UNIT-2_1

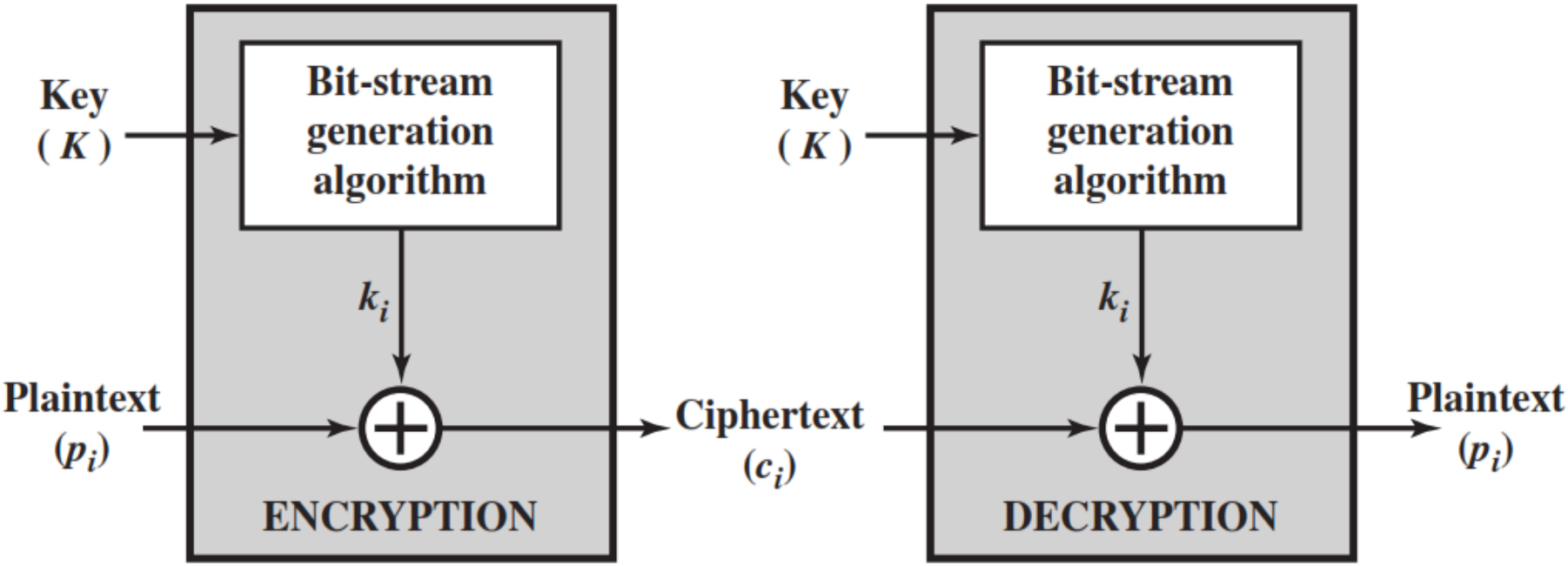
Stream ciphers and block ciphers

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

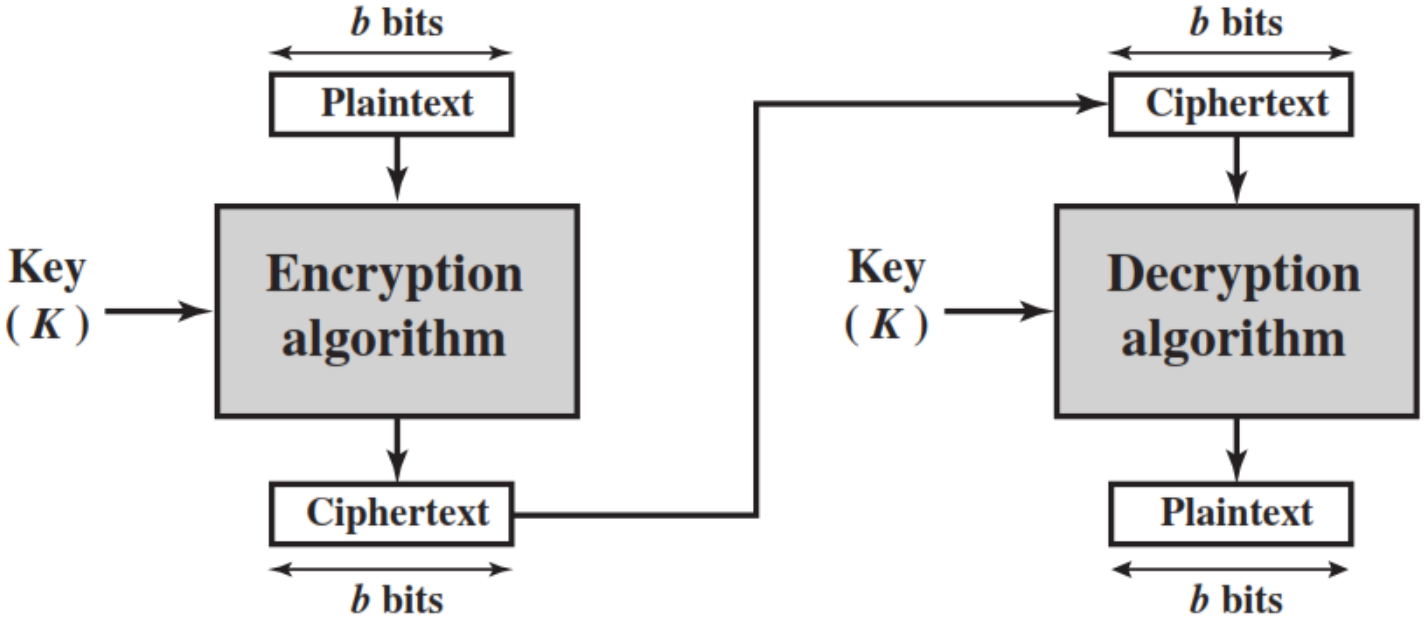
Stream Cipher

- 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

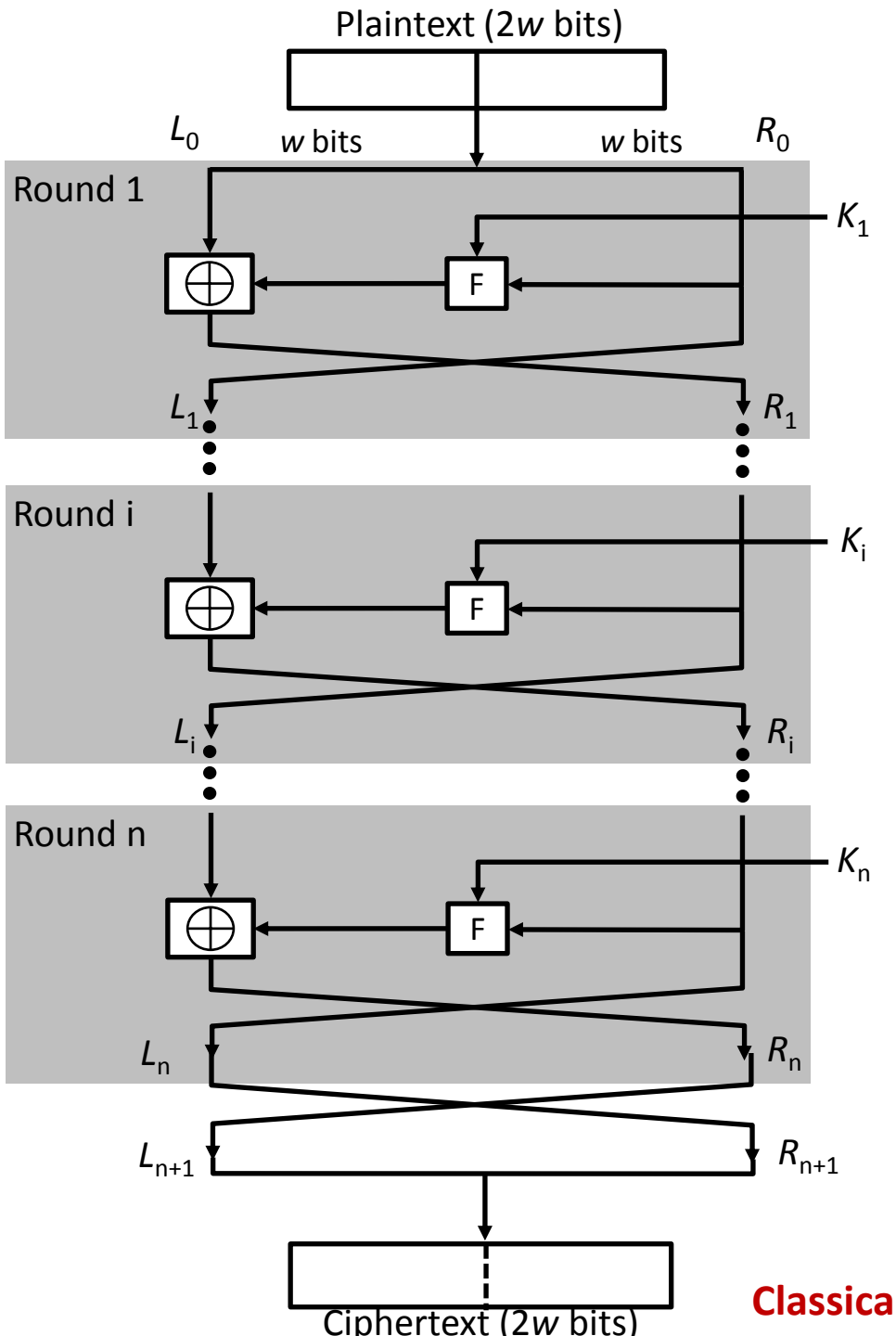
- A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Typically, a block size of **64 or 128** bits is used.
- Examples are Feistel Cipher, DES, Triple DES and AES



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



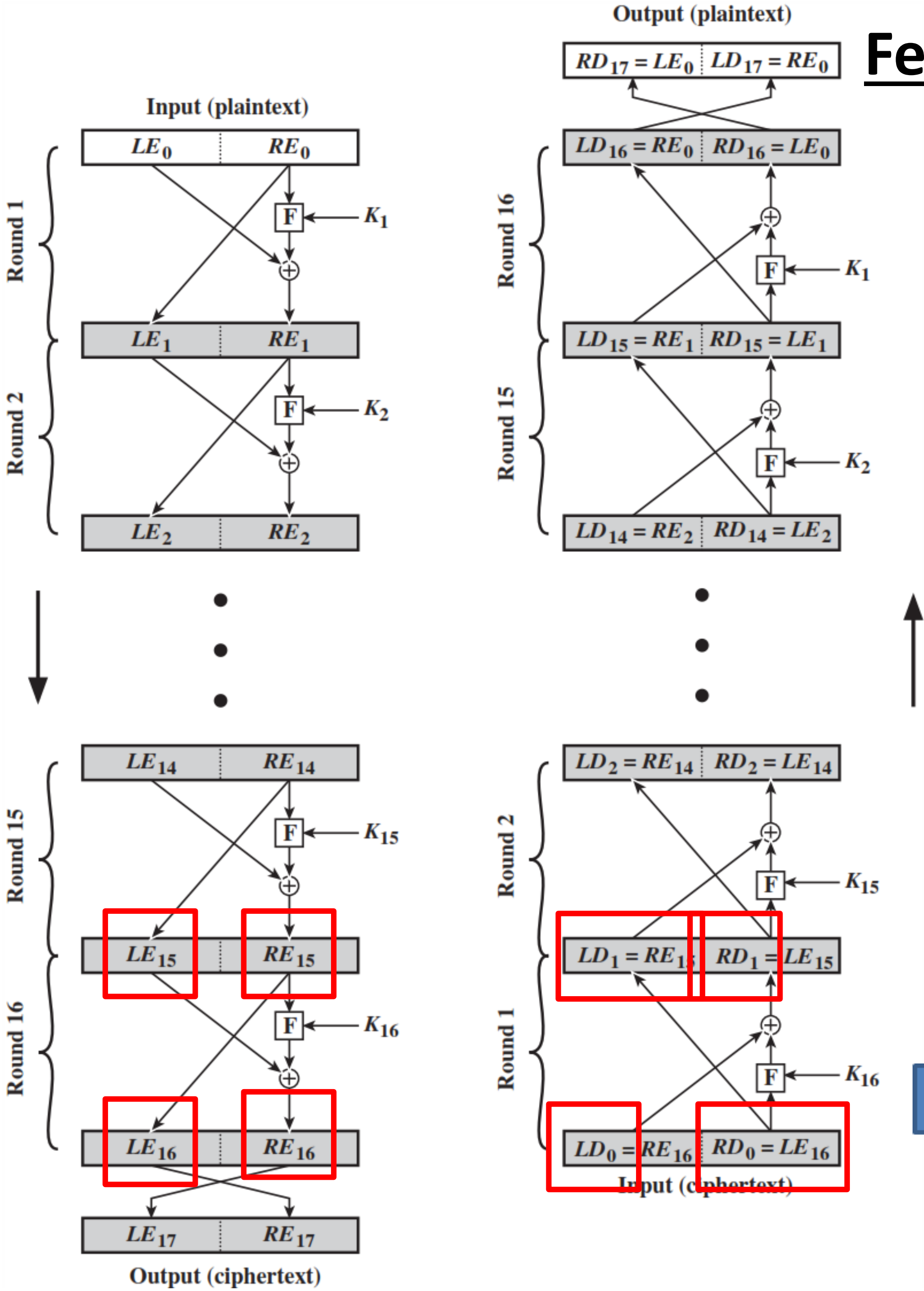
Classical Feistel Network

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 16^{th} 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:

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

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

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

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

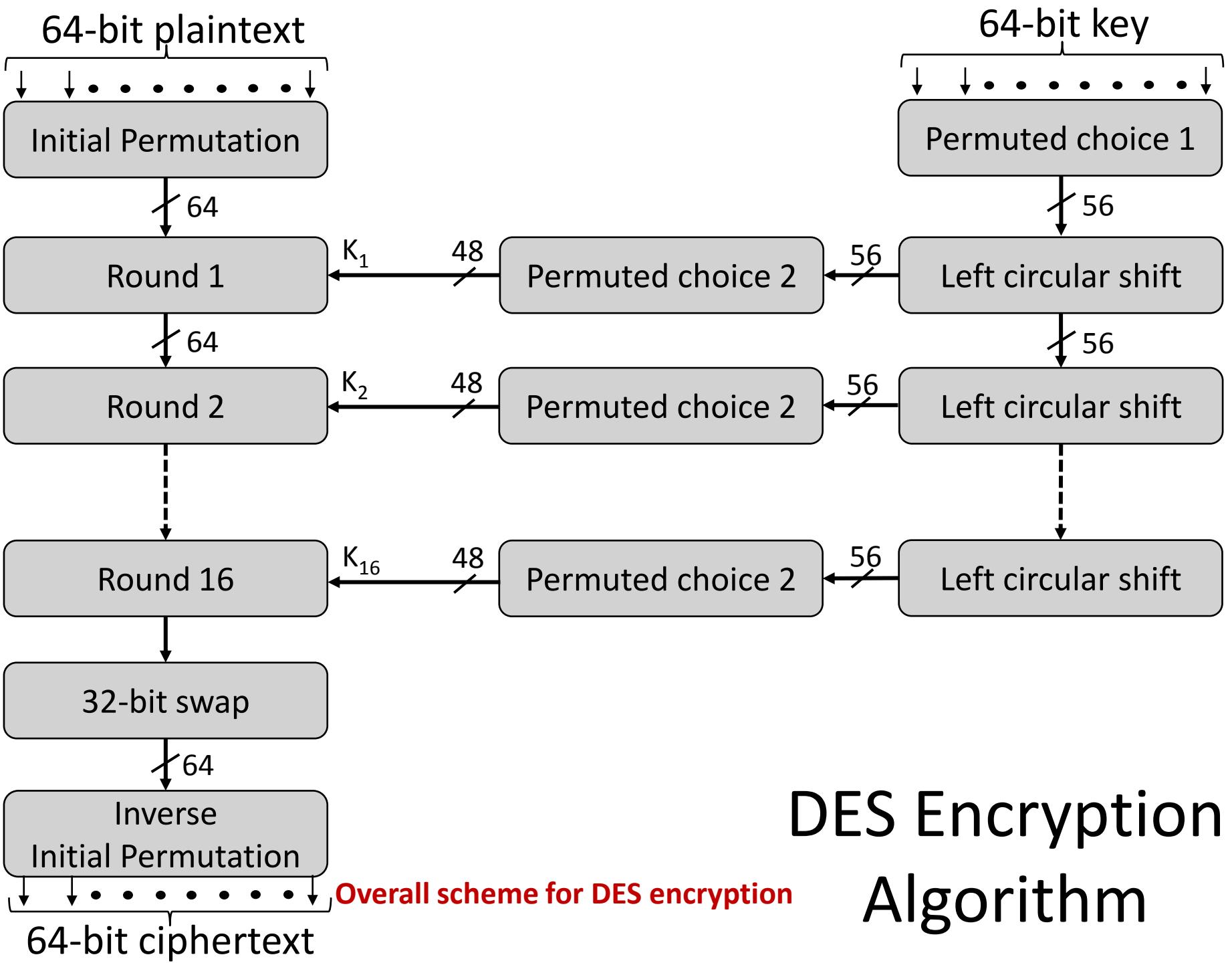
$\oplus\ C]$

$$\textit{Thus},$$

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

Data Encryption Standard (DES)

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



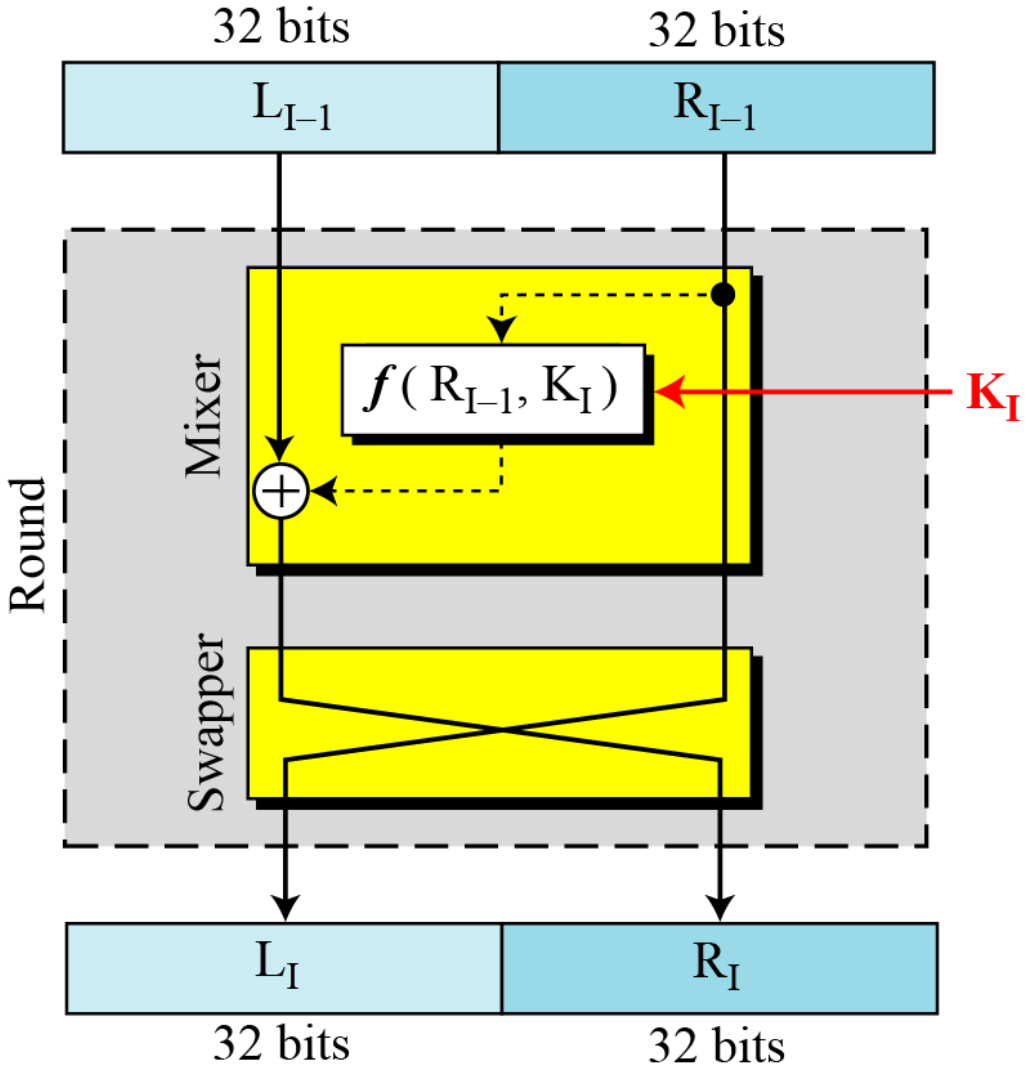
Overall scheme for DES encryption

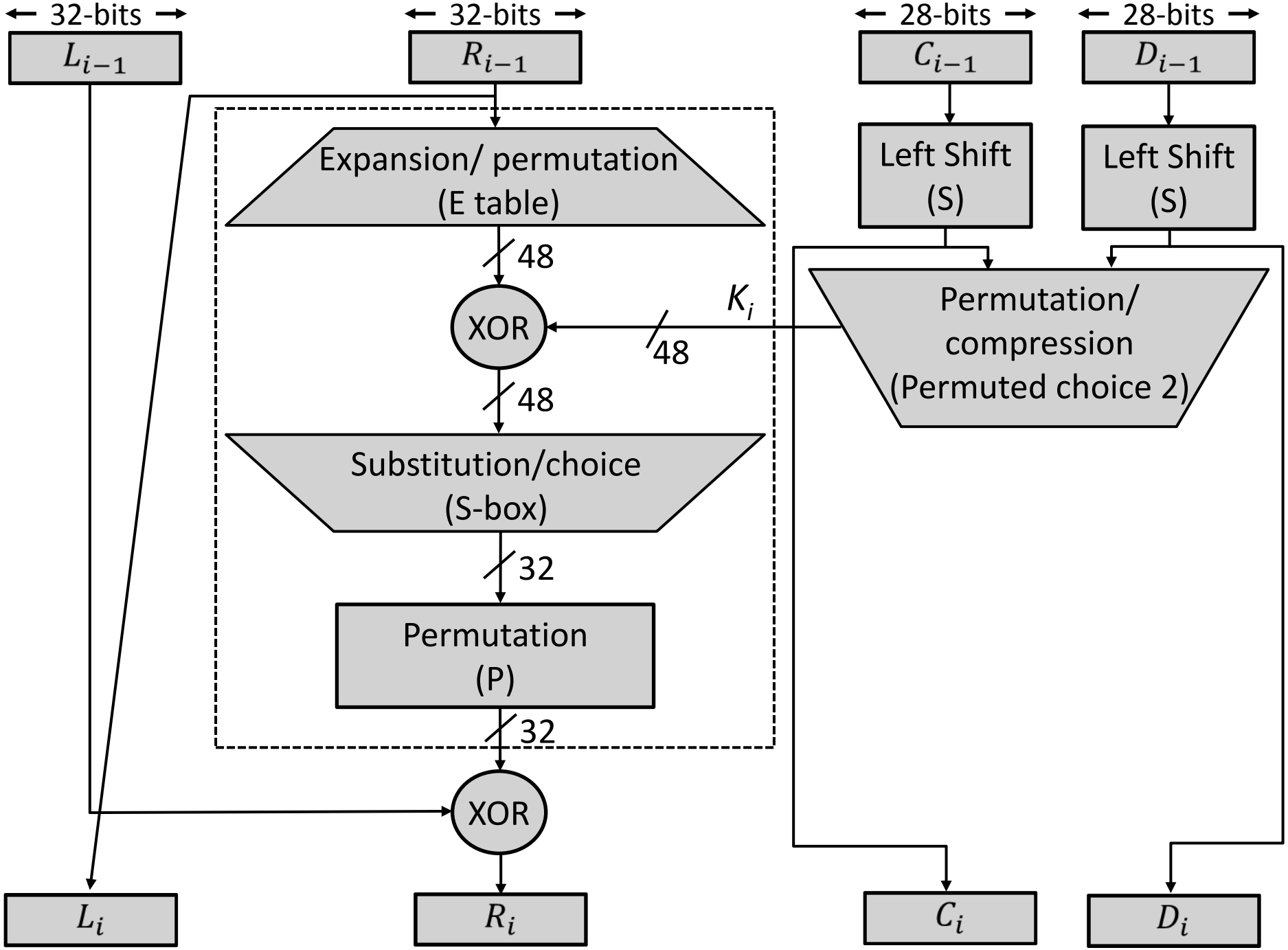
DES Encryption Algorithm

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



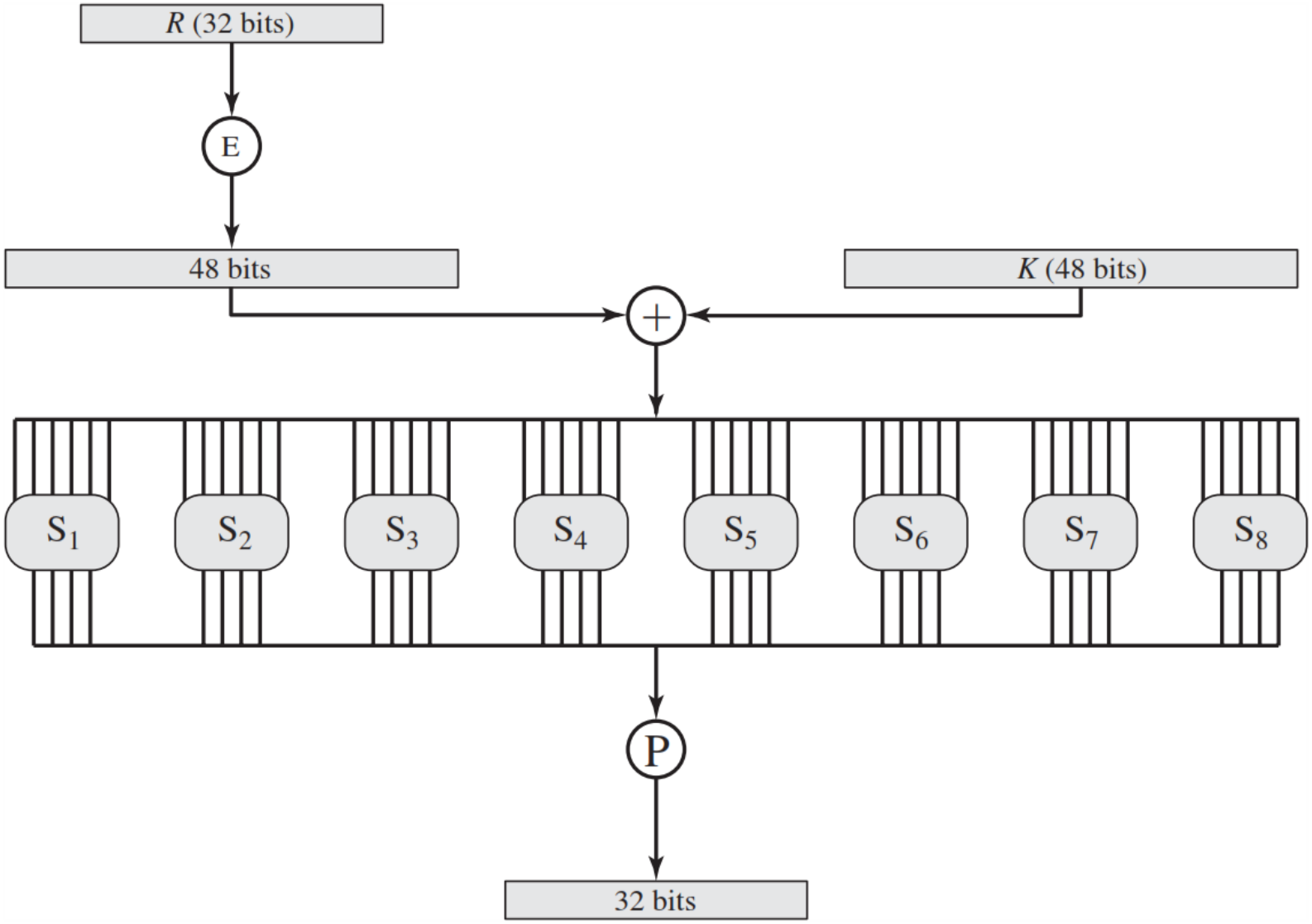


Single Round of DES in Detail

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



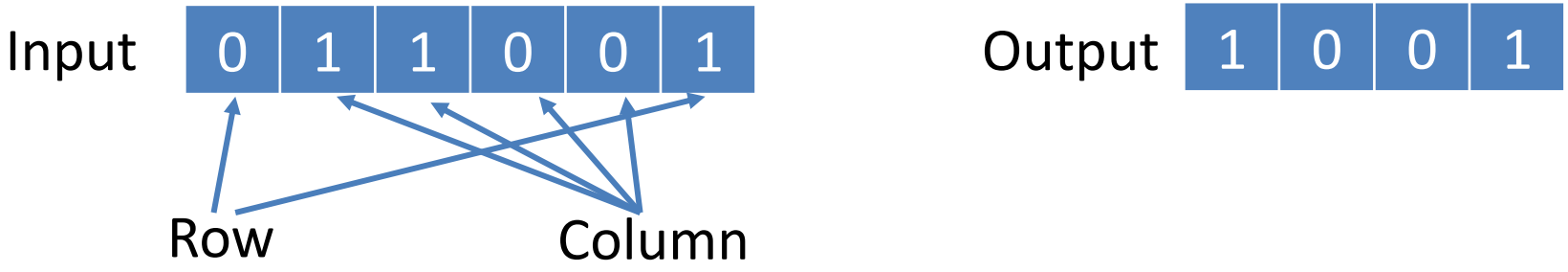
Role of S-box (Cont…)

- 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:



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: 0000000000000001	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.

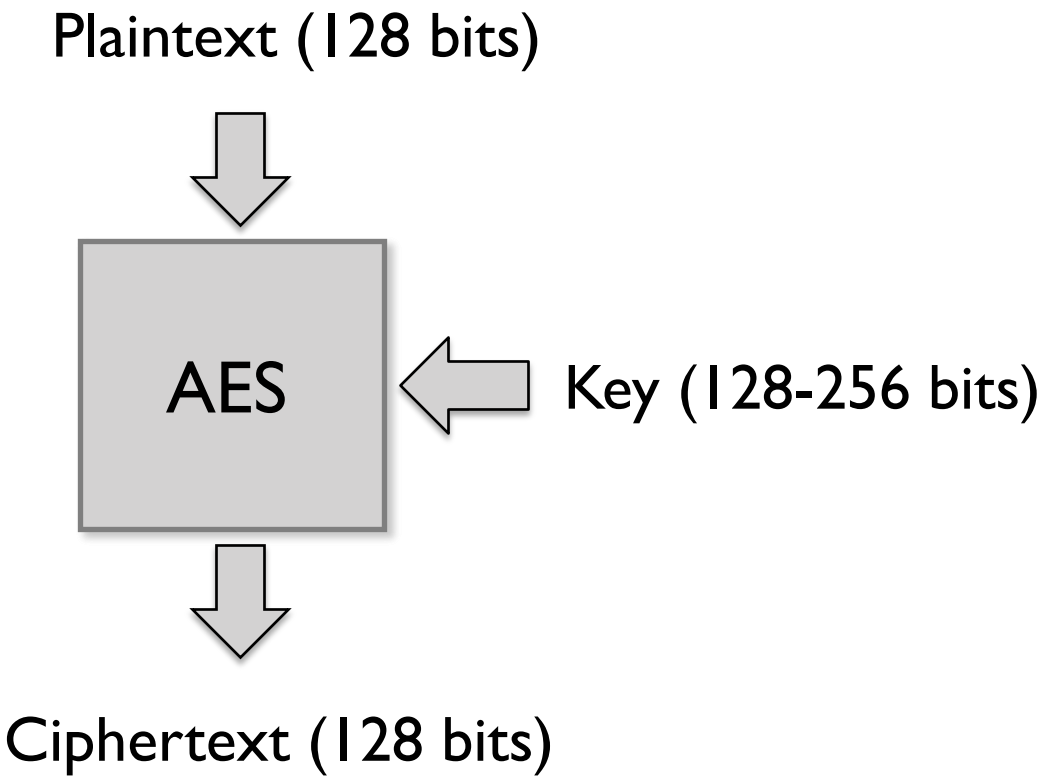
AES (Advanced Encryption Standard)

- 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

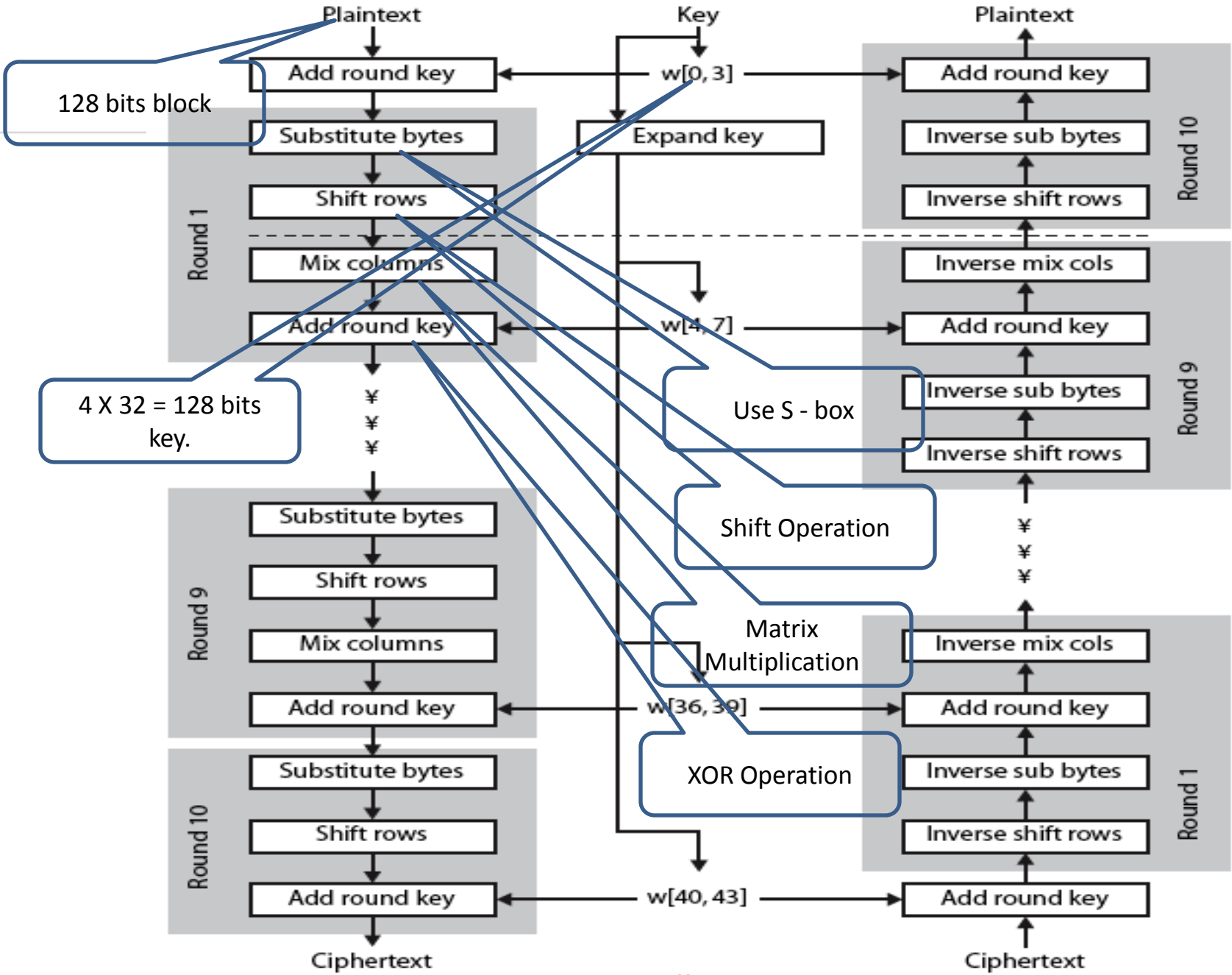
Initialization

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

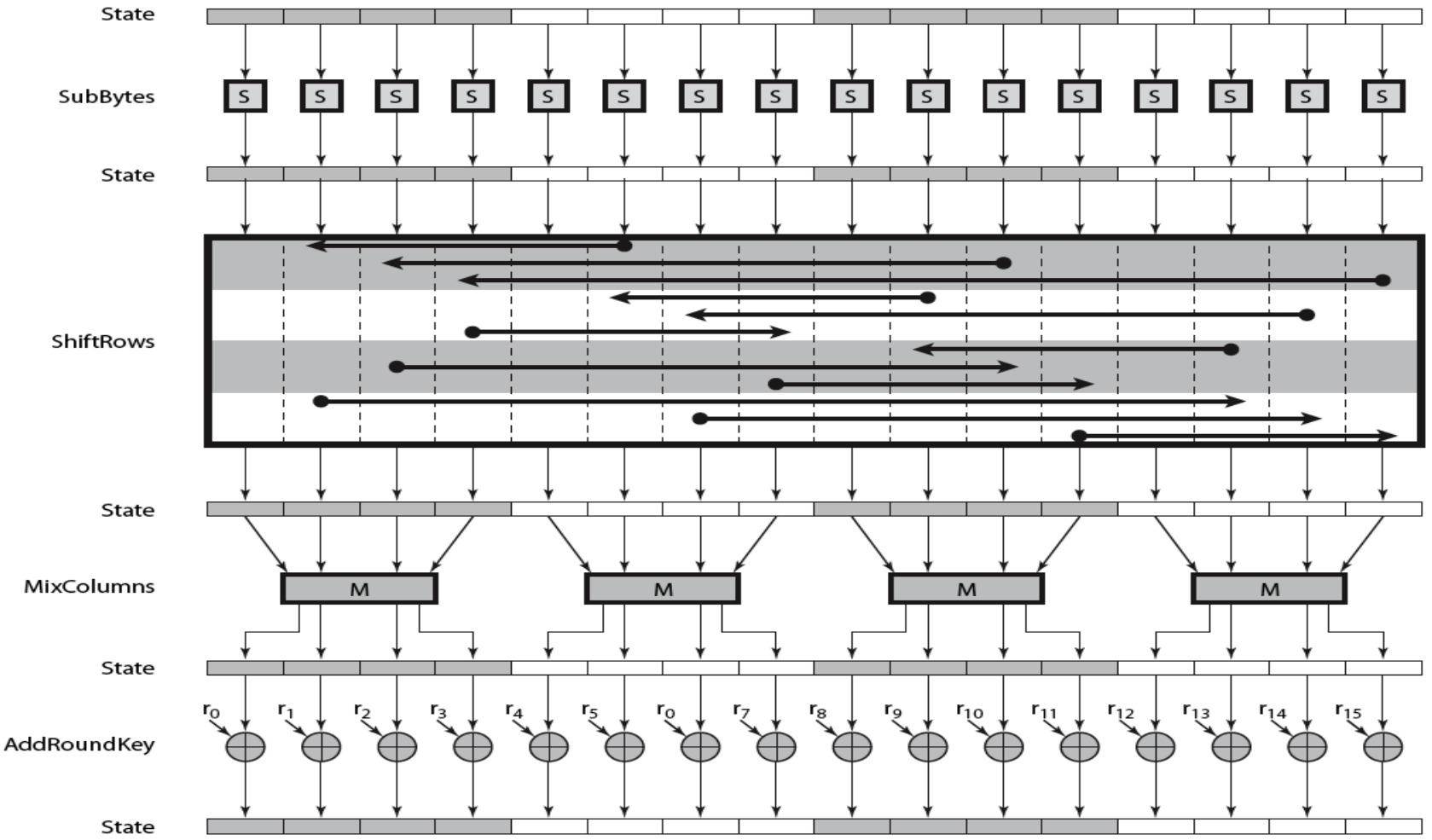
- The first n-1 rounds consist of four distinct transformation functions.

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.



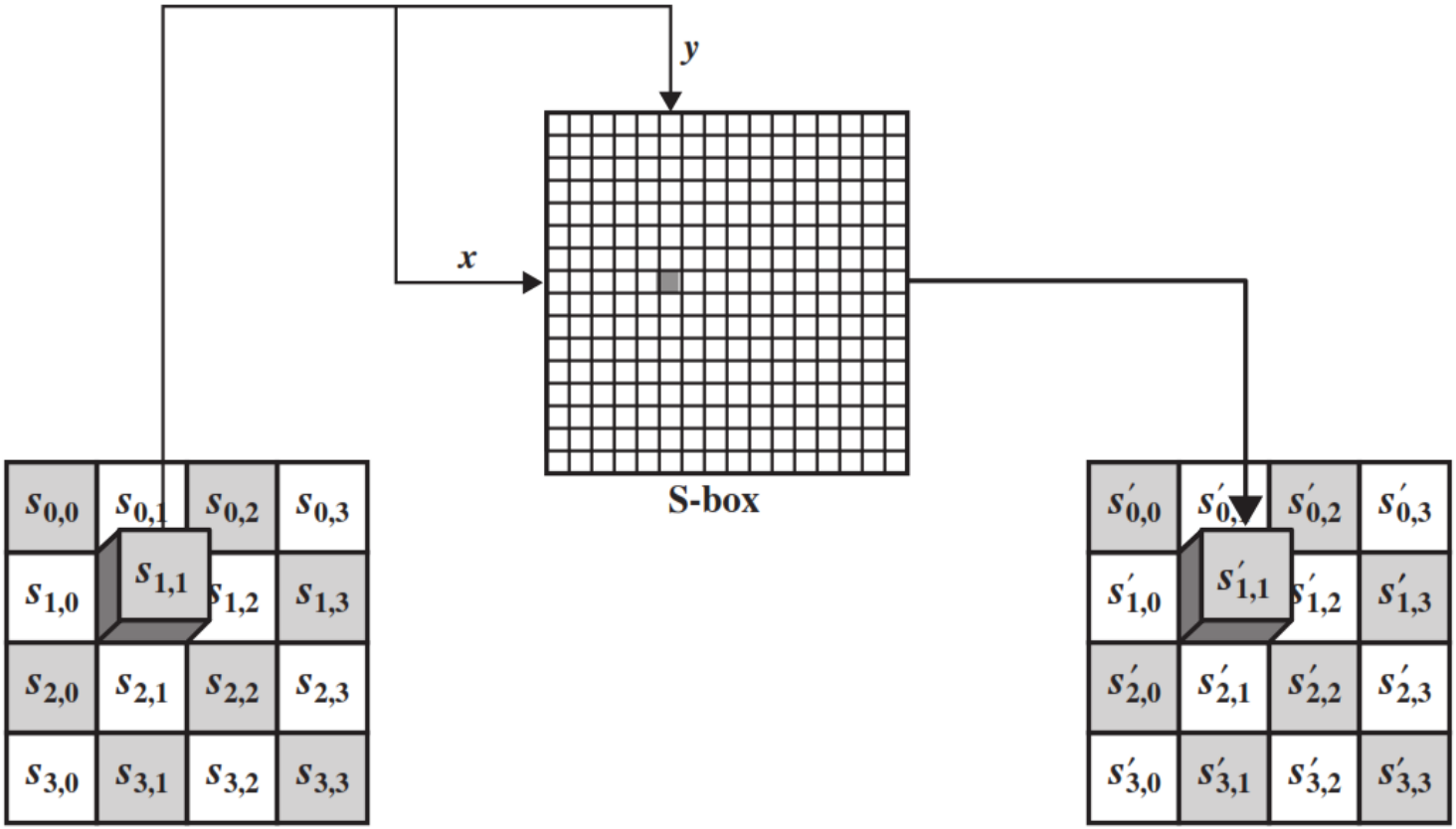
AES Overall Structure

AES Round


State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
r0
r1
r2
r3
r4
r5
r0
r7
r8
r9
r10
r11
r12
r13
r14
r15
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.

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

State

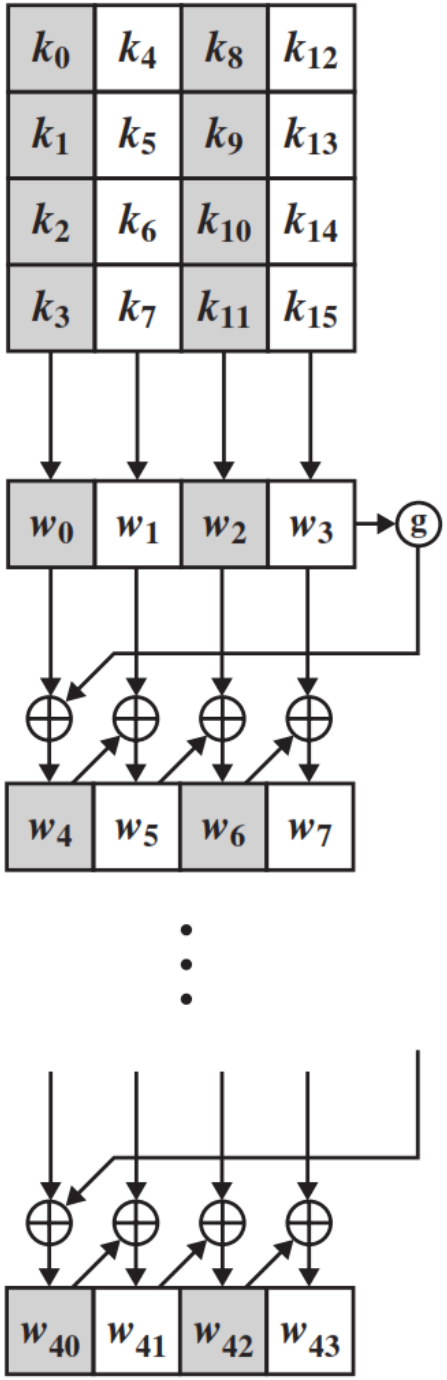
$\oplus$

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

Round Key

$=$

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



AES Key Expansion

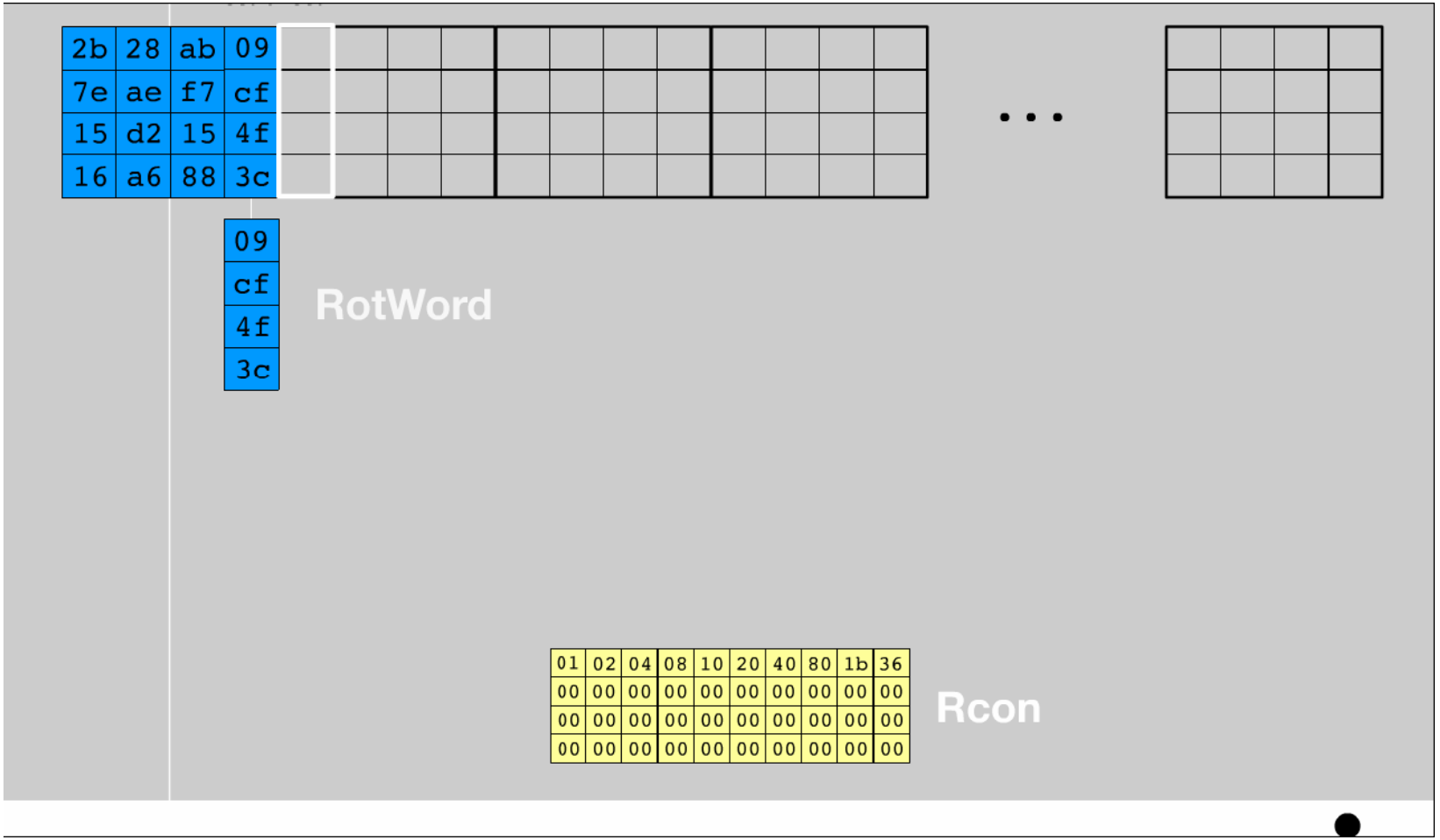
- The AES key expansion algorithm takes as input a four-word (16-byte) key and produces a linear array of **44 words** (176 bytes).
- Each added word **w[i]** depends on the immediately preceding word, w[i - 1].
- In three out of four cases, a simple XOR is used.

Key Expansion Example

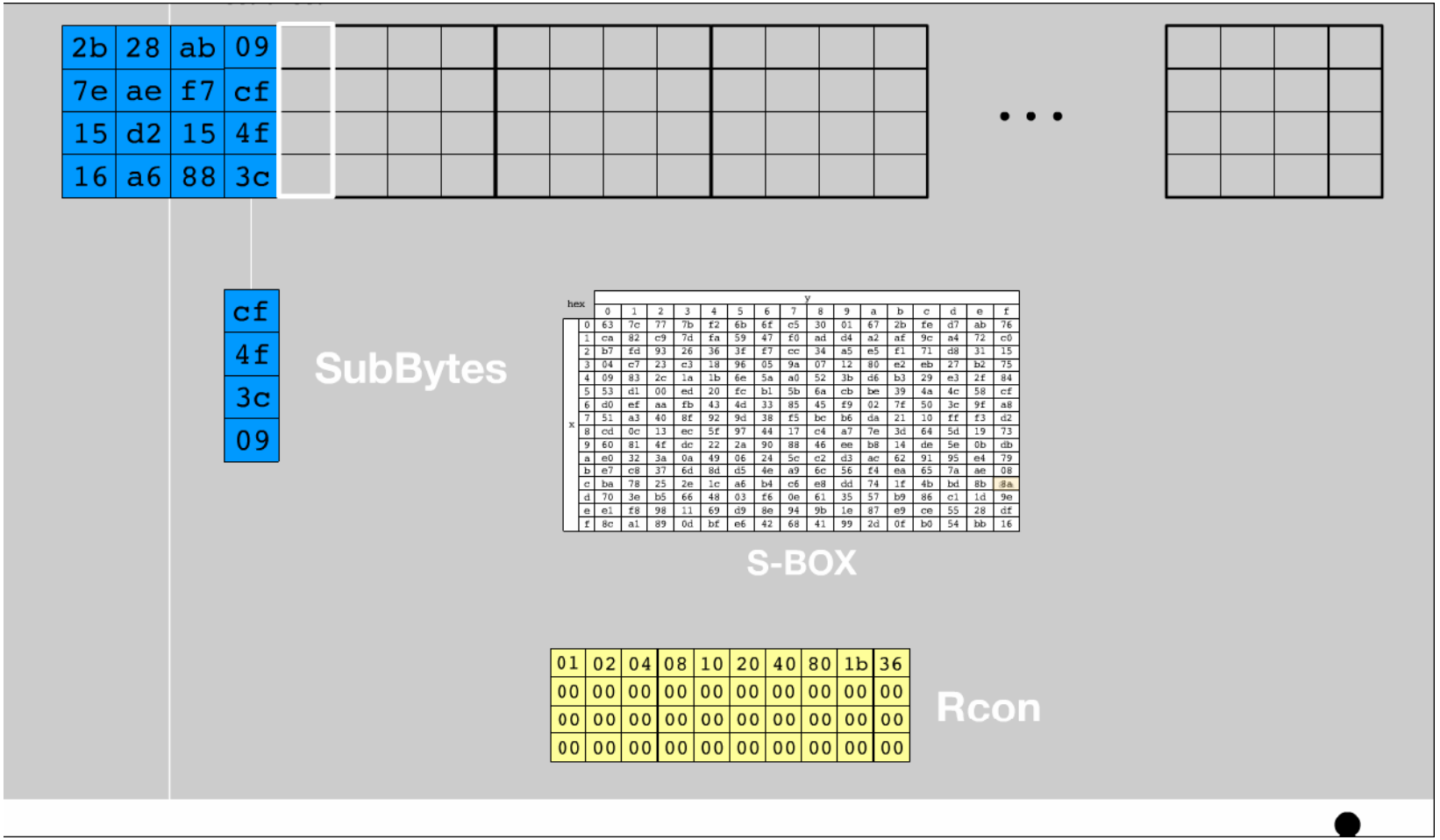
Plaintext:	0123456789abcdeffedcba9876543210
Key:	0f1571c947d9e8590cb7add6af7f6798
Ciphertext:	ff0b844a0853bf7c6934ab4364148fb9

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

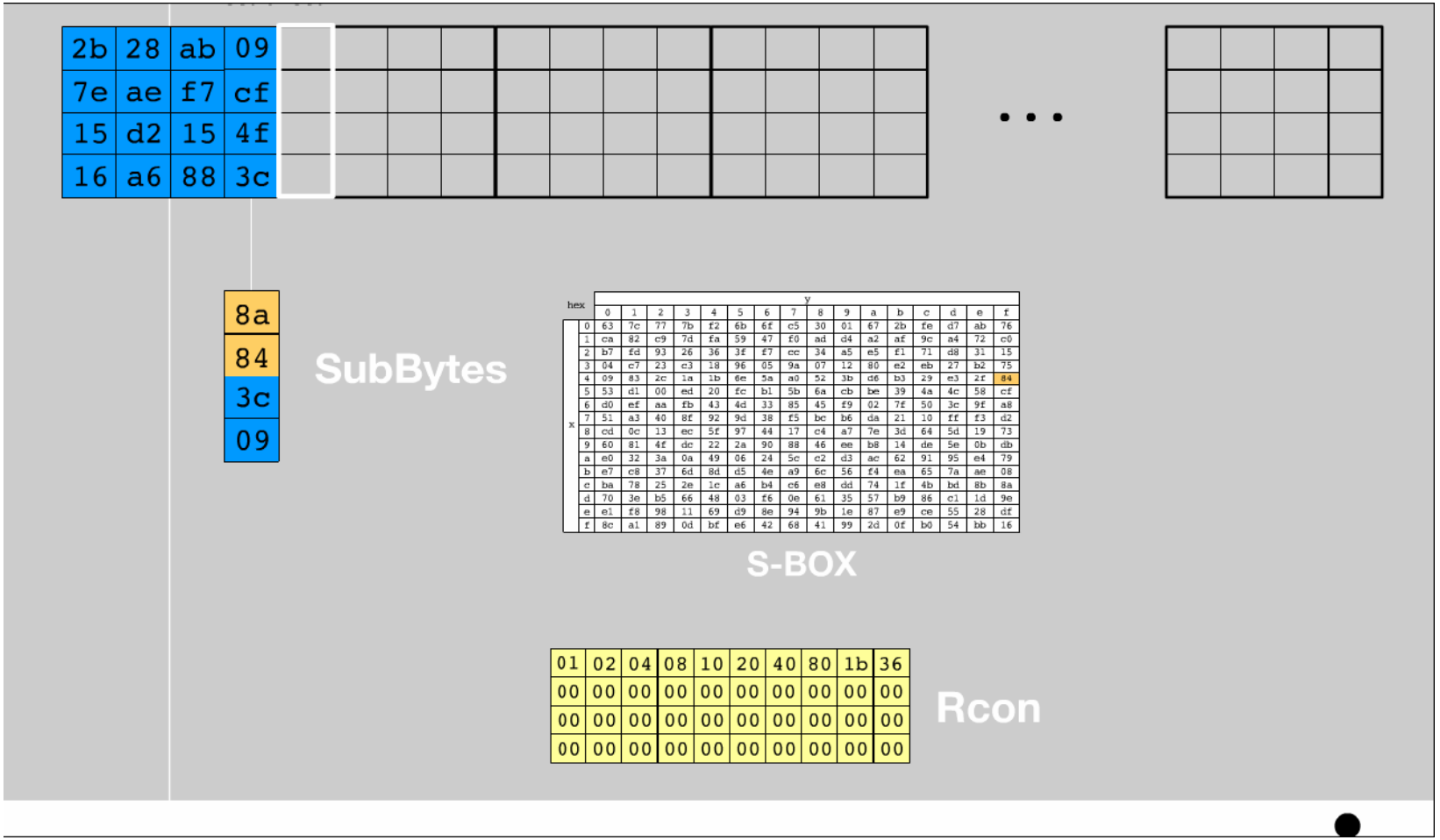
Key Schedule Generation (For reference)



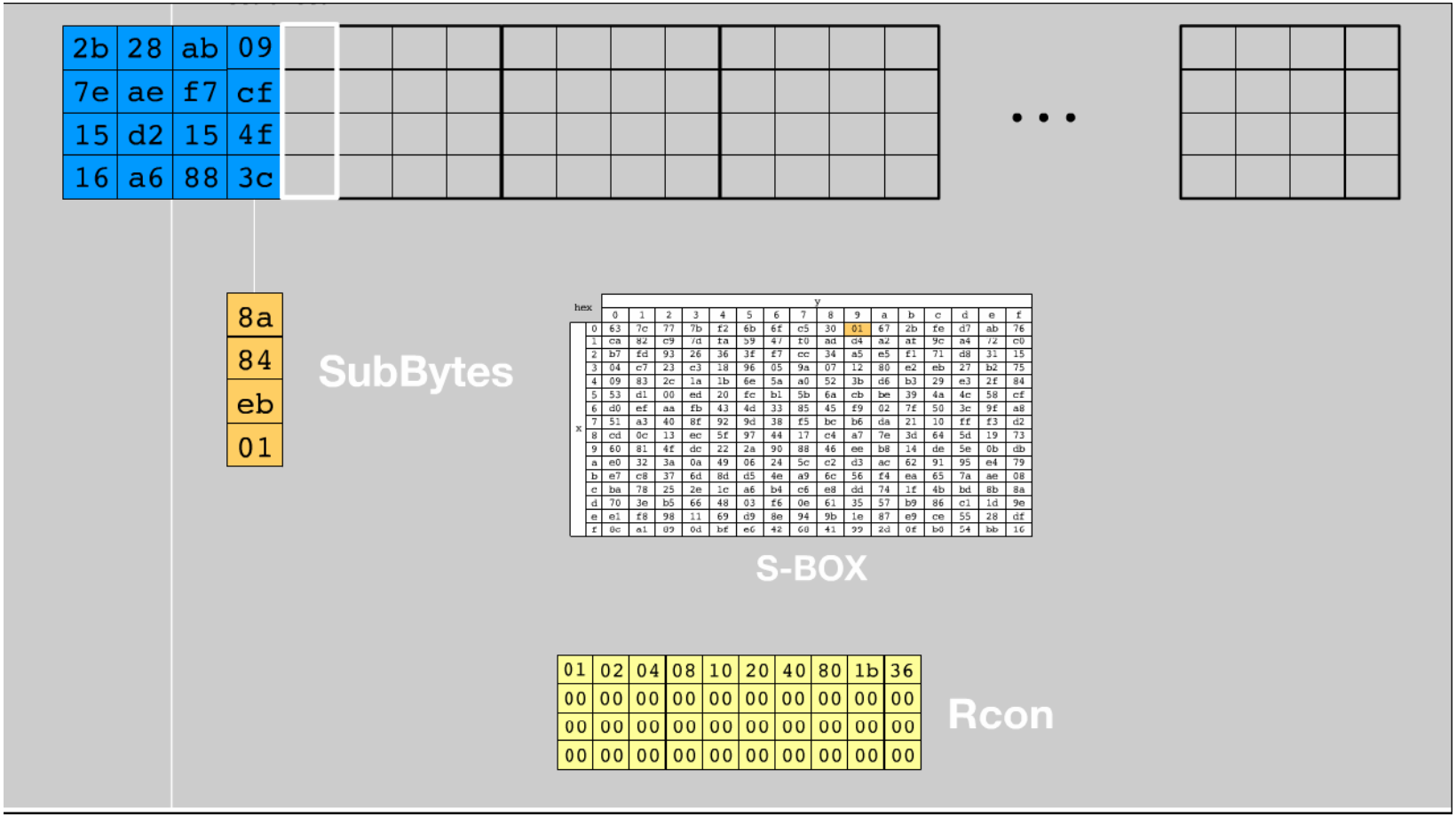
Key Schedule Generation (Cont.)



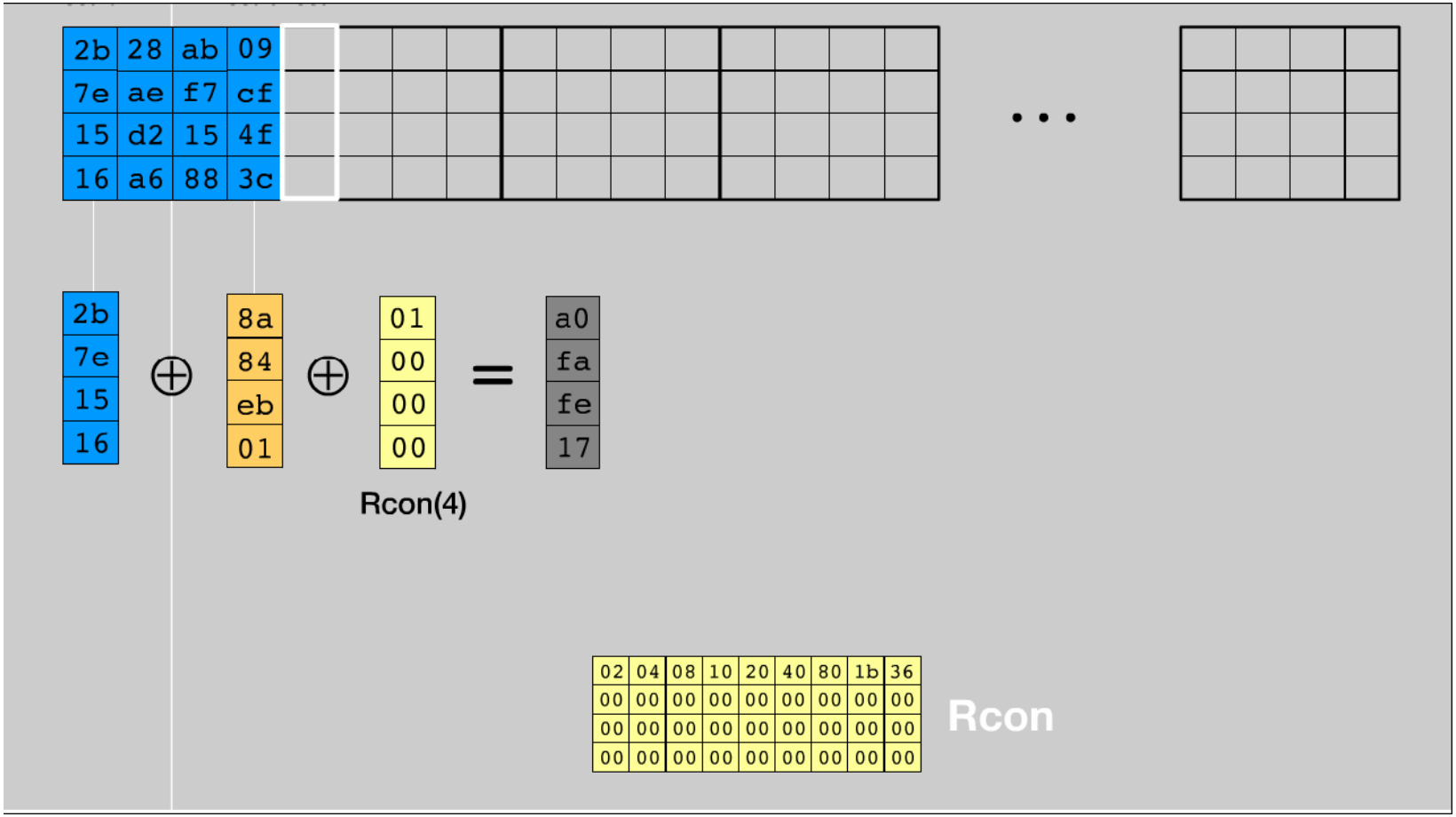
Key Schedule Generation (Cont.)



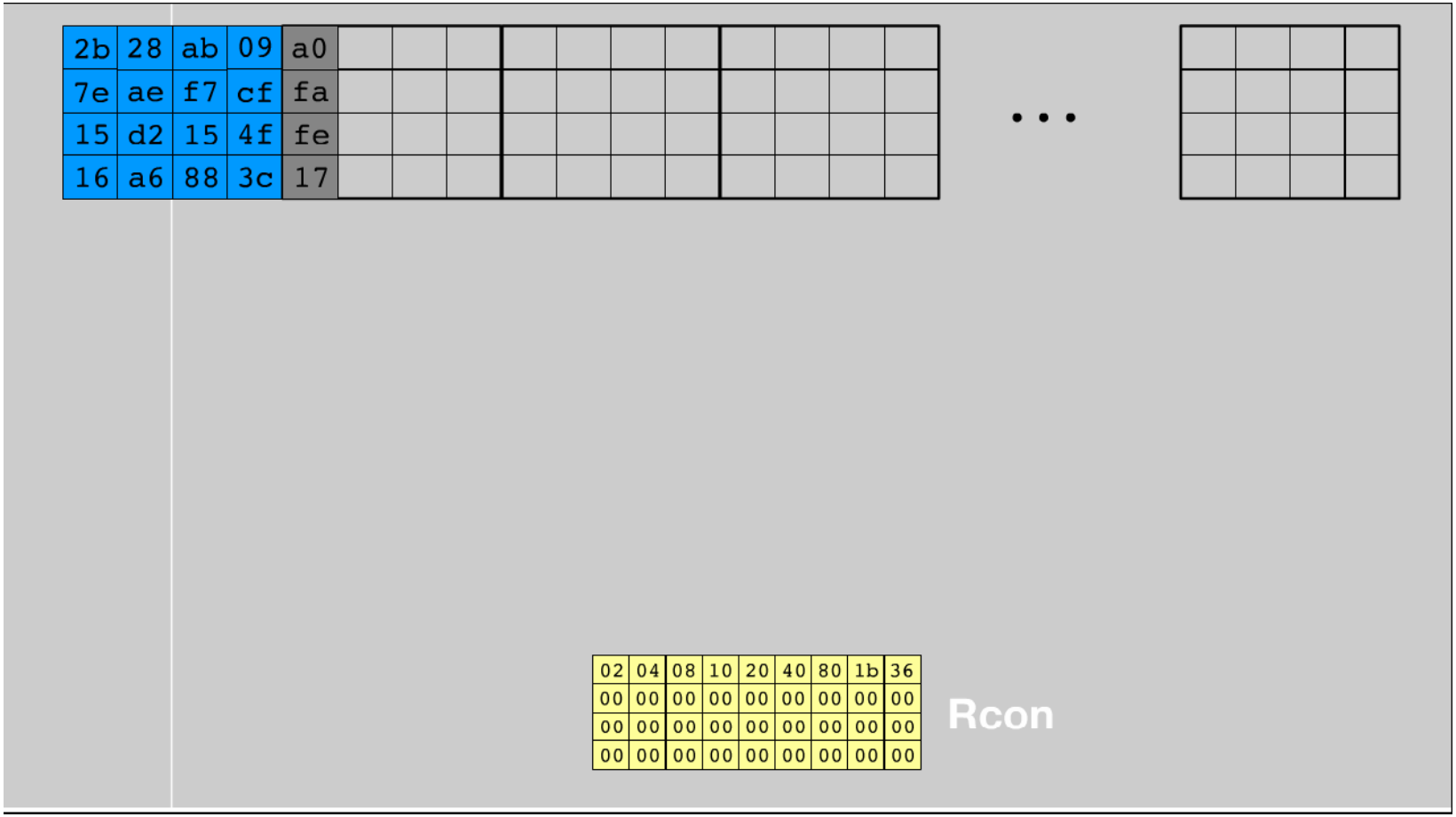
Key Schedule Generation (Cont.)



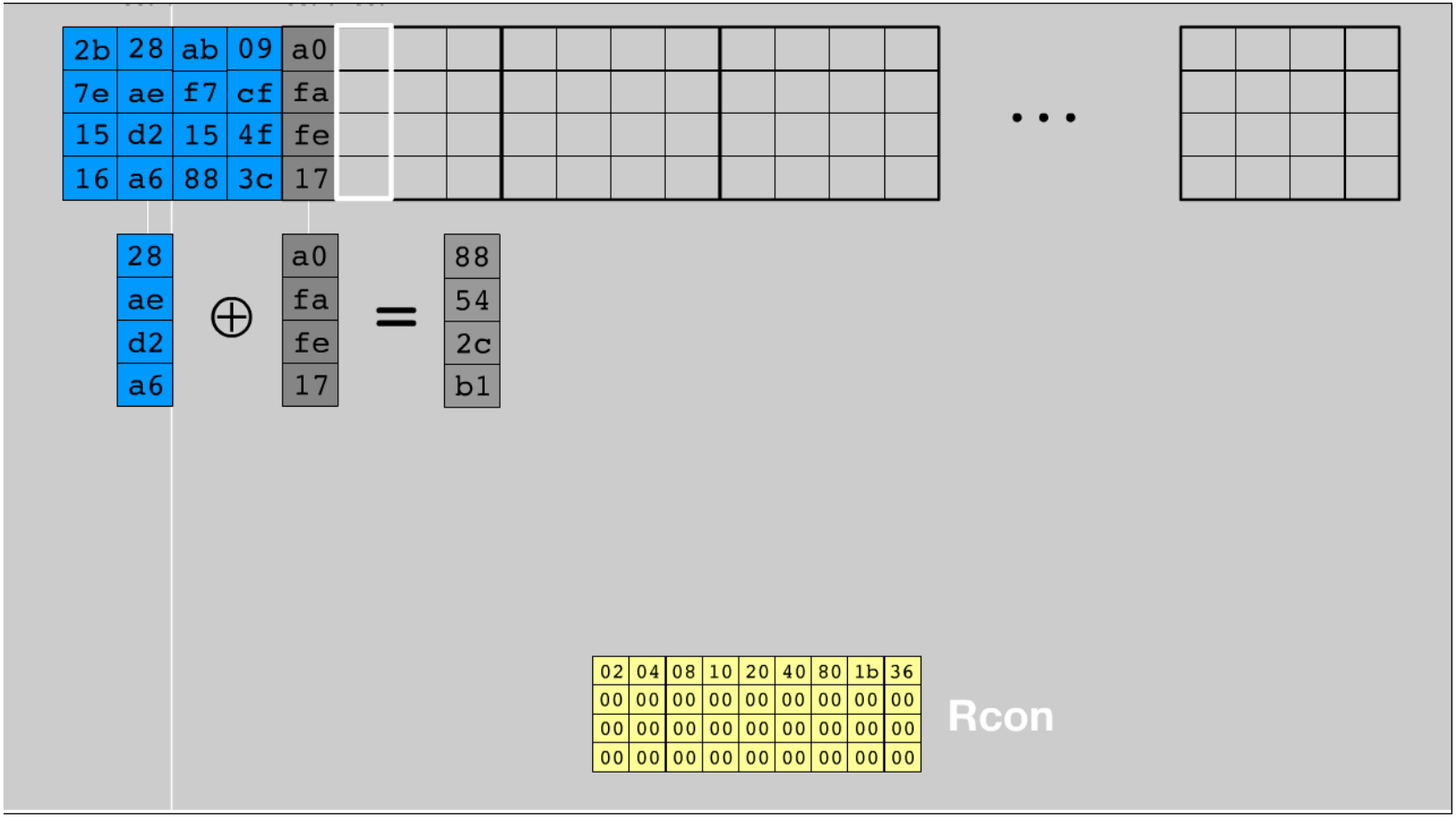
Key Schedule Generation (Cont.)



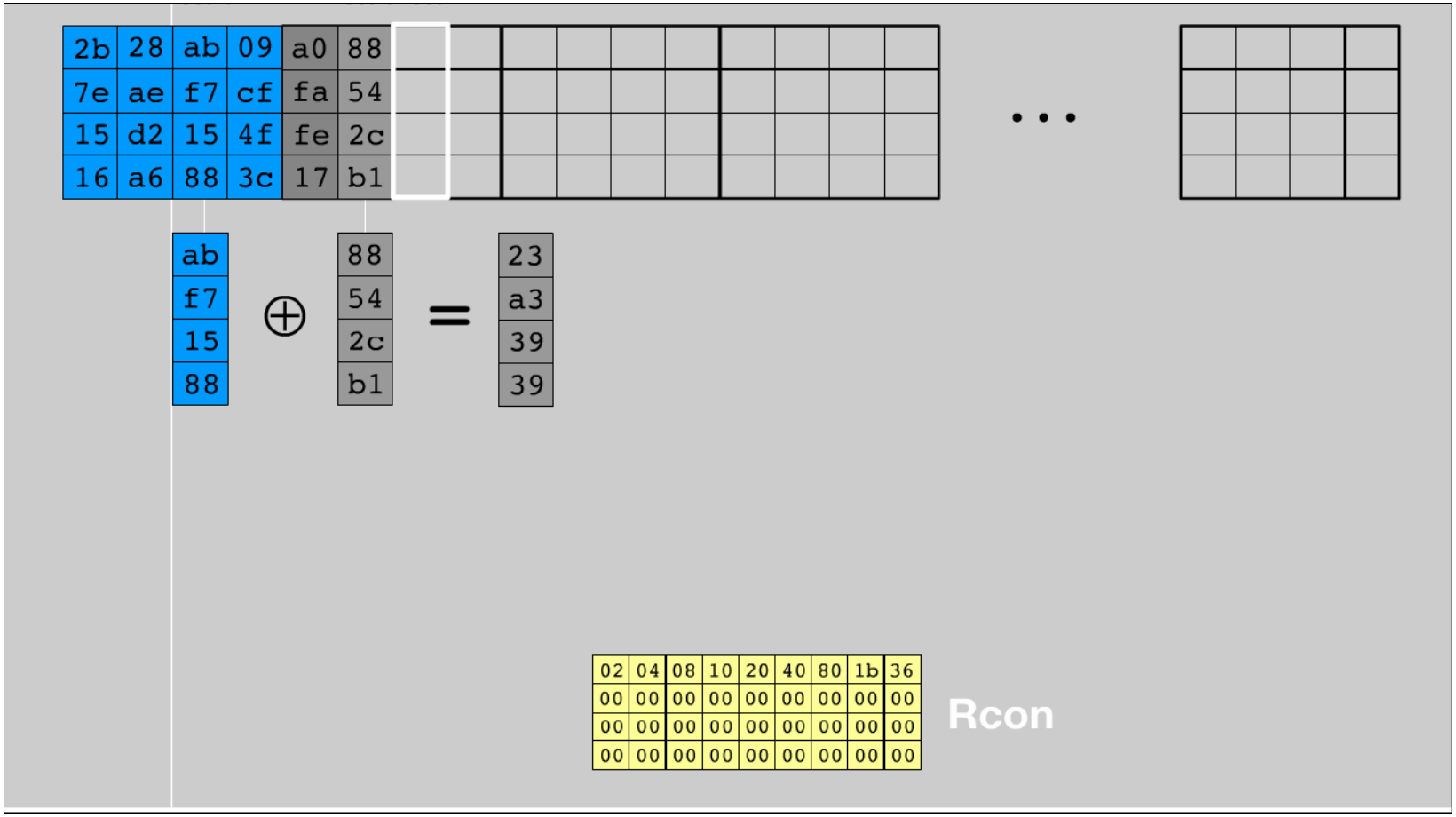
Key Schedule Generation (Cont.)



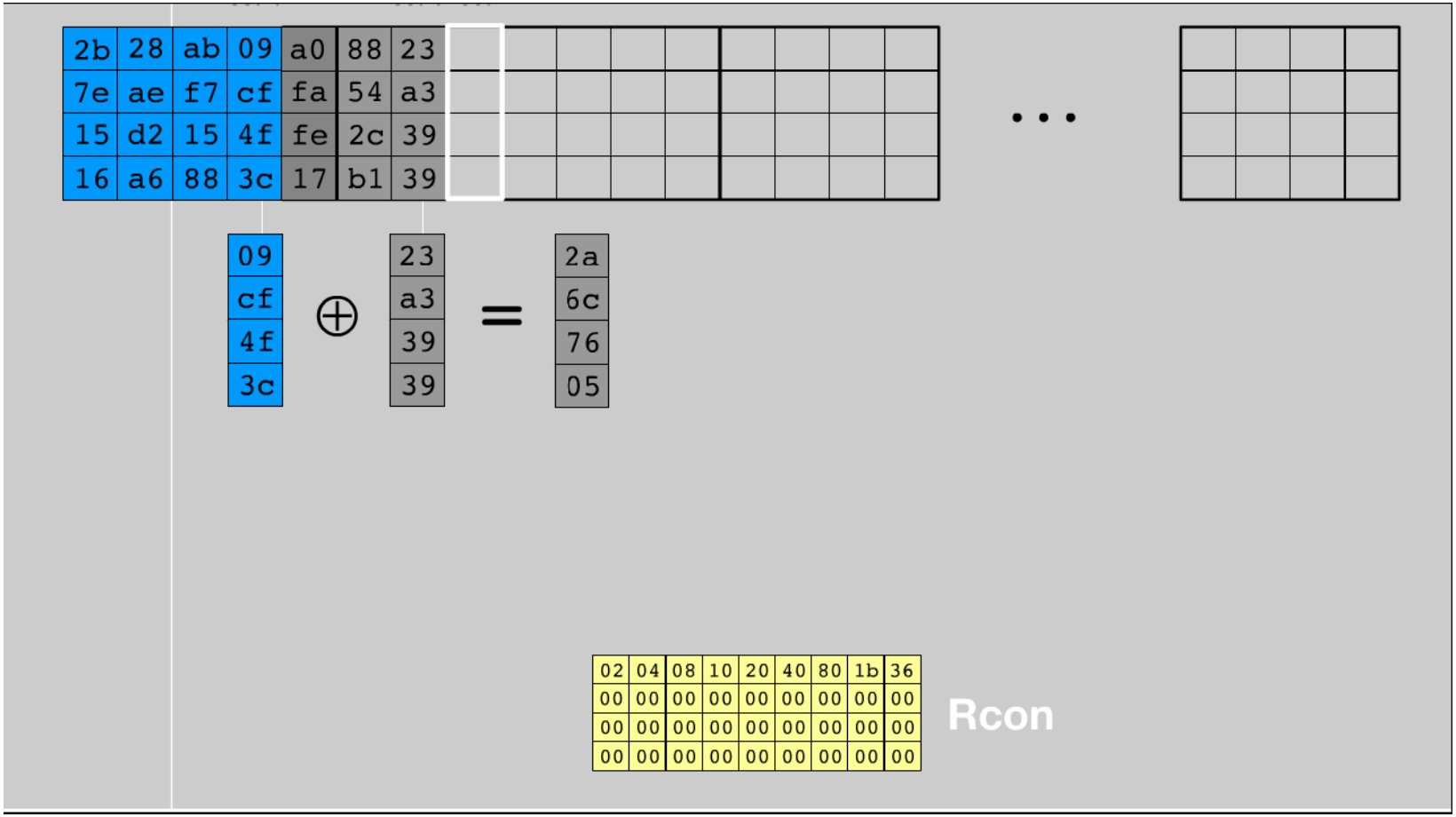
Key Schedule Generation (Cont.)



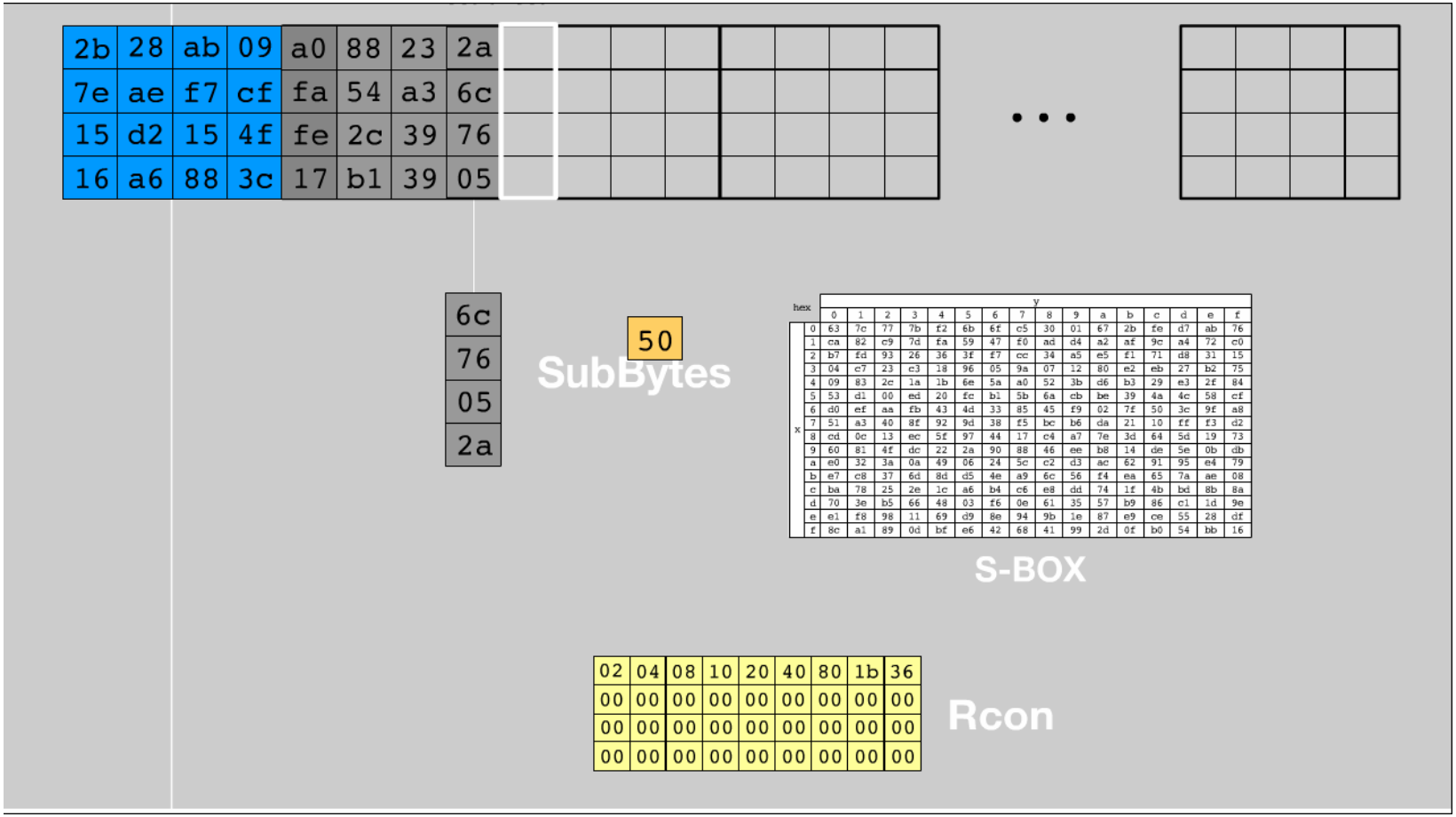
Key Schedule Generation (Cont.)



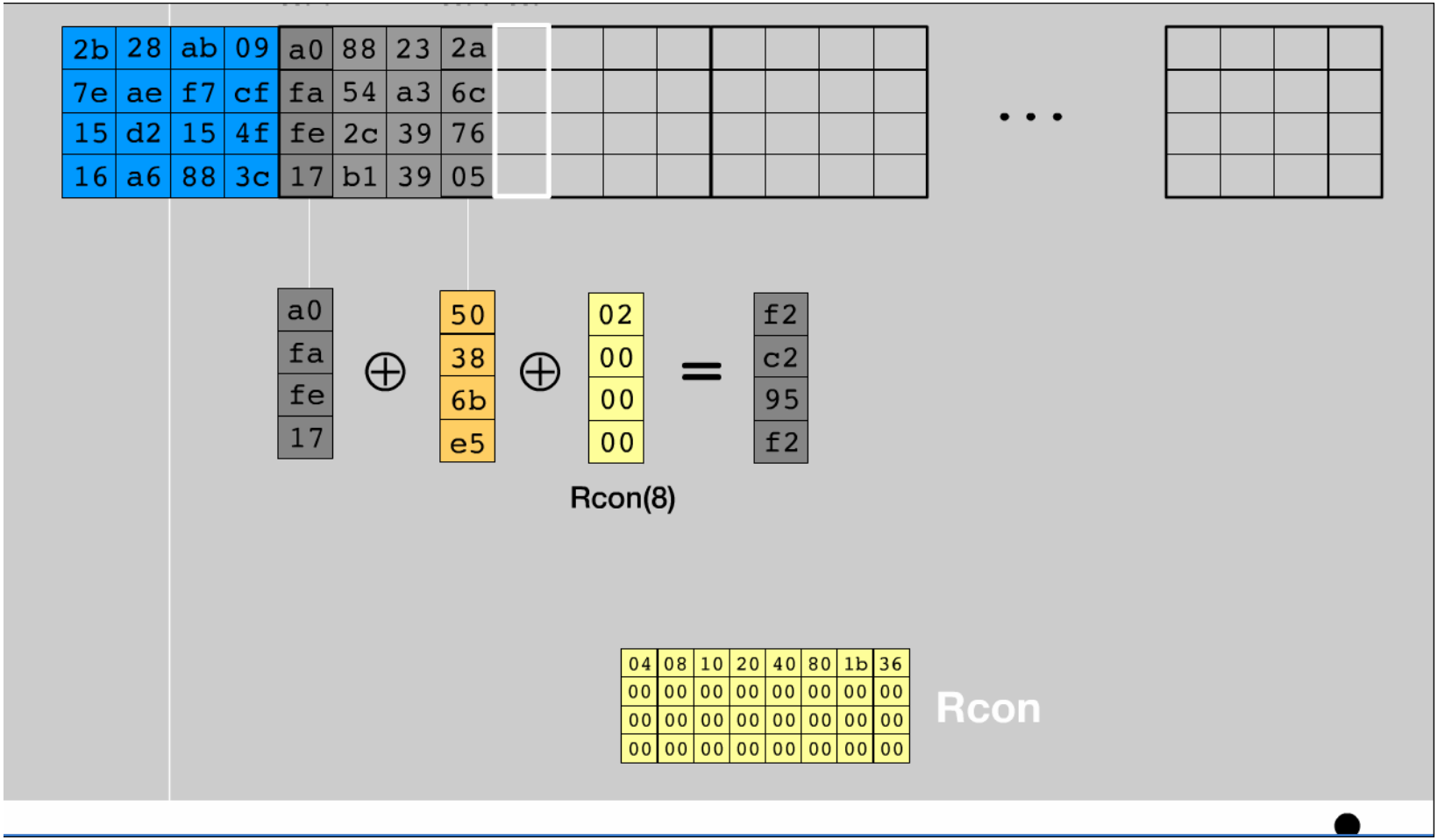
Key Schedule Generation (Cont.)



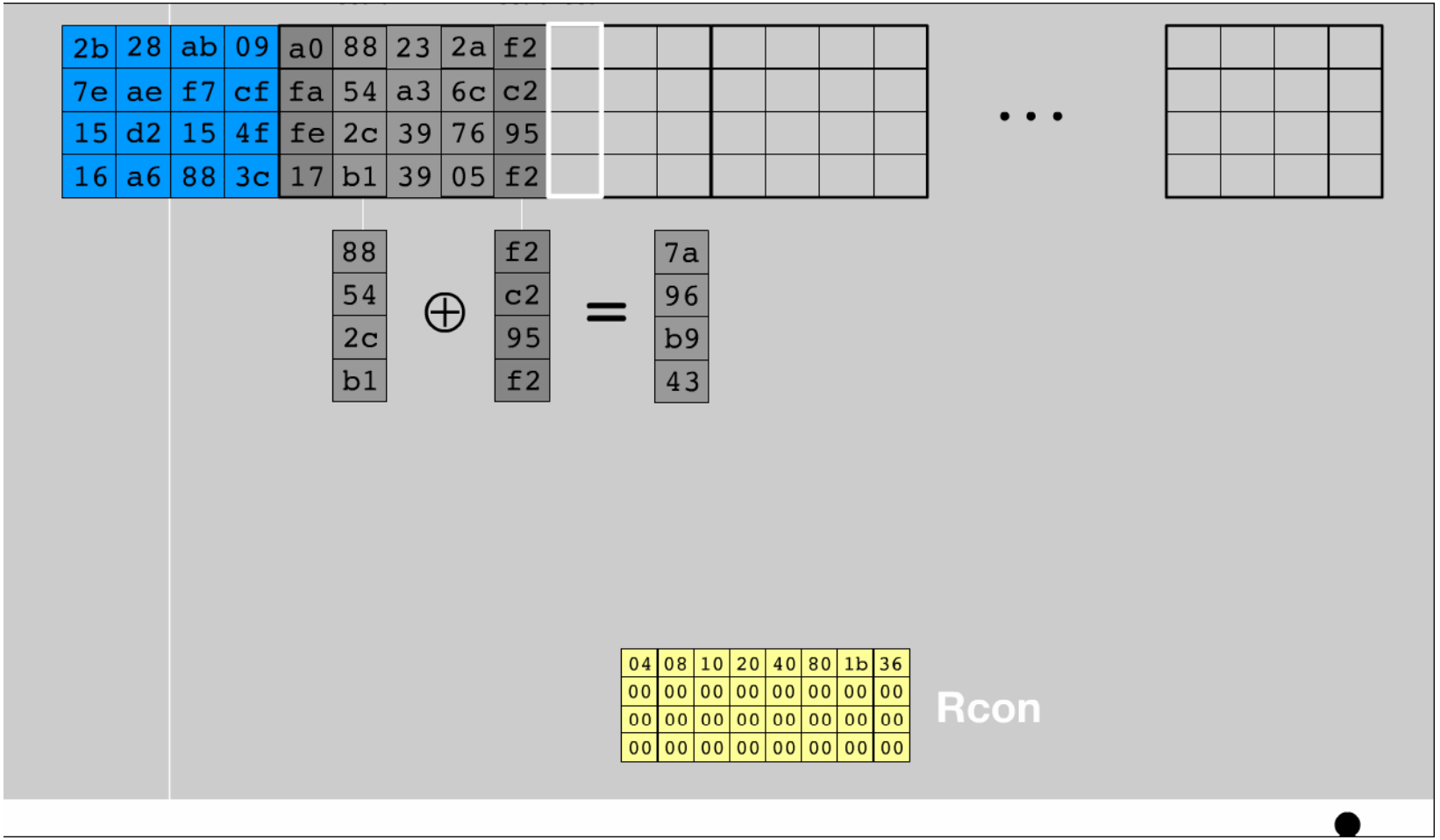
Key Schedule Generation (Cont.)



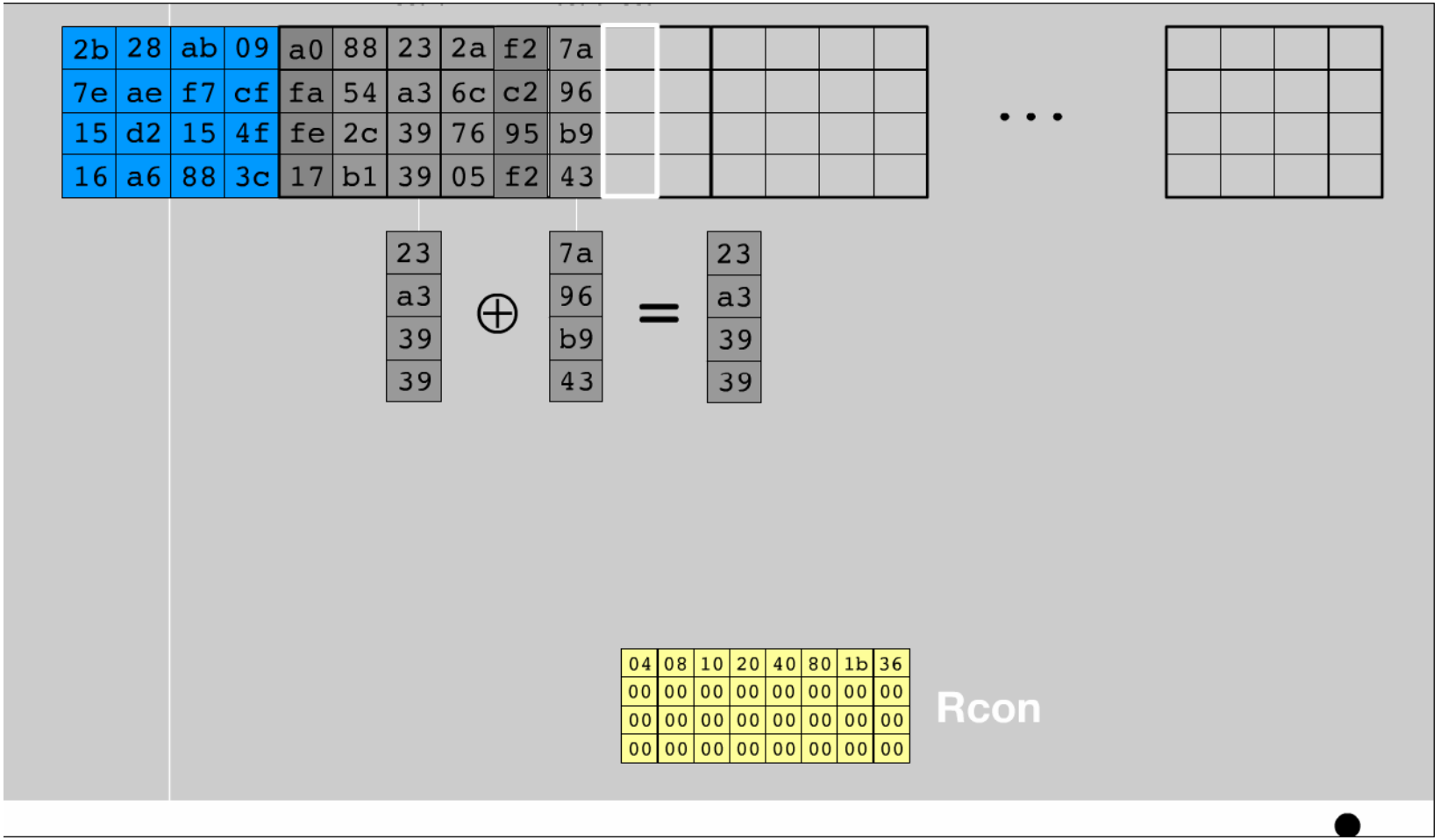
Key Schedule Generation (Cont.)



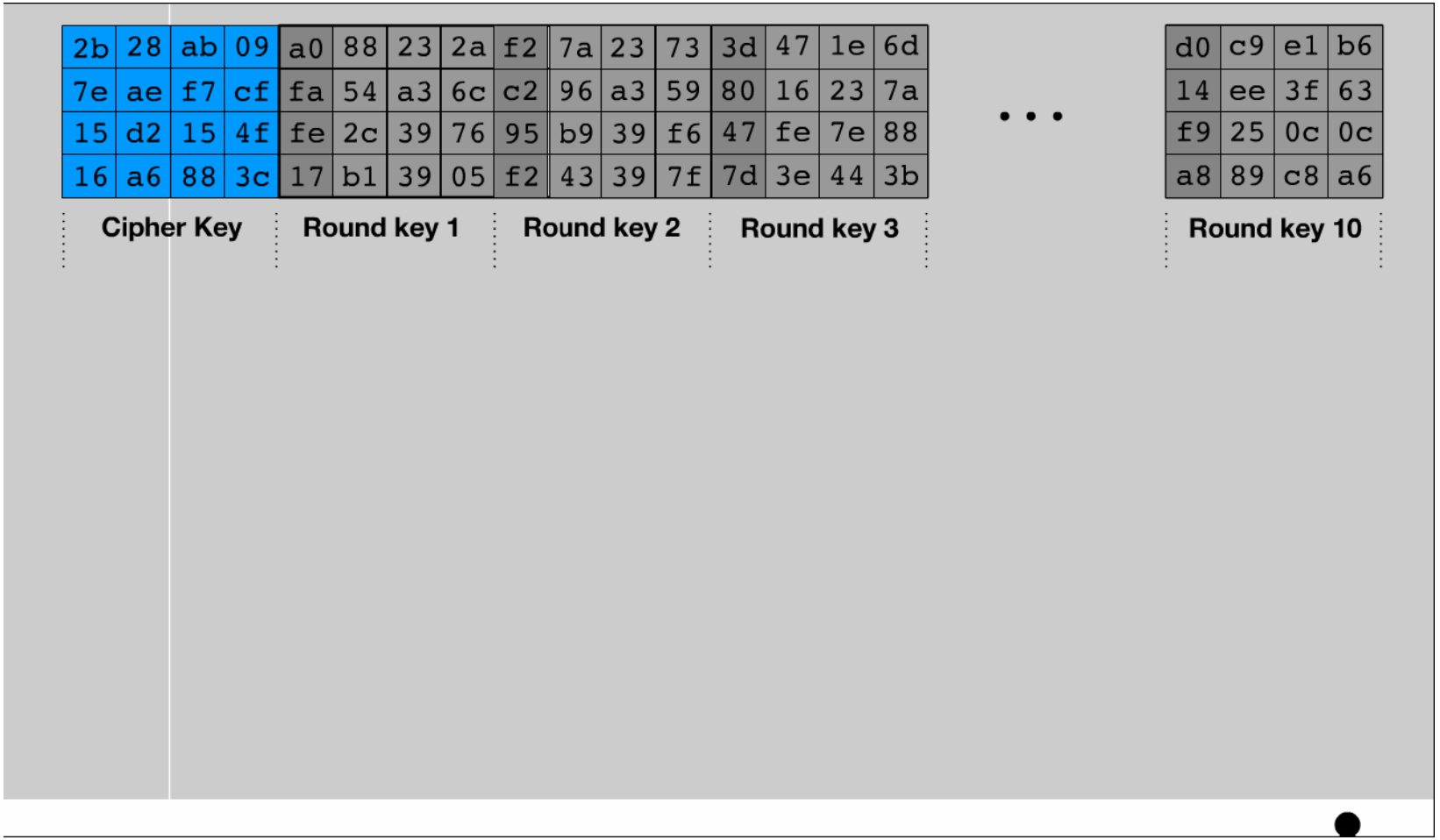
Key Schedule Generation (Cont.)



Key Schedule Generation (Cont.)



Key Schedule Generation (Cont.)



Unit 2

Modes of Operations, Multiple encryptions and triple DES

Outline

- Block cipher modes of operations
  - Electronic Code Book Mode
  - Cipher Block Chaining Mode
  - Cipher Feedback Mode
  - Output Feedback Mode
  - Counter Mode
- Multiple encryptions
- Triple DES

Block Cipher Modes of Operations

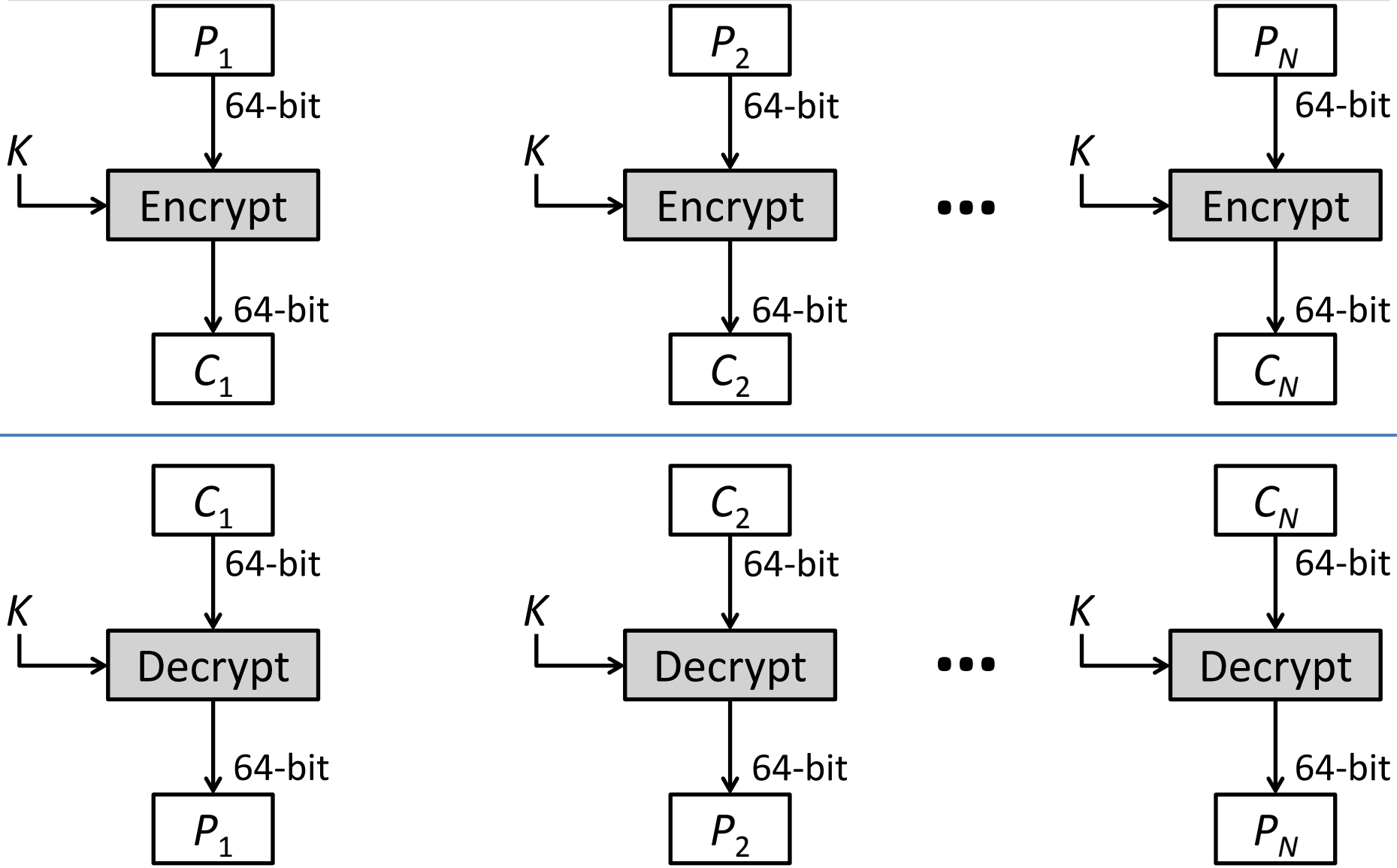
- To apply a block cipher in a **variety of applications**, five "modes of operation" have been defined.
- The **five modes** are intended to cover a wide variety of applications of encryption for which a block cipher could be used.
- These modes are intended for use with any symmetric block cipher, including triple DES and AES.

1. Electronic Code Book (ECB)
2. Cipher Block Chaining (CBC)
3. Cipher Feedback (CFB)
4. Output Feedback (OFB)
5. Counter (CTR)

1. Electronic Code Book (ECB)

- In **ECB** Mode Plaintext is handled one block at a time and each block of plaintext is encrypted using the same key.
- The term **codebook** is used because, for a given key, there is a unique ciphertext for every block of plaintext.

1. ECB Encryption & Decryption



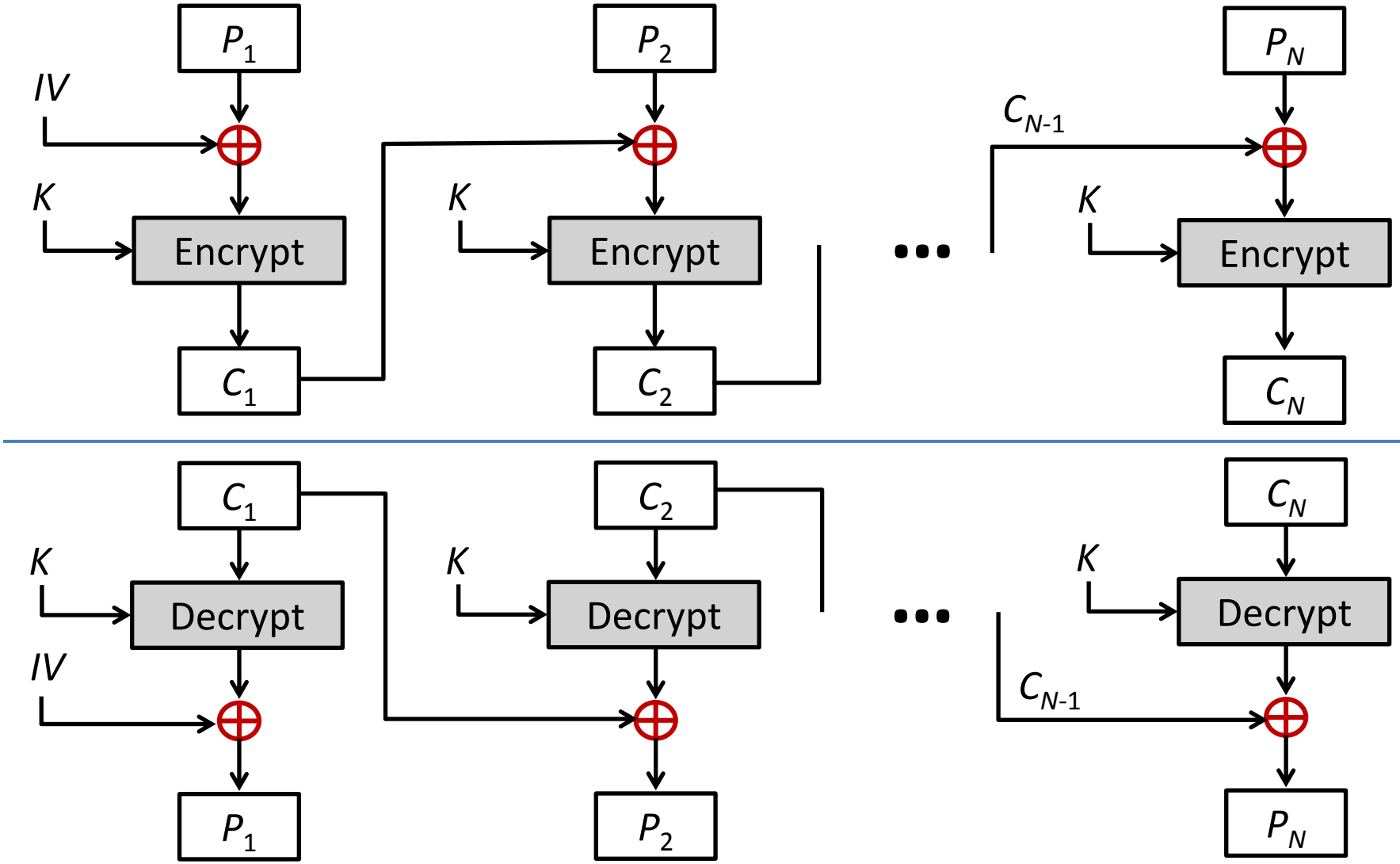
Electronic Code Book - Cont…

- **Strength:** it's simple.
- **Weakness:**
  - Repetitive information contained in the plaintext may show in the ciphertext also.
  - If the message has repetitive elements with a period of repetition a multiple of b bits, then these elements can be identified by the analyst.
- **Typical application:**
  - Secure transmission of short pieces of information (e.g. a temporary encryption key)

2. Cipher Block Chaining (CBC)

- **CBC** is a technique in which the same plaintext block, if repeated, produces different ciphertext blocks.
- In this scheme, the input to the encryption algorithm is the **XOR** of the current plaintext block and the preceding ciphertext block; the same key is used for each block.
- To produce the first block of ciphertext, an **initialization vector (IV)** is XORed with the first block of plaintext.
- On decryption, the **IV** is XORed with the output of the decryption algorithm to recover the first block of plaintext.

2. CBC - Encryption & Decryption



CBC	$C_1 = \mathrm{E}(K, [P_1 \oplus \mathrm{IV}])$ $C_j = \mathrm{E}(K, [P_j \oplus C_{j-1}])\ j = 2, \ldots, N$	$P_1 = \mathrm{D}(K, C_1) \oplus \mathrm{IV}$ $P_j = \mathrm{D}(K, C_j) \oplus C_{j-1}\ j = 2, \ldots, N$

2. Cipher Block Chaining (CBC) – Cont…

- **Strength:** because of the chaining mechanism of CBC, it is an appropriate mode for encrypting messages of length greater than b bits
- **Typical application:**
  - General-purpose block oriented transmission
  - Authentication

Cons:
No Parallelism

Pros:
Confidentiality + Availability

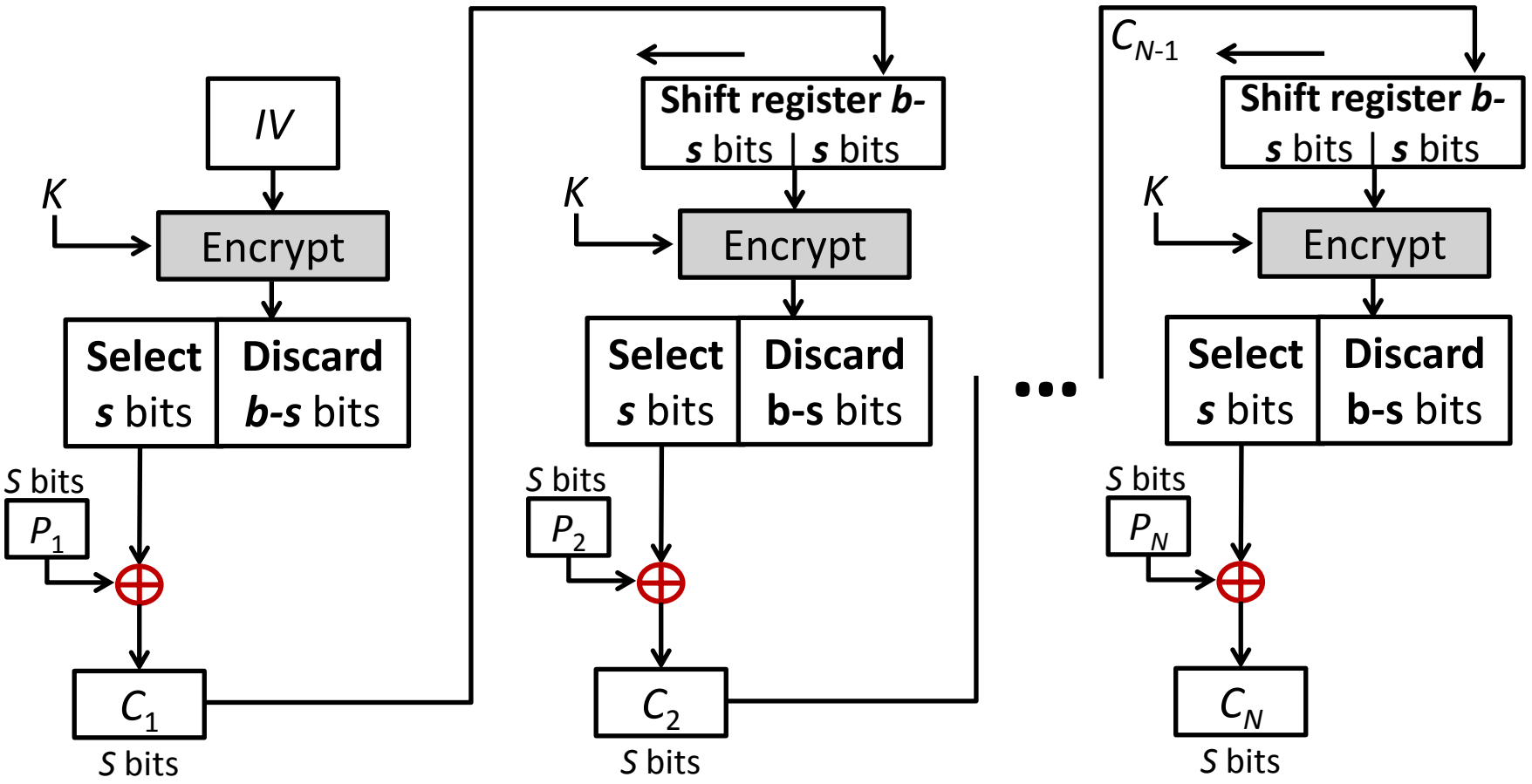
3. Cipher Feedback Mode (CFB)

- For AES, DES, or any block cipher, encryption is performed on a block of b bits. In DES, b = 64 and in AES, b = 128.

- However, it is possible to convert a block cipher into a stream cipher, using cipher feedback (CFB) mode, output feedback (OFB) mode, and counter (CTR) mode.

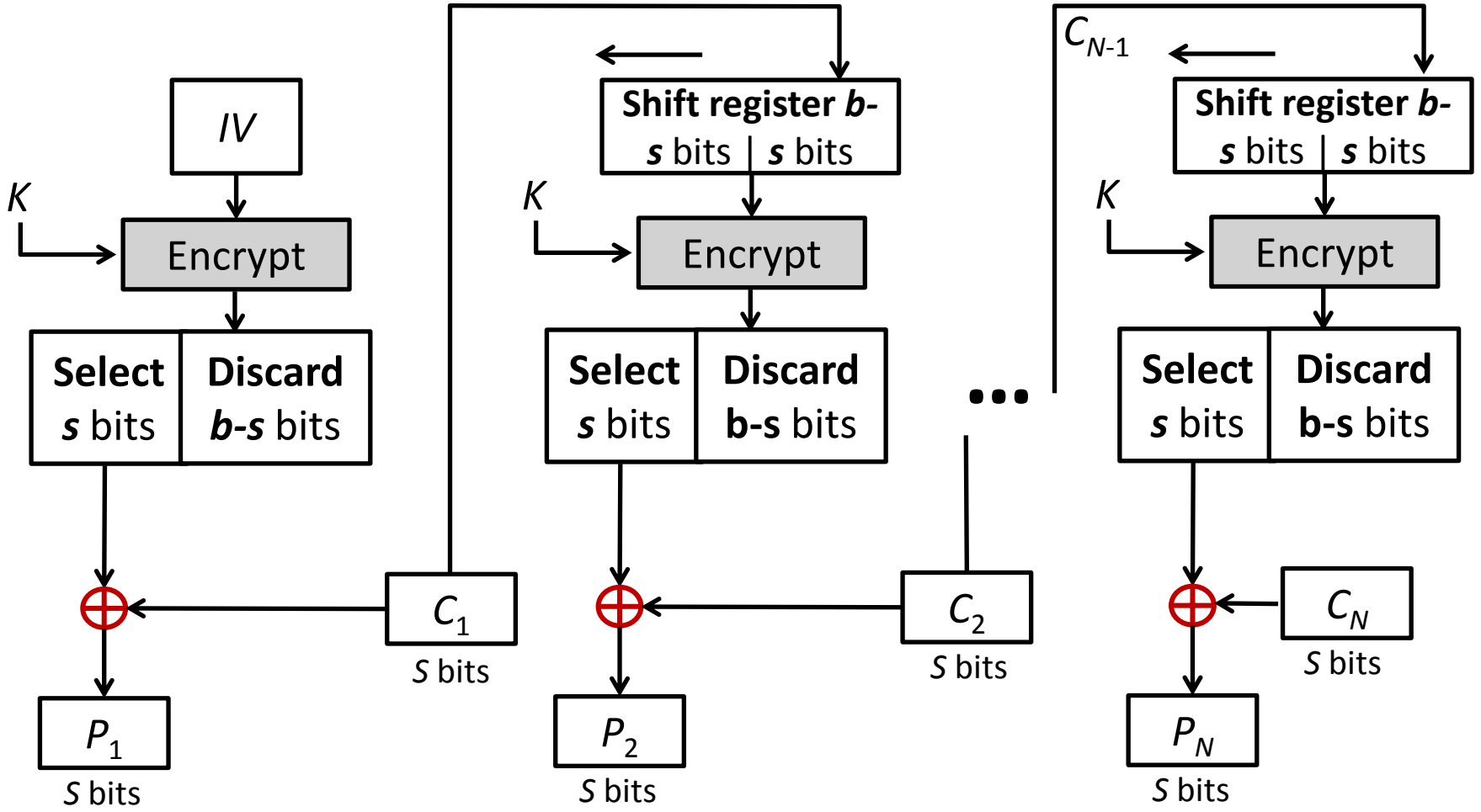
- A stream cipher eliminates the need to pad a message to be an integral number of blocks.

3. CFB Encryption



CFB	$I_1 = IV$ $I_j = \mathrm{LSB}_{b-s}(I_{j-1}) \| C_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $C_j = P_j \oplus \mathrm{MSB}_s(O_j) \quad j = 1, \ldots, N$

3. CFB Decryption



$$
\begin{aligned}
I_1 &= IV \\
I_j &= \mathrm{LSB}_{b-s}(I_{j-1}) \| C_{j-1} \quad j = 2, \ldots, N \\
O_j &= \mathrm{E}(K, I_j) \quad j = 1, \ldots, N \\
P_j &= C_j \oplus \mathrm{MSB}_s(O_j) \quad j = 1, \ldots, N
\end{aligned}
$$

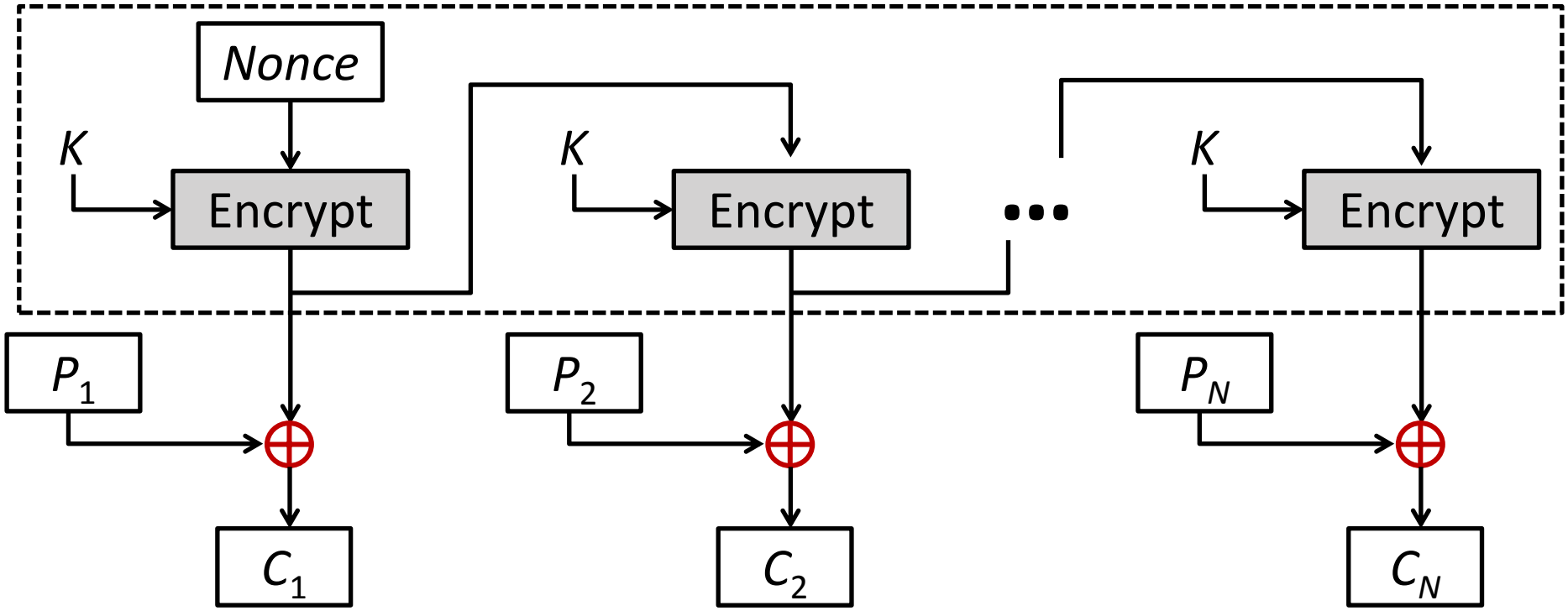
CFB Mode

- The input to the encryption function is a **b-bit shift register** that is initially set to some initialization vector (IV).
- The leftmost (most significant) **s bits** of the output of the encryption function are XORed with the first segment of plaintext **P1** to produce the first unit of ciphertext **C1** , which is then transmitted.
- In addition, the contents of the shift register are shifted left by **s bits**, and C1 is placed in the rightmost (**least significant**) s bits of the shift register.
- For **decryption**, the same scheme is used, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit.

4. Output Feedback Mode (OFB)

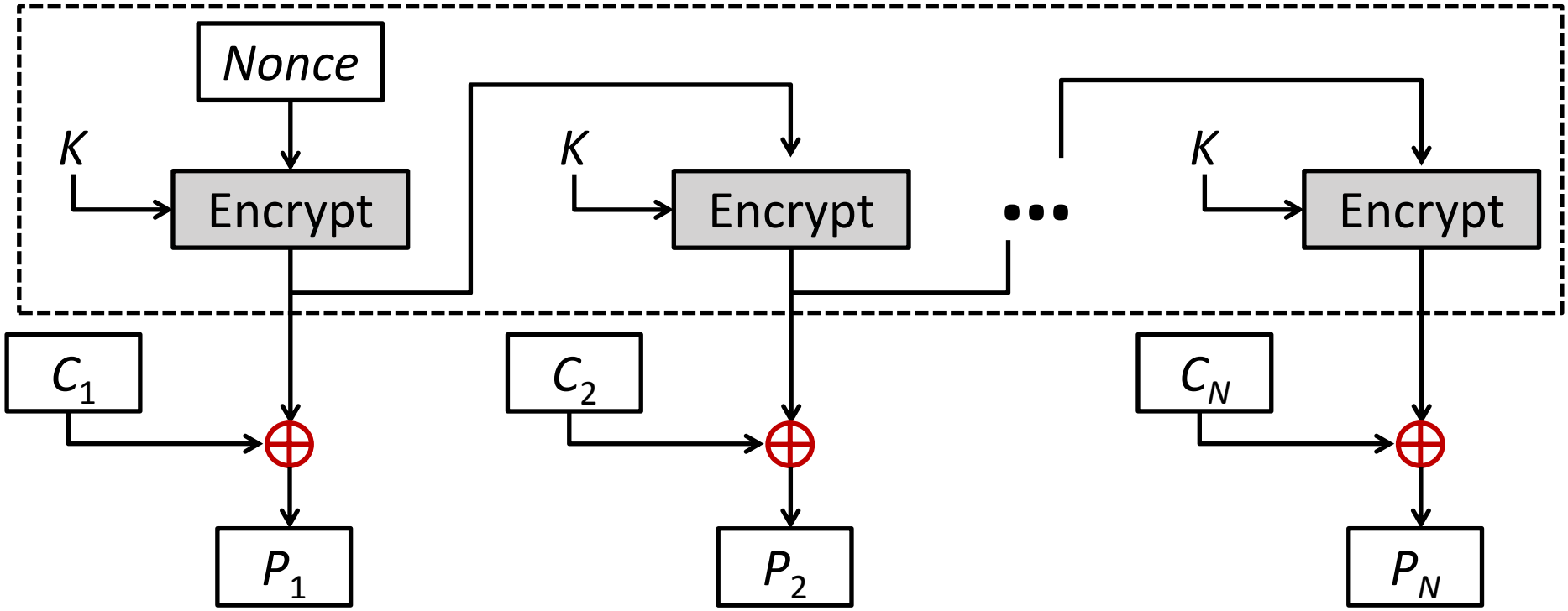
- The output feedback (**OFB**) mode is similar in structure to that of **CFB**.
- For **OFB**, the output of the encryption function is fed back to become the input for encrypting the next block of plaintext.
- In **CFB**, the output of the XOR unit is fed back to become input for encrypting the next block.
- The other difference is that the **OFB** mode operates on full blocks of plaintext and ciphertext, whereas **CFB** operates on an **s-bit** subset.
- **Nonce:** A time-varying value that has at most a negligible chance of repeating, for example, a **random value** that is freshly generated for each use, a timestamp, a sequence number, or some combination of these.

4. OFB Encryption



OFB	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $C_j = P_j \oplus O_j \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$

4. OFB Decryption



$$
\begin{aligned}
I_1 &= Nonce \\
I_j &= O_{j-1} \qquad j = 2, \ldots, N \\
O_j &= \mathrm{E}(K, I_j) \qquad j = 1, \ldots, N \\
P_j &= C_j \oplus O_j \qquad j = 1, \ldots, N-1 \\
P_N^* &= C_N^* \oplus \mathrm{MSB}_u(O_N)
\end{aligned}
$$

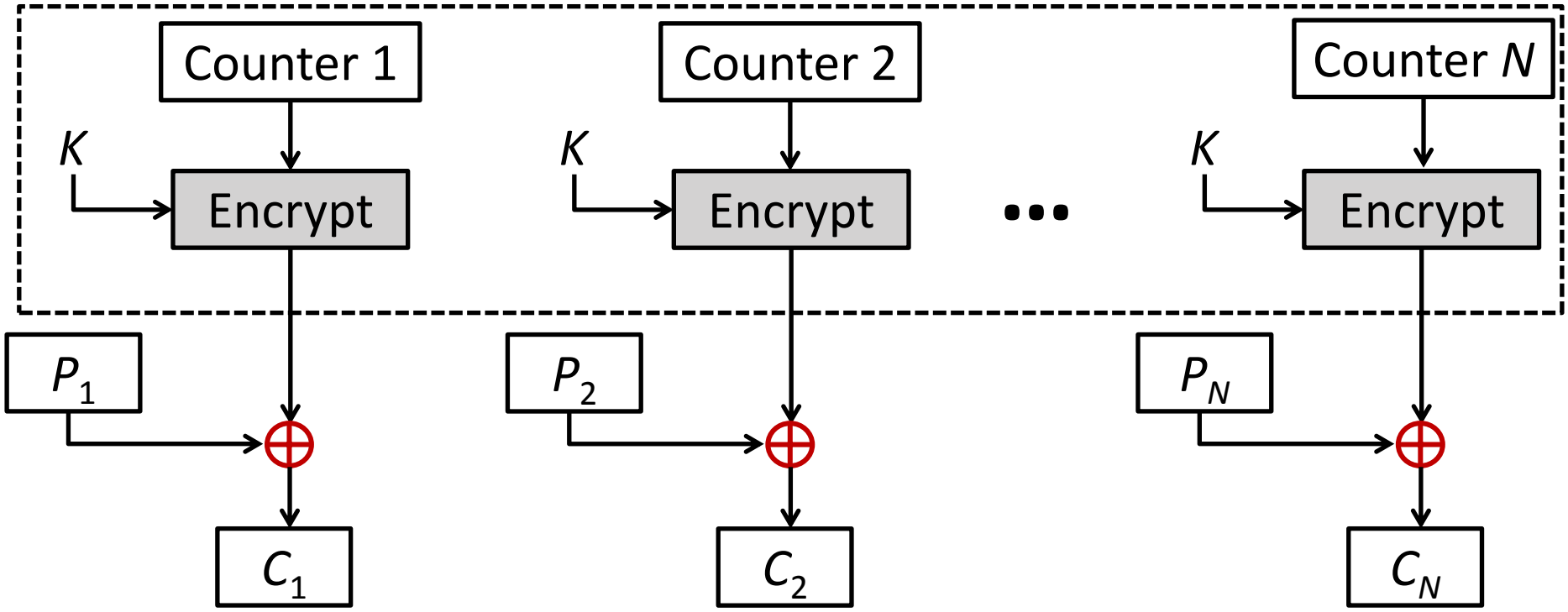
OFB Mode

- Each bit in the ciphertext is independent of the previous bit or bits. i.e. Feedback is independent of transmission
- This **avoids error propagation which**
- It allows many error correcting codes to function normally even when applied before encryption
- It helps recover from ciphertext bit errors, but cannot self-synchronize
- Pre-compute of forward cipher is possible
- Flipping a bit in the ciphertext produces a flipped bit in the plaintext at the same location

5. Counter Mode (CTR)

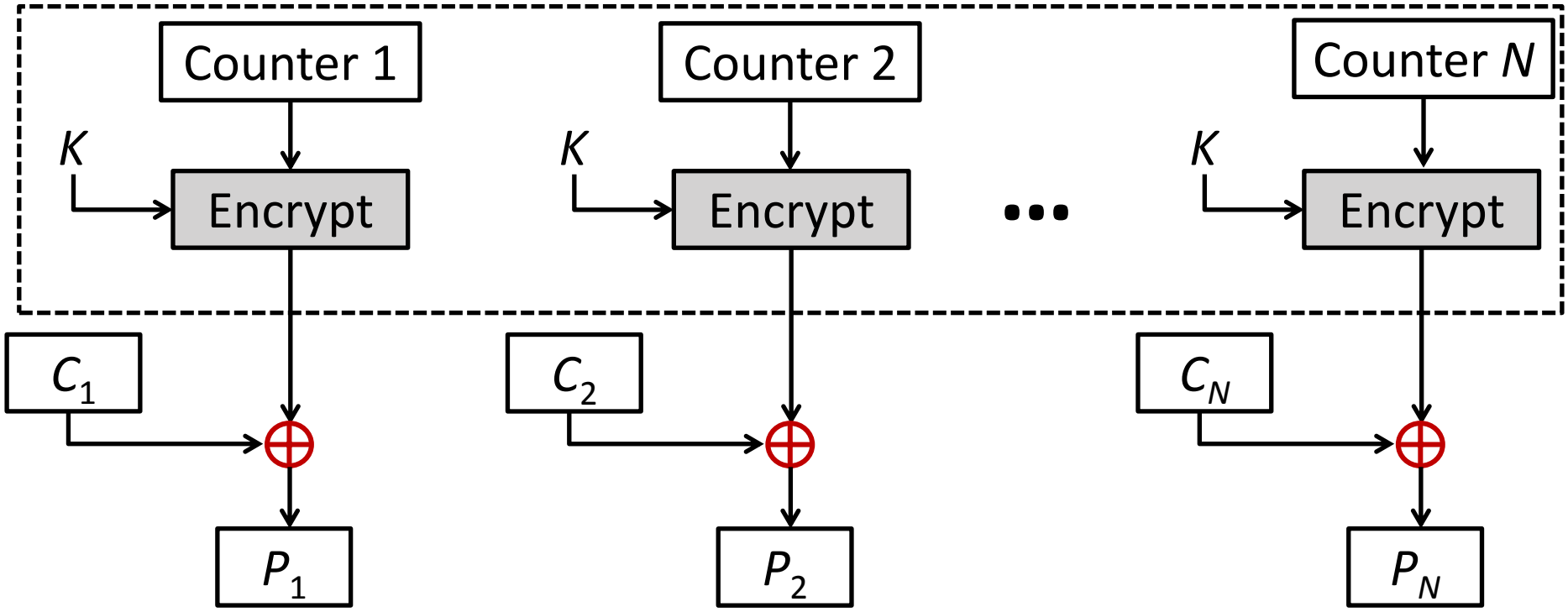
- Counter (**CTR**) mode has increased recently with applications to ATM (asynchronous transfer mode) network security and IP sec (IP security).
- A **counter** equal to the plaintext block size is used.
- The counter value must be different for each plaintext block that is encrypted.
- Typically, the counter is initialized to some value and then incremented by 1 for each subsequent block

5. CTR Encryption



CTR	$C_j = P_j \oplus \mathrm{E}(K, T_j) \qquad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$

4. CTR Decryption



$$P_j = C_j \oplus \mathrm{E}(K, T_j) \qquad j = 1, \ldots, N-1$$
$$P_N^* = C_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$$

Advantages of the CTR Mode

- Strengths:
  - Needs only the encryption algorithm
  - Random access to encrypted data blocks
  - blocks can be processed (encrypted or decrypted) in parallel
  - Simple; fast encryption/decryption

- Counter
  - Must be unknown and unpredictable
  - pseudo-randomness in the key stream is a goal

Summary of All Modes

Operation Mode	Description	Type of Result
ECB	Each n-bit block is encrypted independently with same key	Block Cipher
CBC	Same as ECB, but each block is XORed with previous cipher text	Block Cipher
CFB	Each s-bit block is XORed with s-bit key which is part of previous cipher text	Stream Cipher
OFB	Same as CFB, except that the input to the encryption algorithm is the output of preceding encryption algorithm	Stream Cipher
CTR	Same as OFB, but a counter is used instead of nonce	Stream Cipher

Typical Applications of All Modes

Mode	Typical Application
Electronic Codebook (ECB)	• Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	•General-purpose block-oriented transmission •Authentication
Cipher Feedback (CFB)	•General-purpose stream-oriented transmission •Authentication
Output Feedback (OFB)	•Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	• General-purpose transmission •Useful for high-speed requirements

Use of Stream Ciphers

- Although the five modes of operations enable the use of block ciphers for encipherment of messages or files in large units (ECB, CBC, and CTR) and small units (CFB and OFB), sometimes pure streams are needed for enciphering small units of data such as characters or bits.
- Stream ciphers are more efficient for real-time processing.
- Several stream ciphers have been used in different protocols during the last few decades.
- Examples: RC4, A5/1

RC4

- RC4 means Rivest Cipher 4 invented by Ron Rivest in 1987 for RSA Security. It is a Stream Ciphers.
- Stream Ciphers operate on a stream of data byte by byte.
- RC4 stream cipher is one of the most widely used stream ciphers because of its simplicity and speed of operation.
- It is a variable key-size stream cipher with byte-oriented operations. It uses either 64 bit or 128-bit key sizes.
- It is generally used in applications such as Secure Socket Layer (SSL), Transport Layer Security (TLS), and also used in IEEE 802.11 wireless LAN std.

A5/1

- **A5/1** is a stream cipher used to provide over-the-air communication privacy in the GSM cellular telephone standard.
- It is one of several implementations of the A5 security protocol.

Synchronous & Asynchronous Stream Ciphers

- With **synchronous stream** ciphers, the bits of the key stream do not depend on the ciphertext bits.
- With **asynchronous stream** ciphers the key stream can be inferred (provided one knows the secret key) from previous bits of the cipher stream.
- CTR would be an example of a synchronous streaming mode and CFB would be an example of an asynchronous mode.

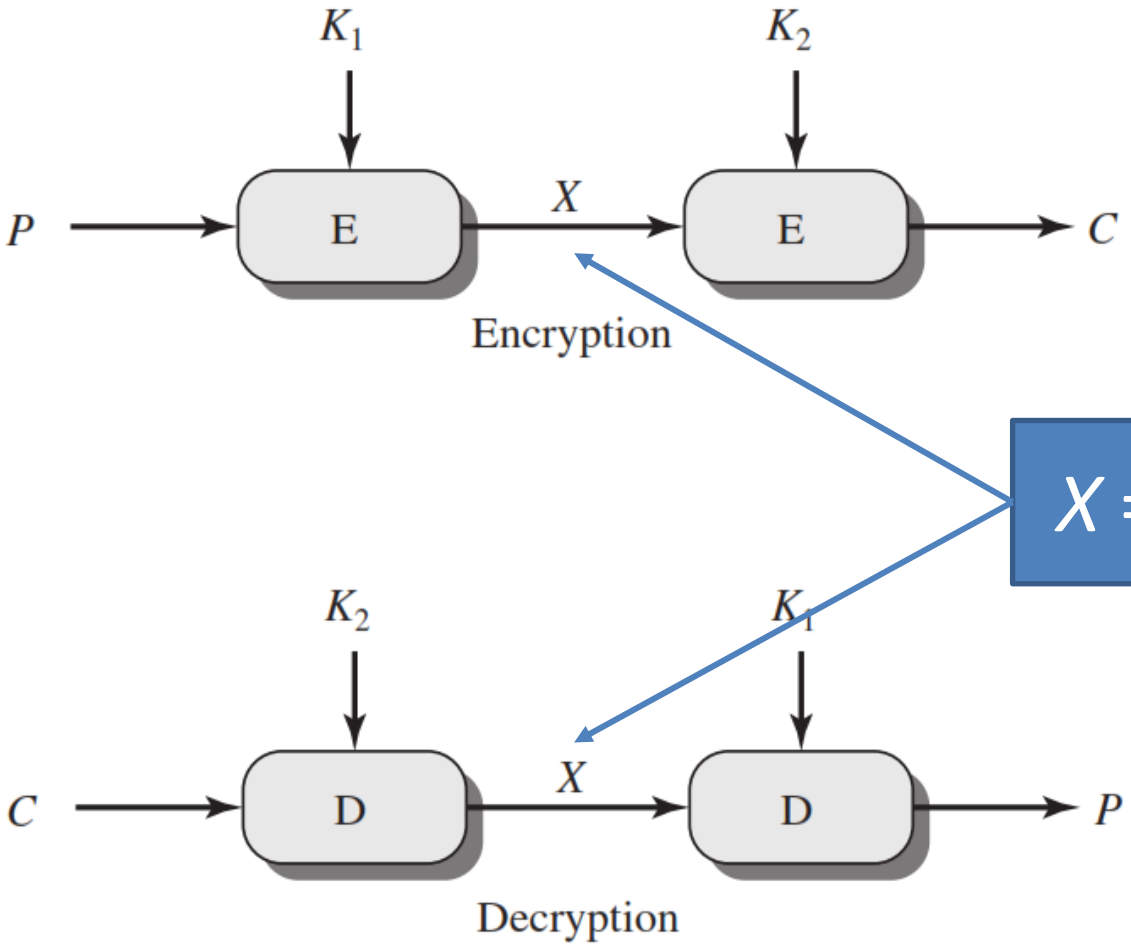
Synchronous & Asynchronous stream ciphers (Cont.)

- Synchronous ciphers have these advantages and disadvantages:
  - Key stream can be pre-computed before plaintext or ciphertext is provided, often with parallelism. typically faster and more efficient than block ciphers when encrypting large volumes of data in real-time
  - As the keystream is deterministic, they have the disadvantage of ciphertext malleability (a known change in ciphertext produces a known change in plaintext) and so will need to be combined with some form of message authentication.
- Asynchronous ciphers have these advantages and disadvantages:
  - The key stream can only be computed once the plaintext/ciphertext is provided. However, changing the ciphertext changes subsequent key stream and so there is considerably less malleability.
  - They also allow easy synchronization at any point in transmission.
  - They are more computationally intensive and slower than synchronous stream ciphers because of the additional complexity in generating the keystream

Multiple Encryption

- Given the **potential vulnerability of DES to a brute-force attack**, there has been considerable interest in finding an alternative.
- One approach is to design a completely new algorithm, of which **AES** is a prime example.
- Another alternative, which would preserve the existing investment in software and equipment, is to use **multiple encryption with DES and multiple keys**.

Double DES



$C=E(K_2,E(K_1, P))$

$X = E(K_1, P) = D(K_2, C)$

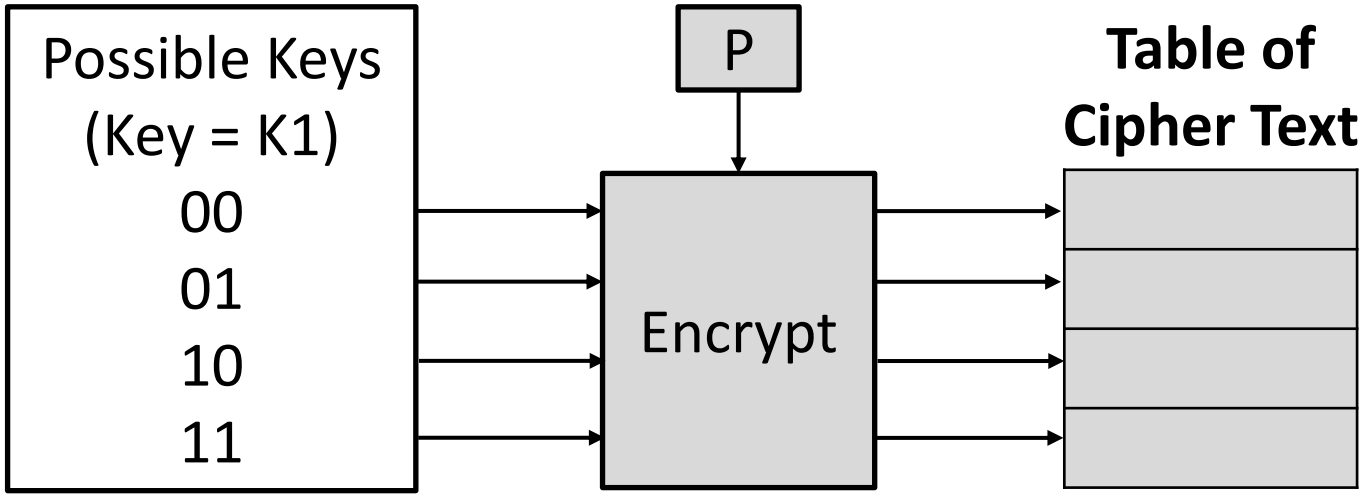
$P=D(K_1,D(K_2, C))$

Meet in the Middle Attack

- **Meet-in-the-middle (MITM) attacks** are often executed to decode multiple data encryption standard (DES) techniques.
- This attack involves encryption from one end, decryption from the other and matching the results in the middle.
- An attacker can use a MITM attack to bruteforce **Double DES**
  - an attacker encrypts the plaintext with all possible keys for the first encryption (2^56 operations) and then decrypts the ciphertext with each possible key for the second encryption (2^56 operations)

Meet in the Middle Attack Step-1

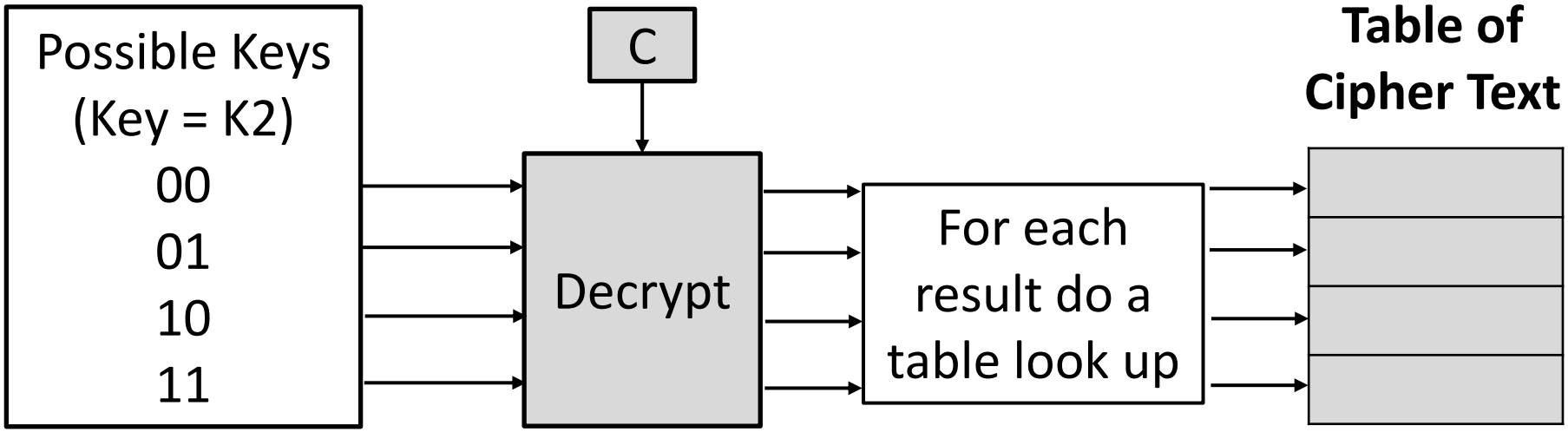
- Suppose cryptanalyst knows **P** and corresponding **C**.
- Now, the aim is to obtain the values of K_1 and K_2.
- For all possible values (2^{56}) of key K1, the cryptanalyst would encrypt the P by performing E(K1,P).
- The cryptanalyst would store output in a table.



Cryptanalyst encryption operation

Meet in the Middle Attack Step-2

- Cryptanalyst decrypts the known **C** with all possible values of **K2**.
- In each case cryptanalyst will **compare** the resulting value with the all values in the table of ciphertext.



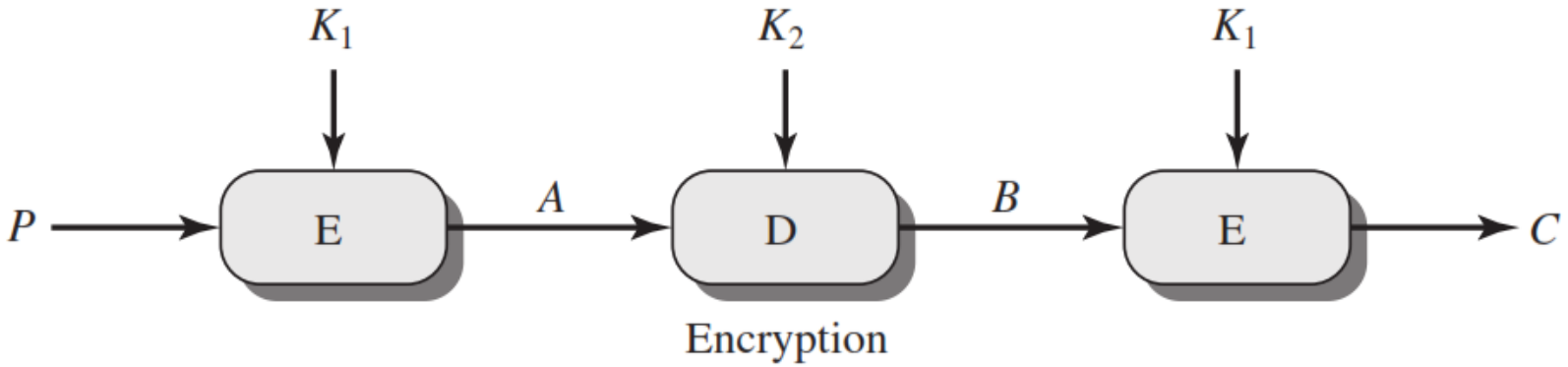
Cryptanalyst decryption operation

Thus, K1 and K2 can be obtained

Triple DES

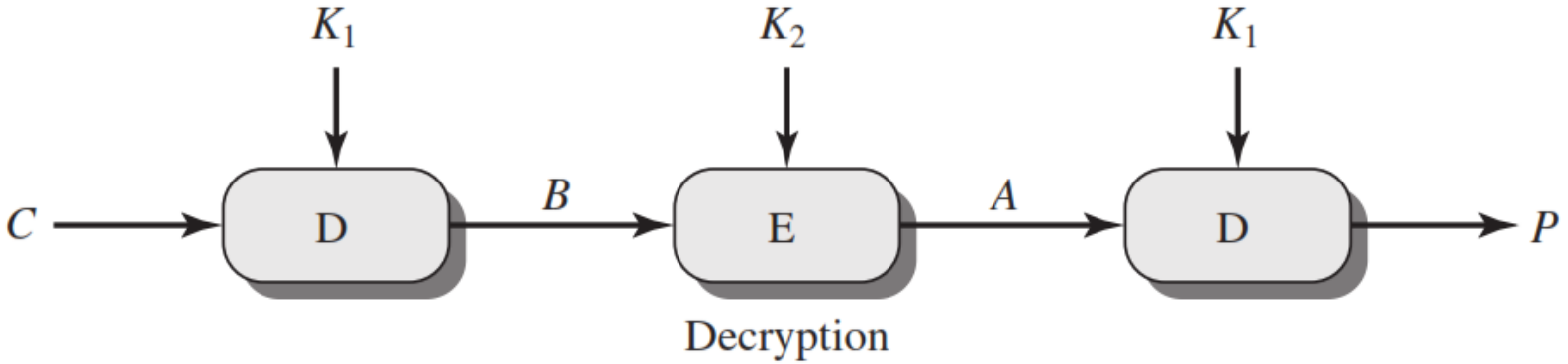
- An obvious counter to the meet in the middle attack is to use 3 stages of encryption with 3 different keys
- However, if 3 different keys are used, it has limitations of requiring a key length of 56x3= 168 bits which may be somewhat unwisely.
- As an alternative, Tuchman proposed a triple encryption method that uses **only 2 keys**.

Triple DES



Encryption

$$C=E(K_1,D(K_2,\ E(K_1,P)))$$



Decryption

$$P=D(K_1,E(K_2,\ D(K_1,C)))$$