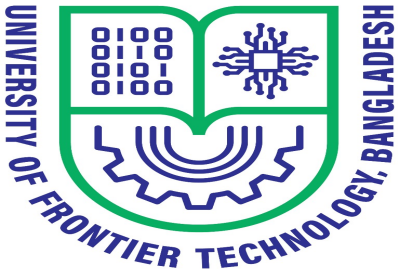


Topic: symmetric-key block cipher

Presented By:

Rakib Hossen

Assistant Professor

INTRODUCTION

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

DES is a block cipher, as shown in Figure 6.1.

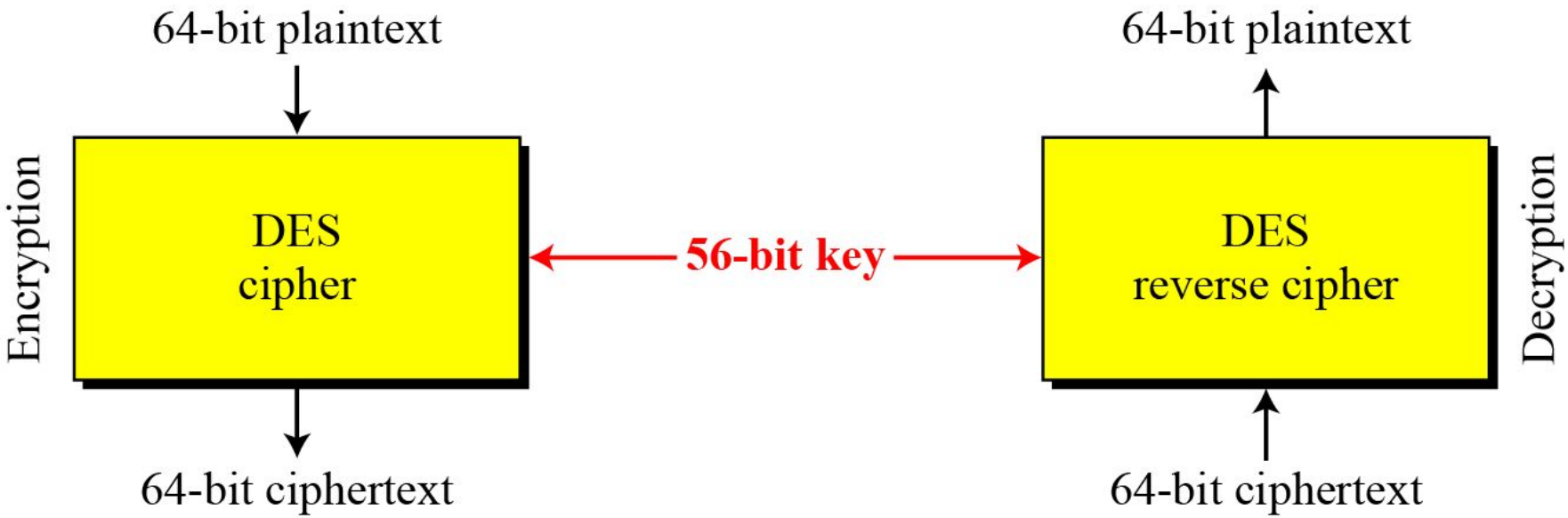


Figure 6.1 **Encryption and decryption with DES**

DES STRUCTURE

The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.

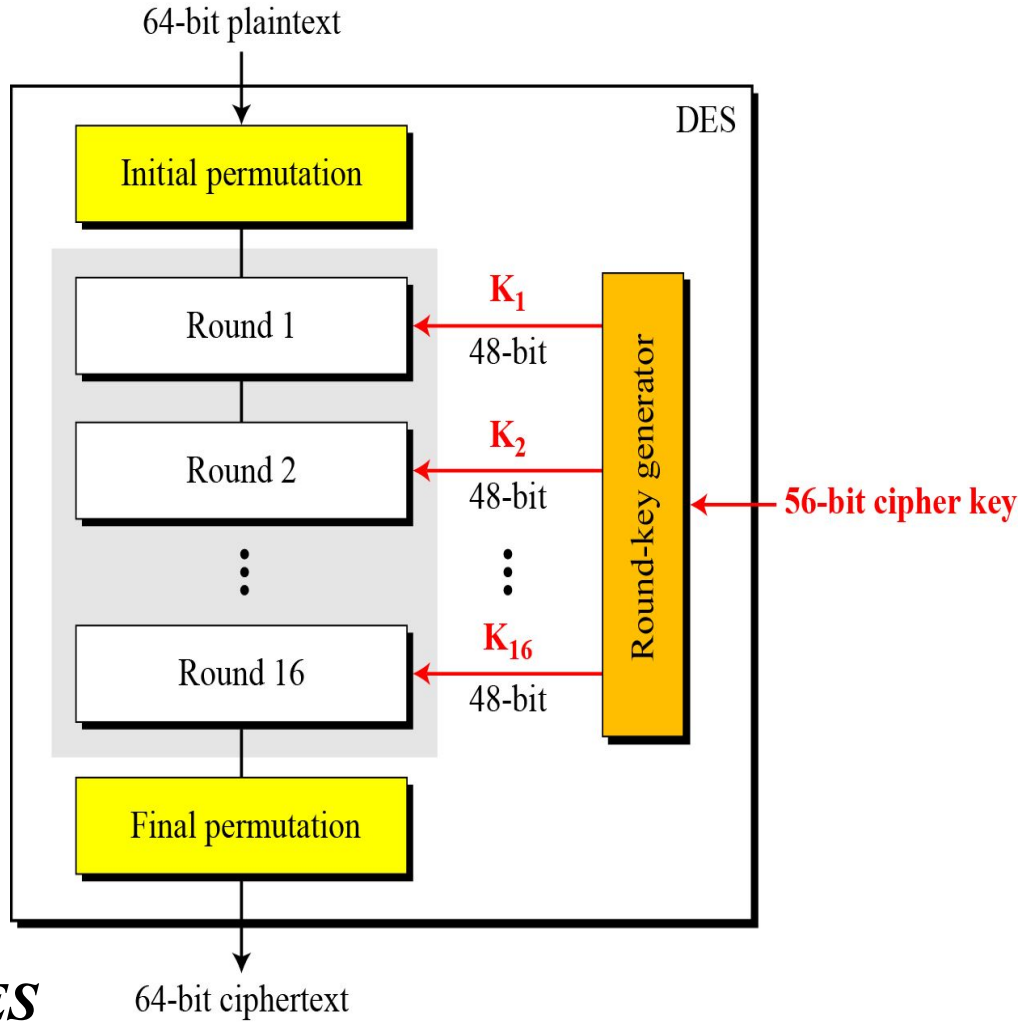
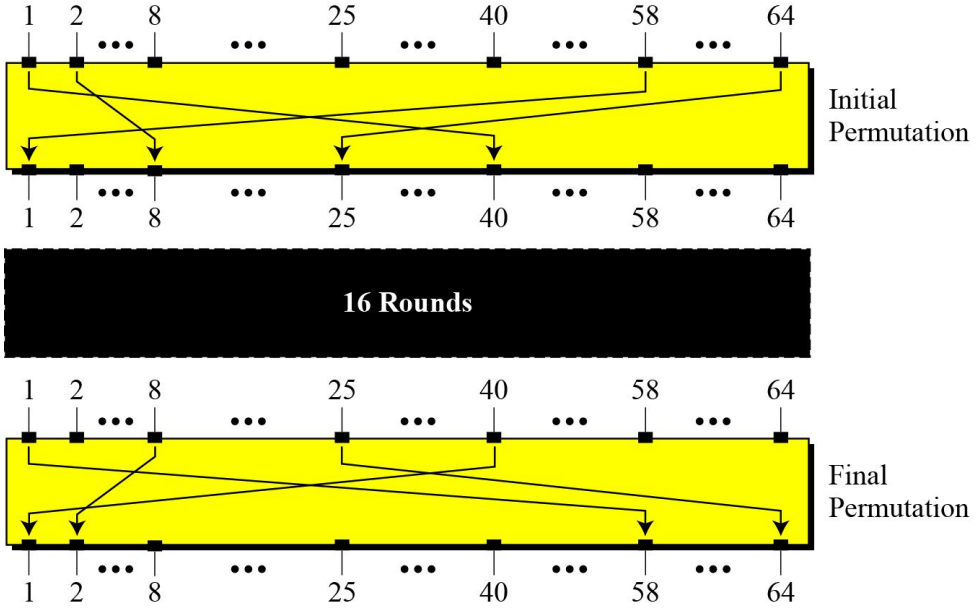


Figure 6.2 *General structure of DES*

Initial and Final Permutations



Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Final Permutation							
40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

Continued

Example 6.1

Find the output of the final permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

Solution

Only bit 25 and bit 63 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

Rounds

DES uses 16 rounds. Each round of DES is a Feistel cipher.

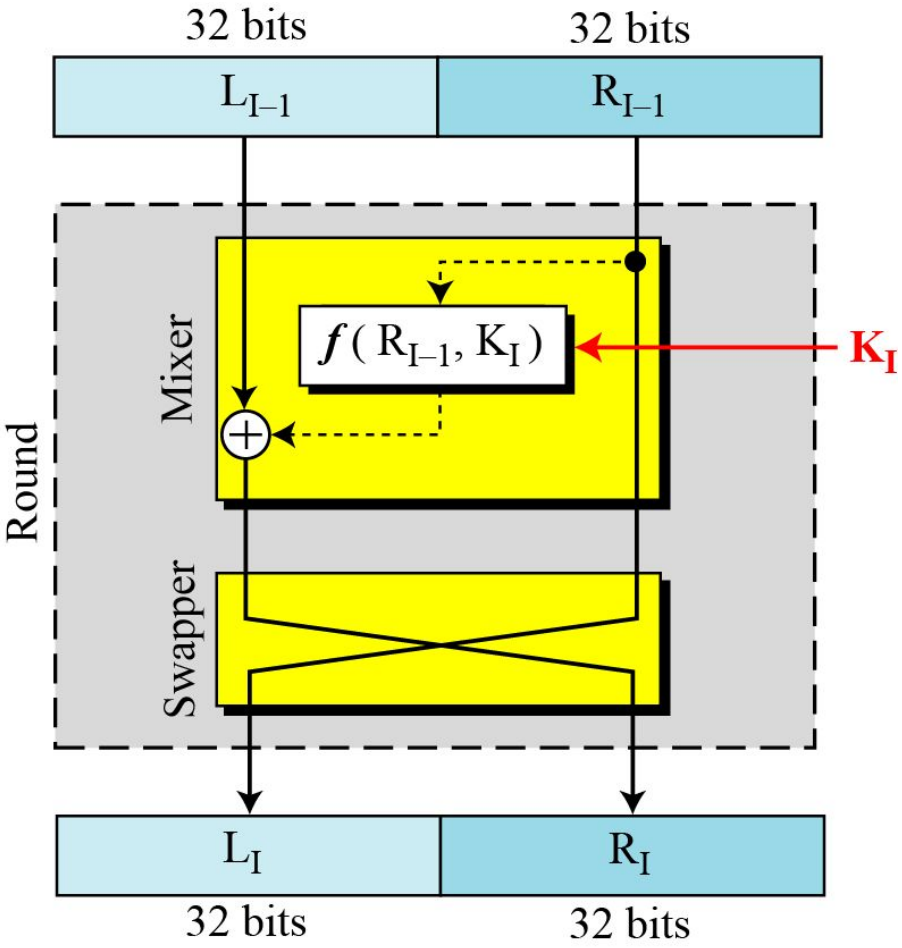


Figure 6.4
A round in DES (encryption site)

DES Function

The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.

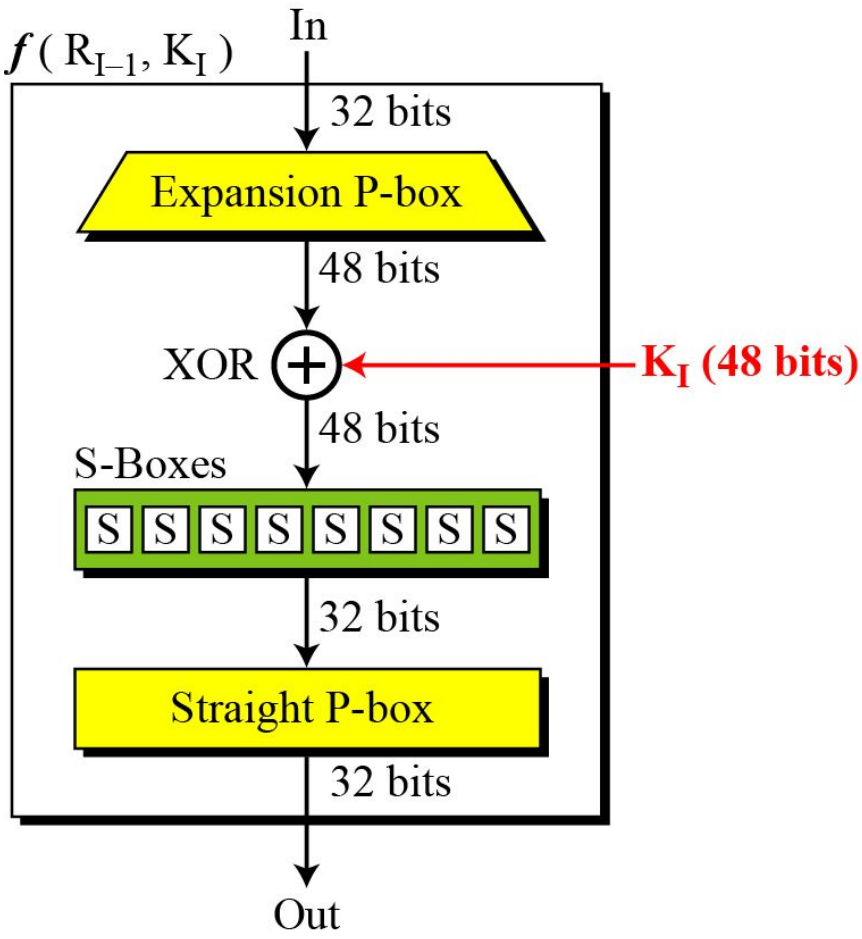
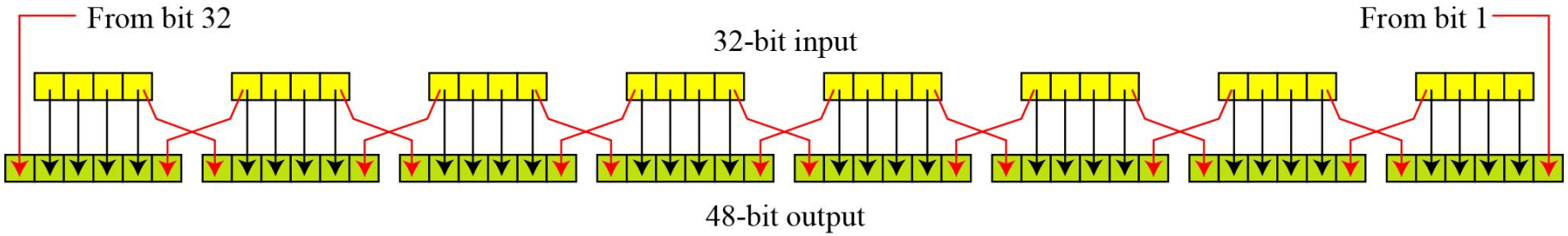


Figure 6.5
DES function

Expansion P-box

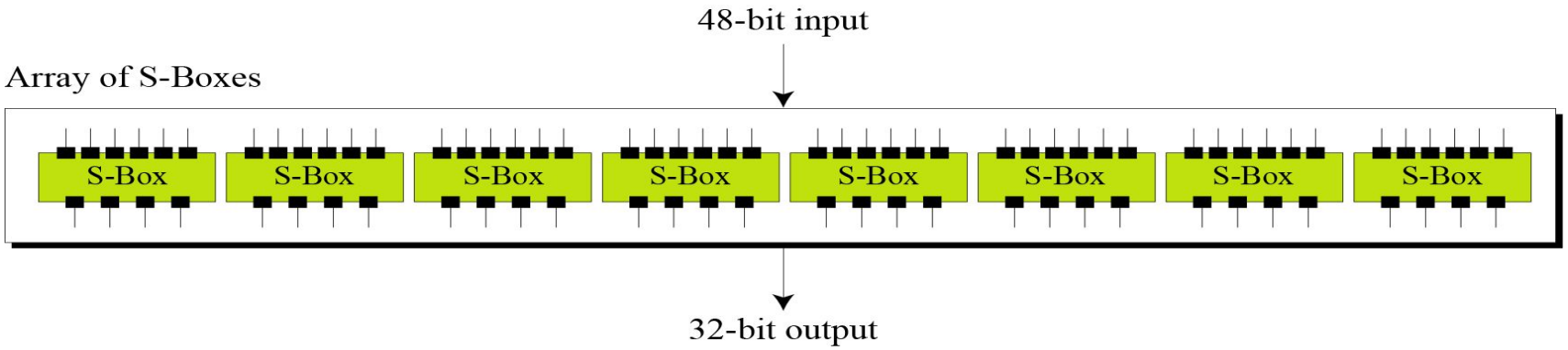
Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.

Figure 6.6 Expansion permutation

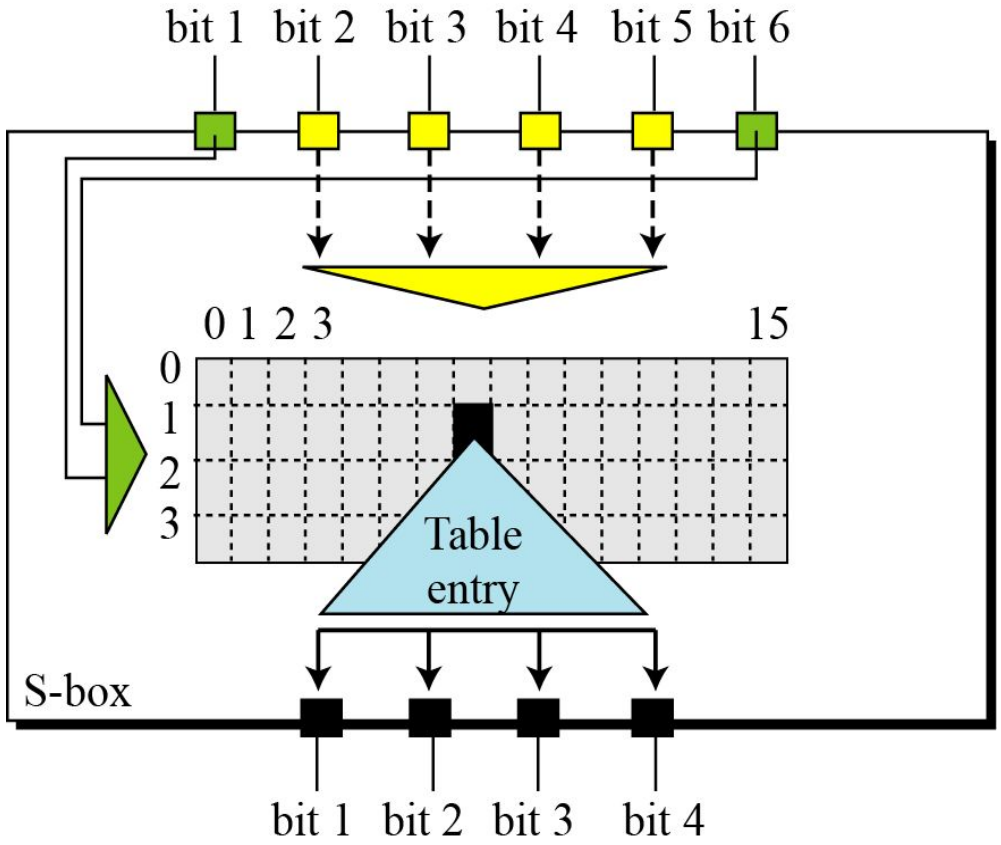


S-Boxes

The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output



S-Boxes



	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

Example 6.3

The input to S-box 1 is 100011. What is the output?

Solution

If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

Key generation

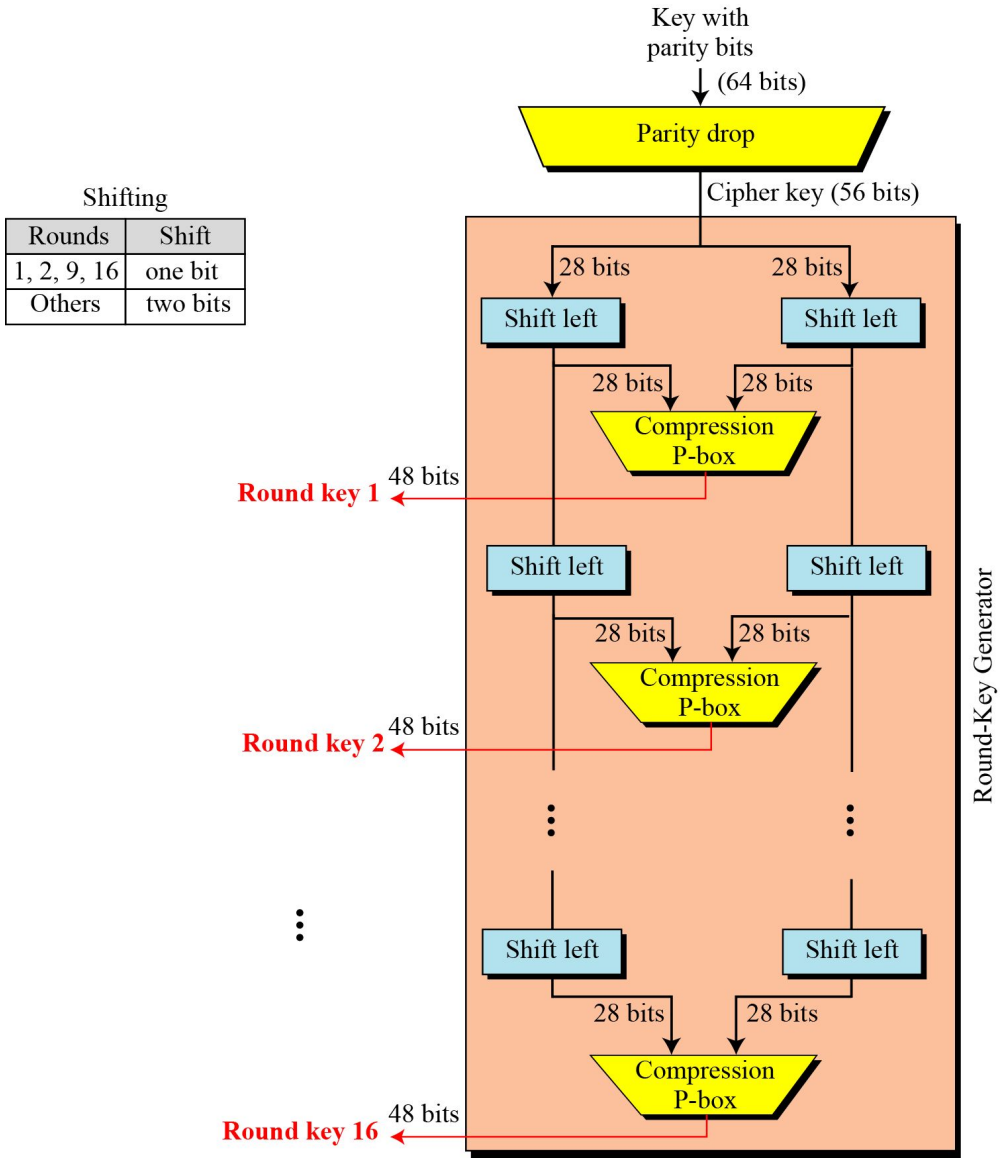


Figure 6.10

Key generation

Advanced Encryption Standard (AES)

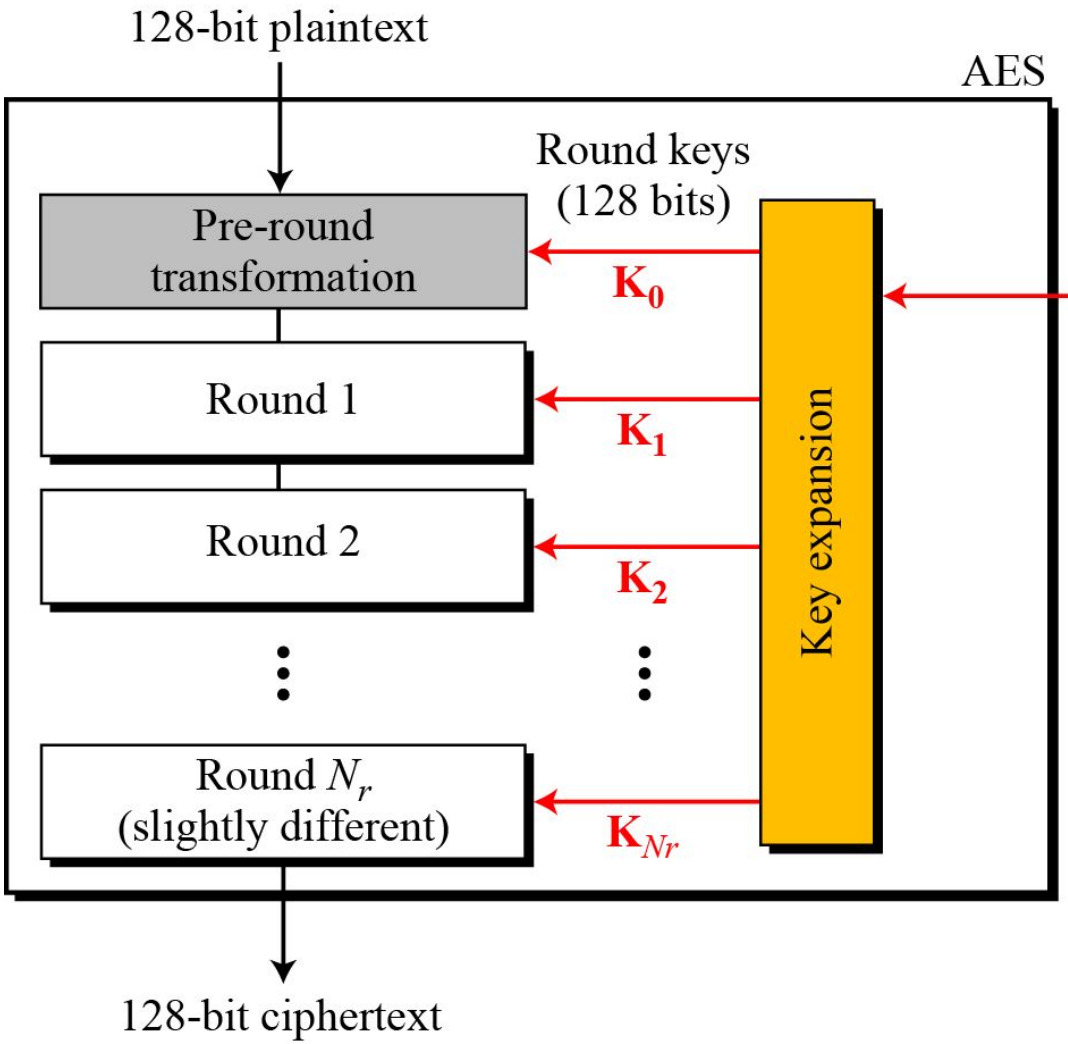
INTRODUCTION

- *The Advanced Encryption Standard (AES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001.*
- *AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the number of rounds.*

AES has defined three versions, with 10, 12, and 14 rounds.
Each version uses a different cipher key size (128, 192, or 256), but the round keys are always 128 bits.

Continue

Figure 7.1 *General design of AES encryption cipher*

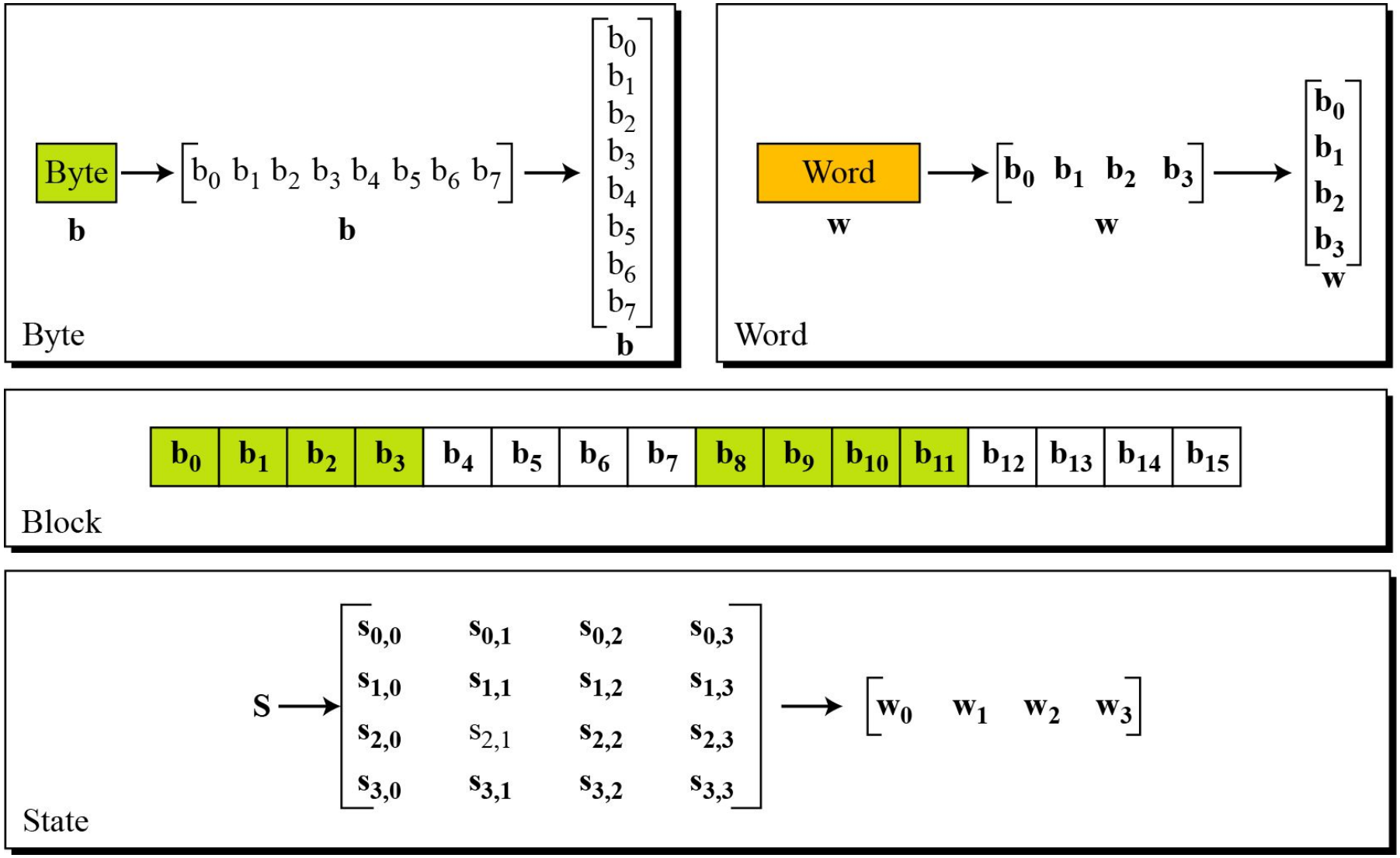


Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

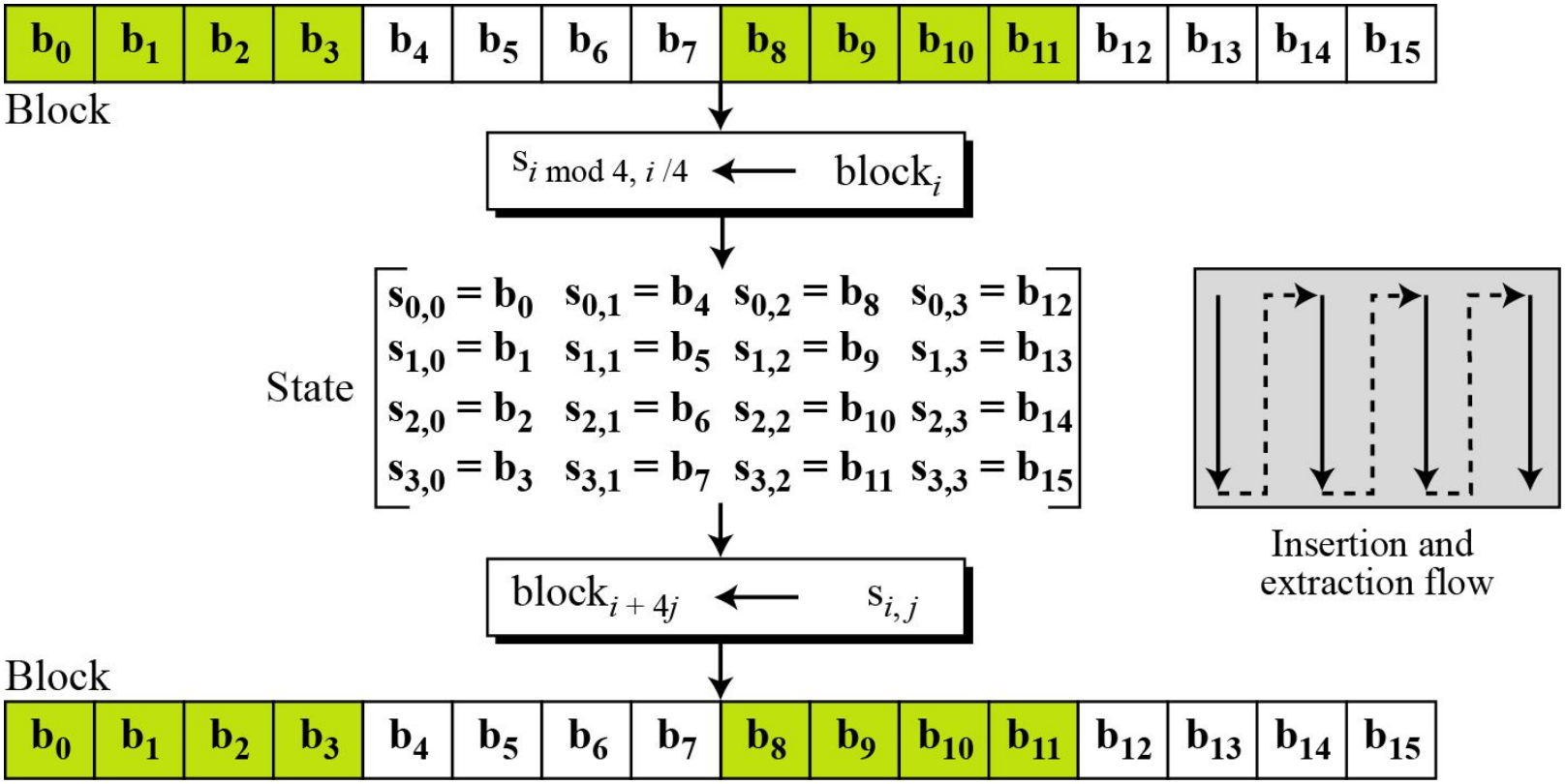
Data Units.

Figure 7.2 *Data units used in AES*



Continue

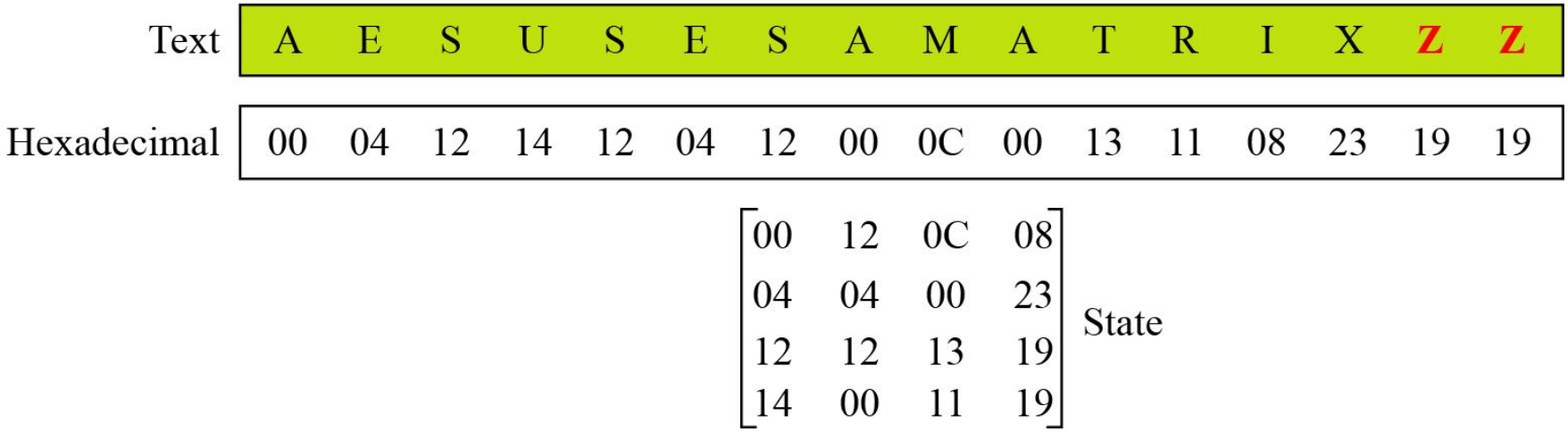
Figure 7.3 ***Block-to-state and state-to-block transformation***



Continue

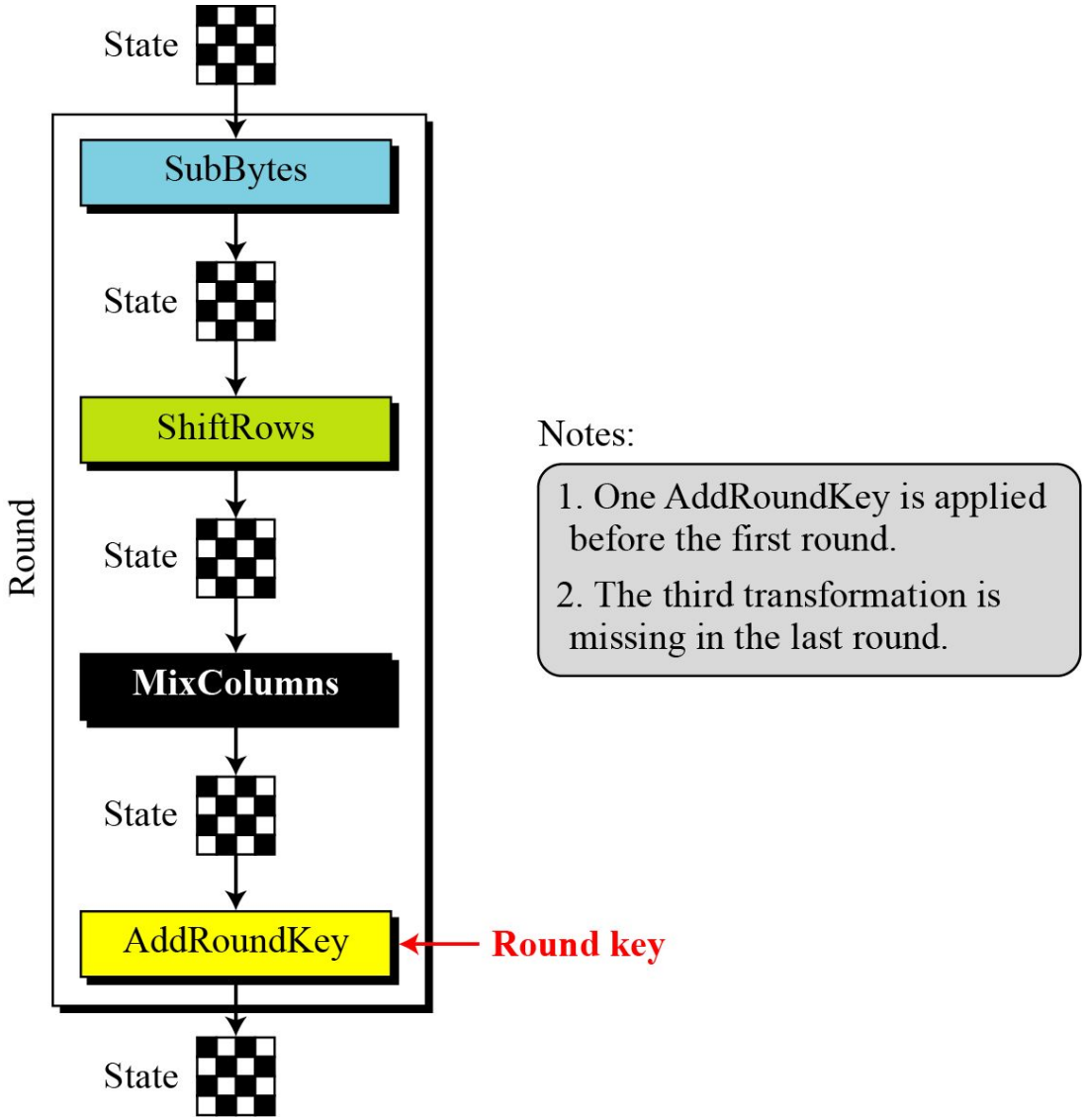
Example 7.1 *Continue*

Figure 7.4 *Changing plaintext to state*



Structure of Each Round

Figure 7.5 *Structure of each round at the encryption site*



Substitution

AES, like DES, uses substitution. AES uses two invertible transformations.

SubBytes

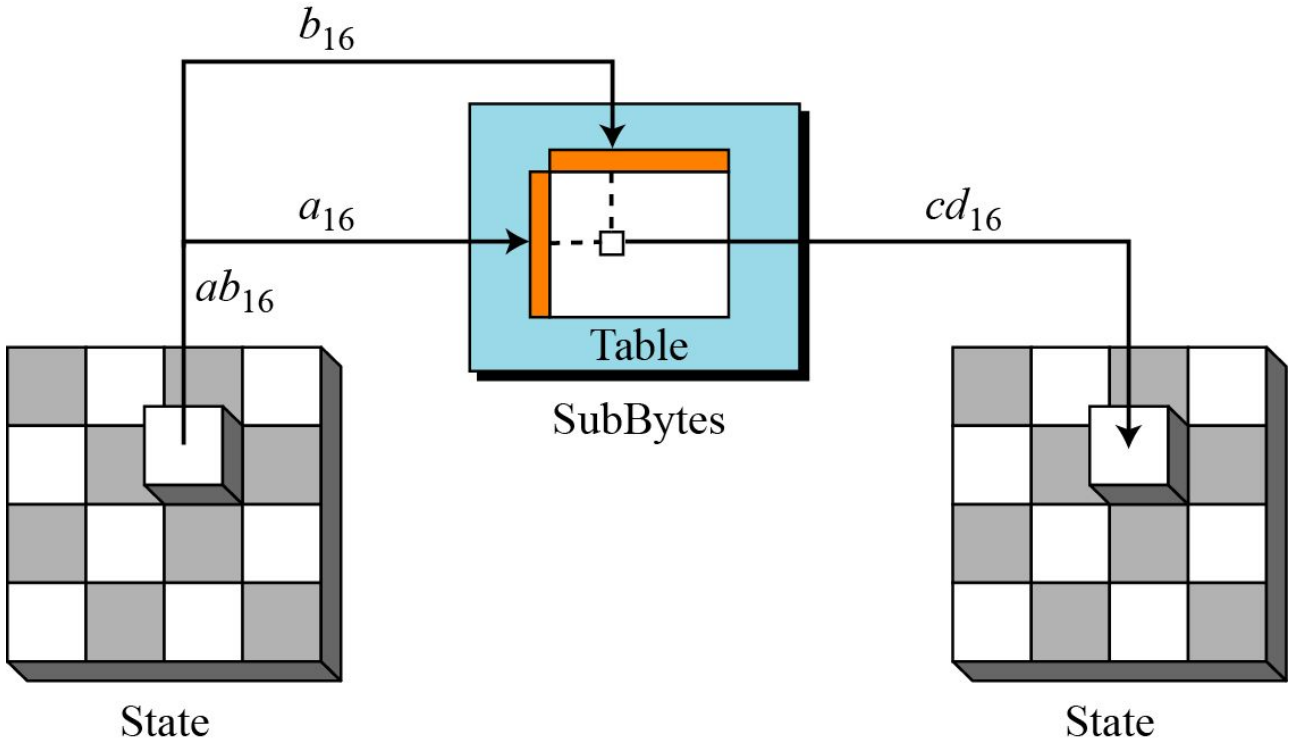
The first transformation, SubBytes, is used at the encryption site. To substitute a byte, we interpret the byte as two hexadecimal digits.

Note

The SubBytes operation involves 16 independent byte-to-byte transformations.

Continue

Figure 7.6 ***SubBytes transformation***



Continue

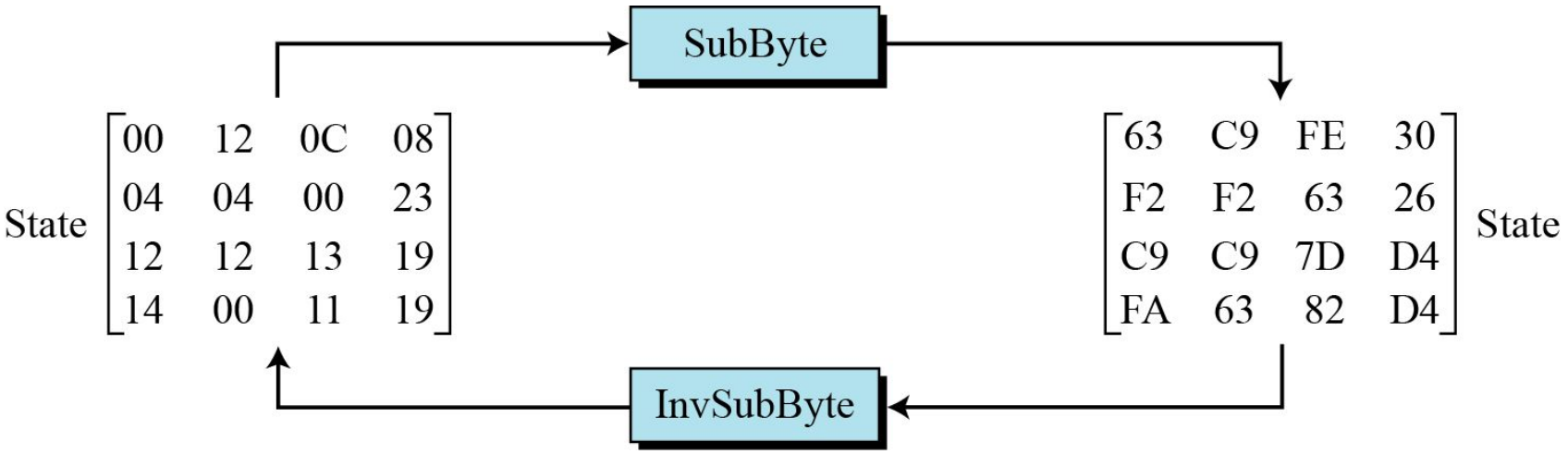
Table 7.1 *SubBytes transformation table (continued)*

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

Example 7.2

Figure 7.7 shows how a state is transformed using the SubBytes transformation. The figure also shows that the InvSubBytes transformation creates the original one. Note that if the two bytes have the same values, their transformation is also the same.

Figure 7.7 ***SubBytes transformation for Example 7.2***



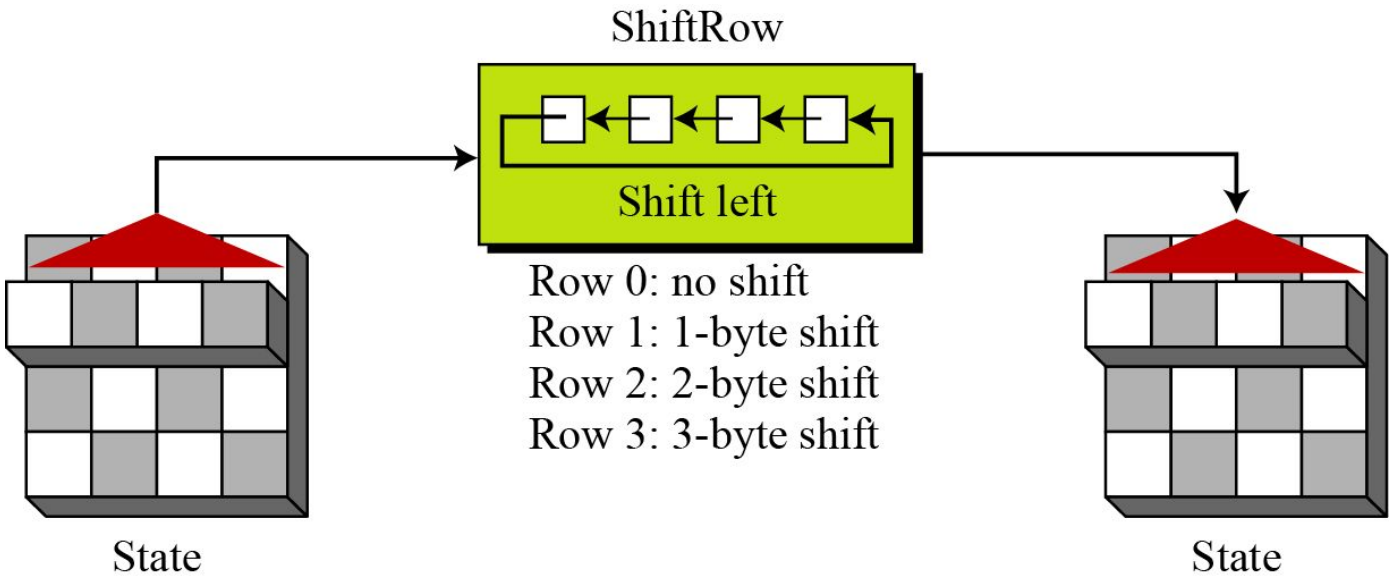
Permutation

Another transformation found in a round is shifting, which permutes the bytes.

ShiftRows

In the encryption, the transformation is called ShiftRows.

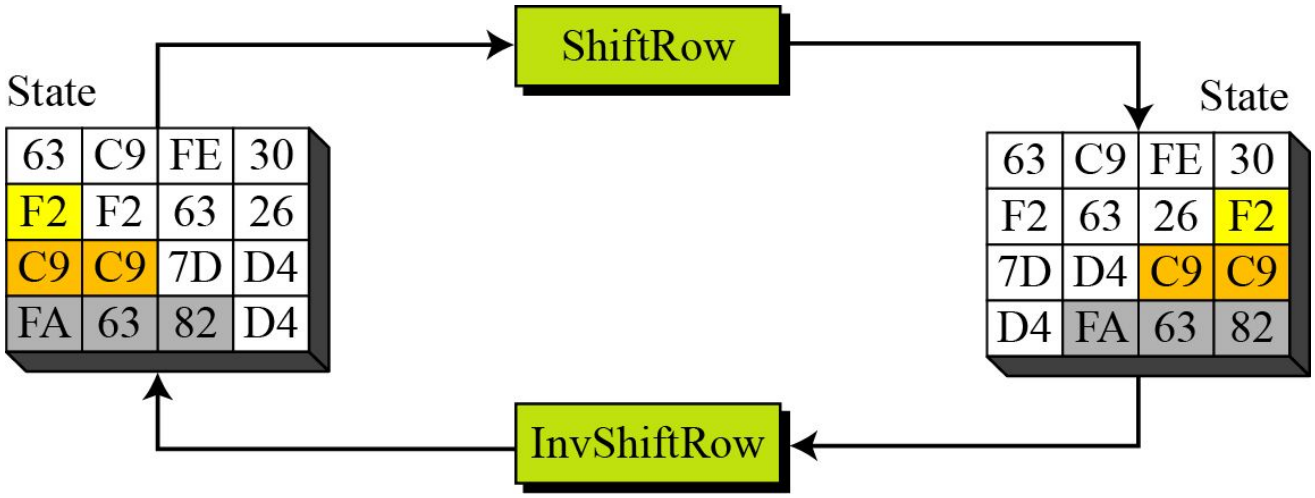
Figure 7.9 ***ShiftRows transformation***



Example 7.4

Figure 7.10 shows how a state is transformed using ShiftRows transformation. The figure also shows that InvShiftRows transformation creates the original state.

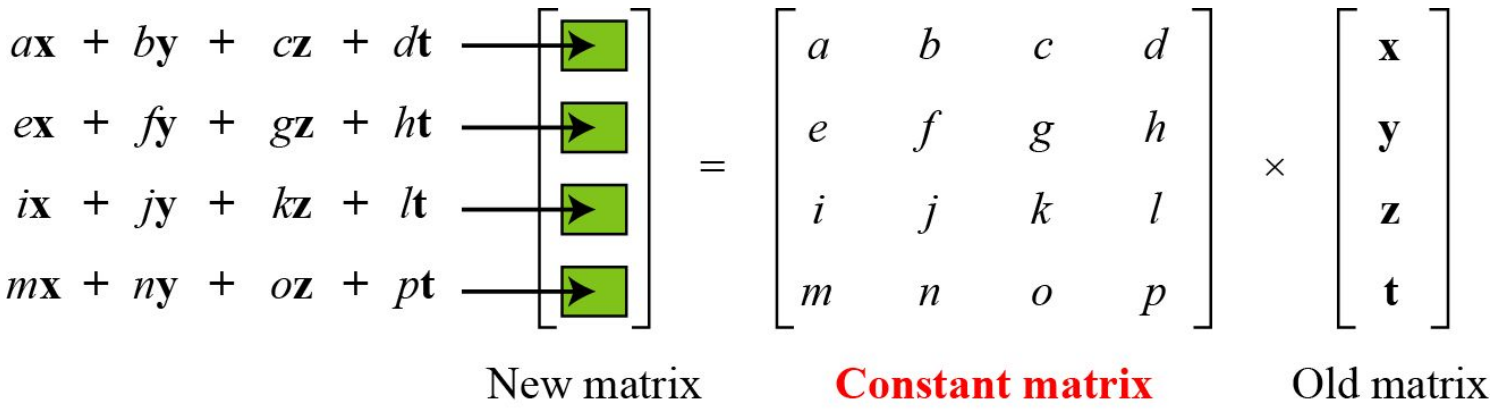
Figure 7.10 ***ShiftRows transformation in Example 7.4***



Mixing

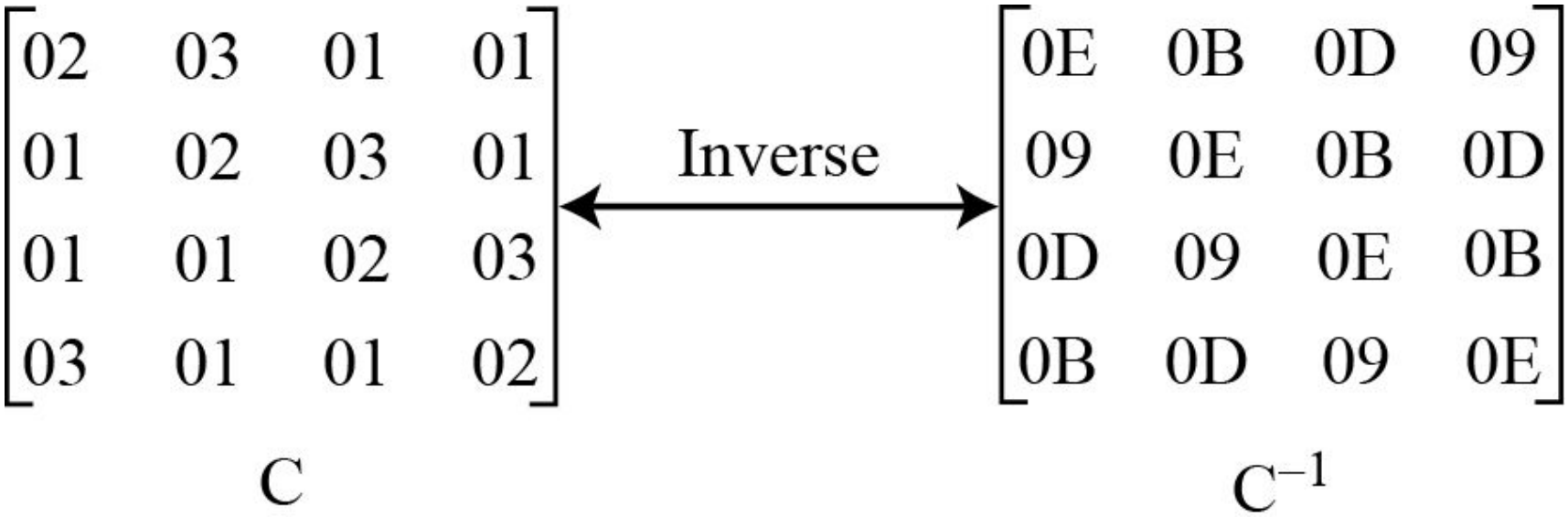
We need an interbyte transformation that changes the bits inside a byte, based on the bits inside the neighboring bytes. We need to mix bytes to provide diffusion at the bit level.

Figure 7.11 ***Mixing bytes using matrix multiplication***



Continue

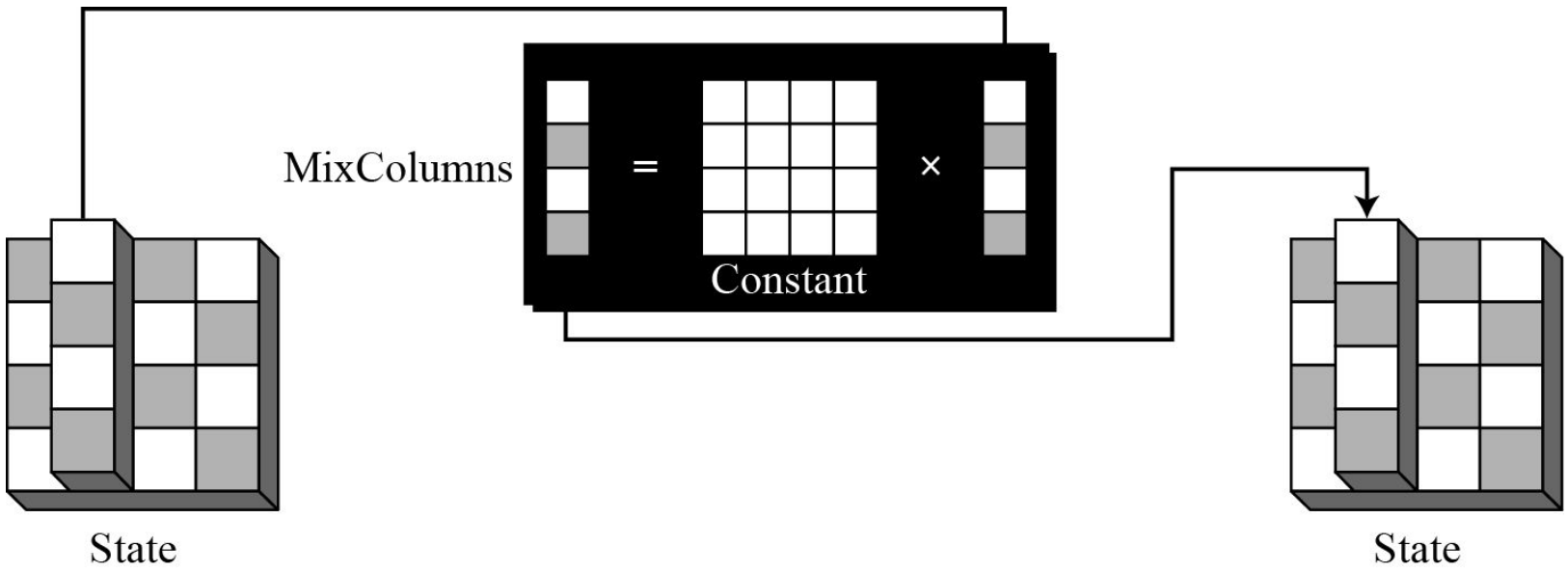
Figure 7.12 *Constant matrices used by MixColumns and InvMixColumns*



MixColumns

The MixColumns transformation operates at the column level; it transforms each column of the state to a new column.

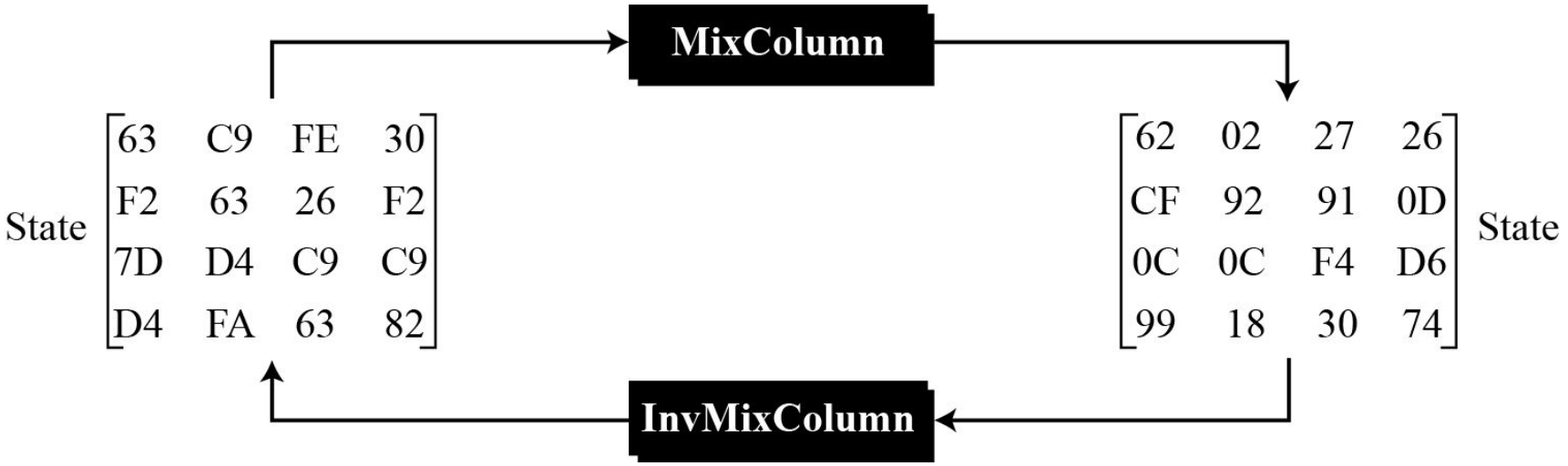
Figure 7.13 *MixColumns transformation*



Example 7.5

Figure 7.14 shows how a state is transformed using the MixColumns transformation. The figure also shows that the InvMixColumns transformation creates the original one.

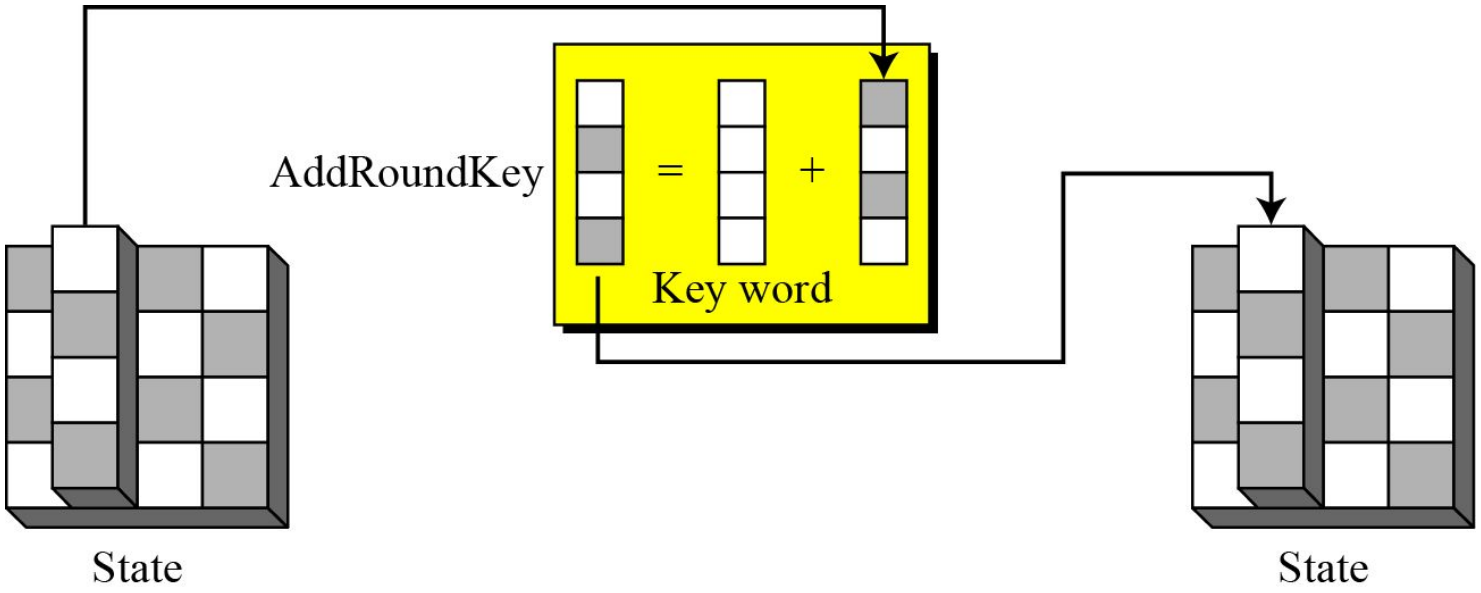
Figure 7.14 *The MixColumns transformation in Example 7.5*



Key Adding

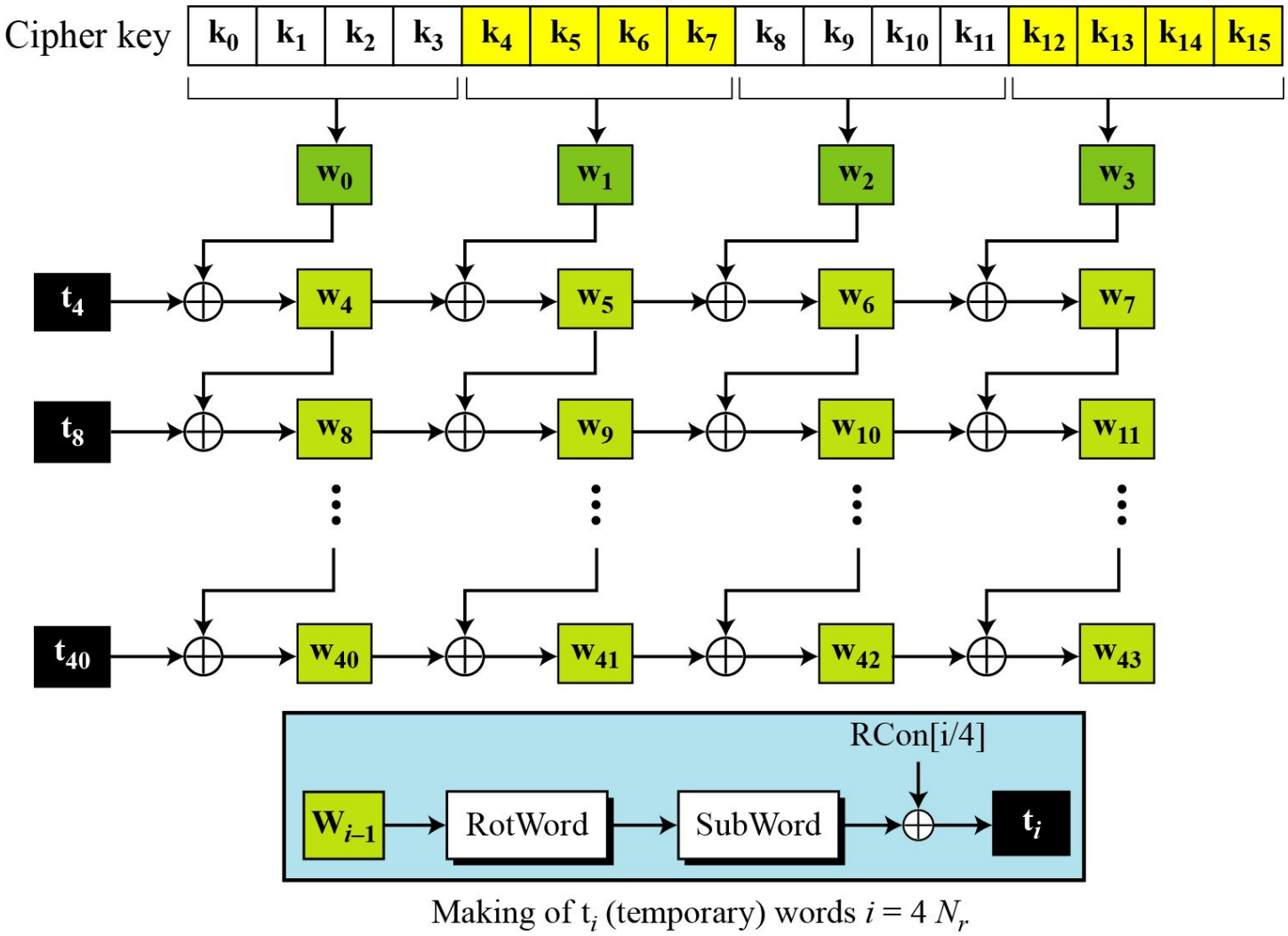
AddRoundKey

AddRoundKey proceeds one column at a time. AddRoundKey adds a round key word with each state column matrix; the operation in AddRoundKey is matrix addition.



Key Expansion in AES-128

Figure 7.16 *Key expansion in AES*



Making of t_i (temporary) words $i = 4\ N_r$