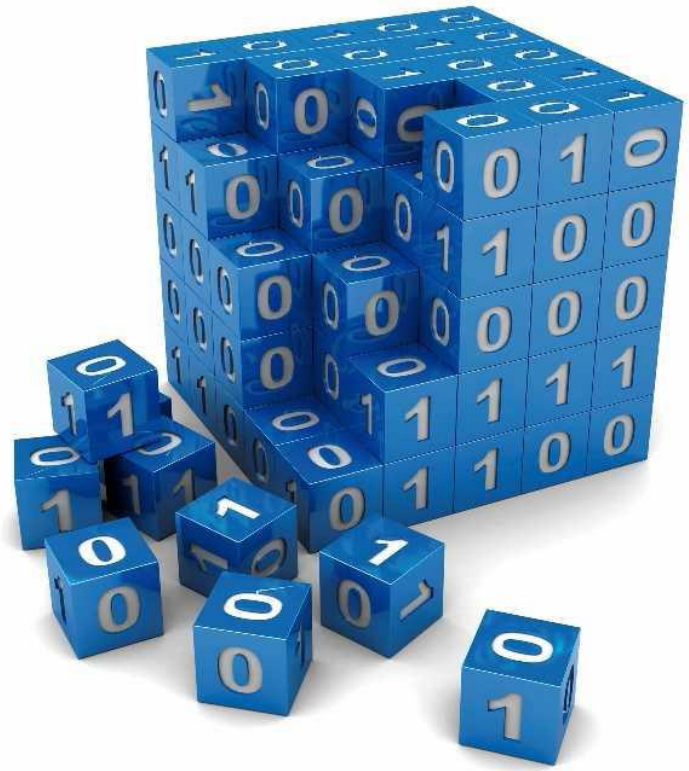
III. MODERN CRYPTOGRAPHY

- Key ?
- Encryption Modes
- Private Key Encryption
- Public Key Encryption
- Hash Function
- Digital Fingerprint
- Cryptanalysis
- PKI (Public Key Infrastructure)



MODERN ENCRYPTION

- Current algorithms manipulate bits instead of characters
- The encoding and decoding of texts are based on the concept of keys (a bit stream).

Key ?

- A key is a value represented by bits and used with a cryptographic algorithm to produce a specific encrypted text.

- The security level of an algorithm depends on the key. The longer the key, the more secure the encryption.

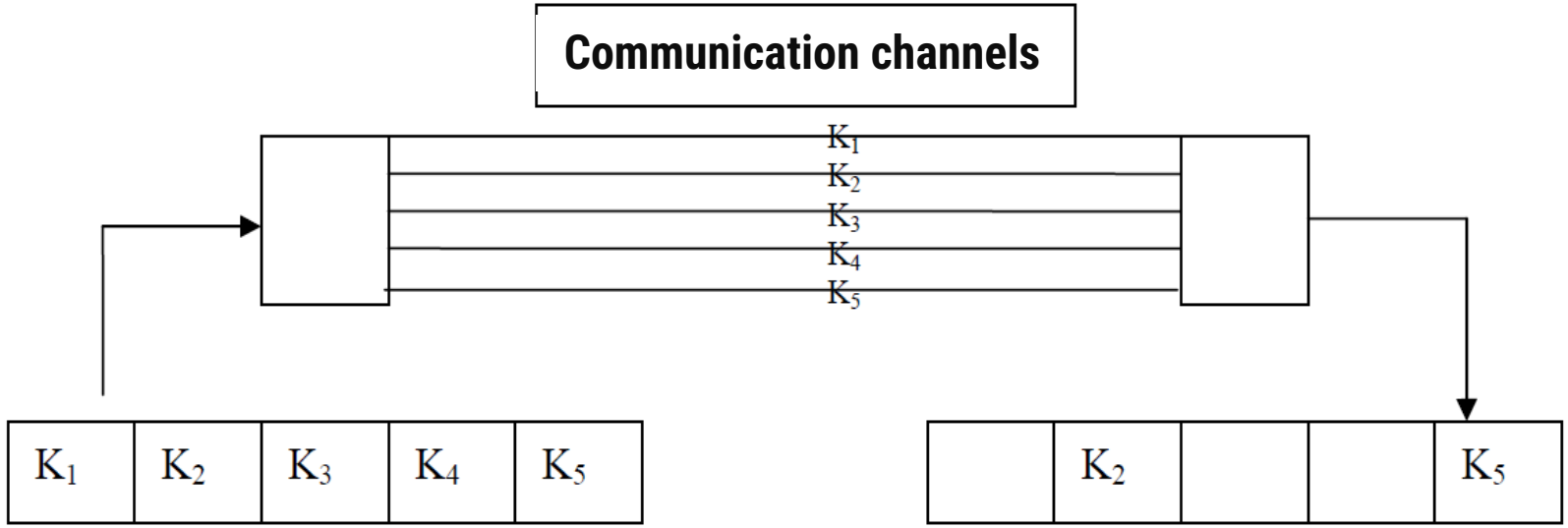
Key management

Key management is the process of securely storing and distributing cryptographic keys to authorized recipients:

- **Transfer**
- **Usage**
- **Storage**
- **Duplication**
- **Longevity**
- **Procurement**

1. Key transfer

- Using a single communication channel
  - **Disadvantage:** listening to the channel
  - **Solution:** Split the key into pieces and send each piece through a different channel: telephone, email...



  - **Disadvantage:** if the key changes every day.
  - **Solution:** send all the keys (for a month) at once:
    - Securely store this list of keys (using a master key).

2. Key usage

- Using software (Encryption tool)

- **Disadvantage:** The encryption application does not run continuously (multitasking operating system)
  - Encryption interruption (backup of the key and encryption program)
    - ➢ Risk of key retrieval by an attacker

- **Solution:** Certificate, Access control,..

3. Key storage

- The key is memorized by the user:

  - Manually enter the key (e.g., 64 bits)

  or

  - Enter the key as a string and use a key hashing technique:
    a) Transform the string into a sequence of bits.
    b) Split the key into two parts: one part memorized by the user and the other part by the system

4. Key duplication

- A responsible individual must know and back up all keys (e.g., those of the employees):
  - **Disadvantage:** Risk of using keys for personal purposes (by the responsible individual)
  - **Solution:** Secret sharing

5. Key longevity

- No key should be used indefinitely

6. Key procurement

- Different ways to obtain someone's public key:
  - Directly from the user.
  - From a centralized database (Certification Authority or trusted authority).
  - From their private database.

7. Key management can be:

- **Centralized** (public key certificate): someone's public key is signed by a person (trusted authority)

- **Distributed:** a group of people sign someone's public key.
  - An example of a system using this method is PGP.

Block Cipher Modes

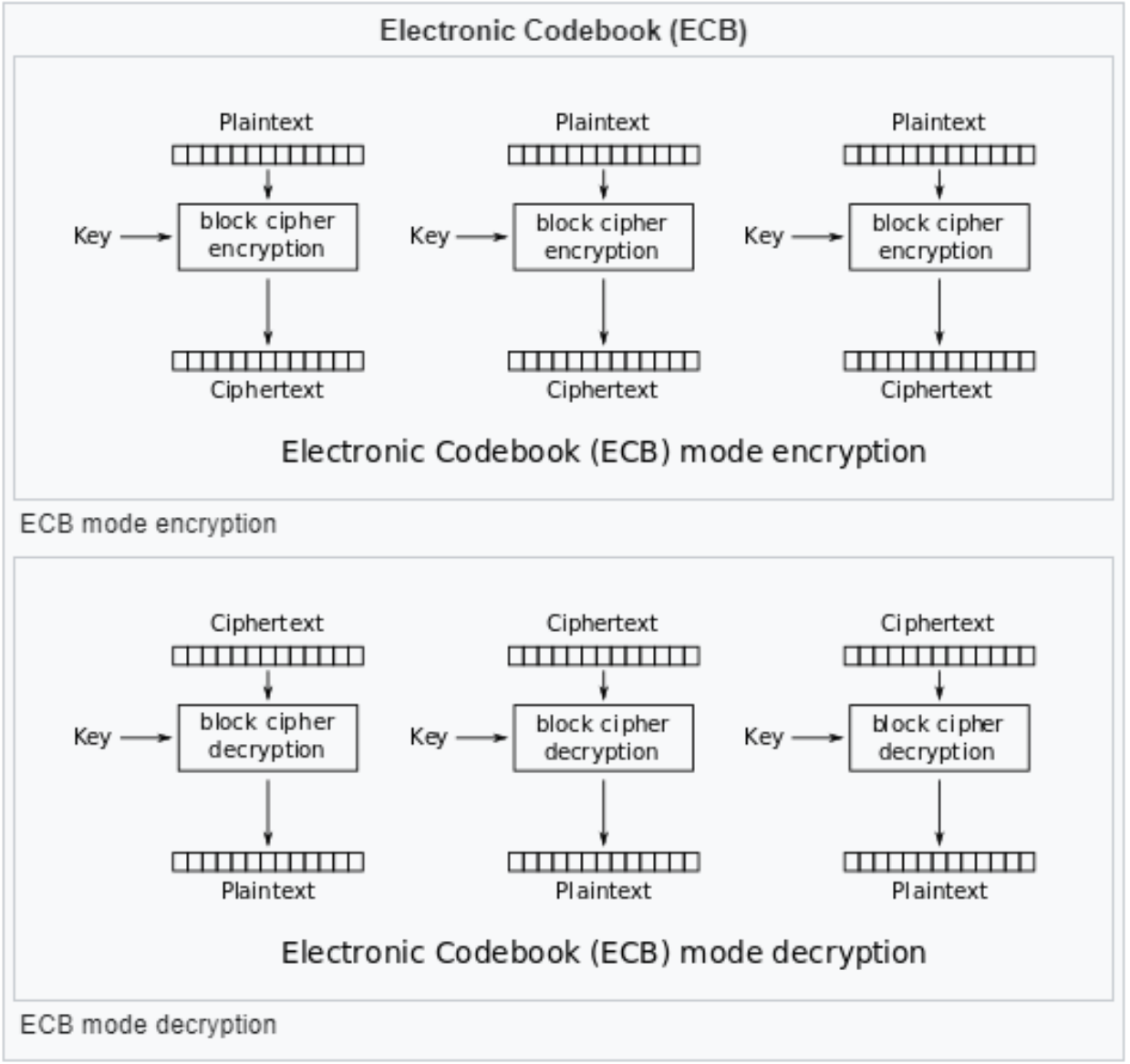
- A **block cipher mode of operation** is an algorithm that uses a block cipher to provide information security such as confidentiality or authenticity

- A block cipher is a cryptographic transformation of **one fixed-length group of bits** called a block

- A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely **transform amounts of data larger than a block**

1-ECB (Electronic codebook)

- The message is divided into blocks (of 64 bits).
- Each block is encrypted separately.
- The same plaintext block is always encrypted by the same ciphertext block (with the same key)
- Block size of 64 bits = a codebook of 2^{64} entries
- For each key = a different codebook.



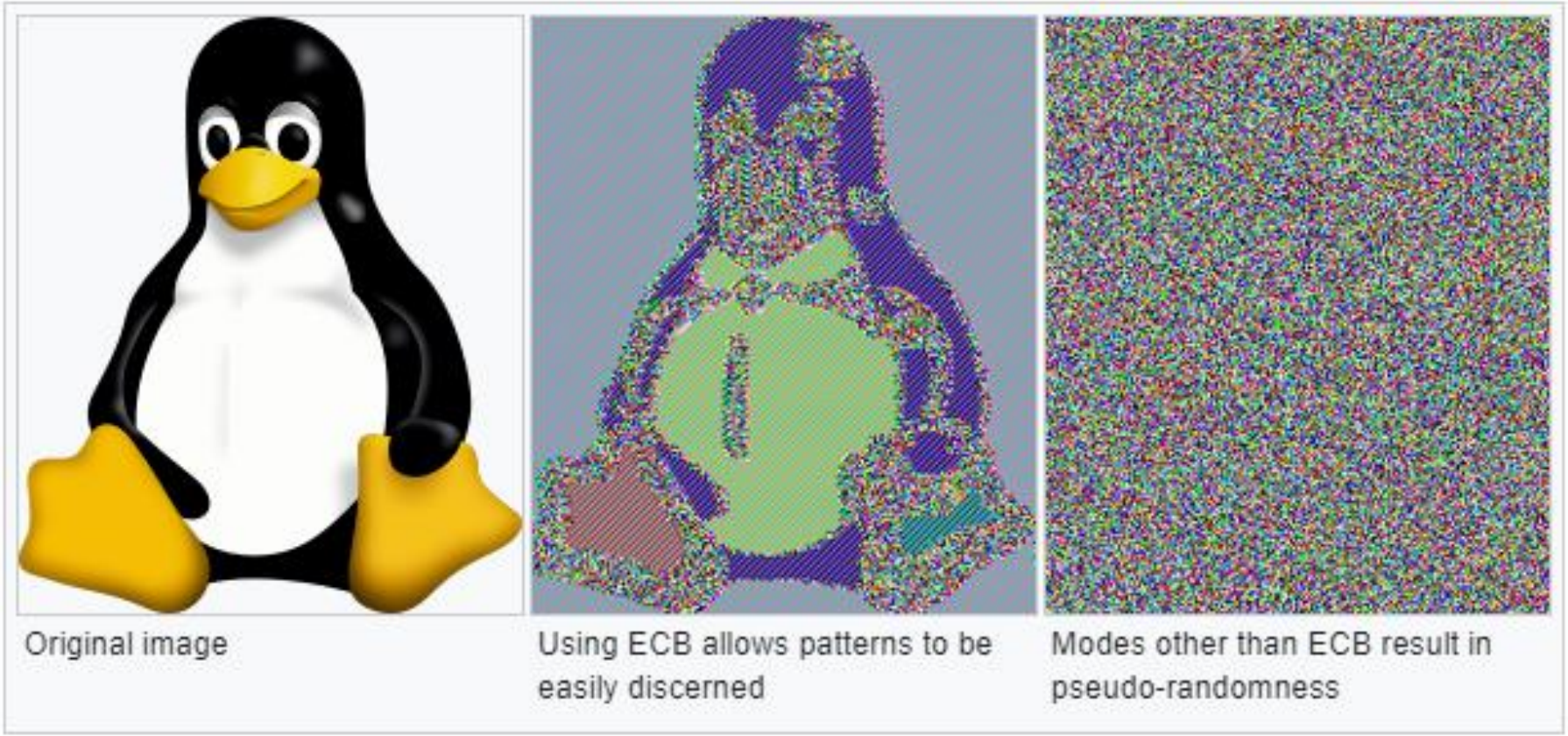
1-ECB (Electronic codebook)

- **Advantages:**
  - Prallelizable encryption/decryption: each block is encrypted independently of the others(non-linear encryption: no order).

- **Disadvantages:**
  - Fails to hide data patterns when it encrypts identical plaintext blocks into identical ciphertext blocks
  - Susceptible to replay attacks, since each block gets decrypted in exactly the same way

1-ECB (Electronic codebook): Disadvantages

- **Example 1:**
  - ECB can leave plaintext data patterns in the ciphertext (can be seen when ECB mode is used to encrypt a bitmap image)



Original image

Using ECB allows patterns to be easily discerned

Modes other than ECB result in pseudo-randomness

1-ECB (Electronic codebook): Disadvantages

- **Example 2:**
  - Identical blocks

The following two messages are encrypted with an ECB mode and a block cipher algorithm that works with one block of two characters at a time. This type of file could correspond to a list of salaries.

```
JOHN__105000
JACK__500000
```

The encryption on the first message gives this:

```
JO|HN|__|10|50|00
Q9|2D|FP|VX|C9|IO
```

And on the second message, we obtain:

```
JA|CK|__|50|00|00
LD|AS|FP|C9|IO|IO
```

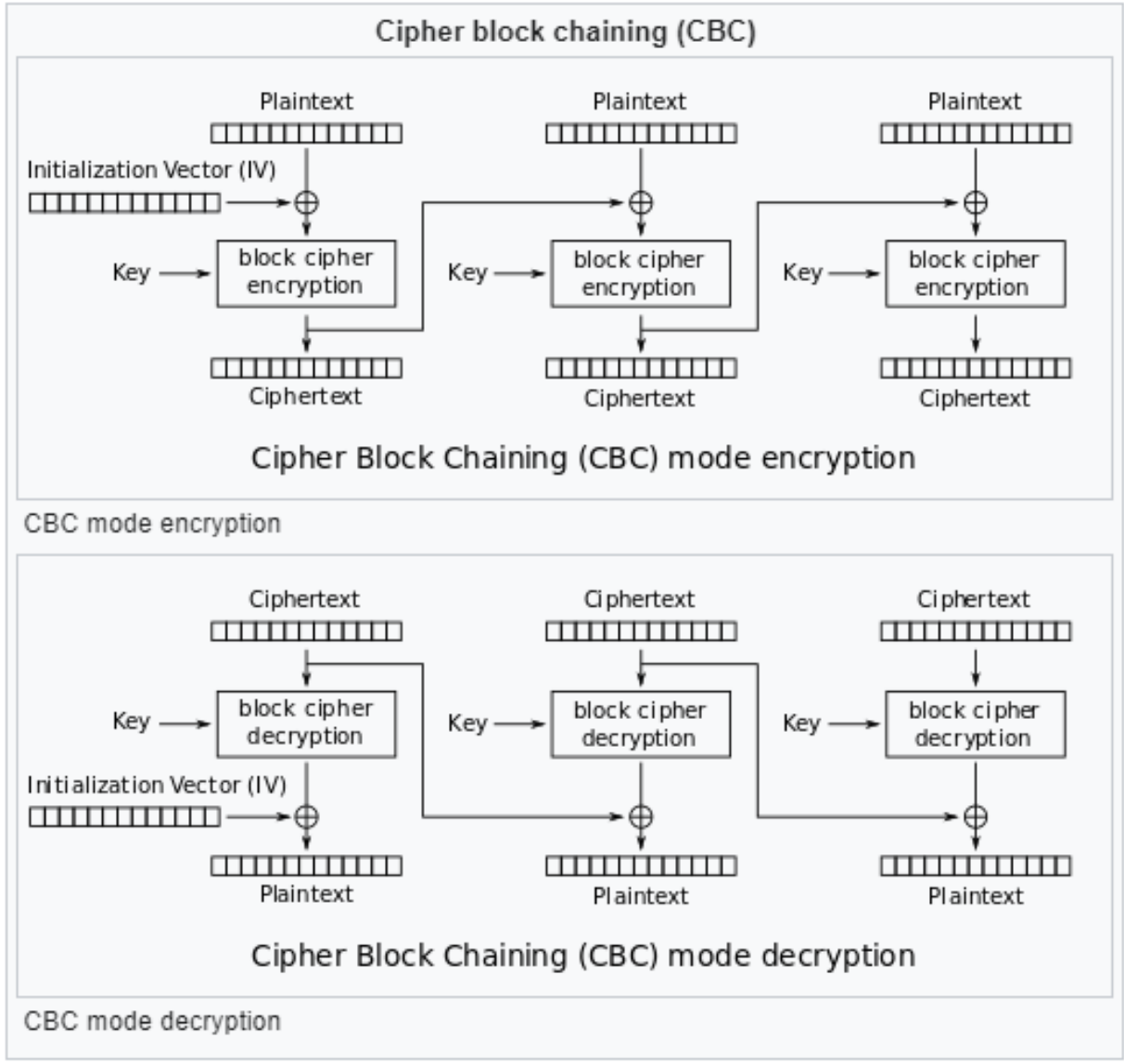
We see that pairs of characters appear in the two encrypted messages, the same is true in the plain text messages:

```
Q9|2D| FP |VX| C9 | IO
LD|AS| FP | C9 | IO|IO
```

Assuming that John knows his salary, he could guess Jack's salary because the sequence "C9" corresponds to "50" and "IO" to "00". John deduces that Jack's salary, encrypted as "C9IOIO", corresponds to "500000".

2-CBC (Cipher Block Chaining)

- Each block of plaintext is XORed with the previous ciphertext block before being encrypted.
- Each ciphertext block depends on all plaintext blocks processed up to that point.
- To make each message unique, an initialization vector must be used in the first block.



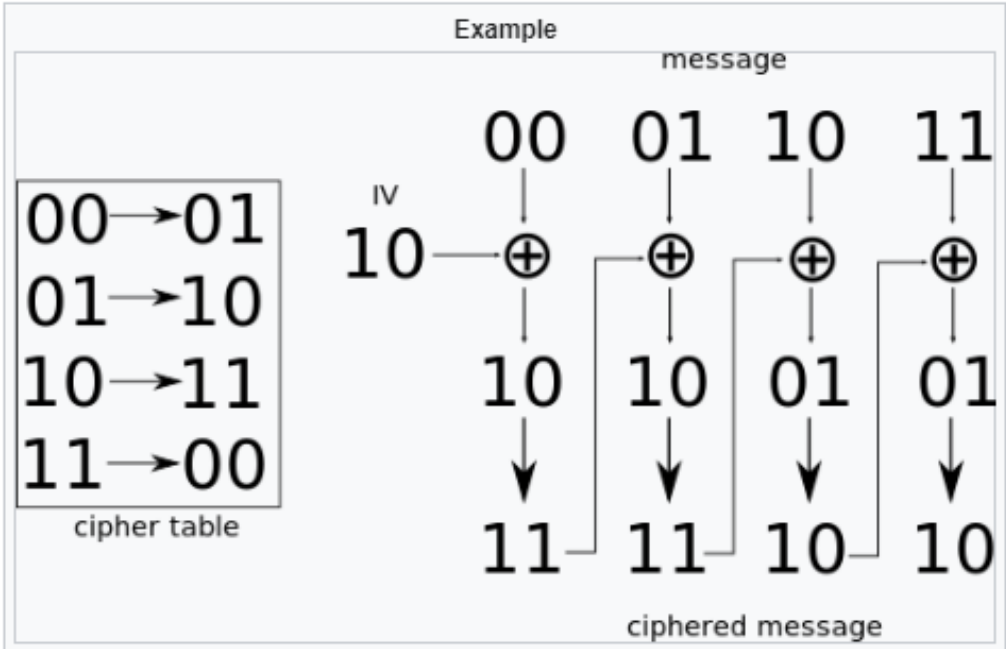
2-CBC (Cipher Block Chaining)

- Encryption: (Not-Parallelizable)

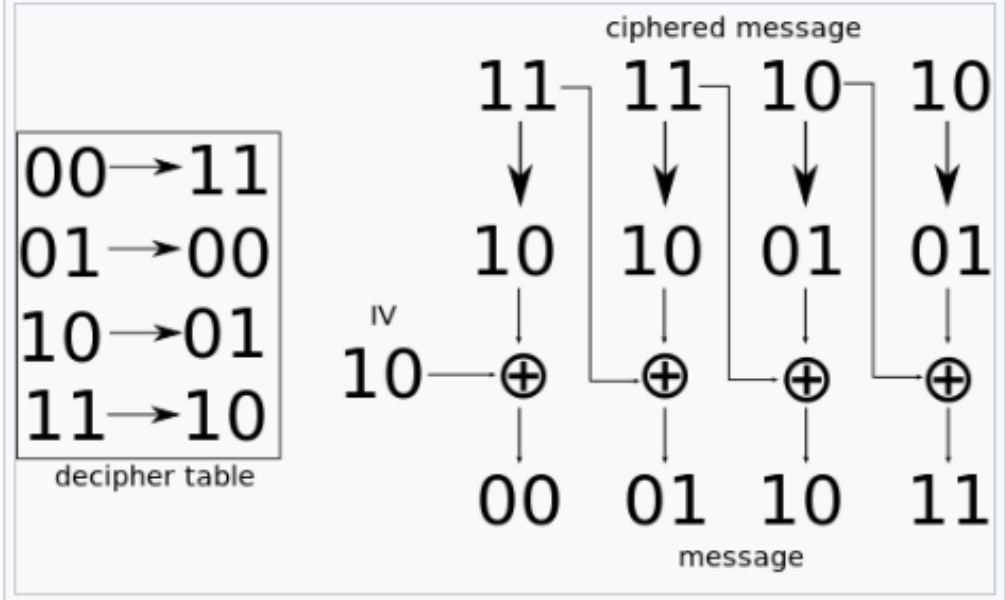
$$C_i = E_K(P_i \oplus C_{i-1}),$$
$$C_0 = IV,$$

- Decryption: (Parallelizable)

$$P_i = D_K(C_i) \oplus C_{i-1},$$
$$C_0 = IV.$$

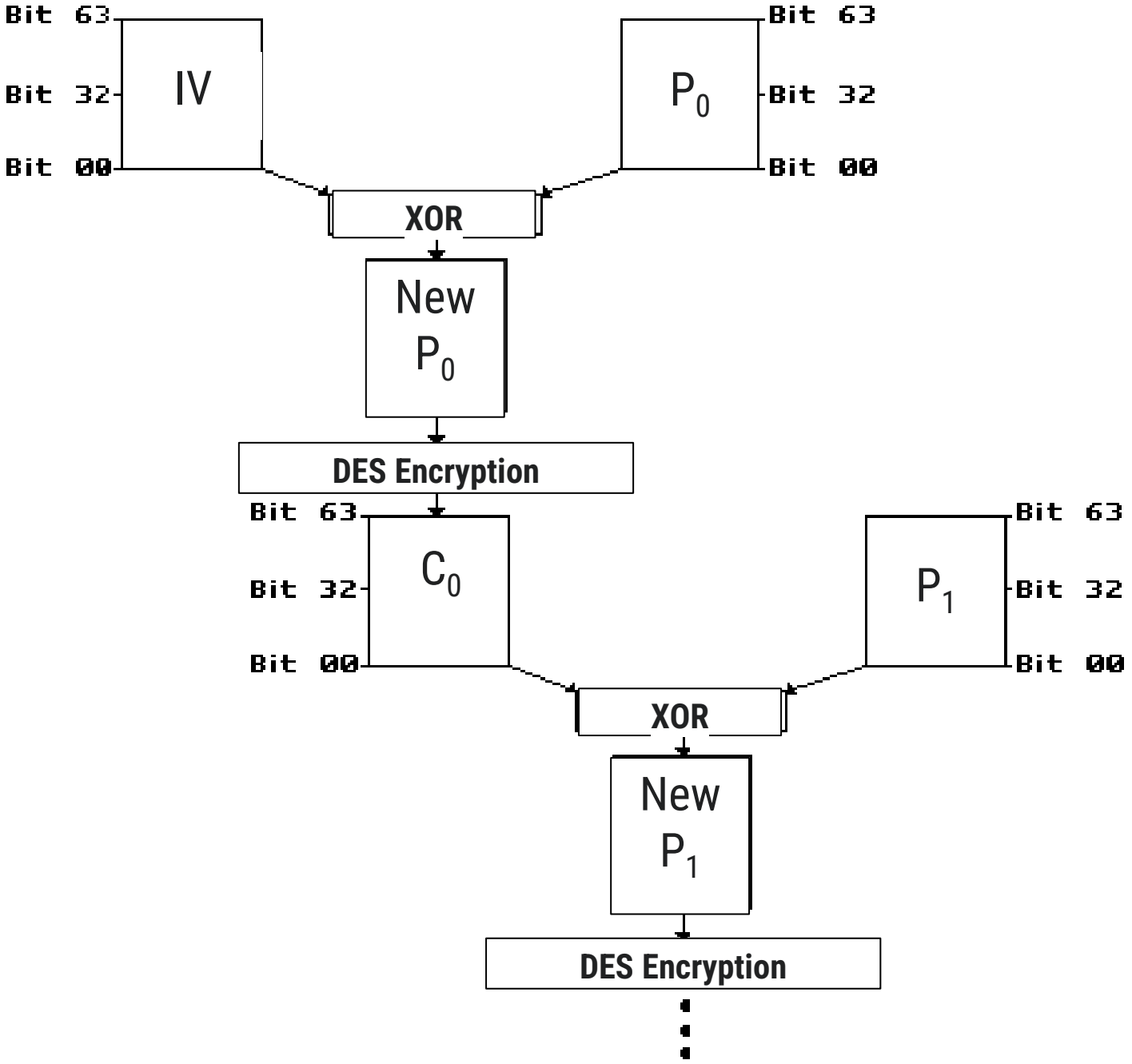


CBC encryption example with a toy 2-bit cipher



CBC decryption example with a toy 2-bit cipher

2-CBC (Cipher Block Chaining): Operational mode



3-PCBC (Propagating Cipher Block Chaining)

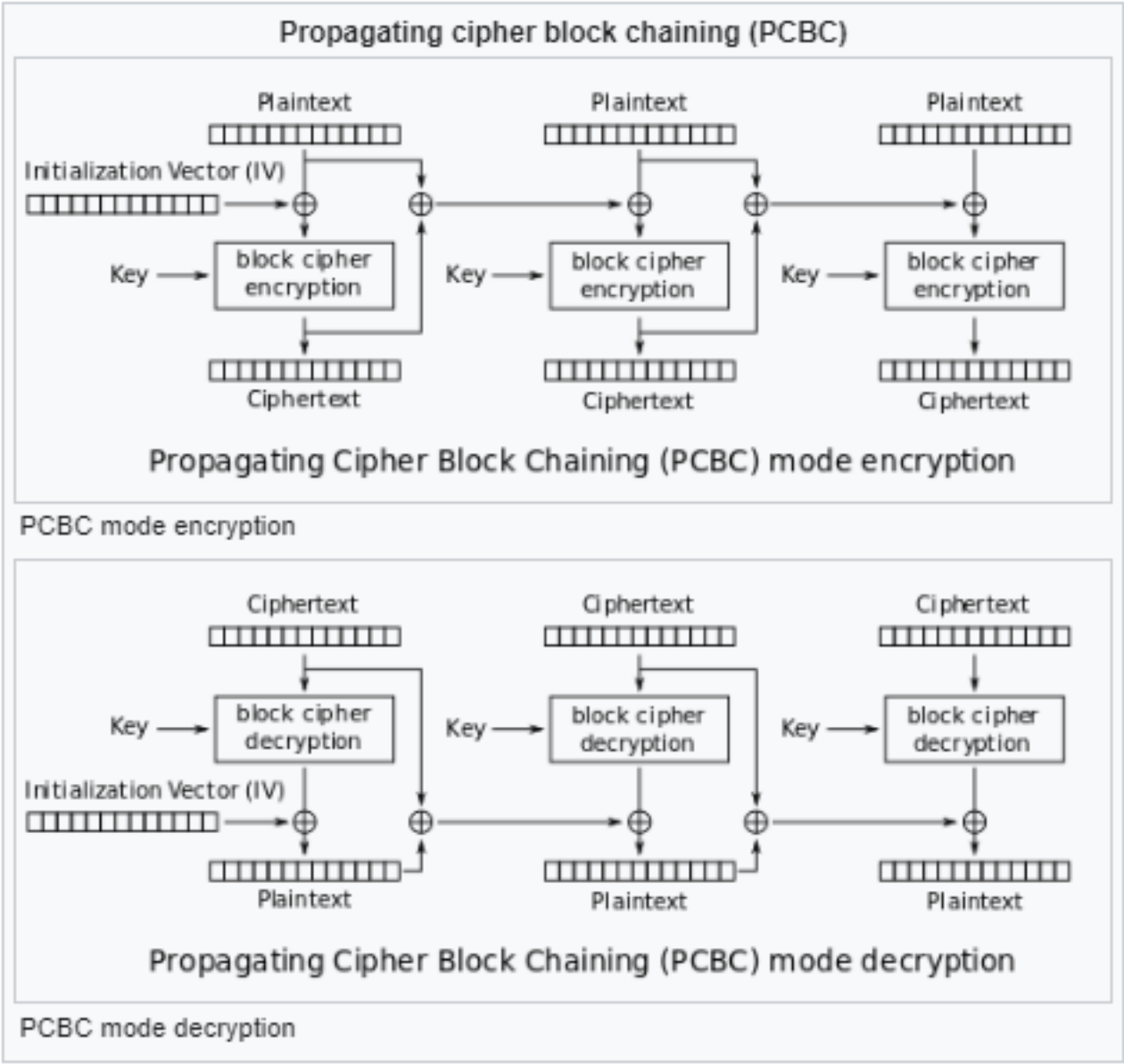
- Encryption: (Not-Parallelizable)

$$C_i = E_K(P_i \oplus P_{i-1} \oplus C_{i-1}), P_0 \oplus C_0 = IV$$

- Decryption: (Not-Parallelizable)

$$P_i = D_K(C_i) \oplus P_{i-1} \oplus C_{i-1}, P_0 \oplus C_0 = IV.$$

PCBC is used in **Kerberos v4** and **WASTE**



4-CFB (Cipher FeedBack)

4.1 Full-block CFB

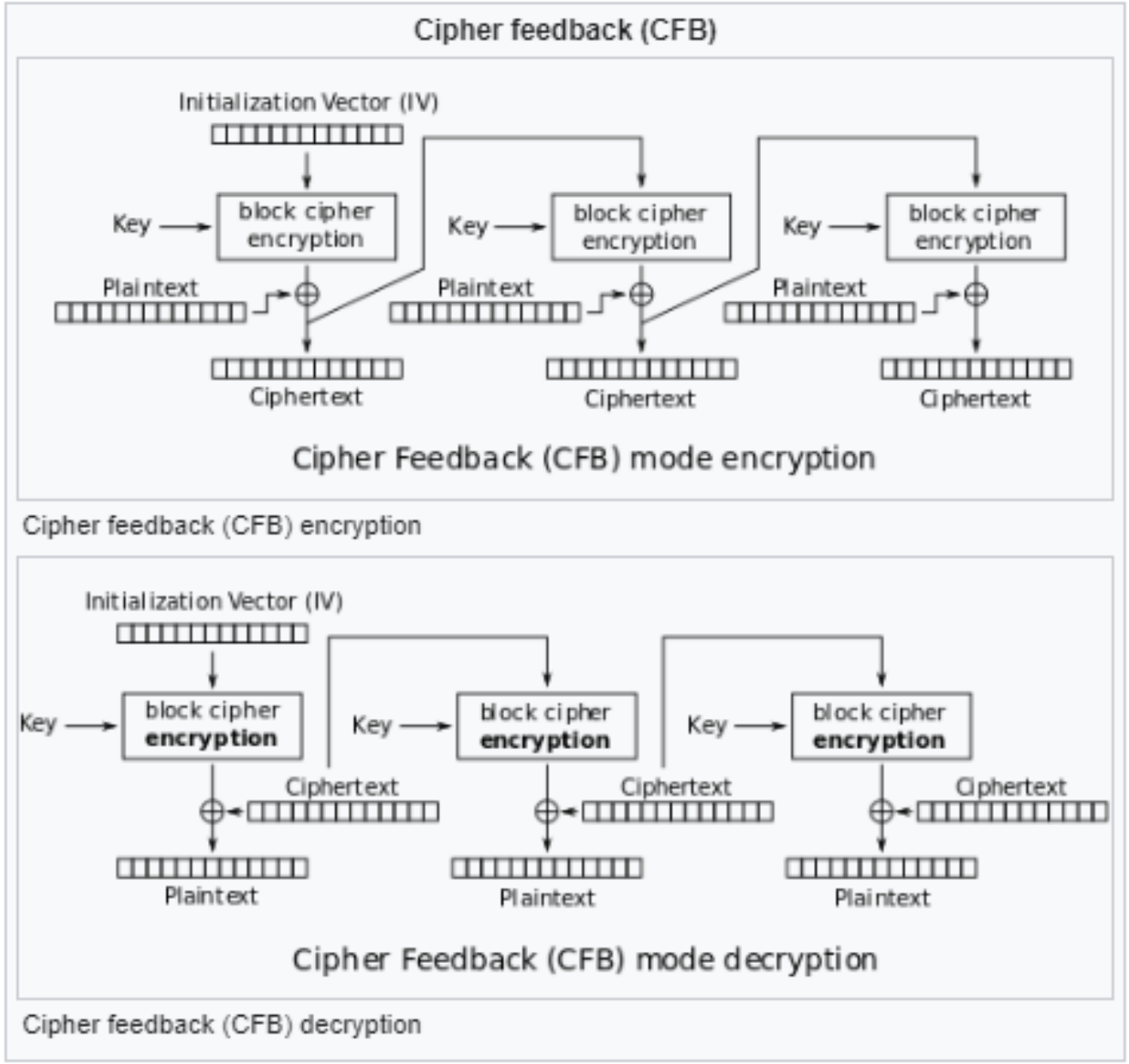
- It uses the entire output of the block cipher
- Identical to CBC

- Encryption: (Not-Parallelizable)

$$C_i = \begin{cases} \mathrm{IV}, & i = 0 \\ E_K(C_{i-1}) \oplus P_i, & \text{otherwise} \end{cases}$$

- Decryption: (Parallelizable)

$$P_i = E_K(C_{i-1}) \oplus C_i$$



4-CFB (Cipher FeedBack)

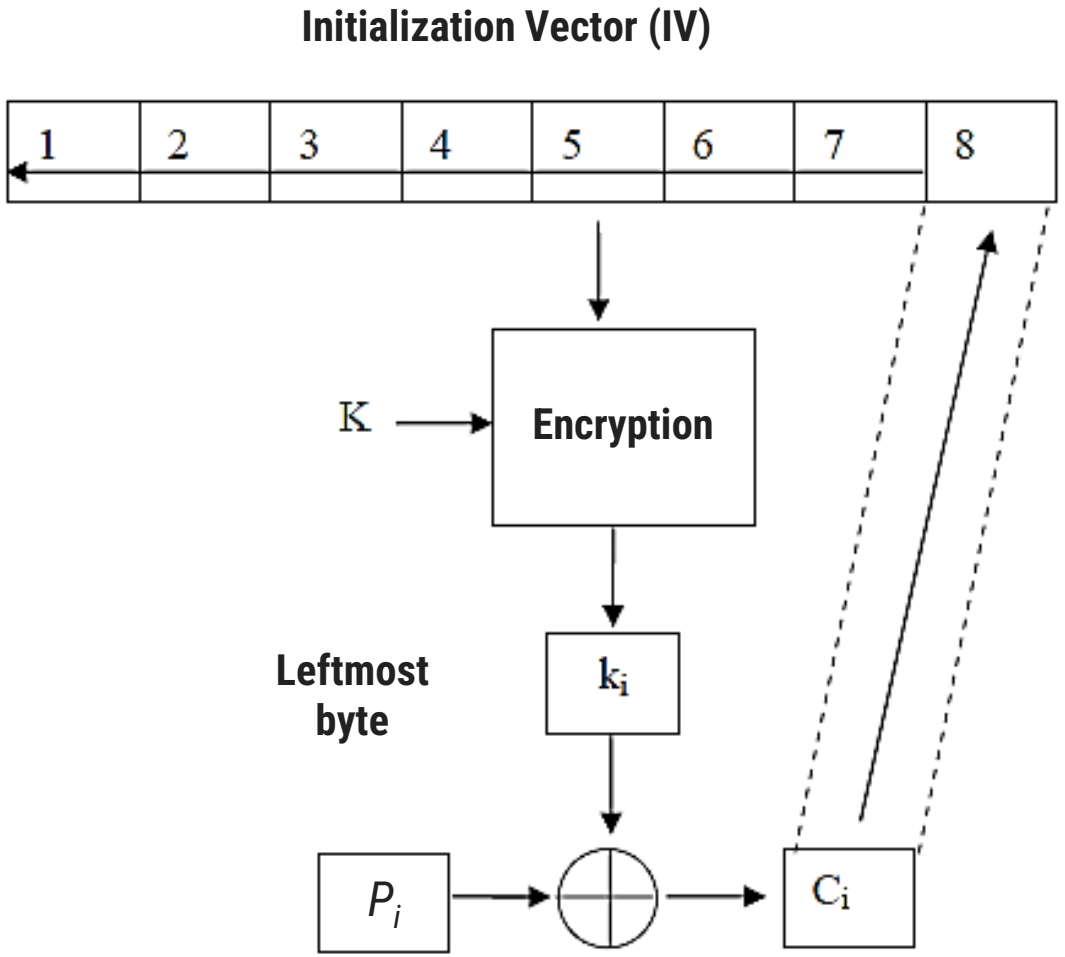
4.2 Partial-block CFB

- CFB with a bit-width (CFB-1, CFB-8, CFB-64,..)
- Requires an integer parameter, denoted **s**, such that **1 ≤ s ≤ b**. (b is full-block size)
- Each plaintext segment (P_i) and ciphertext segment (C_i) consists of **s** bits

- **Encryption/Decryption:**

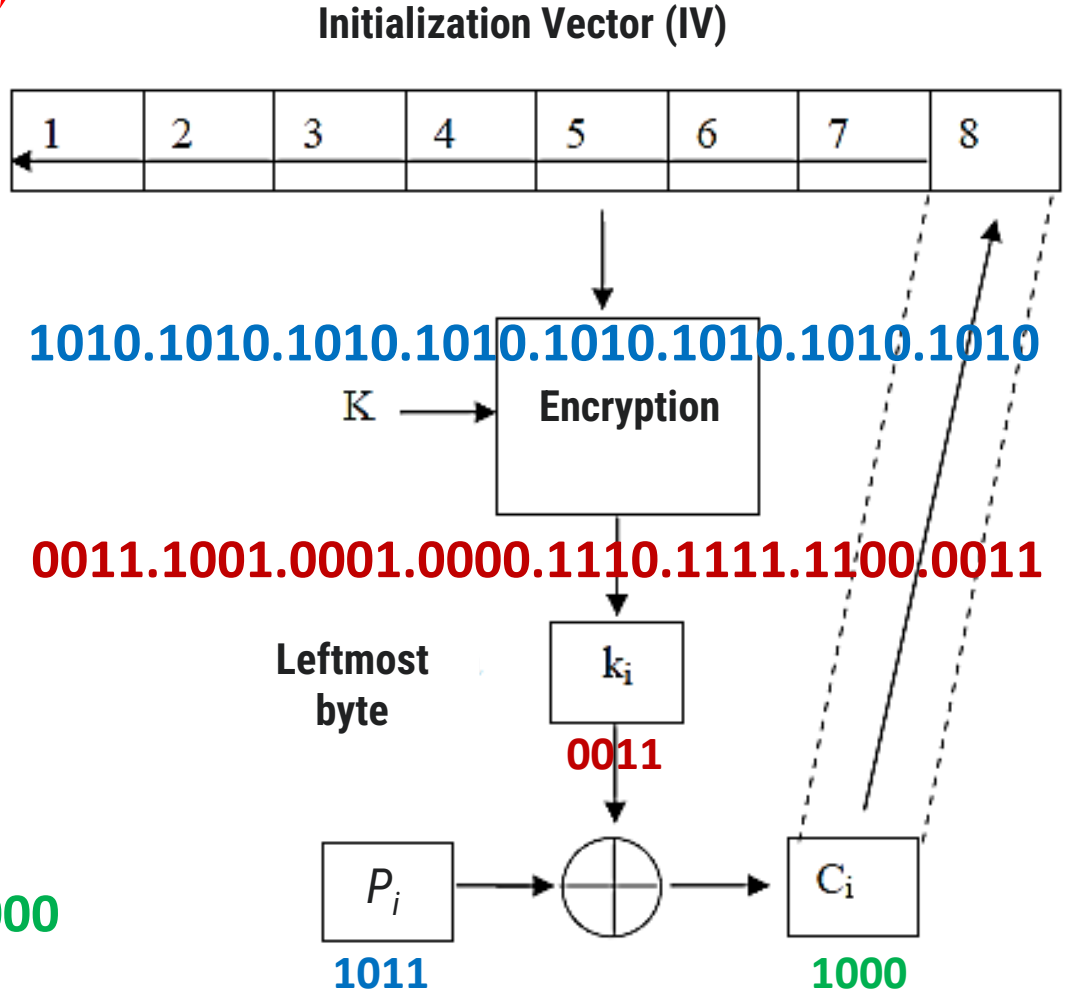
$$I_0 = \mathrm{IV}.$$
$$I_i = \left((I_{i-1} \ll s) + C_i\right) \bmod 2^b,$$
$$C_i = \mathrm{MSB}_s\left(E_K(I_{i-1})\right) \oplus P_i,$$
$$P_i = \mathrm{MSB}_s\left(E_K(I_{i-1})\right) \oplus C_i,$$

CFB-8 illustration:



- **CFB-4 illustration:**
  - M = Hey = 1011.1010.1101
  - K = 1010
  - IV = 1001.0011.1011.1010.0100.0101.0110.1001

1001.0011.1011.1010.0100.0101.0110.1001



~~1001~~.0011.1011.1010.0100.0101.0110.1001.1000

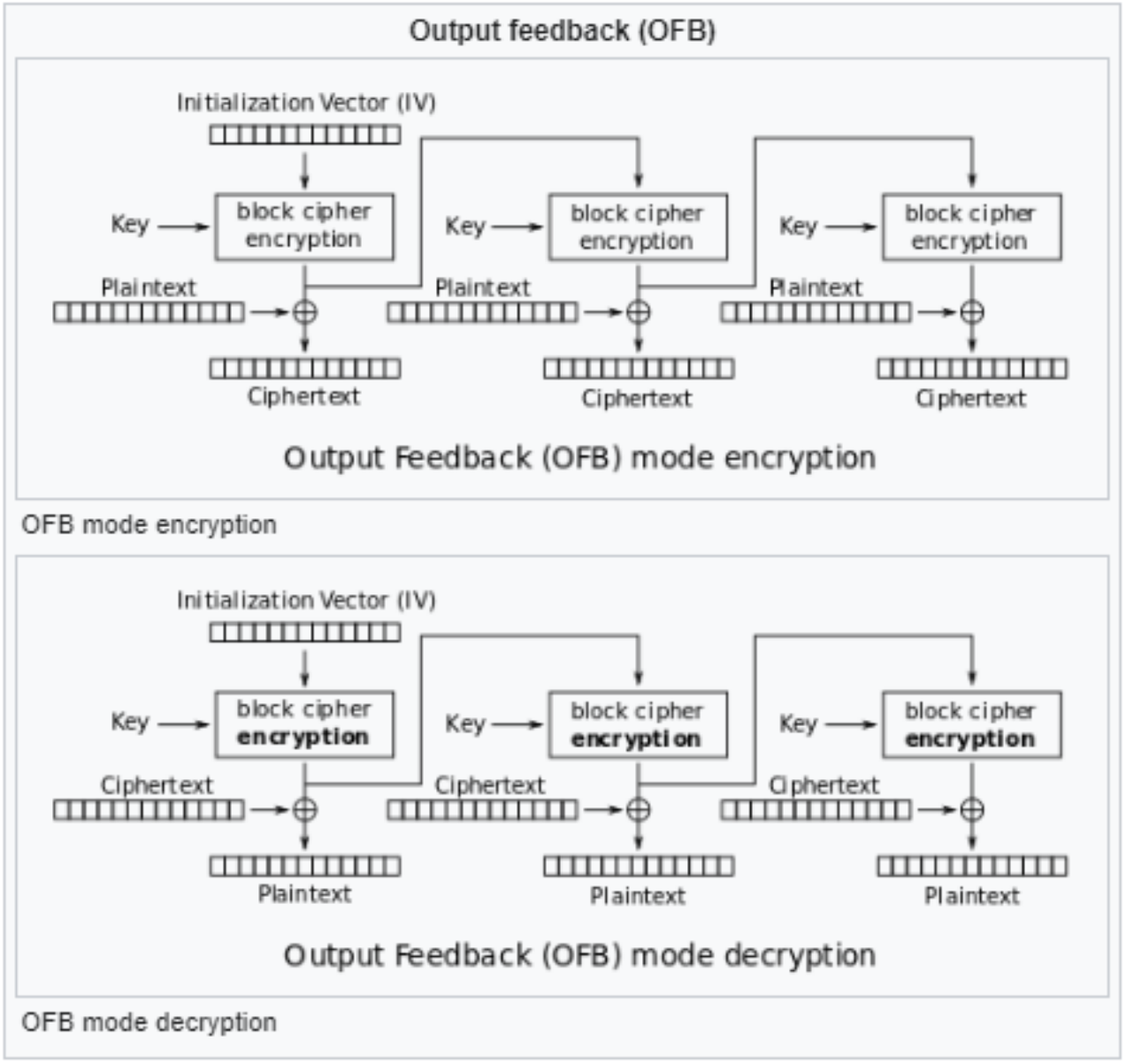
5-OFB (Output FeedBack)

5.1 Full-block OFB

- It uses the entire output of the IV block cipher (similar to CFB but uses K_i instead of C_i as output)

- Encryption/Decryption: (Not-Parallelizable)

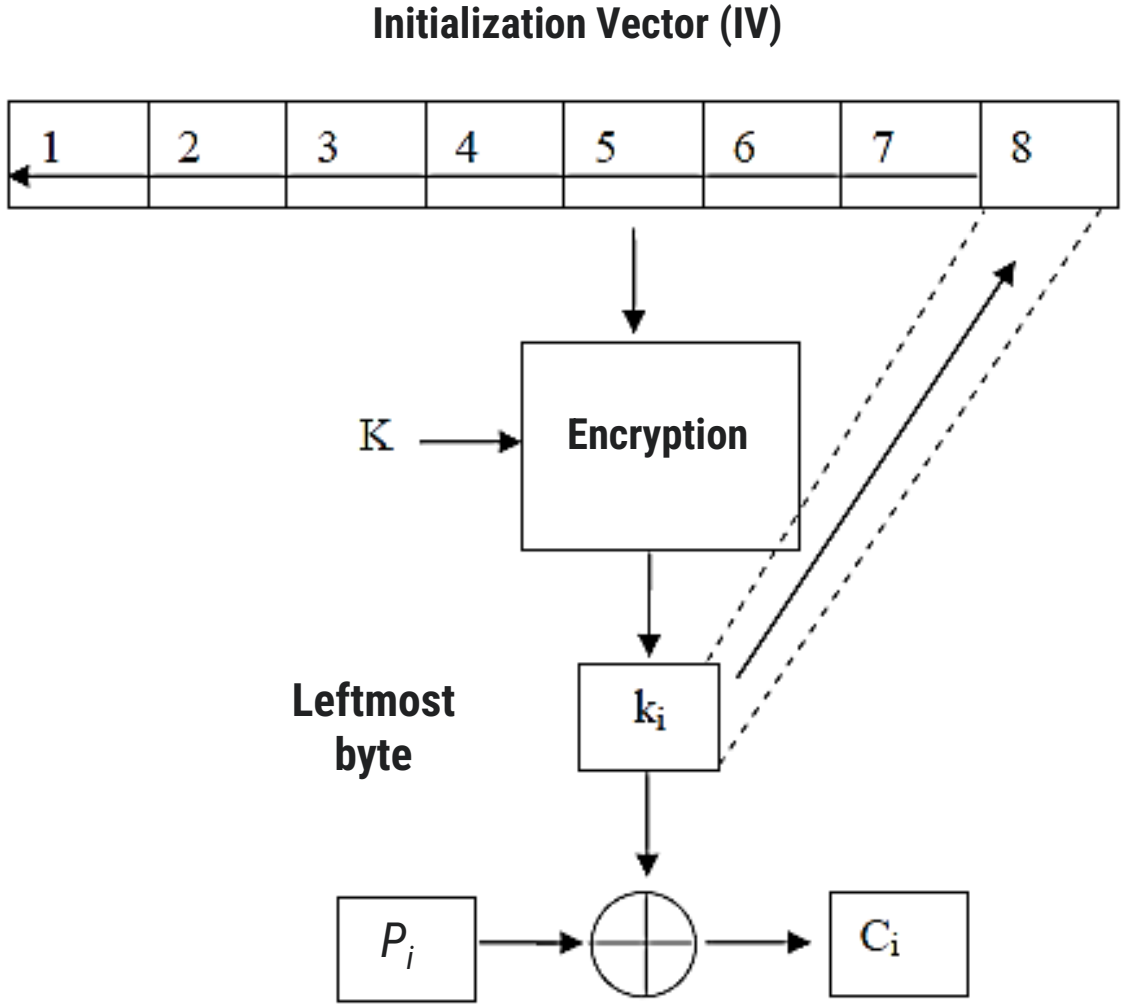
$$C_j = P_j \oplus M_j,$$
$$P_j = C_j \oplus O_j,$$
$$O_j = E_K(I_j),$$
$$I_j = O_{j-1},$$
$$I_0 = \mathrm{IV}.$$



5-OFB (Output FeedBack)

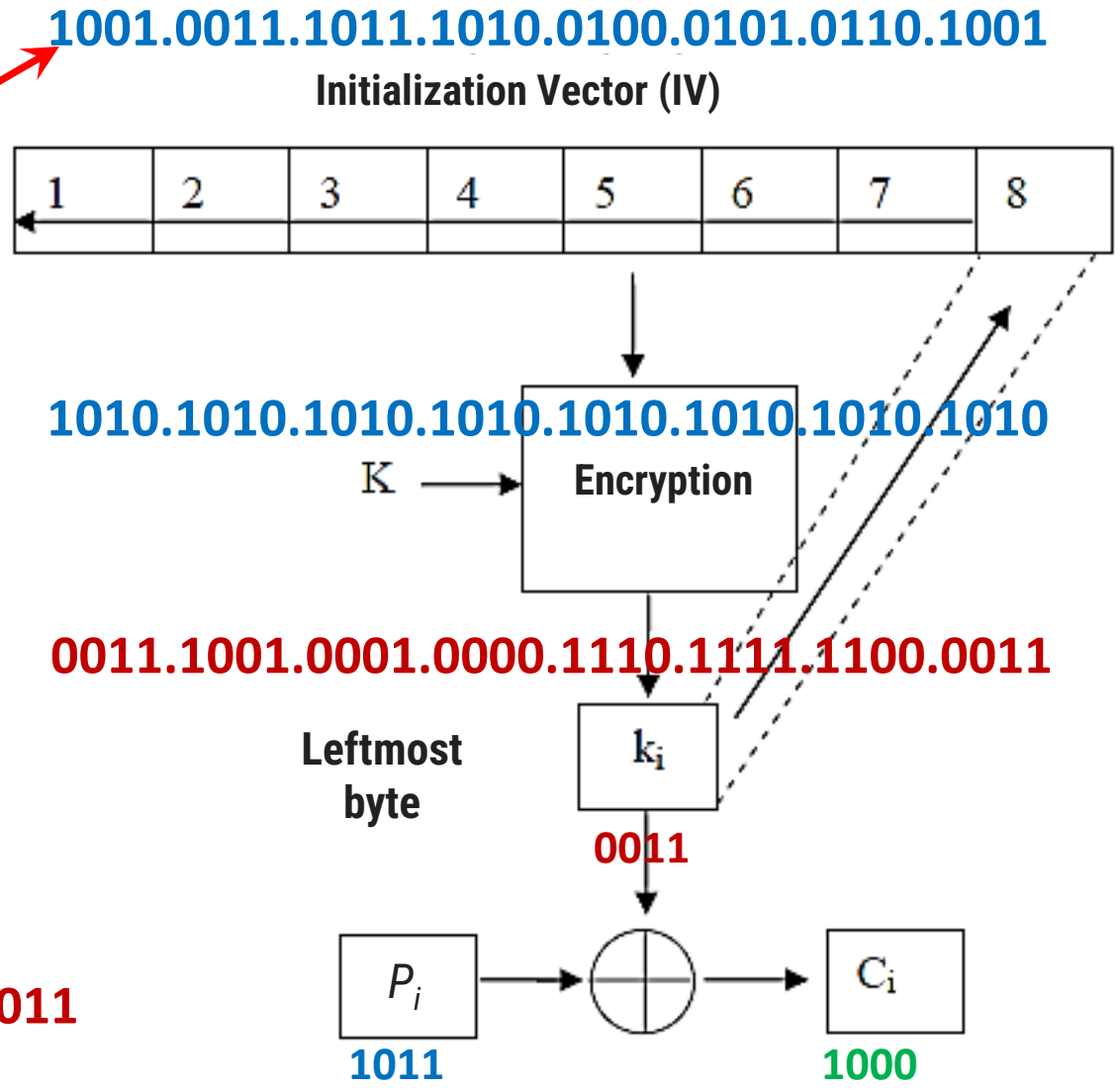
5.2 Partial-block OFB

CFB-8 illustration:



5-OFB-4 illustration

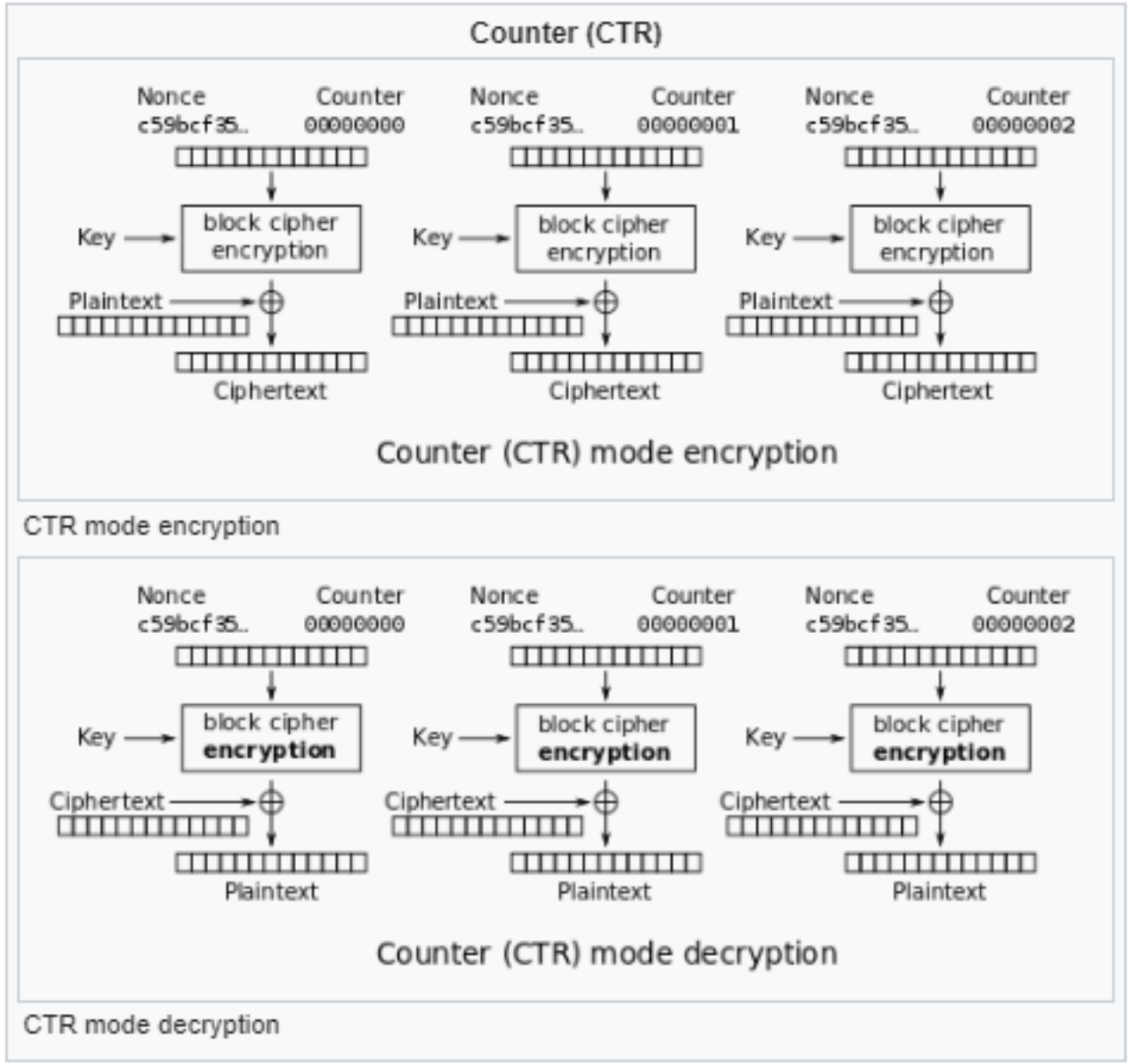
- M = Hey = 1011.1010.1101
- K = 1010
- IV = 1001.0011.1011.1010.0100.0101.0110.1001



~~1001~~.0011.1011.1010.0100.0101.0110.1001.0011

6-Counter (CTR)

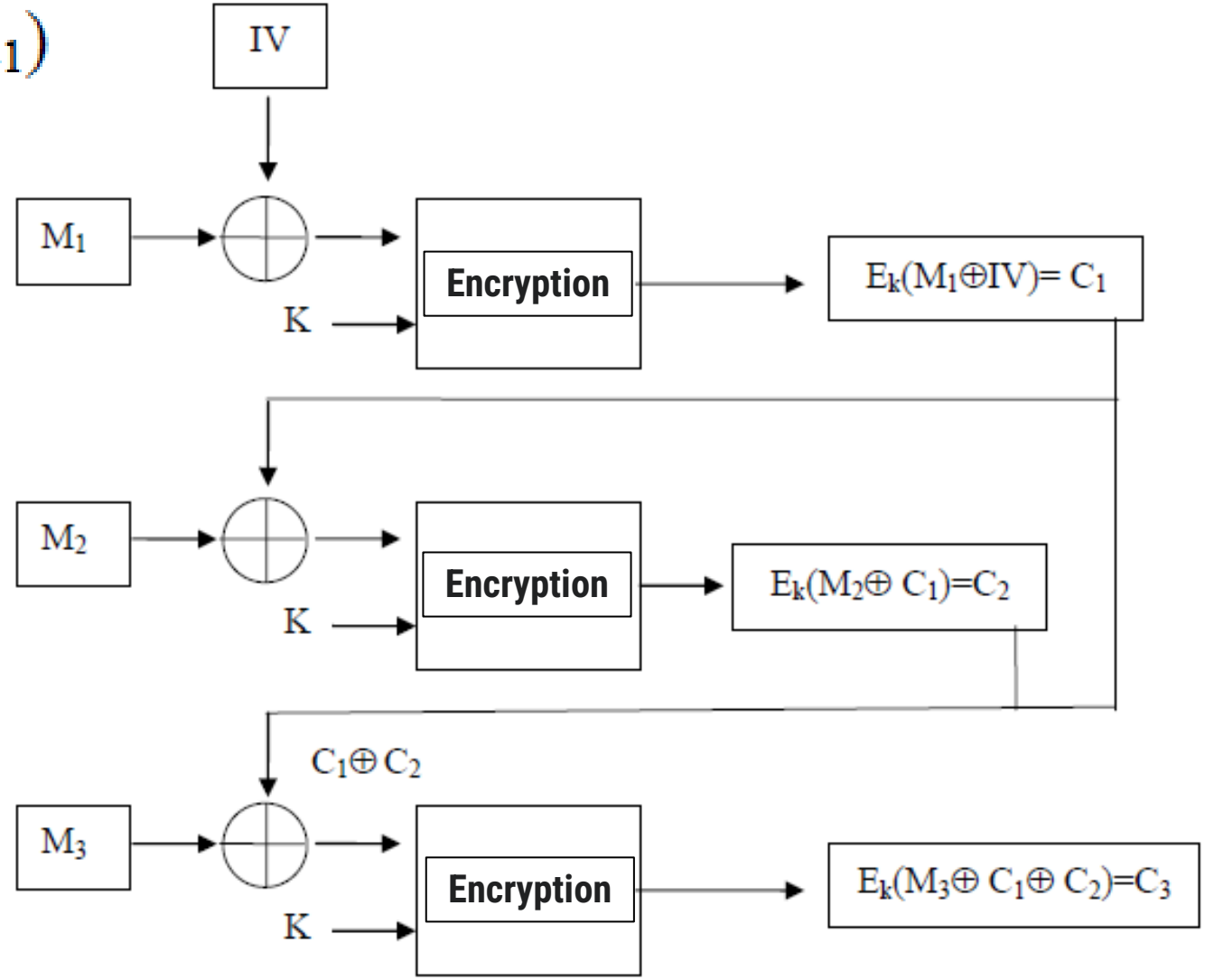
- Instead of using the encryption output k_i to fill the shift register, we use a counter
- After encryption of a block, increment the counter



7-BC (Block Chaining)

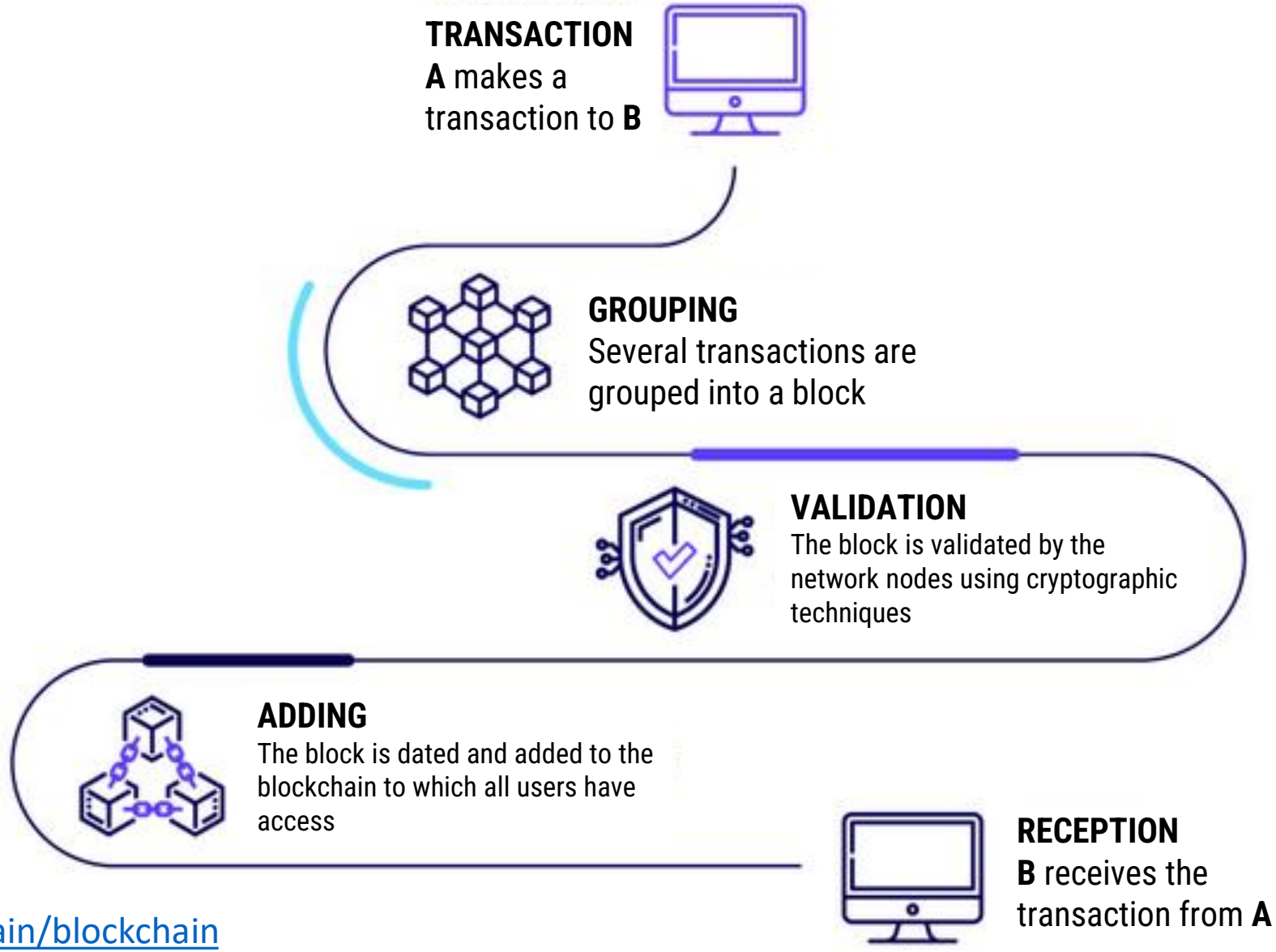
- Each block is encrypted based on all previously encrypted blocks.

$$C_i = E_k(M_i \oplus C_1 \oplus \ldots C_{i-1})$$



7-BC (Block Chaining)

- Operational mode (Cryptocurrency)



https://andersbrownworth.com/blockchain/blockchain

8-Overencryption

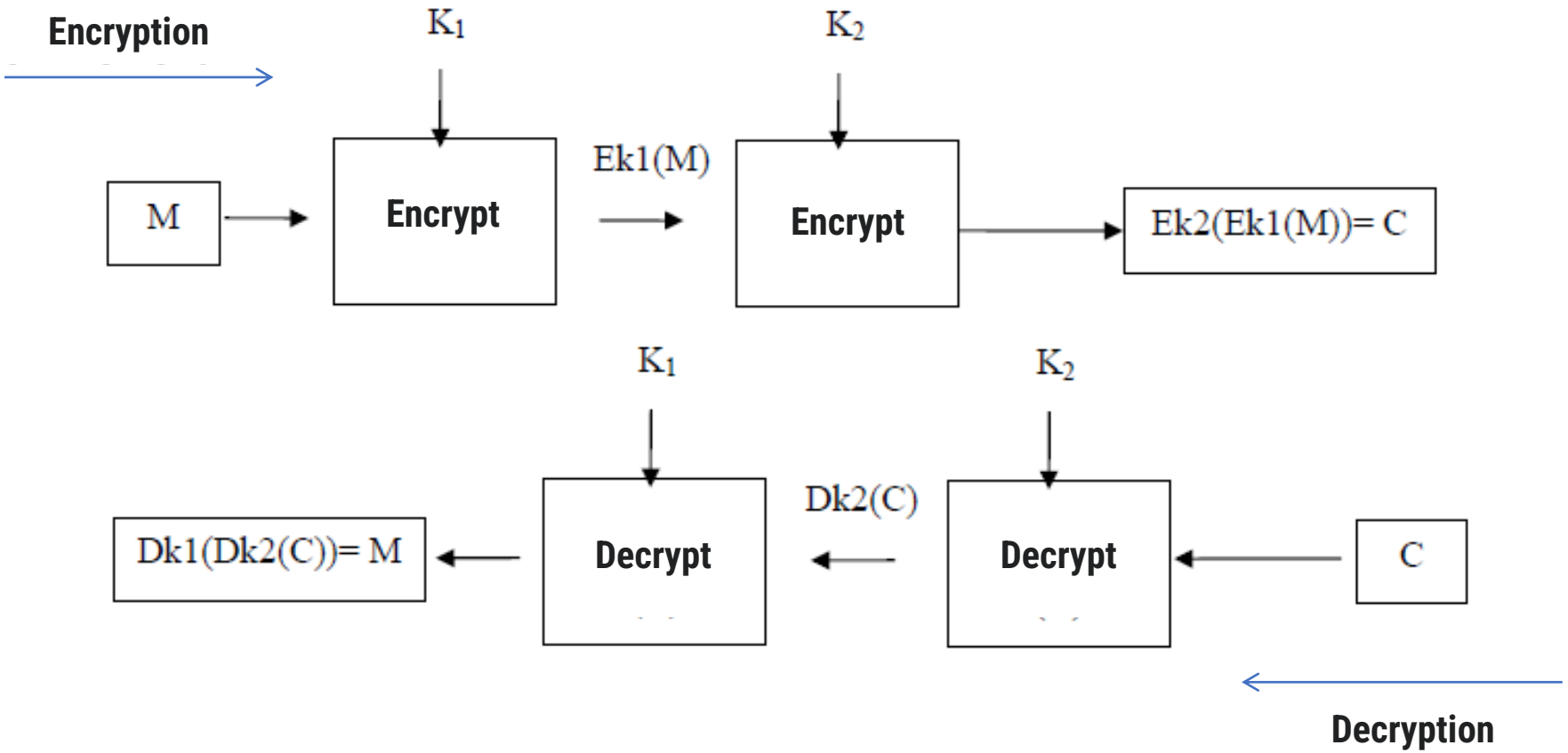
Principle: Encrypting the same plaintext block multiple times.

Example: Encrypting a text twice using the same algorithm and the same key.

Disadvantage: Does not change the complexity of a brute force attack.

Solution: Increase security level by using multiple different keys.

8.1 Double encryption



An exhaustive search requires 2^{2n} attempts
(e.g., a 64-bit block requires 2^{128} attempts)

8.2 Triple encryption (EDE : Encrypt-Decrypt-Encrypt)

- Encrypt a block three times with two different keys:

Encrypt + Decrypt + Encrypt

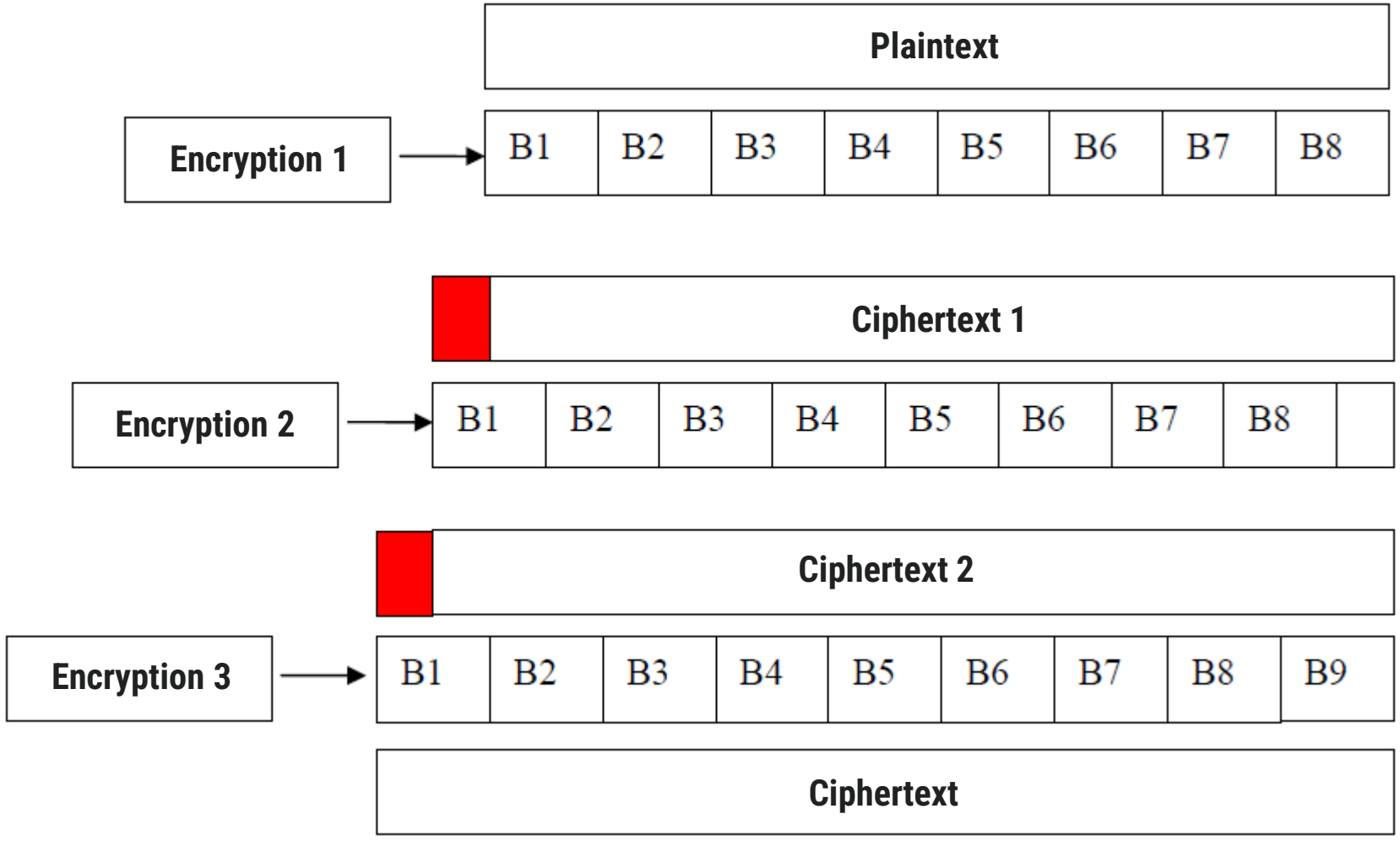
$$C=Ek1(Dk2(Ek1(M)))$$
$$M=Dk1(Ek2(Dk1(C)))$$

- Or encrypt a block with three different keys:

$$C=Ek3(Dk2(Ek1(M)))$$
$$M=Dk1(Ek2(Dk3(C)))$$

8.3 Triple encryption with padding

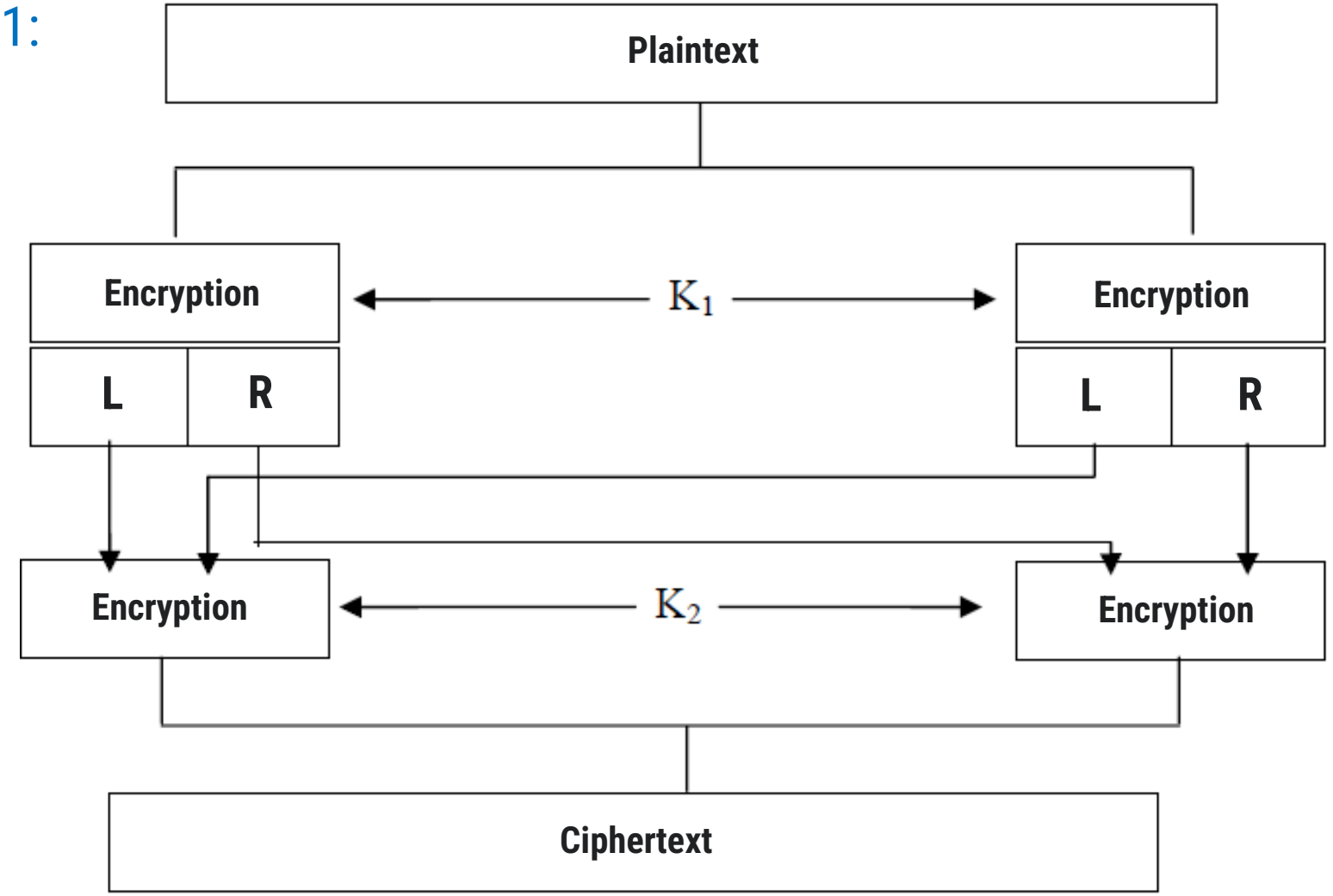
Completing the text with a string of bits (1/2 the length of the block) between the 1st and 2nd encryption, and between the 2nd and 3rd encryption



8.4 Doubling the block length

Double the block length of an algorithm using double encryption

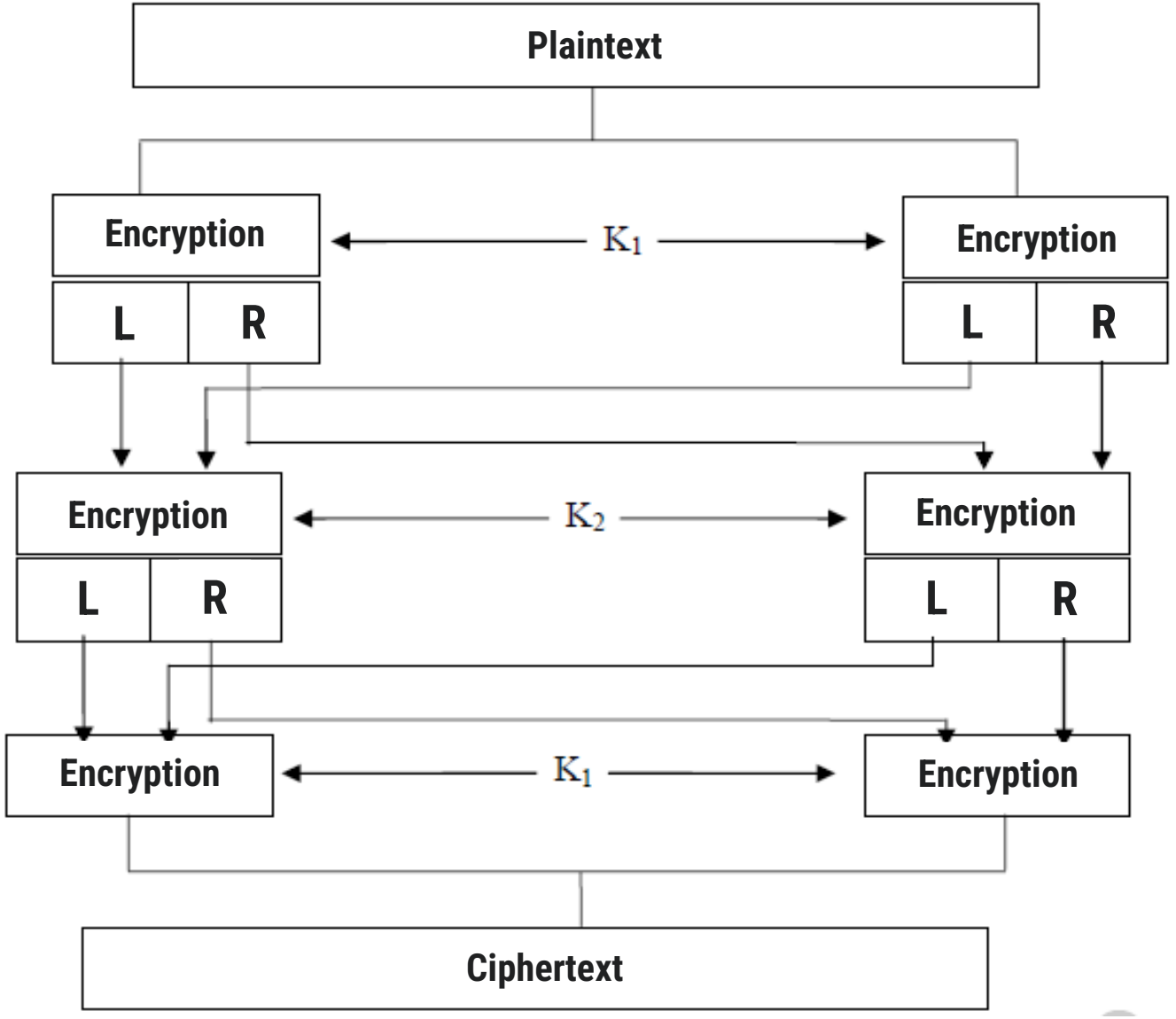
Solution 1:



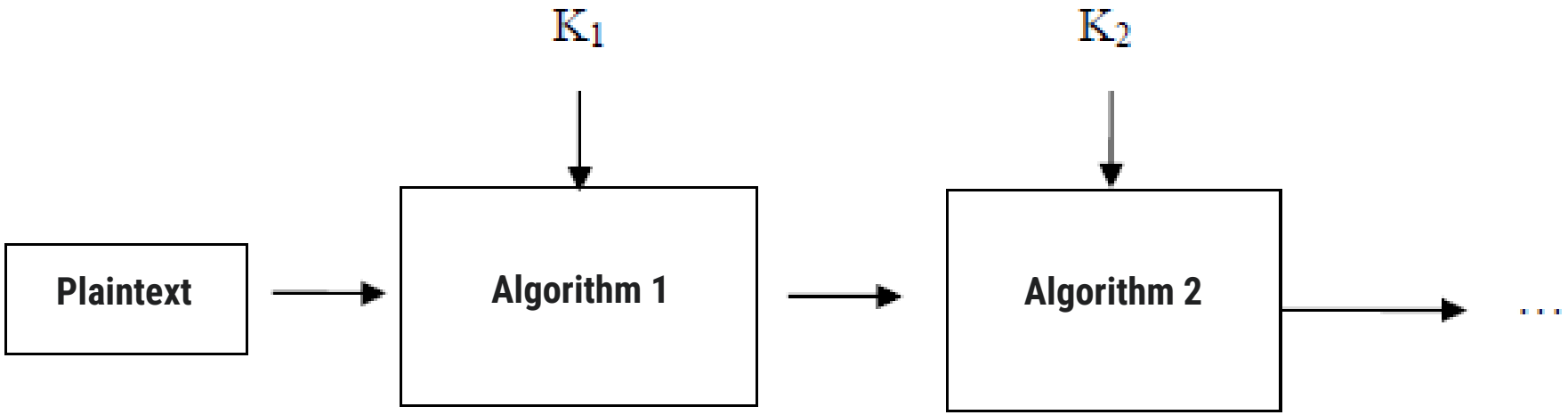
8.4 Doubling the block length

Double the block length of an algorithm using double encryption

Solution 2:



8.5 Over-encryption with several algorithms



9. Continuous encryption

Principle: Encrypts and decrypts 1 bit at a time.

Disadvantages: Software implementation is difficult because manipulating bits is computationally expensive.

Advantages:

- Minimizing Error Propagation: a 1-bit error in encrypted text results in only one erroneous bit in the decryption operation, unlike block encryption where 1 error in encrypted text leads to at least 1 erroneous block in the decryption operation.