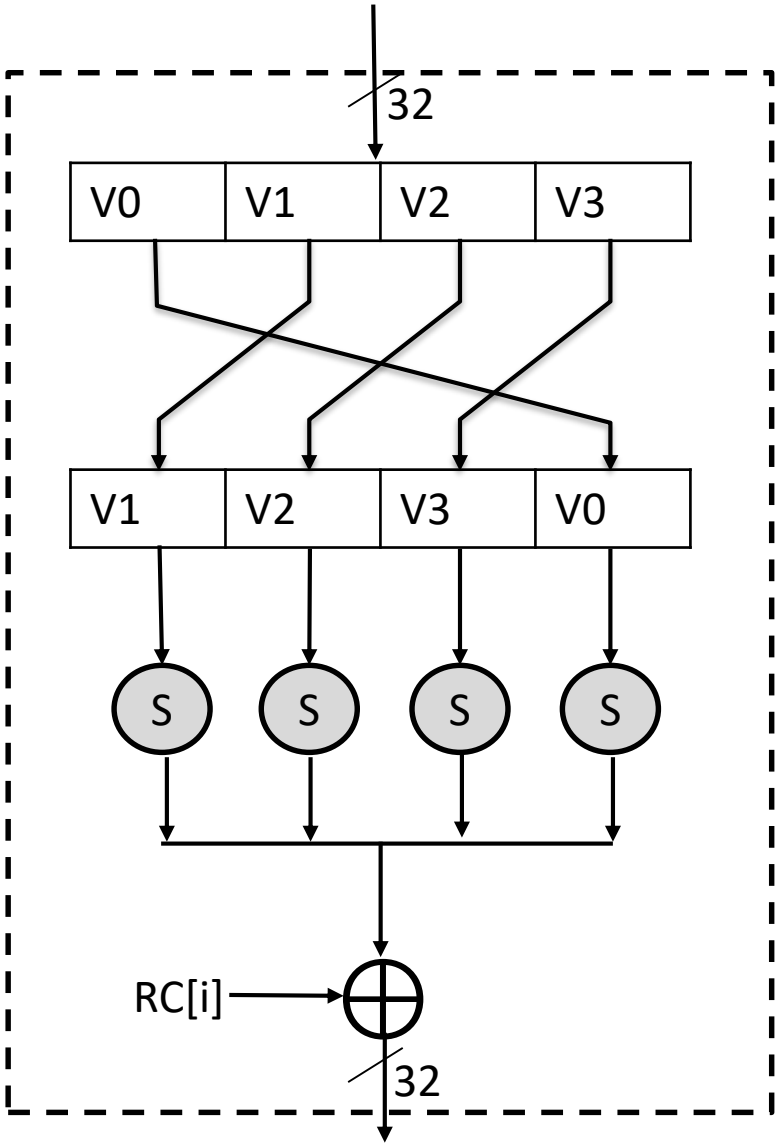


- The AES key expansion algorithm takes as input a four-word (16-byte) key and produces a linear array of **44 words** (176 bytes).
- Each added word **w[i]** depends on the immediately preceding word, w[i - 1].
- In three out of four cases, a simple XOR is used.

g function of key expansion

Rcon Table

1	2	3	4	5	6	7	8	9	10
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



Key Expansion Example

Plaintext:	0123456789abcdeffedcba9876543210
Key:	0f1571c947d9e8590cb7add6af7f6798
Ciphertext:	ff0b844a0853bf7c6934ab4364148fb9

Key Words	Auxiliary Function
w0 = 0f 15 71 c9 w1 = 47 d9 e8 59 w2 = 0c b7 ad d6 w3 = af 7f 67 98	RotWord(w3) = 7f 67 98 af = x1 SubWord(x1) = d2 85 46 79 = y1 Rcon(1) = 01 00 00 00 y1 ⊕ Rcon(1) = d3 85 46 79 = z1
w4 = w0 ⊕ z1 = dc 90 37 b0 w5 = w4 ⊕ w1 = 9b 49 df e9 w6 = w5 ⊕ w2 = 97 fe 72 3f w7 = w6 ⊕ w3 = 38 81 15 a7	RotWord(w7) = 81 15 a7 38 = x2 SubWord(x2) = 0c 59 5c 07 = y2 Rcon(2) = 02 00 00 00 y2 ⊕ Rcon(2) = 0e 59 5c 07 = z2
w8 = w4 ⊕ z2 = d2 c9 6b b7 w9 = w8 ⊕ w5 = 49 80 b4 5e w10 = w9 ⊕ w6 = de 7e c6 61 w11 = w10 ⊕ w7 = e6 ff d3 c6	RotWord(w11) = ff d3 c6 e6 = x3 SubWord(x3) = 16 66 b4 83 = y3 Rcon(3) = 04 00 00 00 y3 ⊕ Rcon(3) = 12 66 b4 8e = z3
w12 = w8 ⊕ z3 = c0 af df 39 w13 = w12 ⊕ w9 = 89 2f 6b 67 w14 = w13 ⊕ w10 = 57 51 ad 06 w15 = w14 ⊕ w11 = b1 ae 7e c0	RotWord(w15) = ae 7e c0 b1 = x4 SubWord(x4) = e4 f3 ba c8 = y4 Rcon(4) = 08 00 00 00 y4 ⊕ Rcon(4) = ec f3 ba c8 = 4

Thank You!!