Cryptography and Network Security—CY53
William Stallings 6th Edition

Stream Ciphers and Block Ciphers

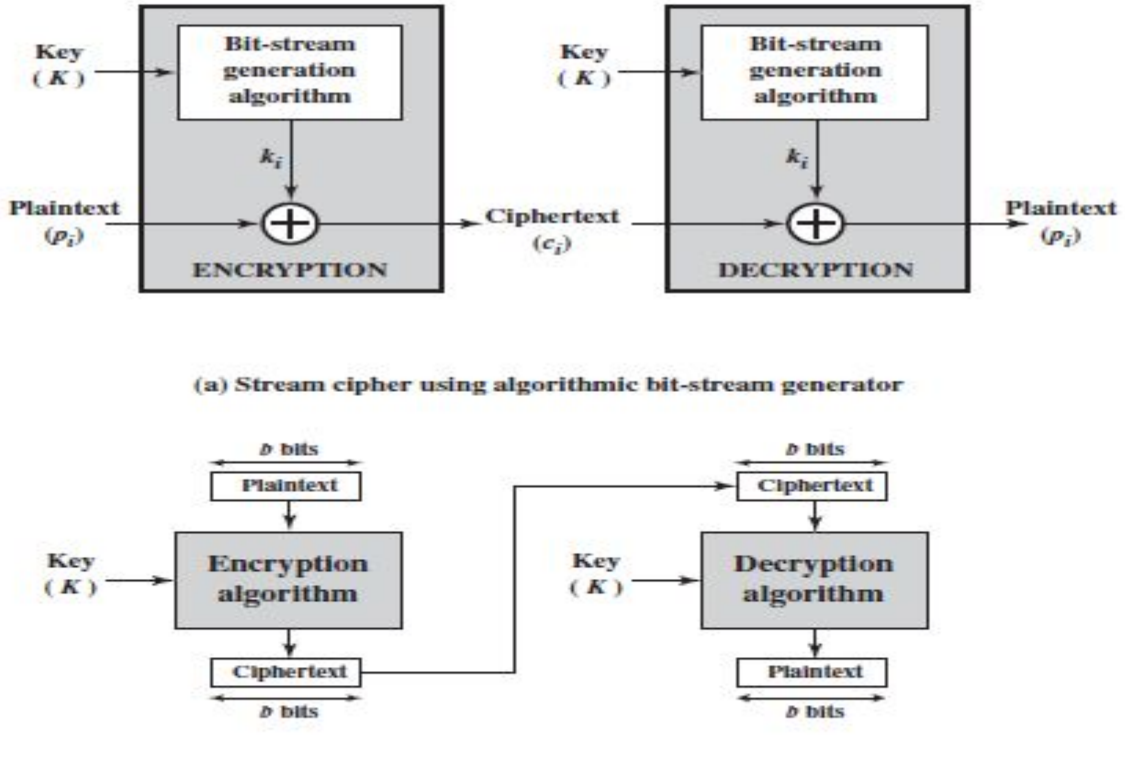
Block Ciphers Principles

Stream Ciphers and Block Ciphers:

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 the auto keyed Vigenère 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.

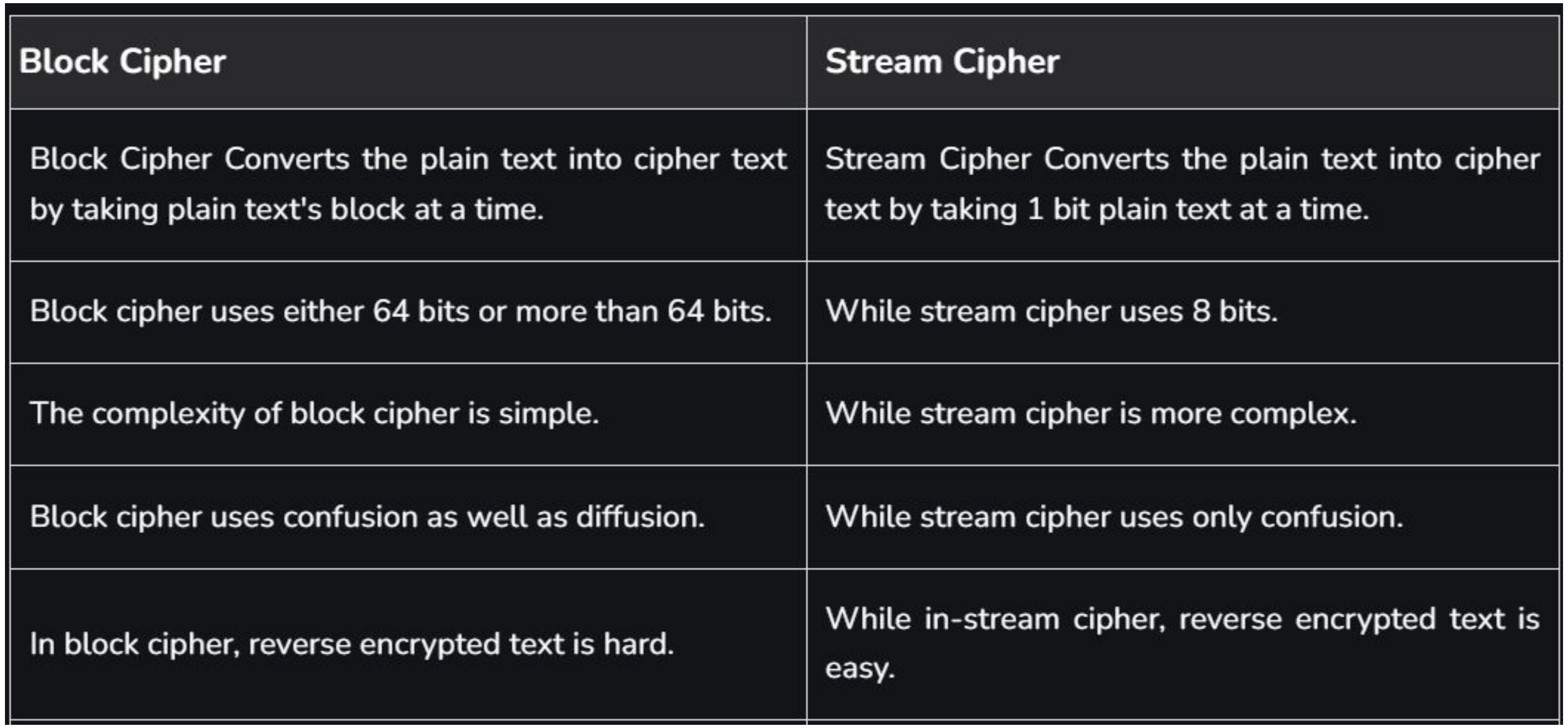
Stream Ciphers and Block Ciphers



(b) Block cipher

Figure 3.1 Stream Cipher and Block Cipher

Stream Ciphers and Block Ciphers



Block Cipher	Stream Cipher
Block Cipher Converts the plain text into cipher text by taking plain text's block at a time.	Stream Cipher Converts the plain text into cipher text by taking 1 bit plain text at a time.
Block cipher uses either 64 bits or more than 64 bits.	While stream cipher uses 8 bits.
The complexity of block cipher is simple.	While stream cipher is more complex.
Block cipher uses confusion as well as diffusion.	While stream cipher uses only confusion.
In block cipher, reverse encrypted text is hard.	While in-stream cipher, reverse encrypted text is easy.

Feistel Cipher

Feistel Cipher model is a structure or a design used to develop many block ciphers such as DES. Feistel cipher may have invertible, non-invertible and self invertible components in its design. A separate key is used for each round. However same round keys are used for encryption as well as decryption.

- **Substitution**: Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.
- **Permutation**: A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

Feistel Cipher

Feistel's is a practical application of proposal by Claude Shannon to develop a product cipher that alternates **Confusion** and **Diffusion** functions.

Diffusion:-: It is a method to frustrate cryptanalysis. In diffusion, the statistical structure of the plaintext is dissipated into long range statistics of the ciphertext. This is achieved by having each plaintext digit affect the value of many ciphertext digits; generally this is equivalent to having each ciphertext digit be affected by many plaintext digits. An example of diffusion is to encrypt a message *M* = *m*1, *m*2, *m*3,... of characters with an averaging operation:

$$y_n = \left(\sum_{i=1}^{k} m_{n+i} \right) \bmod 26$$

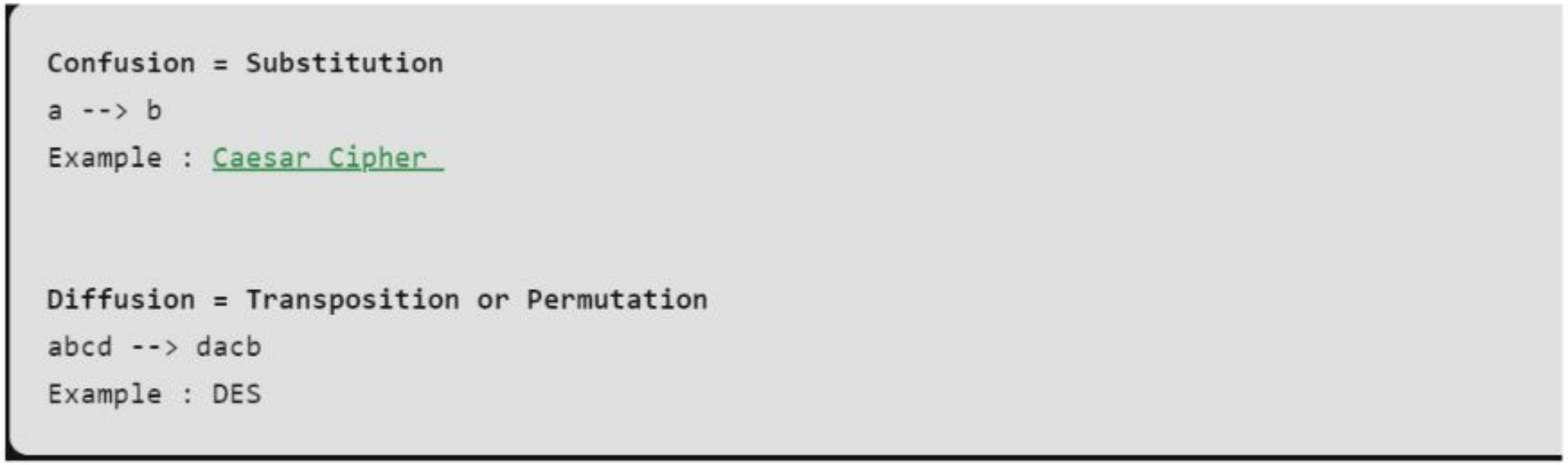
adding *k* successive letters to get a ciphertext letter *yn*. One can show that the statistical structure of the plaintext has been dissipated.

In a binary block cipher, diffusion can be achieved by repeatedly performing some permutation on the data followed by applying a function to that permutation; the effect is that bits from different positions in the original plaintext contribute to a single bit of ciphertext.

Feistel Cipher

Confusion: Confusion defines making the relationship between the key and the cipher as difficult and as included as possible. In other words, the technique provides that the ciphertext provides no clue about the plaintext.

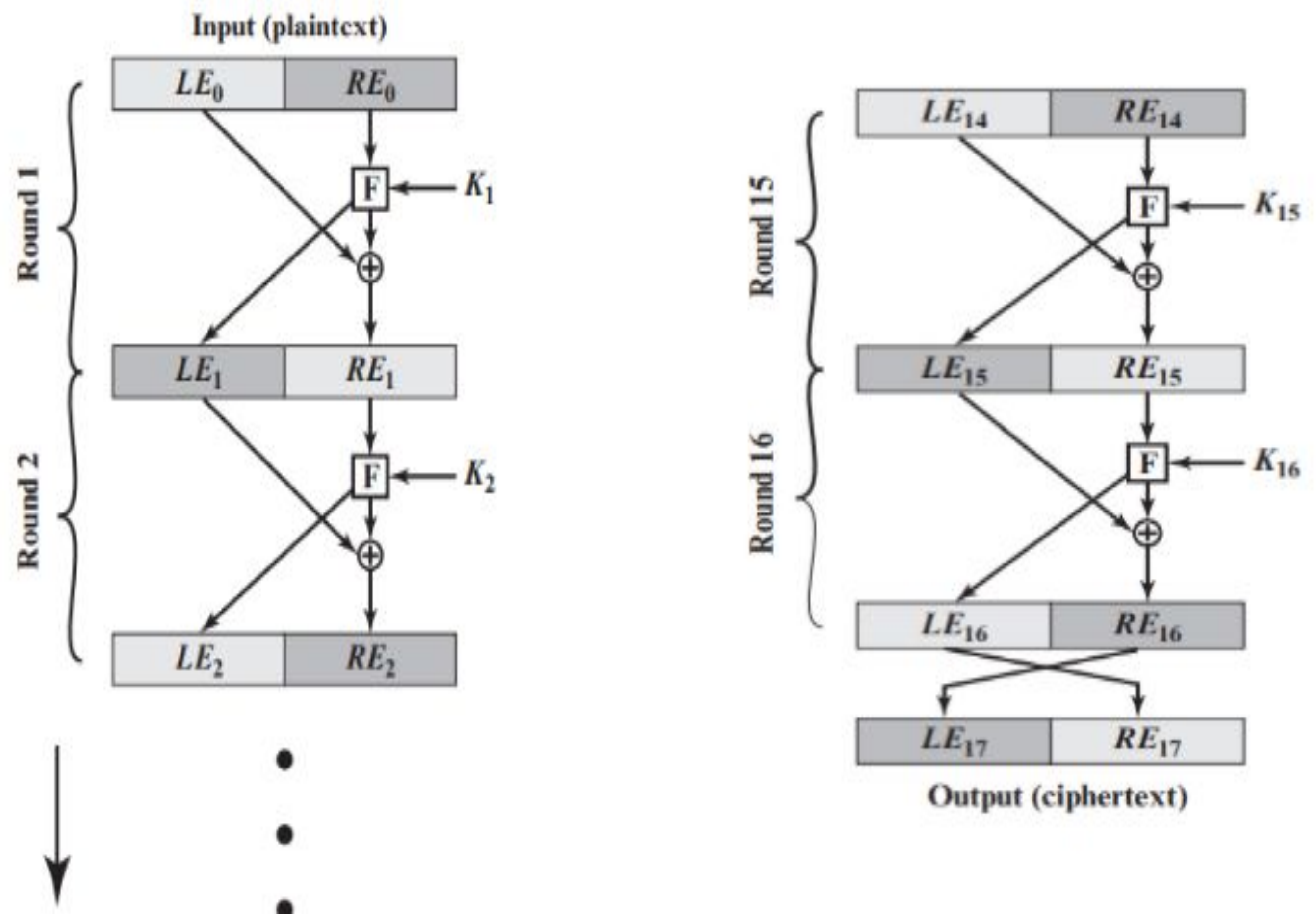
In confusion, it should influence the complete cipher text or change should appear on the complete cipher text and the relationship between the data of the ciphertext and the value of the encryption key is made difficult. It is achieved by substitution.



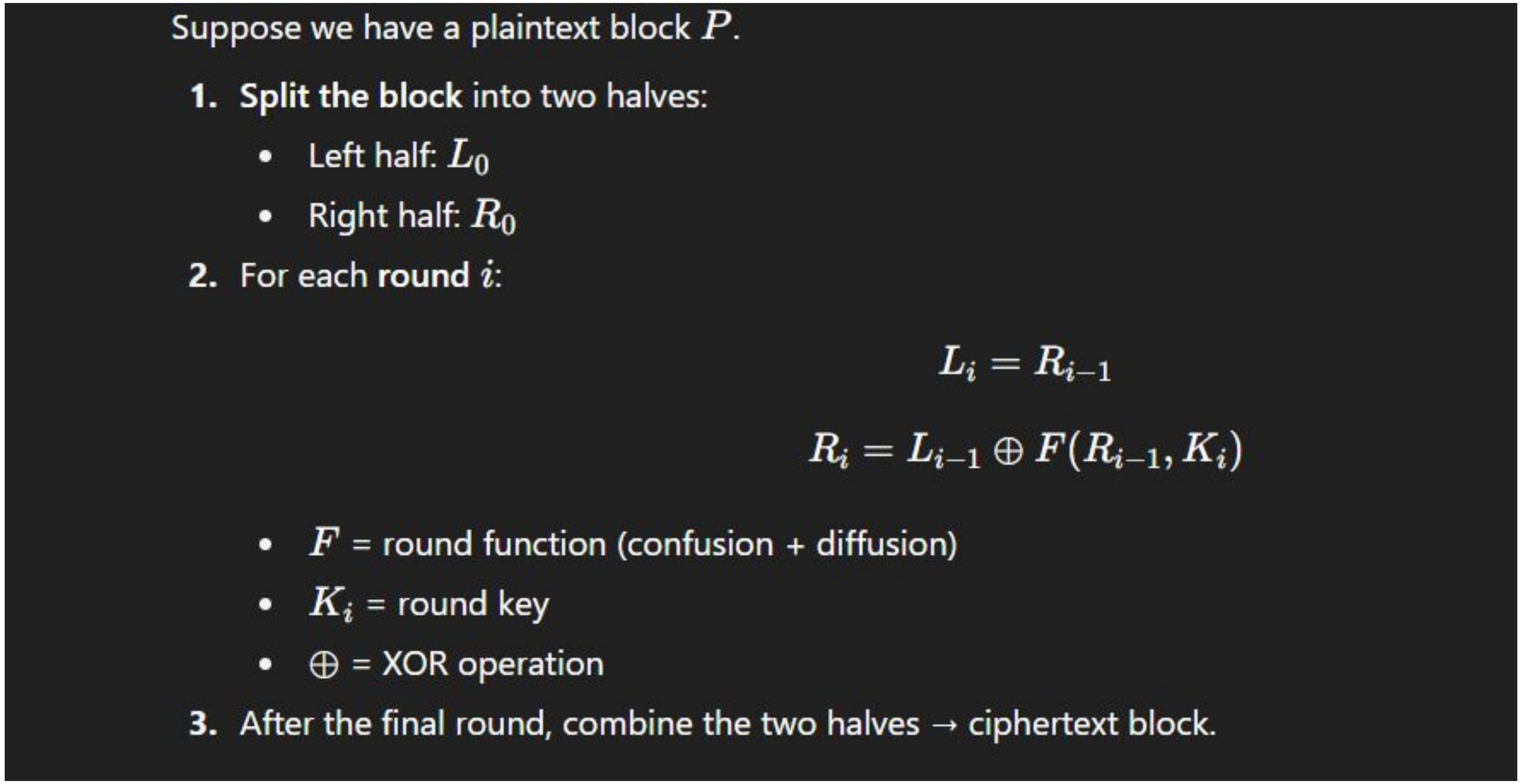
```
Confusion = Substitution
a --> b
Example : Caesar Cipher


Diffusion = Transposition or Permutation
abcd --> dacb
Example : DES
```

Feistel Cipher

Feistel Cipher Encryption



Feistel Cipher



Suppose we have a plaintext block P.

1. **Split the block** into two halves:
   - Left half: L_0
   - Right half: R_0
2. For each **round** i:

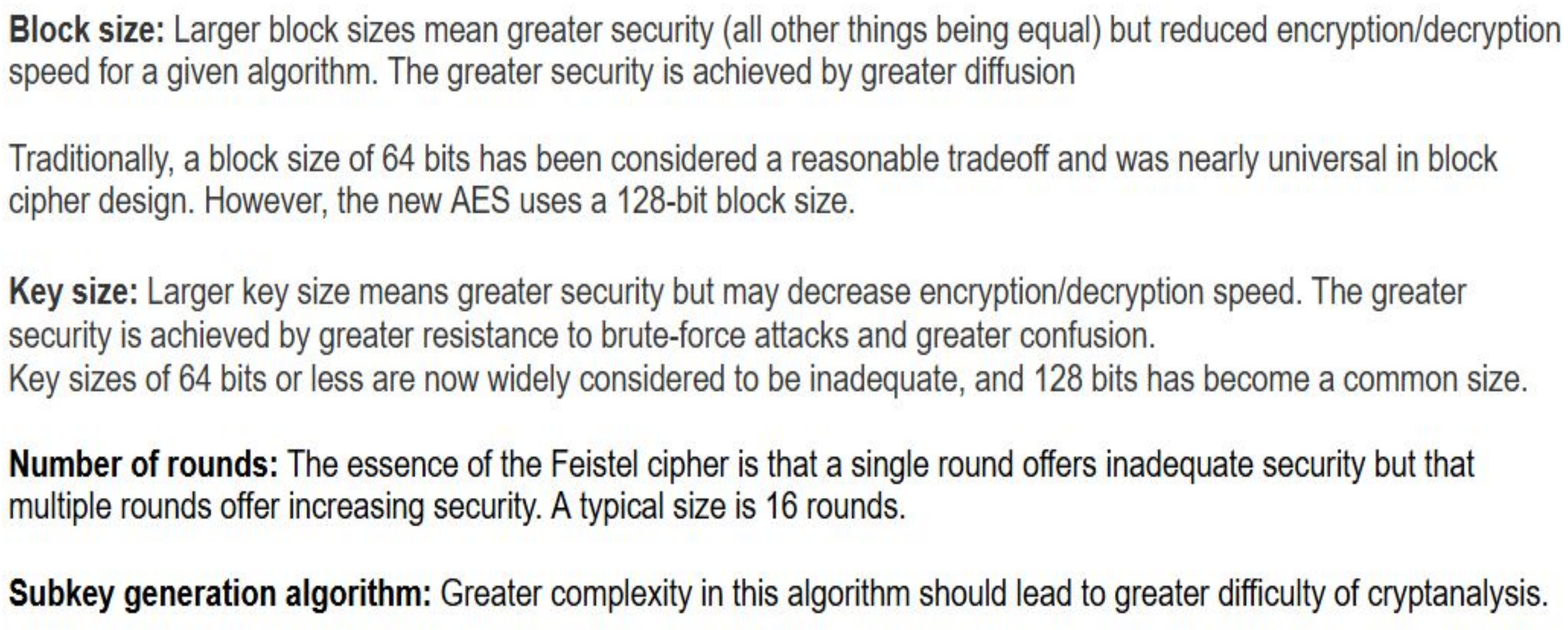
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

   - F = round function (confusion + diffusion)
   - K_i = round key
   - $\oplus$ = XOR operation
3. After the final round, combine the two halves → ciphertext block.

Feistel Cipher

The realization of Feistel network depends on following parameters



Block size: Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion

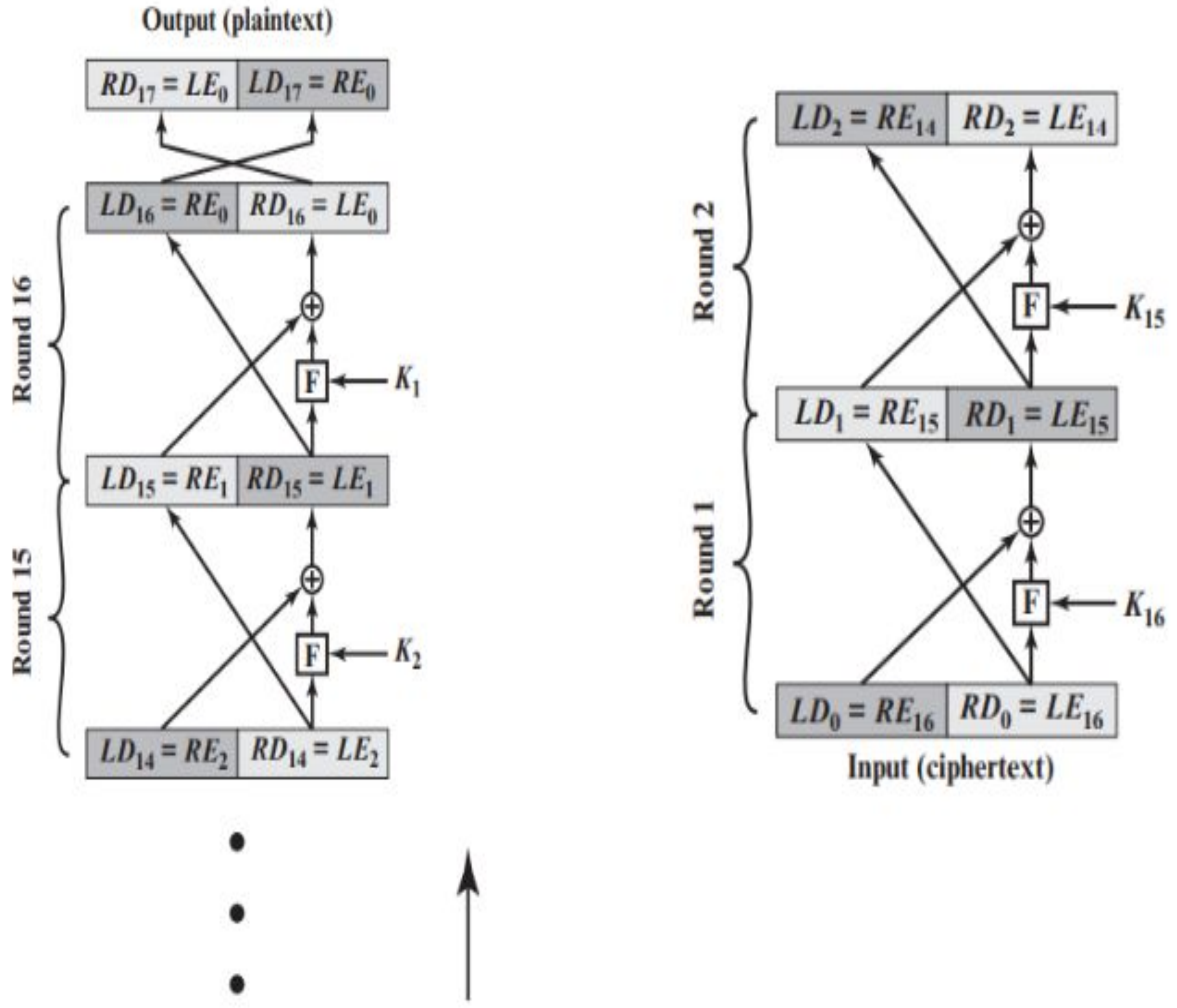
Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design. However, the new AES uses a 128-bit block size.

Key size: Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion.
Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.

Number of rounds: The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

Subkey generation algorithm: Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

Feistel Cipher Decryption



Feistel Cipher Decryption

Now we would like to show that the output of the first round of the decryption process is equal to a 32-bit swap of the input to the sixteenth round of the encryption process. First, consider the encryption process. We see that

$$
\begin{aligned}
LE_{16} &= RE_{15} \\
RE_{16} &= LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})
\end{aligned}
$$

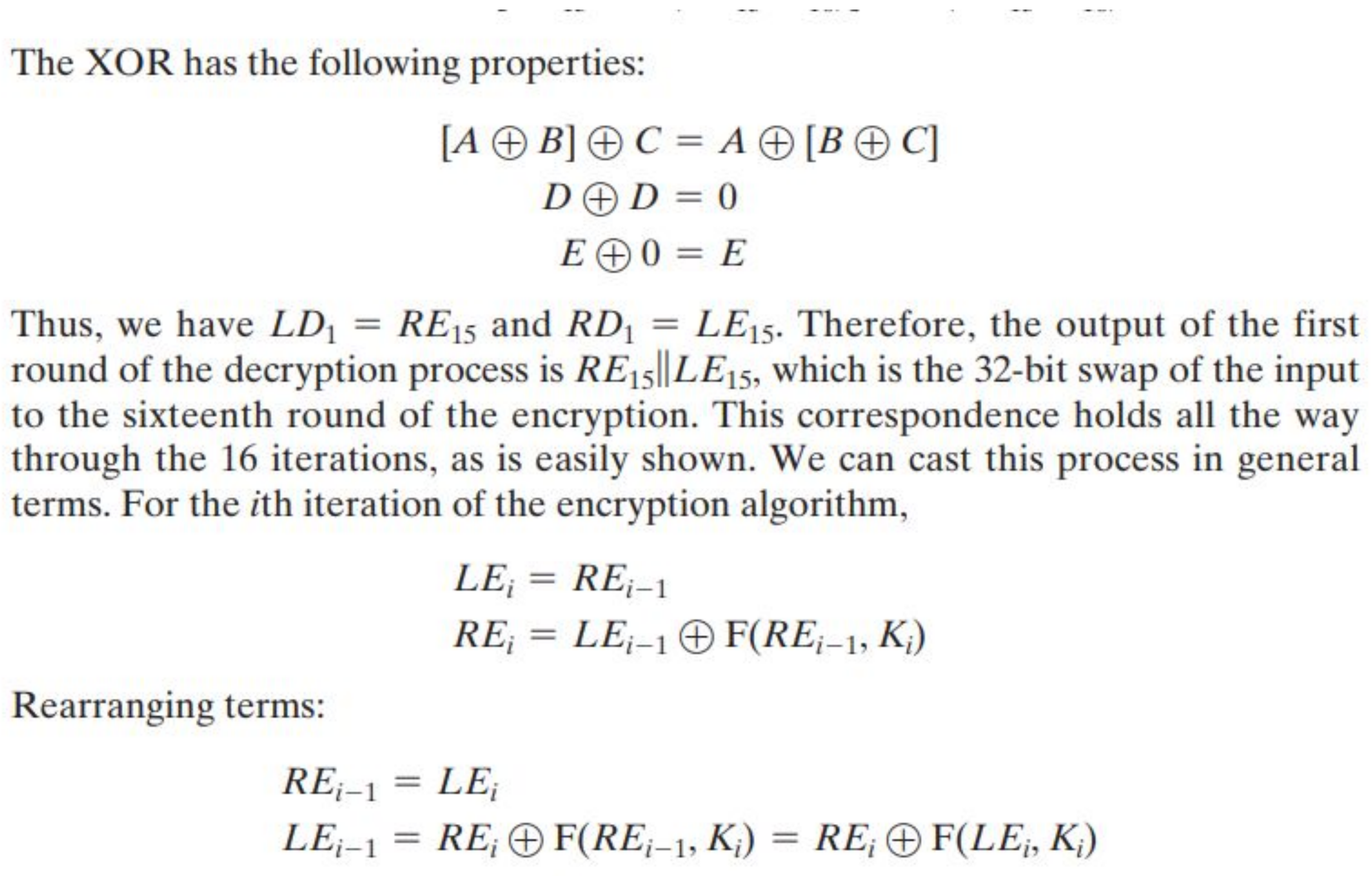
On the decryption side,

$$
\begin{aligned}
LD_1 &= RD_0 = LE_{16} = RE_{15} \\
RD_1 &= LD_0 \oplus \mathrm{F}(RD_0, K_{16}) \\
&= RE_{16} \oplus \mathrm{F}(RE_{15}, K_{16}) \\
&= [LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})] \oplus \mathrm{F}(RE_{15}, K_{16})
\end{aligned}
$$

The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$

Feistel Cipher Decryption



The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$. Therefore, the output of the first round of the decryption process is $RE_{15} \| LE_{15}$, which is the 32-bit swap of the input to the sixteenth round of the encryption. This correspondence holds all the way through the 16 iterations, as is easily shown. We can cast this process in general terms. For the *i*th iteration of the encryption algorithm,

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus \mathrm{F}(RE_{i-1}, K_i)$$

Rearranging terms:

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus \mathrm{F}(RE_{i-1}, K_i) = RE_i \oplus \mathrm{F}(LE_i, K_i)$$

DES-Data Encryption Standard

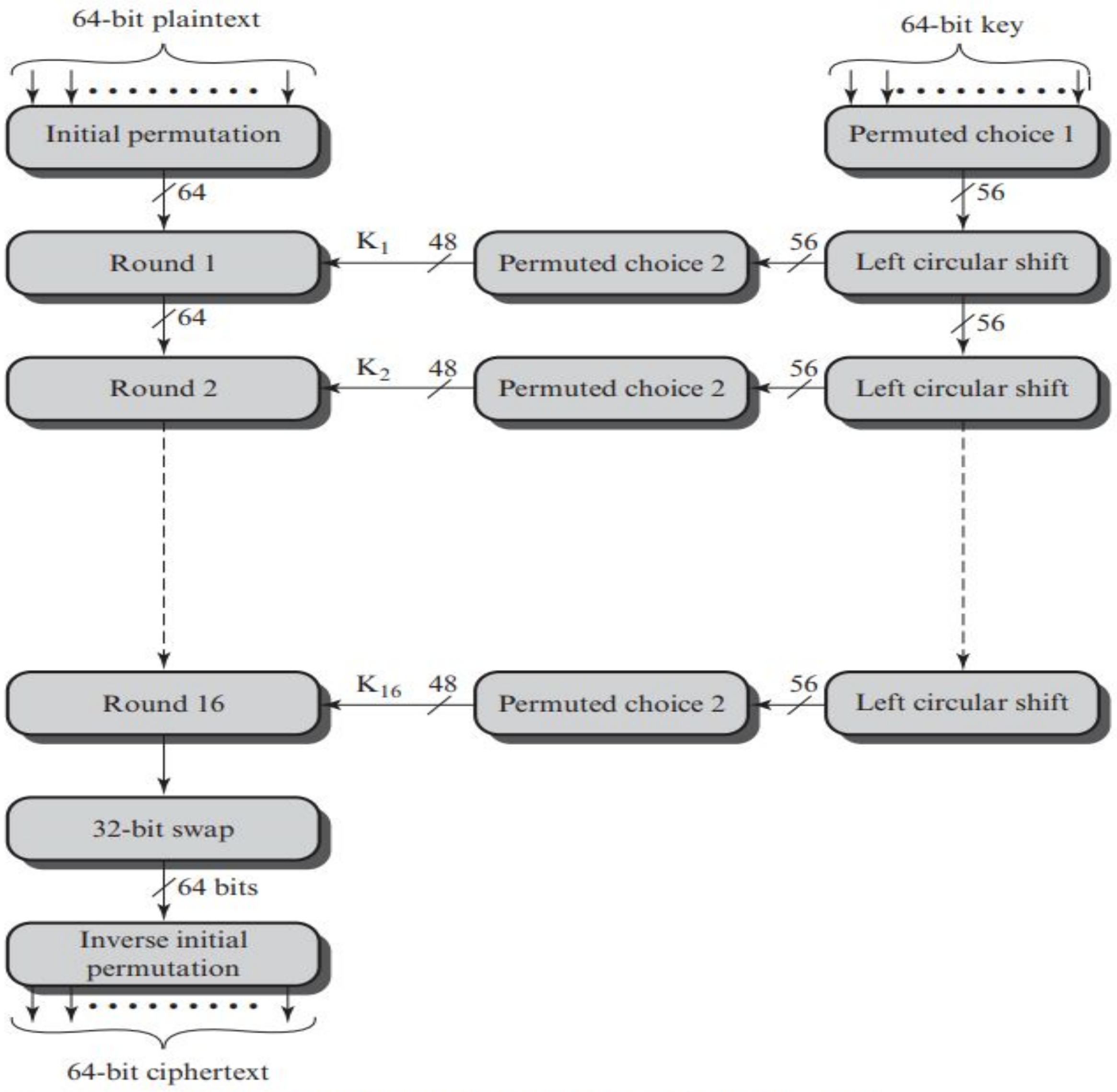


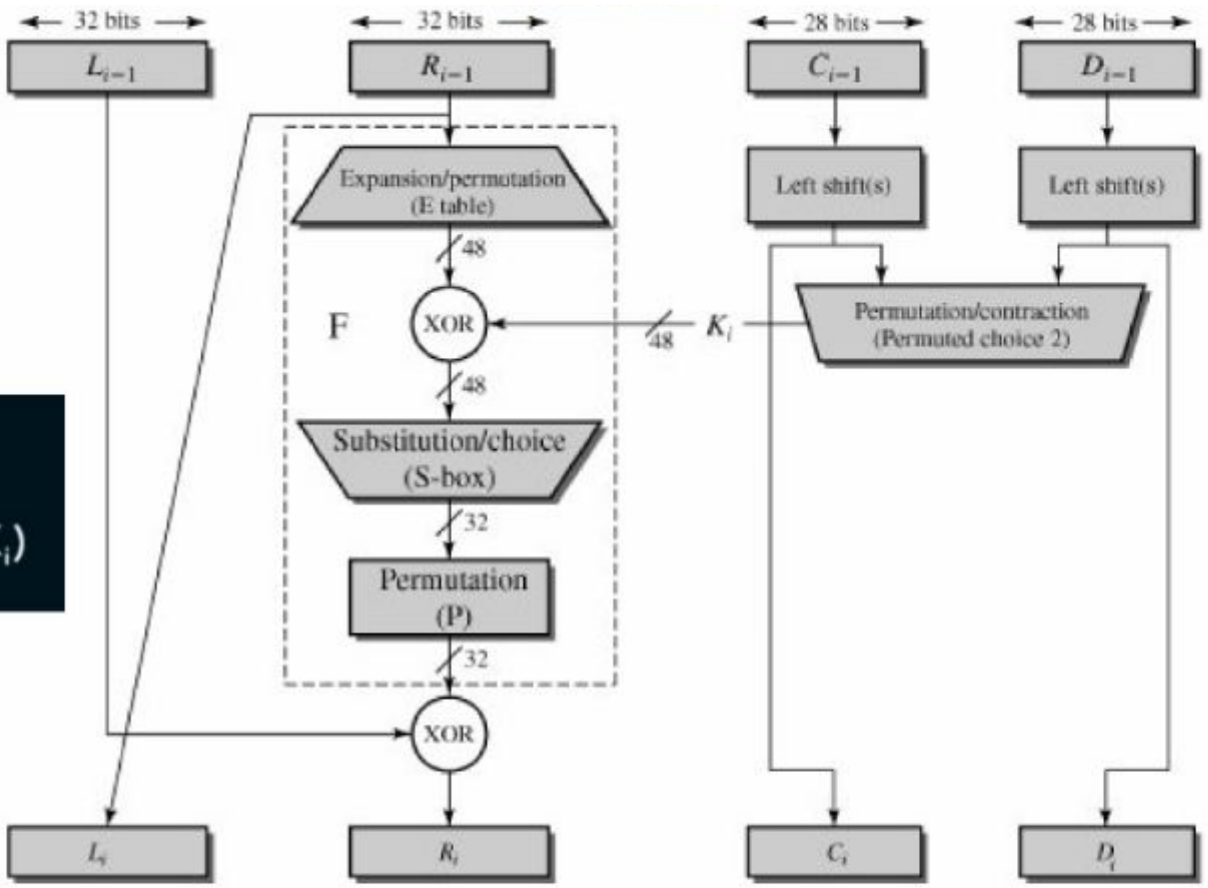
Figure 4.5 General Depiction of DES Encryption Algorithm

DES-Data Encryption Standard

Fig: Single Round of DES algorithm

$L_i = R_{i-1}$

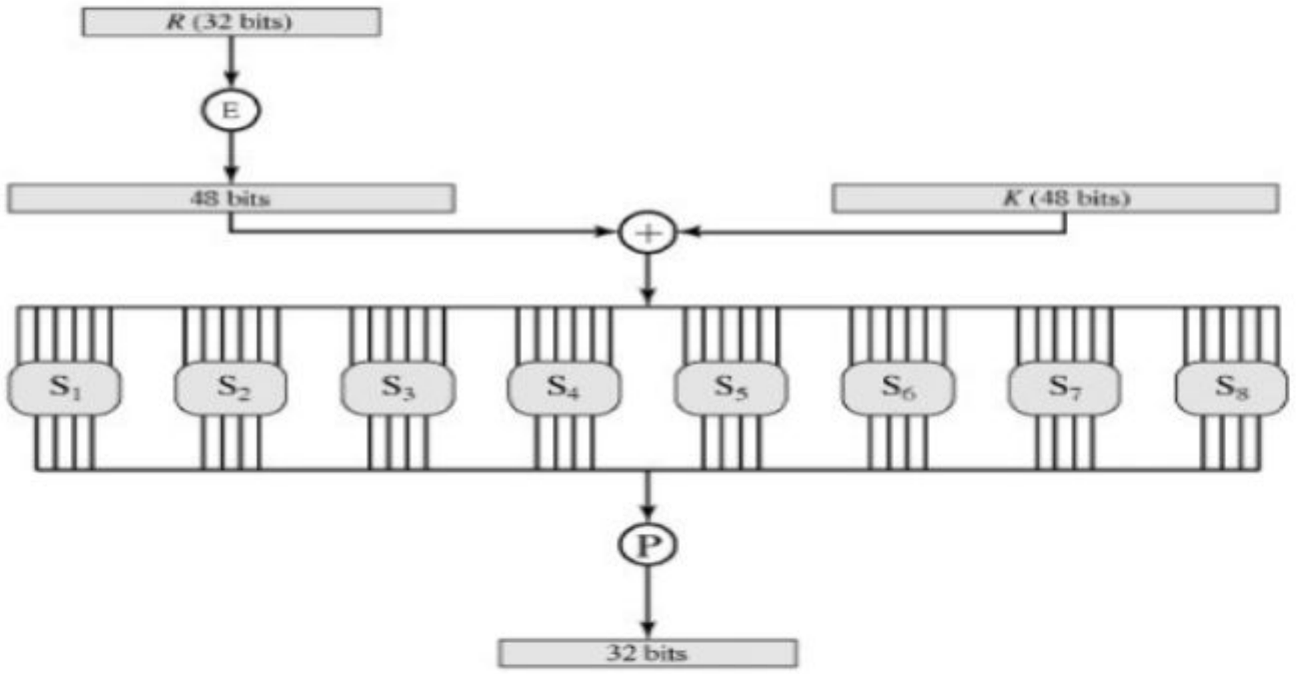
$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$



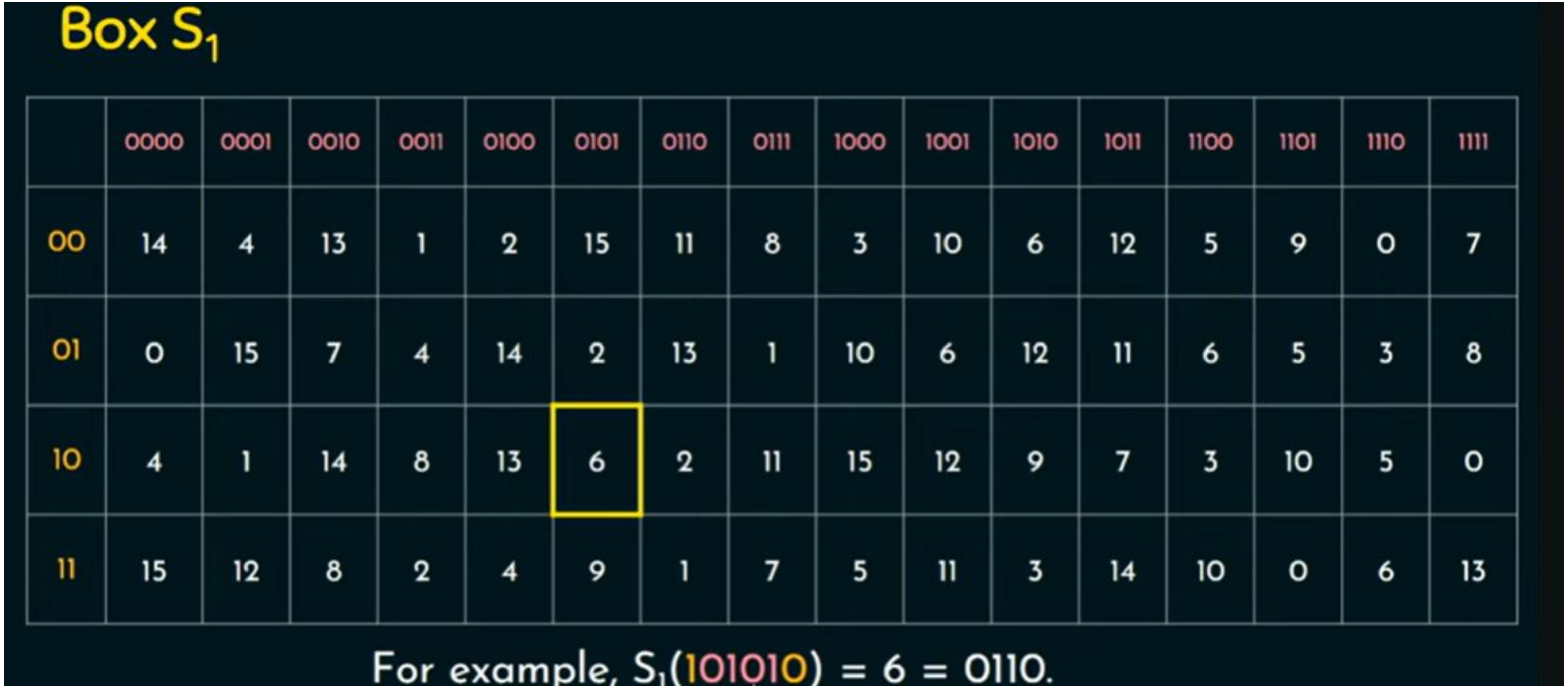
DES-Data Encryption Standard

Fig: Calculation of F(R,K)—F function

Figure 3.6. Calculation of F(R, K)
(This item is displayed on page 78 in the print version)



DES-Data Encryption Standard



Box S_1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
10	4	1	14	8	13	$\boxed{6}$	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

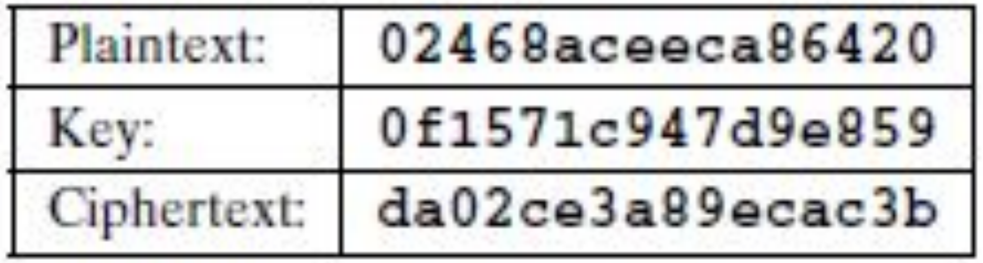
For example, $S_1(101010) = 6 = 0110$.

DES-Data Encryption Standard

DES Decryption:-

Decryption uses the same algorithm as encryption, except that the application of subkeys is reversed. Additionally initial and final permutations are reversed.

DES-Data Encryption Standard Example



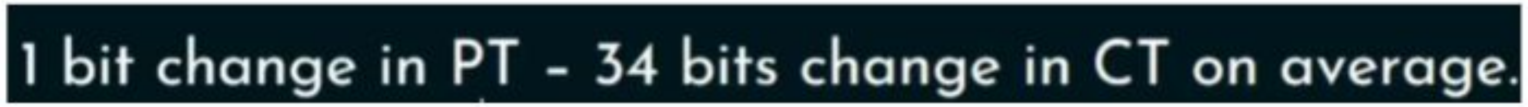
Plaintext:	02468aceeca86420
Key:	0f1571c947d9e859
Ciphertext:	da02ce3a89ecac3b

DES-Data Encryption Standard

The Avalanche Effect:

- A desirable property of any encryption algorithm is that a small change in either plaintext or the key should produce a significant change in the ciphertext.
- A change in one bit of the plaintext or one of the bit of the key should produce a change in many bits of the ciphertext. This is referred as Avalanche Effect.
- If the change were smaller, this might provide a way to reduce the size of the plaintext or key space to be searched.

For Ex:



1 bit change in PT - 34 bits change in CT on average.

DES-Data Encryption Standard

Plaintext

00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

10000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

with the key
0000001 1001011 0100100 1100010 0011100
0011000 0011100 0110010

Plaintext
01101000 10000101 00101111 01111010
00010011 01110110 11101011 10100100

with two keys that differ in only one bit position:

1110010 1111011 1101111 0011000 0011101
0000100 0110001 11011100
0110010 1111011 1101111 0011000 0011101
0000100 0110001 11011100

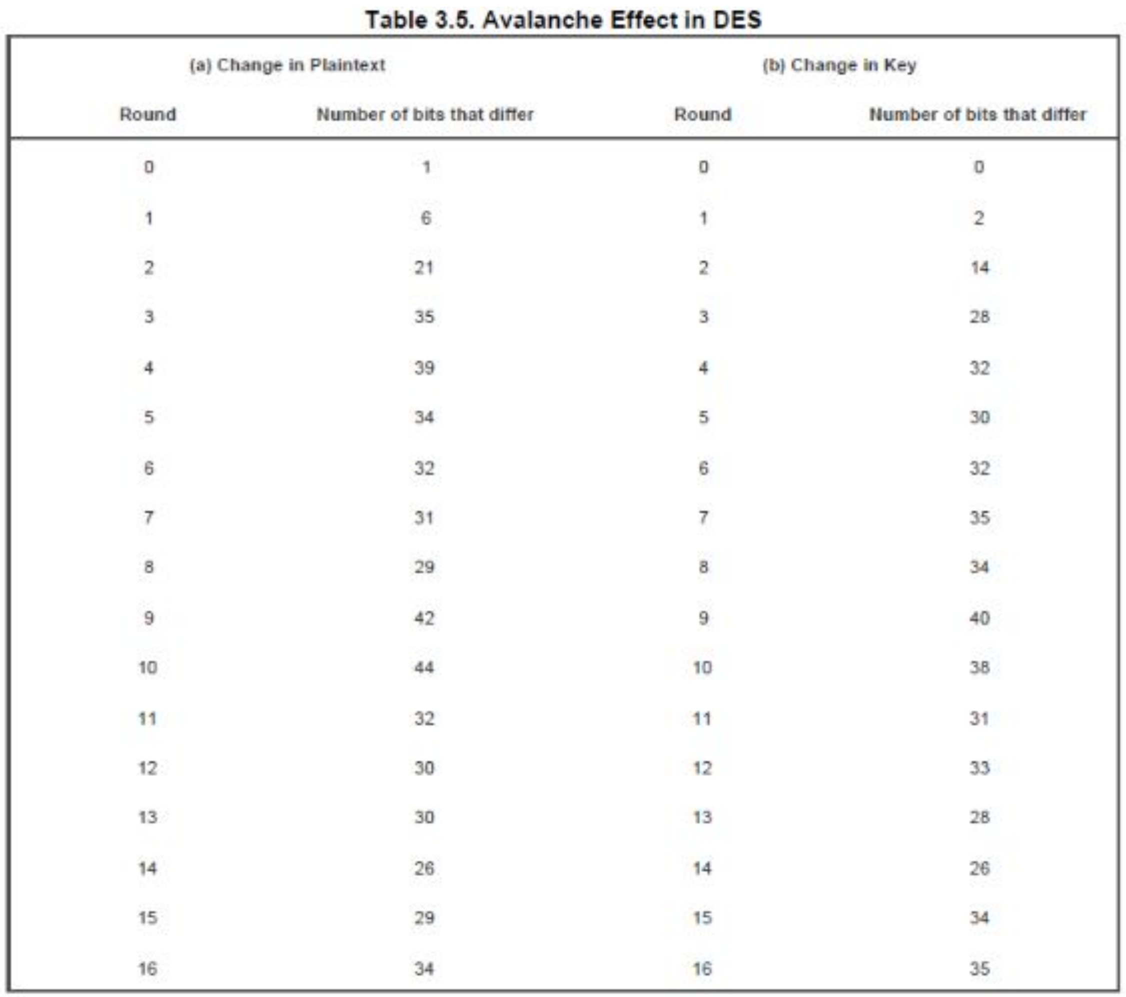
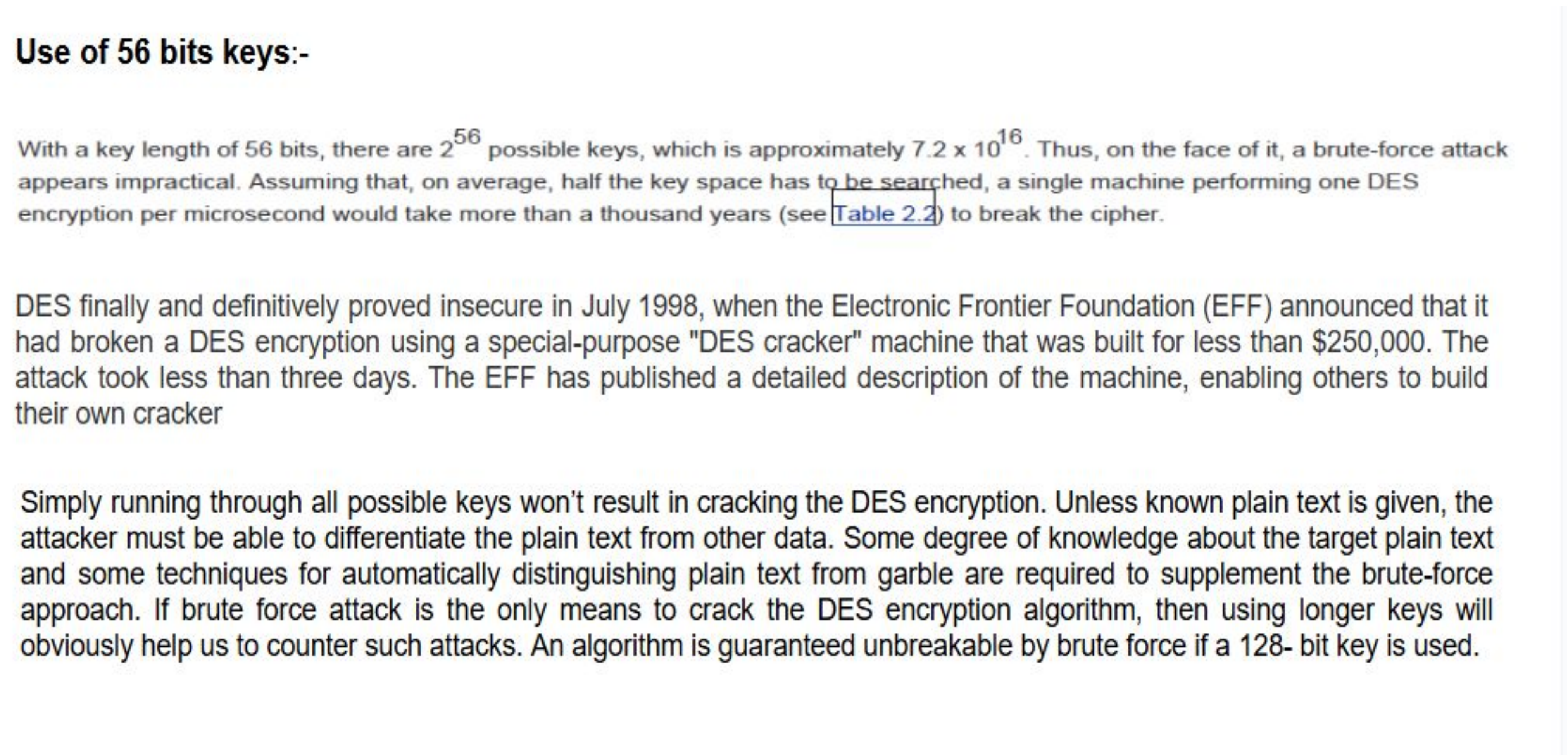


Table 3.5. Avalanche Effect in DES

(a) Change in Plaintext		(b) Change in Key	
Round	Number of bits that differ	Round	Number of bits that differ
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

Strength of DES-Data Encryption Standard



Use of 56 bits keys:-

With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2×10^{16}. Thus, on the face of it, a brute-force attack appears impractical. Assuming that, on average, half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years (see Table 2.2) to break the cipher.

DES finally and definitively proved insecure in July 1998, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose "DES cracker" machine that was built for less than $250,000. The attack took less than three days. The EFF has published a detailed description of the machine, enabling others to build their own cracker

Simply running through all possible keys won't result in cracking the DES encryption. Unless known plain text is given, the attacker must be able to differentiate the plain text from other data. Some degree of knowledge about the target plain text and some techniques for automatically distinguishing plain text from garble are required to supplement the brute-force approach. If brute force attack is the only means to crack the DES encryption algorithm, then using longer keys will obviously help us to counter such attacks. An algorithm is guaranteed unbreakable by brute force if a 128- bit key is used.

Strength of DES-Data Encryption Standard

The Nature of DES Algorithm

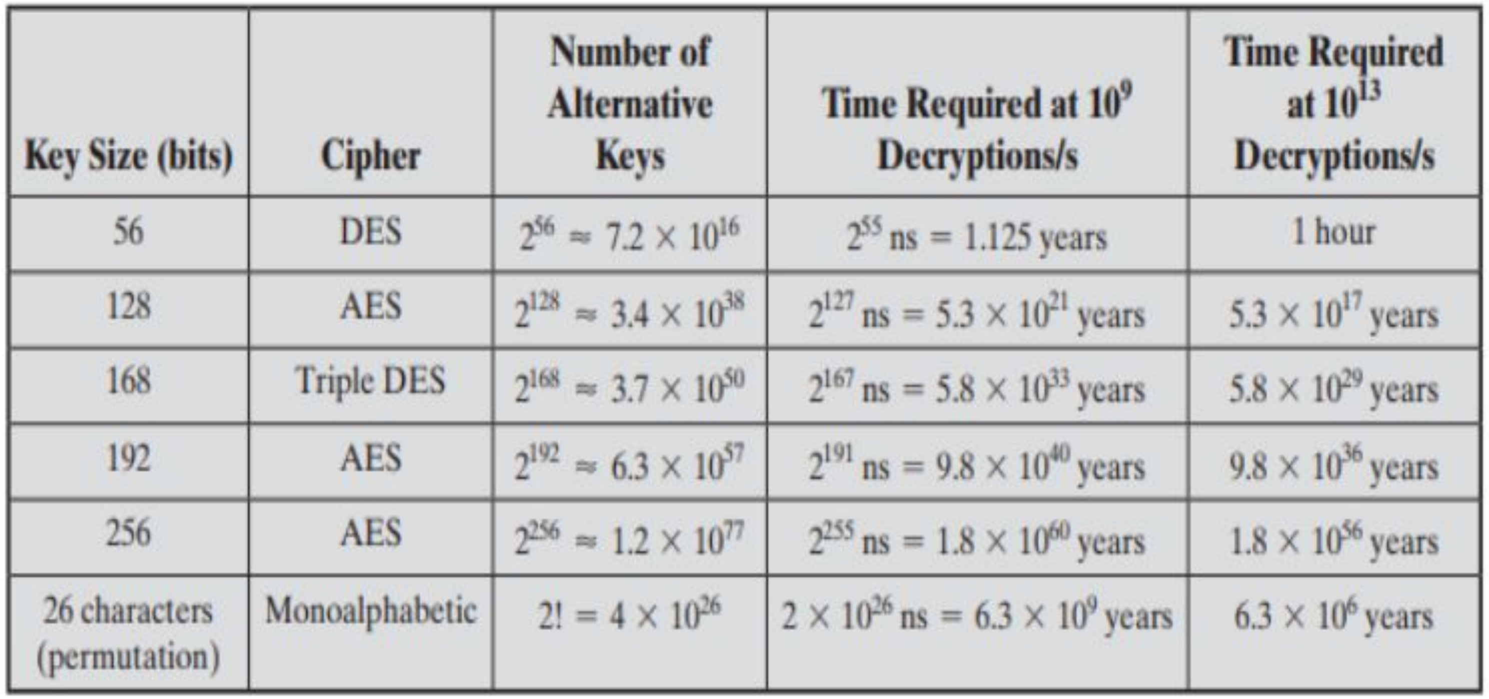
Another concern is the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm. The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration. Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.

Timing Attack

In essence, a timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts. A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs

Strength of DES-Data Encryption Standard

Use of 56 bits keys:-



Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

Block Cipher Design Principles

The cryptographic strength of a Feistel cipher derives from 3 aspects of the design: 3 critical aspects of block cipher design are:

1. **Number of Rounds:**

The greater number of rounds, more difficult it is to perform cryptanalysis, even for a relatively weak F. In general criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater than effort than a simple brute force key search attack.

The number of Rounds is regularly considered in design criteria, it just reflects the number of rounds to be suitable for an algorithm to make it more complex, in DES we have 16 rounds ensuring it to be more secure while in AES we have 10 rounds which makes it more secure.

Block Cipher Design Principles

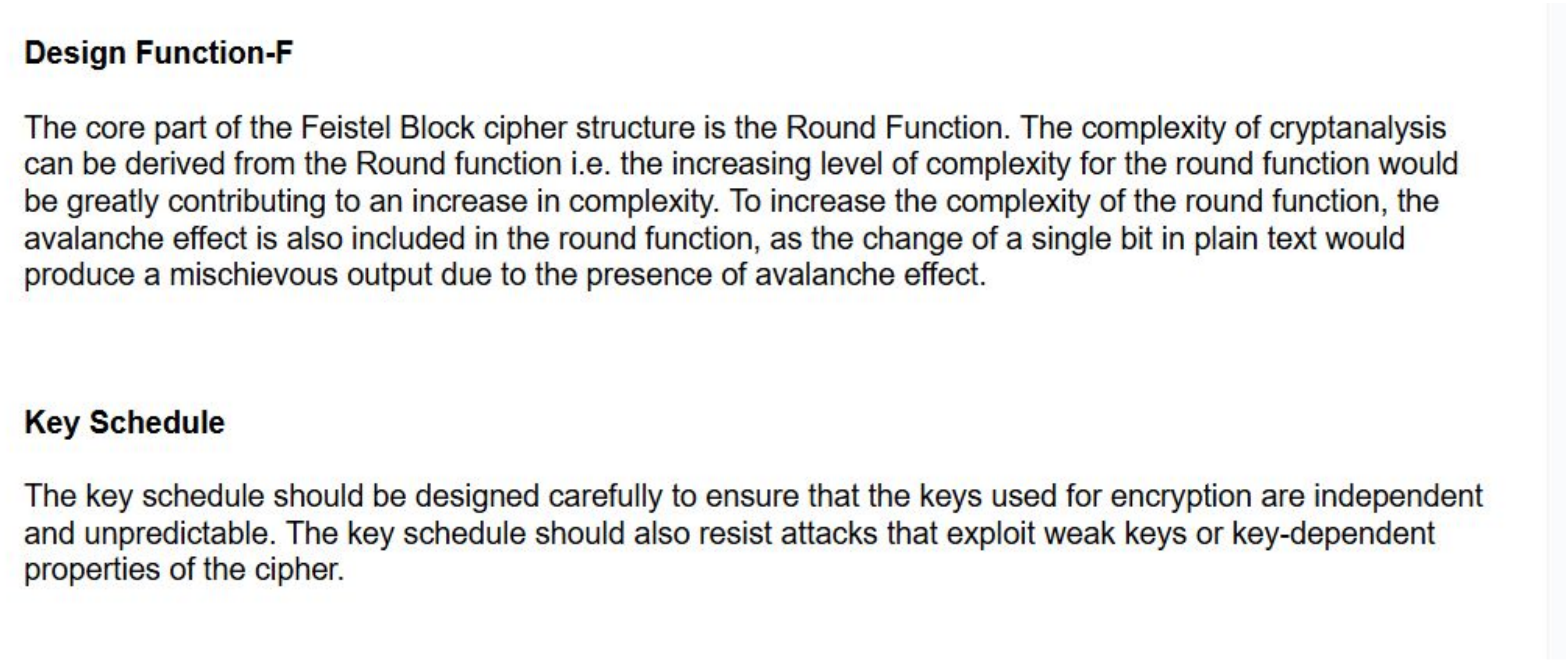
Design Function-F

The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity. To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.

S-Box would be selected based on below criteria

- **Random:** Use some pseudorandom number generation or some table of random digits to generate the entries in the S-boxes. This may lead to boxes with undesirable characteristics for small sizes (e.g., 6 x 4) but should be acceptable for large S-boxes (e.g., 8 x 32).
- **Random with testing:** Choose S-box entries randomly, then test the results against various criteria, and throw away those that do not pass.
- **Human-made:** This is a more or less manual approach with only simple mathematics to support it. It is apparently the technique used in the DES design. This approach is difficult to carry through for large S-boxes.
- **Math-made:** Generate S-boxes according to mathematical principles. By using mathematical construction, S-boxes can be constructed that offer proven security against linear and differential cryptanalysis, together with good diffusion.

Block Cipher Design Principles



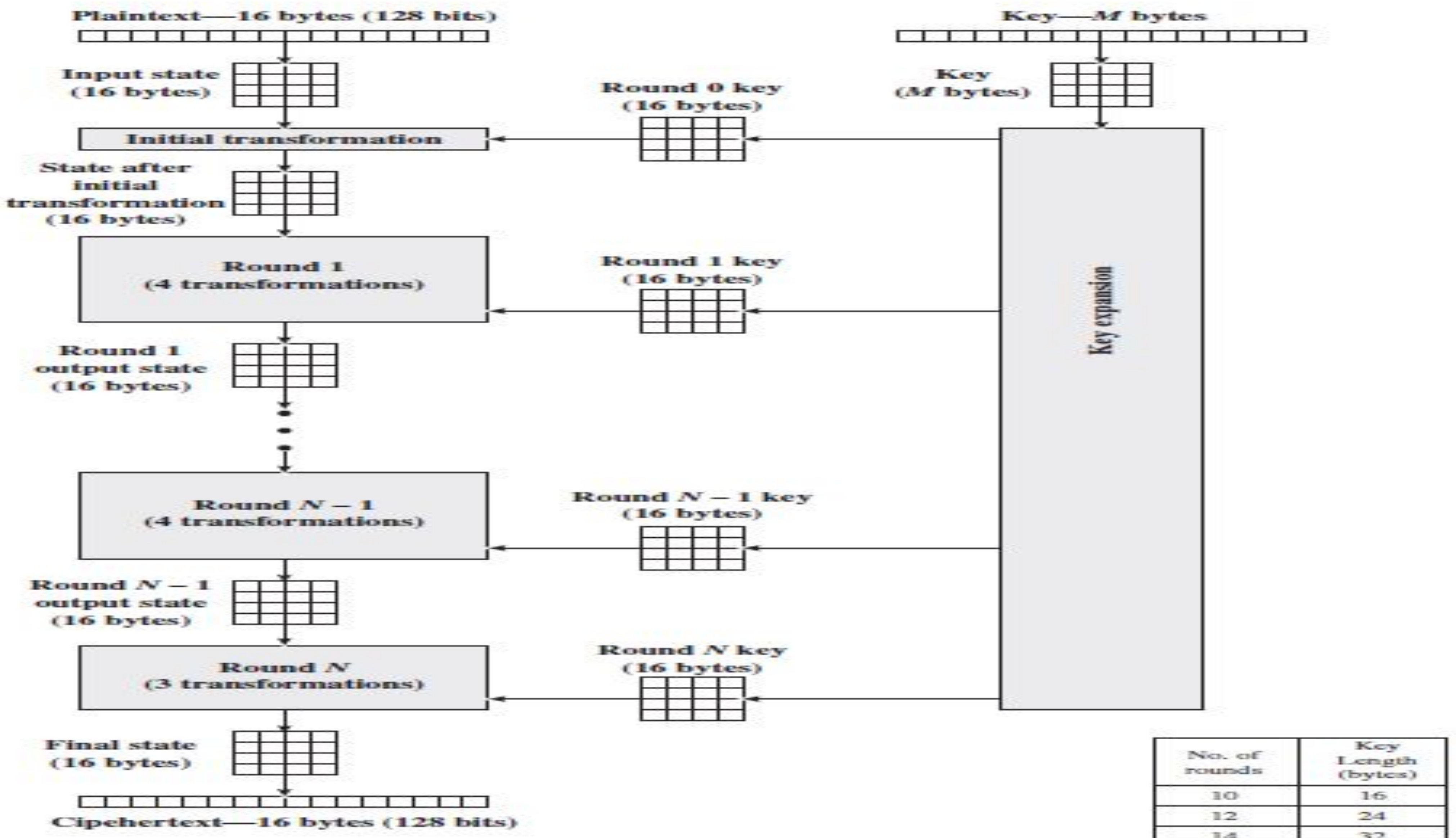
Design Function-F

The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity. To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.

Key Schedule

The key schedule should be designed carefully to ensure that the keys used for encryption are independent and unpredictable. The key schedule should also resist attacks that exploit weak keys or key-dependent properties of the cipher.

Advanced Encryption Standard Structure


Plaintext—16 bytes (128 bits)
Key—M bytes
Input state (16 bytes)
Round 0 key (16 bytes)
Key (M bytes)
Initial transformation
State after initial transformation (16 bytes)
Round 1 (4 transformations)
Round 1 key (16 bytes)
Key expansion
Round 1 output state (16 bytes)
Round N – 1 (4 transformations)
Round N – 1 key (16 bytes)
Round N – 1 output state (16 bytes)
Round N (3 transformations)
Round N key (16 bytes)
Final state (16 bytes)
Cipehertext—16 bytes (128 bits)

No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 5.1 AES Encryption Process

Advanced Encryption Standard Structure

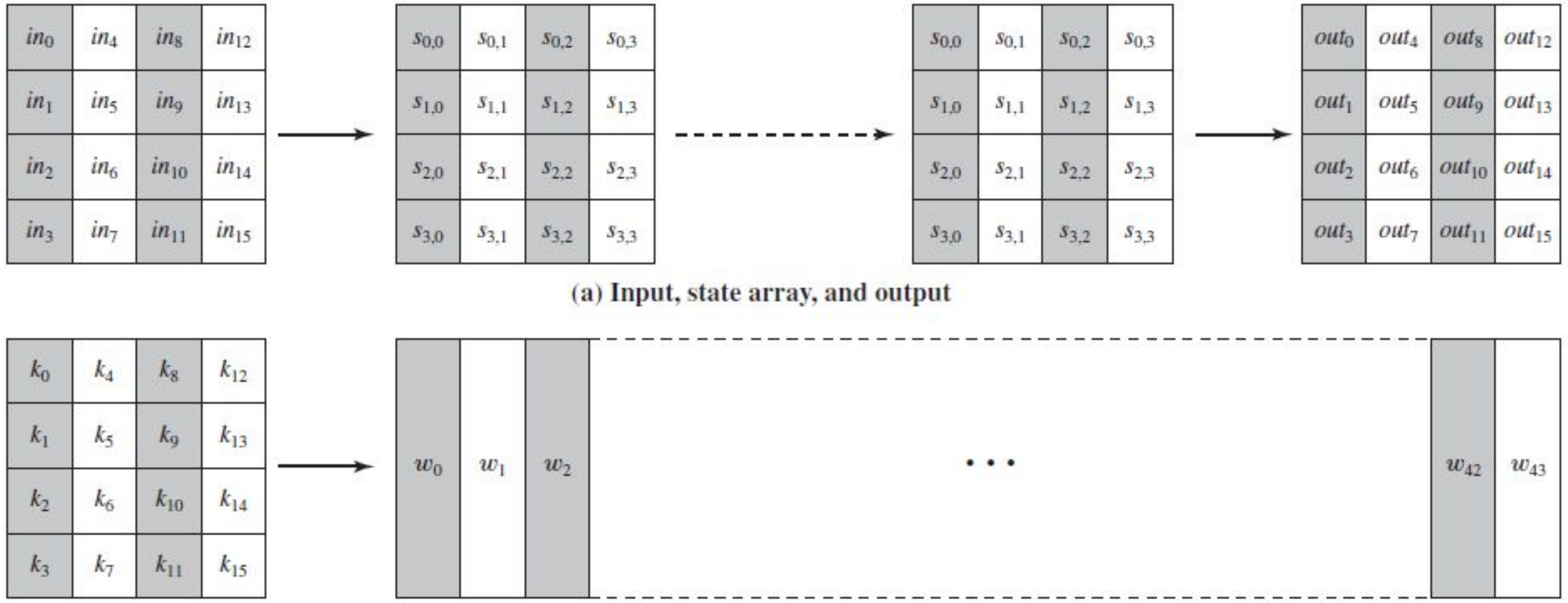


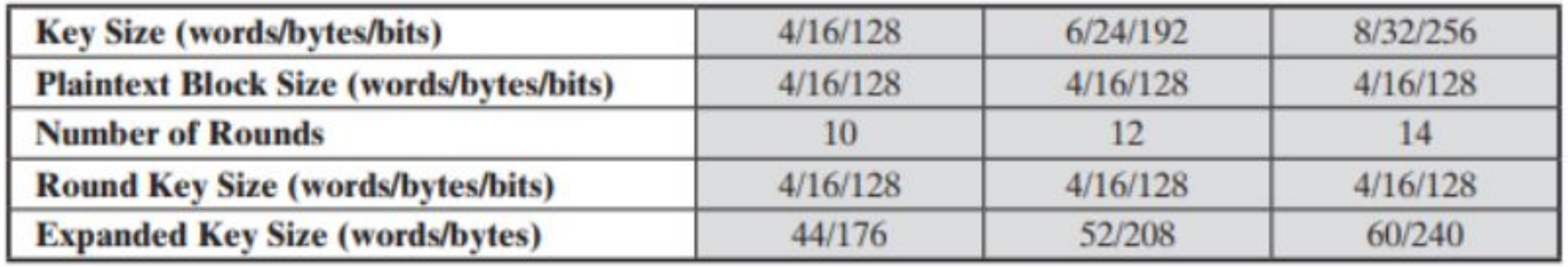
Figure 5.2 AES Data Structures

Advanced Encryption Standard

AES Cipher:-

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. The AES specification uses the same three key size alternatives but limits the block length to 128 bits. A number of AES parameters depend on the key length.

Table 1. AES Parameters



Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

Rijndael was designed to have the following characteristics:

- Resistance against all known attacks
- Speed and code compactness on a wide range of platforms
- Design simplicity

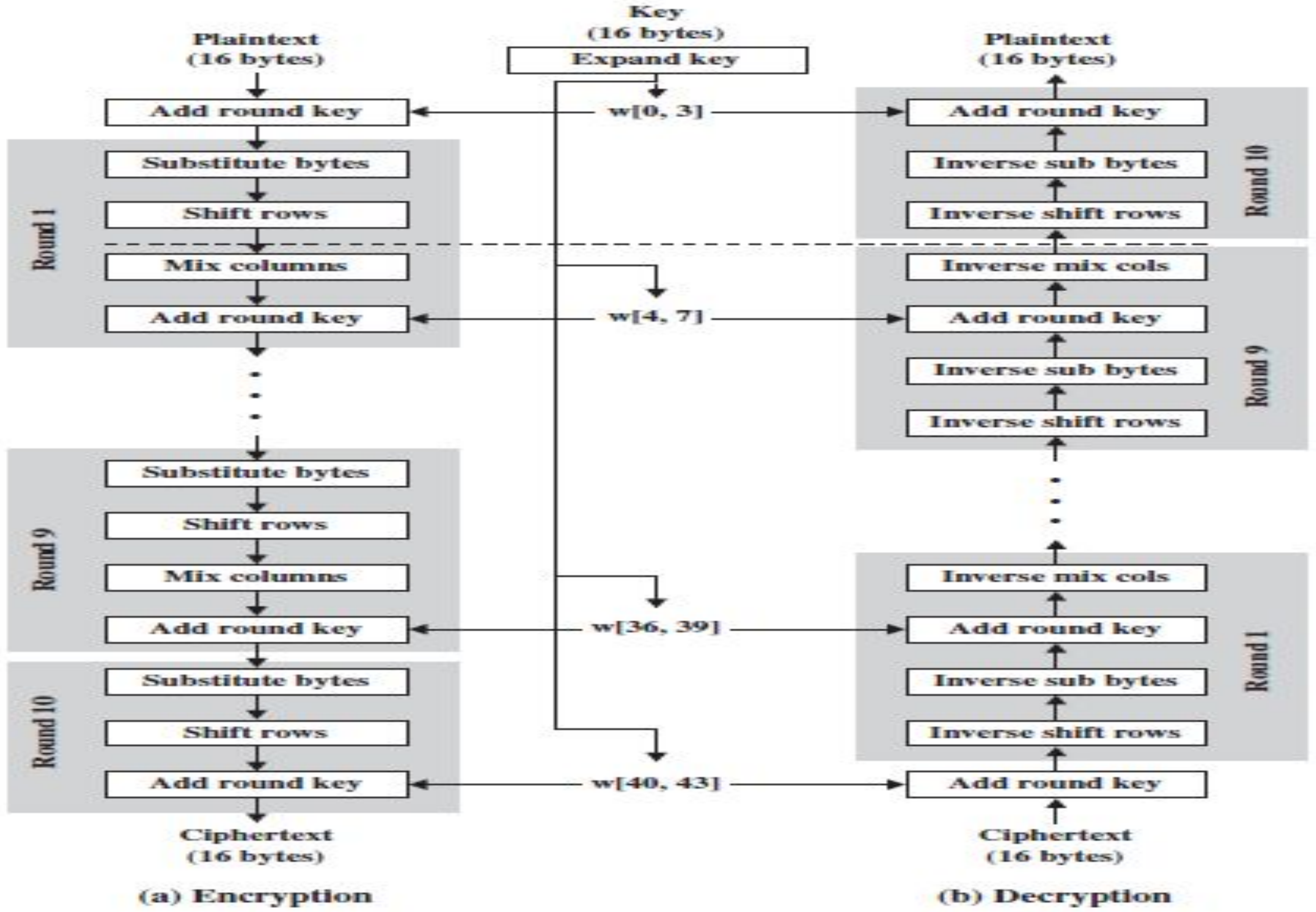
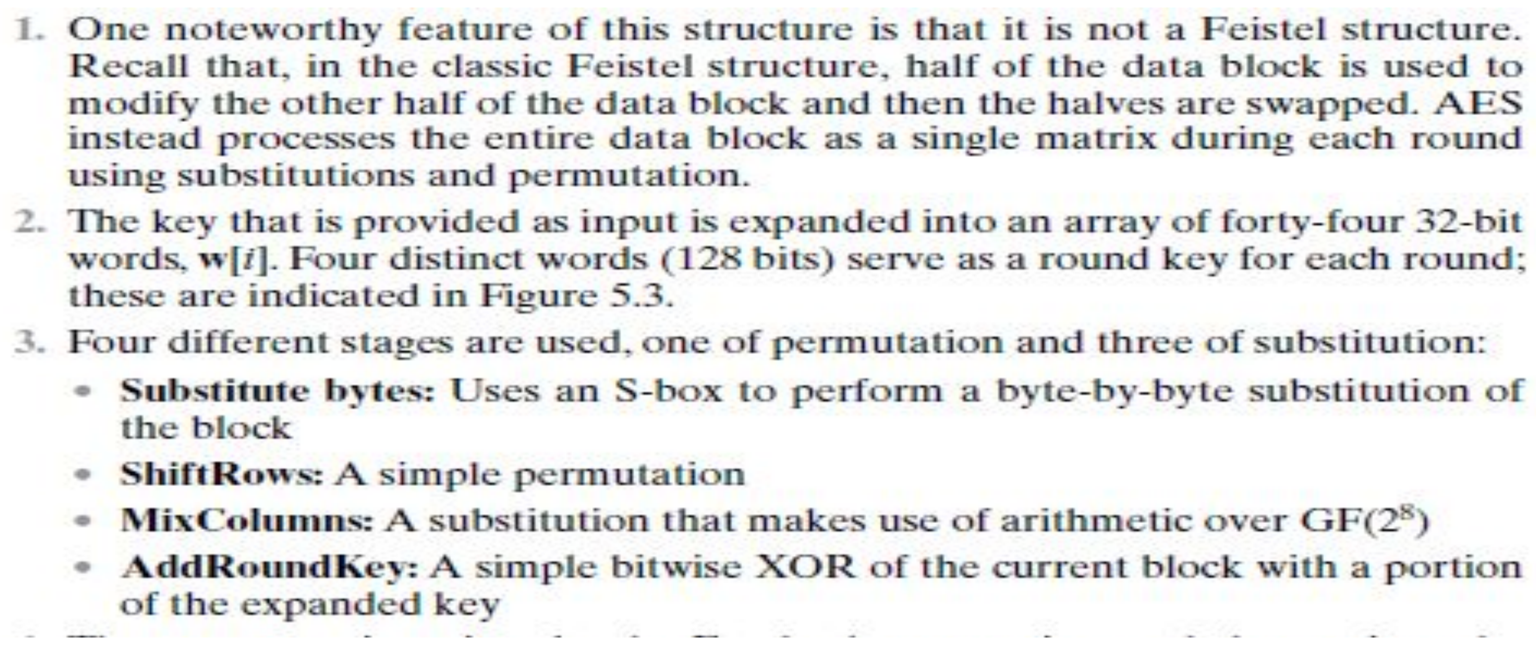


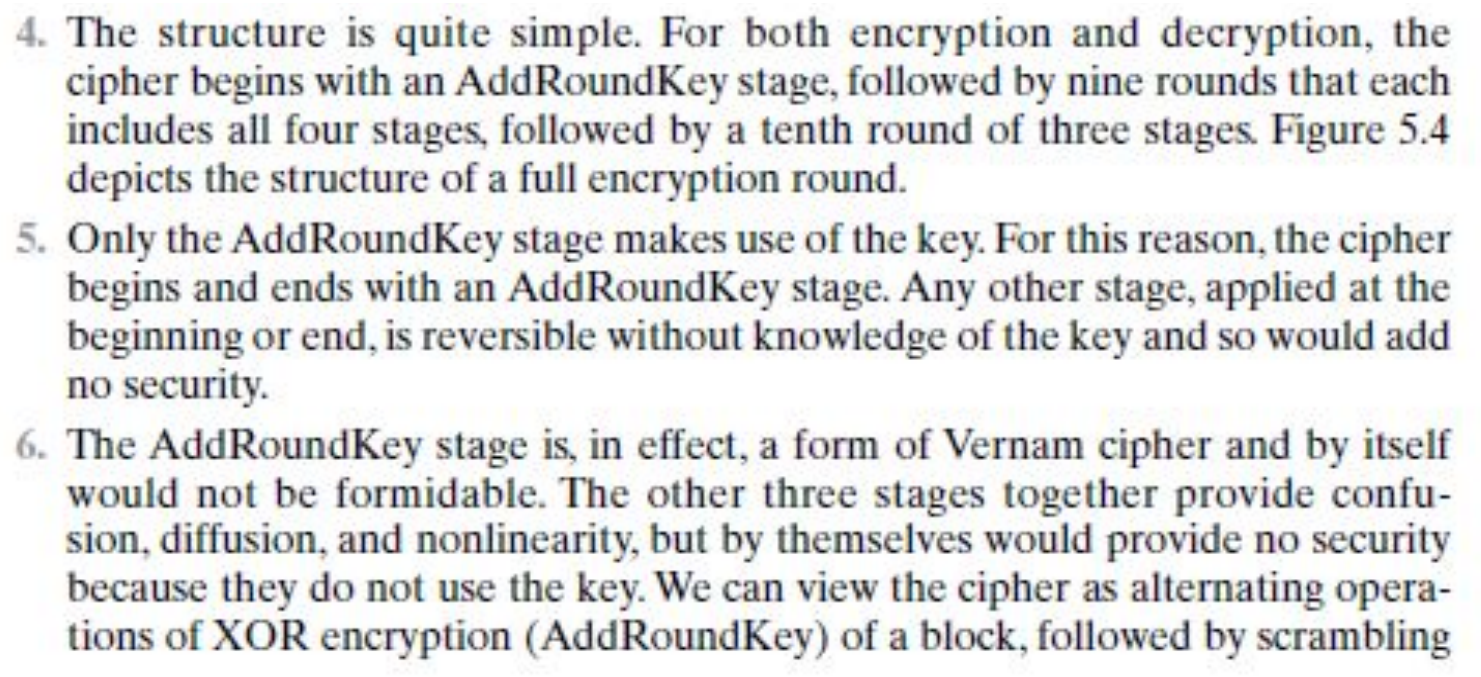
Figure 5.3 AES Encryption and Decryption

Advanced Encryption Standard(AES) Structure



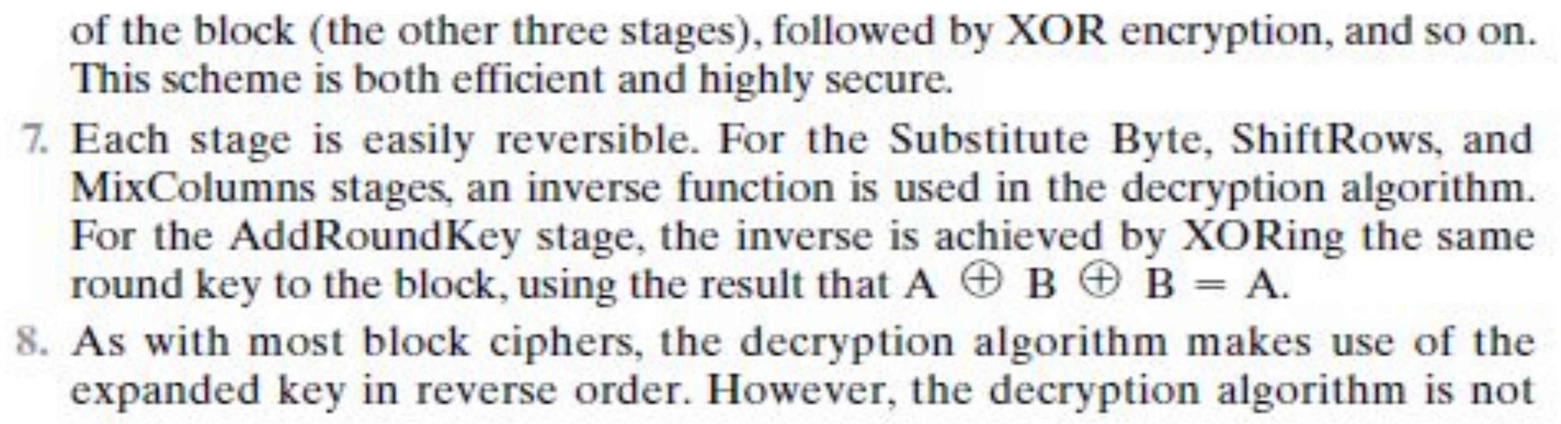
1. One noteworthy feature of this structure is that it is not a Feistel structure. Recall that, in the classic Feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. AES instead processes the entire data block as a single matrix during each round using substitutions and permutation.
2. The key that is provided as input is expanded into an array of forty-four 32-bit words, **w**[*i*]. Four distinct words (128 bits) serve as a round key for each round; these are indicated in Figure 5.3.
3. Four different stages are used, one of permutation and three of substitution:
   - **Substitute bytes:** Uses an S-box to perform a byte-by-byte substitution of the block
   - **ShiftRows:** A simple permutation
   - **MixColumns:** A substitution that makes use of arithmetic over GF(2^8)
   - **AddRoundKey:** A simple bitwise XOR of the current block with a portion of the expanded key

Advanced Encryption Standard(AES) Structure



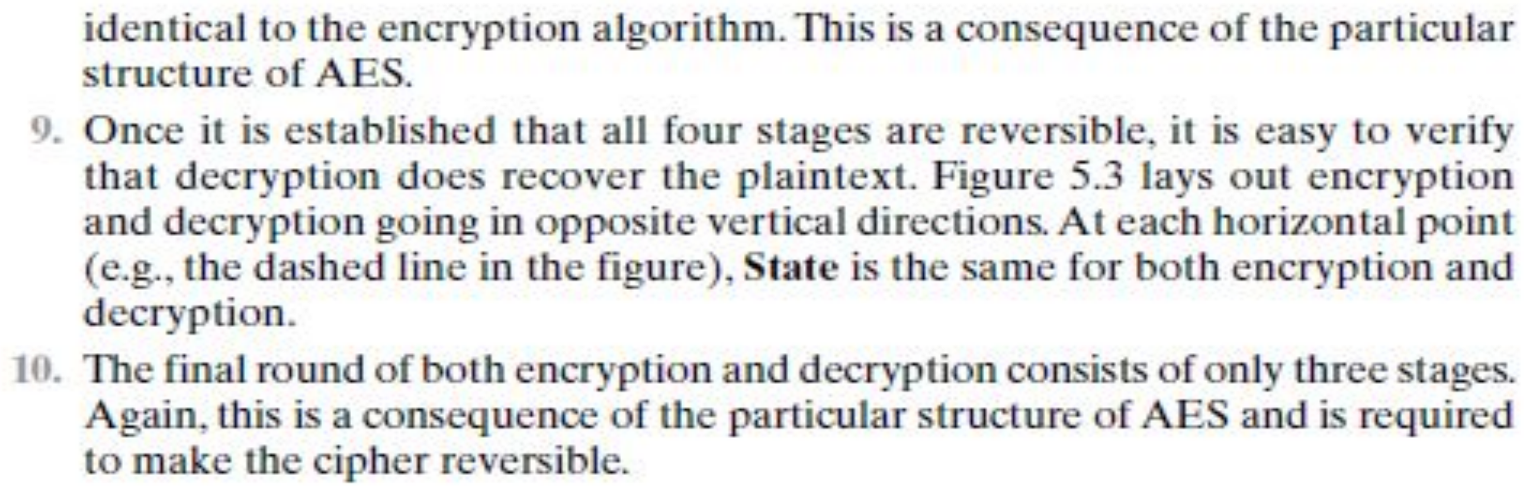
4. The structure is quite simple. For both encryption and decryption, the cipher begins with an AddRoundKey stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages. Figure 5.4 depicts the structure of a full encryption round.
5. Only the AddRoundKey stage makes use of the key. For this reason, the cipher begins and ends with an AddRoundKey stage. Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security.
6. The AddRoundKey stage is, in effect, a form of Vernam cipher and by itself would not be formidable. The other three stages together provide confusion, diffusion, and nonlinearity, but by themselves would provide no security because they do not use the key. We can view the cipher as alternating operations of XOR encryption (AddRoundKey) of a block, followed by scrambling

Advanced Encryption Standard(AES) Structure



of the block (the other three stages), followed by XOR encryption, and so on. This scheme is both efficient and highly secure.

7. Each stage is easily reversible. For the Substitute Byte, ShiftRows, and MixColumns stages, an inverse function is used in the decryption algorithm. For the AddRoundKey stage, the inverse is achieved by XORing the same round key to the block, using the result that $A \oplus B \oplus B = A$.
8. As with most block ciphers, the decryption algorithm makes use of the expanded key in reverse order. However, the decryption algorithm is not



identical to the encryption algorithm. This is a consequence of the particular structure of AES.

9. Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext. Figure 5.3 lays out encryption and decryption going in opposite vertical directions. At each horizontal point (e.g., the dashed line in the figure), **State** is the same for both encryption and decryption.
10. The final round of both encryption and decryption consists of only three stages. Again, this is a consequence of the particular structure of AES and is required to make the cipher reversible.

Advanced Encryption Standard(AES) Structure

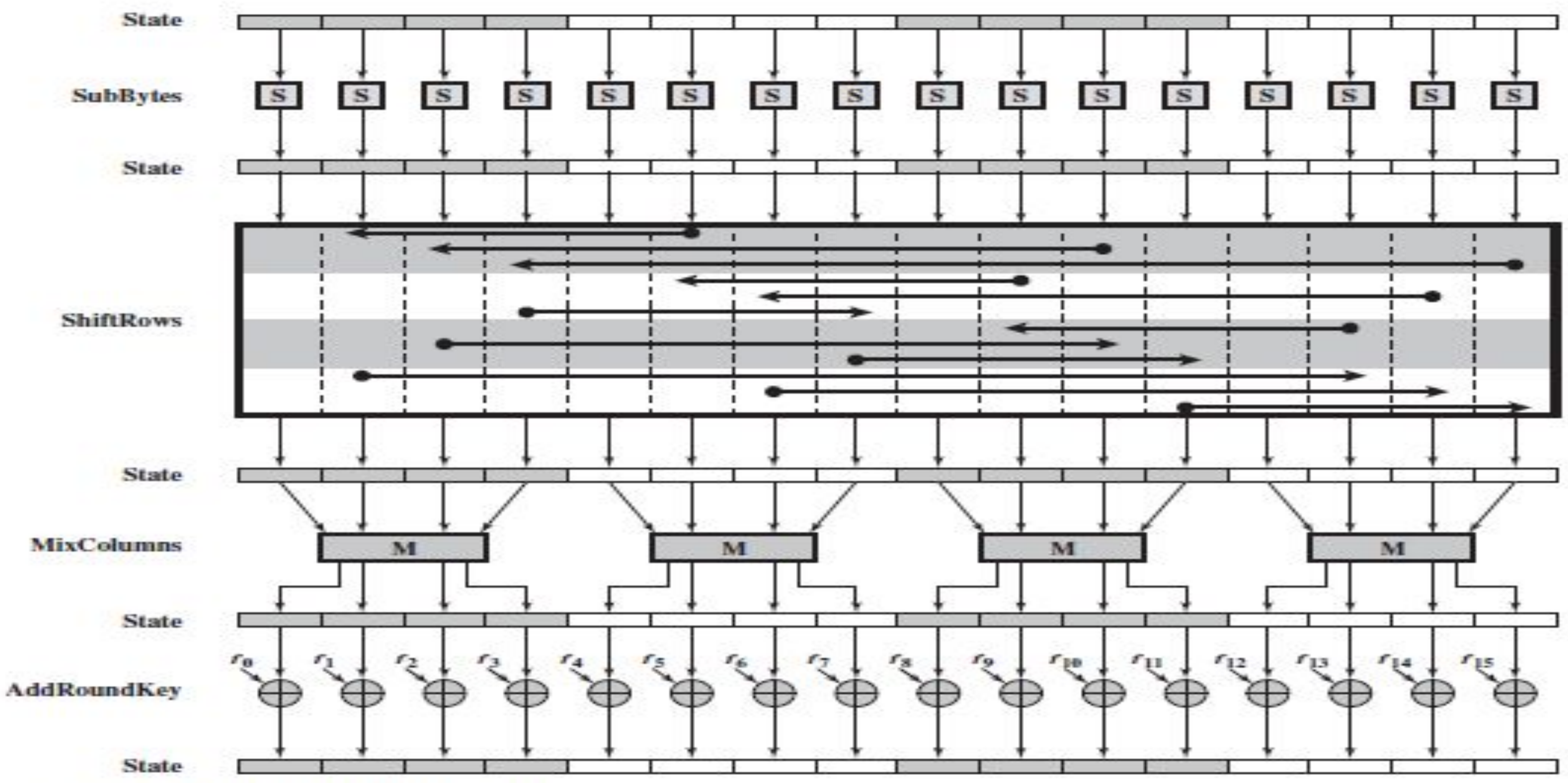
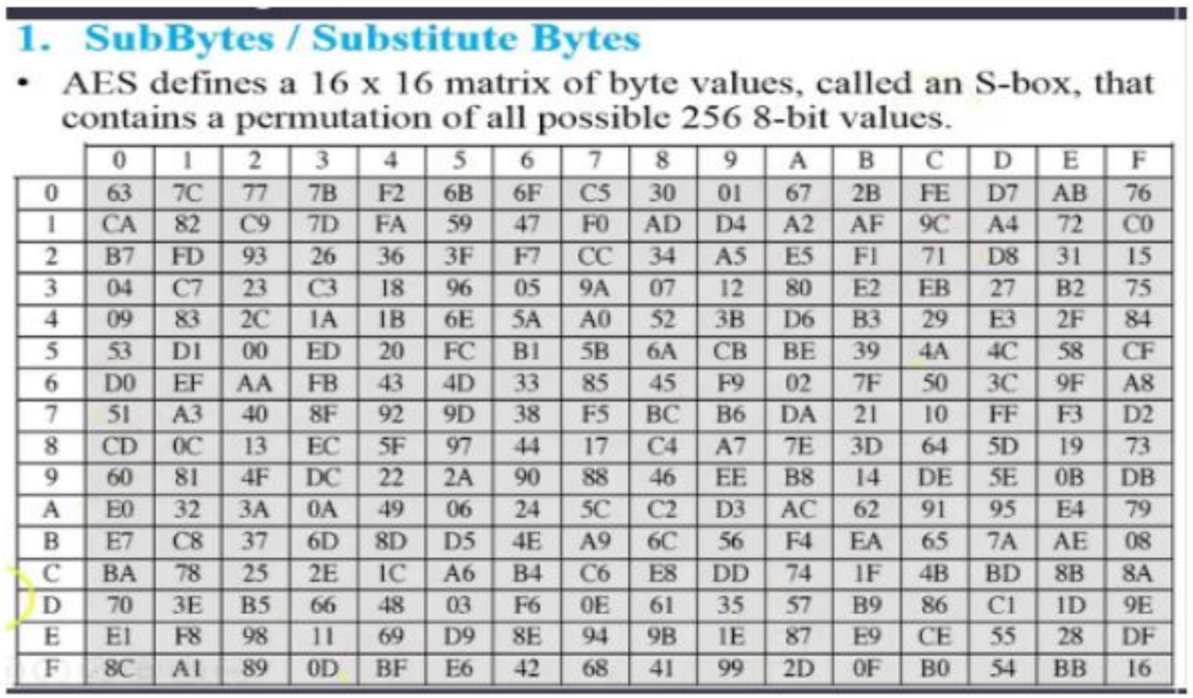


Figure 5.4 AES Encryption Round

Advanced Encryption Standard(AES) Transformations

AES Substitute bytes:-

- First transformation function of AES is called substitute bytes(sub bytes).
- It is also known as forward substitute bytes transformation.
- AES defines 16 X16 matrix of byte values ,called an S-box that contains permutation of all possible 256 8-bit values.
- These row and column values serve as indexes into the S-box to select a unique 8 bit output value.



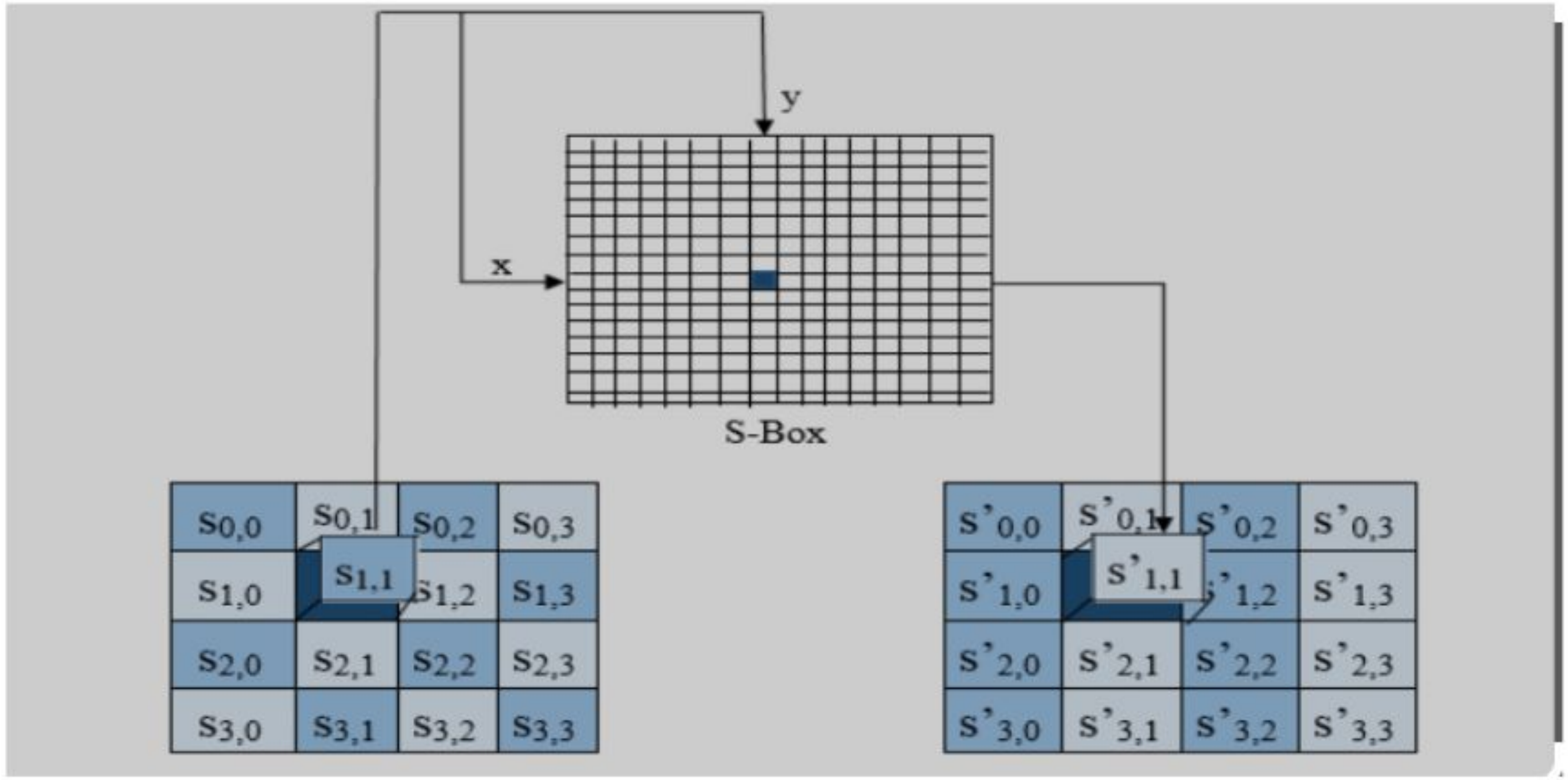
1. SubBytes / Substitute Bytes

- AES defines a 16 x 16 matrix of byte values, called an S-box, that contains a permutation of all possible 256 8-bit values.

	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

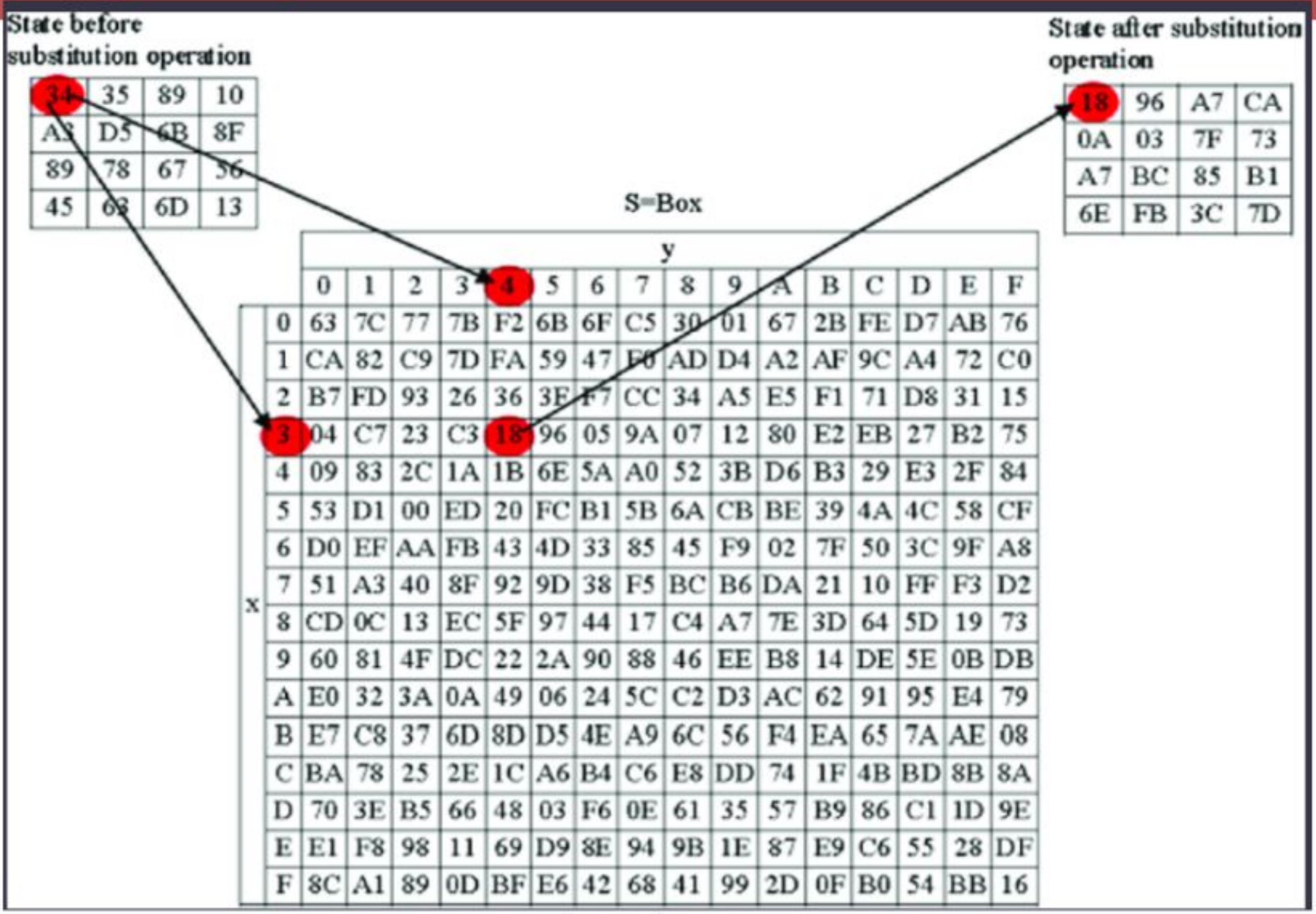
Advanced Encryption Standard(AES) Transformations

AES Substitute Bytes



Advanced Encryption Standard(AES) Transformations

AES Substitute Bytes Example:



State before substitution operation

34	35	89	10
A3	D5	6B	8F
89	78	67	56
45	63	6D	13

State after substitution operation

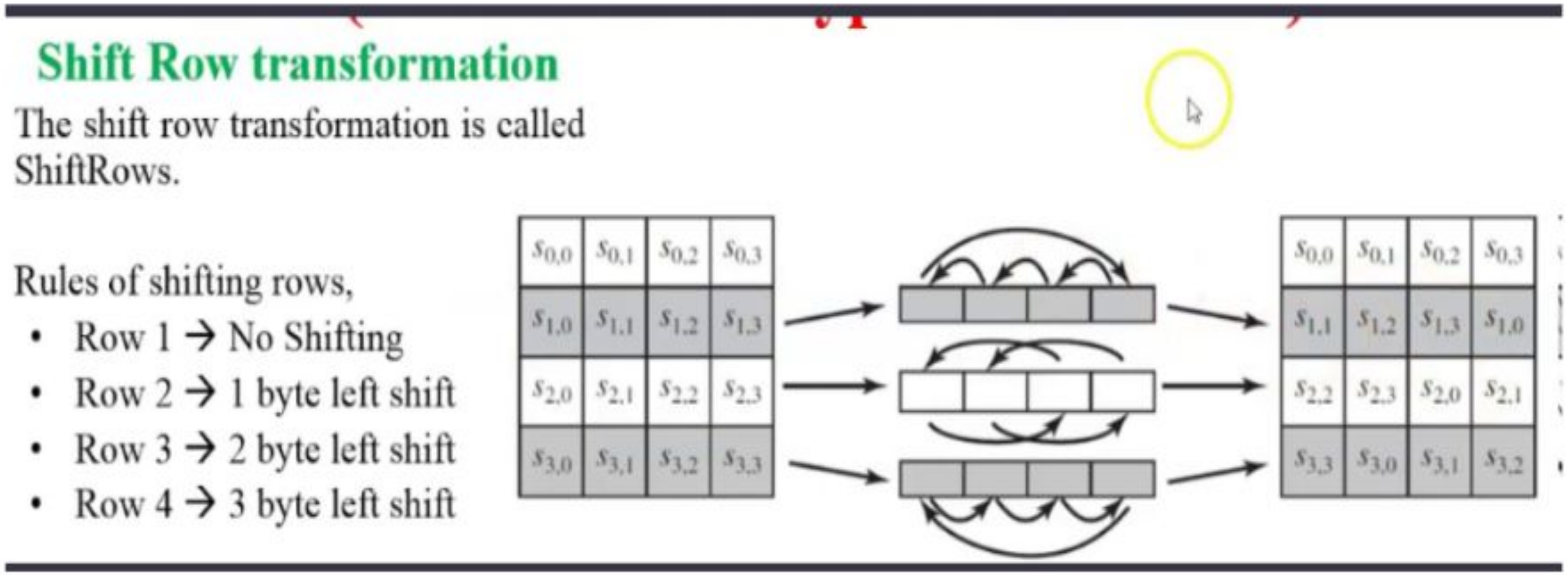
18	96	A7	CA
0A	03	7F	73
A7	BC	85	B1
6E	FB	3C	7D

S=Box

x \ y	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	3E	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	C6	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Advanced Encryption Standard(AES) Transformations

AES Shift Rows



Shift Row transformation

The shift row transformation is called ShiftRows.

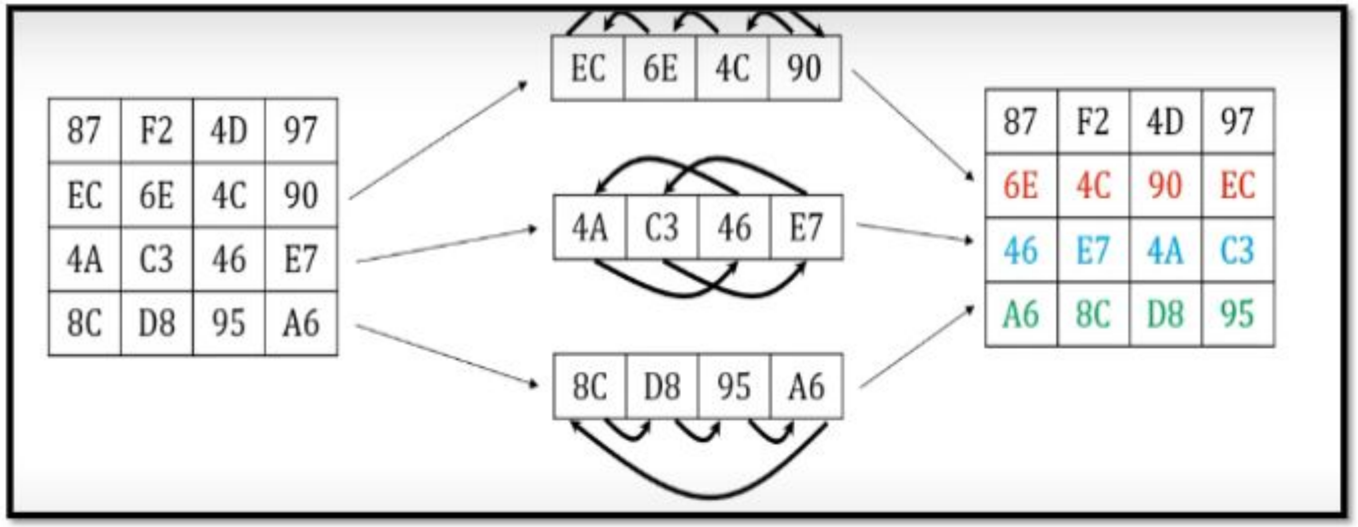
Rules of shifting rows,

- Row 1 → No Shifting
- Row 2 → 1 byte left shift
- Row 3 → 2 byte left shift
- Row 4 → 3 byte left shift

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{bmatrix}$$

Advanced Encryption Standard(AES) Transformations

AES Shift Rows

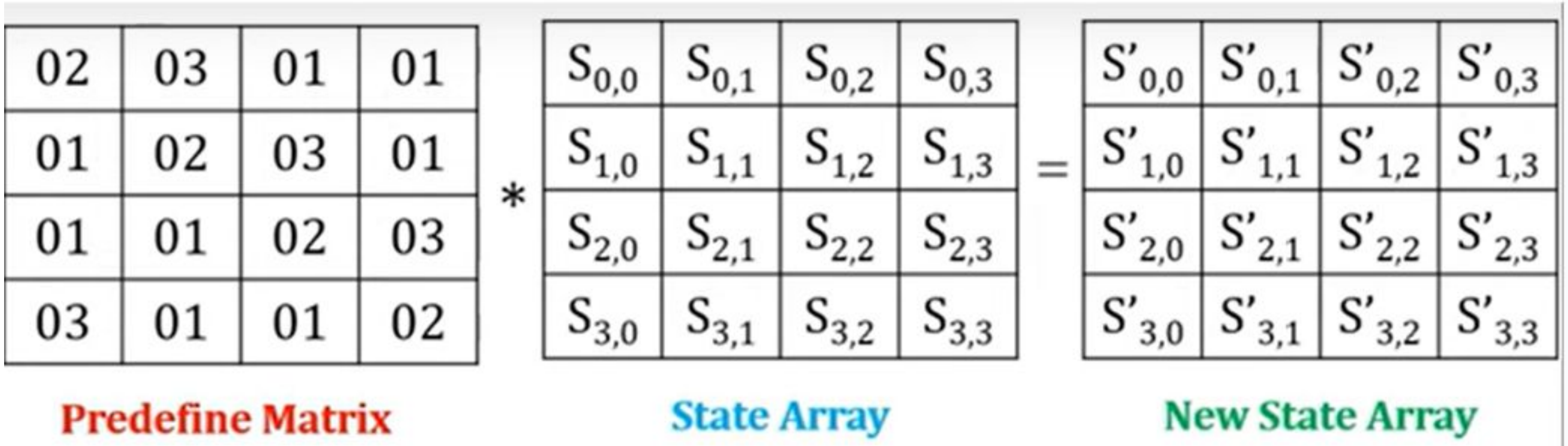


- Row 1 → No Shifting
- Row 2 → 1 byte circular left shift
- Row 3 → 2 byte circular left shift
- Row 4 → 3 byte circular left shift

Advanced Encryption Standard(AES) Transformations

AES Mix Columns:

Mix Columns operates on each column individually.



$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} * \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

State Array = Output of Shift Rows function.

Advanced Encryption Standard(AES) Transformations

Mix Columns operates on each column individually.

$$S'_{0,j} = (2 * S_{0,j}) \oplus (3 * S_{1,j}) \oplus S_{2,j} \oplus S_{3,j}$$

$$S'_{1,j} = S_{0,j} \oplus (2 * S_{1,j}) \oplus (3 * S_{2,j}) \oplus S_{3,j}$$

$$S'_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (2 * S_{2,j}) \oplus (3 * S_{3,j})$$

$$S'_{3,j} = (3 * S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 * S_{3,j})$$

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} = \begin{bmatrix} ? & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\}$$

Advanced Encryption Standard(AES) Transformations

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} = \begin{bmatrix} ? & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

$$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\}$$

$02 = 0000\ 0010 = X$

$87 = 1000\ 0111 = X^7 + X^2 + X + 1$

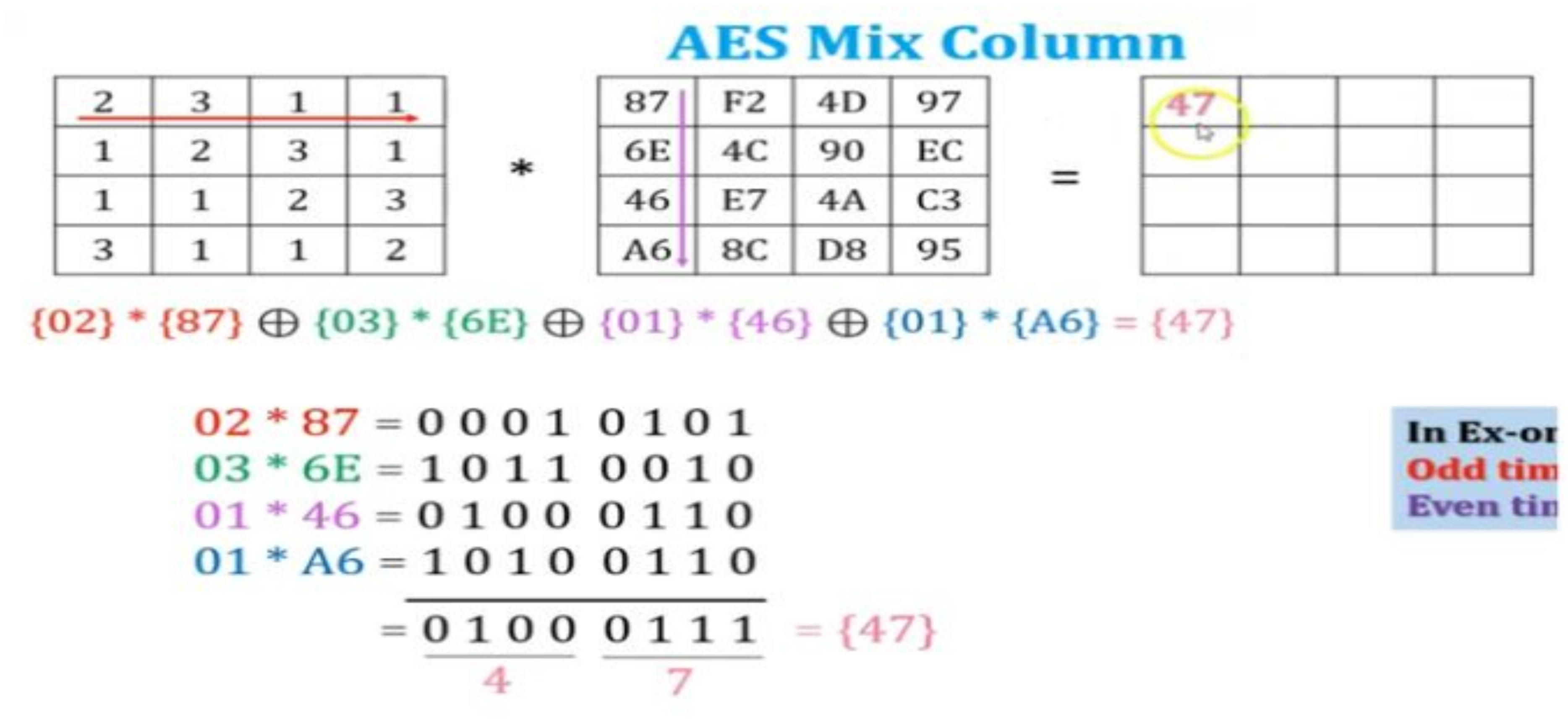
$$\begin{aligned} 02 * 87 &= X * (X^7 + X^2 + X + 1) \\ &= X^8 + X^3 + X^2 + X \\ &= X^4 + \cancel{X^3} + \cancel{X} + 1 + \cancel{X^3} + X^2 + \cancel{X} \\ &= X^4 + X^2 + 1 \\ &= 0001\ 0101 \end{aligned}$$

$03 = 0000\ 0011 = X + 1$

$6E = 0110\ 1110 = X^6 + X^5 + X^3 + X^2 + X$

$$\begin{aligned} 03 * 6E &= (X+1) * (X^6 + X^5 + X^3 + X^2 + X) \\ &= X^7 + \cancel{X^6} + X^4 + \cancel{X^3} + \cancel{X^2} + \cancel{X^6} + X^5 + \cancel{X^3} + \cancel{X^2} + X \\ &= X^7 + X^5 + X^4 + X \end{aligned}$$

Advanced Encryption Standard(AES) Transformations



AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

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

=

47			

$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\} = \{47\}$

```
02 * 87 = 0 0 0 1  0 1 0 1
03 * 6E = 1 0 1 1  0 0 1 0
01 * 46 = 0 1 0 0  0 1 1 0
01 * A6 = 1 0 1 0  0 1 1 0
        ___________________
        = 0 1 0 0  0 1 1 1  = {47}
            4        7
```

In Ex-or
Odd tim
Even tin

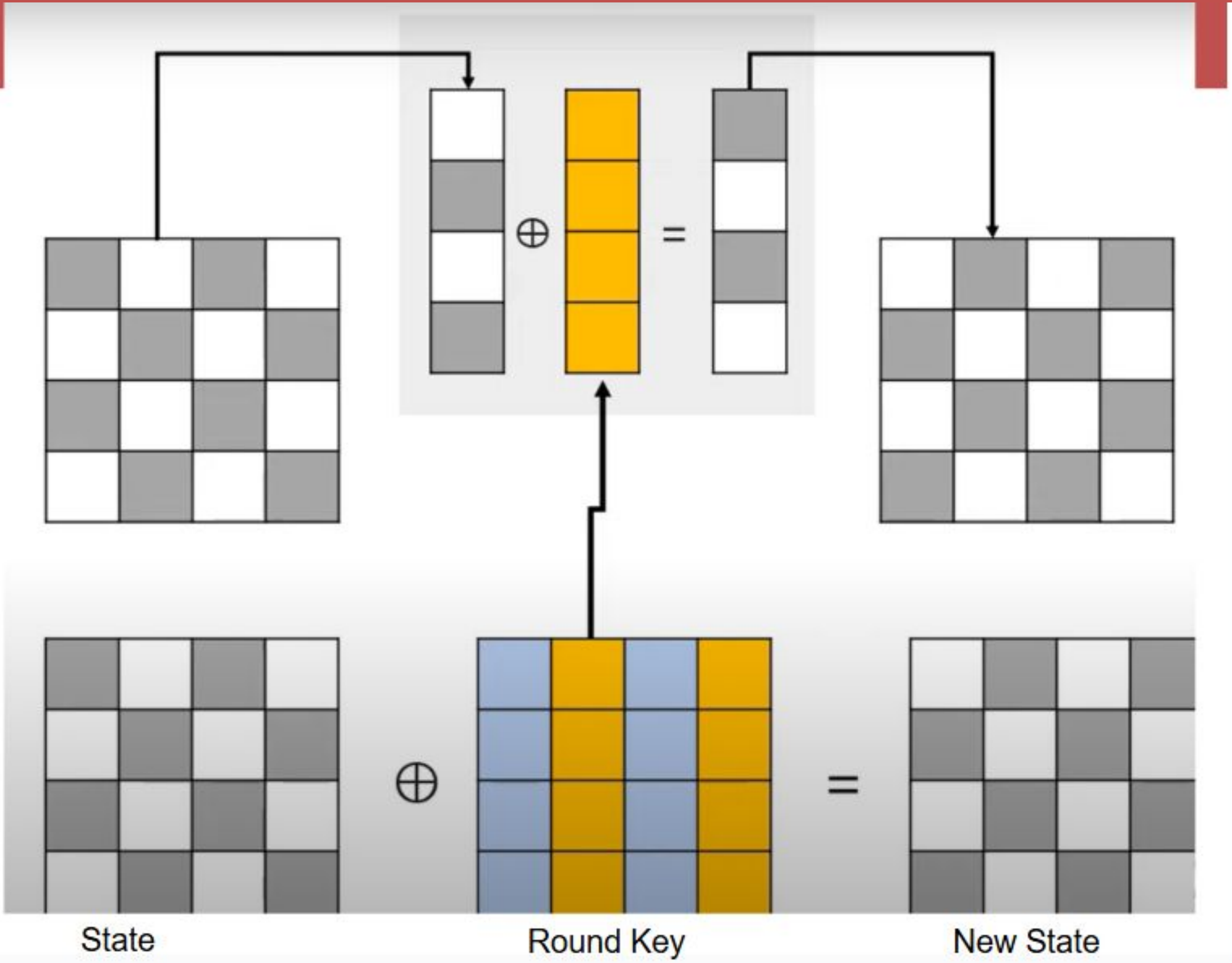
Advanced Encryption Standard(AES) Transformations

AES Mix Columns

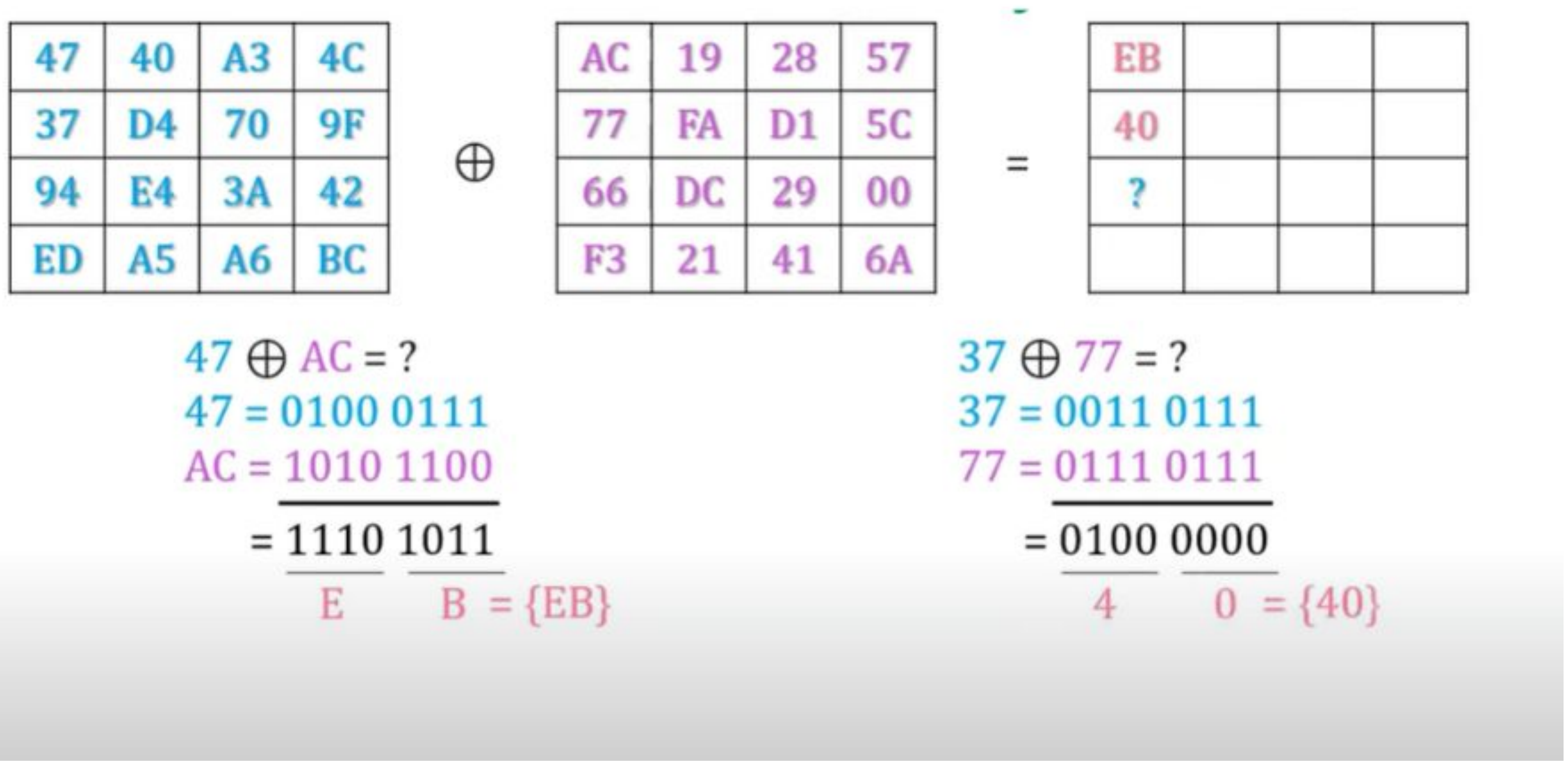
$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} = \begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}$$

Advanced Encryption Standard(AES) Transformations

AES Add Round Key:



Advanced Encryption Standard(AES) Transformations



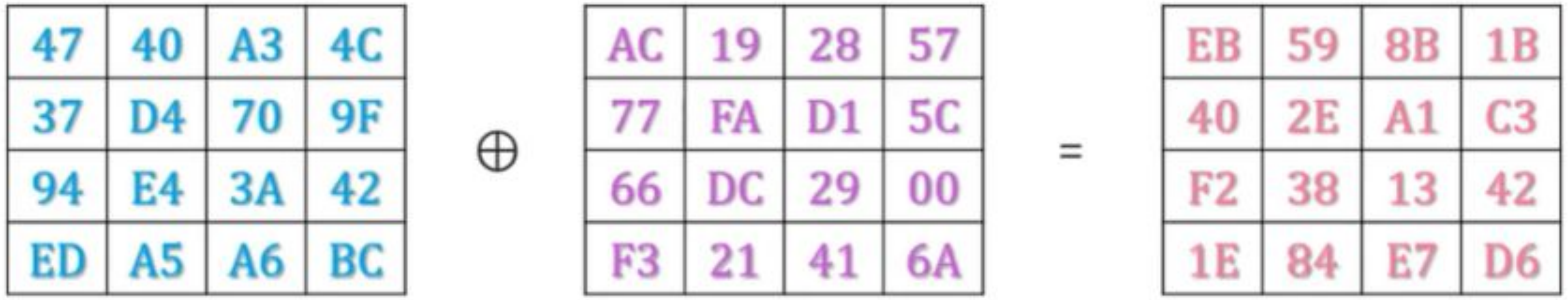
$$\begin{array}{|c|c|c|c|}\hline 47 & 40 & A3 & 4C \\ \hline 37 & D4 & 70 & 9F \\ \hline 94 & E4 & 3A & 42 \\ \hline ED & A5 & A6 & BC \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline AC & 19 & 28 & 57 \\ \hline 77 & FA & D1 & 5C \\ \hline 66 & DC & 29 & 00 \\ \hline F3 & 21 & 41 & 6A \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline EB & & & \\ \hline 40 & & & \\ \hline ? & & & \\ \hline & & & \\ \hline \end{array}$$

$$47 \oplus AC = ?$$
$$\begin{array}{r} 47 = 0100\ 0111 \\ AC = \underline{1010\ 1100} \\ = \underbrace{1110}_{E}\ \underbrace{1011}_{B} = \{EB\} \end{array}$$

$$37 \oplus 77 = ?$$
$$\begin{array}{r} 37 = 0011\ 0111 \\ 77 = \underline{0111\ 0111} \\ = \underbrace{0100}_{4}\ \underbrace{0000}_{0} = \{40\} \end{array}$$

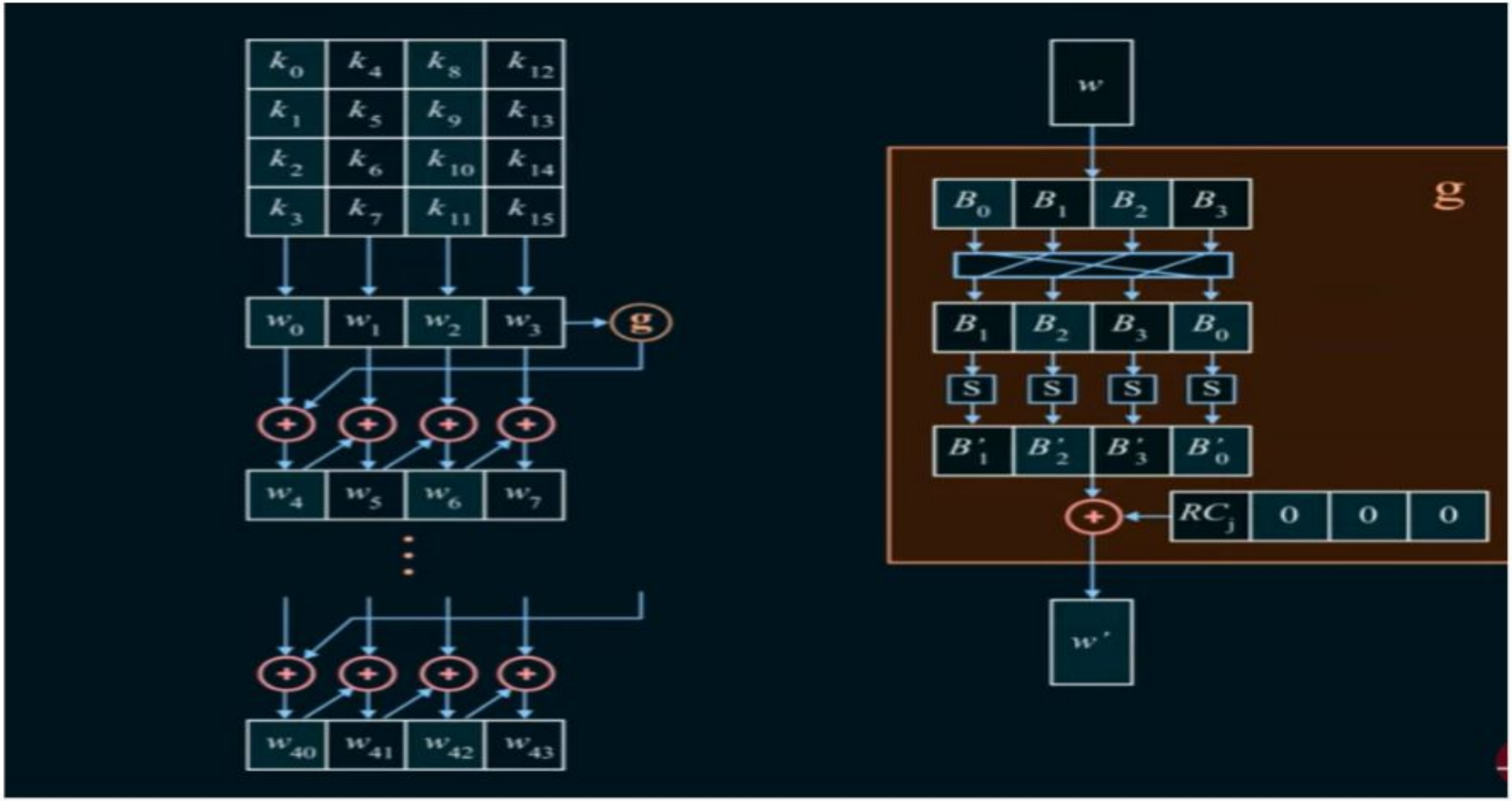
Advanced Encryption Standard(AES) Transformations

AES Add Round Key



$$\begin{array}{|c|c|c|c|}\hline 47 & 40 & A3 & 4C \\ \hline 37 & D4 & 70 & 9F \\ \hline 94 & E4 & 3A & 42 \\ \hline ED & A5 & A6 & BC \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|}\hline AC & 19 & 28 & 57 \\ \hline 77 & FA & D1 & 5C \\ \hline 66 & DC & 29 & 00 \\ \hline F3 & 21 & 41 & 6A \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline EB & 59 & 8B & 1B \\ \hline 40 & 2E & A1 & C3 \\ \hline F2 & 38 & 13 & 42 \\ \hline 1E & 84 & E7 & D6 \\ \hline \end{array}$$

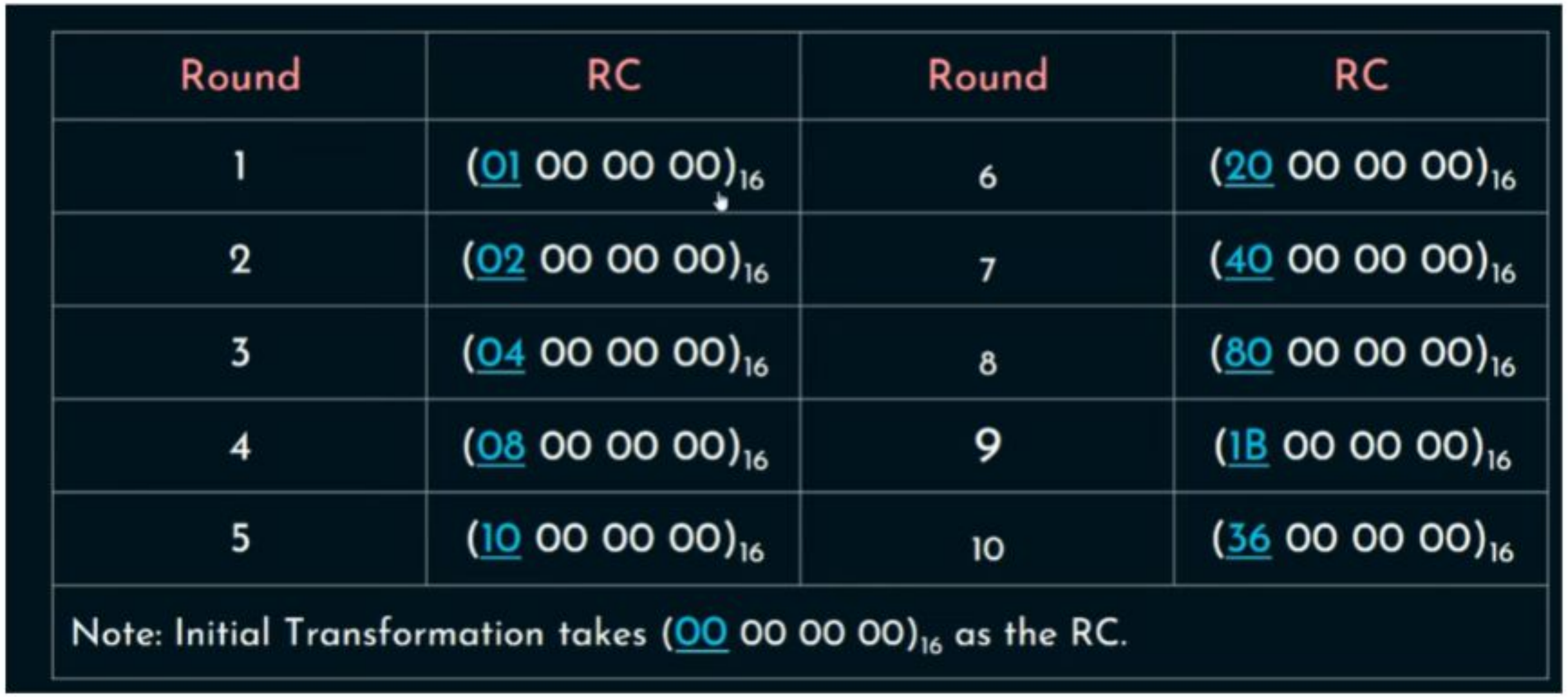
Advanced Encryption Standard(AES) Key Expansion



k_0 k_4 k_8 k_{12}
k_1 k_5 k_9 k_{13}
k_2 k_6 k_{10} k_{14}
k_3 k_7 k_{11} k_{15}
w_0 w_1 w_2 w_3
g
w_4 w_5 w_6 w_7
w_{40} w_{41} w_{42} w_{43}
w
g
B_0 B_1 B_2 B_3
B_1 B_2 B_3 B_0
S S S S
B'_1 B'_2 B'_3 B'_0
RC_j 0 0 0
w'

Advanced Encryption Standard(AES) Key Expansion

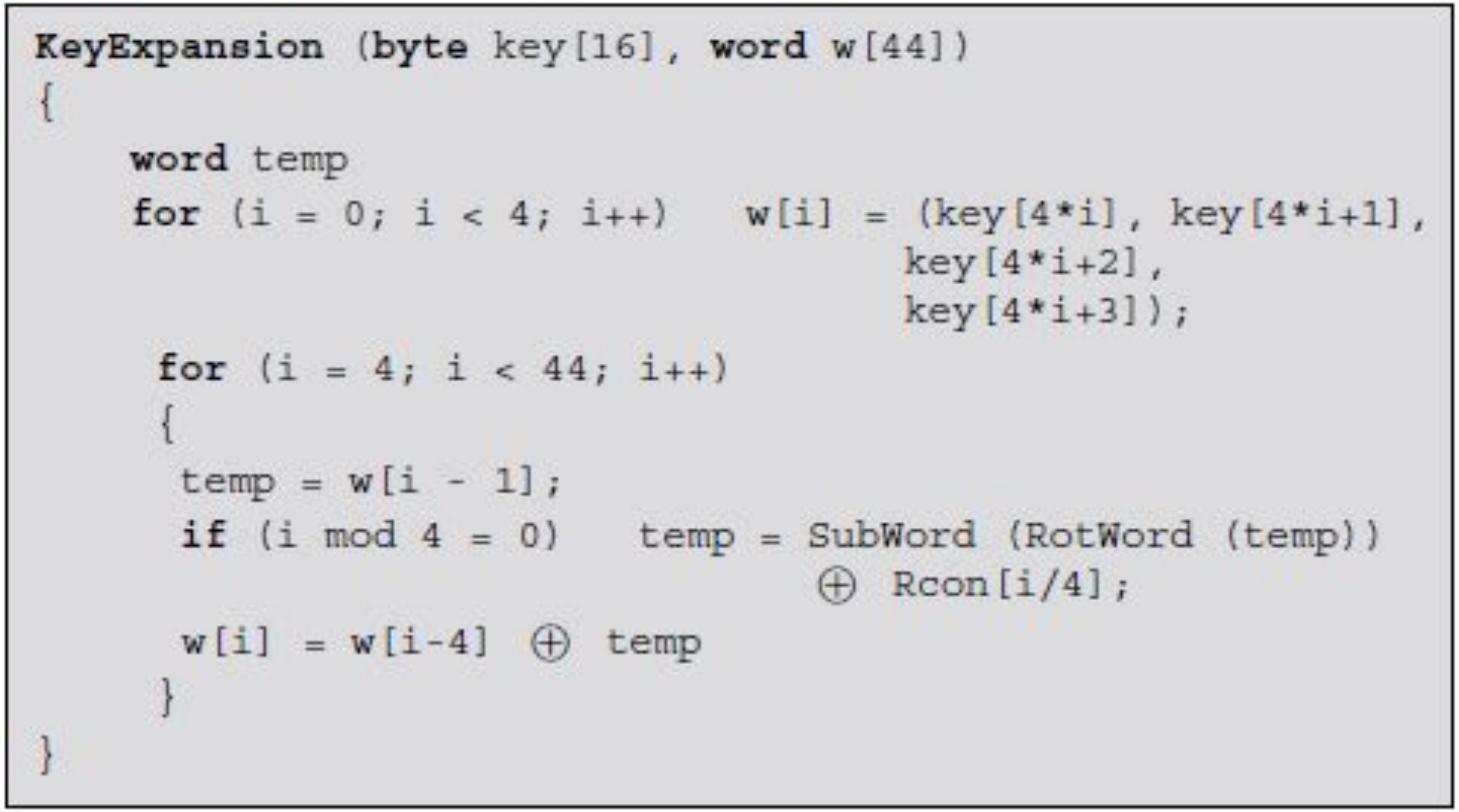
Round Constant in AES



Round	RC	Round	RC
1	$(\underline{01}\ 00\ 00\ 00)_{16}$	6	$(\underline{20}\ 00\ 00\ 00)_{16}$
2	$(\underline{02}\ 00\ 00\ 00)_{16}$	7	$(\underline{40}\ 00\ 00\ 00)_{16}$
3	$(\underline{04}\ 00\ 00\ 00)_{16}$	8	$(\underline{80}\ 00\ 00\ 00)_{16}$
4	$(\underline{08}\ 00\ 00\ 00)_{16}$	9	$(\underline{1B}\ 00\ 00\ 00)_{16}$
5	$(\underline{10}\ 00\ 00\ 00)_{16}$	10	$(\underline{36}\ 00\ 00\ 00)_{16}$

Note: Initial Transformation takes $(\underline{00}\ 00\ 00\ 00)_{16}$ as the RC.

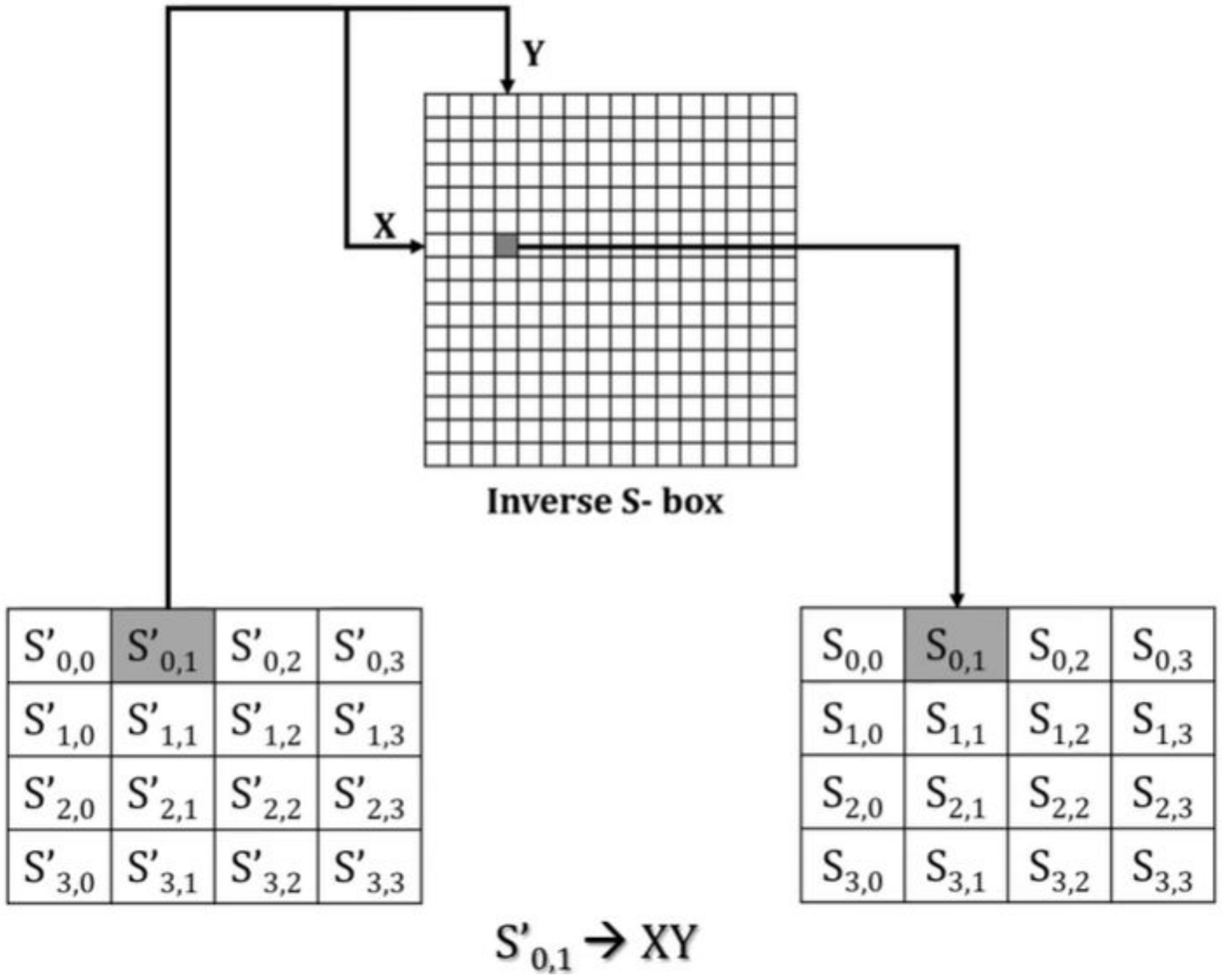
Advanced Encryption Standard(AES) Key Expansion



```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++)   w[i] = (key[4*i], key[4*i+1],
                                      key[4*i+2],
                                      key[4*i+3]);

     for (i = 4; i < 44; i++)
     {
      temp = w[i - 1];
      if (i mod 4 = 0)   temp = SubWord (RotWord (temp))
                                ⊕ Rcon[i/4];
      w[i] = w[i-4] ⊕ temp
     }
}
```

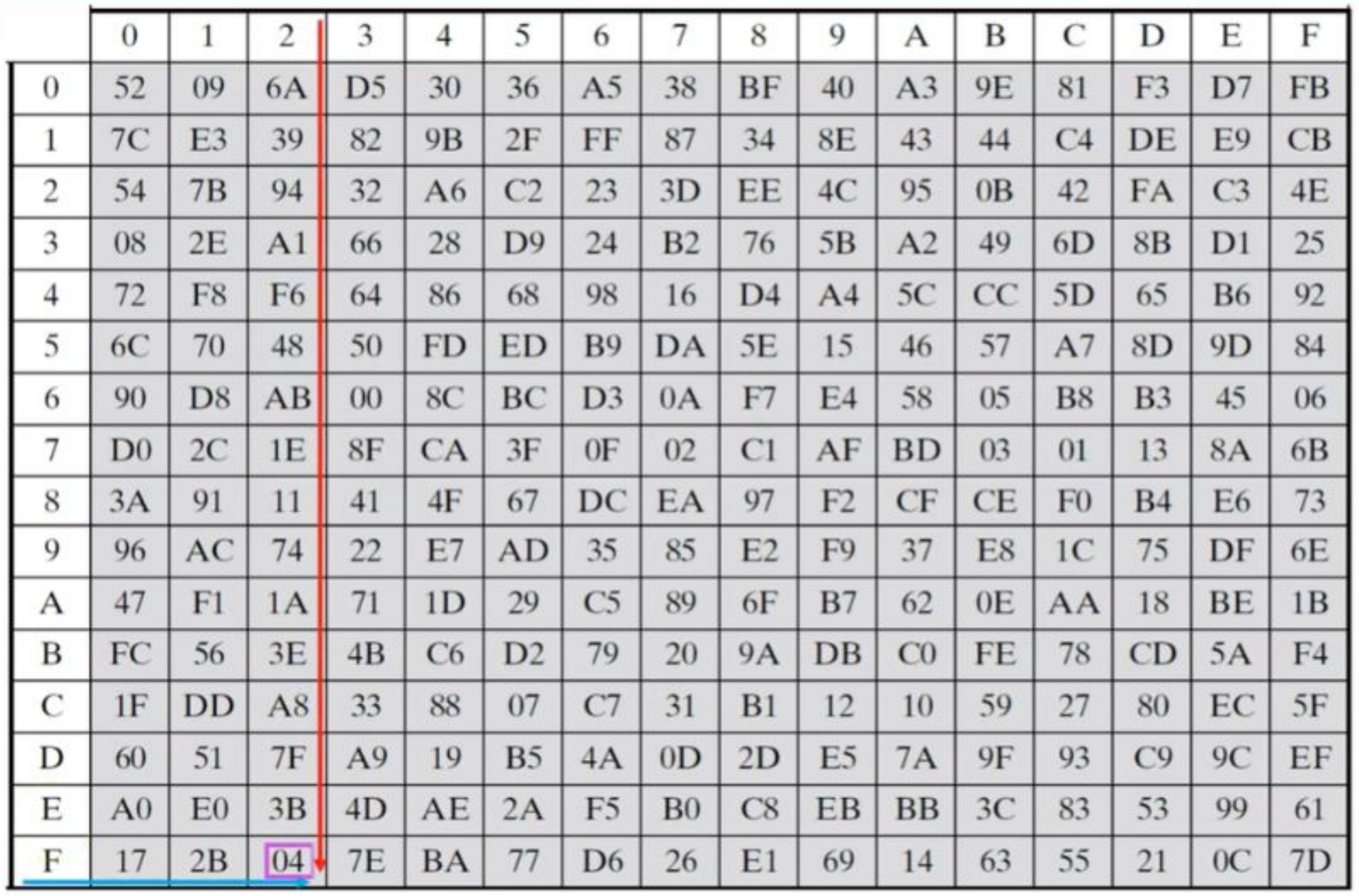
Advanced Encryption Standard(AES) Key Expansion

Inverse Sub Bytes

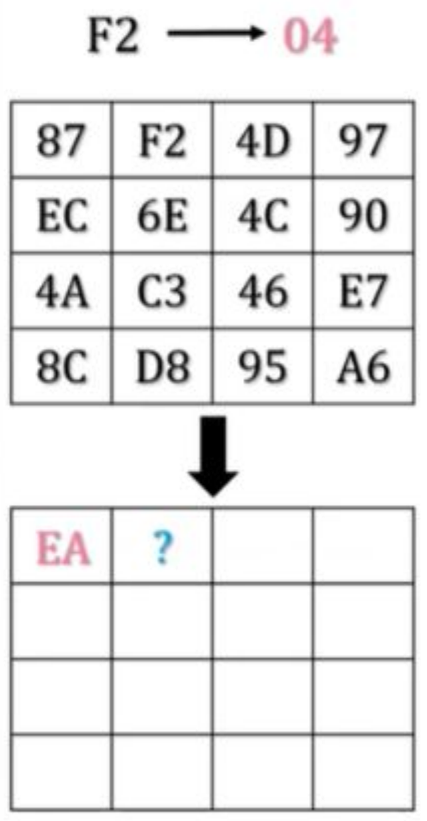


Advanced Encryption Standard(AES) Key Expansion

Inverse Sub Bytes



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D



$F2 \longrightarrow 04$

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

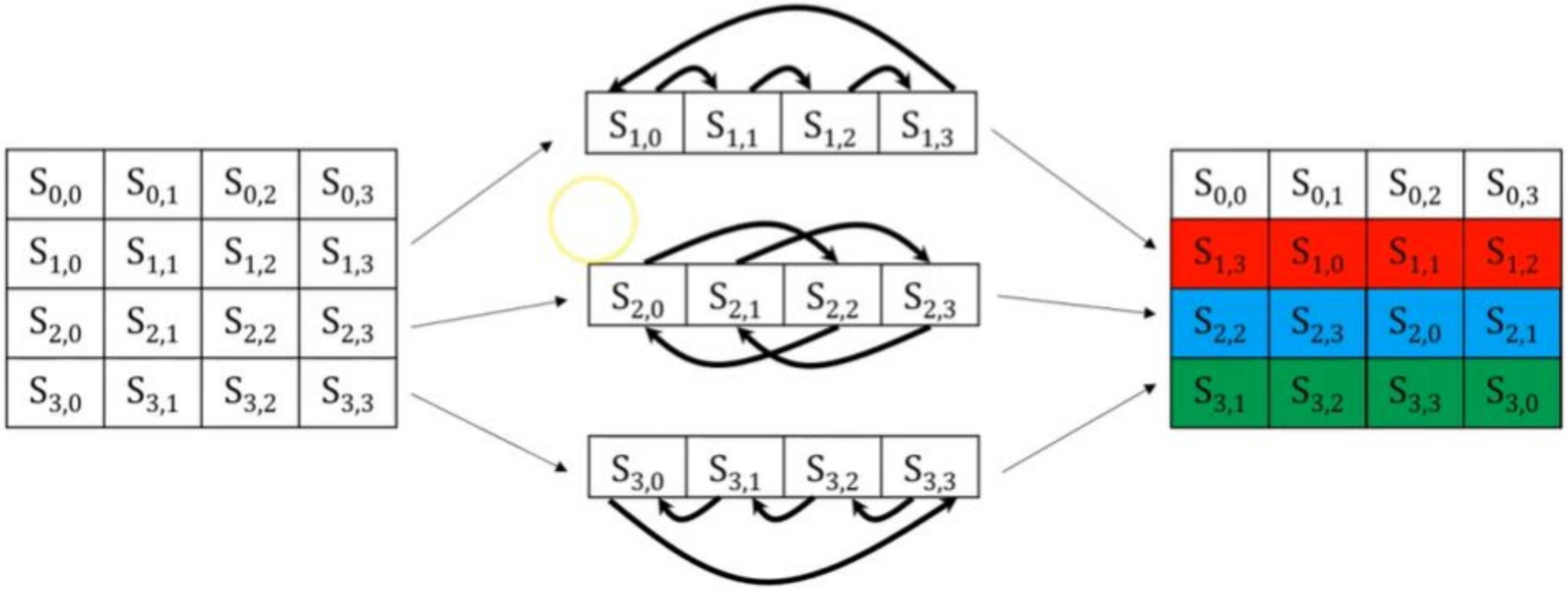
$\Downarrow$

EA	?		

Advanced Encryption Standard(AES) Key Expansion

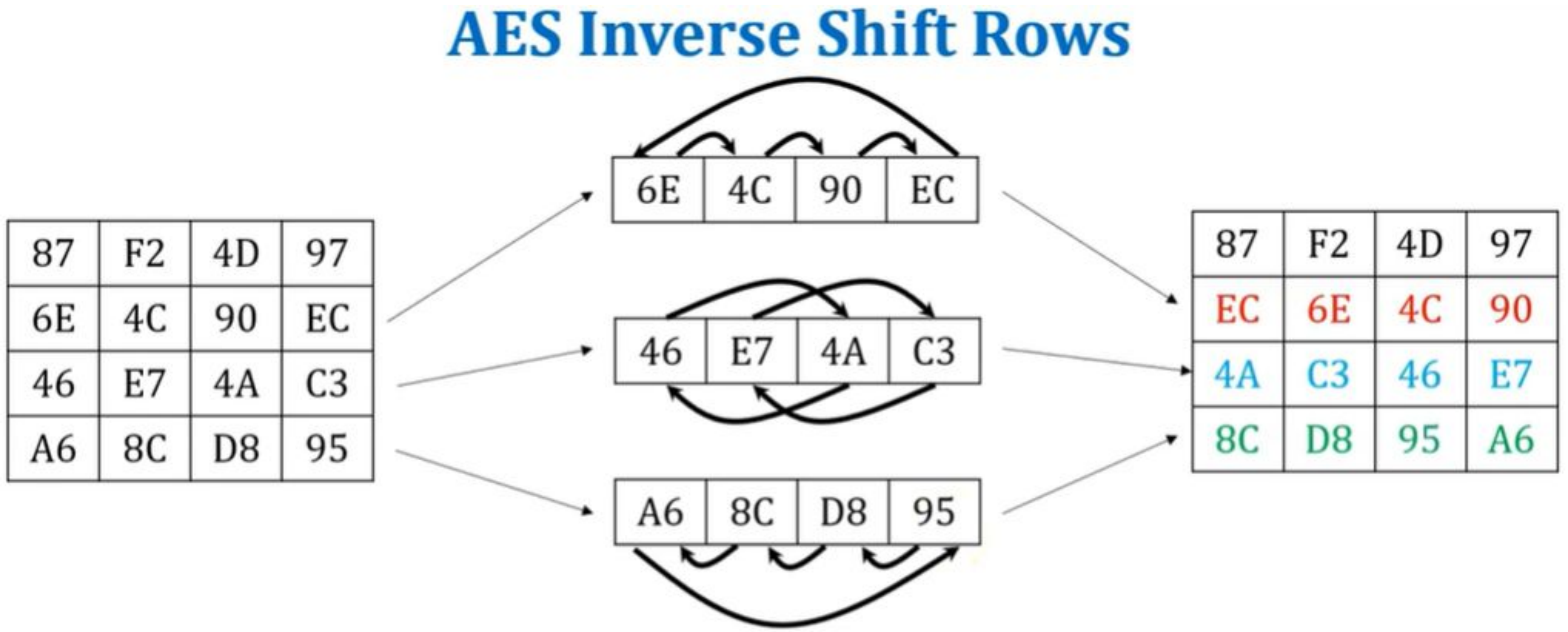
AES Inverse Shift Rows

- **Rules of Inverse shifting rows,**
  - Row 1 → No Shifting
  - Row 2 → 1 byte circular right shift
  - Row 3 → 2 byte circular right shift
  - Row 4 → 3 byte circular right shift



$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,3}$ $S_{1,0}$ $S_{1,1}$ $S_{1,2}$
$S_{2,2}$ $S_{2,3}$ $S_{2,0}$ $S_{2,1}$
$S_{3,1}$ $S_{3,2}$ $S_{3,3}$ $S_{3,0}$

Advanced Encryption Standard(AES) Key Expansion



Advanced Encryption Standard(AES) Key Expansion

Inverse Mix Columns

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} * \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

Predefine Matrix For decryption **State Array** **New State Array**

State Array = Output of previous transformation function in decryption process

$$S'_{0,j} = (0E * S_{0,j}) \oplus (0B * S_{1,j}) \oplus (0D * S_{2,j}) \oplus (09 * S_{3,j})$$

$$S'_{1,j} = (09 * S_{0,j}) \oplus (0E * S_{1,j}) \oplus (0B * S_{2,j}) \oplus (0D * S_{3,j})$$

$$S'_{2,j} = (0D * S_{0,j}) \oplus (09 * S_{1,j}) \oplus (0E * S_{2,j}) \oplus (0B * S_{3,j})$$

$$S'_{3,j} = (0B * S_{0,j}) \oplus (0D * S_{1,j}) \oplus (09 * S_{2,j}) \oplus (0E * S_{3,j})$$

Advanced Encryption Standard(AES) Key Expansion

Inverse Mix Columns

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} * \begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix} = \begin{bmatrix} ? & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

$$\boxed{\{0E\} * \{47\}} \oplus \{0B\} * \{37\} \oplus \{0D\} * \{94\} \oplus \{09\} * \{ED\}$$

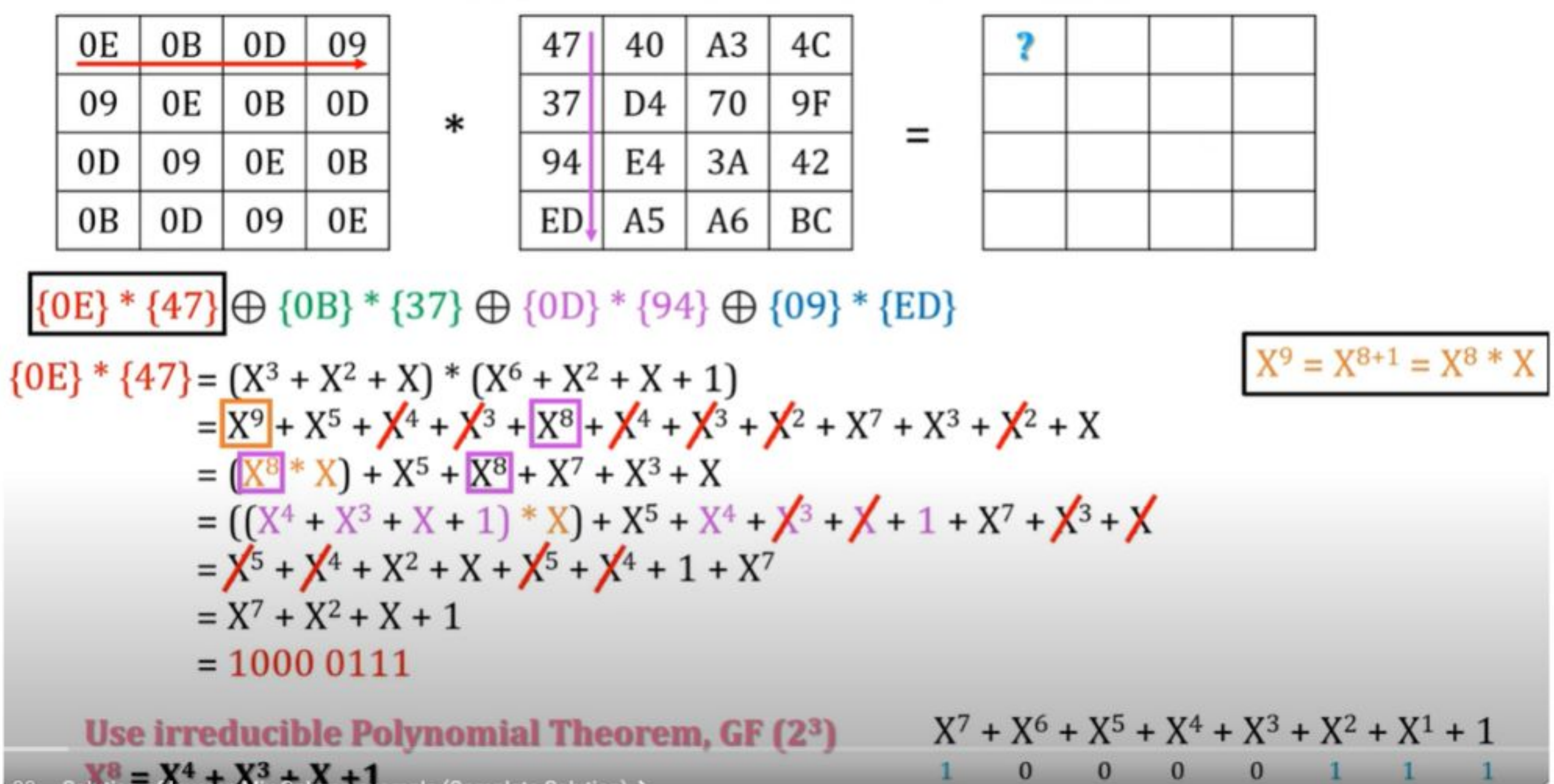
$$\begin{array}{cccccccc} X^7 + & X^6 + & X^5 + & X^4 + & \boxed{X^3} + & \boxed{X^2} + & \boxed{X^1} + & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & \boxed{1} & \boxed{1} & 0 \end{array}$$

$GF(2^3)$

$0E = \overbrace{0000\ 1110} = X^3 + X^2 + X$

$47 = 0100\ 0111$

Advanced Encryption Standard(AES) Key Expansion



0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

*

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

=

?			

$\boxed{\{0E\} * \{47\}} \oplus \{0B\} * \{37\} \oplus \{0D\} * \{94\} \oplus \{09\} * \{ED\}$

$\boxed{X^9 = X^{8+1} = X^8 * X}$

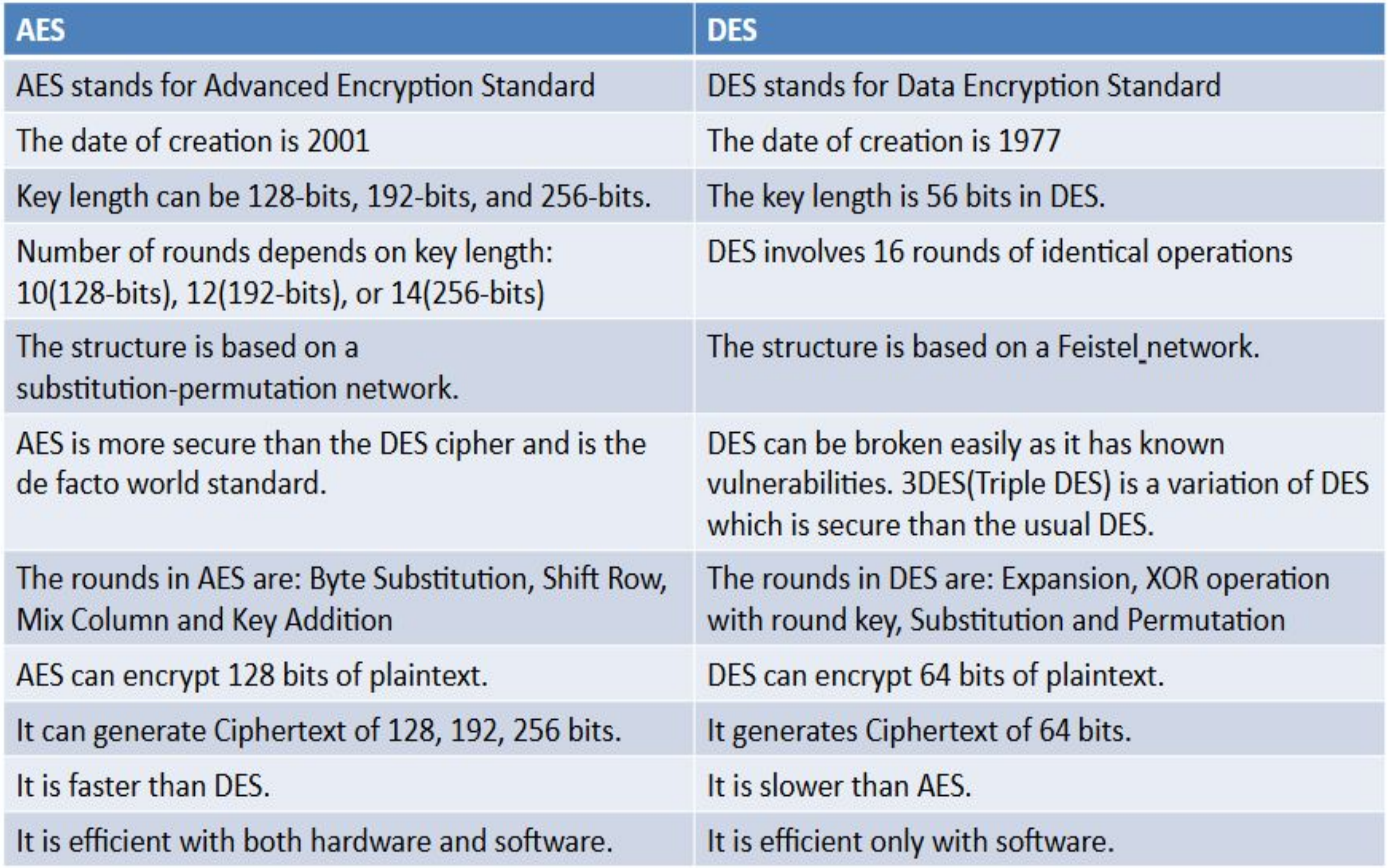
$$
\begin{aligned}
\{0E\} * \{47\} &= (X^3 + X^2 + X) * (X^6 + X^2 + X + 1) \\
&= \boxed{X^9} + X^5 + \cancel{X^4} + \cancel{X^3} + \boxed{X^8} + \cancel{X^4} + \cancel{X^3} + \cancel{X^2} + X^7 + X^3 + \cancel{X^2} + X \\
&= (\boxed{X^8} * X) + X^5 + \boxed{X^8} + X^7 + X^3 + X \\
&= ((X^4 + X^3 + X + 1) * X) + X^5 + X^4 + \cancel{X^3} + \cancel{X} + 1 + X^7 + \cancel{X^3} + \cancel{X} \\
&= \cancel{X^5} + \cancel{X^4} + X^2 + X + \cancel{X^5} + \cancel{X^4} + 1 + X^7 \\
&= X^7 + X^2 + X + 1 \\
&= 1000\ 0111
\end{aligned}
$$

Use irreducible Polynomial Theorem, GF (2^3)

$X^8 = X^4 + X^3 + X + 1$

X^7	X^6	X^5	X^4	X^3	X^2	X^1	1
1	0	0	0	0	1	1	1

Difference between AES and DES



AES	DES
AES stands for Advanced Encryption Standard	DES stands for Data Encryption Standard
The date of creation is 2001	The date of creation is 1977
Key length can be 128-bits, 192-bits, and 256-bits.	The key length is 56 bits in DES.
Number of rounds depends on key length: 10(128-bits), 12(192-bits), or 14(256-bits)	DES involves 16 rounds of identical operations
The structure is based on a substitution-permutation network.	The structure is based on a Feistel_network.
AES is more secure than the DES cipher and is the de facto world standard.	DES can be broken easily as it has known vulnerabilities. 3DES(Triple DES) is a variation of DES which is secure than the usual DES.
The rounds in AES are: Byte Substitution, Shift Row, Mix Column and Key Addition	The rounds in DES are: Expansion, XOR operation with round key, Substitution and Permutation
AES can encrypt 128 bits of plaintext.	DES can encrypt 64 bits of plaintext.
It can generate Ciphertext of 128, 192, 256 bits.	It generates Ciphertext of 64 bits.
It is faster than DES.	It is slower than AES.
It is efficient with both hardware and software.	It is efficient only with software.