UNIT-2_1

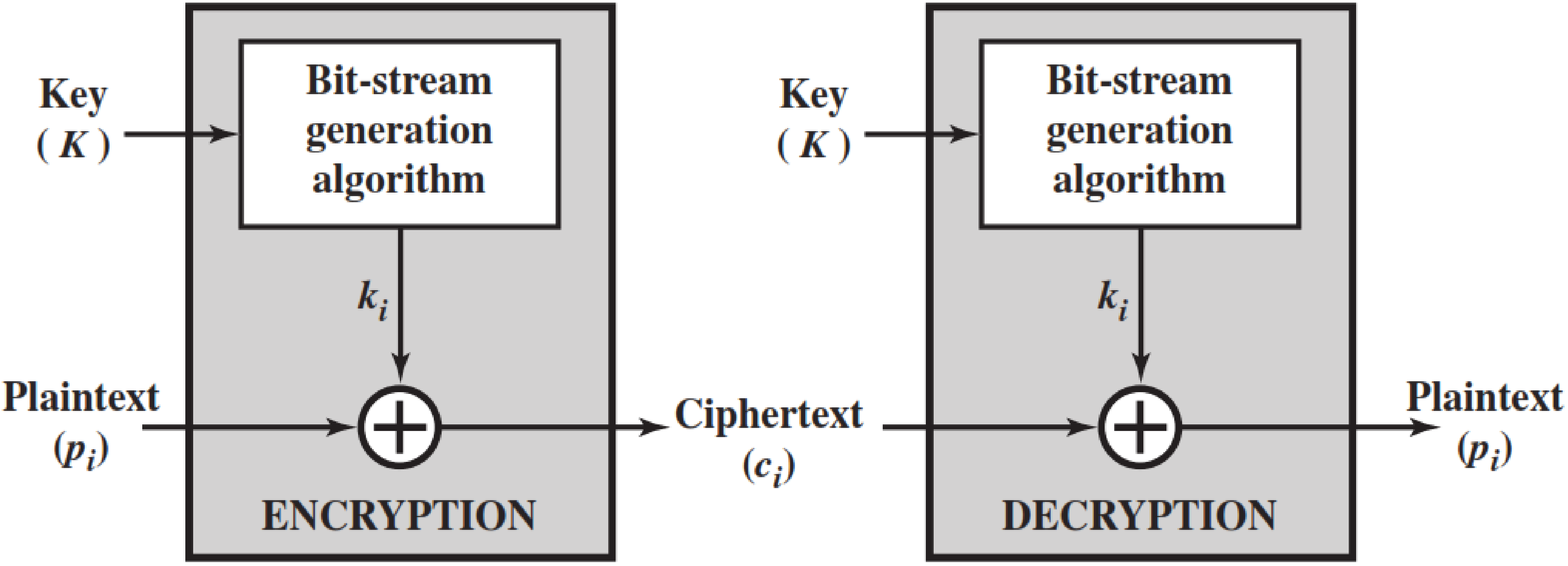
Stream ciphers and block ciphers

Unit-2

- Stream ciphers and block ciphers
- Block Cipher structure
- Data Encryption standard (DES)
- Design principles of block cipher
- AES with structure
- AES Transformation functions
- Key expansion

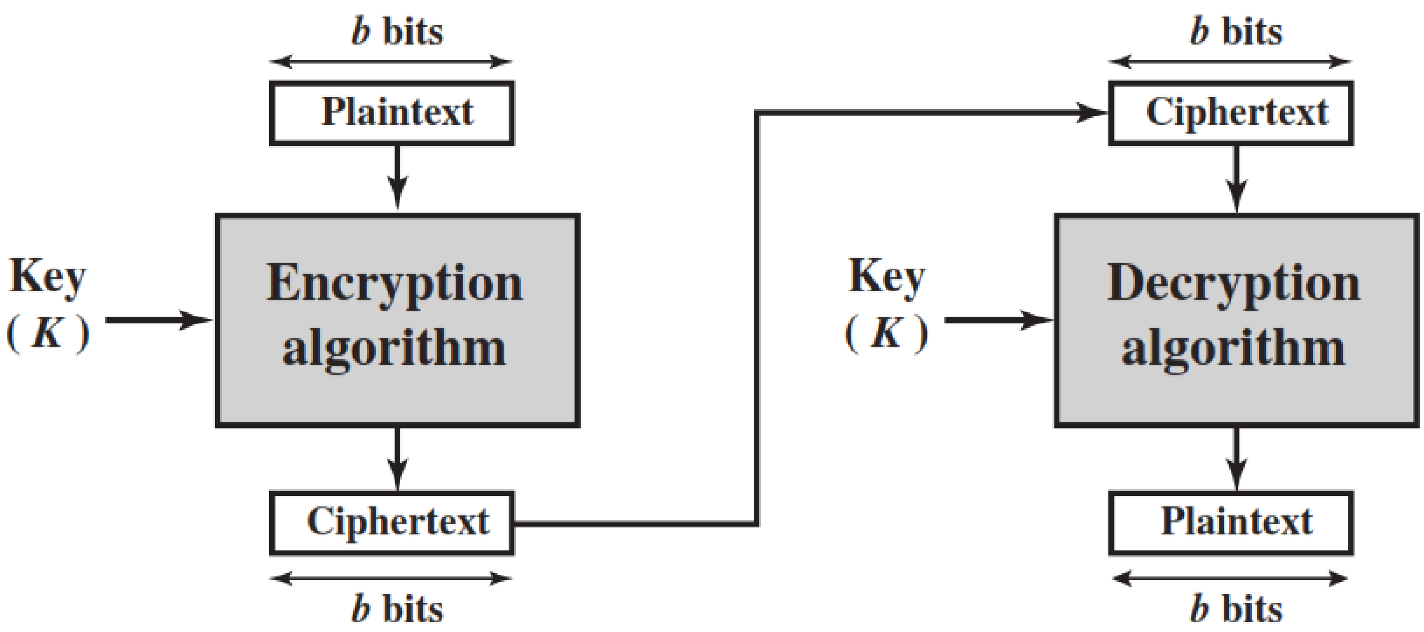
Stream Cipher

- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time.
- Examples of classical stream ciphers are Autokeyed Vigenère cipher ,A5/1, RC4 and Vernam cipher.



Block Cipher

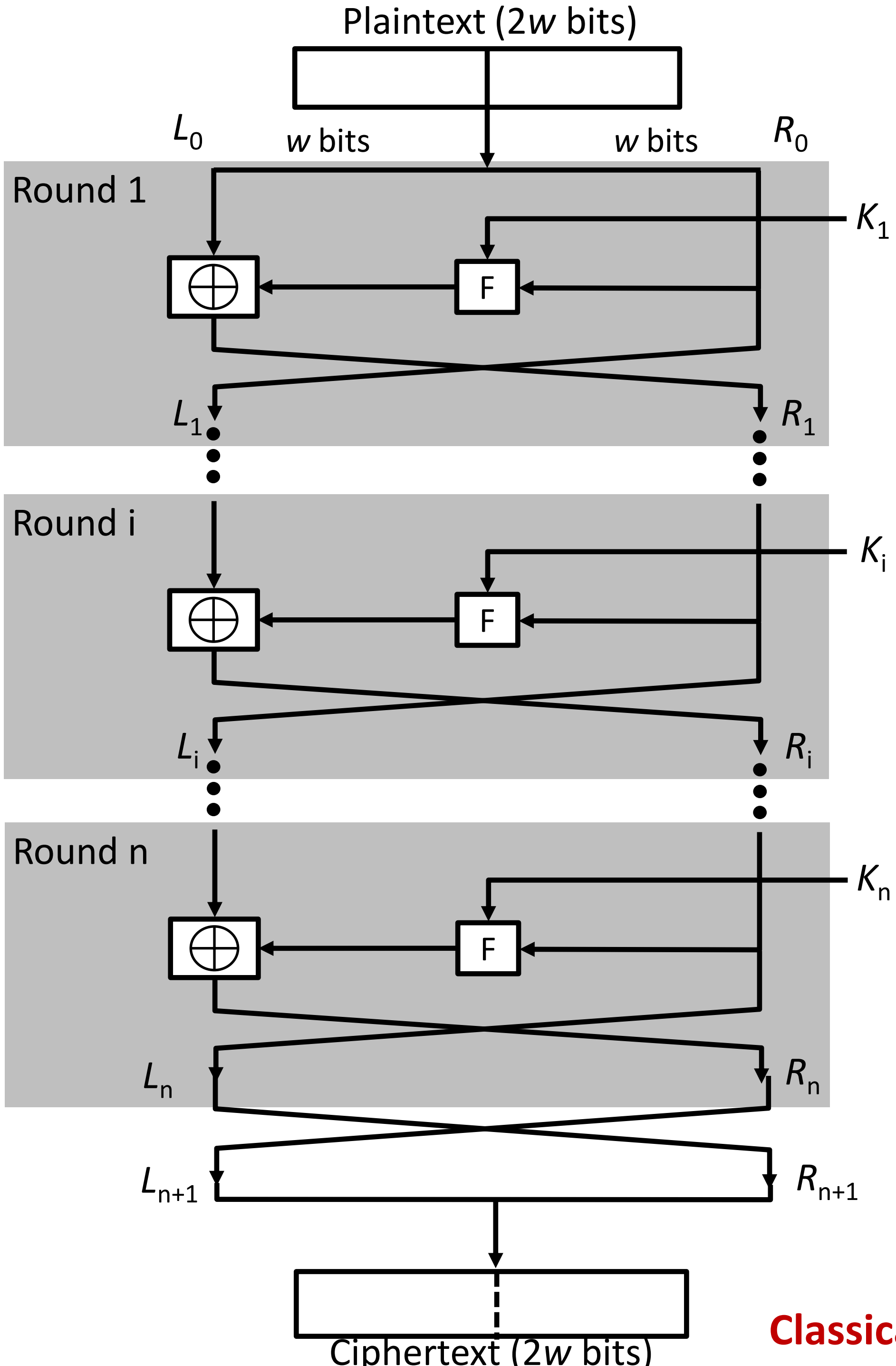
- A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Typically, a block size of **64 or 128** bits is used.
- Examples are Feistel Cipher, DES, Triple DES and AES



Diffusion and Confusion

- **Diffusion** hides the relationship between the ciphertext and the plaintext.
- This is achieved by having each plaintext digit affect the value of many ciphertext digits.
- **Confusion** hides the relationship between the ciphertext and the key.
- This is achieved by the use of a complex substitution algorithm.

Feistel Cipher Structure Or Block Cipher Structure



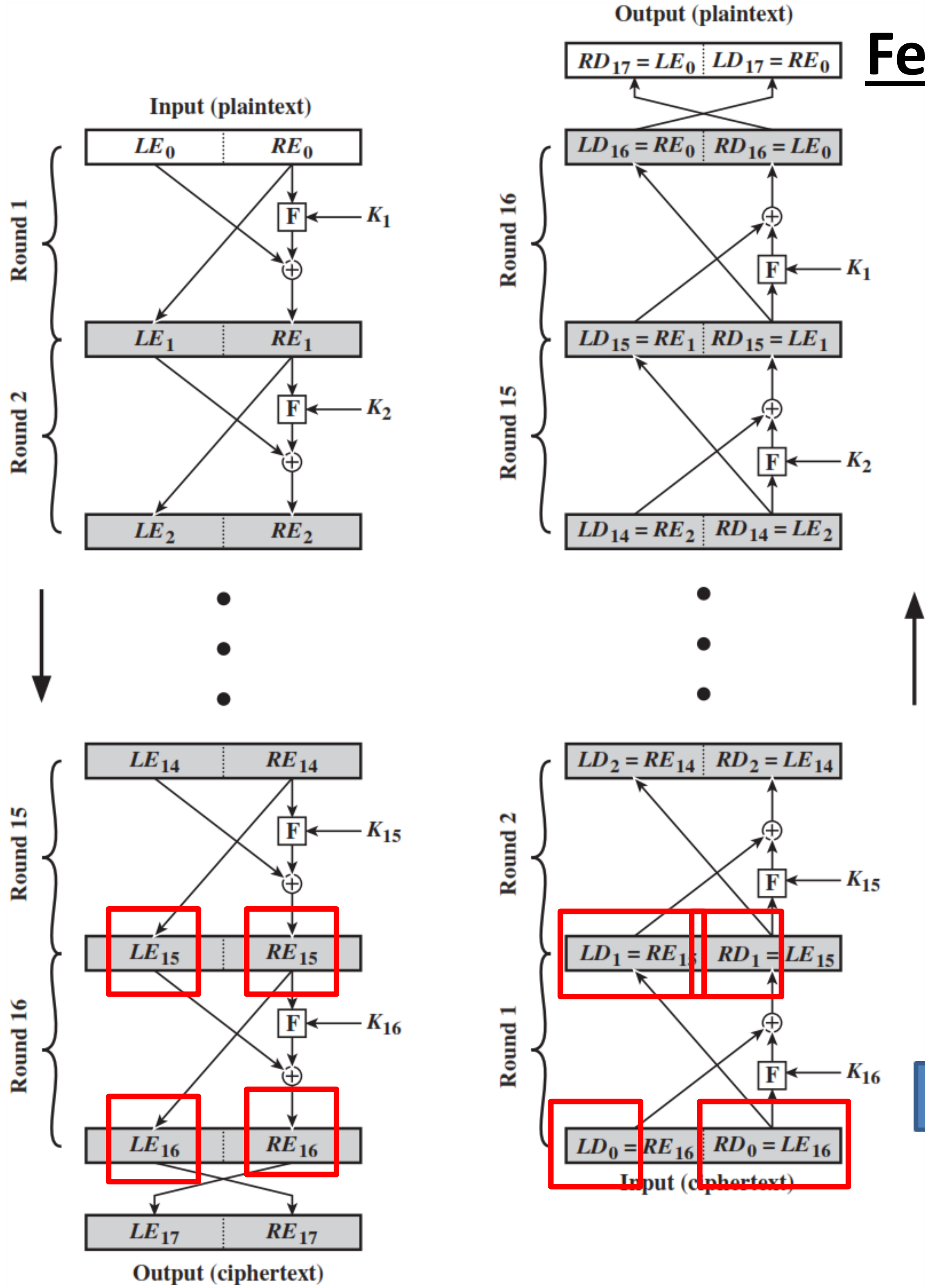
Classical Feistel Network

Feistel Cipher Structure

- Input plaintext block of length 2w bits
- key K = n bits , Sub-keys: K_1, K_2, …, K_n (Derived from K)
- All rounds have the same structure.
- A **substitution** is performed by taking exclusive-OR on left half(Li) of the data and the output of round function F which has inputs right half(Ri) and sub key ki.
- A **permutation** is performed that consists of interchange of two halves of data.
- This structure is called **Substitution-Permutation Network** (SPN)

Feistel Network Factors

- **Block size:** Common block size of 64-bit. However, the new algorithms uses a 128-bit, 256-bit block size.
- **Key size:** Key sizes of 64 bits or less are now widely considered to be insufficient, These days at least 128 bit, more better, e.g. 192 or 256 bit
- **Number of rounds:** A typical size is 16 rounds.
- **Round function F:** Again, greater complexity generally means greater resistance to cryptanalysis.
- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.



Feistel Encryption & Decryption

- Prove that o/p of first round of Decryption is equal to 32-bit swap of i/p of 16th round of Encryption
- $LD_1=RE_{15}$ & $RD_1=LE_{15}$
- On Encryption Side:

$$LE_{16} = RE_{15}$$

$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$

- On Decryption Side:

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$

$$RD_1 = LD_0 \oplus F(RD_0, K_{16})$$

$$= RE_{16} \oplus F(RE_{15}, K_{16})$$

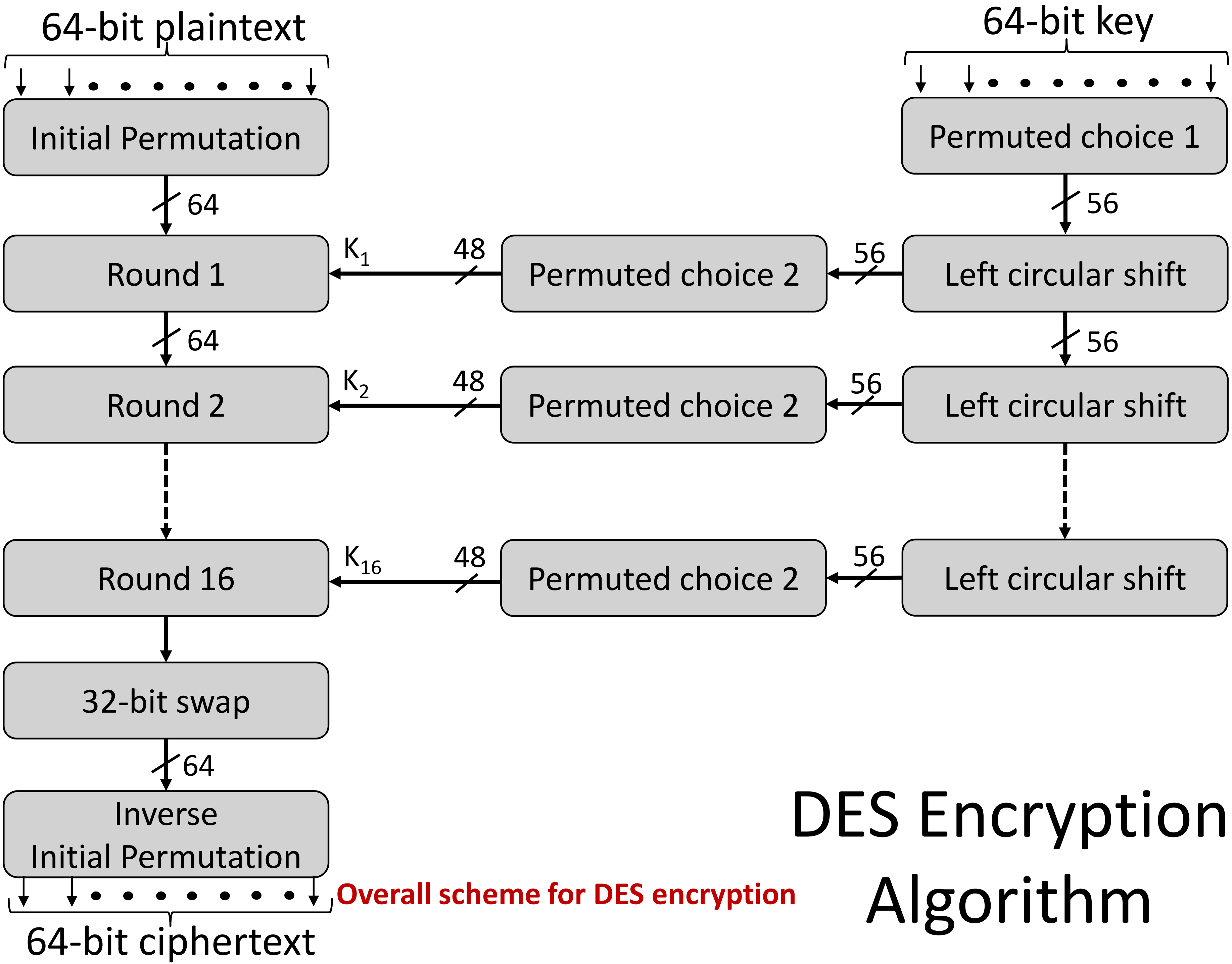
$$= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16})$$

Thus,

$$LD_1 = RE_{15} \;\&\; RD_1 = LE_{15}$$

Data Encryption Standard (DES)

- Type: Block Cipher
- Block Size : 64-bit
- Key Size: 64-bit, with only 56-bit effective
- Number of Rounds: 16



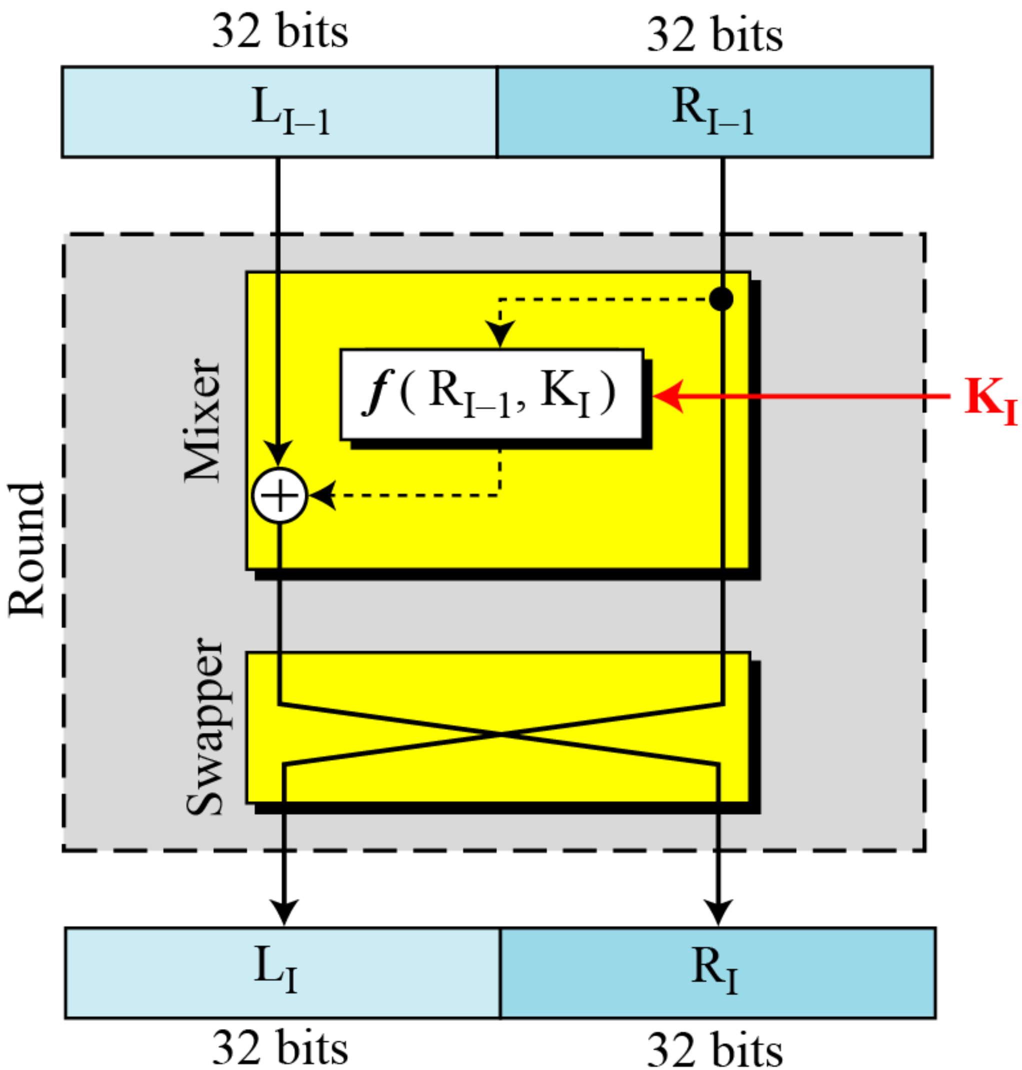
Overall scheme for DES encryption

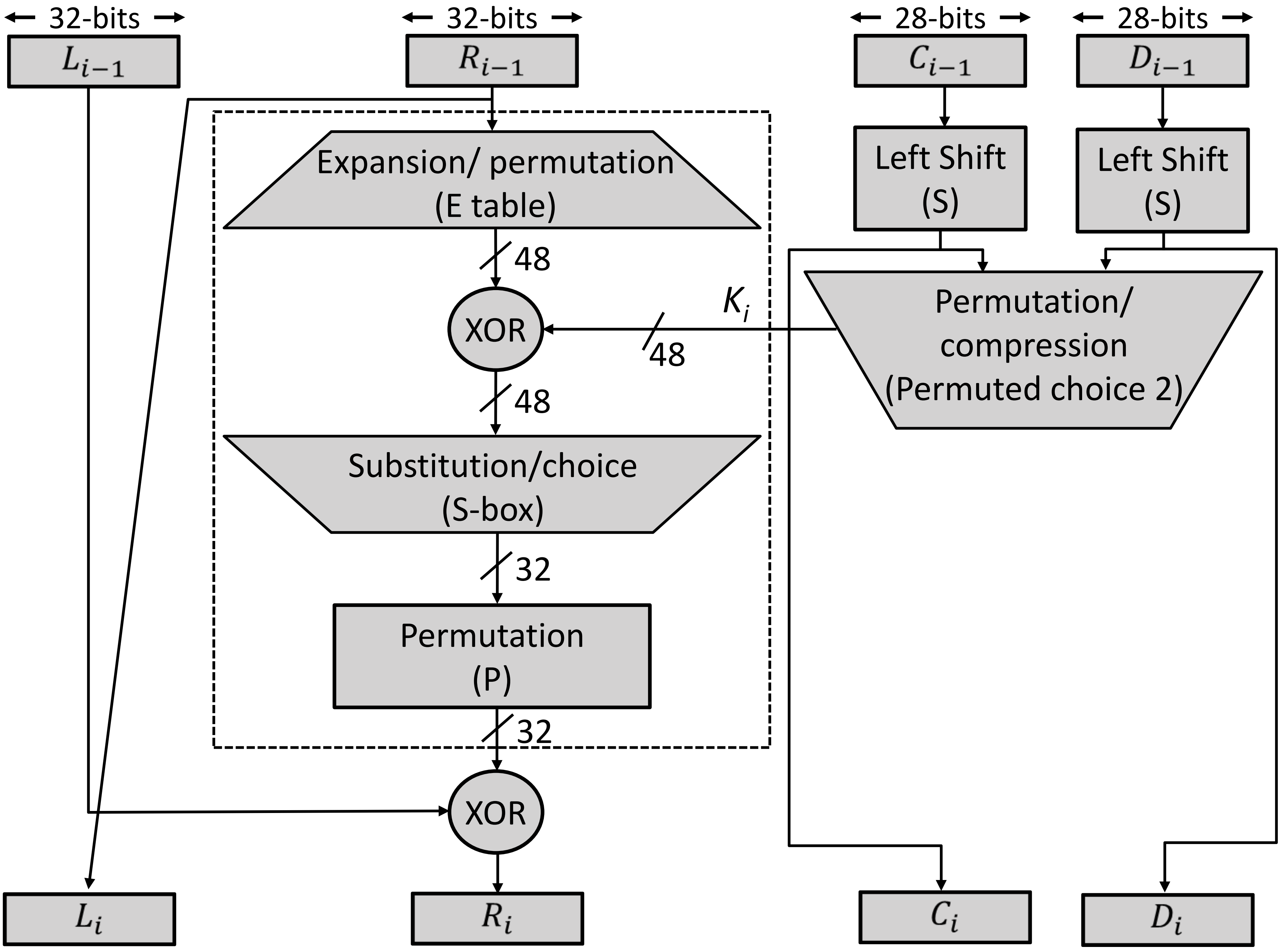
DES Encryption Algorithm

DES Encryption Algorithm (Cont…)

- First, the 64-bit plaintext passes through an **initial permutation** (IP) that rearranges the bits to produce the permuted input.
- This is followed by a phase consisting of sixteen rounds of the same function, which involves both **permutation** and **substitution** functions.
- Finally, the preoutput is passed through a permutation that is the **inverse of the initial permutation** function, to produce the 64-bit ciphertext.
- The 56-bit key is passed through a **permutation function**.
- For each of the sixteen rounds, a subkey (K_i) is produced by the combination of a **left circular shift** and a **permutation**.

DES Single Round



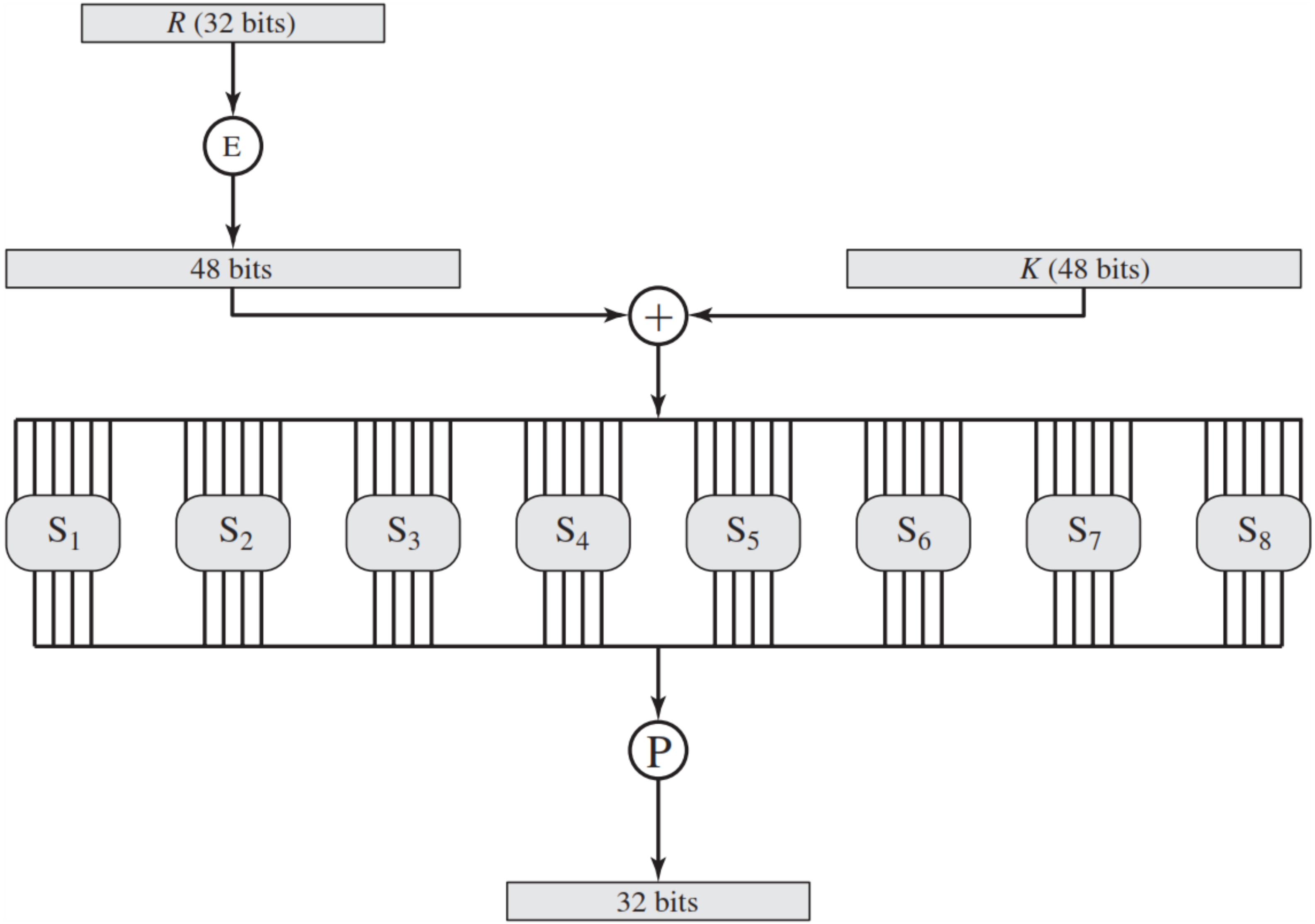


Single Round of DES in Detail

DES Single Round (Cont…)

1. Key Transformation
   - Permutation of selection of sub-key from original key
2. Expansion Permutation (E-table)
   - Right half is expanded from 32-bits to 48-bits
3. S-box Substitution
   - Accepts 48-bits from XOR operation and produce 32-bits using 8 substitution boxes (each S-boxes has a 6-bit i/p and 4-bit o/p).
4. P-Box Permutation
5. XOR and Swap

Role of S-boxes in the function F



Role of S-box (Cont…)

- The outer two bits of each group select one row of an S-box.
- Inner four bits selects one column of an S-box.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

S-box 1

- Example:

Input | 0 | 1 | 1 | 0 | 0 | 1 |

Output | 1 | 0 | 0 | 1 |

Row

Column

Avalanche Effect

- Desirable property of any encryption algorithm is that a change in one bit of the plaintext or of the key should produce a change in many bits of cipher text.
- DES performs strong **avalanche effect**.

Plaintext: 0000000000000000	Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1	
Plaintext: 000000000000000<u>1</u>	Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3	

- Although the two plaintext blocks differ only in the rightmost bit, the ciphertext blocks differ in 29 bits.
- This means that changing approximately 1.5 % of the plaintext creates a change of approximately 45 % in the ciphertext.

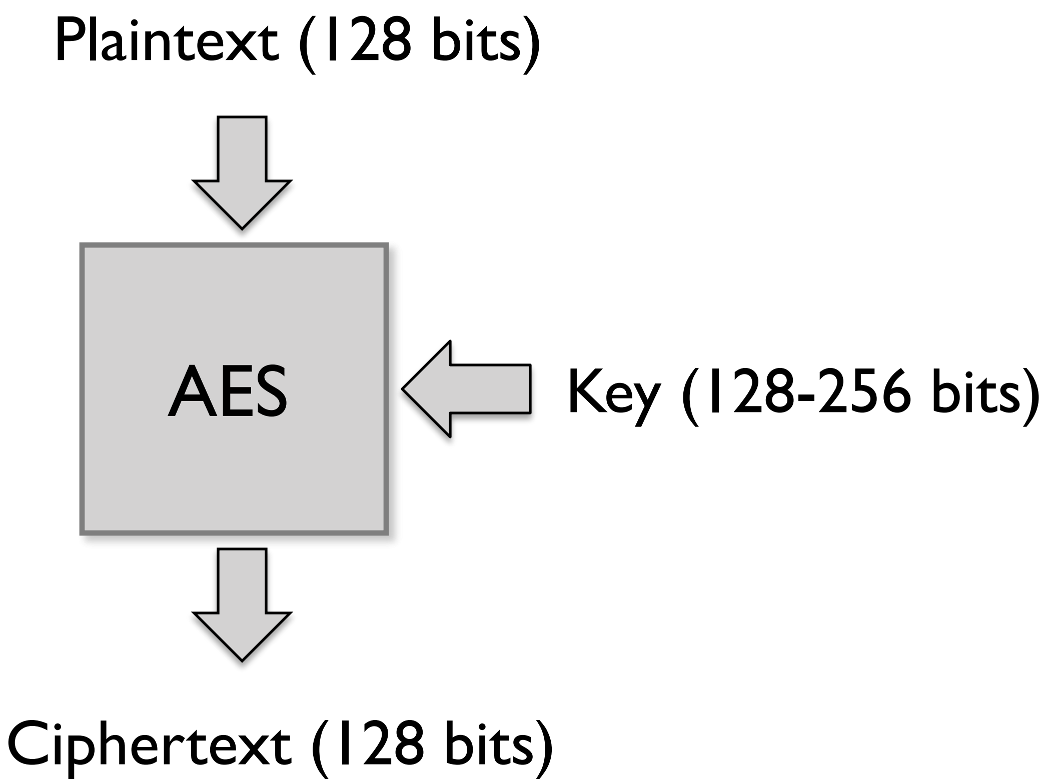
AES (Advanced Encryption Standard)

- 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**
Expanded Key Size (words)	**44**	**52**	**60**

- 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 Overview

- Simple Repeating structure
- Cipher begins and ends with Add Round Key
  - Forms a Vernam Cipher or "One Time Pad"
    - Any other stage applied at the beginning or end is reversible without the key
- Other three stages provide confusion, diffusion and nonlinearity
- n standard rounds, n is 10,12 or 14
- The first n-1 rounds are similar consisting of
  - ByteSub
  - ShiftRow
  - MixColumn
  - AddRoundKey
- The last round only perform the transformations
  - ByteSub
  - ShiftRow
  - AddRoundKey

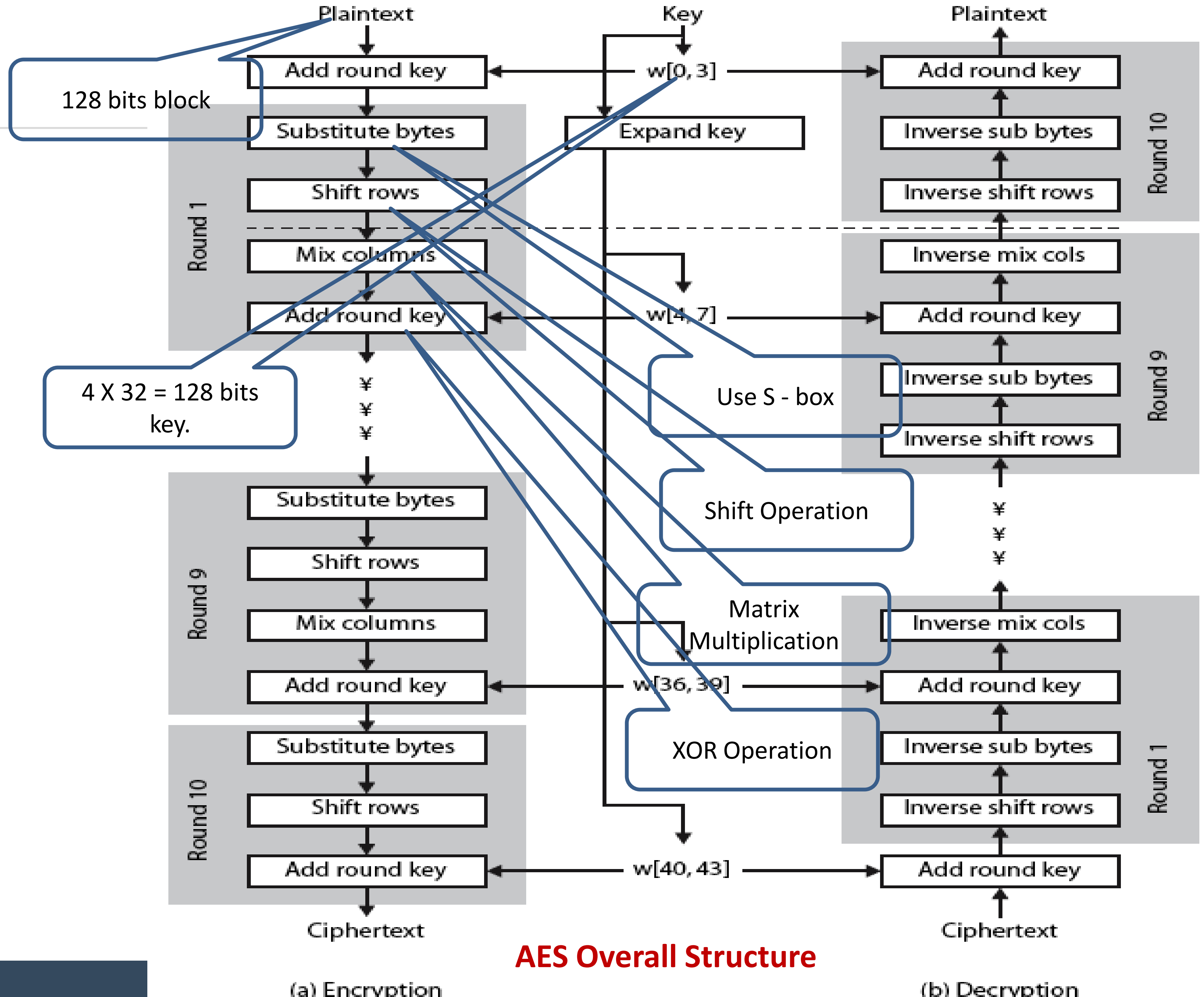
Initialization

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**.

AES Structure

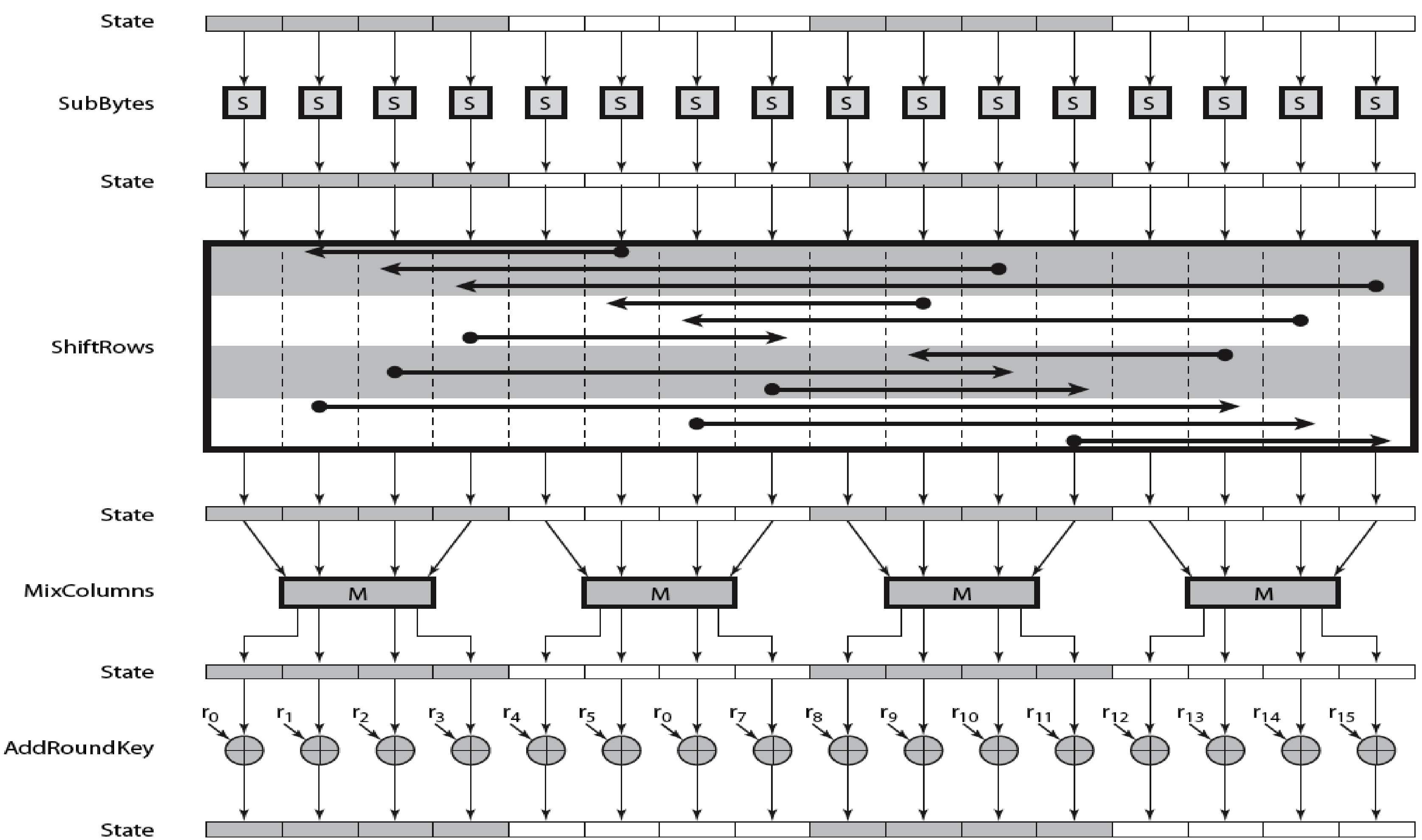
- The first n-1 rounds consist of four distinct transformation functions.

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 Overall Structure

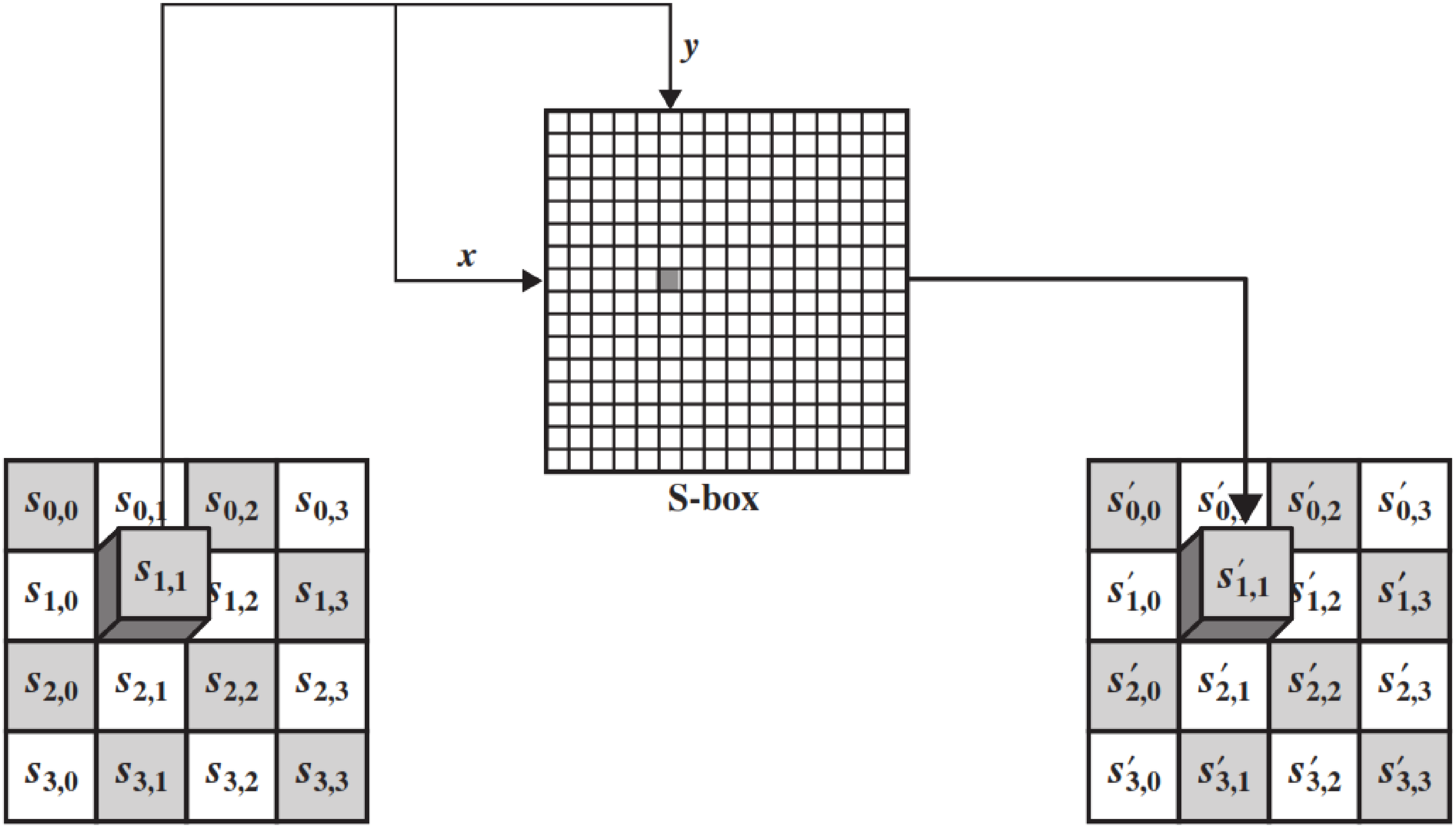
AES Round



State
SubBytes
S S S S S S S S S S S S S S S S
State
ShiftRows
State
MixColumns
M M M M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_0 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

SubByte Transformation

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



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.

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

MixColumns

- Each byte of a column is mapped into a new value that is a function of all four bytes in that column.
- Mix Columns performs matrix multiplication according to Galois field arithmetic

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

GF(2^8)

- Finite Field/ Galois fields : A field with finite number of elements
- Finite fields play a key role in cryptography
- The finite field with p^n elements is denoted GF(p^n) and is also called the **Galois field** of order p^n
- Rijndael (standardised as AES) uses the characteristic 2 finite field with 256 elements, which can also be called the Galois field GF(2^8)
- Byte b7b6b5b4b3b2b1b0 will have the representation as

  $b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$

- Therefore, 01010111 would have the representation as

  $x^6 + x^4 + x^2 + x + 1$

Addition on Bytes

- The sum of two elements is the polynomial with coefficients that are given by the sum modulo 2 (i.e., 1+1=0) of the coefficients of the two terms.
- Example: 57+83=?
  - $57= x^6+x^4+x^2+x+1$
  - $83= x^7+x+1$
  - $(x^6+x^4+x^2+x+1)+(x^7+x+1)=x^7+x^6+x^4+x^2$
  - $x^7+x^6+x^4+x^2 = D4$

Multiplication

- Multiplication is performed using a special polynomial called the irreducible polynomial.
- The modulus used for these operations is typically a specific irreducible polynomial of degree 8, which ensures that the resulting values remain within the field.
- Example: 57 • 83=?
  - $57= x^6+x^4+x^2+x+1$
  - $83= x^7+x+1$
  - $(x^6+x^4+x^2+x+1)\bullet(x^7+x+1)= x^{13}+x^{11}+x^9+x^8+x^7+ x^7+x^5+x^3+x^2+x+ x^6+x^4+x^2+x+1$

    AES uses arithmetic in the finite field GF(2^8) with irreducible (prime) polynomial which is $x^8+x^4+x^3+x+1$ (11B)
  - $x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1$ modulo $x^8+x^4+x^3+x+1$

  ….
  - x^7+x^6+1=C1

AddRoundKey

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

State

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

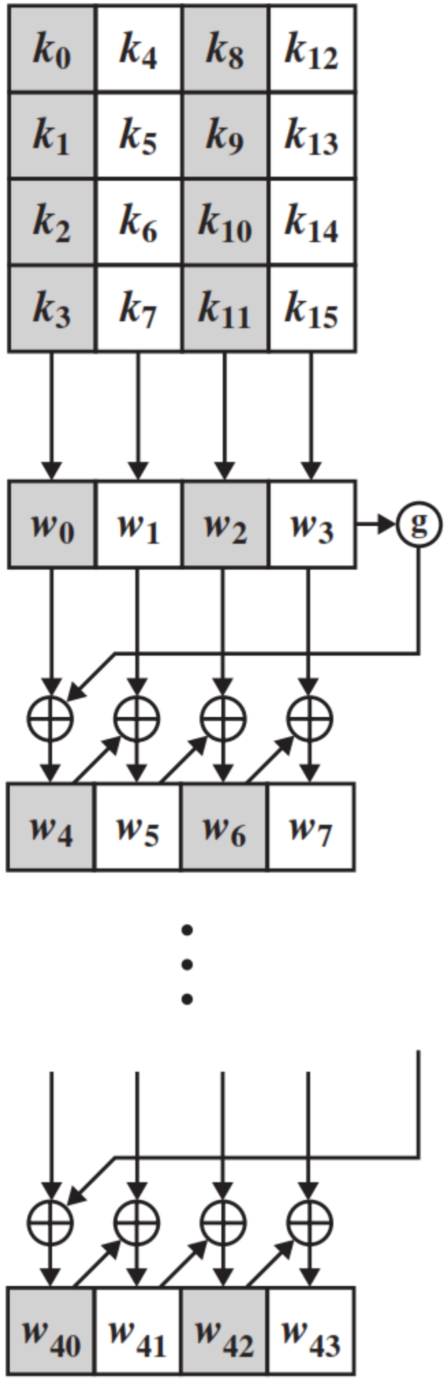
$\oplus$

Round Key

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

$=$

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6



AES Key Expansion

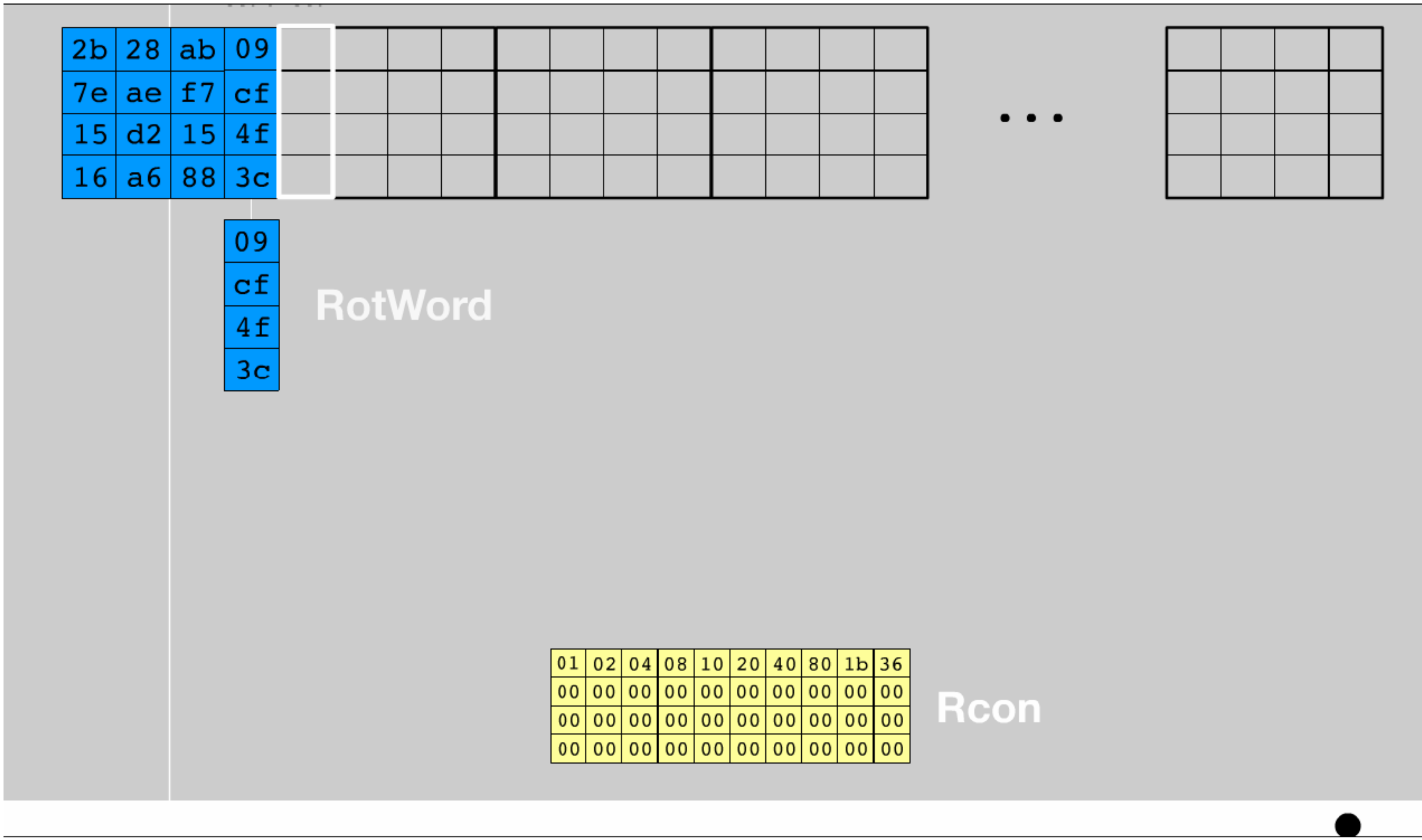
- The AES key expansion algorithm takes as input a four-word (16-byte) key and produces a linear array of **44 words** (176 bytes).
- Each added word **w[i]** depends on the immediately preceding word, w[i - 1].
- In three out of four cases, a simple XOR is used.

Key Expansion Example

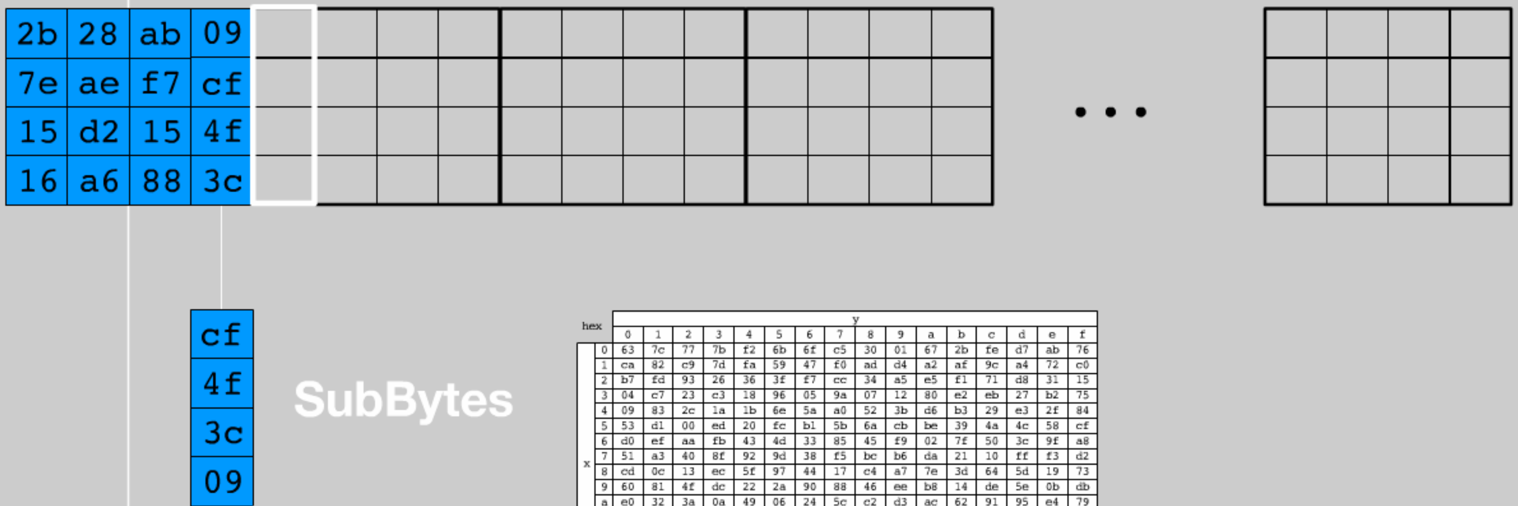
Plaintext:	0123456789abcdeffedcba9876543210
Key:	0f1571c947d9e8590cb7add6af7f6798
Ciphertext:	ff0b844a0853bf7c6934ab4364148fb9

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

Key Schedule Generation (For reference)



Key Schedule Generation (Cont.)



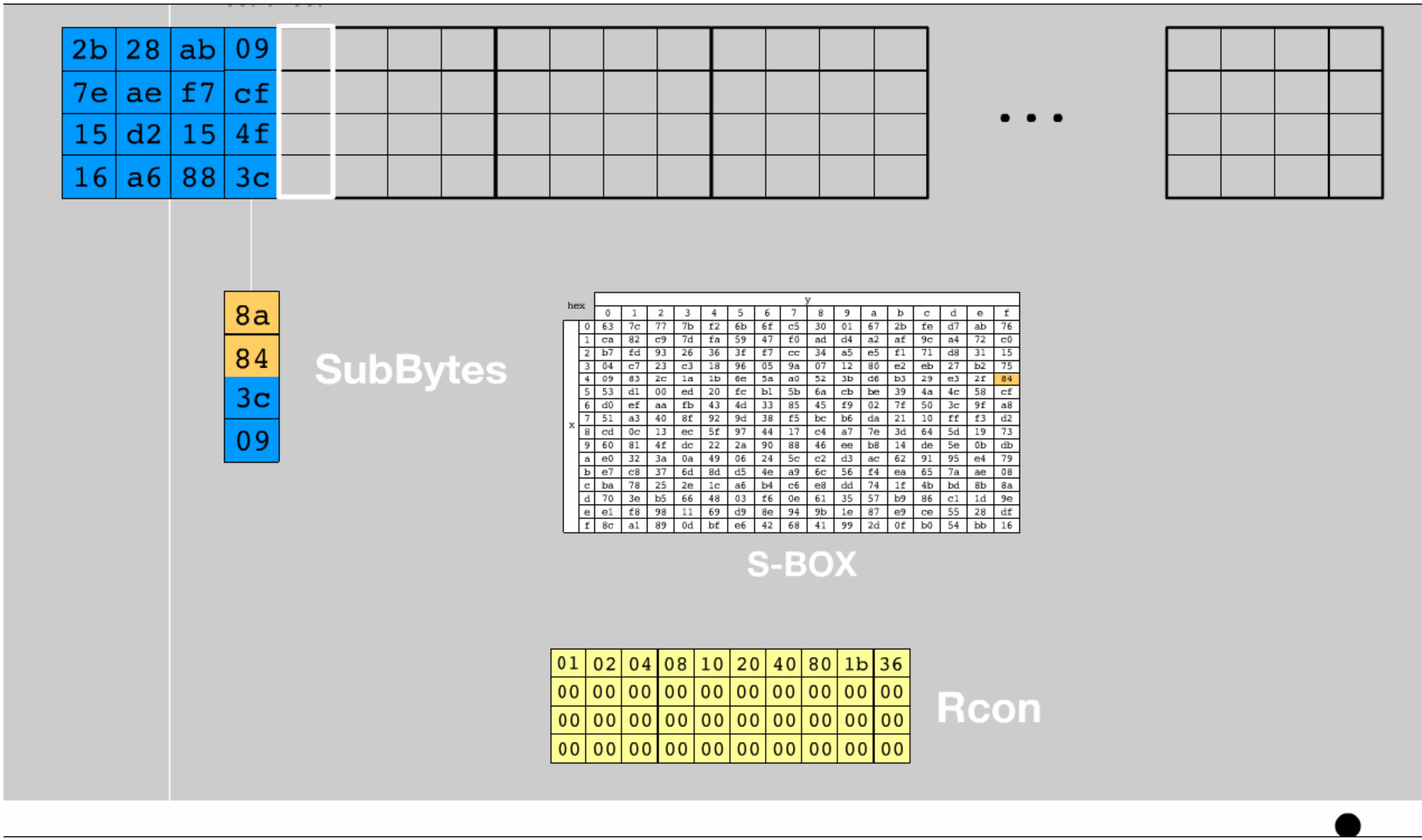
hex	y: 0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x: 0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00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

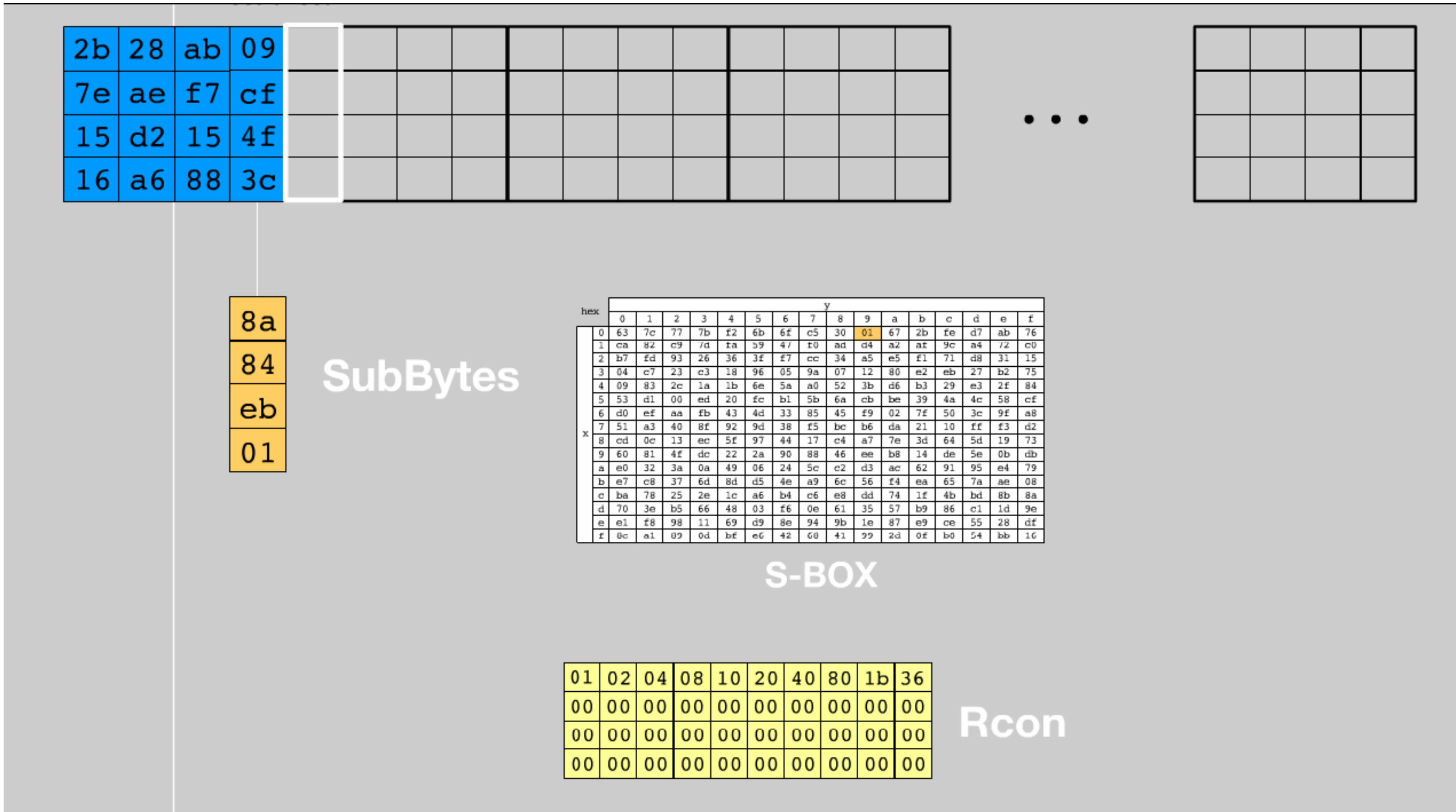
01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Rcon

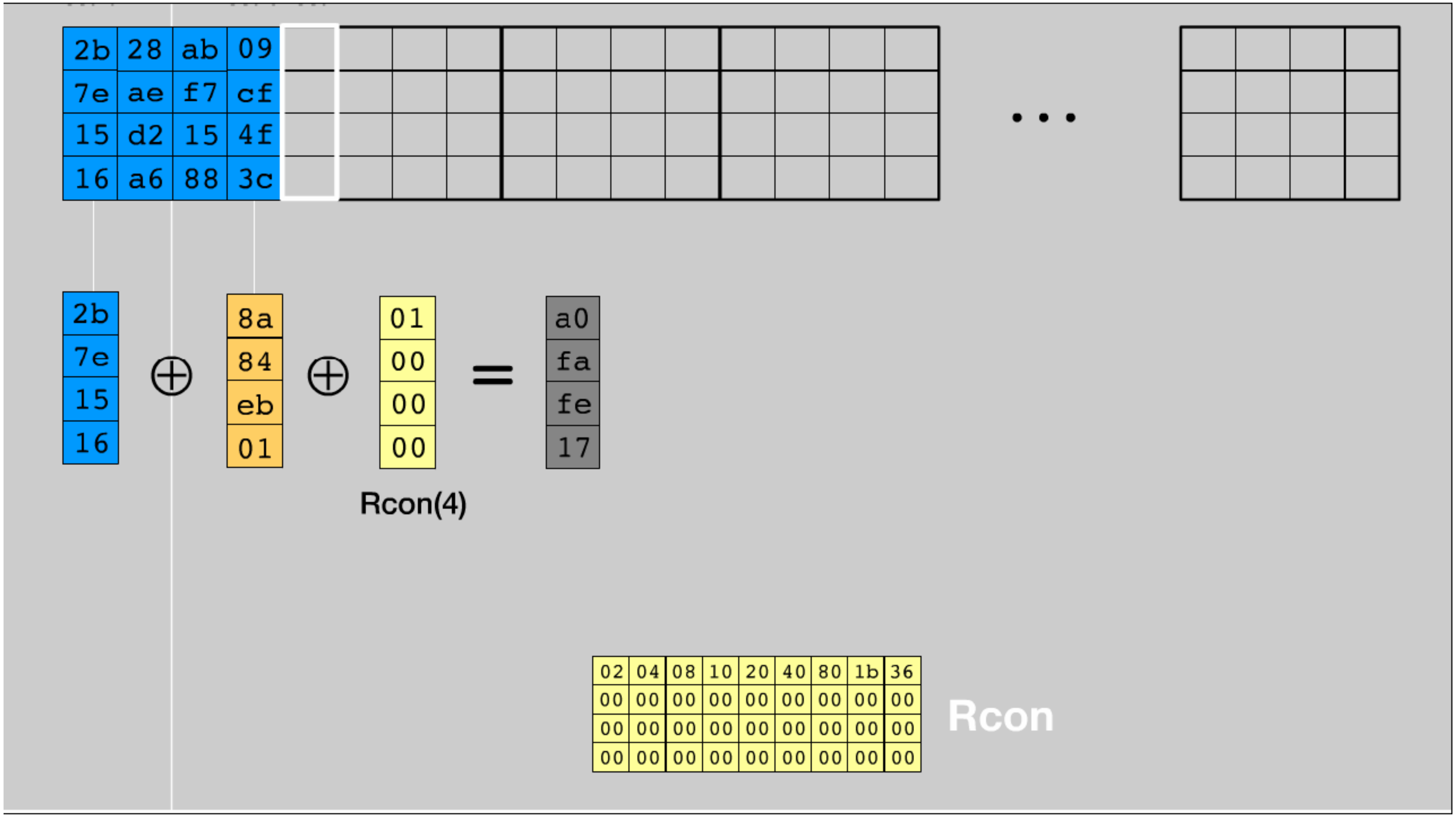
Key Schedule Generation (Cont.)



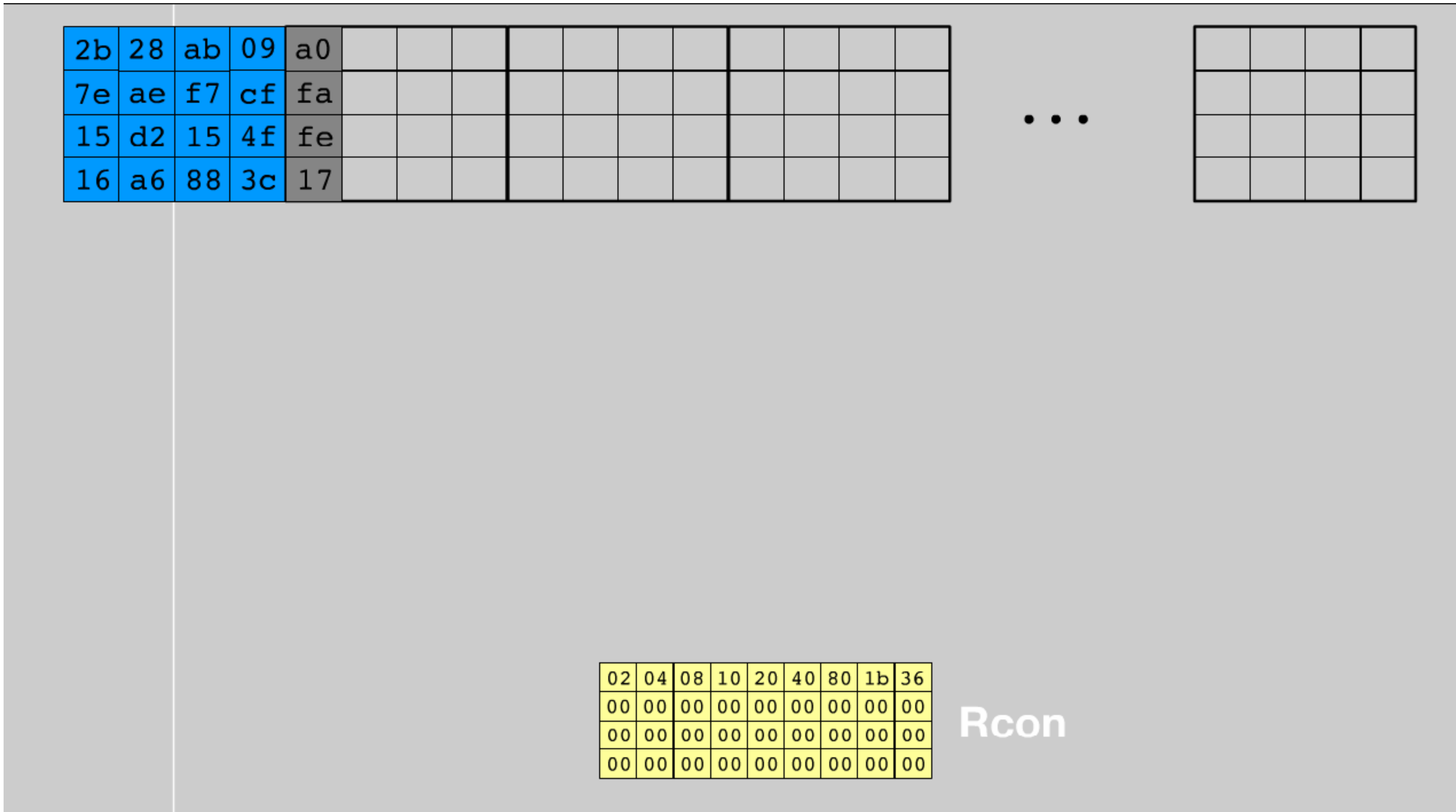
Key Schedule Generation (Cont.)



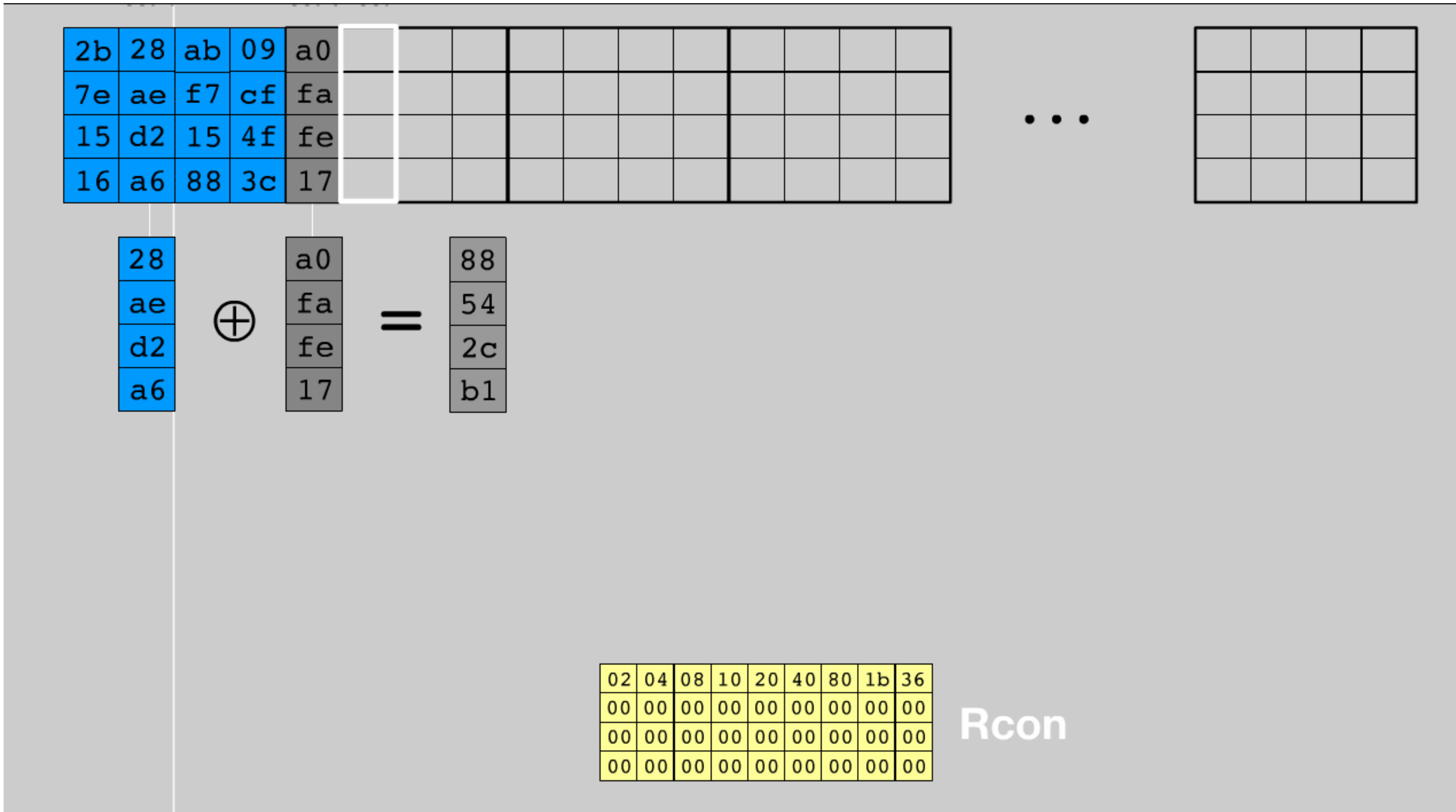
Key Schedule Generation (Cont.)



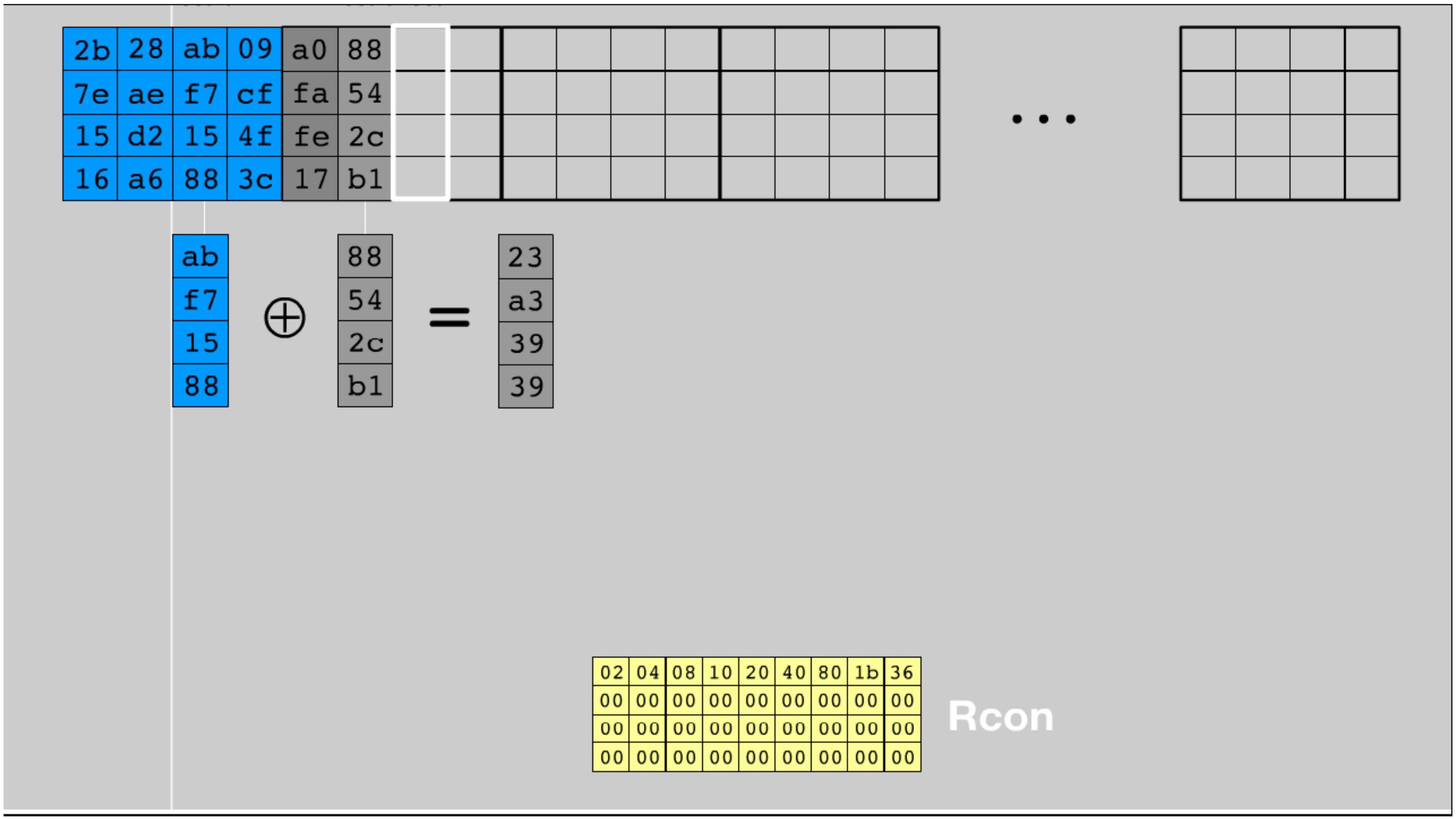
Key Schedule Generation (Cont.)



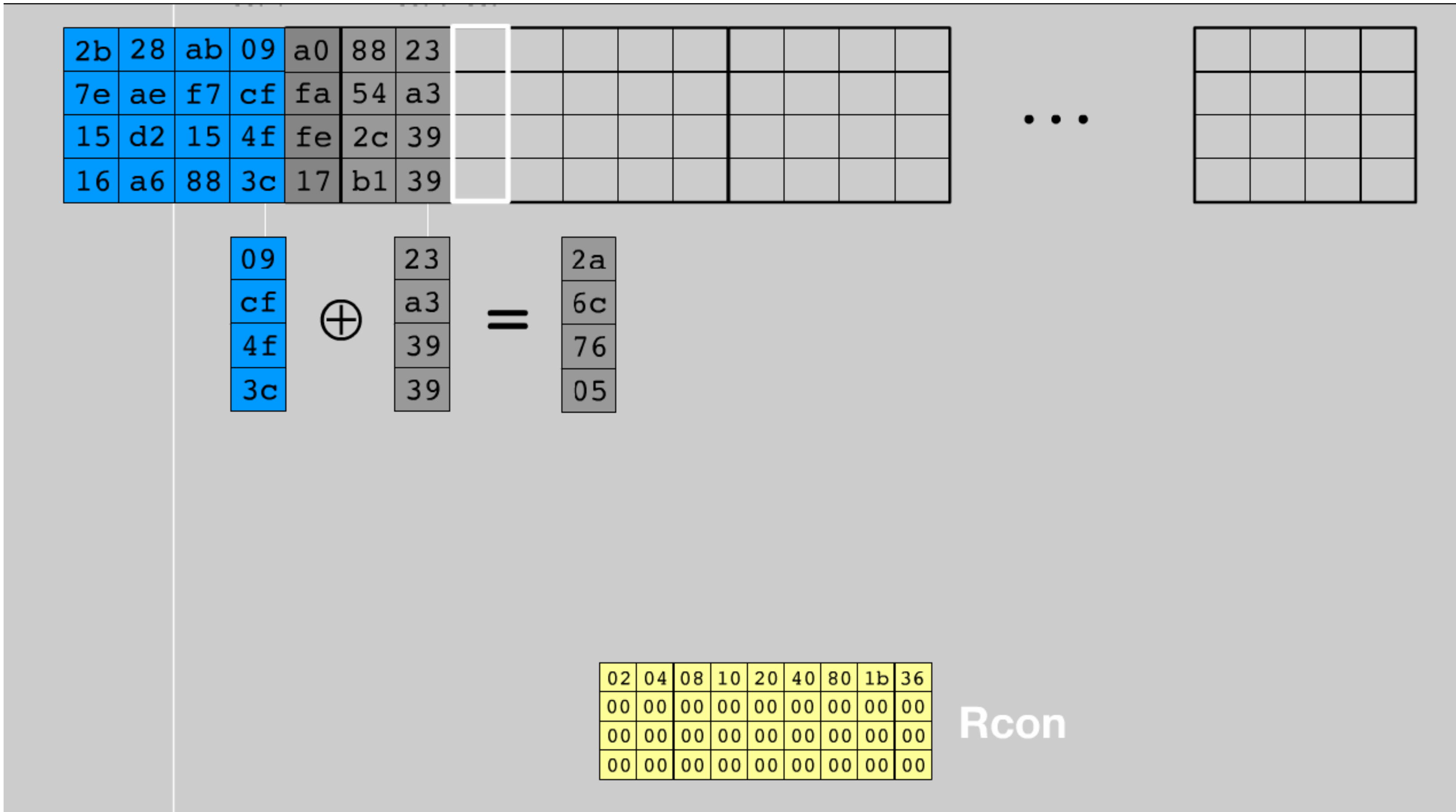
Key Schedule Generation (Cont.)



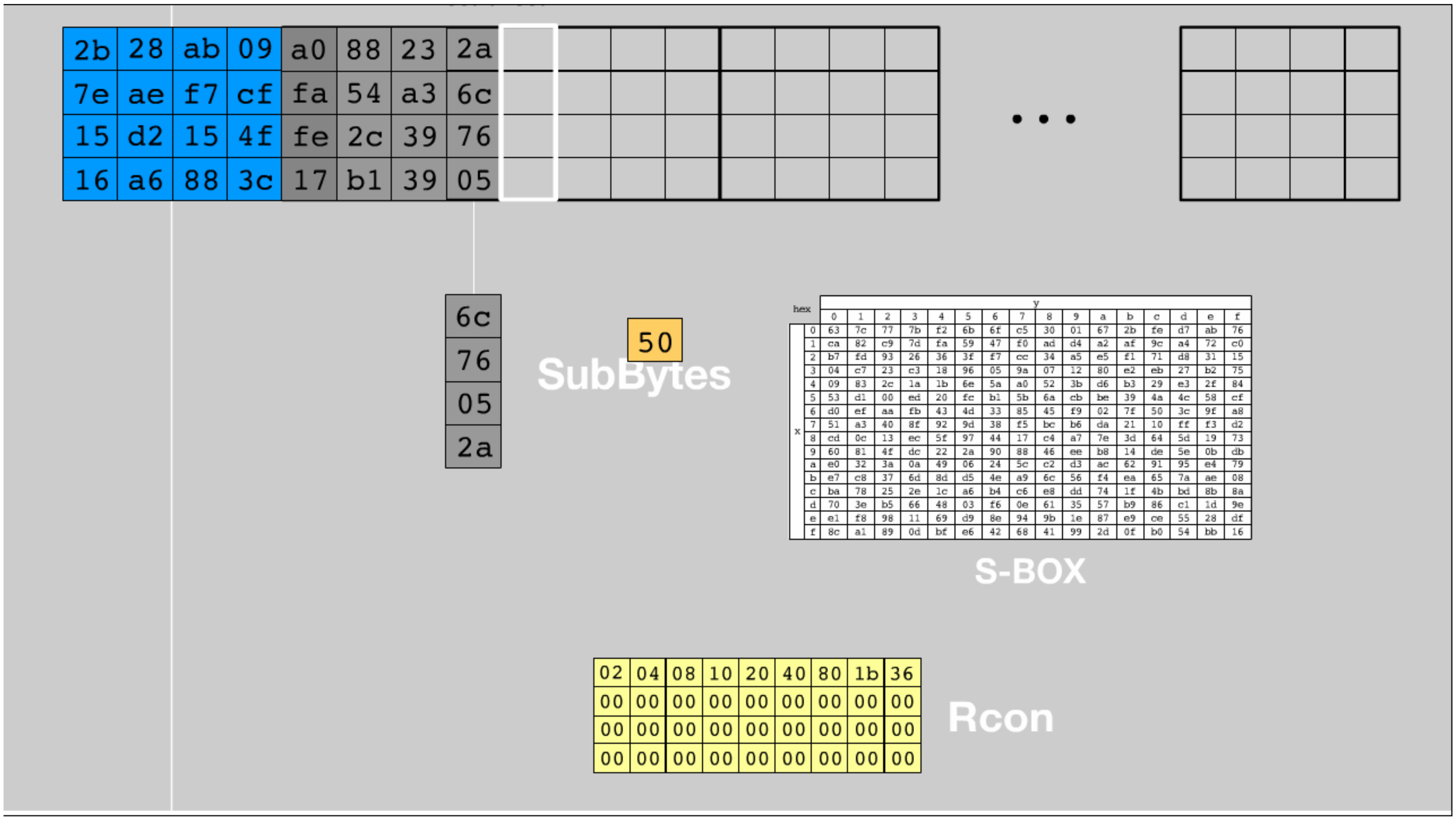
Key Schedule Generation (Cont.)



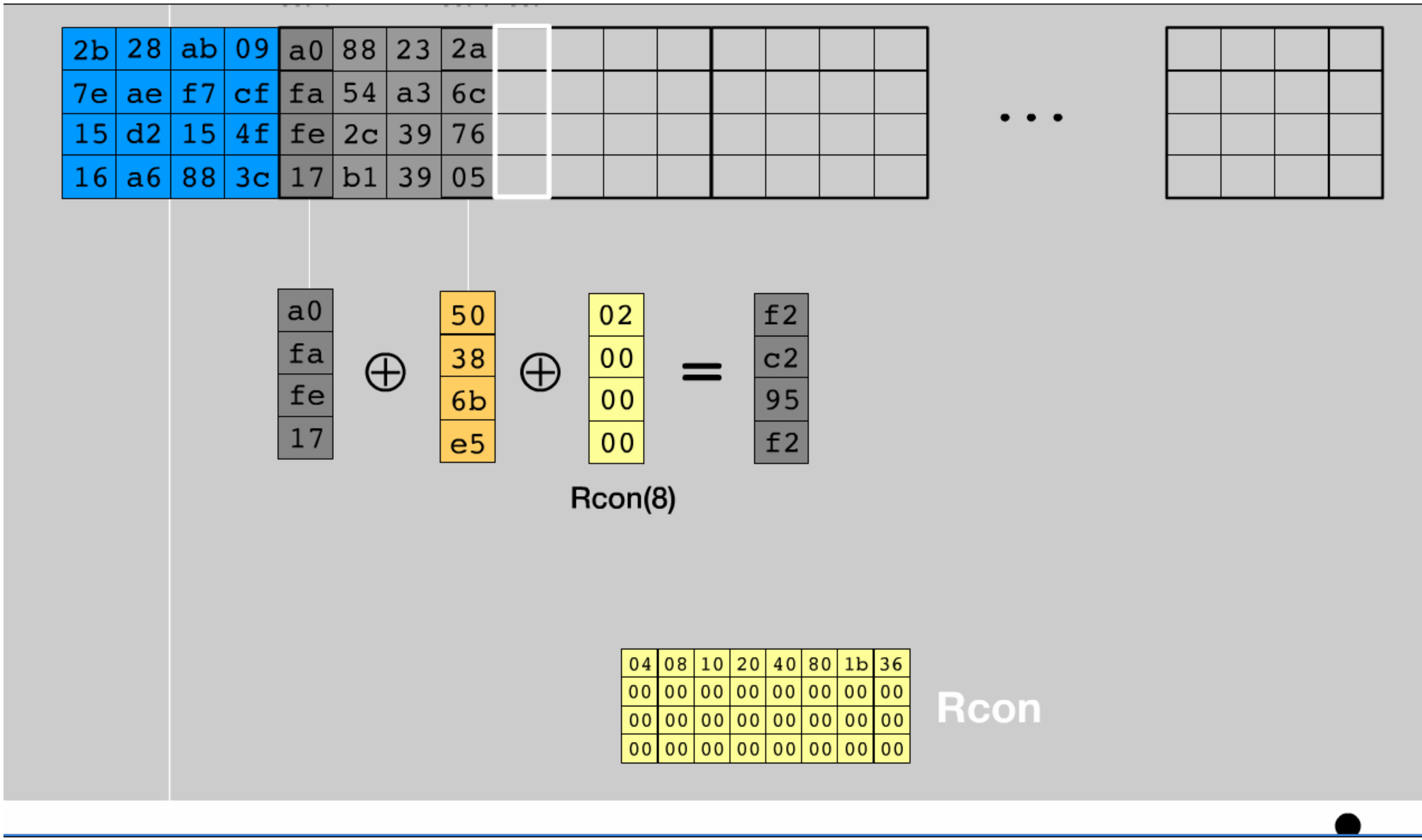
Key Schedule Generation (Cont.)



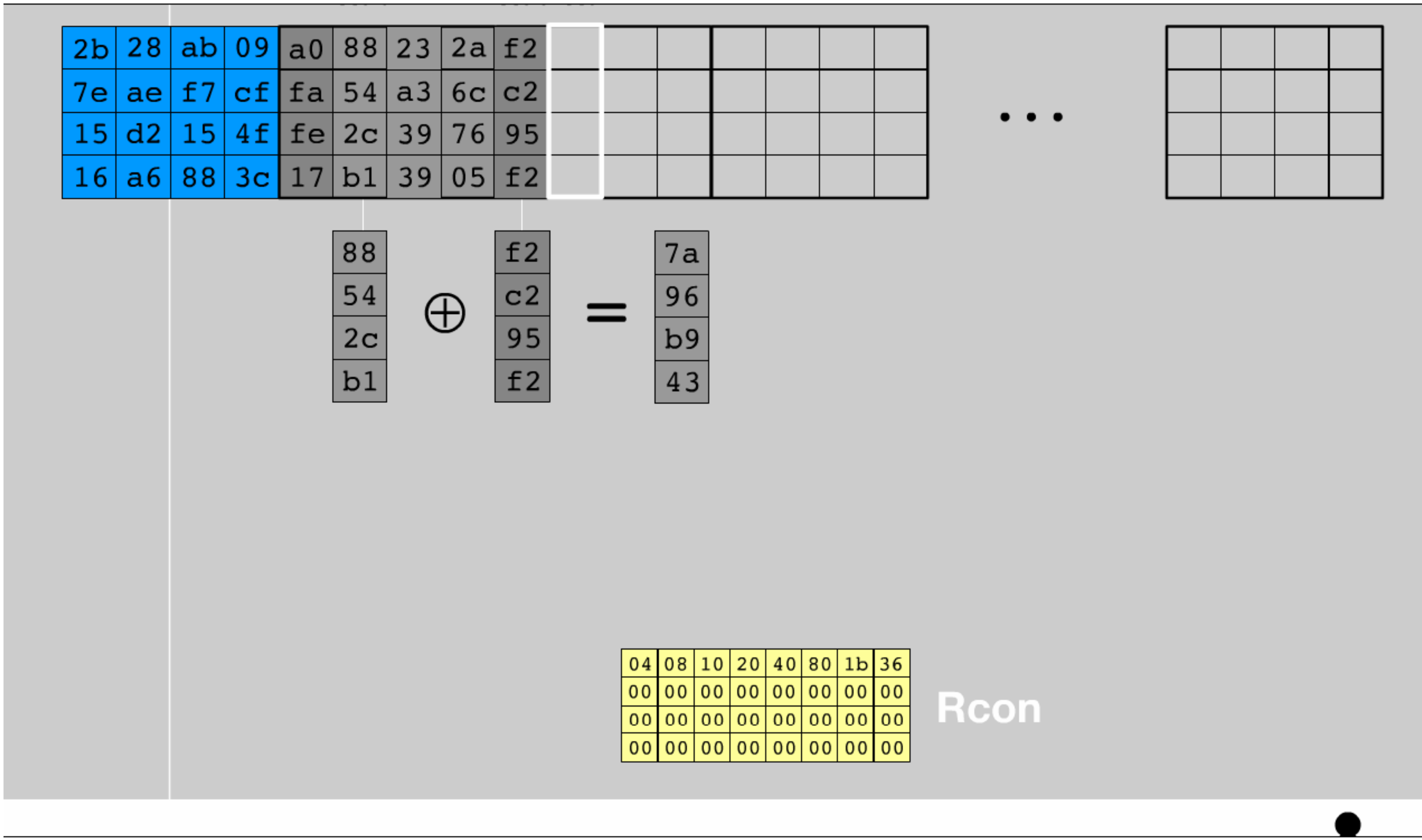
Key Schedule Generation (Cont.)



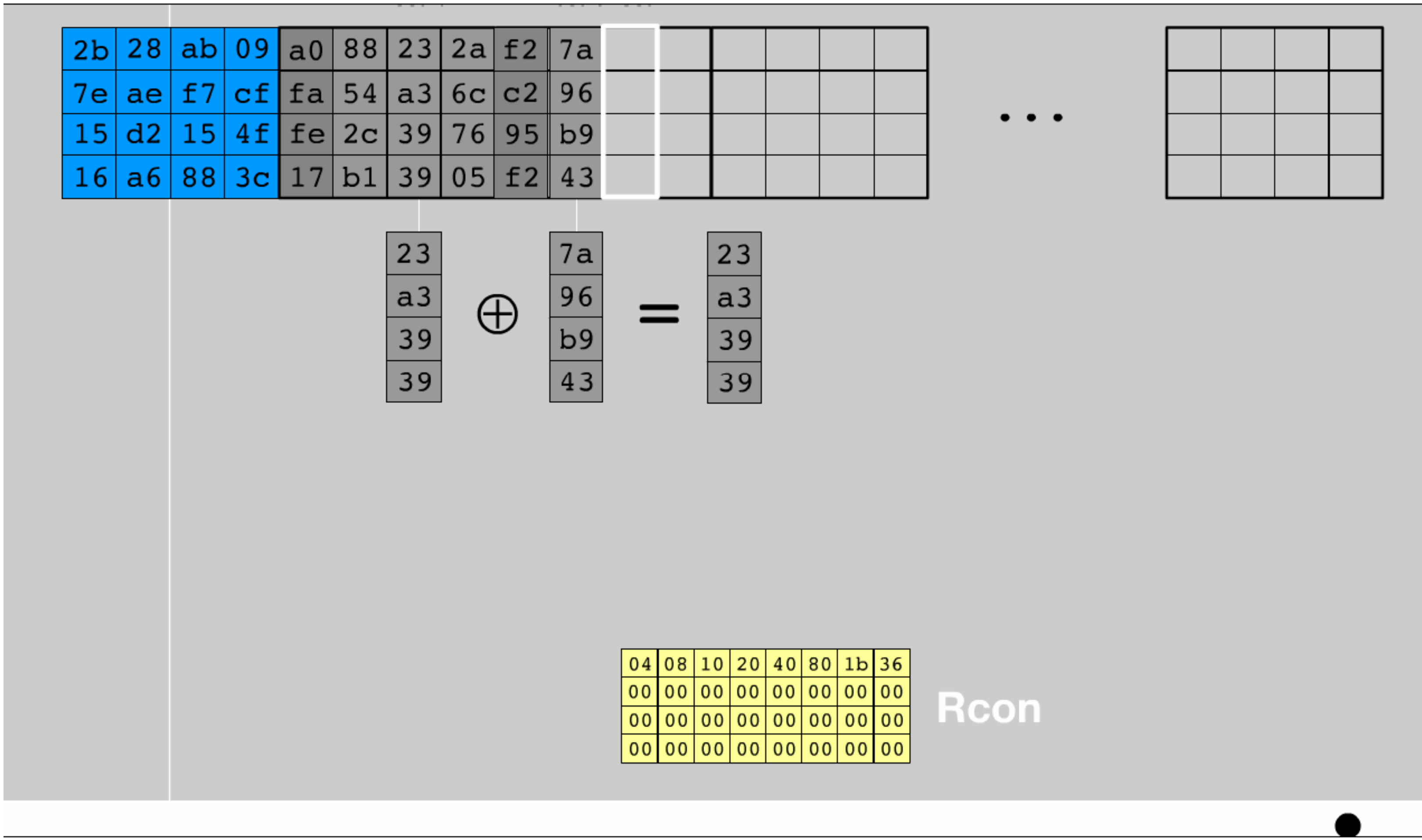
Key Schedule Generation (Cont.)



Key Schedule Generation (Cont.)



Key Schedule Generation (Cont.)



Key Schedule Generation (Cont.)

Cipher Key				Round key 1				Round key 2				Round key 3					Round key 10			
2b	28	ab	09	a0	88	23	2a	f2	7a	23	73	3d	47	1e	6d	…	d0	c9	e1	b6
7e	ae	f7	cf	fa	54	a3	6c	c2	96	a3	59	80	16	23	7a		14	ee	3f	63
15	d2	15	4f	fe	2c	39	76	95	b9	39	f6	47	fe	7e	88		f9	25	0c	0c
16	a6	88	3c	17	b1	39	05	f2	43	39	7f	7d	3e	44	3b		a8	89	c8	a6