Stream Cipher, Block cipher, Modes of Operation

Dr. N.Gopika rani
CSE Department

S.NO	Block Cipher	Stream Cipher
1.	Block Cipher Converts the plain text into cipher text by taking plain text's block at a time.	Stream Cipher Converts the plain text into cipher text by taking 1 byte of plain text at a time.
2.	Block cipher uses either 64 bits or more than 64 bits.	While stream cipher uses 8 bits.
3.	The complexity of block cipher is simple.	While stream cipher is more complex.
4.	Block cipher Uses confusion as well as diffusion.	While stream cipher uses only confusion.
5.	In block cipher, reverse encrypted text is hard.	While in-stream cipher, reverse encrypted text is easy.
6.	The algorithm modes which are used in block cipher are ECB (Electronic Code Book) and CBC (Cipher Block Chaining).	The algorithm modes which are used in stream cipher are CFB (Cipher Feedback) and OFB (Output Feedback).
7.	Block cipher works on transposition techniques like rail-fence technique, columnar transposition technique, etc.	While stream cipher works on substitution techniques like Caesar cipher, polygram substitution cipher, etc.
8.	Block cipher is slow as compared to a stream cipher.	While stream cipher is fast in comparison to block cipher.

Difference Between Confusion and Diffusion

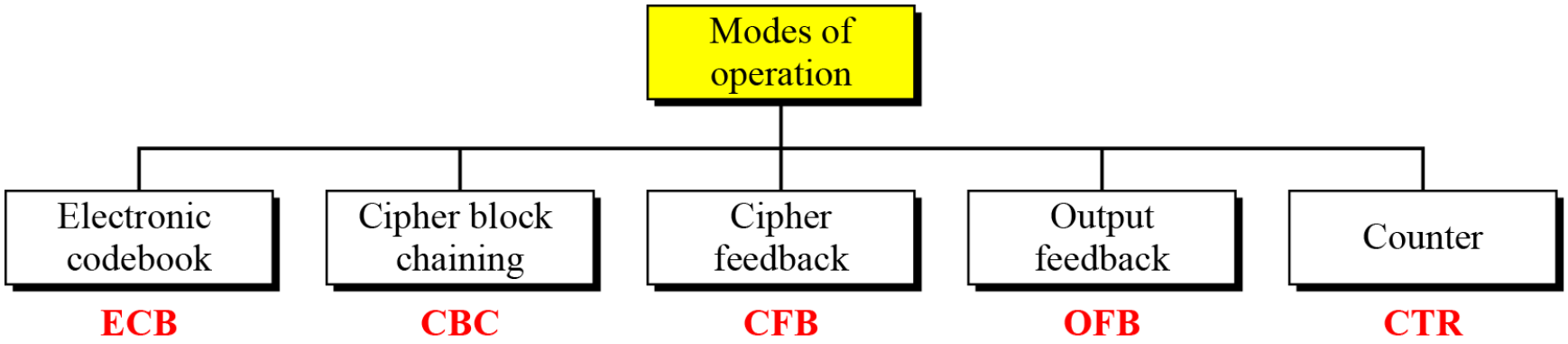
Confusion	Diffusion
Confusion obscures the relationship between the plaintext and ciphertext.	Diffusion spreads the plaintext statistics through the ciphertext.
A one-time pad relies entirely on confusion while a simple substitution cipher is another (weak) example of a confusion-only cryptosystem.	A double transposition is the classic example of a diffusion-only cryptosystem.
Confusion hides the relation between the ciphertext and key.	Diffusion hides the relation between the ciphertext and the plaintext.
If a single bit in the key is changed, most or all bits in the ciphertext will also be changed.	If a single symbol in the plaintext is changed, several or all symbol in the ciphertext will also be changed
In confusion, the relationship between the statistics of the ciphertext and the value of the encryption key is made complex. It is achieved by substitution.	In diffusion, the statistical structure of the plain text is dissipated into long-range statistics of the ciphertext This is achieved by permutation.

Topics

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

Modes of Operation Taxonomy

- Current well-known modes of operation



Mode 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

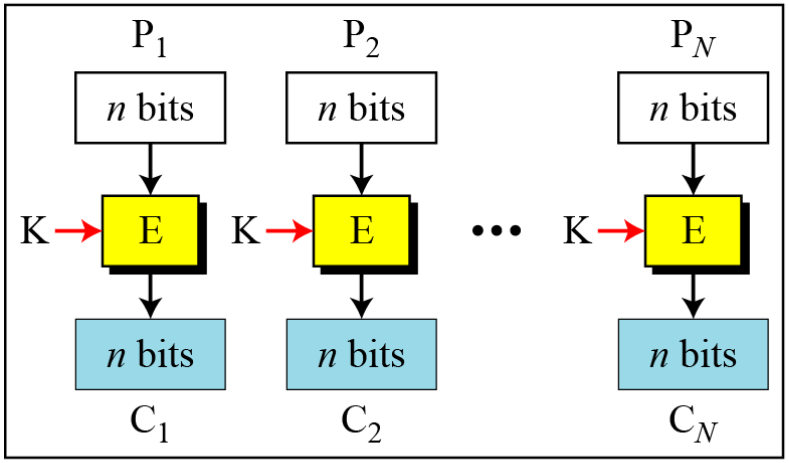
Topics

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

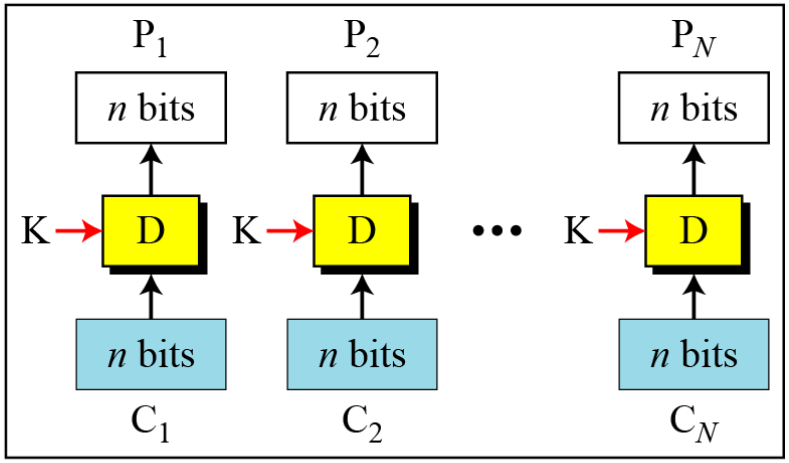
ECB Scheme

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

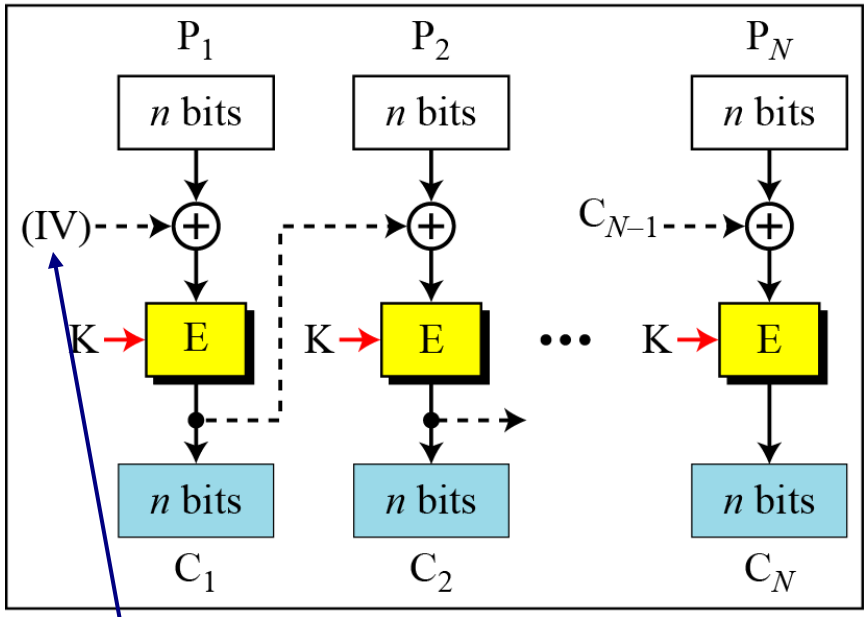
$$C_i = E_K (P_i \text{ XOR } C_{i-1})$$
$$C_0 = IV$$

- Uses: bulk data encryption, authentication

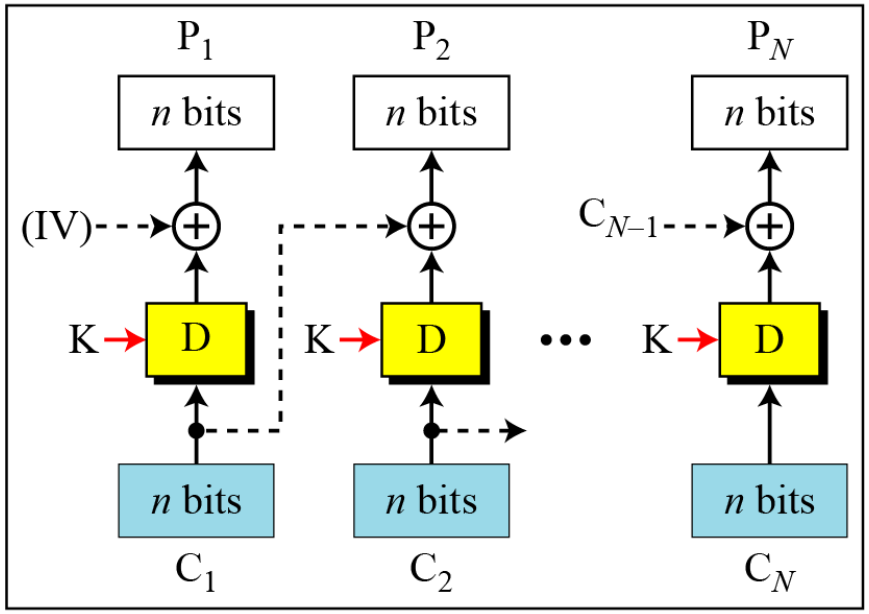
CBC scheme

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

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



Encryption



Decryption

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

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

- Uses: stream data encryption, authentication

CFB Scheme

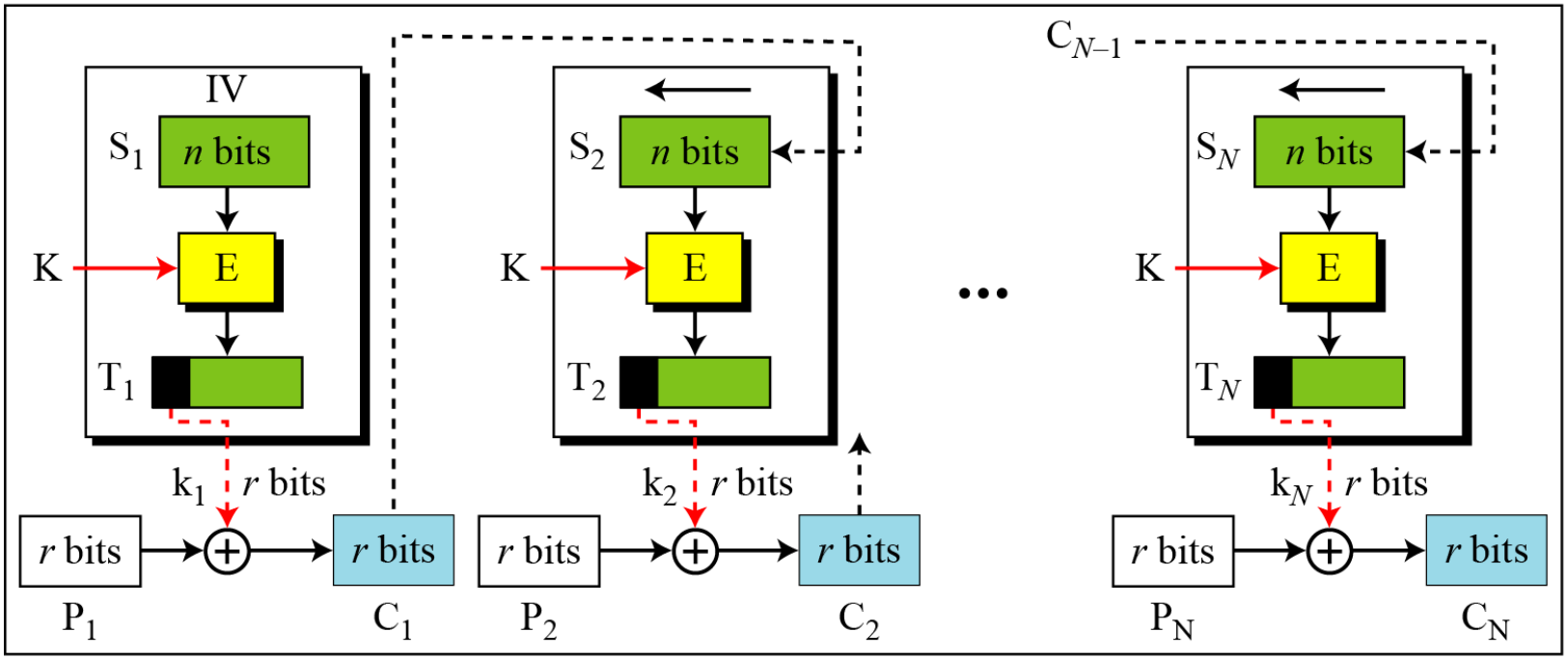
Encryption: $C_i = P_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$

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

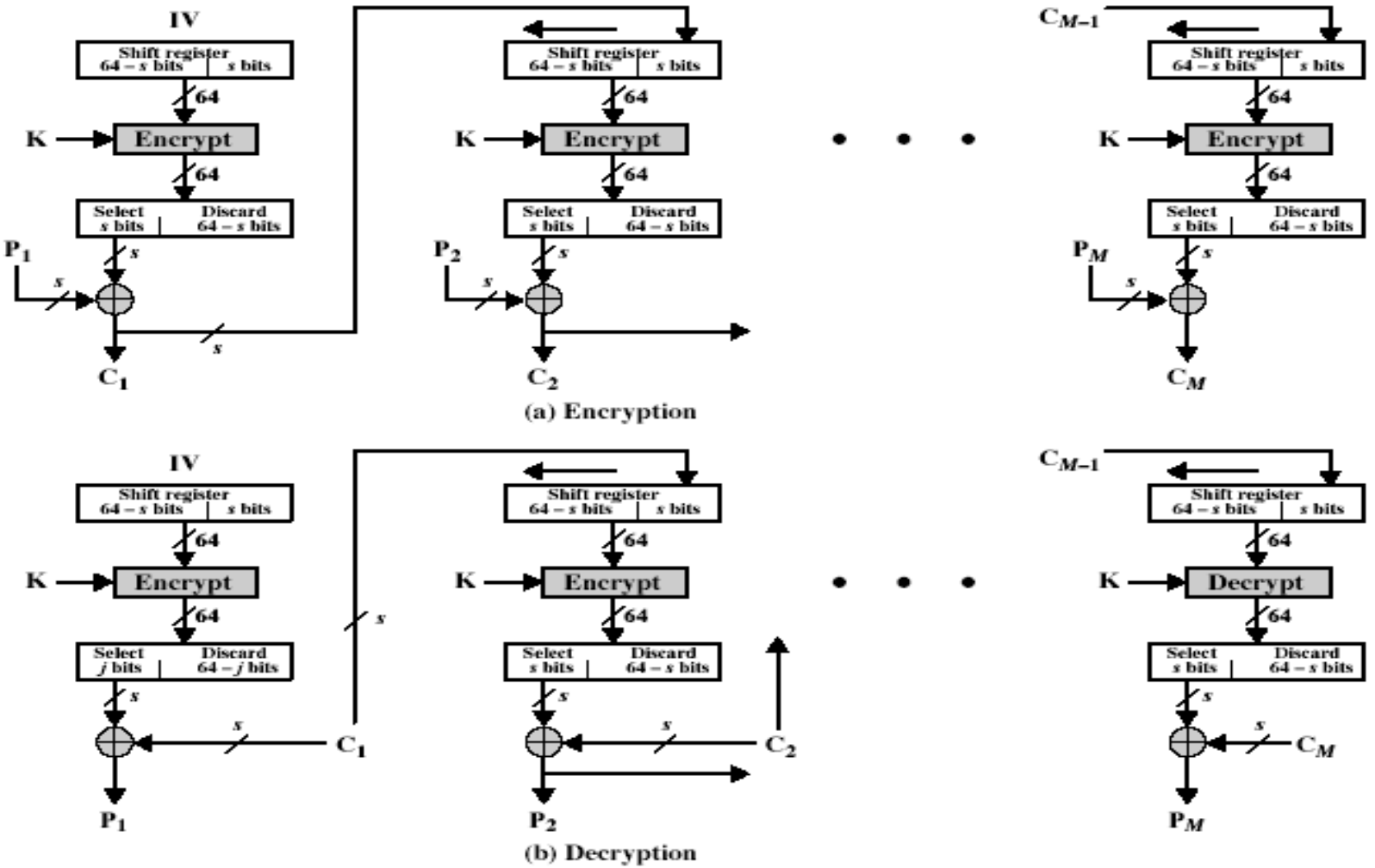
D : Decryption
C_i: Ciphertext block i
IV: Initial vector (S_1)

S_i: Shift register
T_i: Temporary register



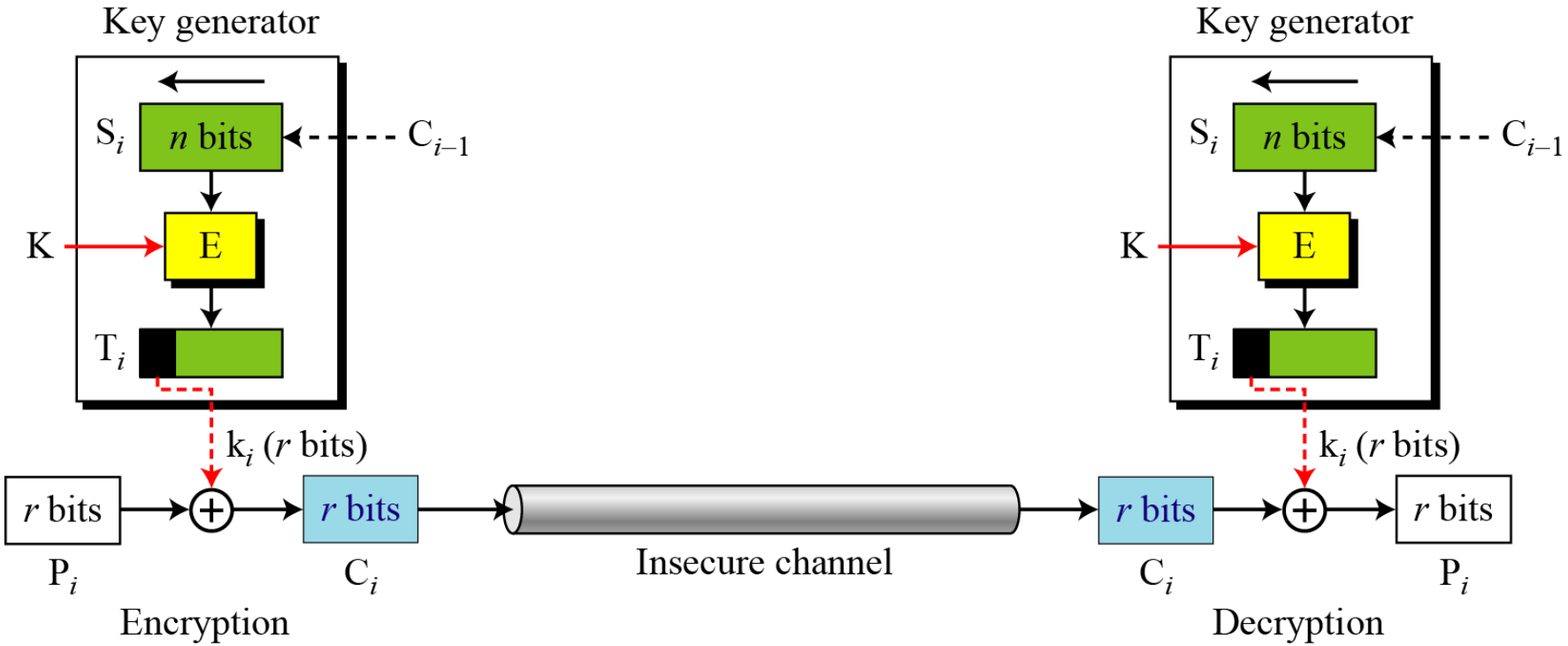
Encryption

CFB Encryption/Decryption



CFB as a Stream Cipher

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



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

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

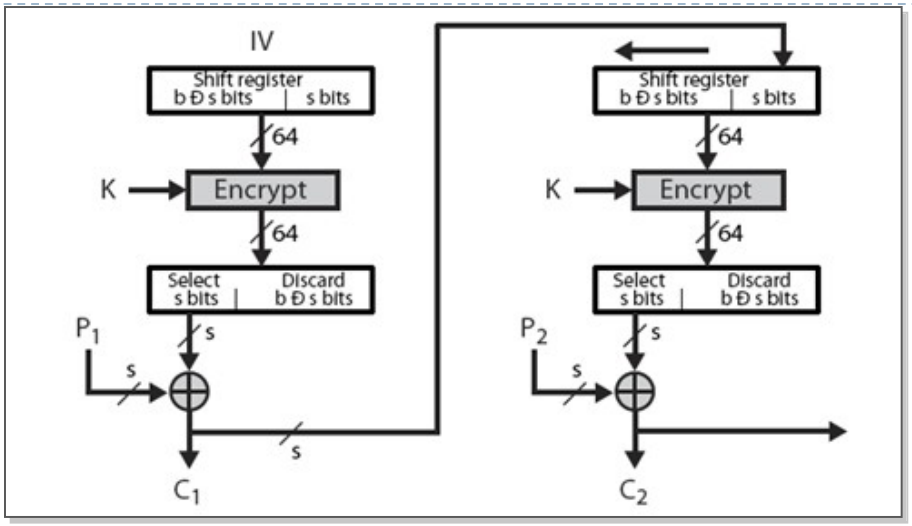
$$O_i = E_K(O_{i-1})$$

$$O_0 = IV$$

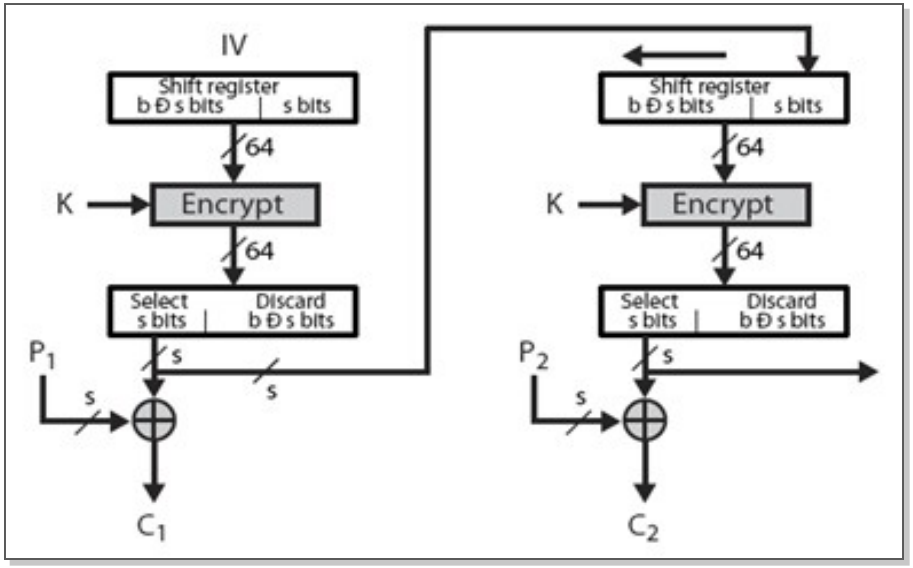
- Uses: stream encryption over noisy channels

CFB V.S. OFB

Cipher Feedback



Output Feedback

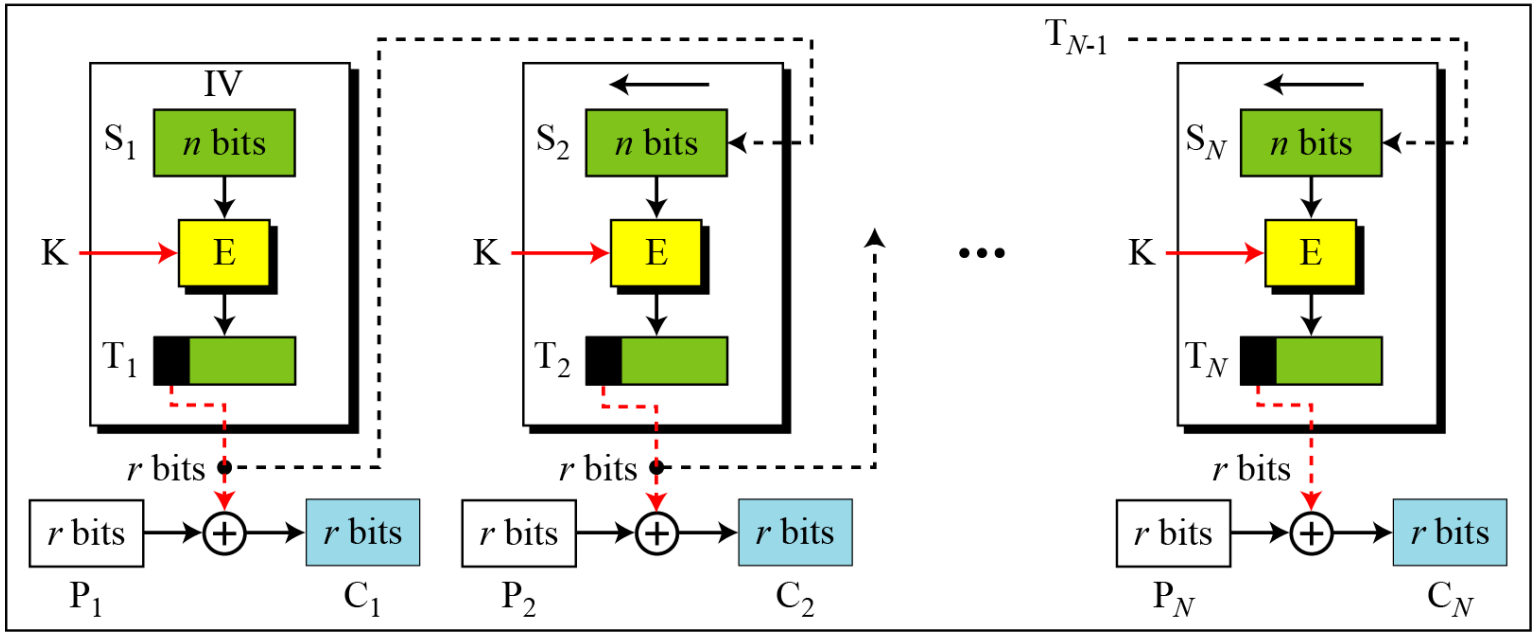


OFB Scheme

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

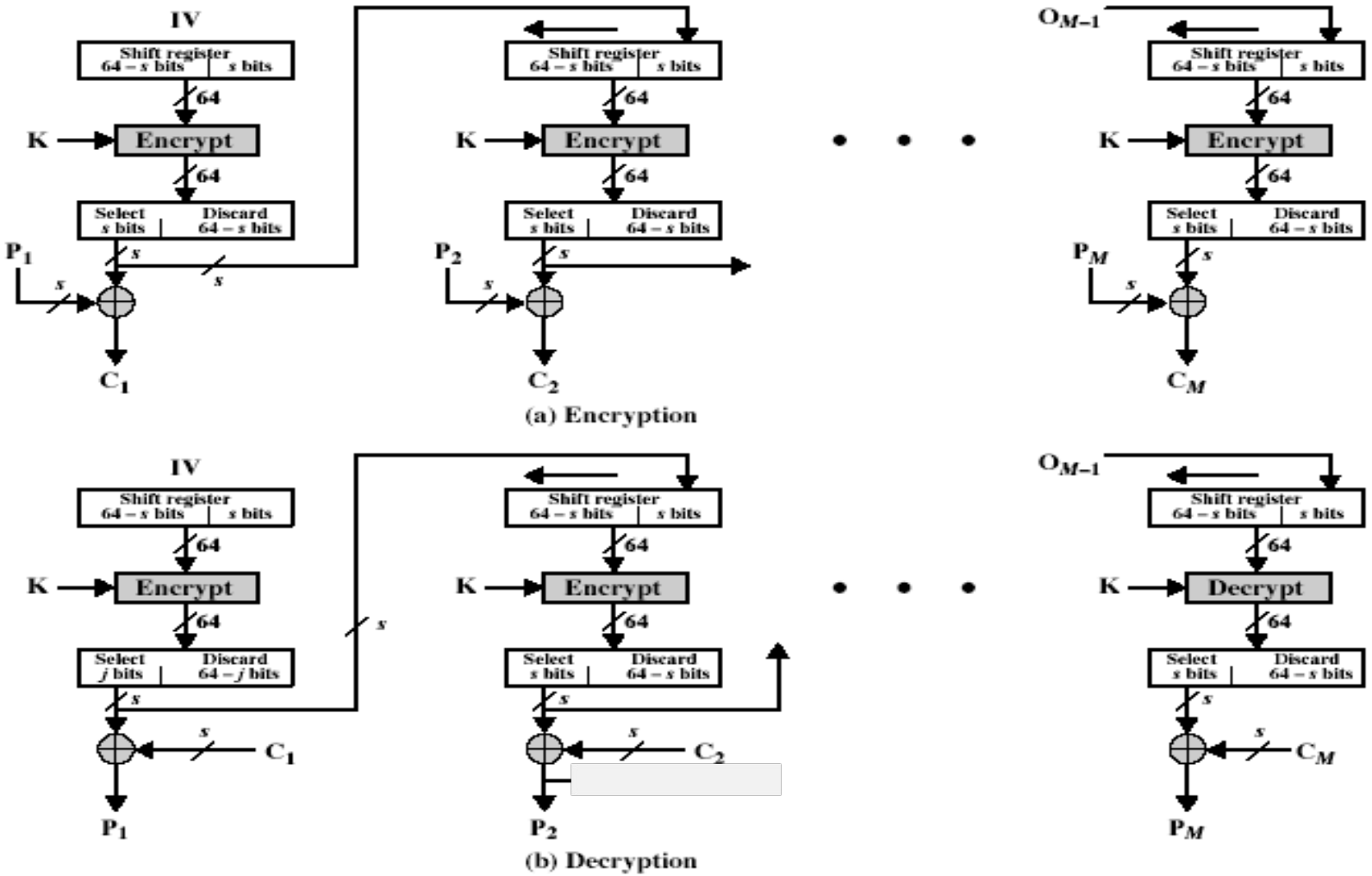
D : Decryption
C_i: Ciphertext block i
IV: Initial vector (S_1)

S_i: Shift register
T_i: Temporary register



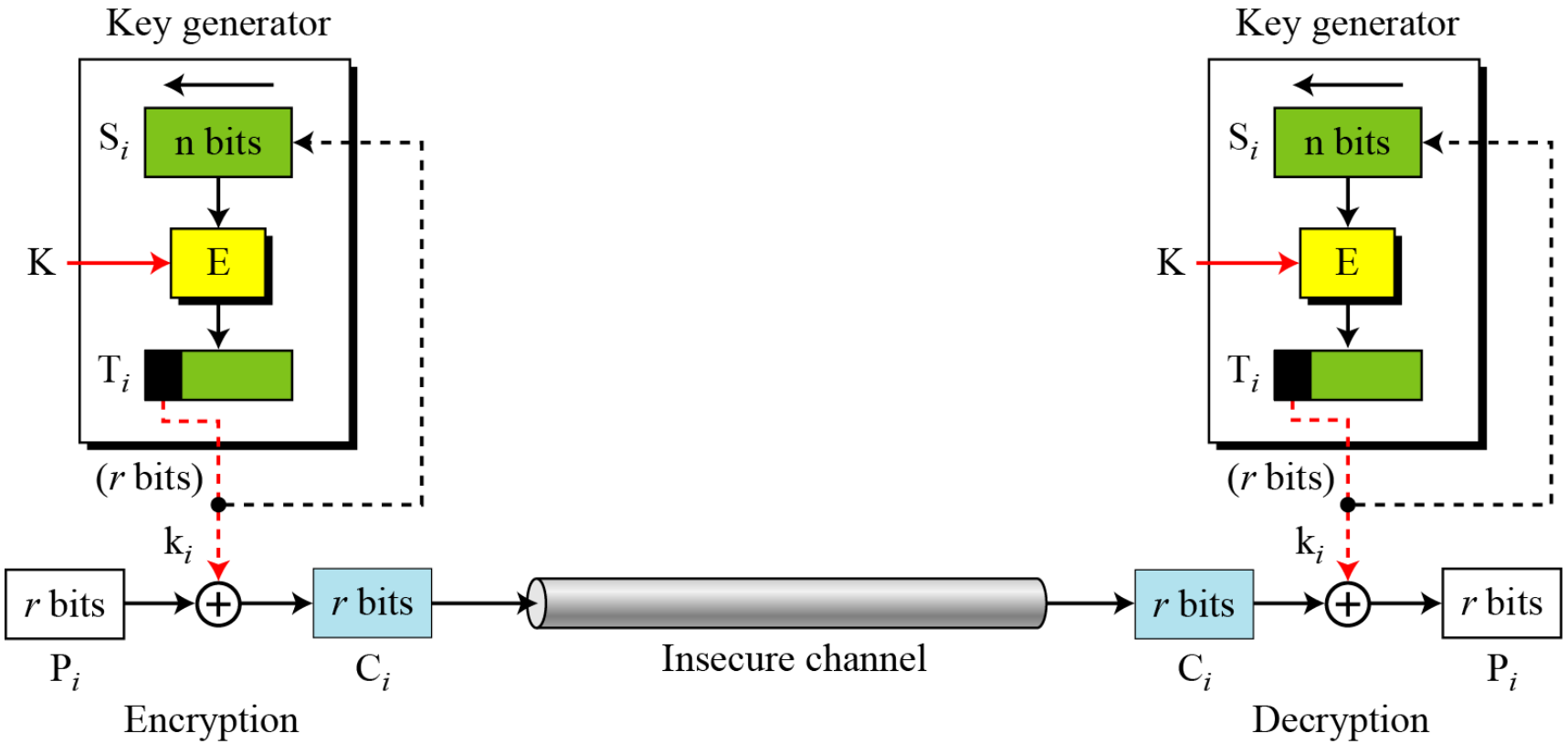
Encryption

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

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

$$O_i = E_K(i)$$

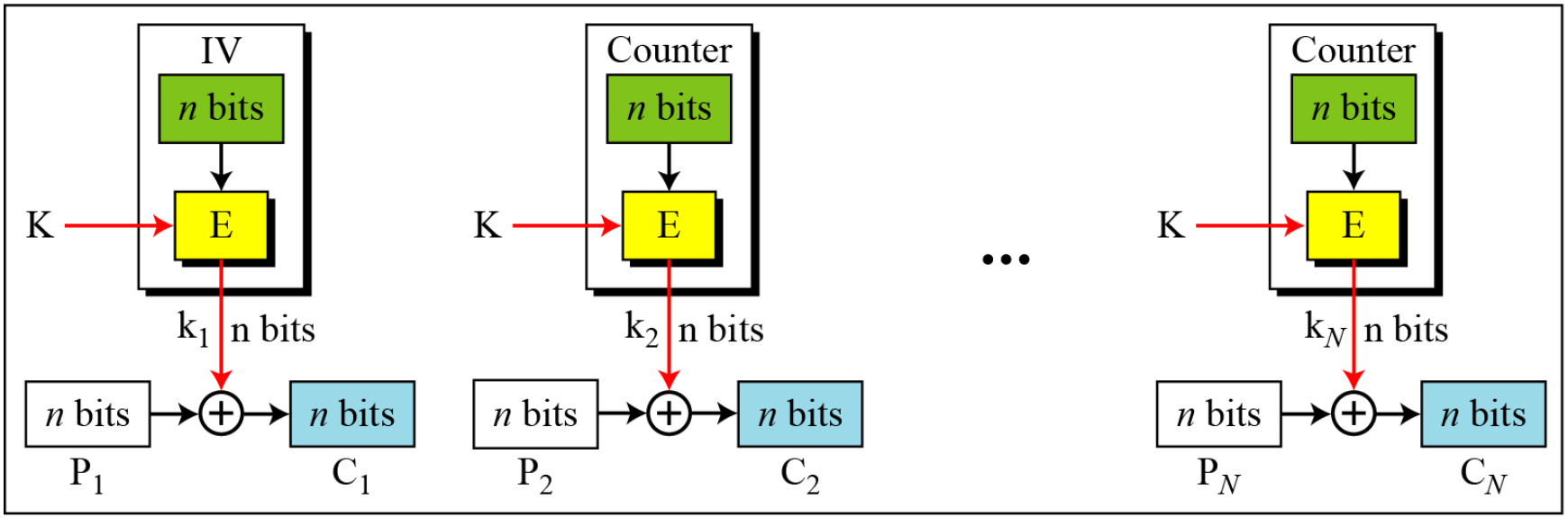
- Uses: high-speed network encryptions

CTR Scheme

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

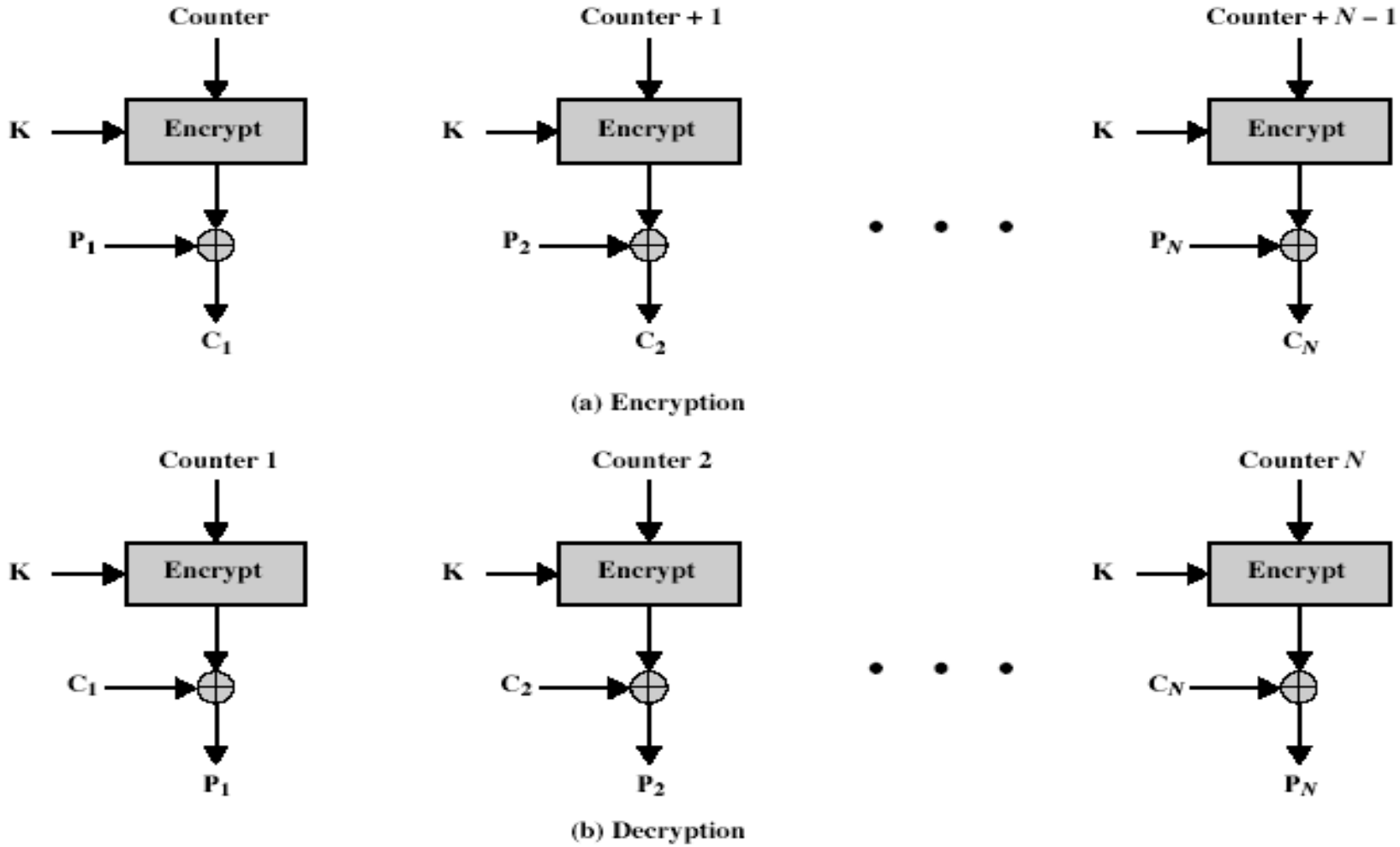
IV: Initialization vector
C_i : Ciphertext block *i*
k_i : Encryption key *i*

The counter is incremented for each block.



Encryption

CTR Encryption and Decryption



Remark on CTR

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

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

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

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

Comparison of Modes

Mode	Description	Application
ECB	64-bit plaintext block encoded separately	Secure transmission of encryption key
CBC	64-bit plaintext blocks are XORed with preceding 64-bit ciphertext	Commonly used method. Used for authentication
CFB	s bits are processed at a time and used similar to CBC	Primary stream cipher. Used for authentication

Comparison of Modes

Mode	Description	Application
OFB	Similar to CFB except that the output is fed back	Stream cipher well suited for transmission over noisy channels
CTR	Key calculated using the nonce and the counter value. Counter is incremented for each block	General purpose block oriented transmission. Used for high-speed communications