COMPSCI4062&5063: Cyber Security Fundamentals

# Topic 4: Cryptography I

Dongzhu Liu

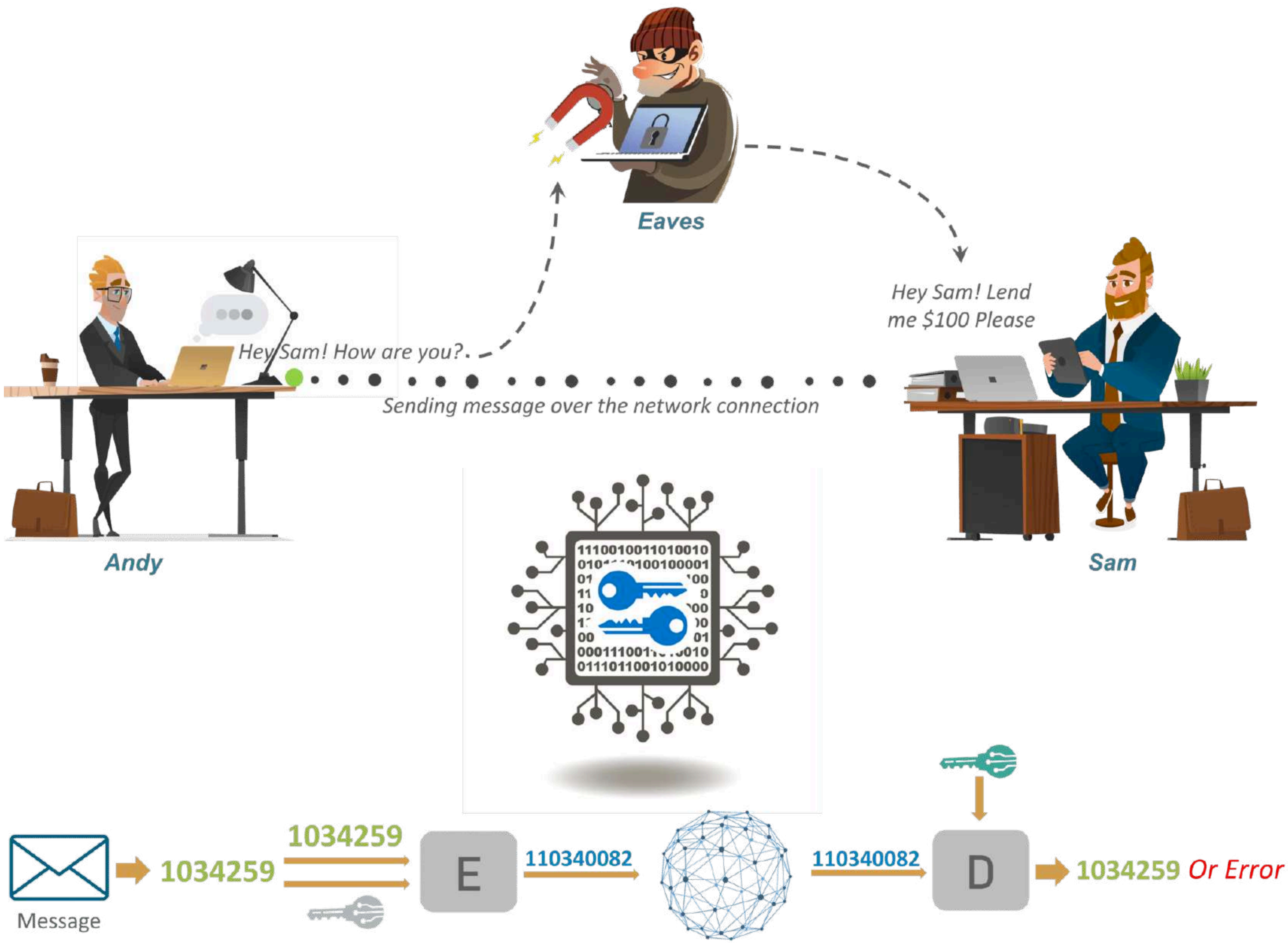
Email: dongzhu.liu@glasgow.ac.uk
Office: SAWB 510 (b)

University of Glasgow | School of Computing Science

# Overview

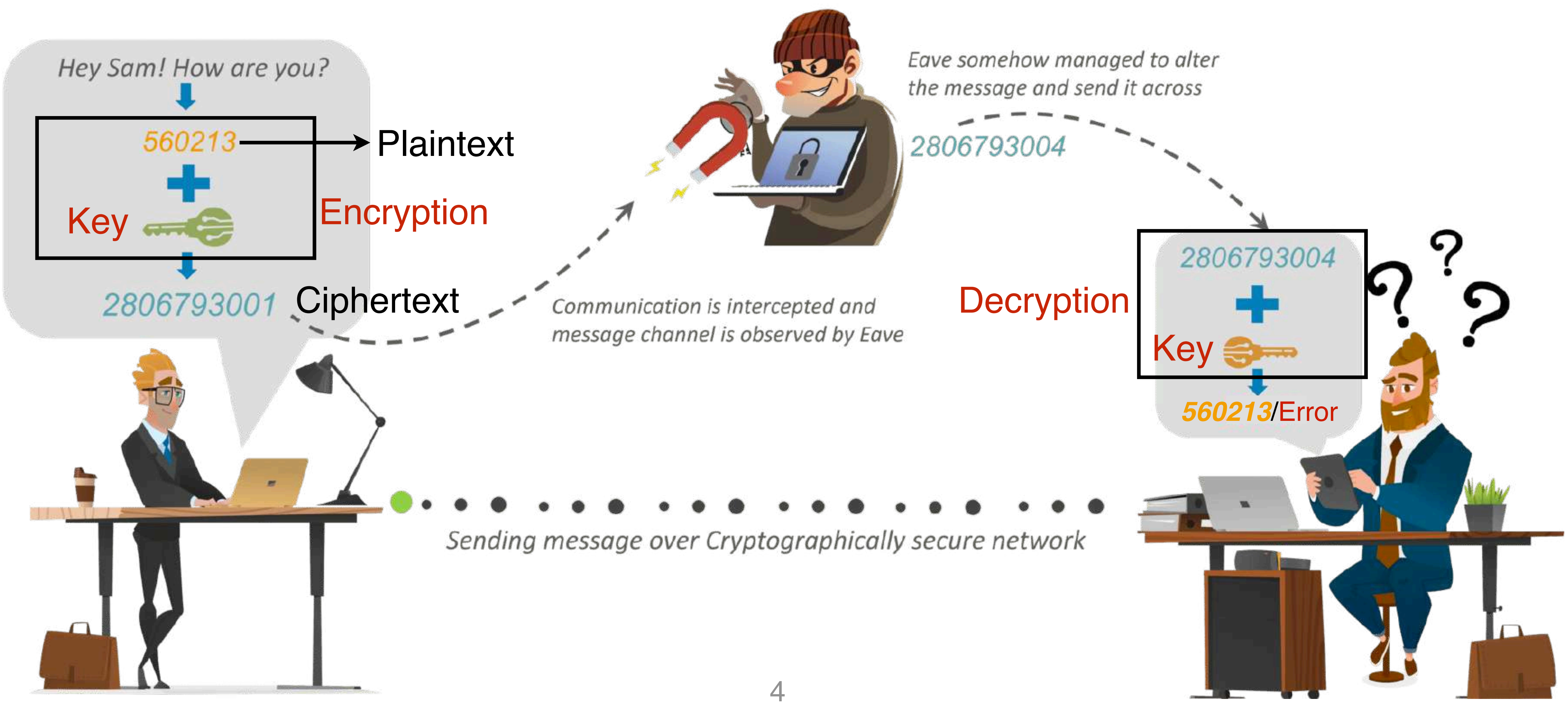
- Cryptography
  - Context
  - Ingredients
  - Classification
  - Attacks
- Symmetric Encryption: Block Cipher
  - DES/Triple DES (Feistel Cipher Structure)
  - AES
- Symmetric Encryption: Stream Cipher
  - RC4
- Cipher Block Modes of Operation

# Cryptography: Context



# Cryptography: Ingredients

- Plaintext: original message
- Encryption algorithm: substitution/transformations on the plain text
- Secret key: algorithm input, substitution/transformations depends on the key
- Ciphertext: algorithm output, depends on plain text and secret key
- Decryption algorithm: reverse of encryption



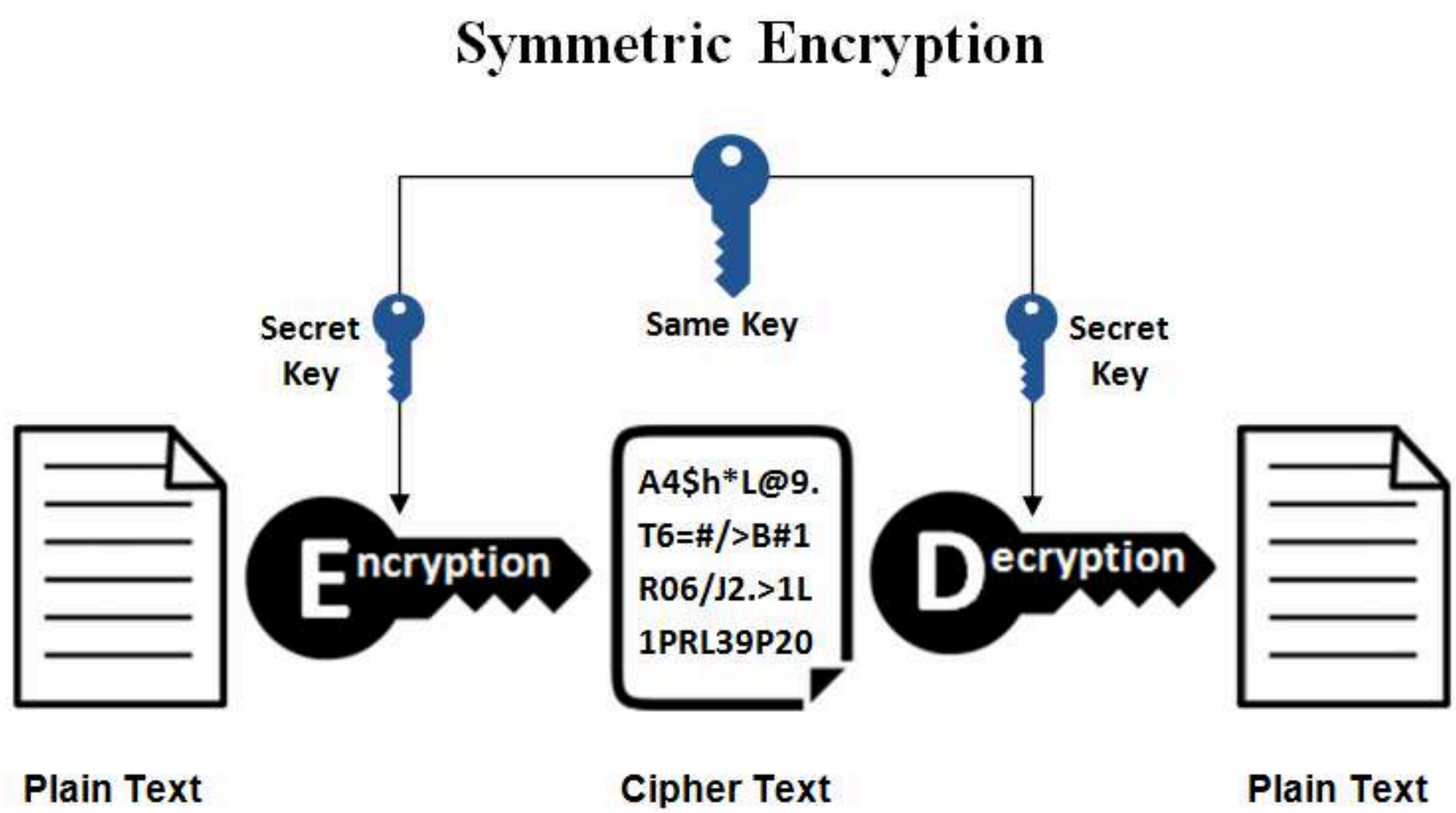
# Cryptography: Classification

- The type of operations used for transforming plaintext to cipher text
  - Substitution: each element in the plaintext is mapped into another element
  - Transposition: elements in the plaintext are rearranged

  Fundamental requirement: No information be lost

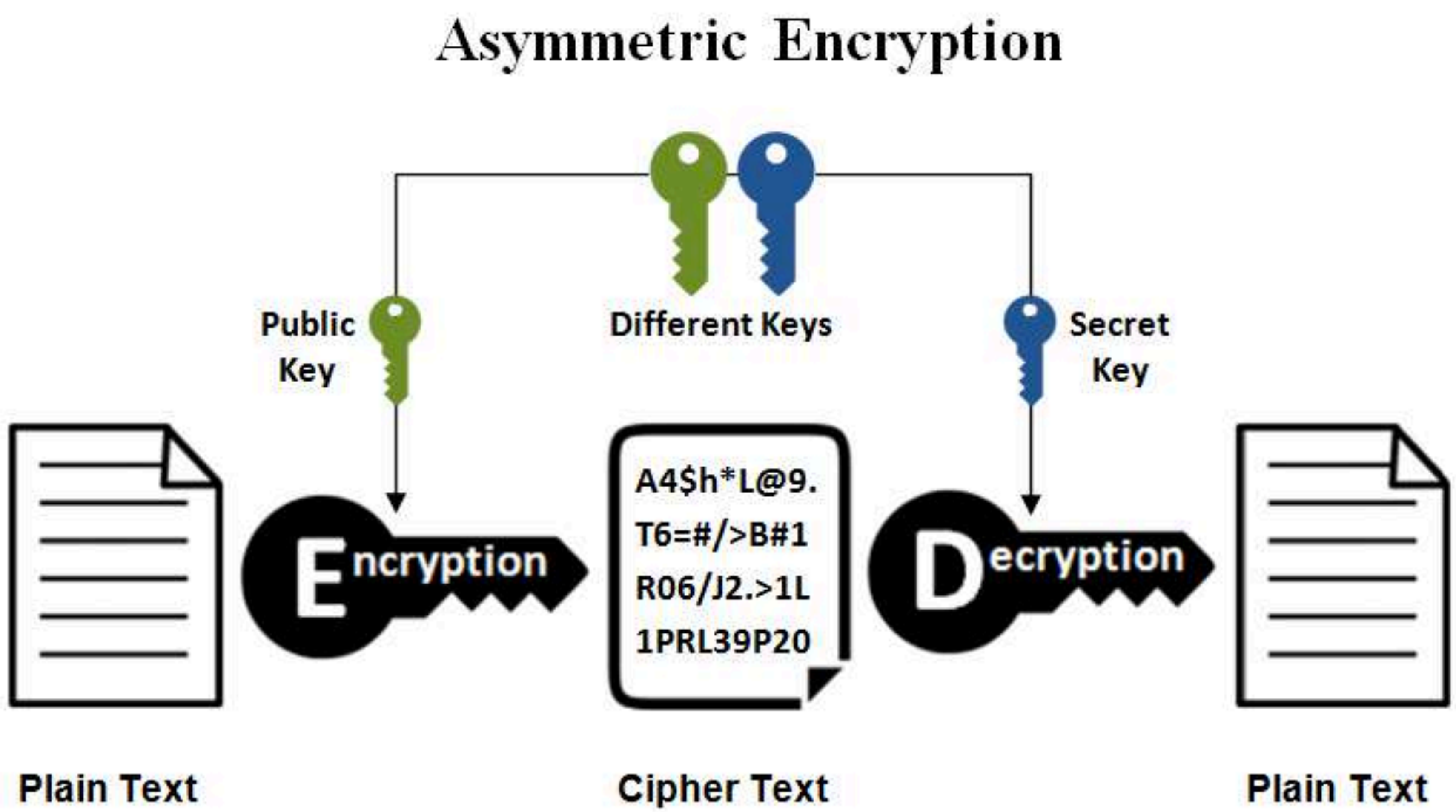
- The way in which the plaintext is processed
  - Block cipher: processes the input of one block of elements at a time
  - Stream cipher: processes the input elements continuously
- The number of keys used
  - Same key at sender and receiver — **Symmetric Encryption**
  - Different keys at sender and receiver — **Asymmetric Encryption**

# Symmetric Encryption




- Communication overhead to share the key
- To receive information from multiple sender, the secret key is shared among them, or create different keys for each sender.

# Asymmetric Encryption




- Public key is freely available to anyone who is a sender
- Encryption by public key can only be decrypted by secrete key
- What about encryption by secret key?

# Types of Attacks on Encrypted Messages

| Type of Attack | Known to Cryptanalyst |
|---|---|
| Ciphertext only | • Encryption algorithm<br>• Ciphertext to be decoded |
| Known plaintext | • Encryption algorithm<br>• Ciphertext to be decoded<br>• One or more plaintext-ciphertext pairs formed with the secret key |
| Chosen plaintext | • Encryption algorithm<br>• Ciphertext to be decoded<br>• Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key |
| Chosen ciphertext | • Encryption algorithm<br>• Ciphertext to be decoded<br>• Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |
| Chosen text | • Encryption algorithm<br>• Ciphertext to be decoded<br>• Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key<br>• Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |

Easier (defend)

Harder (defend)

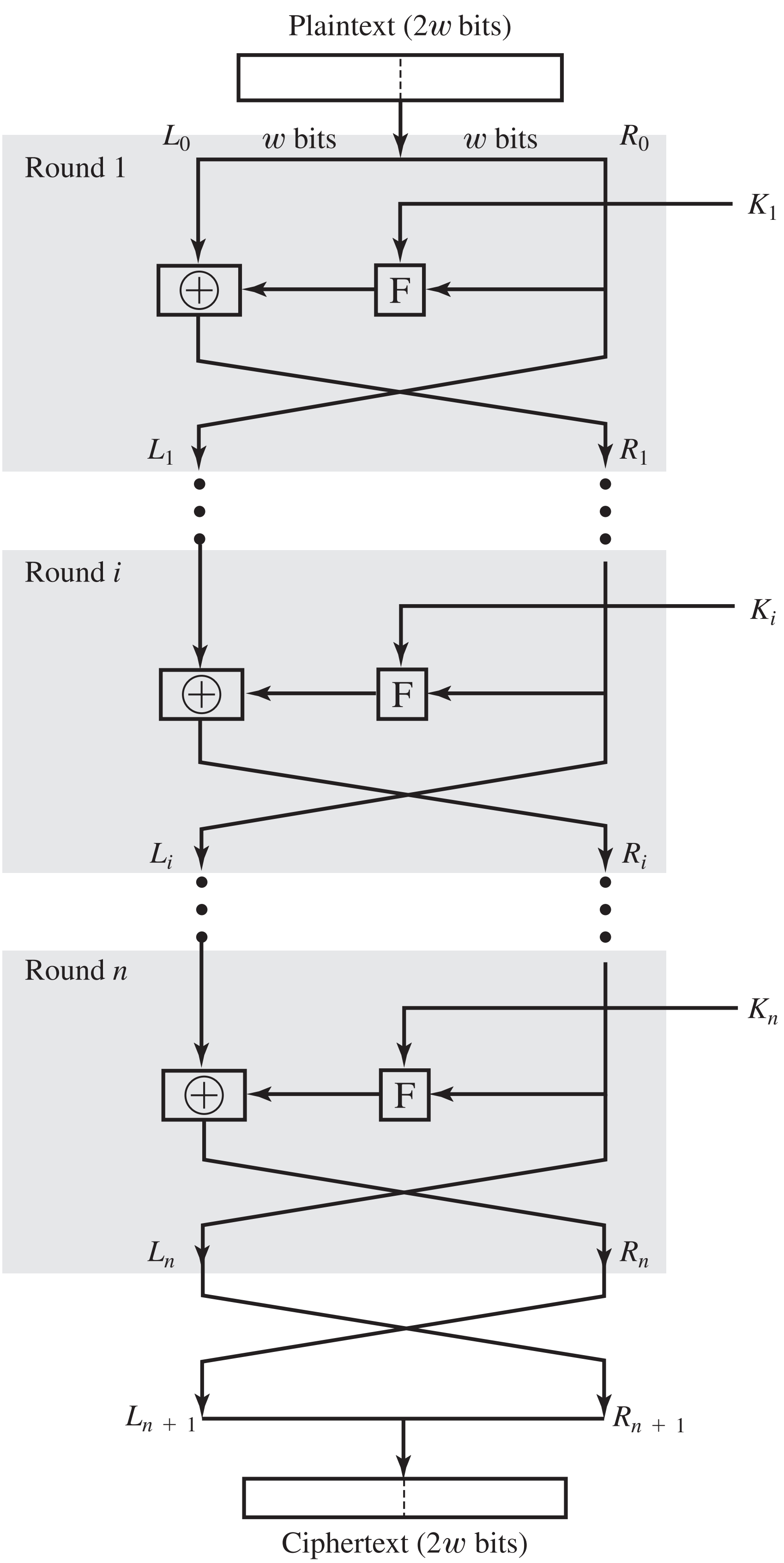
# Types of Attacks on Encrypted Messages

- Ciphertext only attack
  - Encryption algorithm
  - Ciphertext to be decoded

How to decrypt the message?

# Symmetric Encryption: Feistel Cipher Structure

- A structure for symmetric block encryption algorithm
  - The plaintext block is divided into two halves, $L_0$ and $R_0$
  - The two halves of the data pass through *n* rounds of processing and then combine to produce the ciphertext block.
  - The subkeys $K_i$ are different from each other
  - Applying a round function F to the right half of the data and then taking the XOR of the output of F and the left half of the data (substitution on the left half)

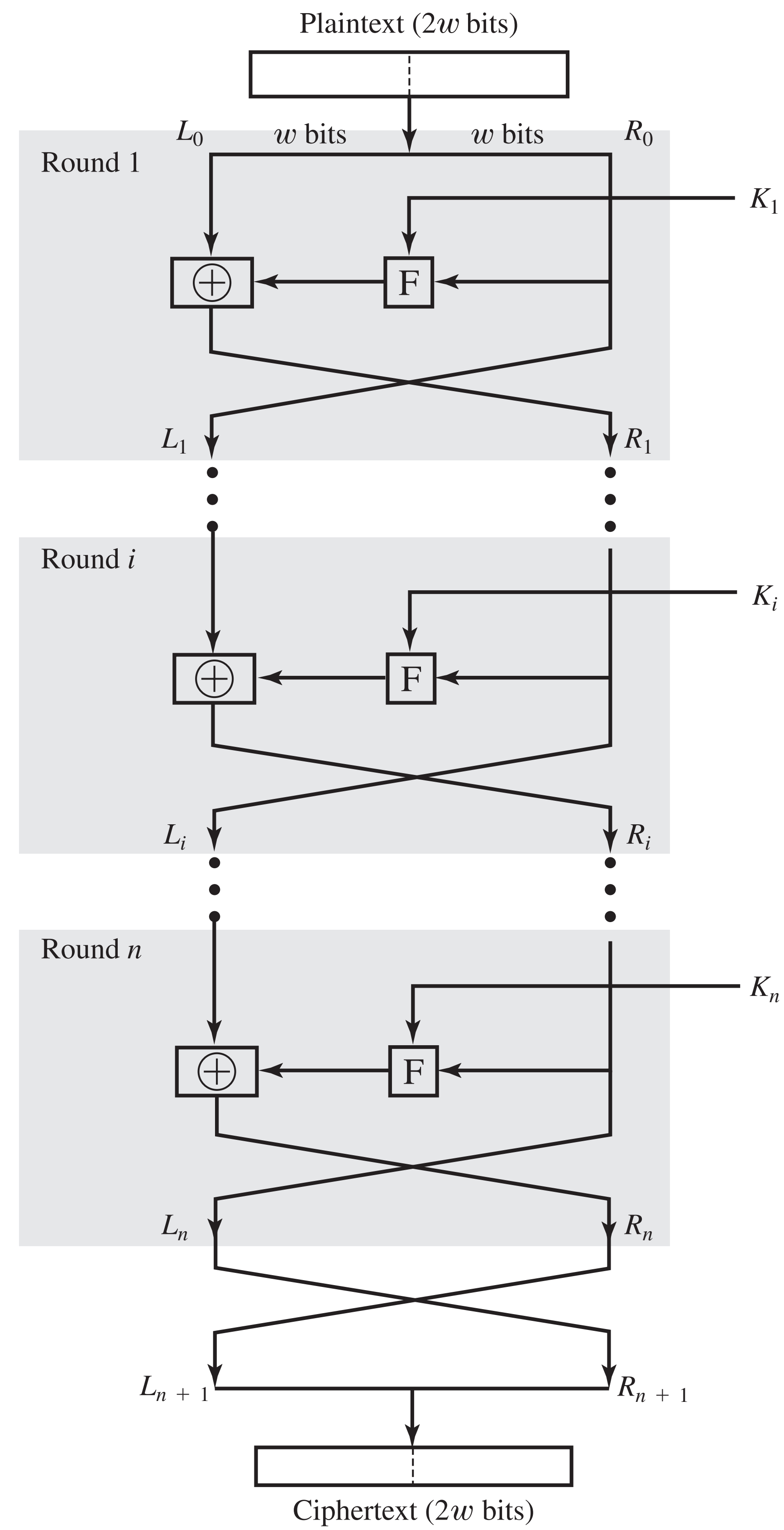


# Symmetric Encryption: Design Features

- Block Size: larger block sizes mean greater security but reduced encryption/decryption speed. A block size of 128 bits is a reasonable tradeoff and is nearly universal among recent block cipher designs.
- Key Size: similar to block size, most common key length 128 bits.
- Number of rounds: 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.
- Rond function: similar to subkey generation
- Fast software encryption/decryption: In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation. Accordingly, the speed of execution of the algorithm becomes a concern.
- Ease of Analysis: if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength.

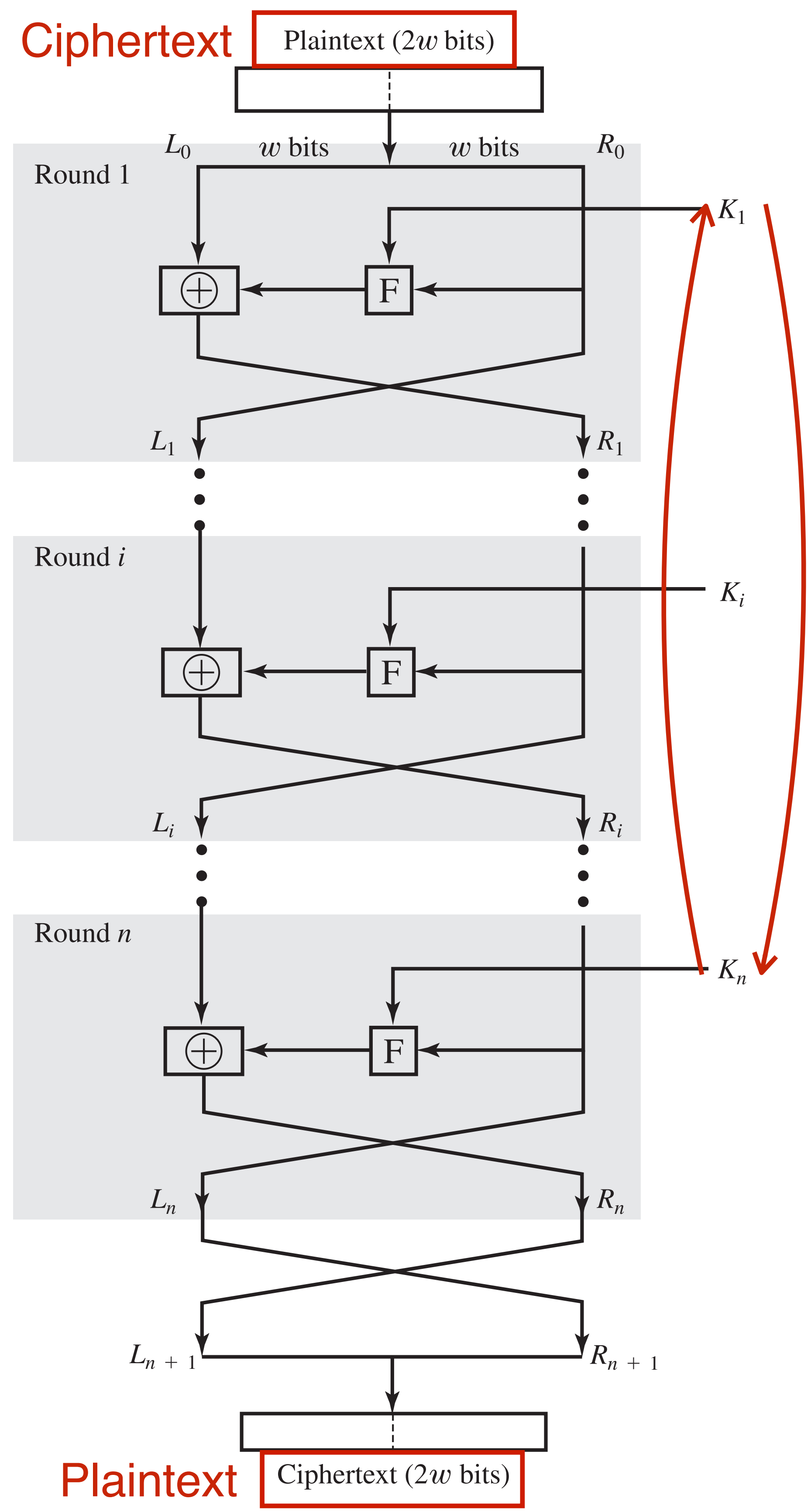
# Symmetric Encryption: DES

- Data Encryption Standard
  - Symmetric block cipher
  - Plaintext 64 bits, longer plaintext are processed in 64-bit blocks
  - Key 56 bits
  - 16 rounds of processing
  - 16 subkeys are generated from the original key



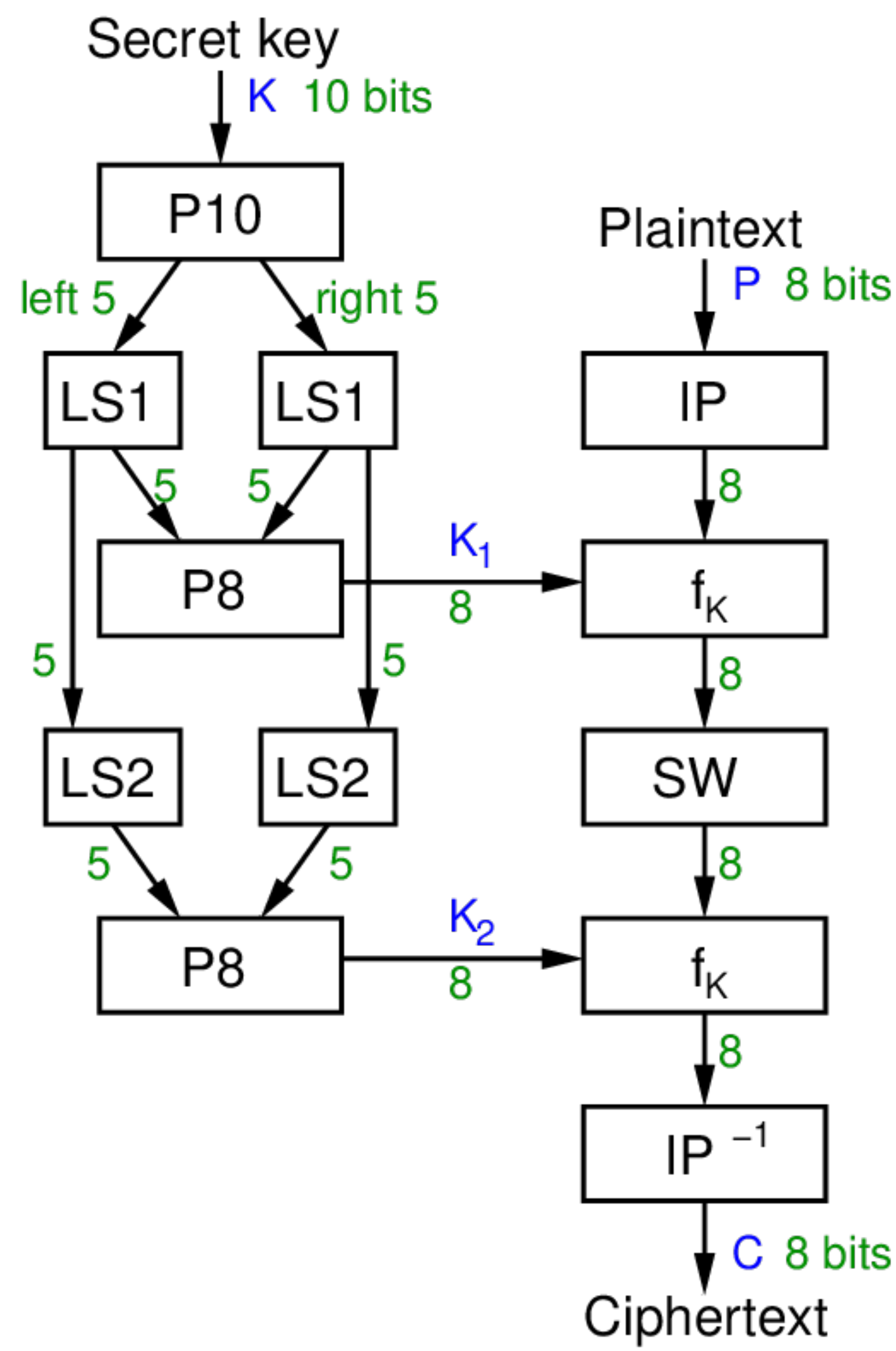
# Symmetric Encryption: DES

- Decryption DES
  - Use the ciphertext as input to the DES algorithm
  - Use the subkeys $K_i$ in reverse order



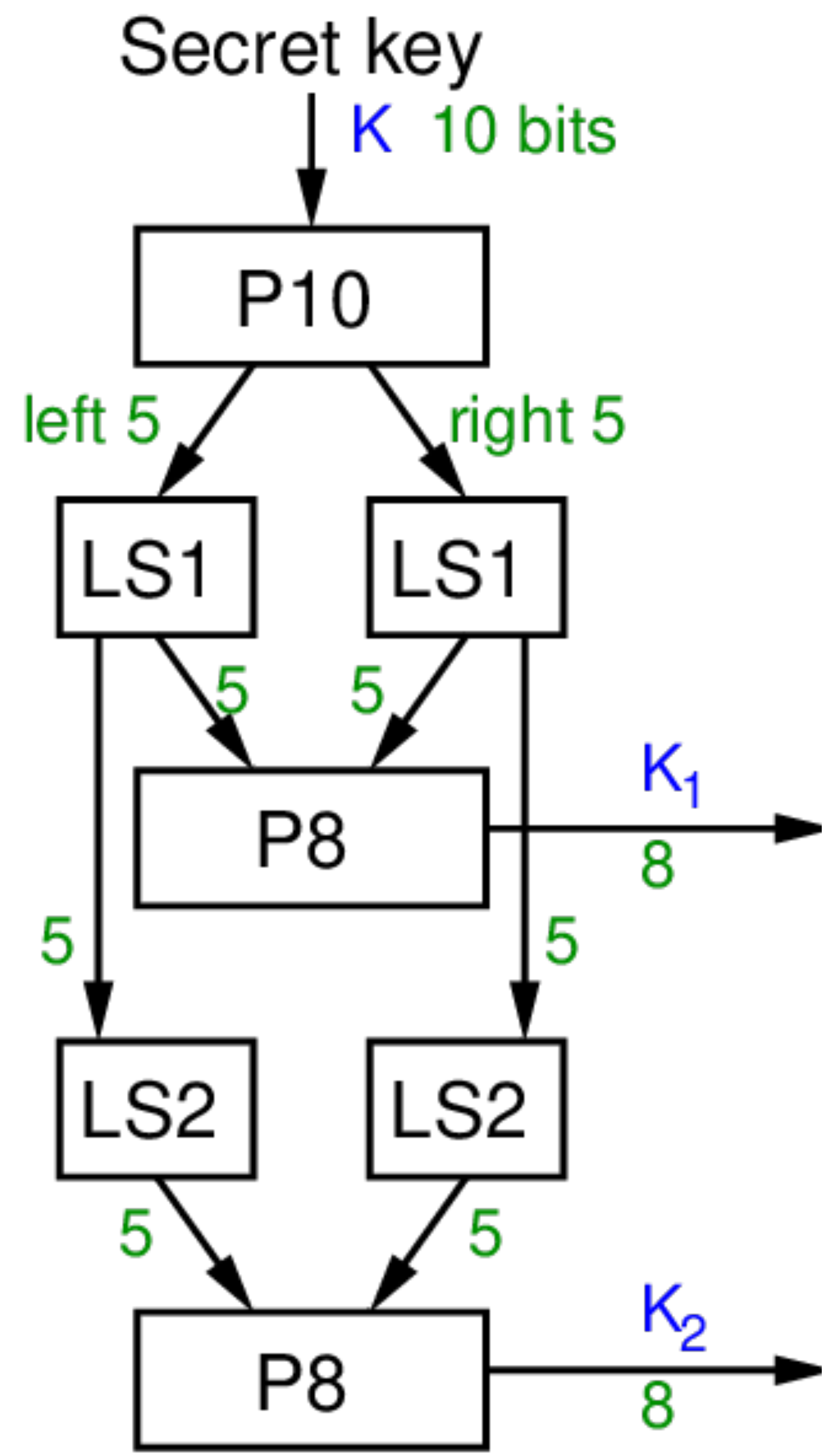
# Symmetric Encryption: DES

- A simplified example for DES
  - Plaintext 8 bits
  - Ciphertext 8 bits
  - Key 10 bits
  - Rounds 2
  - Subkeys generated using permutations and left shifts
  - Encryption: initial permutation, round function, switch halves
  - Decryption: Same as encryption, except round keys used in opposite order



# Symmetric Encryption: DES

- A simplified example for DES
  - Key generation
  - Original key: 1 0 1 0 0 0 0 0 1 0
  - After P10: 1 0 0 0 0 0 1 1 0 0
  - Left 5: 1 0 0 0 0   Right 5: 0 1 1 0 0
  - LS1(left 5): 0 0 0 0 1   LS1(right 5): 1 1 0 0 0
  - Input P8: 0 0 0 0 1 1 1 0 0 0
  - K1: 1 0 1 0 0 1 0 0




P10 (permutate)

```
Input : 1  2  3  4  5  6  7  8  9  10
Output: 3  5  2  7  4  10 1  9  8  6
```

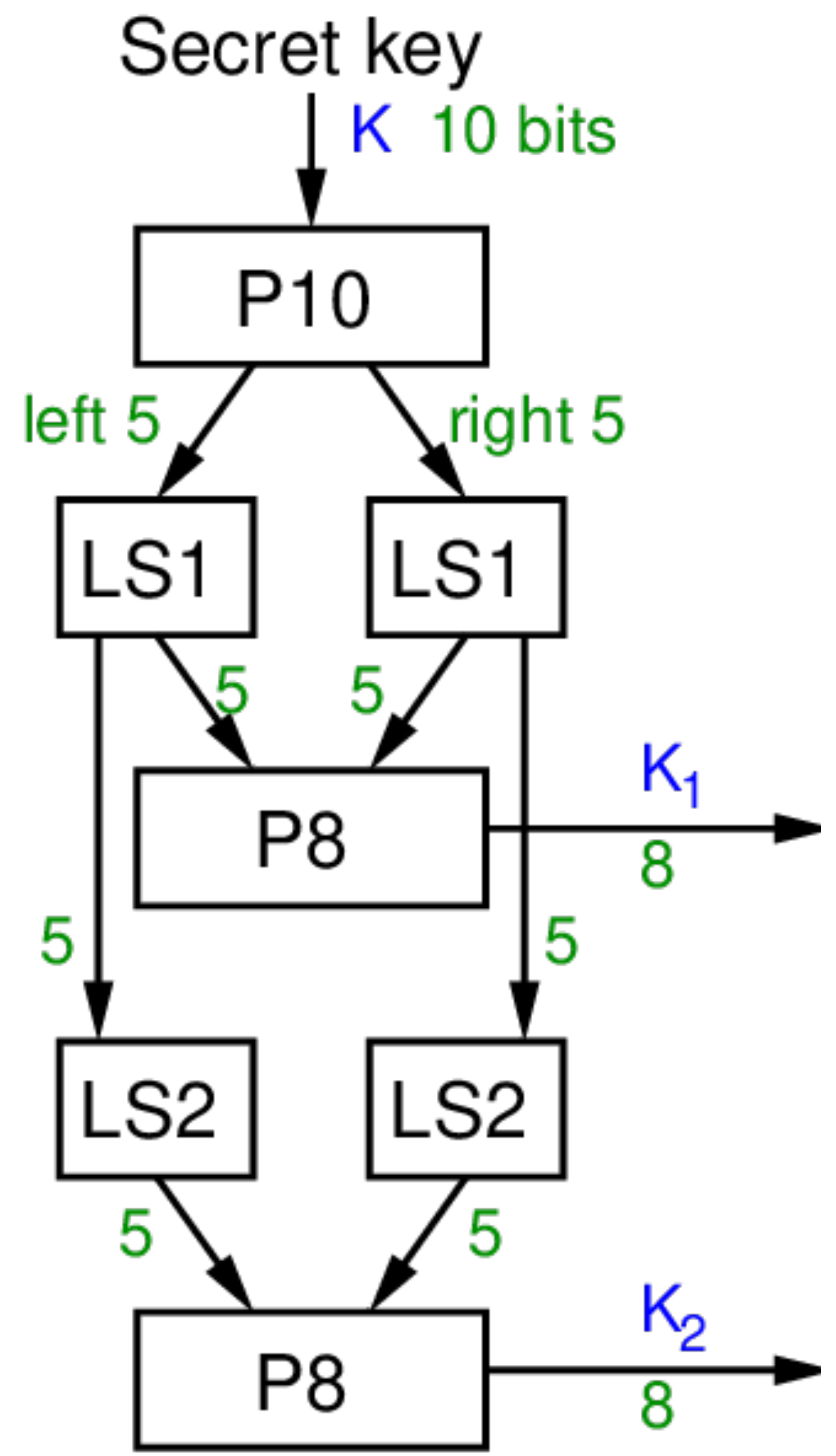
P8 (select and permutate)

```
Input : 1  2  3  4  5  6  7  8  9  10
Output: 6  3  7  4  8  5  10 9
```

LS-1: left shift by 1 position
LS-2: left shift by 2 positions

# Symmetric Encryption: DES

- A simplified example for DES
  - Key generation
  - LS1(left 5): 0 0 0 0 1   LS1(right 5): 1 1 0 0 0
  - Exercise: How to get K2?




P10 (permutate)

```
Input : 1 2 3 4 5 6 7 8 9 10
Output: 3 5 2 7 4 10 1 9 8 6
```

P8 (select and permutate)

```
Input : 1 2 3 4 5 6 7 8 9 10
Output: 6 3 7 4 8 5 10 9
```

LS-1: left shift by 1 position

LS-2: left shift by 2 positions

# Symmetric Encryption: DES

- A simplified example for DES
  - Encryption: plaintext 0 1 1 1 0 0 1 0
  - K1: 1 0 1 0 0 1 0 0    K2: 0 1 0 0 0 0 1 1

IP (initial permutation)

Input : 1 2 3 4 5 6 7 8

Output: 2 6 3 1 4 8 5 7

IP-1: inverse of IP, such that X = IP-1(IP(X))

P4 (permutate)

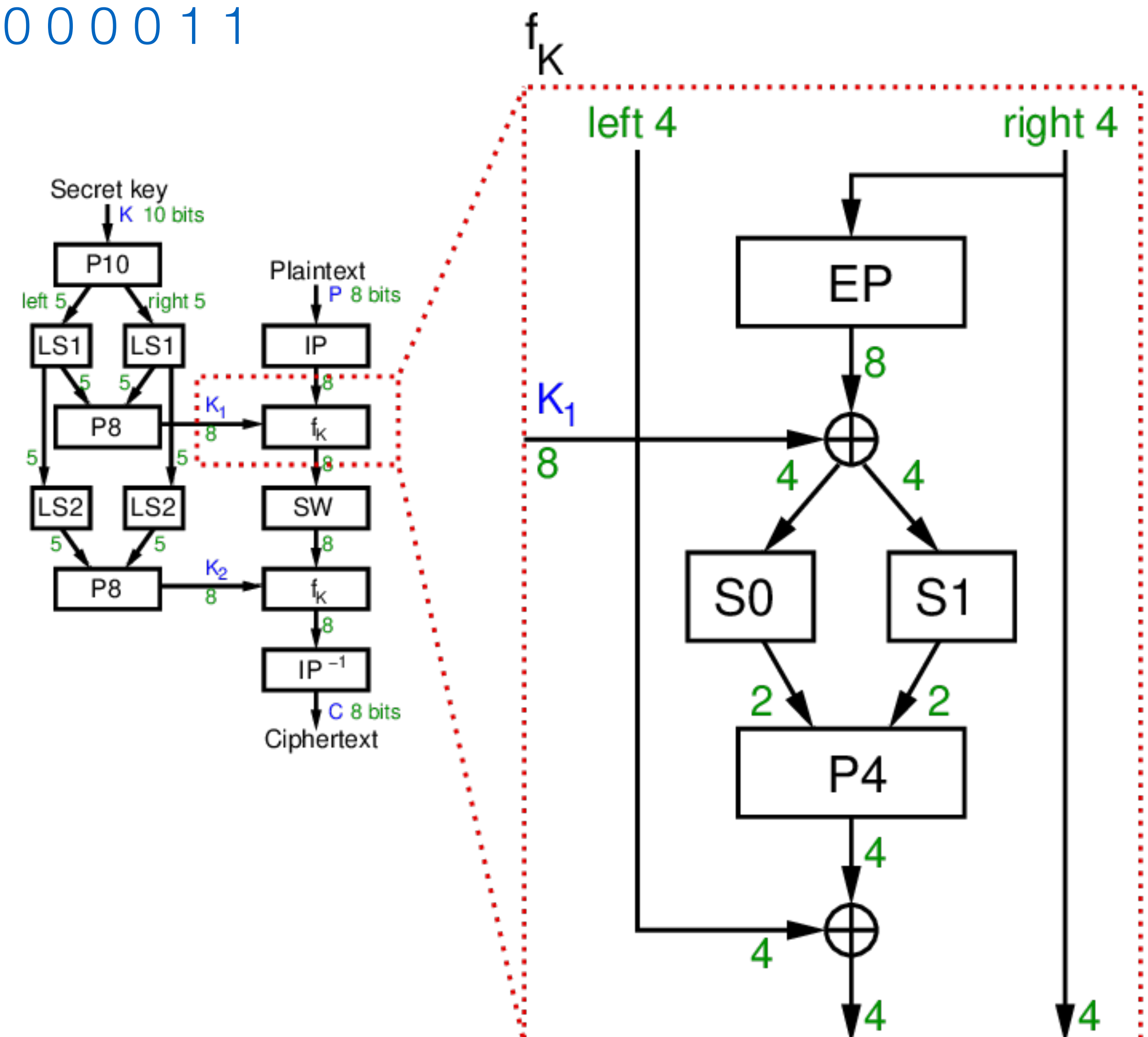
Input : 1 2 3 4

Output: 2 4 3 1

EP (expand and permutate)

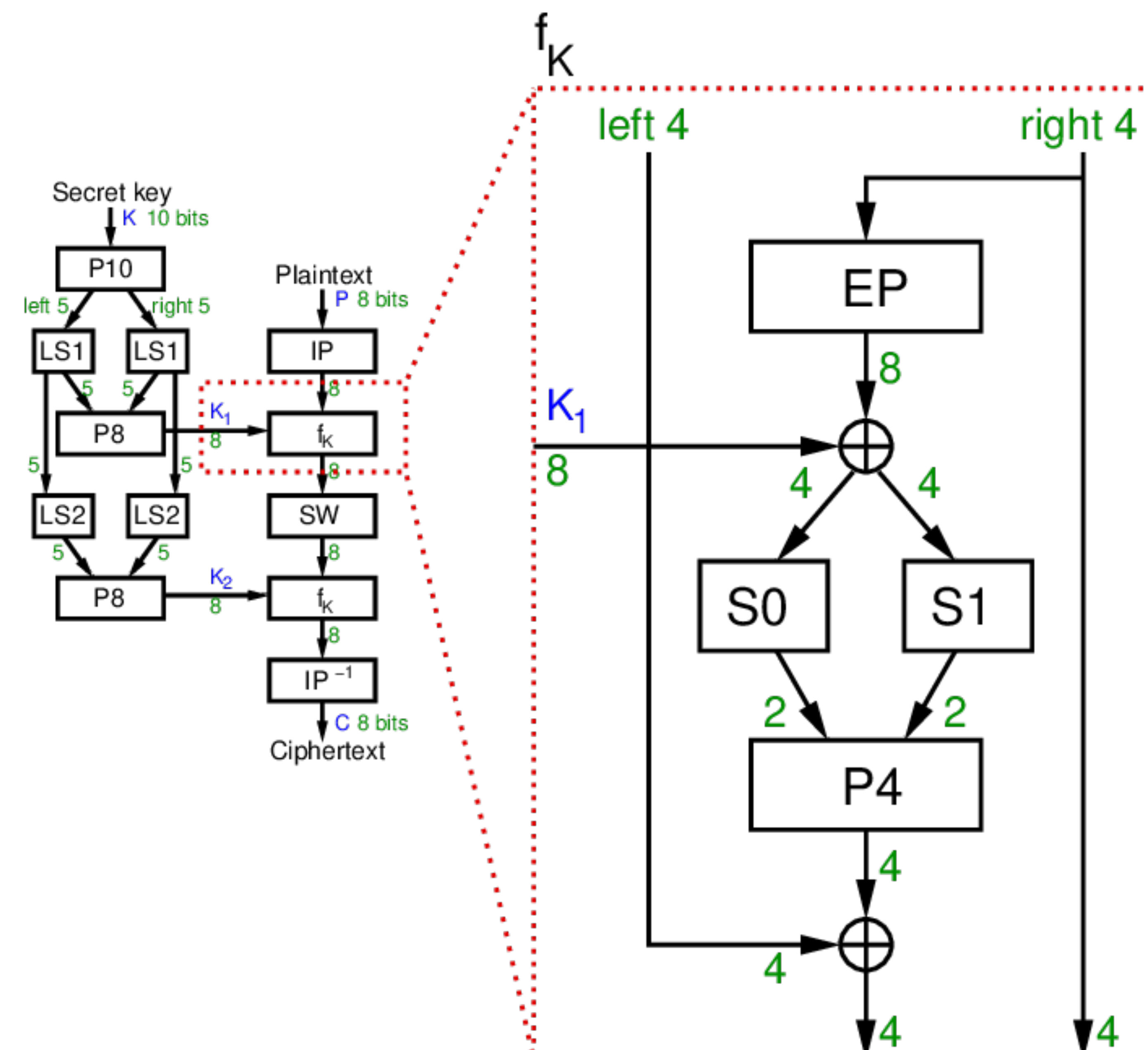
Input : 1 2 3 4

Output: 4 1 2 3 2 3 4 1



# Symmetric Encryption: DES

- A simplified example for DES
  - Encryption: plaintext 0 1 1 1 0 0 1 0
  - IP: 1 0 1 0 1 0 0 1
  - EP: 1 1 0 0 0 0 1 1
  - XOR K1 (1 0 1 0 0 1 0 0):
    - 0 1 1 0 0 1 1 1




IP (initial permutation)

```
Input : 1 2 3 4 5 6 7 8
Output: 2 6 3 1 4 8 5 7
```

IP-1: inverse of IP, such that X = IP-1(IP(X))

EP (expand and permutate)

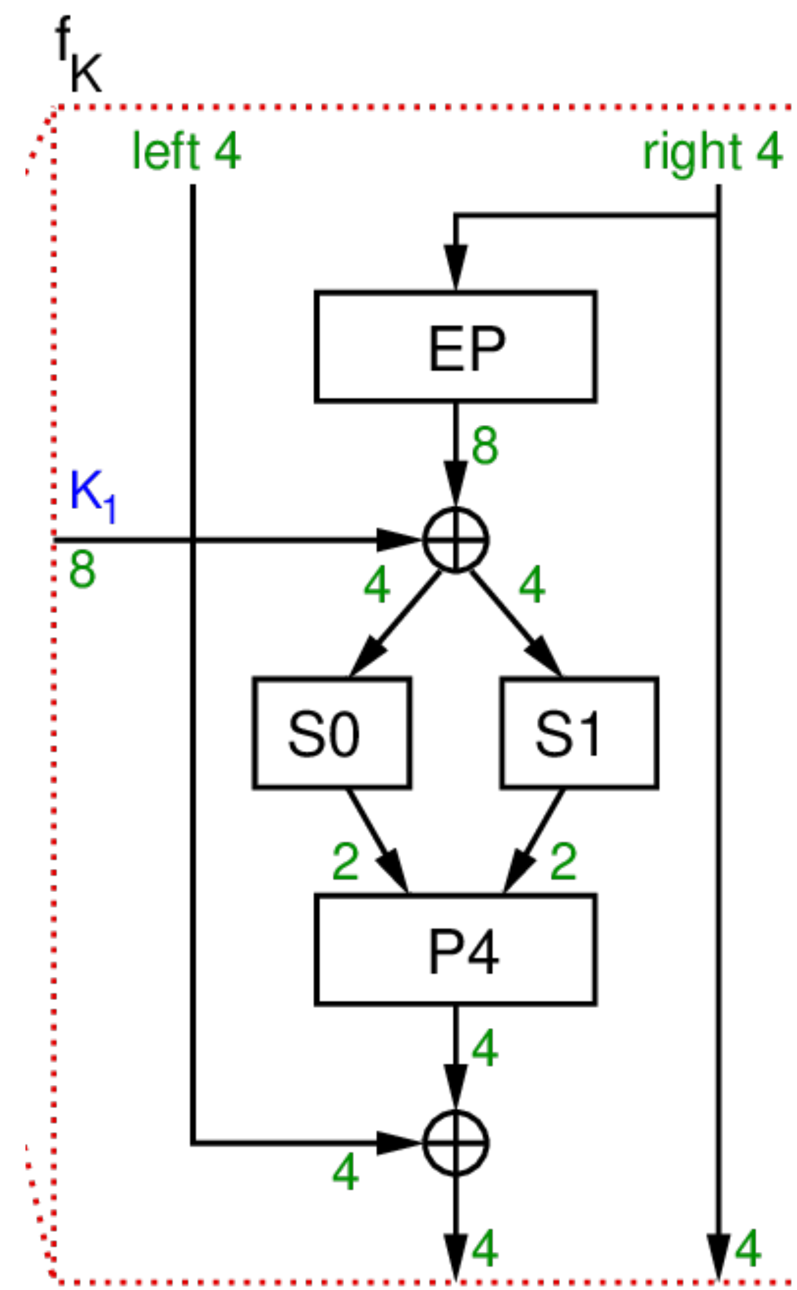
```
Input : 1 2 3 4
Output: 4 1 2 3 2 3 4 1
```

**XOR truth table**

| Input | | Output |
|---|---|---|
| A | B | |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Symmetric Encryption: DES

- A simplified example for DES
  - Output XOR: 0 1 1  0  0 1 1 1
  - Input S0: 0 1 1  0
  - S0: row 00  and  column 11
  - Output S0: 10



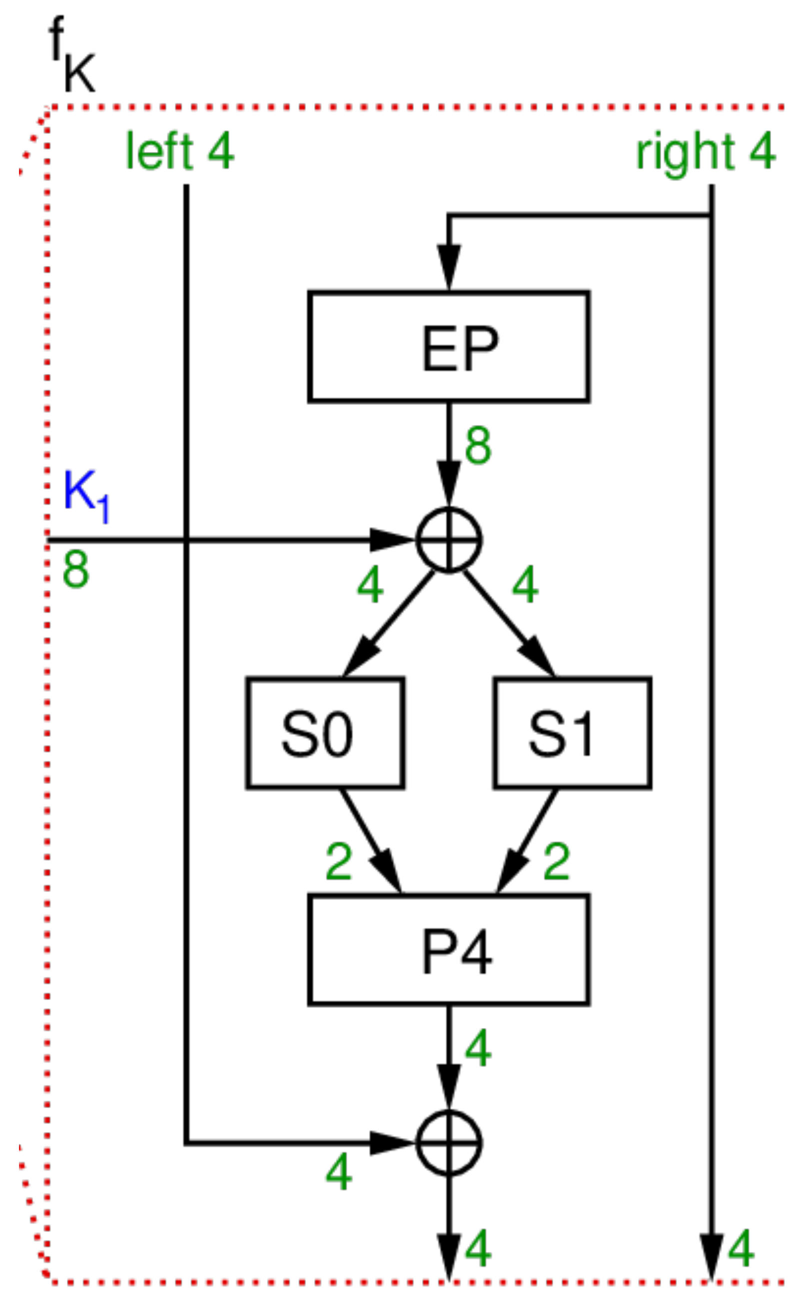

## S-DES S-Boxes

- S-DES (and DES) perform substitutions using S-Boxes
- S-Box considered as a matrix: input used to select row/column; selected element is output
- 4-bit input: $bit_1, bit_2, bit_3, bit_4$
- $bit_1 bit_4$ specifies row (0, 1, 2 or 3 in decimal)
- $bit_2 bit_3$ specifies column
- 2-bit output

$$S0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \quad S1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

# Symmetric Encryption: DES

- A simplified example for DES
  - Output XOR: 0 1 1 0 0 1 1 1
  - Input S1: 0 1 1 1
  - <u>Exercise: Output S1?</u>



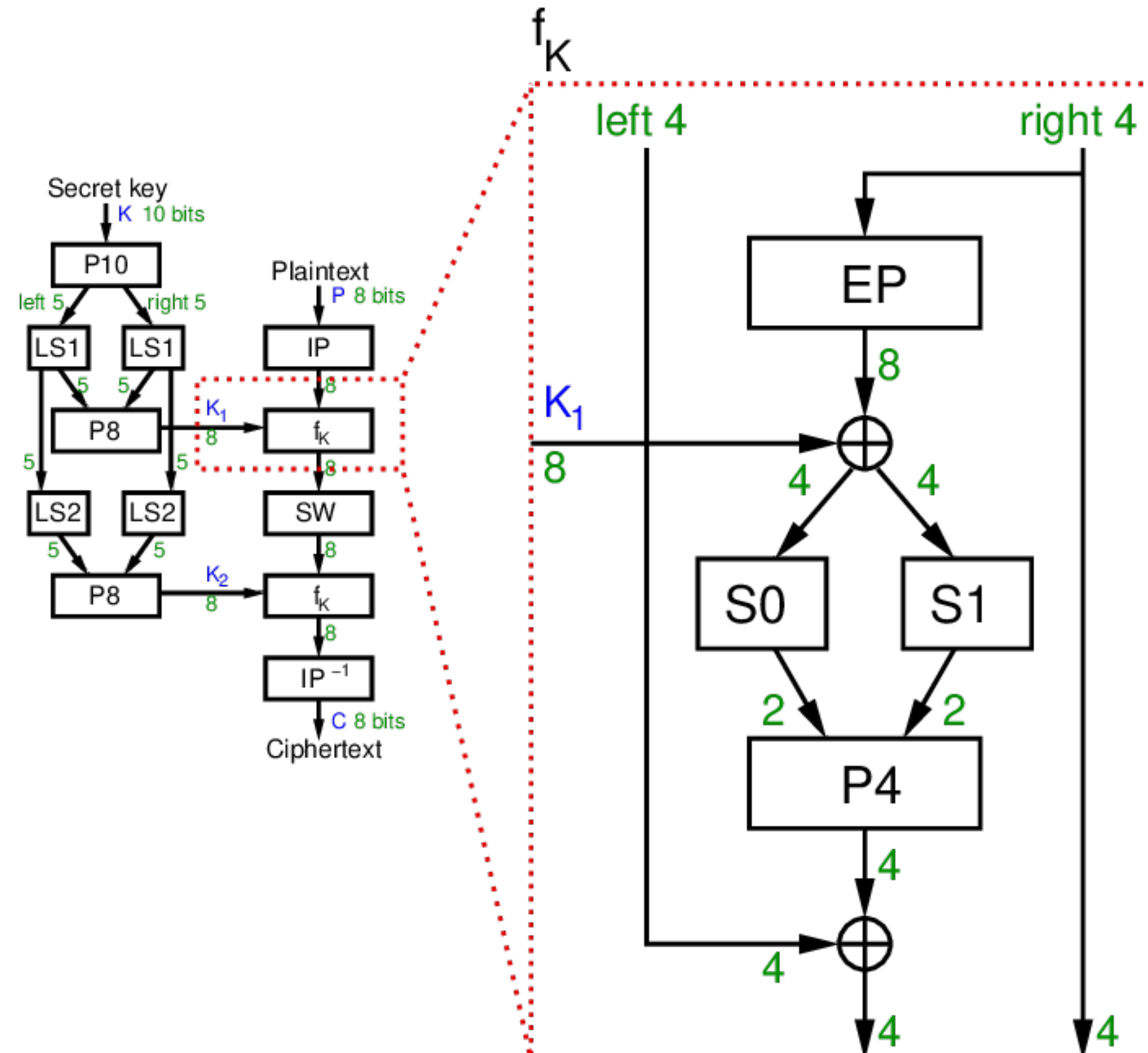

## S-DES S-Boxes

- S-DES (and DES) perform substitutions using S-Boxes
- S-Box considered as a matrix: input used to select row/column; selected element is output
- 4-bit input: $bit_1, bit_2, bit_3, bit_4$
- $bit_1 bit_4$ specifies row (0, 1, 2 or 3 in decimal)
- $bit_2 bit_3$ specifies column
- 2-bit output

$$S0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \quad S1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

# Symmetric Encryption: DES

- A simplified example for DES
  - S0: 10   S1: 11
  - P4: 0 1 1 1
  - IP: 1 0 1 0 1 0 0 1
  - P4 XOR (1 0 1 0): 1 1 0 1
  - left: 1101 right: 1001




P4 (permutate)

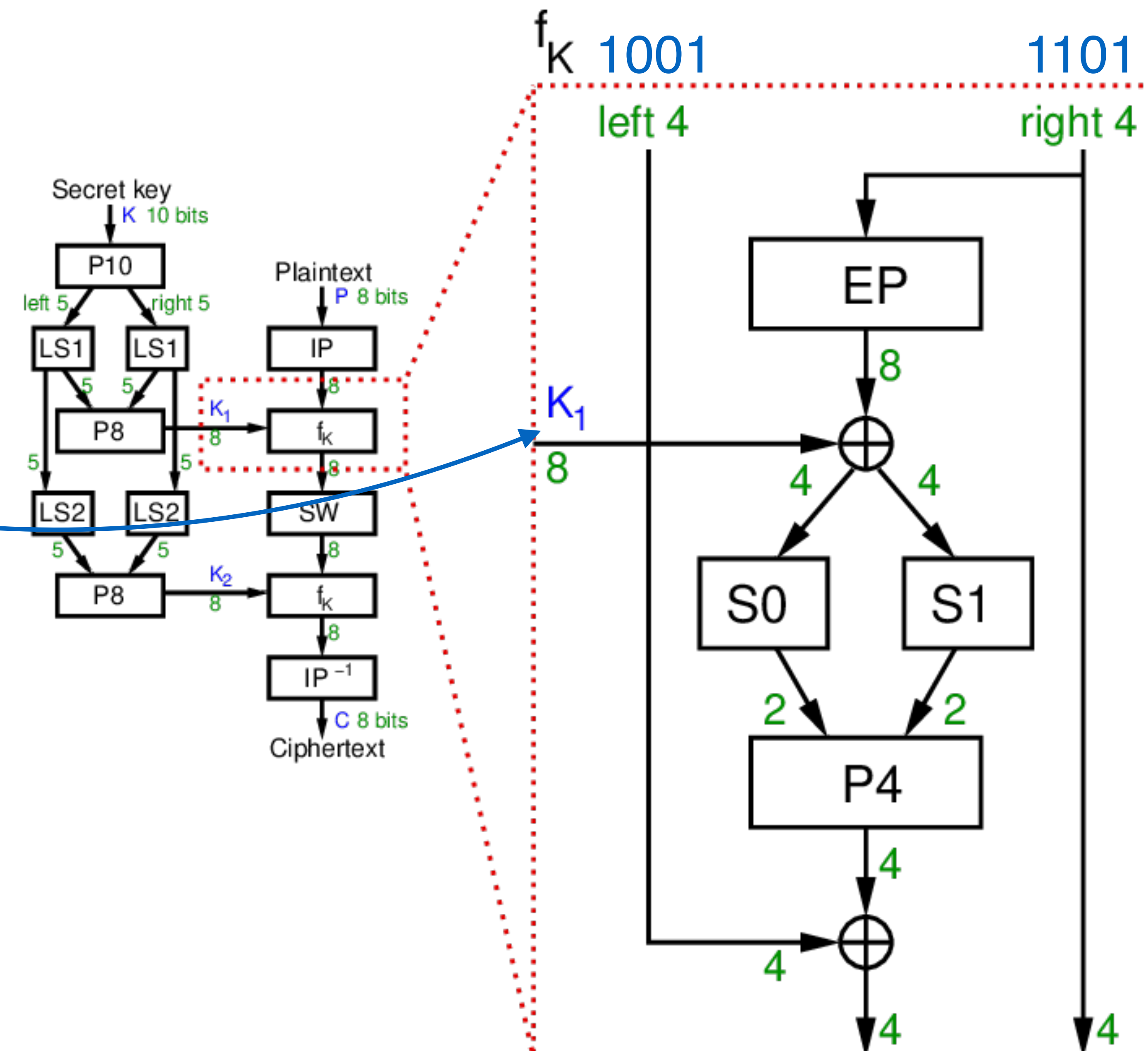
```
Input : 1 2 3 4
Output: 2 4 3 1
```

**XOR truth table**

| Input | | Output |
|---|---|---|
| **A** | **B** | |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

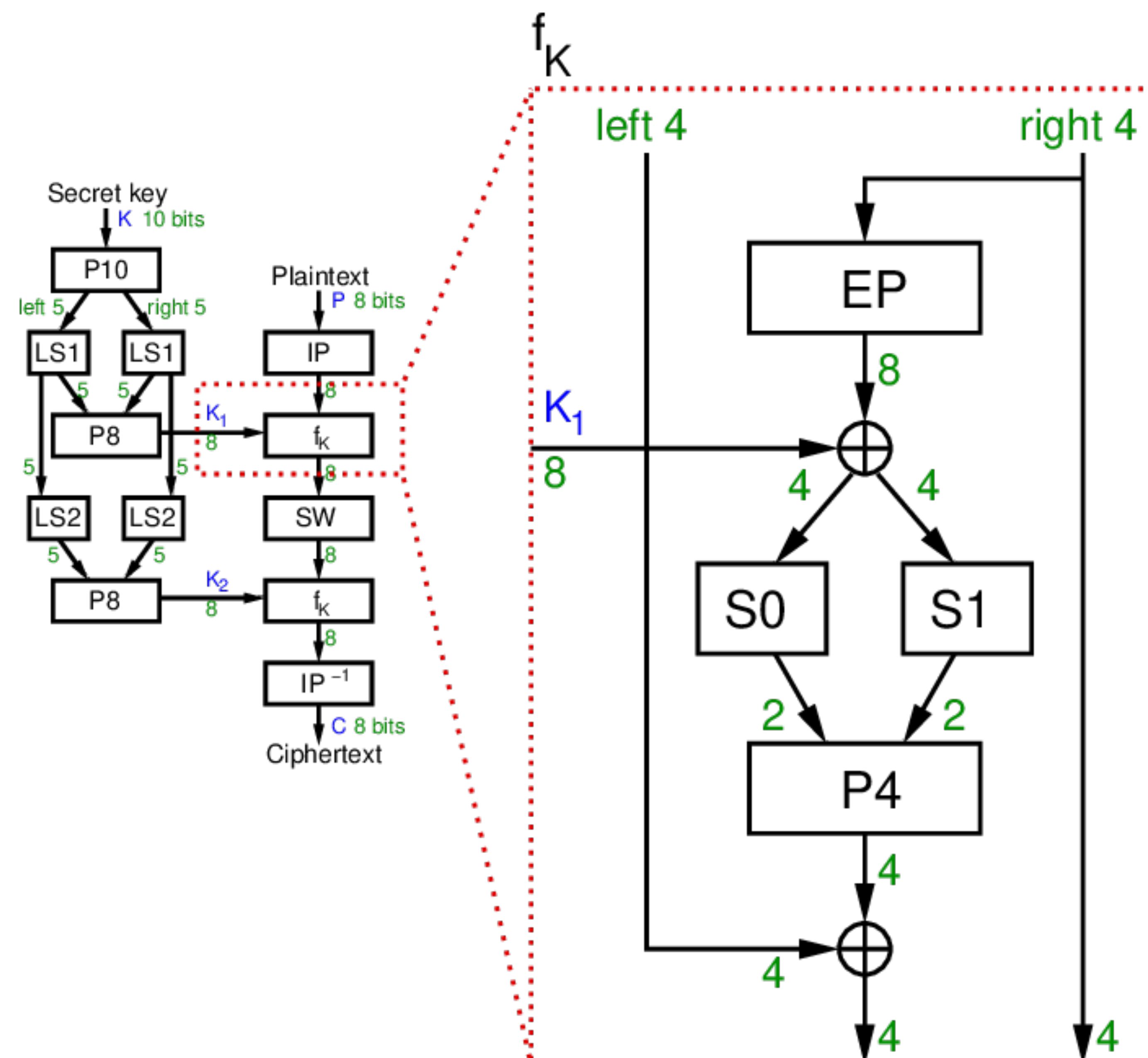
# Symmetric Encryption: DES

- A simplified example for DES
  - left: 1101 right: 1001
  - Swap: 1001 1101
  - <u>Exercise: Output of round 2?</u>
    - <u>K2: 0 1 0 0 0 0 1 1</u>



# Symmetric Encryption: DES

- A simplified example for DES
  - left: 1101 right: 1001
  - Swap: 1001 1101
  - Output f2: 1 1 1 0 1 1 0 1
  - Ciphertext: 0 1 1 1 0 1 1 1



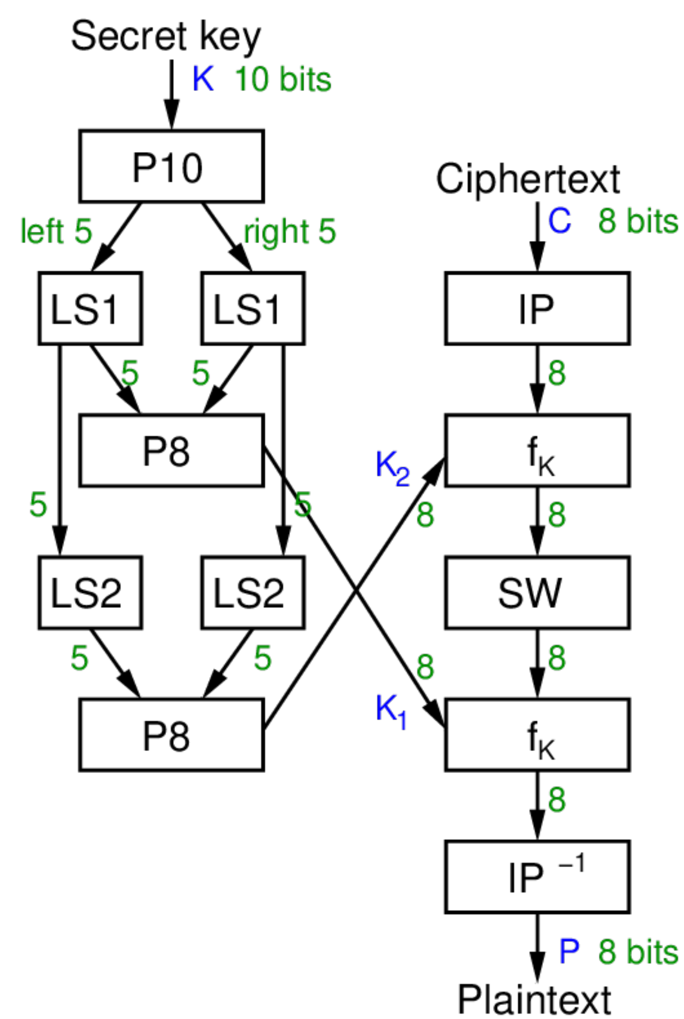

IP (initial permutation)

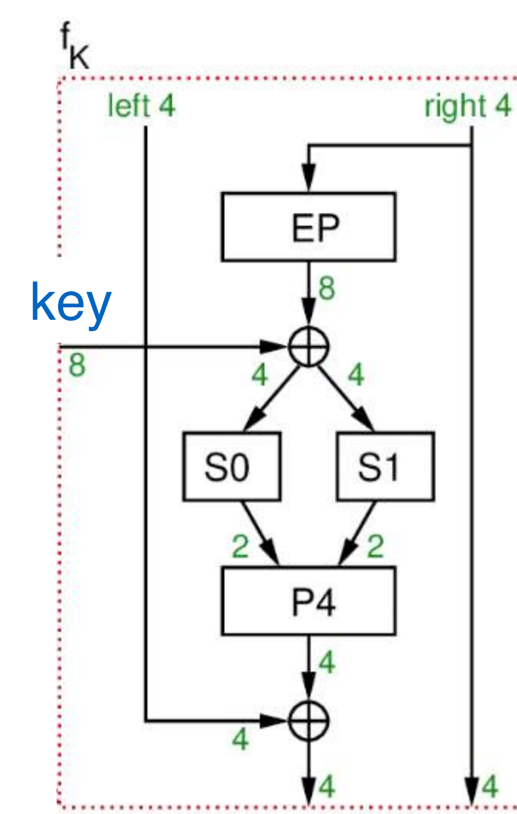
```
Input : 1 2 3 4 5 6 7 8
Output: 2 6 3 1 4 8 5 7
```

IP-1: inverse of IP, such that X = IP-1(IP(X))

# Symmetric Encryption: DES

- A simplified example for DES
  - Decryption ciphertext: 0 1 1 1 0 1 1 1
  - K1: 10100100   K2: 01000011
  - <u>Exercise: Can you recover plaintext 0 1 1 1 0 0 1 0?</u>




EP (expand and permutate)

```
Input : 1 2 3 4

Output: 4 1 2 3 2 3 4 1
```

IP (initial permutation)

```
Input : 1 2 3 4 5 6 7 8

Output: 2 6 3 1 4 8 5 7
```

IP-1: inverse of IP, such that X = IP-1(IP(X))

## S-DES S-Boxes

- S-DES (and DES) perform substitutions using S-Boxes
- S-Box considered as a matrix: input used to select row/column; selected element is output
- 4-bit input: $bit_1, bit_2, bit_3, bit_4$
- $bit_1 bit_4$ specifies row (0, 1, 2 or 3 in decimal)
- $bit_2 bit_3$ specifies column
- 2-bit output

$$S0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \quad S1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

P4 (permutate)

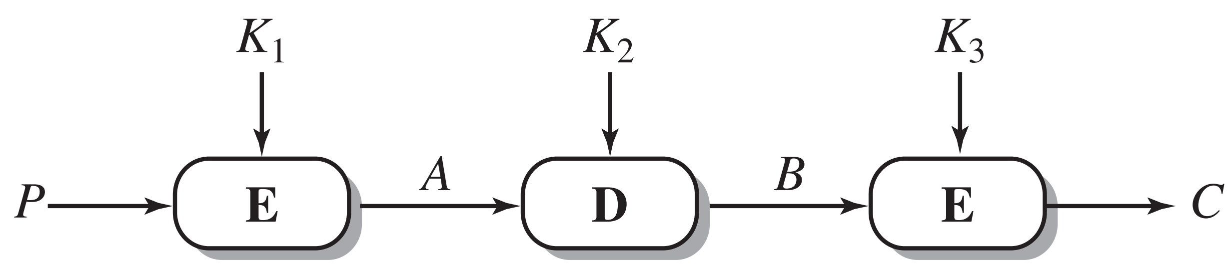
```
Input : 1 2 3 4

Output: 2 4 3 1
```

**XOR truth table**

| Input | | Output |
|---|---|---|
| **A** | **B** | |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Symmetric Encryption: Triple DES

- 3 DES uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence

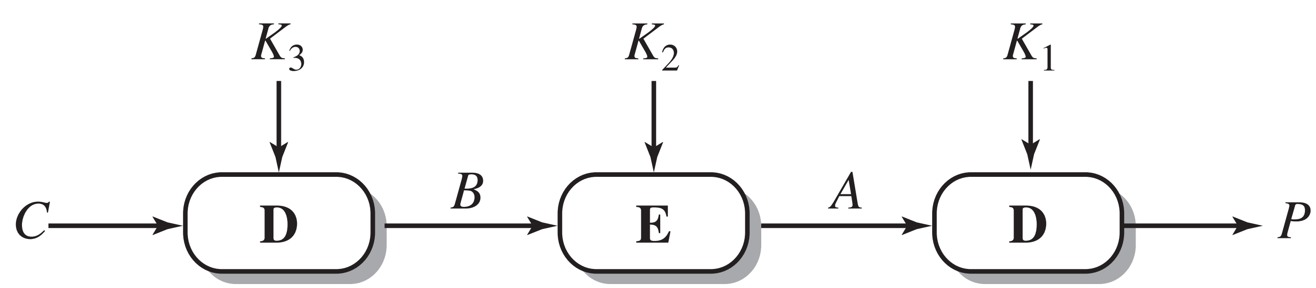



(a) Encryption

$$C = \mathrm{E}(K_3, \mathrm{D}(K_2, \mathrm{E}(K_1, p)))$$

$C$ = ciphertext
$P$ = plaintext
$\mathrm{E}[K, X]$ = encryption of $X$ using key $K$
$\mathrm{D}[K, Y]$ = decryption of $Y$ using key $K$

- Decryption is simply the same operation with the key reversed

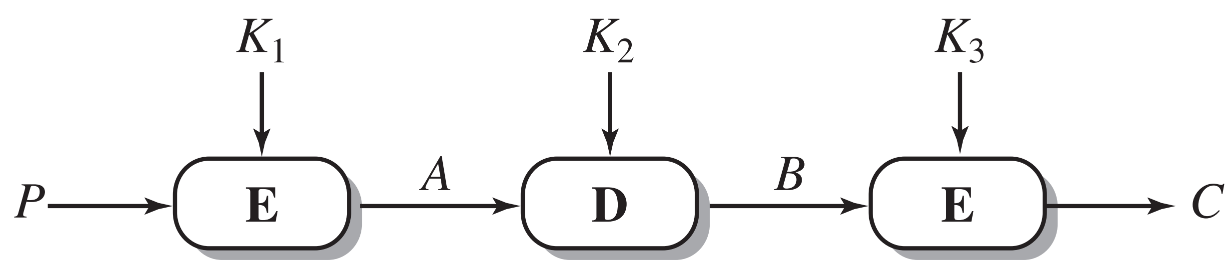



(b) Decryption

$$P = \mathrm{D}(K_1, \mathrm{E}(K_2, \mathrm{D}(K_3, \mathrm{C})))$$

# Symmetric Encryption: Triple DES

- 3 DES uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence




(a) Encryption

$$C = \mathrm{E}(K_3, \mathrm{D}(K_2, \mathrm{E}(K_1, p)))$$

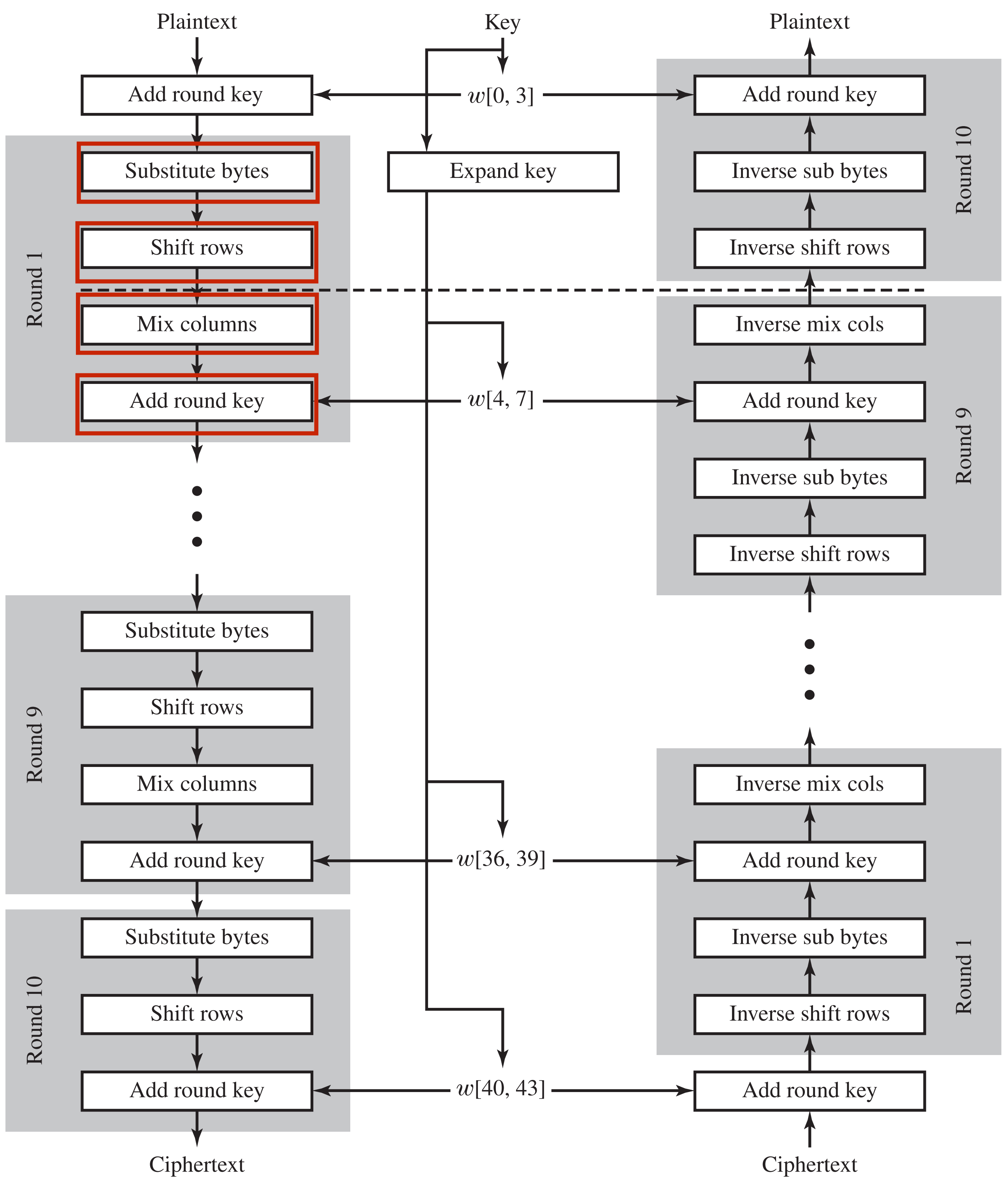
$C$ = ciphertext
$P$ = plaintext
$\mathrm{E}[K, X]$ = encryption of $X$ using key $K$
$\mathrm{D}[K, Y]$ = decryption of $Y$ using key $K$

- <u>Can triple DES reduce to single DES?</u>

# Symmetric Encryption: AES

- Advanced encryption standard (AES) [more secure and efficient]
  - Block length 128 bits
  - Key length can be 128, 192, or 256 bits
  - Not a Feistel structure, processes the entire data block in parallel during each round using substitutions and permutation. [Feistel structure: half of the data block is used to modify the other half, and then the halves are swapped.]
  - Key operations: substitute bytes, shift rows, mix columns, and add round key.
  - All operations are reversible: for substitute bytes, shift rows, mix columns, an inverse function is used in the decryption; for add round key, the inverse is achieved by XOR the same round key to the block — $\mathbf{A} \bigoplus \mathbf{A} \bigoplus \mathbf{B} = \mathbf{B}$.
  - It is easy to verify that decryption does recover the plaintext by reversible operations.

# Symmetric Encryption: AES




(a) Encryption (b) Decryption

# Symmetric Encryption: AES

- Substitute Bytes Transformation
  - AES defines a 16·16 matrix of byte values, called an S-box

| | | y | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| x | 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| | 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| | 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| | 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| | 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| | 5 | 53 | D1 | 00 | ED | 20 | FC | BI | 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 |

  - Leftmost 4 bits — row index, rightmost 4 bits — column index, check S-box for 8-bit output value

Example:

| | | | |
|---|---|---|---|
| EA | 04 | 65 | 85 |
| 83 | 45 | 5D | 96 |
| 5C | 33 | 98 | B0 |
| F0 | 2D | AD | C5 |

$\longrightarrow$

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

# Symmetric Encryption: AES

- Inverse Substitute Bytes Transformation
  - Inverse S-box

| | | y | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| x | 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 | FA |
| | 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 |

-A low correlation between input bits and output bits

-The output cannot be described as a simple mathematical function of the input.

  - Leftmost 4 bits — row index, rightmost 4 bits — column index, check S-box for 8-bit output value

Example:

| | | | |
|---|---|---|---|
| EA | 04 | 65 | 85 |
| 83 | 45 | 5D | 96 |
| 5C | 33 | 98 | B0 |
| F0 | 2D | AD | C5 |

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

# Symmetric Encryption: AES

- Shift Row Transformation
  - First row of state is not altered
  - Second row: 1-byte circular left shift
  - Third row: 2-byte circular left shift
  - ….
- Inverse Shift Row Transformation
  - Circular shifts in the opposite direction

Example:

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

$\longrightarrow$

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

# Symmetric Encryption: AES

- Mix Column Transformation

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

Predefine Matrix  State Array  New State Array

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

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

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

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

Example:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} * \begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$$

Predefine Matrix

# Symmetric Encryption: AES

- Mix Column Transformation

Example:

$$
\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}
*
\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}
\longrightarrow
\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}
$$

Predefine Matrix

$$01 * x = x$$

$02 * x$ x highest digit is 0 —> binary multiplication

x highest digit is 1 —> shift left by 1, then XOR with 1B (0001 1011)

$$03 * x = (02 \oplus 01) * x = (02 * x) \oplus x$$

87=1000 0111 02*87: 1) shift left by 1: 0000 1110 2) XOR with 1B: 0001 0101

6E=0110 1110 02*6E=1101 1100 03*6E=(02*6E) XOR 6E= 1011 0010

46=0100 0110 01*46=0100 0110

A6=1010 0110 01*A6=1011 0110

XOR

0101 0111 —> 47

# Symmetric Encryption: AES

- Add Round Key Transformation
  - State XOR Round Key (element-wise)

- Inverse Add Round Key Transformation
  - Identical to the forward, i.e., State XOR Round Key. XOR operation is its own inverse.

$$0100\ 0000 \oplus 0001\ 1001 = 0101\ 1001$$

Example:

| | | | |
|---|---|---|---|
| 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 |
| ED | A5 | A6 | BC |

Key

$=$

| | | | |
|---|---|---|---|
| EB | 59 | 8B | 1B |
| 40 | 2E | A1 | C3 |
| F2 | 38 | 13 | 42 |
| 1E | 84 | E7 | D2 |

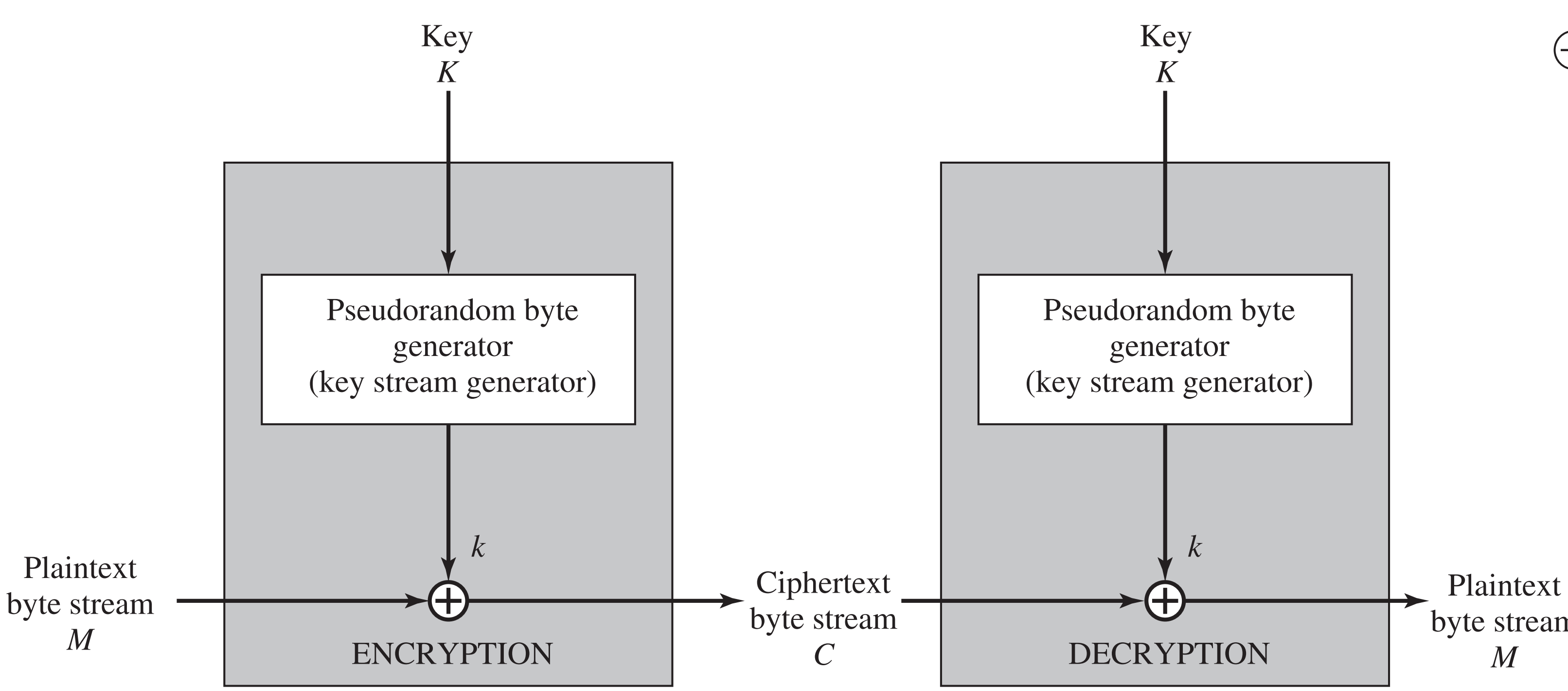
# Symmetric Encryption: Stream Cipher

- Stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.
- A key is input to a pseudorandom bit generator that produces a stream of numbers — key stream
- A key stream XOR plaintext stream (bitwise)

```
  11001100    plaintext
⊕ 01101100    key stream
  10100000    ciphertext
```

- Decryption requires the use of the same pseudorandom sequence

```
  10100000    ciphertext
⊕ 01101100    key stream
  11001100    plaintext
```



# Symmetric Encryption: Stream Cipher

- As secure as block cipher of comparable key length
- Faster and use far less code
- Reuse key incurs security issue (block cipher can reuse keys)
  - Example: Two plaintexts are encrypted with the same key using a stream cipher, then XOR of two ciphertexts is the XOR of the original plaintexts

# Symmetric Encryption: RC4

- Stream cipher
- A variable length key 1-256 bytes
- Use key to initialize state vector S (permutation)
- Once S vector is initialized, the input key is no longer used.
- Key stream is generated by S (cycling, swapping… ) and take XOR with plaintext for encryption

# Cipher Block Modes of Operation

- DES/3DES the block length is 64 bits
- How to tackle a longer plaintext?

# Cipher Block Modes of Operation

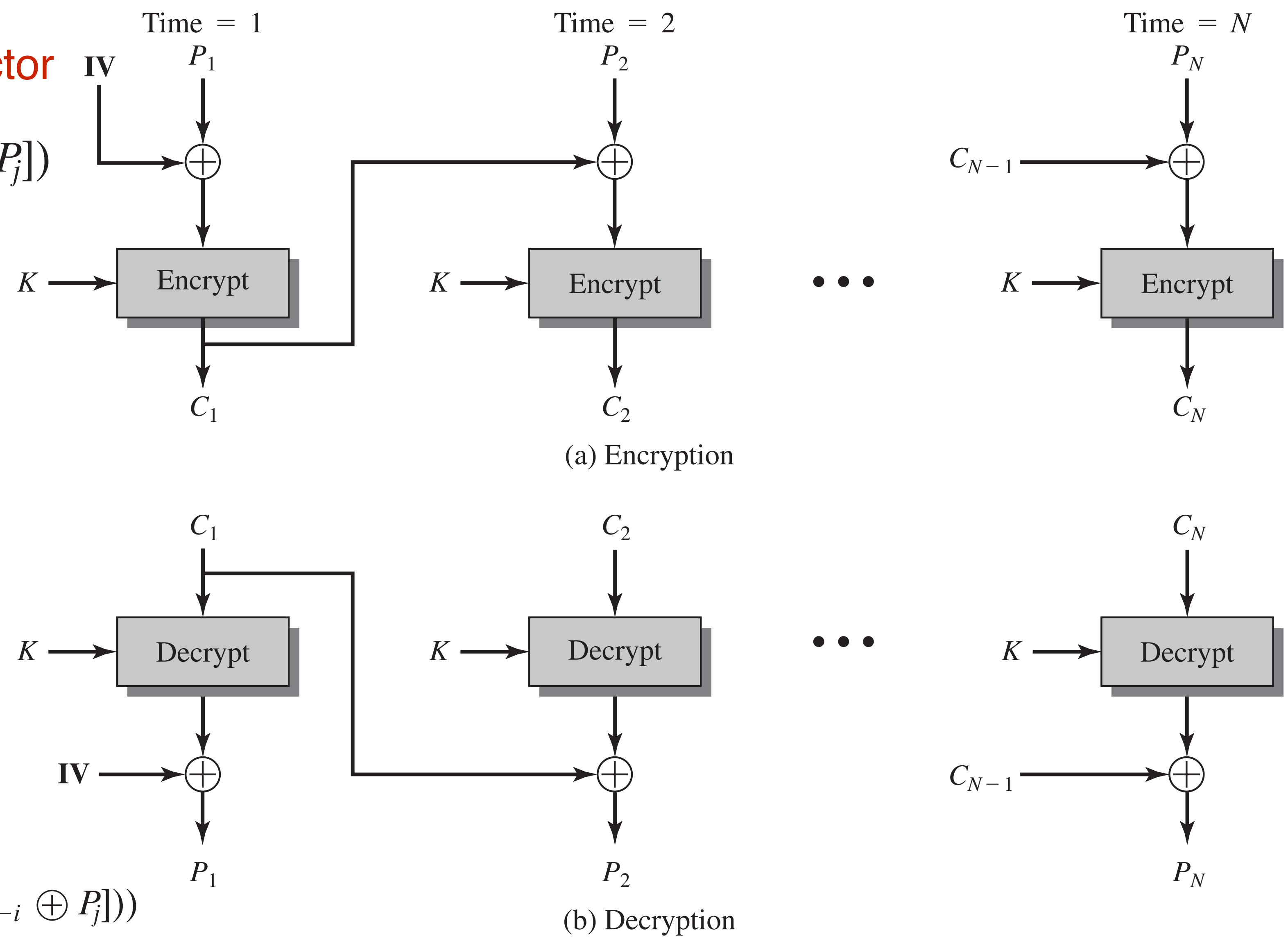
- Electronic Code book
  - Description: Each block of 64 plaintext bits is encoded independently using the same key.
  - Typical Application: Secure transmission of single values (e.g., an encryption key)

# Cipher Block Modes of Operation

- Cipher Block Chaining (CBC)

Initialization vector

$$C_j = \mathrm{E}(K, [C_{j-1} \oplus P_j])$$



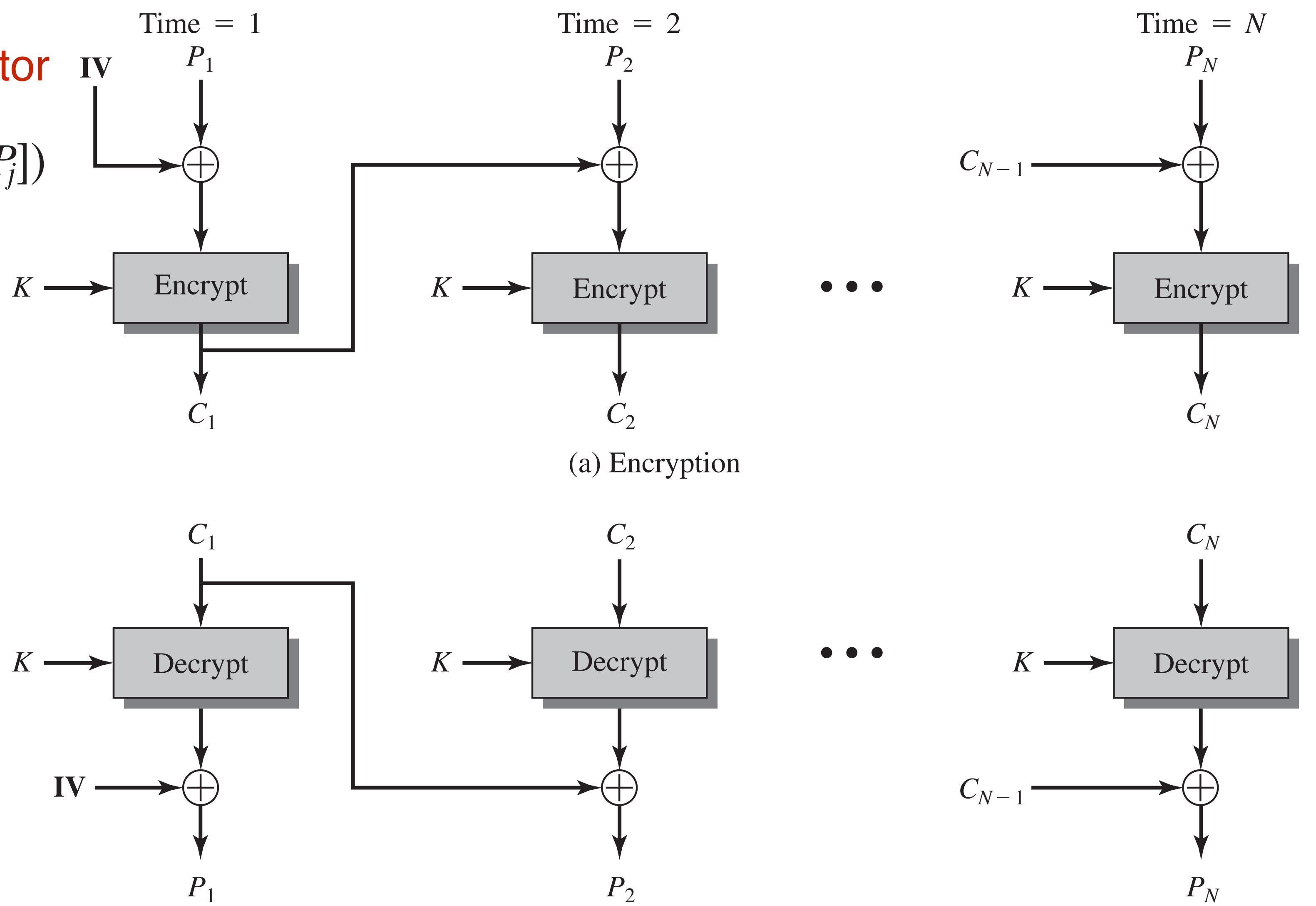

$$\mathrm{D}(K, C_j) = \mathrm{D}(K, \mathrm{E}(K, [C_{j-i} \oplus P_j]))$$
$$\mathrm{D}(K, C_j) = C_{j-1} \oplus P_j$$
$$C_{j-1} \oplus \mathrm{D}(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j$$

# Cipher Block Modes of Operation

- Cipher Block Chaining (CBC)

Initialization vector

$$C_j = \mathrm{E}(K, [C_{j-1} \oplus P_j])$$




(a) Encryption

(b) Decryption

$$\mathrm{D}(K, C_j) = \mathrm{D}(K, \mathrm{E}(K, [C_{j-i} \oplus P_j]))$$

$$\mathrm{D}(K, C_j) = C_{j-1} \oplus P_j$$

$$C_{j-1} \oplus \mathrm{D}(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j$$

# Cipher Block Modes of Operation

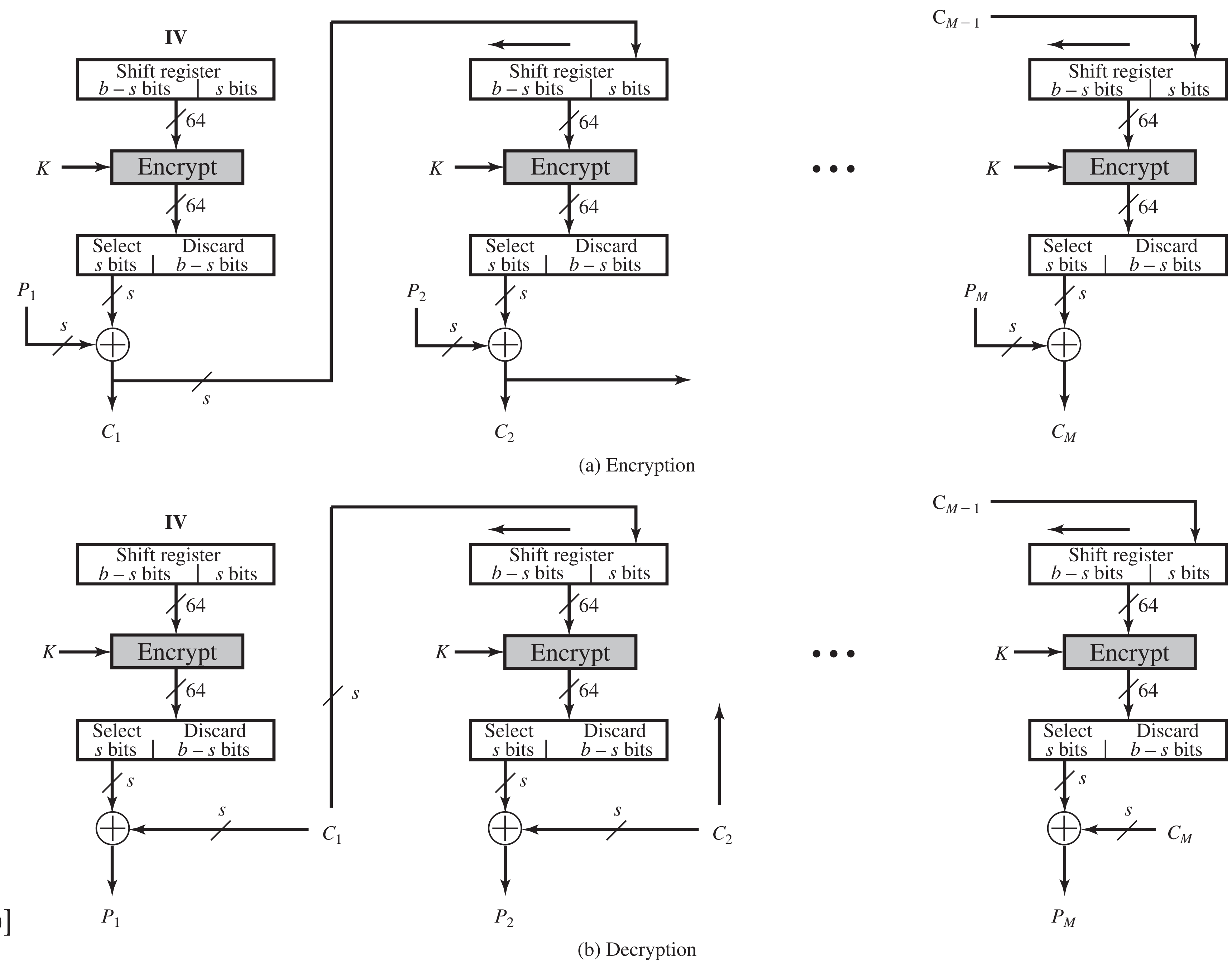
- Cipher Block Chaining (CBC)
  - Initialization vector (IV) must be known to both the sender and receiver
  - IV should be protected as key, e.g., adversary change bits in IV, then the plaintext in block 1 can be changed.

  - Typical application: General-purpose block-oriented transmission; authentication

# Cipher Block Modes of Operation

- Cipher Feedback (CFB)
  - convert block cipher into a stream cipher

$$C_1 = P_1 \oplus S_s[E(K, IV)]$$



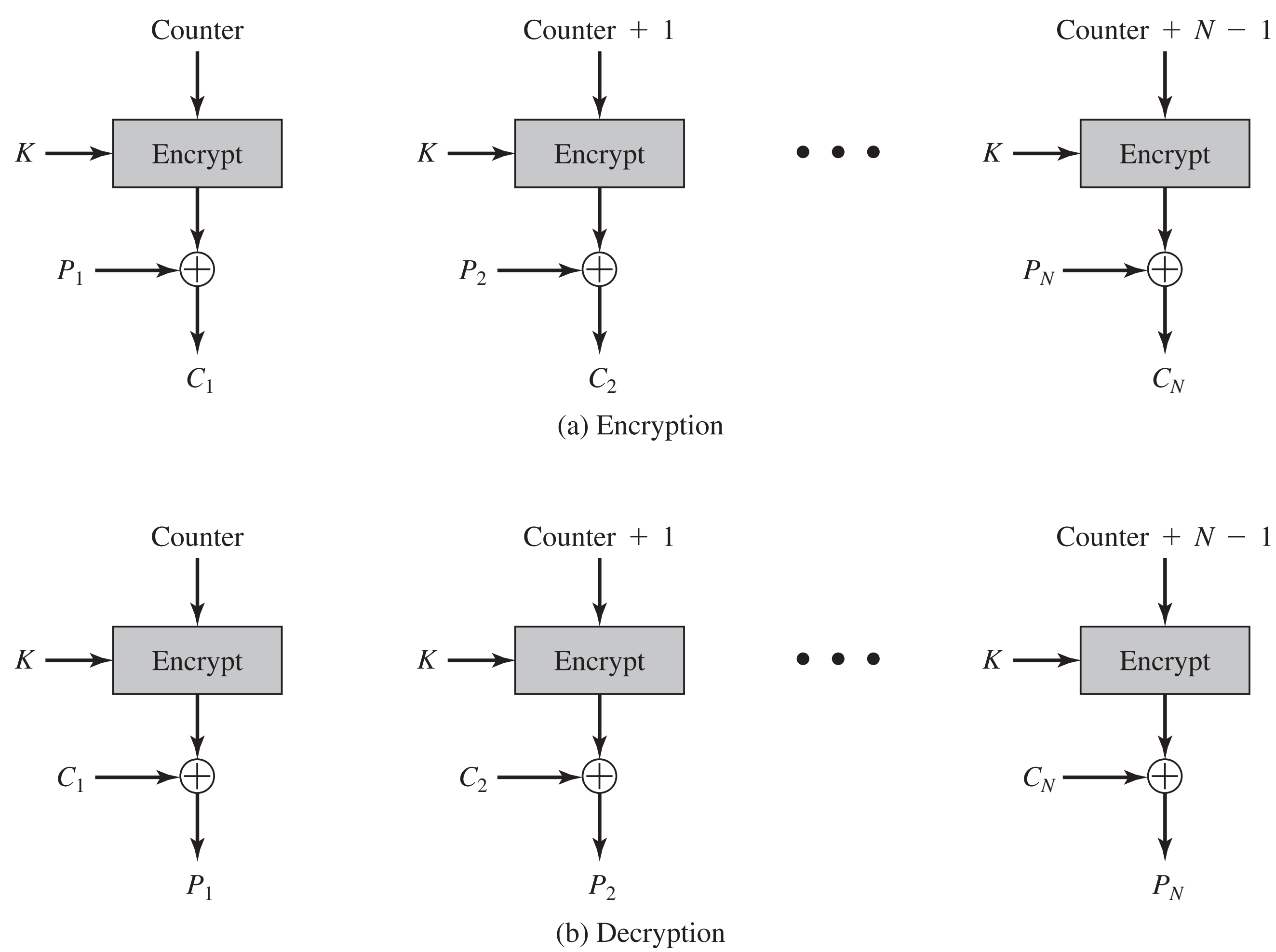

$$P_1 = C_1 \oplus S_s[E(K, IV)]$$

# Cipher Block Modes of Operation

- Cipher Feedback (CFB)
  - Decryption also use the encryption function
  - Typical application: General-purpose stream-oriented transmission; authentication

# Cipher Block Modes of Operation

- Counter Mode
  - Counter size = plaintext block size
  - Counter is initialized to some value and then incremented by 1 for each subsequent block



# Cipher Block Modes of Operation

- Counter (CTR)
  - No chain, multiple blocks process in parallel
  - Counter at i block cannot be computed until i-1 prior blocks are computed
  - Only need encryption algorithm

  - Typical application: General-purpose block-oriented transmission; Useful for high-speed requirements

# Summary

- Cryptography
  - Context, Ingredients, Classification, Attacks
- Symmetric Encryption: Block Cipher
  - DES/Triple DES (Feistel Cipher Structure)
  - AES
- Symmetric Encryption: Stream Cipher
  - RC4
- Cipher Block Modes of Operation
  - Electronic Code book (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Counter

# Thank You

# Quiz Time

- 15 minutes