Unit II. Symmetric Ciphers
Modes of Operation

Er. Kobid Karkee
Himalaya College of Engineering

Topics

- **Overview of Modes of Operation**
- ECB, CBC, CFB, OFB, CTR
- Notes and Remarks on each modes

Modes of Operation

- Block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks, with 56-bit key
- Need way to use in practise, given data usually have arbitrary amount of length to encrypt
  - Partition message into separate block for ciphering
- A **mode of operation** describes the process of encrypting each of these blocks **under a single key**
- Some modes may use randomized addition input value

Quick History

1981
- Early modes of operation: **ECB, CBC, CFB, OFB**
  - DES Modes of operation

    http://www.itl.nist.gov/fipspubs/fip81.htm

2001
- Revised and including **CTR** mode and AES
  - Recommendation for Block Cipher Modes of Operation

    http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

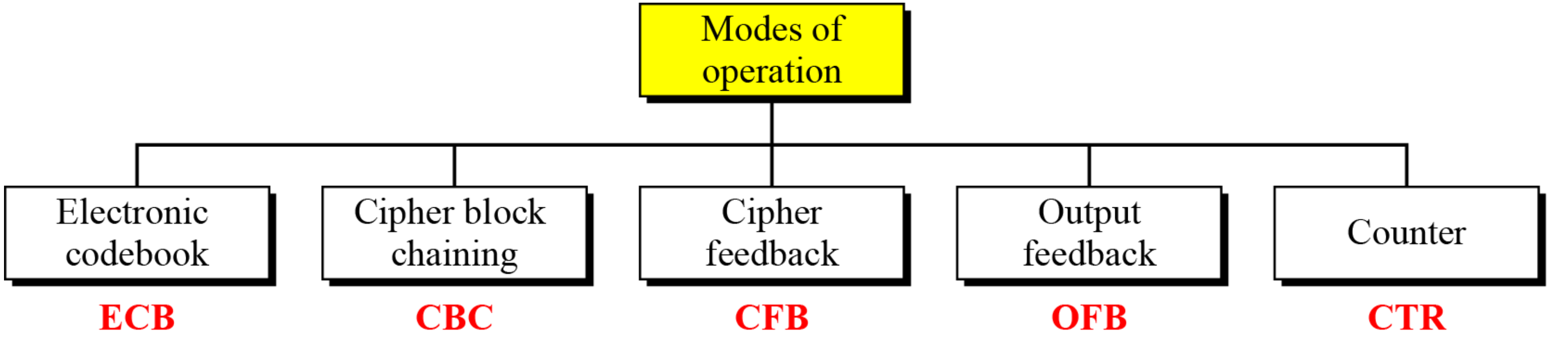
2010
- New Mode : **XTS-AES**
  - Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices

    http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf

Modes of operation are nowadays defined by a number of national and internationally recognized standards bodies such as ISO, IEEE, ANSI and IETF. The most influential source is the US NIST

Modes of Operation Taxonomy

- Current well-known modes of operation



Most Popular

More Technical Notes

- Initialize Vector (IV)
  - a block of bits to randomize the encryption and hence to produce distinct ciphertext
- Nonce : Number (used) Once
  - Random of psuedorandom number to ensure that past communications can not be reused in replay attacks
  - Some also refer to initialize vector as nonce
- Padding
  - final block may require a padding to fit a block size
  - Method
    - Add null Bytes
    - Add 0x80 and many 0x00
    - Add the *n* bytes with value *n*

Electronic Codebook Book (ECB)

- Message is broken into independent blocks which are encrypted
- Each block is a value which is substituted, like a codebook, hence name
- Each block is encoded independently of the other blocks

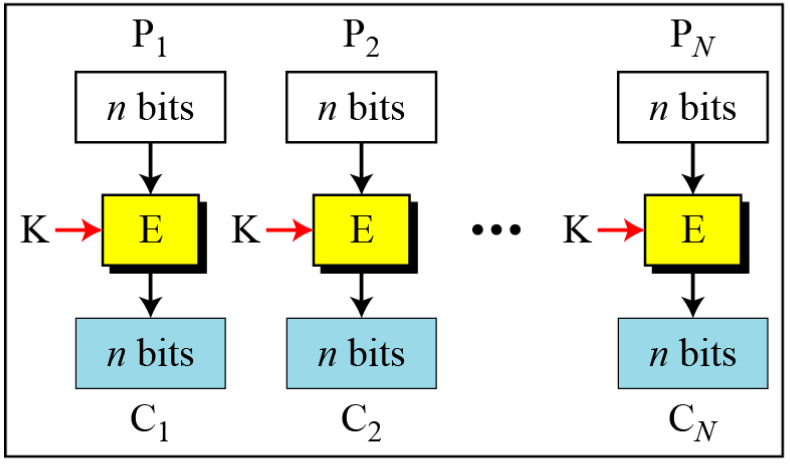
$$C_i = E_K(P_i)$$

- Uses: secure transmission of single values

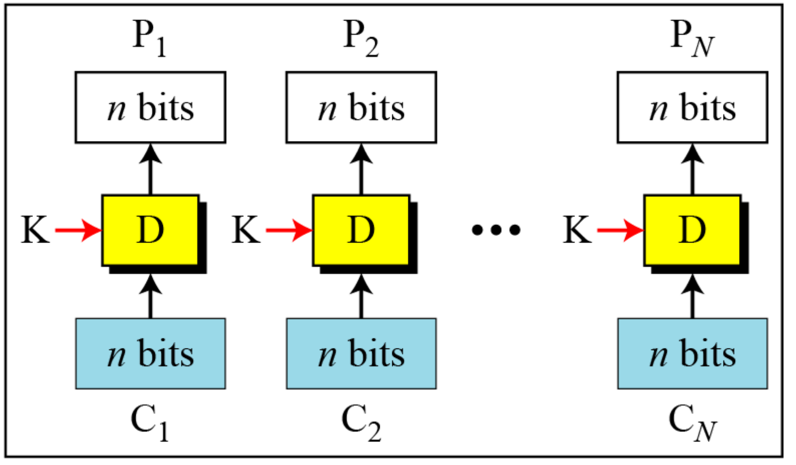
ECB Scheme

Encryption: $C_i = E_K(P_i)$ Decryption: $P_i = D_K(C_i)$

E: Encryption D: Decryption
P_i: Plaintext block *i* C_i: Ciphertext block *i*
K: Secret key



Encryption



Decryption

Remarks on ECB

- Strength: it's simple.
- Weakness:
  - Repetitive information contained in the plaintext may show in the ciphertext, if aligned with blocks.
  - If the same message is encrypted (with the same key) and sent twice, their ciphertext are the same.
- Typical application:
  - secure transmission of short pieces of information (e.g. a temporary encryption key)

Cipher Block Chaining (CBC)

- Solve security deficiencies in ECB
  - Repeated same plaintext block result different ciphertext block

- Each previous cipher blocks is chained to be input with current plaintext block, hence name

- Use Initial Vector (IV) to start process

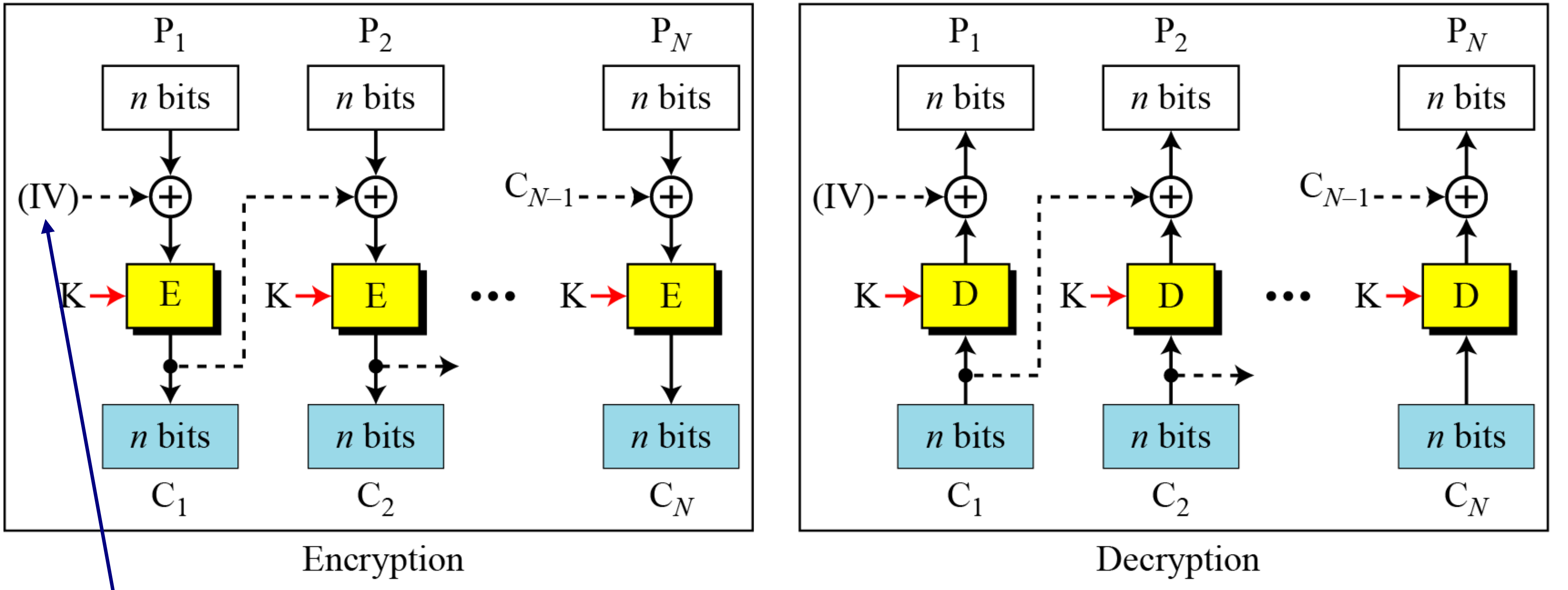
$$C_i = E_K(P_i \text{ XOR } C_{i-1})$$
$$C_0 = IV$$

- Uses: bulk data encryption, authentication

CBC scheme

E: Encryption
P_i: Plaintext block i
K: Secret key

D : Decryption
C_i : Ciphertext block i
IV: Initial vector (C_0)



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

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

Remarks on CBC

- The encryption of a block depends on the current and **all** blocks before it.
- So, repeated plaintext blocks are encrypted differently.
- Initialization Vector (IV)
  - May sent encrypted in ECB mode before the rest of ciphertext

Cipher FeedBack (CFB)

- Use Initial Vector to start process
-
- Encrypt previous ciphertext , then combined with the plaintext block using X-OR to produce the current ciphertext
- Cipher is fed back (hence name) to concatenate with the rest of IV
- Plaintext is treated as a stream of bits
  - Any number of bit (1, 8 or 64 or whatever) to be feed back (denoted CFB-1, CFB-8, CFB-64)
- Relation between plaintext and ciphertext

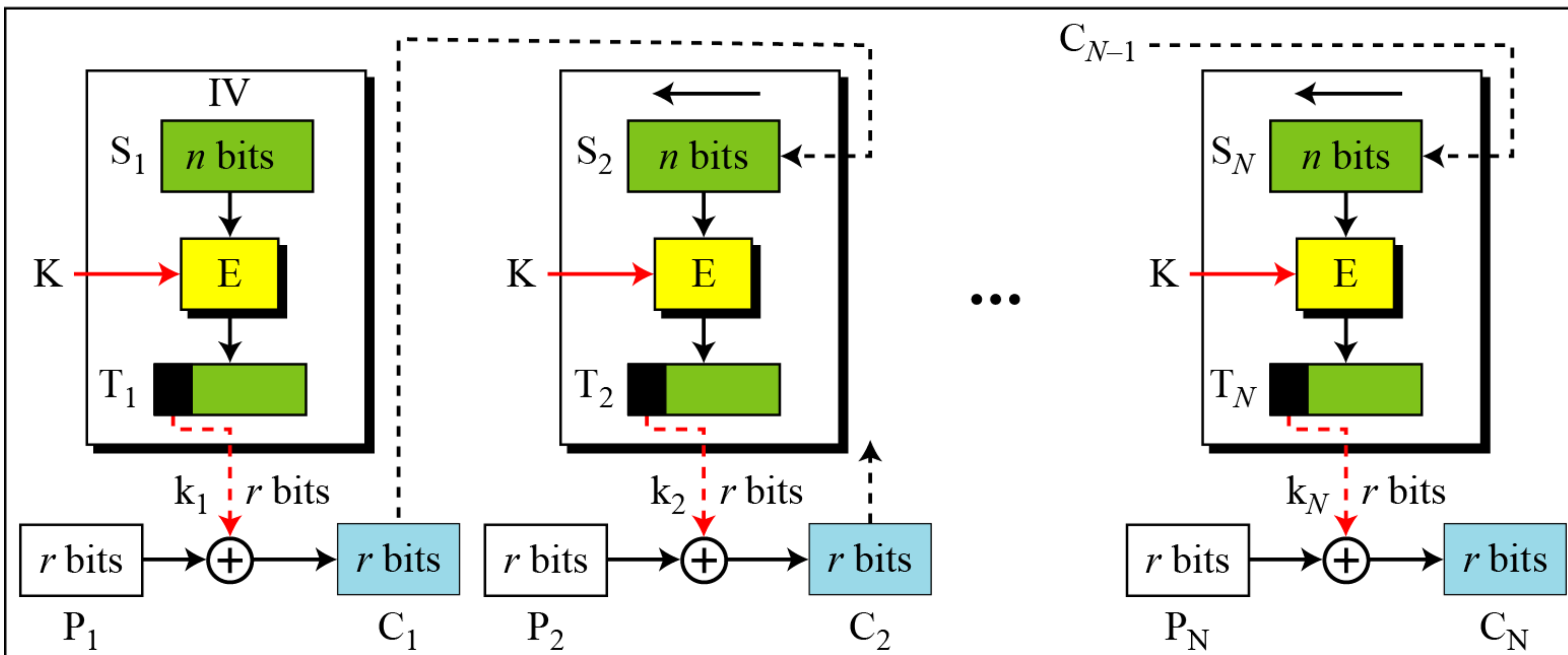
$$C_i = P_i \text{ XOR SelectLeft}(E_K(\text{ShiftLeft}(C_{i-1})))$$
$$C_0 = IV$$

- Uses: stream data encryption, authentication

CFB Scheme

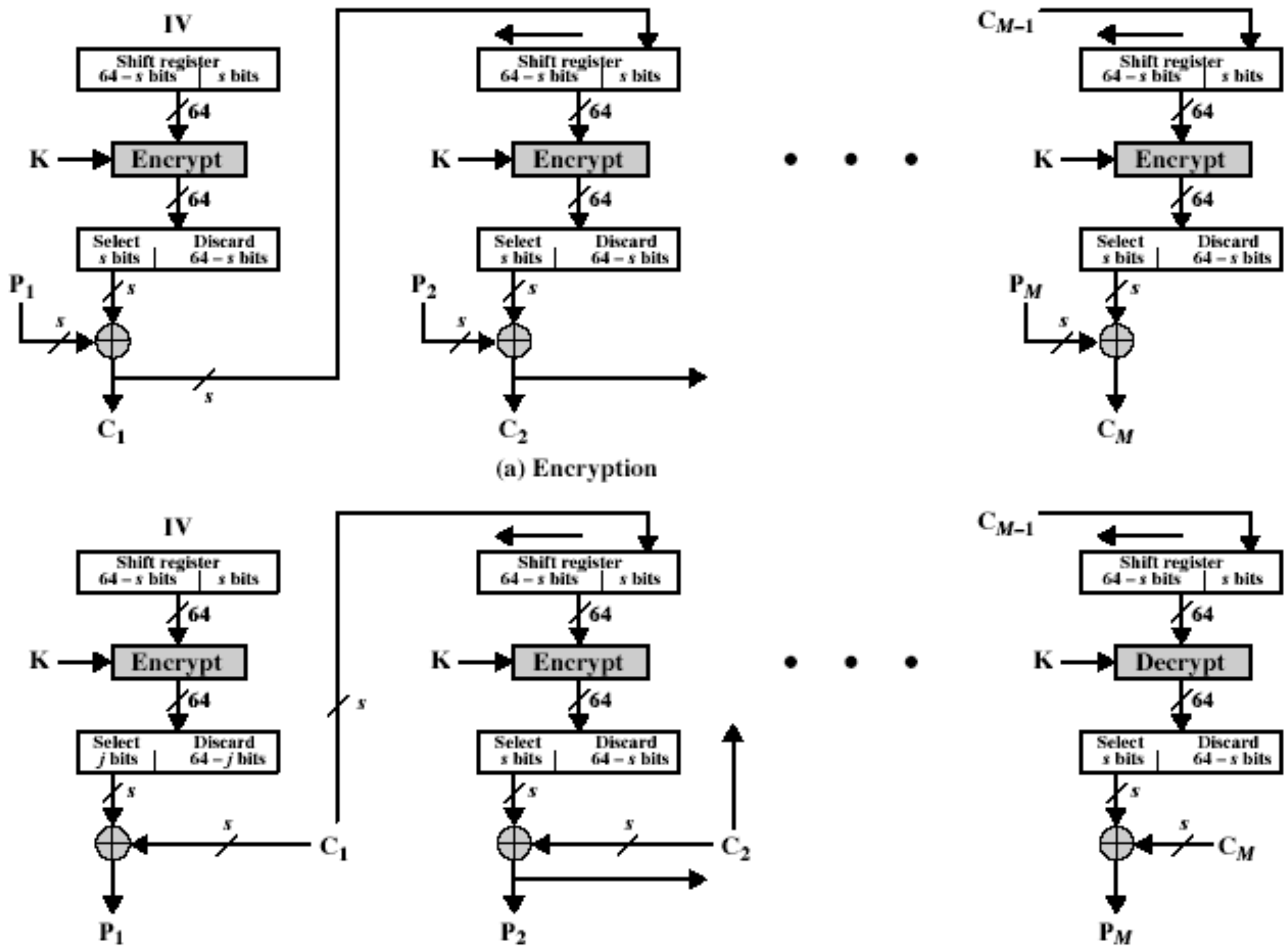
Encryption: $C_i = P_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$
Decryption: $P_i = C_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$

E : Encryption
P_i: Plaintext block i
K: Secret key
D : Decryption
C_i: Ciphertext block i
IV: Initial vector (S_1)
S_i: Shift register
T_i: Temporary register



Encryption

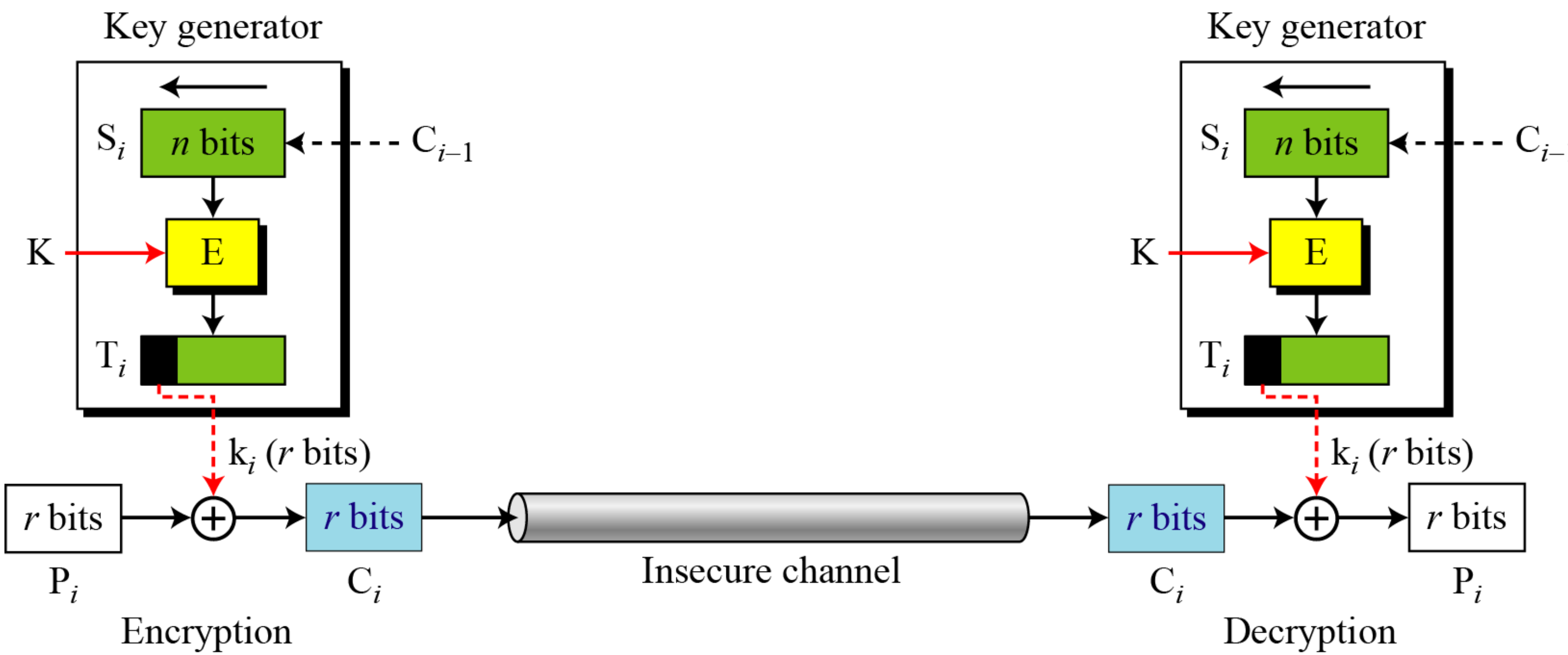
CFB Encryption/Decryption



(a) Encryption
(b) Decryption

CFB as a Stream Cipher

- In CFB mode, encipherment and decipherment use the encryption function of the underlying block cipher.



Remark on CFB

- The block cipher is used as a stream cipher.
  - enable to encrypt any number of bits e.g. single bits or single characters (bytes)
  - S=1 : bit stream cipher
  - S=8 : character stream cipher)
- A ciphertext segment depends on the current and all preceding plaintext segments.
- A corrupted ciphertext segment during transmission will affect the current and next several plaintext segments.

Output FeedBack (OFB)

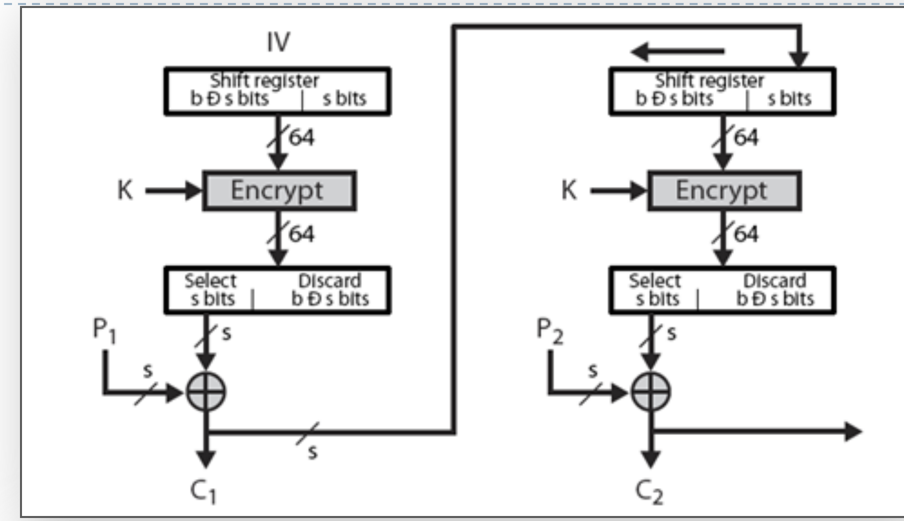
- Very similar to CFB
- But output of the encryption function output of cipher is fed back (hence name), instead of ciphertext
- Feedback is independent of message
- Relation between plaintext and ciphertext

$$C_i = P_i \text{ XOR } O_i$$
$$O_i = E_K(O_{i-1})$$
$$O_0 = IV$$

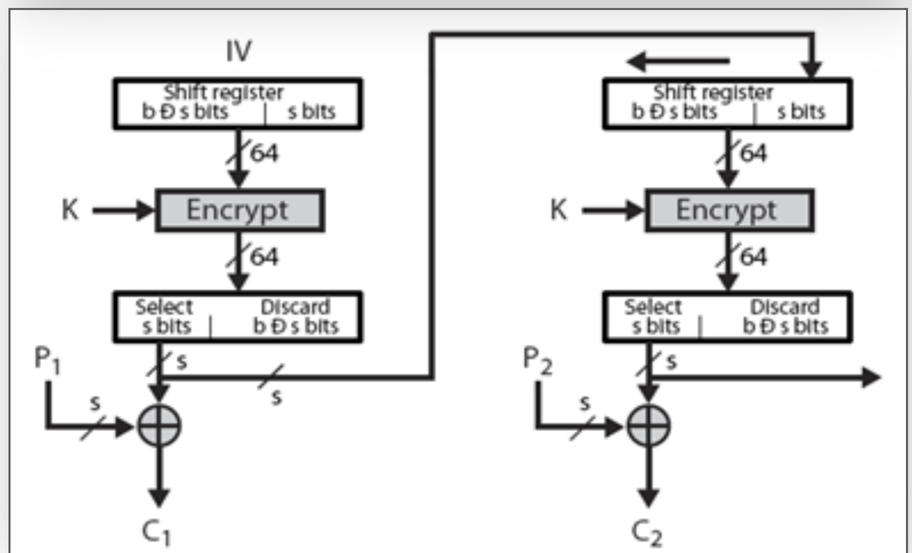
- Uses: stream encryption over noisy channels

CFB V.S. OFB

Cipher Feedback



Output Feedback

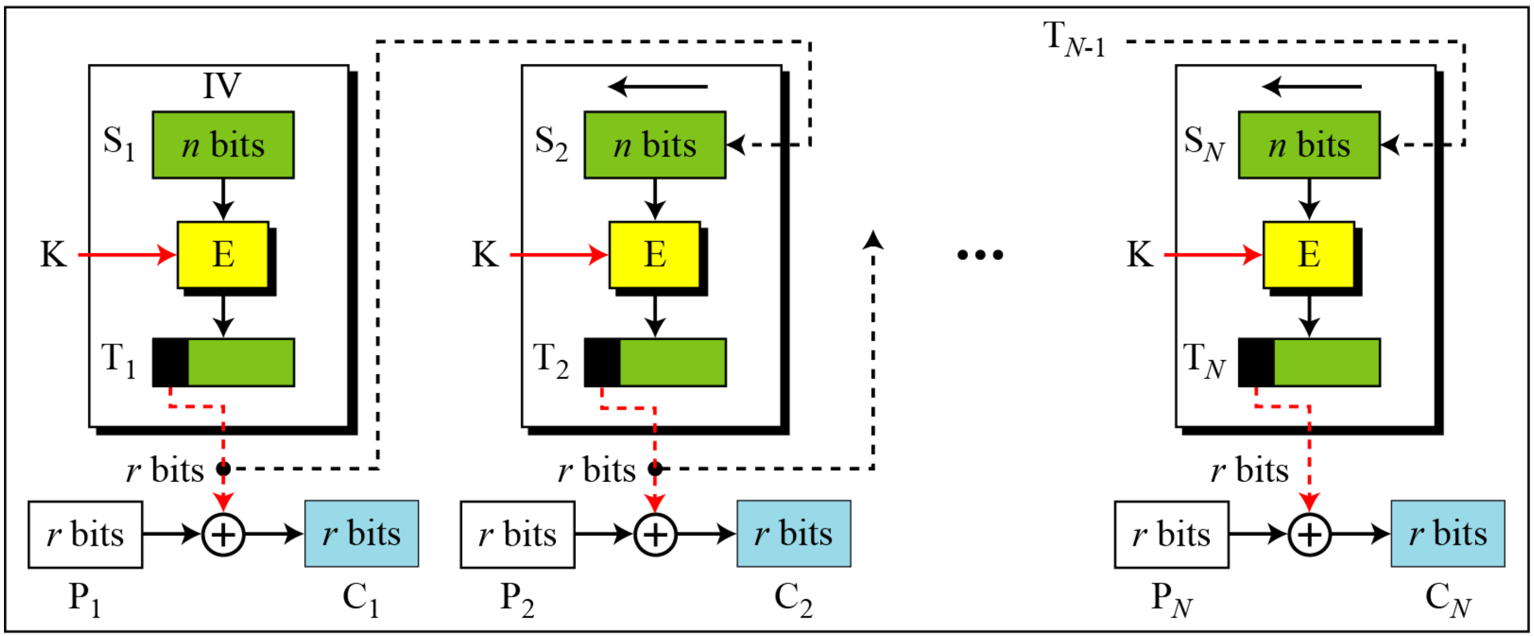


OFB Scheme

E : Encryption
P_i: Plaintext block i
K: Secret key

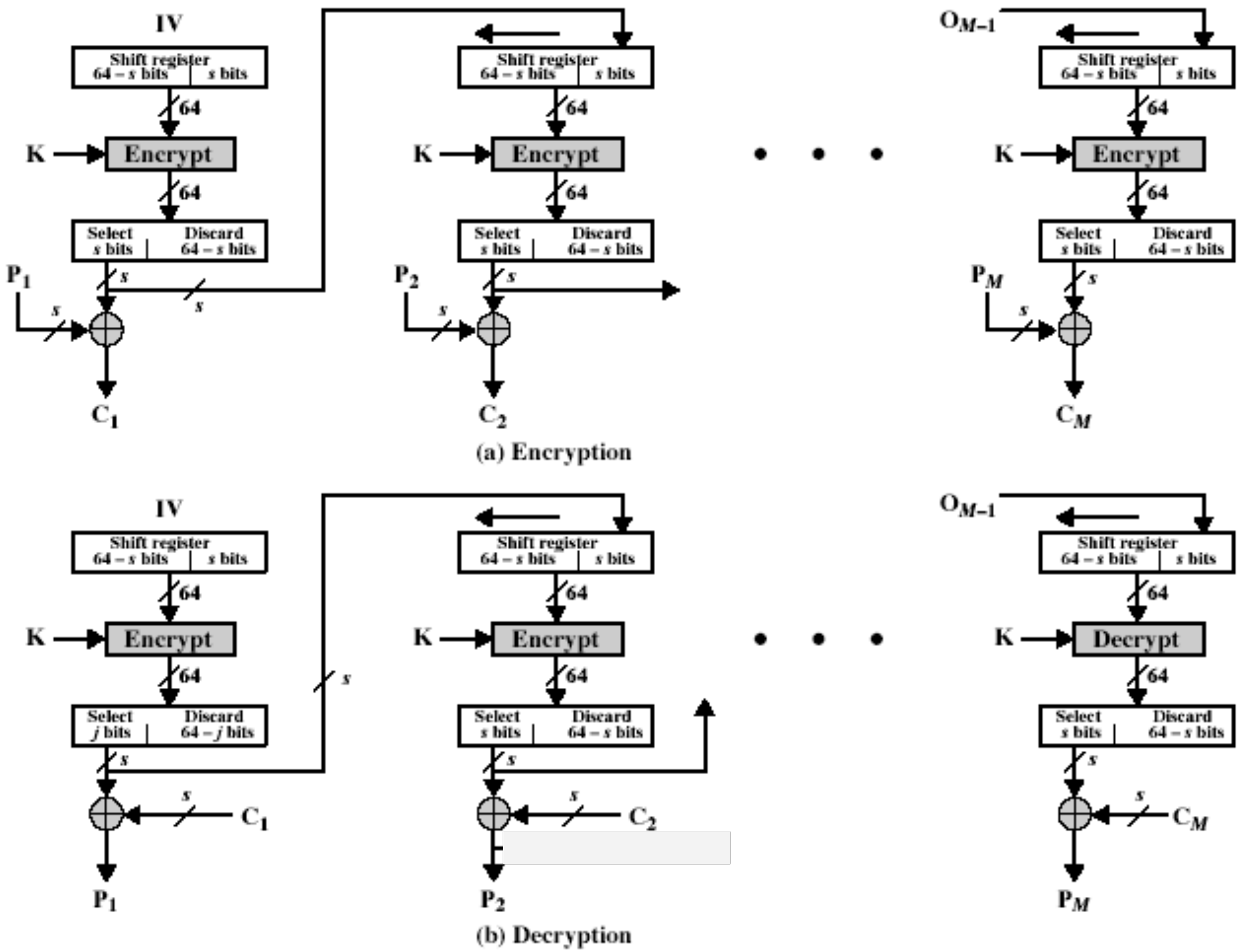
D : Decryption
C_i : Ciphertext block i
IV: Initial vector (S_1)

S_i: Shift register
T_i: Temporary register



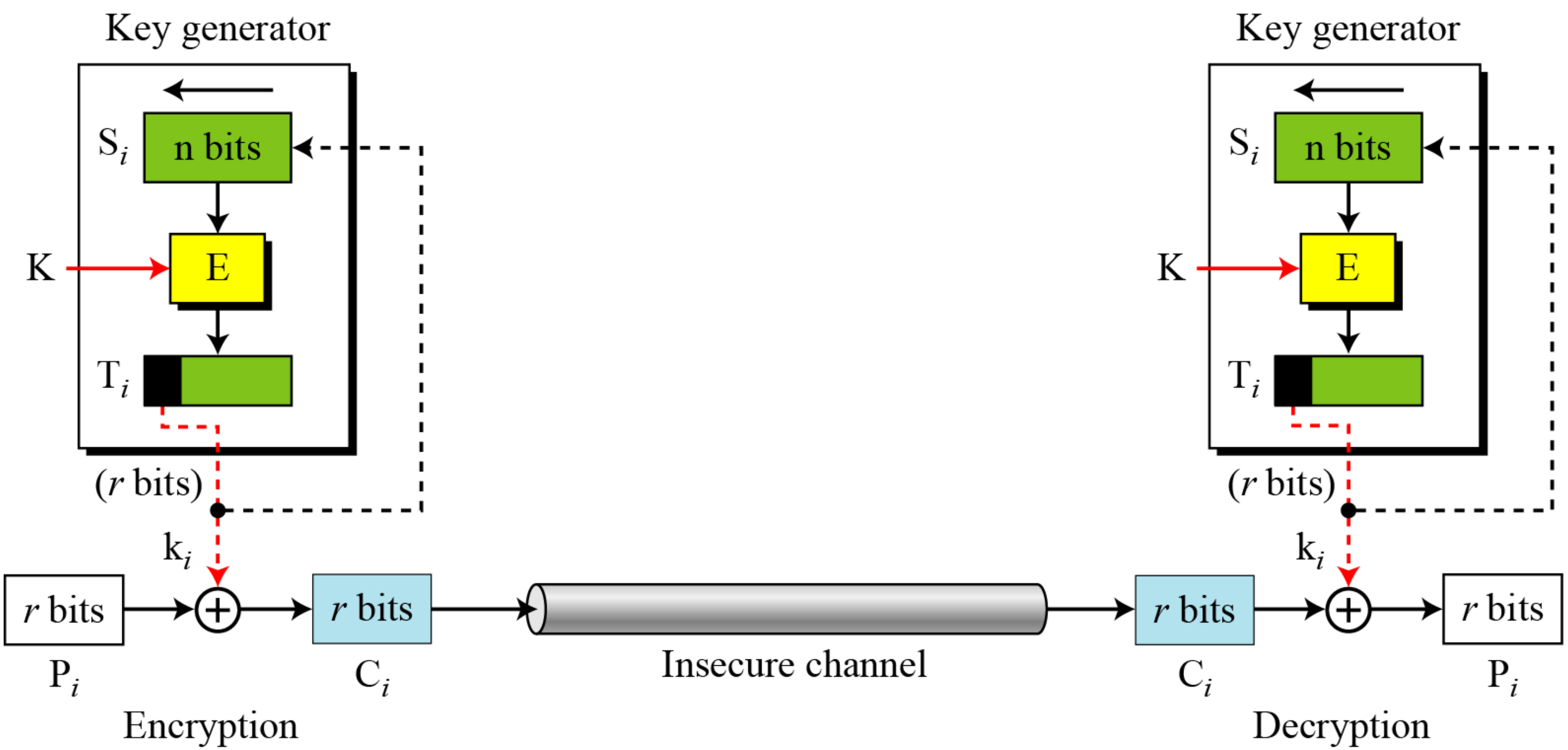
Encryption

OFB Encryption and Decryption



OFB as a Stream Cipher

- In OFB mode, encipherment and decipherment use the encryption function of the underlying block cipher.



Remarks on OFB

- Each bit in the ciphertext is independent of the previous bit or bits. This avoids error propagation

- Pre-compute of forward cipher is possible

- Security issue
  - when j^{th} plaintext is known, the j^{th} output of the forward cipher function will be known
  - Easily cover j^{th} plaintext block of other message with the same IV

- Require that the IV is a nonce

Counter (CTR)

- Encrypts counter value with the key rather than any feedback value (no feedback)

- Counter for each plaintext will be different
  - can be any function which produces a sequence which is guaranteed not to repeat for a long time

- Relation

$$C_i = P_i \text{ XOR } O_i$$

$$O_i = E_K(i)$$

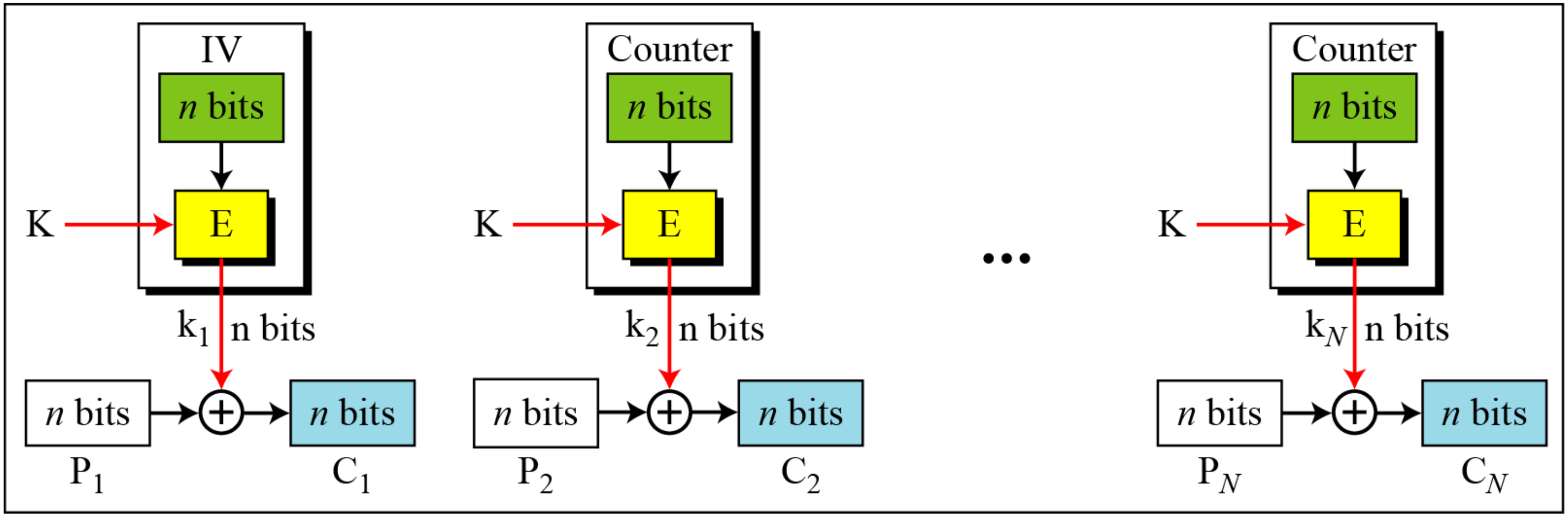
- Uses: high-speed network encryptions

CTR Scheme

E : Encryption
P_i : Plaintext block i
K : Secret key

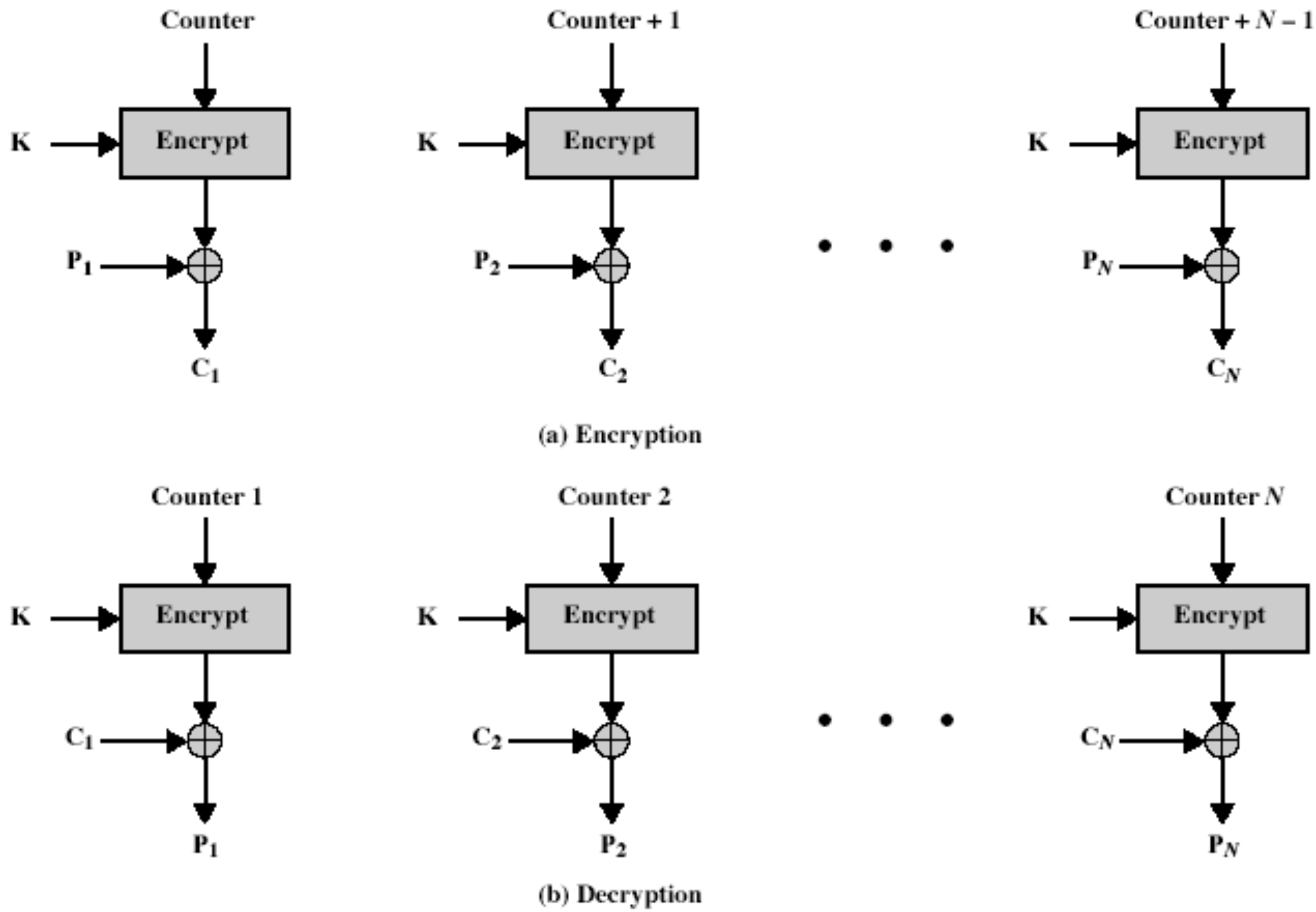
IV: Initialization vector
C_i : Ciphertext block i
k_i : Encryption key i

The counter is incremented for each block.

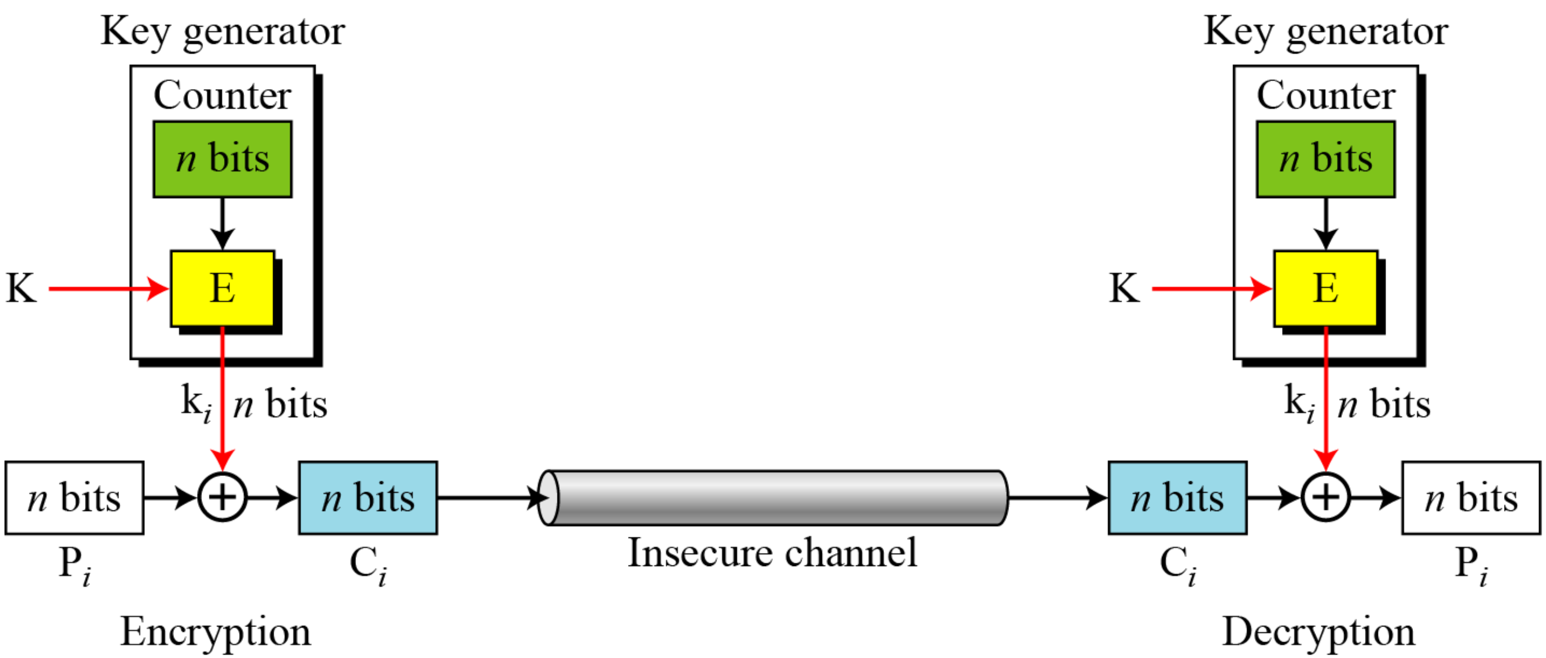


Encryption

CTR Encryption and Decryption



OFB as a Stream Cipher



Remark on CTR

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

Topics

- Overview of Modes of Operation
- EBC, CBC, CFB, OFB, CTR
- **Notes and Remarks on each modes**

Remark on each mode

- Basically two types:
  - block cipher
  - stream cipher
- CBC is an excellent block cipher
- CFB, OFB, and CTR are stream ciphers
- CTR is faster because simpler and it allows parallel processing

Modes and IV

- An IV has different security requirements than a key
- Generally, an IV will not be reused under the same key
- CBC and CFB
  - reusing an IV leaks some information about the first block of plaintext, and about any common prefix shared by the two messages
- OFB and CTR
  - reusing an IV completely destroys security

CBC and CTR comparison

CBC	CTR
Padding needed	No padding
No parallel processing	Parallel processing
Separate encryption and decryption functions	Encryption function alone is enough
Random IV or a nonce	Unique nonce
Nonce reuse leaks some information about initial plaintext block	Nonce reuse will leak information about the entire message

Comparison of Different Modes

Mode	Advantages	Disadvantages
Electronic Code Book (ECB)	• Simple • Fast • Support for parallel (encryption/decryption)	• Duplicate data in plaintext will be reflected in the ciphertext . • The plaintext can be operated by deleting or replacing the ciphertext. • If the ciphertext packet is damaged, it will affect the plaintext. • Can't resist replay attacks • Should not be used
Cipher Block Chaining (CBC)	• Support for parallel computing (decryption) • Ability to decrypt any ciphertext packet • Duplicate data in plaintext will not be reflected in the ciphertext	• Do not Support for parallel computing (Encryption) • Wrong blocks affect all following blocks
Cipher Feedback (CFB)	• No padding • Support for parallel computing (decryption) • Ability to decrypt any ciphertext packet • Can be prepared for encryption and decryption first	• Do not Support for parallel computing (Encryption) • Can't resist replay attacks • Wrong blocks affect all following blocks
Output Feedback (OFB)	• No padding • Can prepare encryption and decryption in advance • Encryption and decryption use the same structure • Bad blocks only affect the current block	• Do not Support for parallel computing • Malicious attacker can change some ciphertext damaged plaintext
Counter	• No padding • Can prepare encryption and decryption in advance • Support for parallel computing • Encryption and decryption use the same structure • Bad blocks only affect the current block	• Malicious attacker can change some ciphertext damaged plaintext

Table 6.1 Block Cipher Modes of Operation

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	• Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	Input is processed *s* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	• Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	• General-purpose block-oriented transmission • Useful for high-speed requirements

Comparison of Different Modes

Operation Mode	*Description*	*Type of Result*	*Data Unit Size*
ECB	Each n-bit block is encrypted independently with the same cipher key.	Block cipher	n
CBC	Same as ECB, but each block is first exclusive-ored with the previous ciphertext.	Block cipher	n
CFB	Each r-bit block is exclusive-ored with an r-bit key, which is part of previous cipher text	Stream cipher	$r \leq n$
OFB	Same as CFB, but the shift register is updated by the previous r-bit key.	Stream cipher	$r \leq n$
CTR	Same as OFB, but a counter is used instead of a shift register.	Stream cipher	n

Final Notes

- ECB, CBC, OFB, CFB, CTR, and XTS modes only provide confidentiality
- To ensure an encrypted message is not accidentally modified or maliciously tampered requires a separate Message Authentication Code (MAC)
- Several MAC schemes
  - HMAC, CMAC and GMAC
- But.. compositing a confidentiality mode with an authenticity mode could be difficult and error prone
- New modes combined confidentiality and data integrity into a single cryptographic primitive
  - CCM, GCM, CWC, EAX, IAPM and OCB