Chapter 6

Block Cipher Operation

Dr. Shin-Ming Cheng

NTUST
CONNECTIVITY LAB

Block Ciphers

- A block cipher is much more than just an encryption algorithm
  - build different types of block-based encryption schemes
  - realize stream ciphers
  - construct hash functions
  - make message authentication codes
  - build key establishment protocols
  - make a pseudo-random number generator
- The security of block cipher is increased by
  - key whitening
  - multiple encryption

Content of this Chapter

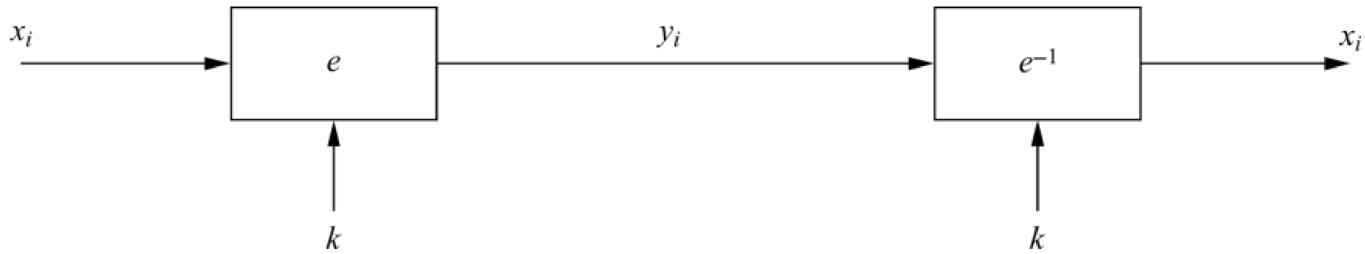
- Encryption with Block Ciphers/Modes of Operation
  - Electronic Code Book mode (ECB)
  - Cipher Block Chaining mode (CBC)
  - Output Feedback mode (OFB)
  - Cipher Feedback mode (CFB)
  - Counter mode (CTR)
- Exhaustive Key Search Revisited
- Increasing the Security of Block Ciphers

Encryption with Block Ciphers

- Ways of encrypting long plaintexts with a block cipher (modes of operation)
  - Electronic Code Book mode (ECB) / Cipher Block Chaining mode (CBC) / Output Feedback mode (OFB) /Cipher Feedback mode (CFB) / Counter mode (CTR)
- All of the 5 modes have one goal:
  - In addition to confidentiality, they provide authenticity/認證性 and integrity/完整性:
    - Is the message really coming from the original sender? (authenticity)
    - Was the ciphertext altered during transmission? (integrity)

Electronic Code Book mode (ECB)

- $e_k(x_i)$ denotes the encryption of a b-bit plaintext block x_i with key k
- $e_k^{-1}(y_i)$ denotes the decryption of b-bit ciphertext block y_i with key k
- Messages which exceed b bits are partitioned into b-bit blocks
- Each Block is encrypted separately



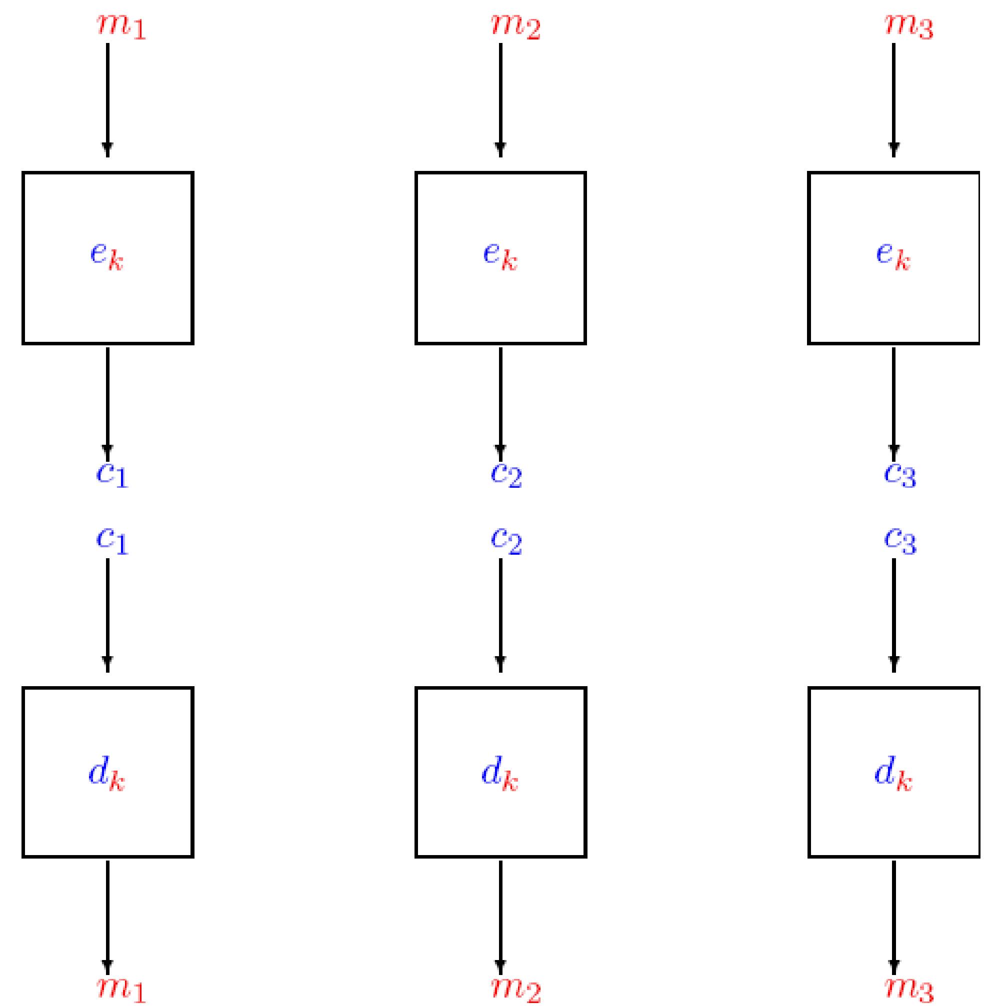
Encryption: $y_i = e_k(x_i),\ i \geq 1$

Decryption: $x_i = e_k^{-1}(y_i) = e_k^{-1}(e_k(x_i)),\ i \geq 1$

ECB Mode

› Encryption

› Decryption



ECB Mode

- Advantages
  - Bit errors caused by noisy channels only affect the corresponding block but not succeeding blocks
  - Block cipher operating can be parallelized
    - advantage for high-speed implementations

ECB Mode

- Disadvantages
  - ECB encrypts highly deterministically
    - identical plaintexts result in identical ciphertexts
    - an attacker recognizes if the same message has been sent twice
    - plaintext blocks are encrypted independently of previous blocks
      - an attacker may reorder ciphertext blocks which results in valid plaintext

Substitution Attack on ECB (1/2)

- Once a particular plaintext to ciphertext block mapping $x_i \rightarrow y_i$ is known, a sequence of ciphertext blocks can easily be manipulated
- Suppose an electronic bank transfer

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount $

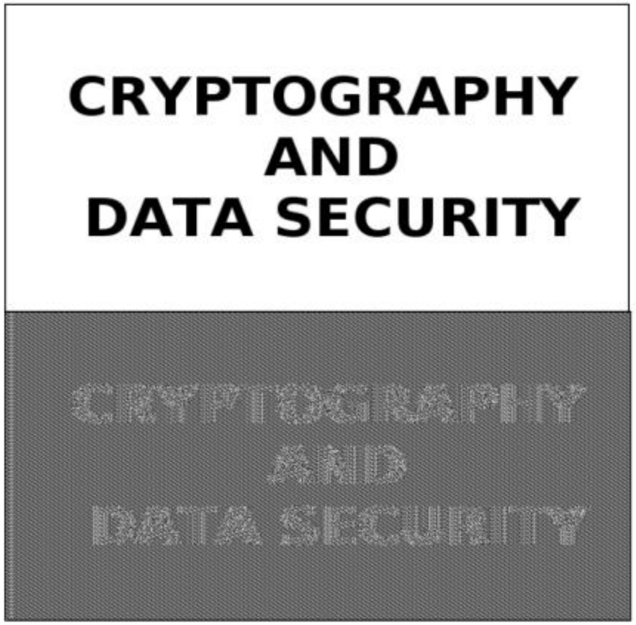
  - the encryption key between the two banks does not change too frequently

Substitution Attack on ECB (2/2)

- The attacker sends $1.00 transfers from his account at bank A to his account at bank B repeatedly
  - He can check for ciphertext blocks that repeat, and he stores blocks 1,3 and 4 of these transfers
- He now simply replaces block 4 of other transfers with the block 4 that he stored before
  - all transfers from some account of bank A to some account of bank B are redirected to go into the attacker's B account!

Example of encrypting bitmaps in ECB mode

› Identical plaintexts are mapped to identical ciphertexts



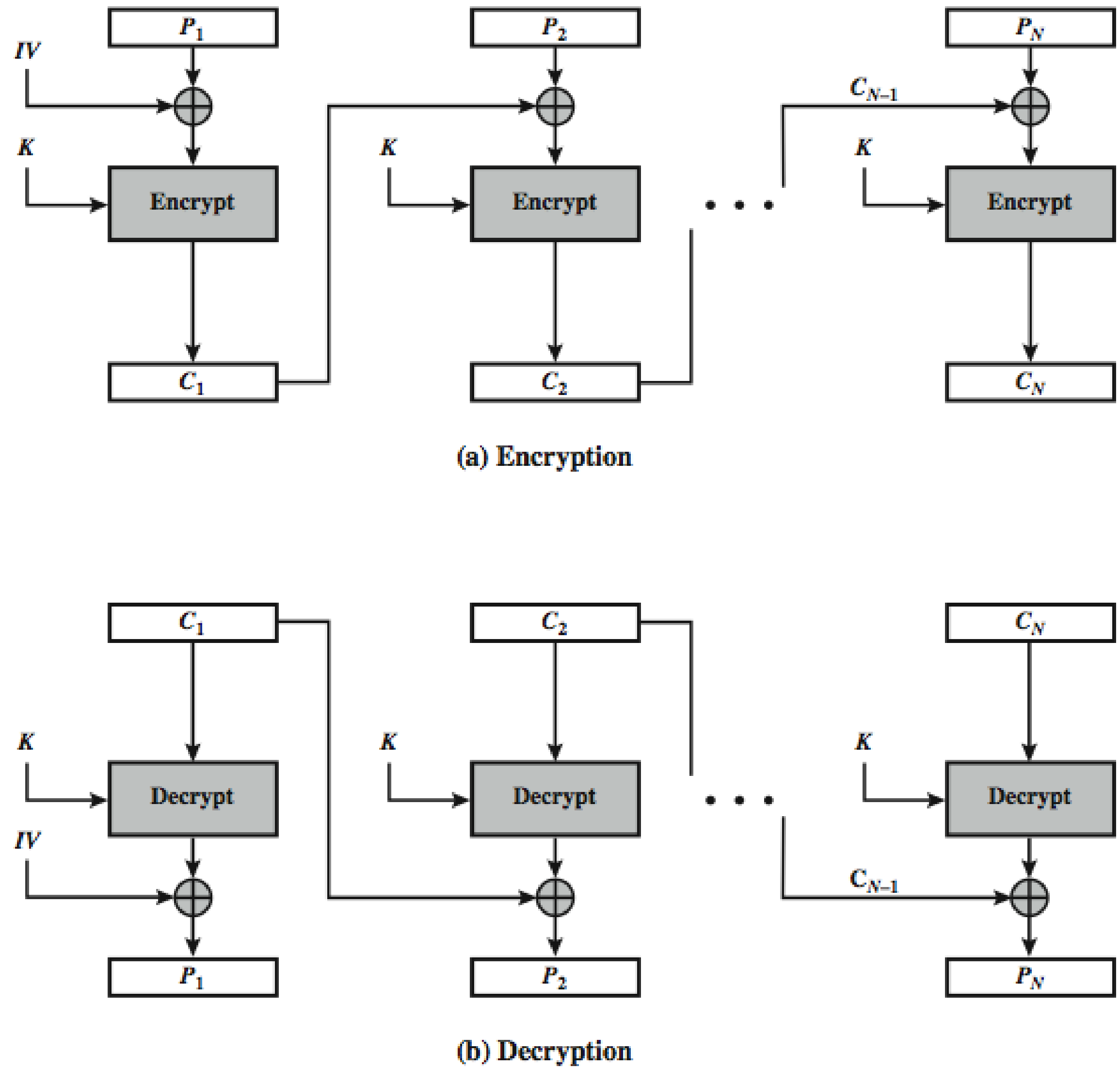
› Statistical properties in the plaintext are preserved in the ciphertext

Cipher Block Chaining mode (CBC)

- Two main ideas behind the CBC mode:
  - The encryption of all blocks are chained together
    - ciphertext y_i depends not only on block x_i but on all previous plaintext blocks as well
  - Initialization vector (IV)
    - Randomizes the encryption
      - Same plaintext can get different ciphertext
    - Can be public

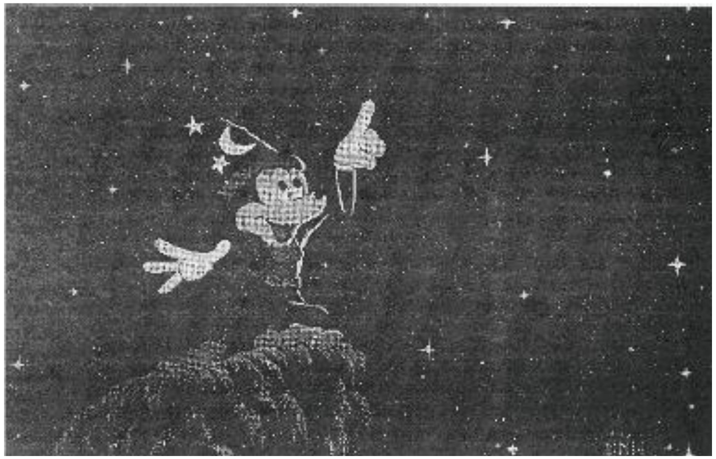
Encryption (first block):	$y_1 = e_k(x_1 \oplus \mathrm{IV})$
Encryption (general block):	$y_i = e_k(x_i \oplus y_{i-1}),\ \ i \geq 2$
Decryption (first block):	$x_1 = e_k^{-1}(y_1) \oplus \mathrm{IV}$
Decryption (general block) :	$x_i = e_k^{-1}(y_i) \oplus y_{i-1},\ \ i \geq 2$

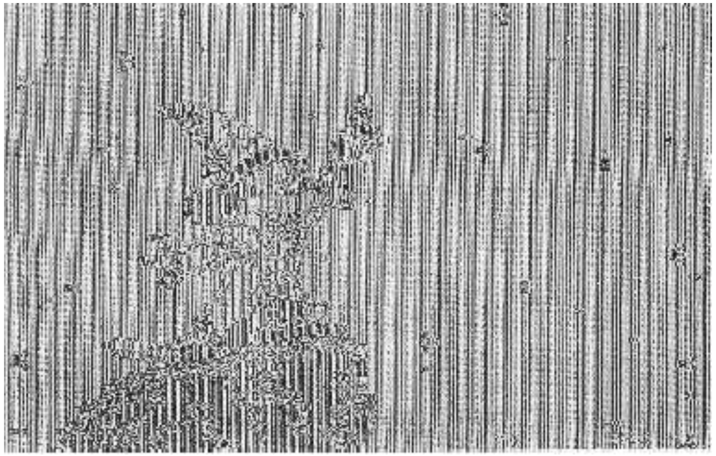
CBC Mode

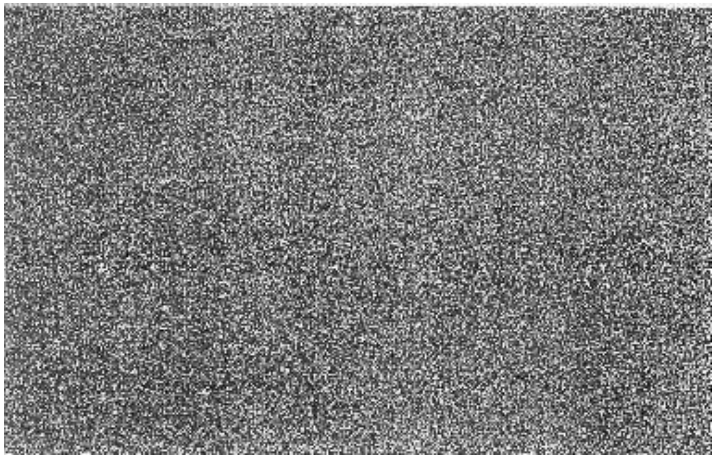

P1
P2
PN
IV
K
CN-1
Encrypt
Encrypt
Encrypt
C1
C2
CN
(a) Encryption
C1
C2
CN
K
K
K
Decrypt
Decrypt
Decrypt
IV
CN-1
P1
P2
PN
(b) Decryption

ECB vs CBC

- Plaintext
  - Original →
- Ciphertext
  - Encrypted by ECB ↘
  - Encrypted by CBC ↓



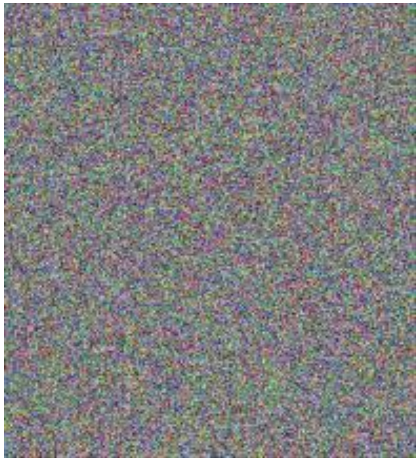


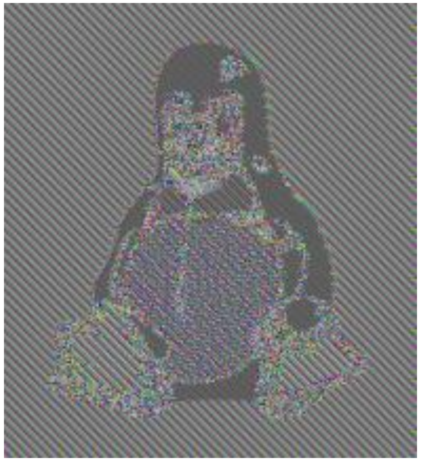


ECB vs CBC

› Plaintext：
  - Original →

› Ciphertext：
  - Encrypted by ECB ↘
  - Encrypted by CBC ↓





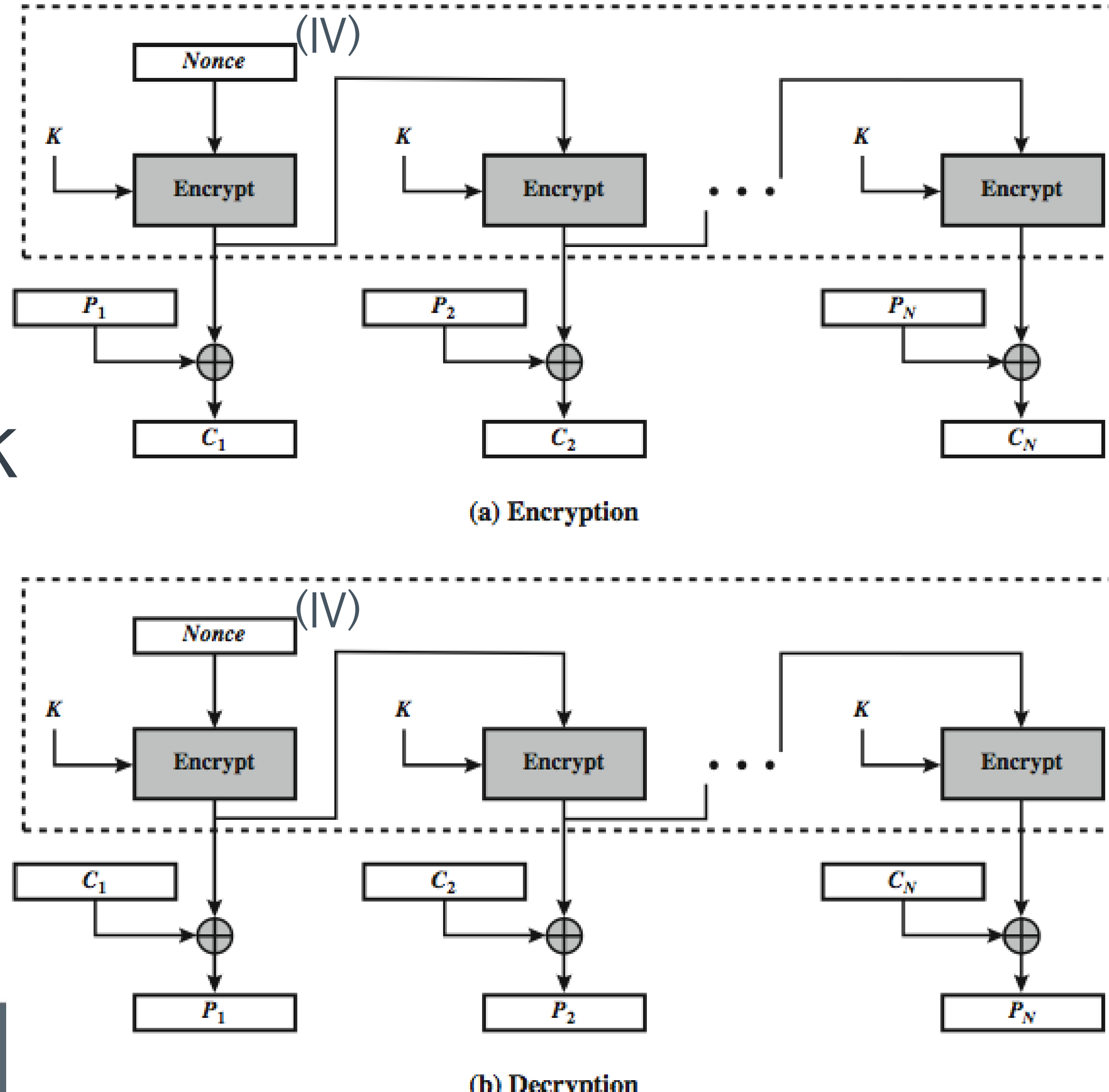
Substitution Attack on CBC (1/2)

- Suppose the last example (electronic bank transfer)
  - If the IV is properly chosen for every wire transfer, the attack will not work at all
  - If the IV is kept the same for several transfers, the attacker would recognize the transfers from his account at bank A to back B

Substitution Attack on CBC (2/2)

- If we choose a new IV every time we encrypt, the CBC mode becomes a probabilistic encryption scheme
  - two encryptions of the same plaintext look entirely different
- It is not needed to keep the IV secret!
- Typically, the IV should be a non-secret nonce (value used only once)

Output Feedback mode (OFB)



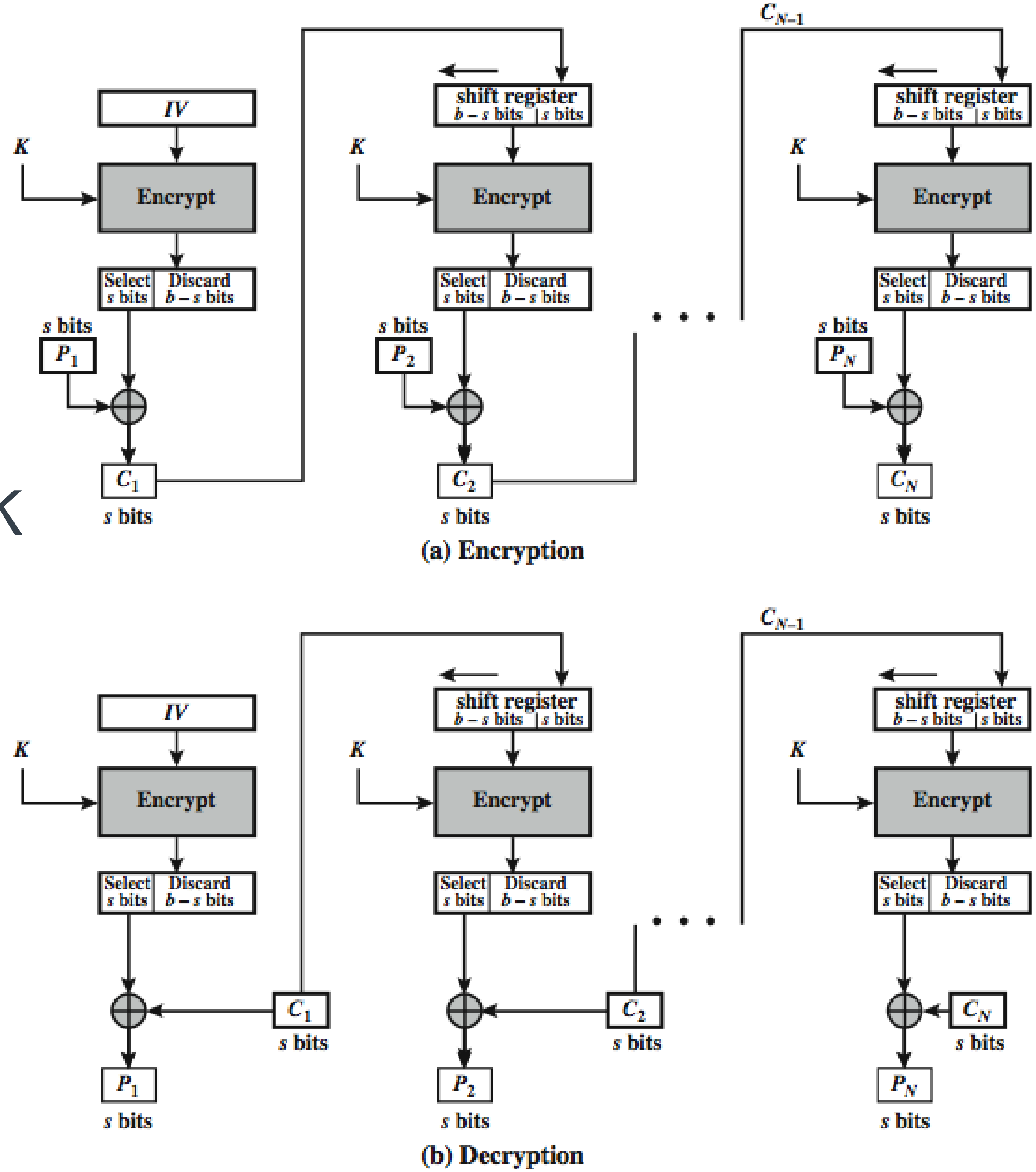
Output Feedback mode (OFB)

- It is used to build a synchronous stream cipher from a block cipher.
- The key stream is not generated bitwise but instead in a blockwise fashion
- The output of the cipher gives us key stream bits S_i with which we can encrypt plaintext bits using the XOR operation

Output Feedback mode (OFB)

- Advantages
  - Encryption and decryption can have immediacy
  - If there is a bit error in the plaintext => ciphertext only one bit error
  - All blocks are only used to encrypt (can reduce the cost of the hardware device)
- Disadvantages
  - Keystream generated in cycles

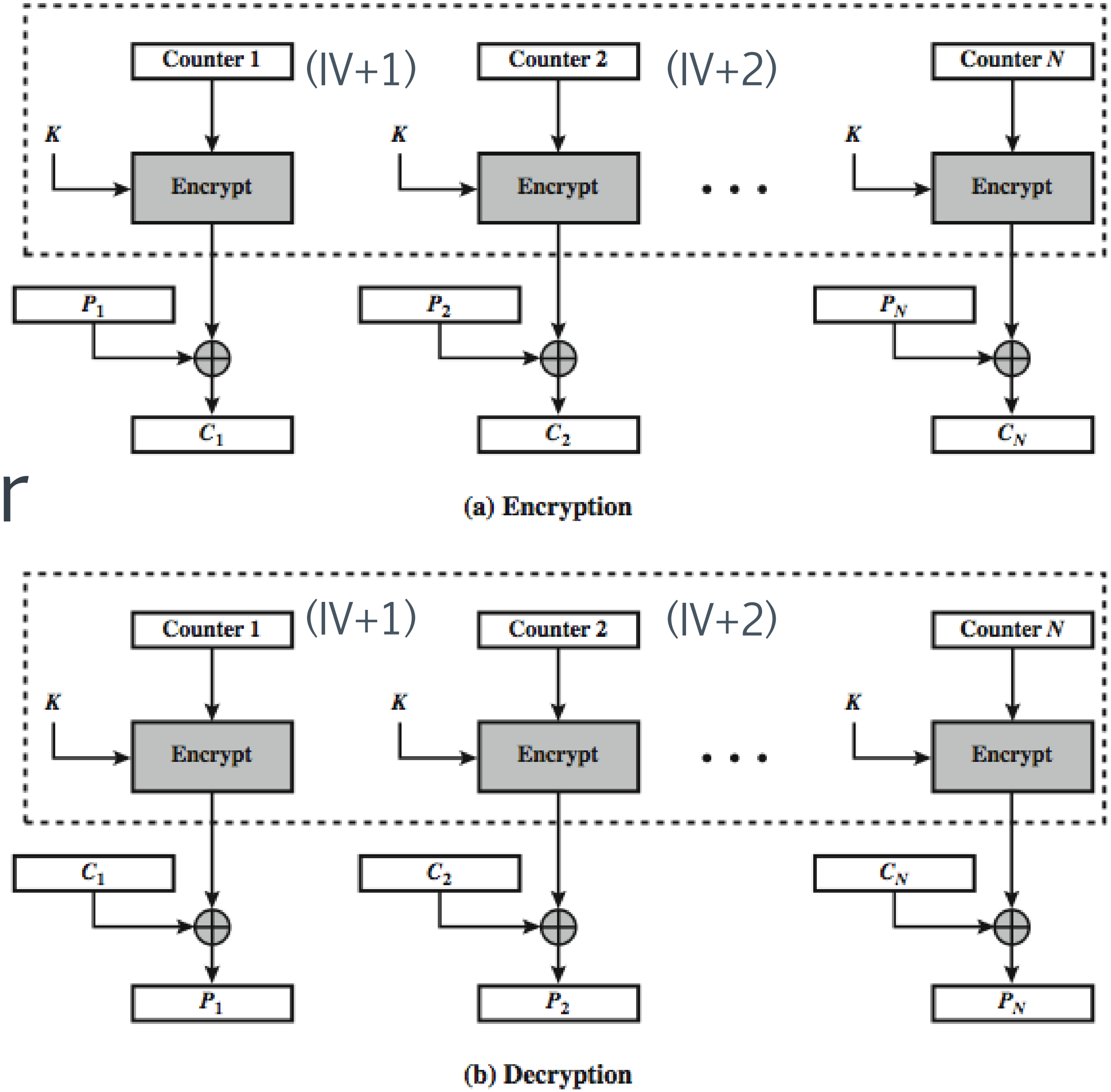
Cipher Feedback mode (CFB)



Cipher Feedback mode (CFB)

- Ciphertext Feedback Mode
  - It uses a block cipher as a building block for an asynchronous **stream cipher** (similar to the OFB mode)
  - The key stream S_i is generated in a blockwise fashion and is also a function of the ciphertext
- As a result of the use of an IV, the CFB encryption is also nondeterministic
- It can be used in situations where short plaintext blocks are to be encrypted

Counter mode (CTR)



Counter mode (CTR)

- It uses a block cipher as a stream cipher (like the OFB and CFB modes)
- The key stream is computed in a blockwise fashion
- The input to the block cipher is a counter which assumes a different value every time the block cipher computes a new key stream block
- Unlike CFB and OFB modes, the CTR mode can be parallelized since the 2nd encryption can begin before the 1st one has finished

Exhaustive Key Search Revisited (1/4)

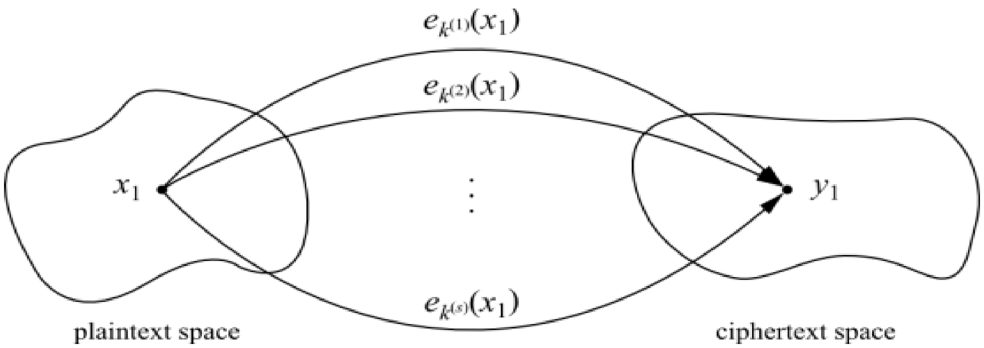
- A simple exhaustive search for a DES key knowing one pair (x_1, y_1):

$$DES_k^{(i)}(x_1) \stackrel{?}{=} y_1, i = 0,1, \dots, 2^{56} - 1$$

- However, for most other block ciphers a key search is somewhat more complicated
- A brute-force attack can produce **false positive results /假正面結果** (block<key)
  - keys k_i that are found are not the one used for the encryption

Exhaustive Key Search Revisited (2/4)

- The likelihood of this is related to the relative size of the key space and the plaintext space
- A brute-force attack is still possible, but several pairs of plaintext–ciphertext are needed



Exhaustive Key Search Revisited (3/4)

- Assume a cipher with a block width of 64 bit and a key size of 80 bit
- If we encrypt x_1 under all possible 2^{80} keys, we obtain 2^{80} ciphertexts
  - However, there exist only 2^{64} different ones
- If we run through all keys for a given plaintext–ciphertext pair, we find on average $\frac{2^{80}}{2^{64}} = 2^{16}$ keys that perform the mapping $e_k(x_1) = y_1$

Exhaustive Key Search Revisited (4/4)

> Given a block cipher with a key length of k bits and block size of n bits, as well as t plaintext-ciphertext pairs $(x_1, y_1), \ldots, (x_t, y_t)$, the expected number of *false* keys which encrypt all plaintexts to the corresponding ciphertexts is:
>
> $$2^{k-tn}$$ 期望值

- Assuming two plaintext-ciphertext pairs, the likelihood is

$$2^{80-2*64} = 2^{-48}$$

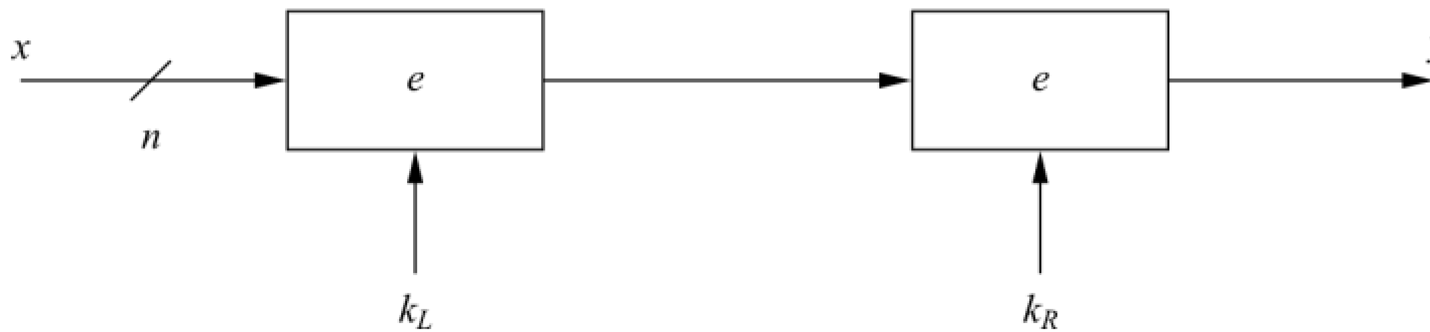
  - for almost all practical purposes two plaintext-ciphertext pairs are sufficient

Increasing the Security of Block Ciphers

- In some situations we wish to increase the security of block ciphers, e.g., if a cipher such as DES is available in hardware or software for legacy reasons in a given application
- Two approaches are possible
  - Multiple encryption/多重加密
    - theoretically much more secure, but sometimes in practice increases the security very little
  - Key whitening/增加key的長度

Double Encryption

- A plaintext x is first encrypted with a key k_L, and the resulting ciphertext is encrypted again using a second key k_R



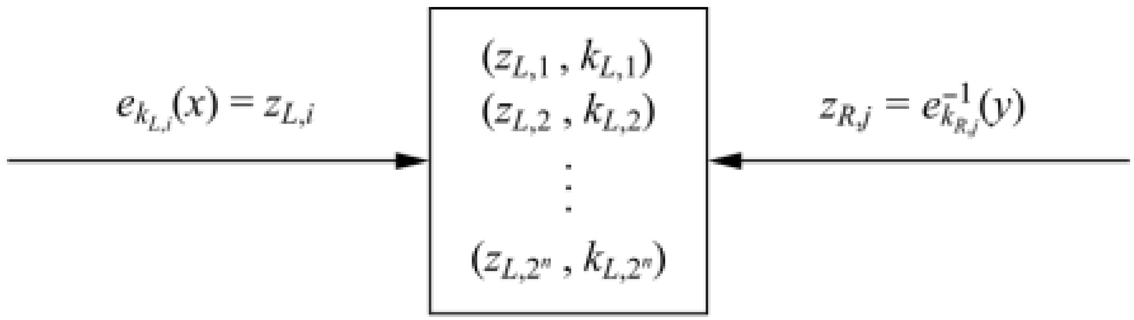
- Assuming a key length of k bits, an exhaustive key search would require $2^k \cdot 2^k = 2^{2k}$ encryptions or decryptions

Pigeonhole principle/鴿籠原理

› If n items are put into m pigeonholes with $n > m$, then at least one pigeonhole must contain more than one item.

Meet-in-the-Middle Attack (1/2)

- A Meet-in-the-Middle attack requires $2^k + 2^k = 2^{k+1}$ operations!



  - **Phase I:** for the given (x_1, y_1) the **left** encryption is brute-forced for all $k_{L,i}$, $i = 1, 2, \dots, 2^k$ and a lookup table with 2^k entry (each $n + k$ bits wide) is computed
    - the lookup table should be ordered by the result of the encryption ($z_{L,i}$)

Meet-in-the-Middle Attack (2/2)

- **Phase II:** the **right** encryption is brute-forced (using decryption) and for each $z_{R,i}$ it is checked whether $z_{R,i}$ is equal to any $z_{L,i}$ value in the table of the first phase

› Pigeonhole principle/鴿籠原理

- If n items are put into m pigeonholes with $n > m$, then at least one pigeonhole must contain more than one item.

› Computational Complexity

- Number of encryptions and decryptions = $2^k + 2^k = 2^{k+1}$
- Number of storage location = 2^k

› Double encryption is not much more secure then single encryption!

Triple Encryption (1/2)

- The encryption of a block three times

$$y = e_{k3}(e_{k2}(e_{k1}(x)))$$

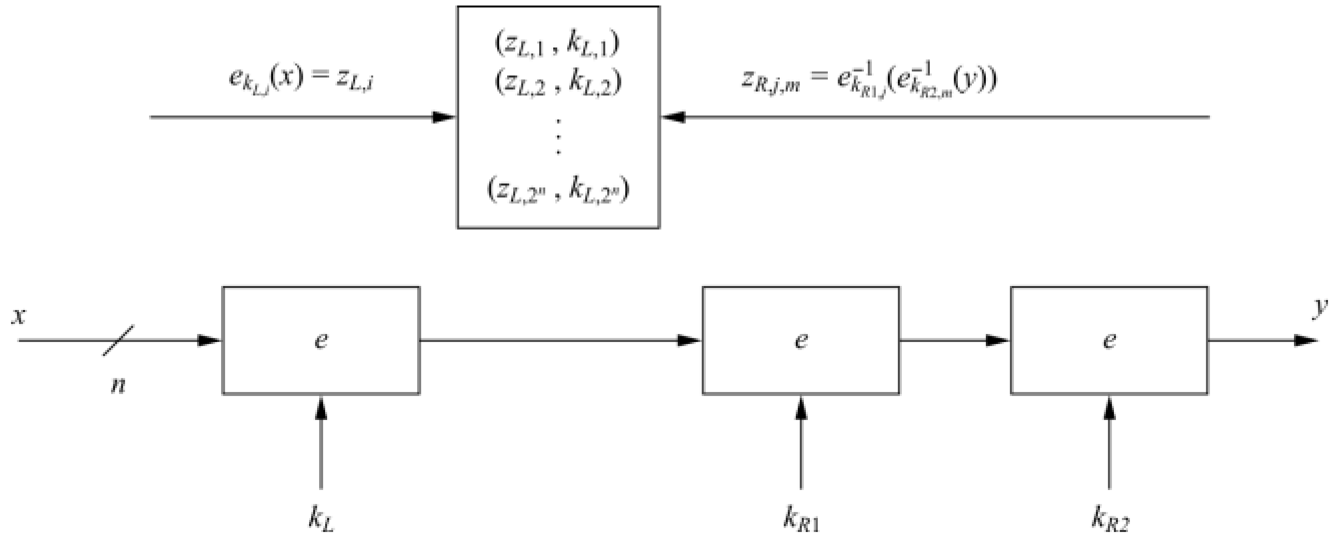
- In practice a variant scheme is often used EDE (encryption-decryption-encryption)

$$y = e_{k3}(e_{k2}^{-1}(e_{k1}(x)))$$

- Advantage: choosing $k1 = k2 = k3$ performs single DES encryption

Triple Encryption (2/2)

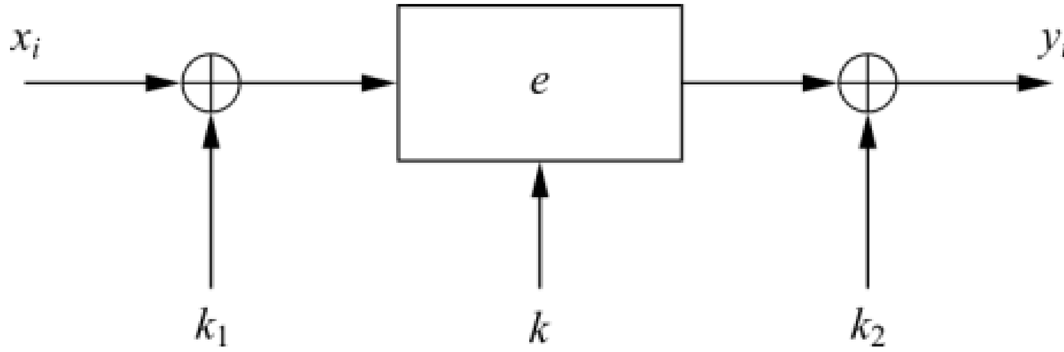
- Still we can perform a meet-in-the middle attack, and it reduces the effective key length of triple encryption from $3K$ to $2K$!
  - The attacker must run 2^{112} tests in the case of 3DES



- Triple encryption effectively doubles the key length

Key Whitening (1/2)

- Makes block ciphers such as DES much more resistant against brute-force attacks
- n addition to the regular cipher key k, two whitening keys k_1 and k_2 are used to XOR-mask the plaintext and ciphertext



Key Whitening (2/2)

- It does not strengthen block ciphers against most analytical attacks such as linear and differential cryptanalysis
- It is not a "cure" for inherently weak ciphers
- The additional computational load is negligible
- Its main application is ciphers that are relatively strong against analytical attacks but possess too short a key space especially DES
  - a variant of DES which uses key whitening is called DESX