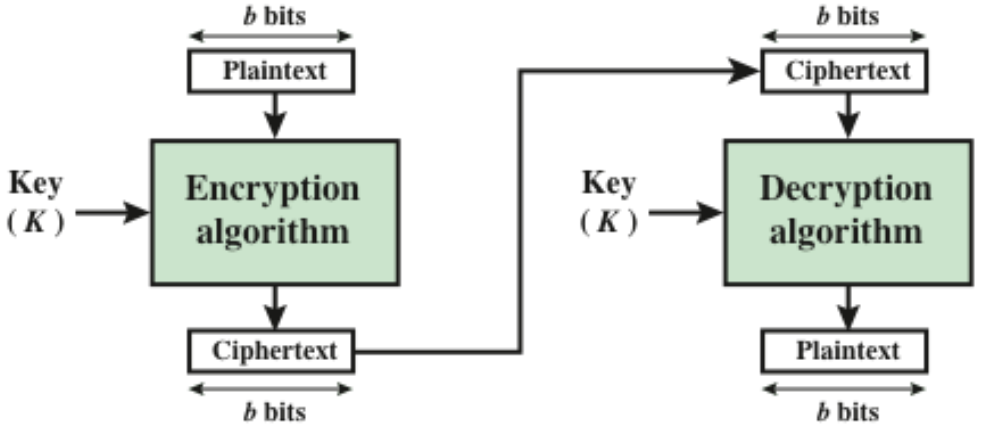
Block Cipher Operation Modes

Dr. Mana Khatua
Assistant Professor
Dept. of CSE, IIT Guwahati
Email: manaskhatua@iitg.ac.in

Why Different Modes?



- A block cipher takes a fixed-length block of text of length *b* bits and a key as **input** and produces a *b*-bit block of ciphertext as **output**.

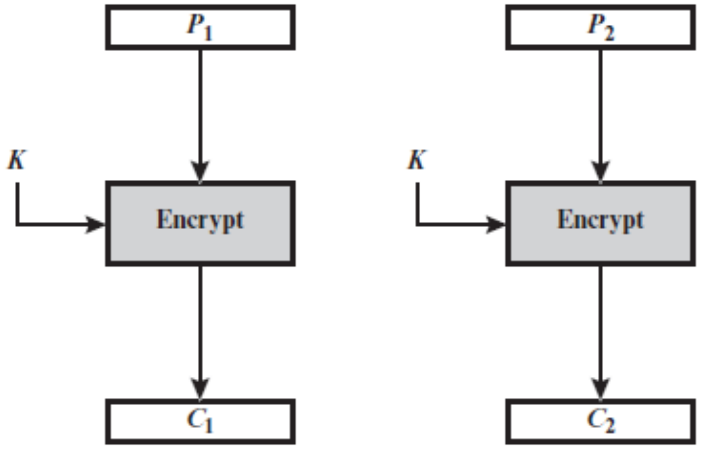
- What will happen when:
  - Input text size > b
  - Few segments appear repeatedly in input text
- If we use same key for multiple blocks, the cryptanalysis will be easier

- **Solution:**
  - NIST defines five modes of operations for any symmetric block cipher

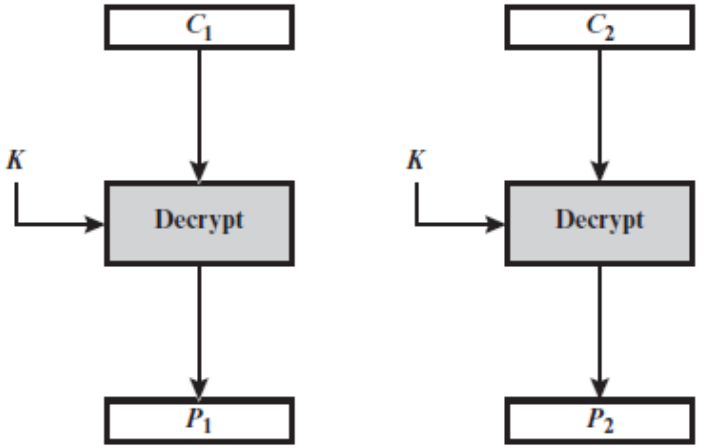
Five Modes

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 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 block of plaintext and the preceding block 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

Electronic CodeBook (ECB) mode



Encryption

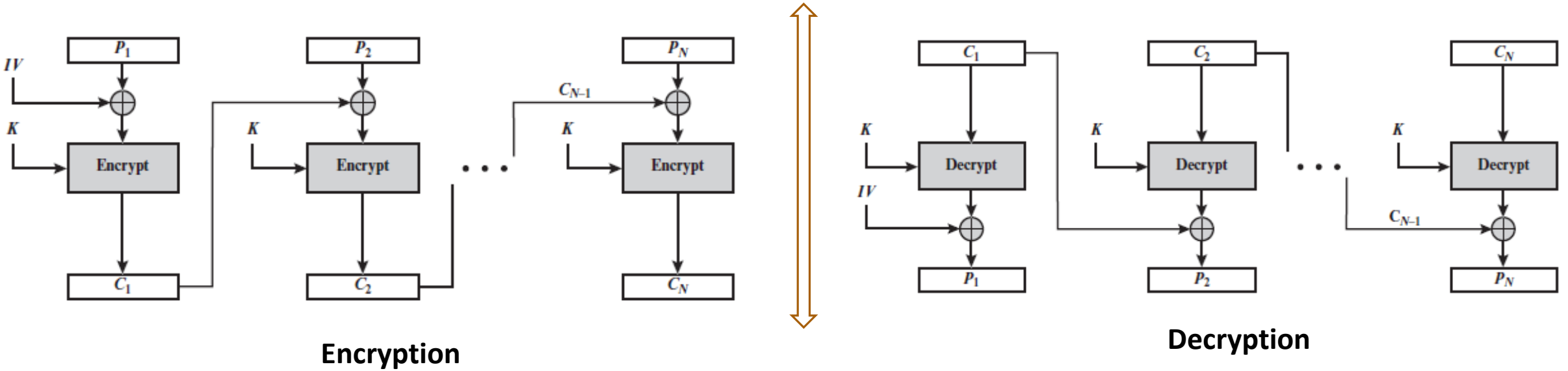


Decryption

- Plaintext is handled one block at a time and each block of plaintext is encrypted using the same key
- For a message longer than b bits, the procedure is simply to break the message into *b*-bit blocks, padding the last block if necessary.
- Significant characteristic: If the same b-bit block of plaintext appears more than once in the message, it always produces the same ciphertext.
- Suggested Use: only to secure messages shorter than a single block of underlying cipher

Cipher Block Chaining (CBC) mode

Requirement: The same plaintext block, if repeated, need to produce different ciphertext blocks.



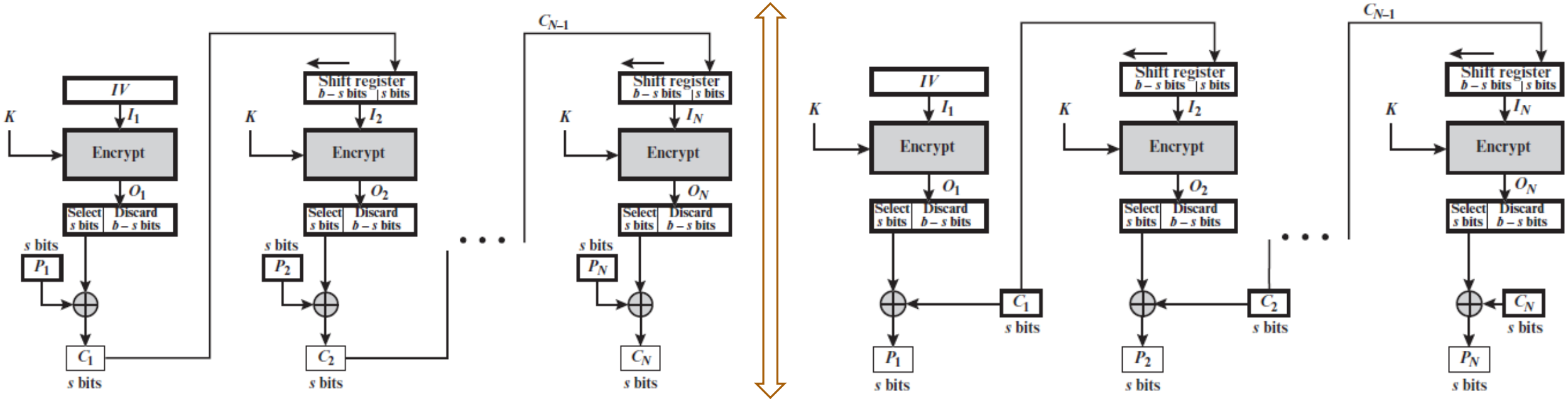
Encryption **Decryption**

- ✓ Each ciphertext block depends on **all** message blocks
- ✓ The IV must be known to both the sender and receiver but be unpredictable by a third party.
- ✓ One reason for protecting the IV while transmitted
  - If an opponent is able to fool the receiver into using a different value for IV, then the opponent is able to invert selected bits in the first block of plaintext.
- ✓ CBC mode is used to achieve confidentiality as well as authentication.

Cipher Feedback (CFB) mode

It is possible to convert a block cipher into a stream cipher.

- ✓ A stream cipher eliminates the need to pad a message to be an integral number of blocks.
- ✓ It also can operate in real time.



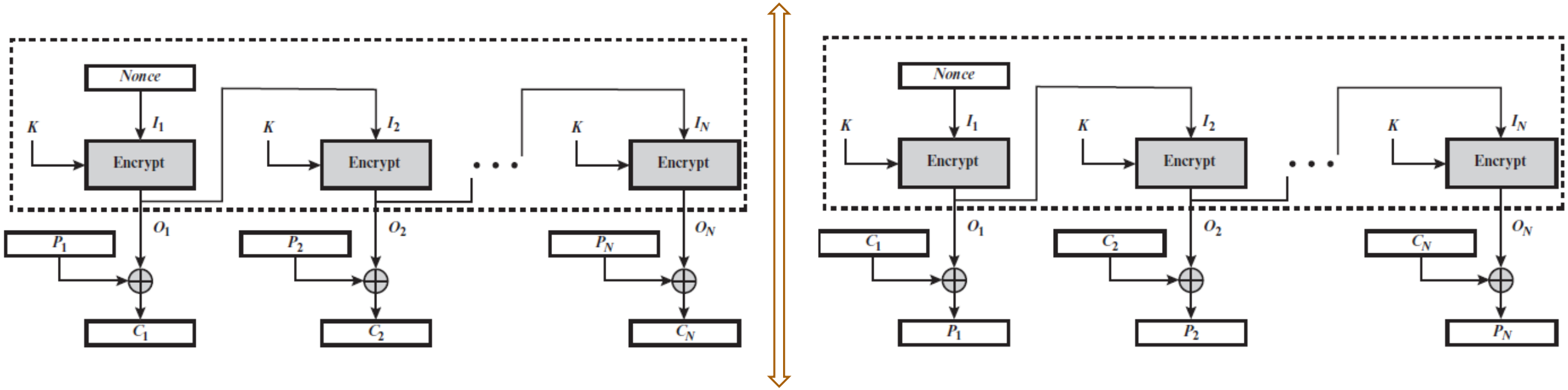
Cont…

CFB	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $C_j = P_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $P_j = C_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$

- ✓ Note that it is the *encryption* function that is used, not the *decryption* function during decryption.
- ✓ Although CFB can be viewed as a stream cipher, it does not conform to the typical construction of a stream cipher.
- ✓ Disadvantage 1: In CFB encryption, multiple forward cipher operations cannot be performed in parallel
  - as the input block to each forward cipher function (except the first) depends on the result of the previous forward cipher function
- ✓ However, parallel operation is possible for CFB decryption.
- ✓ Disadvantage 2: Bit errors in transmission (of C_i) gets propagated

Output Feedback (OFB) mode

- ✓The output of the encryption function is fed back to become the input for encrypting the next block of plaintext
- ✓OFB mode operates on full blocks of plaintext and ciphertext



Encryption **Decryption**

- ✓ In OFB, IV must be nonce, i.e. IV must be unique to each execution of the encryption operation.
- ✓ Advantage: Bit errors in transmission do not propagate

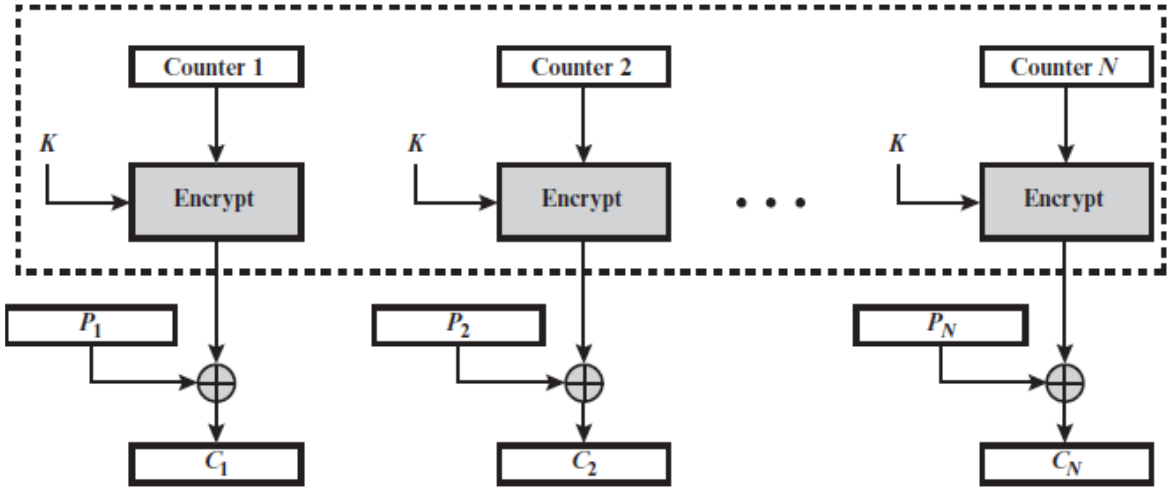
Cont…

OFB	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $C_j = P_j \oplus O_j \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $P_j = C_j \oplus O_j \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u(O_N)$

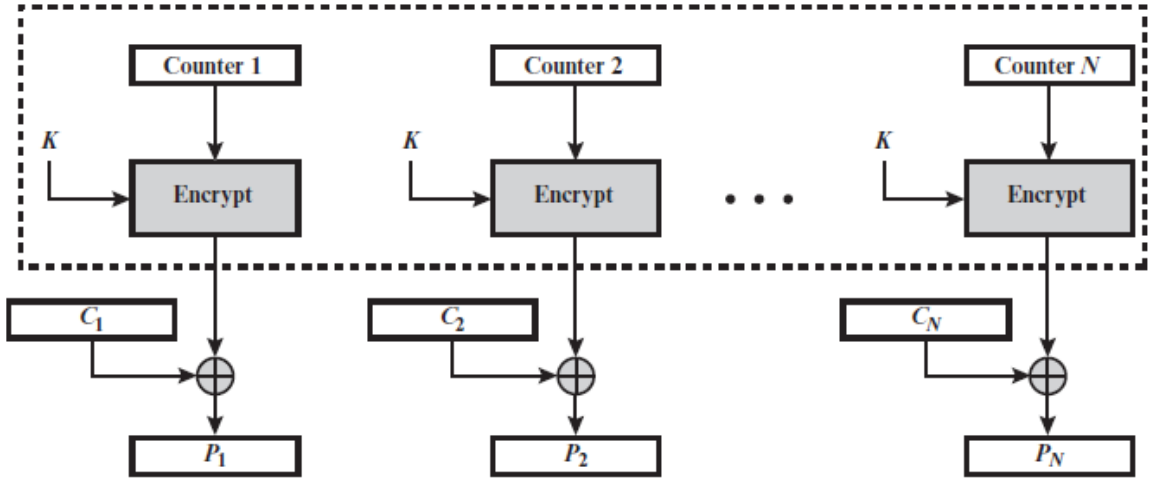
- Let the size of a block be *b*. If the last block of plaintext contains *u* bits with $u < b$, the most significant *u* bits of the last output block used for the XOR operation; the remaining $b - u$ bits of the last output block are discarded.
  - ✓ So, unlike ECB, CBC and CFB modes, it does not need padding.

- Disadvantage: it is more vulnerable to a message stream modification attack than is CFB

Counter (CTR) mode



Encryption



Decryption

- ✓A counter equal to the plaintext block size is used.
- ✓there is no chaining
- ✓Given a sequence of counters T_1, T_2, …., T_N, we can define CTR mode as follows.

CTR	$C_j = P_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$	$P_j = C_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$

Cont...

- Unlike the ECB, CBC, and CFB modes, we do not need to use padding because of the structure of the CTR
- T_1 must be different for all of the messages encrypted using the same key
- All T_i values across all messages must be unique

Advantages of CTR mode

- ✓ Hardware efficiency – Encryption / decryption in CTR mode can be done parallel on multiple blocks of plaintext or ciphertext
- ✓ Software efficiency – processors that supports parallel features can be utilized
- ✓ Pre-processing – if sufficient memory is available and security is maintained, pre-processing can be used to prepare the output of the encryption boxes that feed into the XOR functions
- ✓ Random access – The i-th block of plaintext or ciphertext can be processed in random-access fashion.
- ✓ Provable security – It can be shown that CTR is at least as secure as the other modes
- ✓ Simplicity – Unlike ECB and CBC modes, CTR mode requires only the implementation of encryption algorithm and not the decryption algo.


Thank
You