Block Cipher modes

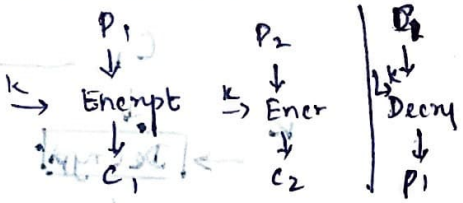
* Block of text → b bits + key ⇒ b bits cipher text
* if plaintext > b bits then it is broken down to b bit blocks.

Modes of operation → enhance the cryptographic algo

Simplest mode – electronic code-book (ECB) mode

* Each block of plaintext is encrypted using the same key
* 'Code book' is used as diff ciphertext to b-bit block plaintext for the key
* For message longer than b bits, the procedure is to break the message into b-bits * padding the last block.
* Decryption is performed one block at a time

ECB : $C_j = E(K, P_j)$	$P_j = D(K, C_j)$



* Used for short amt of data, DES, AES → key transfer
* Same plaintext is used > once, then same ciphertext
* Not secure for lengthy message

Cryptanalyst – Difficult for highly structured msg

Overhead : Additional opr. for encryption / Decryption

Error recovery : error in i^{th} ciphertext is inherited for P.T block

Error propagation

Diffusion – Low Entropy is not reflected to C.T blocks

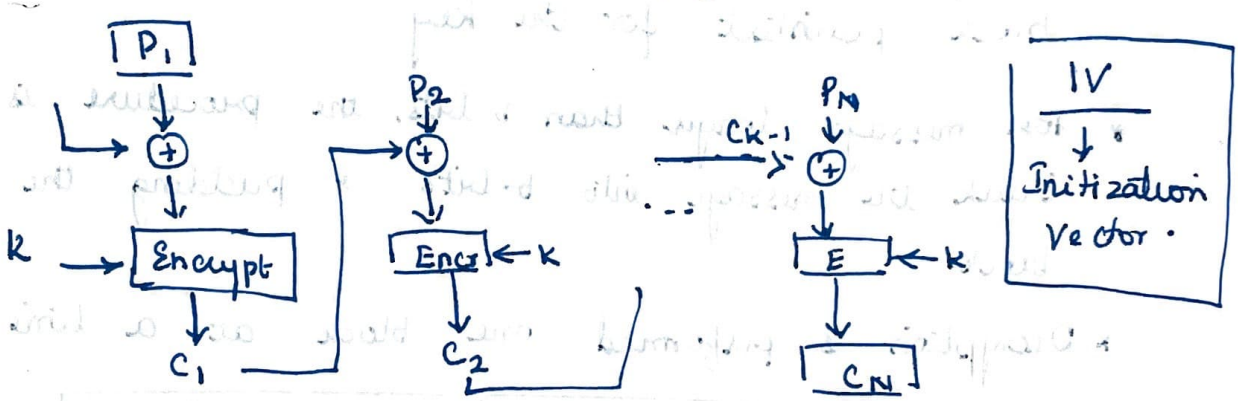
Security

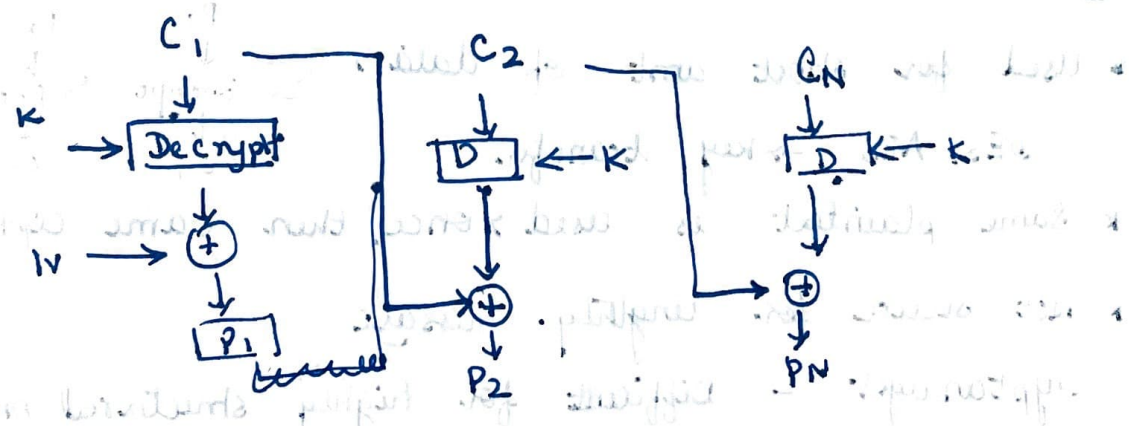
④ Cipher block chaining mode:

* Overcome security deficiencies, same text block if repeated produces diff. cipher block.
* Input to the encryption algo. XOR of the current plaintext and the preceding ciphertext, same key for each block.
* CBC → last block is fully padded.

For decryption, each cipher block is passed through a decryption algo. Result XORed with ciphertext block.

$$C_j = E\,[K, [C_{j-1} \oplus P_j)]$$





$$D\,(k, c_j) = D\,(k, E\,(K, c_{j-1} \oplus P_j))$$

IV – Init vector – first block of data

IV is known to the sender & receiver

* Not possible to predict the IV

IV is send using ECB

2 methods

↳ Apply encryption fn, under the same key to nonce (TS/seq. no)

↳ Random data block using random number generator

③ Cipher feedback mode

* Block cipher – encryption on a block of b bits.

DES, b = 64 bits AES, b = 128

Convert block cipher into stream cipher using 3 mods

↳ Cipher feedback mode (CFB)

↳ O/p feedback (OFB)

↳ Counter mode (CTR)

* Stream cipher eliminates, need to pad

CFB, unit of transmission is s bits, s = 8

plaintext – chained together, ciphertext is fn of preceding plaintext, but P.T is divided into s bits

Input to encryption fn is b bit shift register

Set to initialization vector (IV)

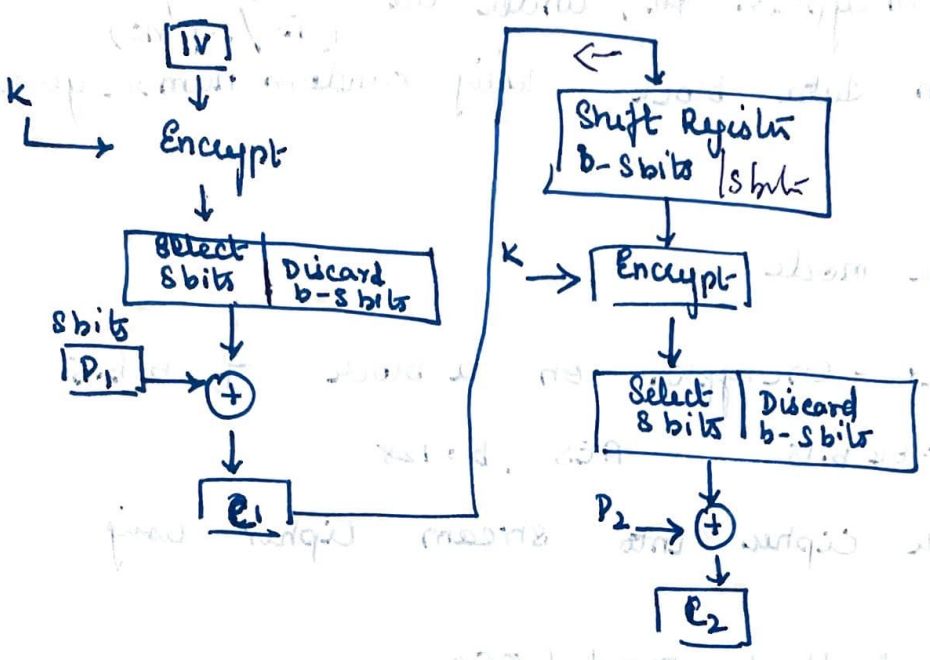
Leftmost s bits of O/p of encryption fn ⊕ first seg. of plain text ⇒ Ciphertext

Shift register is shifted by s bits (left) & C_1 is placed in rightmost of s bits of shift register

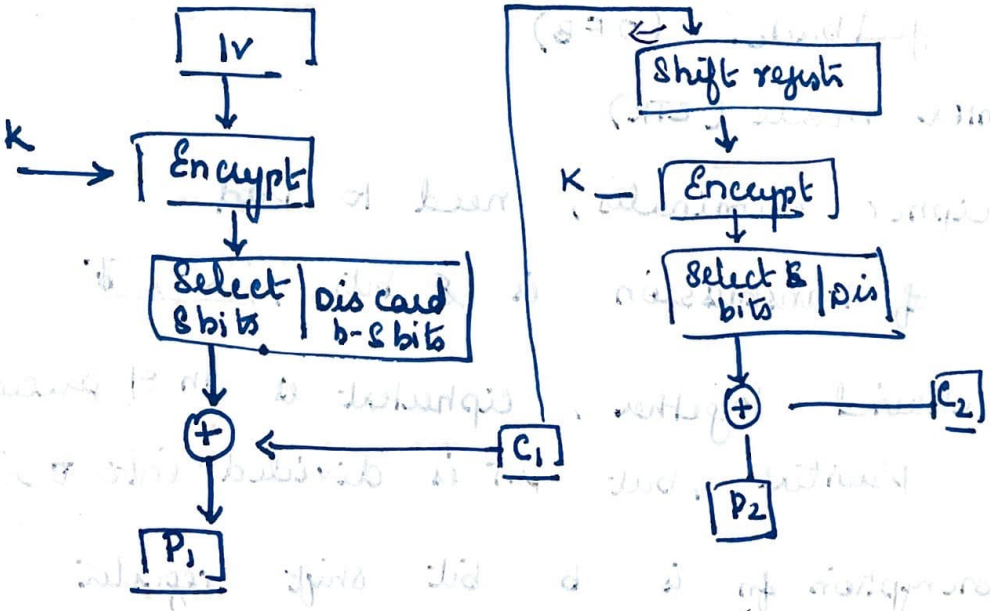
* Decryption, Ciphertext is ⊕ with O/p of encryption fn to produce plaintext-

$$C_1 = P_1 \oplus MSB_s [E(K, IV)]$$

$$P_1 = C_1 \oplus MSB_s [E(K, IV)]$$



Preceding C.T is used to encrypt alg to produce pseudo random O/p. → ⊕ PT ∴ to produce C.T



$I_1 = iv$

$I_j = LSB_{b-s}(I_{j-1}) \| C_{j-1}$

$O_j = E(K, I_j)$

$C_j = P_j + MSB(O_j)$

$I_1 = IV$

$I_j = LSB_{b-s}(I_{j-1}) \| C_{j-1}$

$O_j = E(K, I_j)$

$P_j = C_j \oplus MSB(O_j)$

Output feedback Mode

Ilar to CFB.

OFB, output of encryption fn is fedback to become I/p for encrypting next block of plaintext

OFB, mode operates on full blocks of plaintext & Cipher text

$$C_j = P_j \oplus E(K, O_{j-1})$$

$$O_{j-1} = E(K, O_{j-2})$$

$$\therefore C_j = P_j \oplus E(K, [C_{j-1} \oplus P_{j-1}])$$

Illy $P_j = C_j \oplus E(K, [C_{j-1} \oplus P_{j-1}])$

$I_1 = \text{nonce}$

$I_j = O_{j-1}$

$O_j = E(K, I_j)$

$C_j = P_j \oplus O_j$

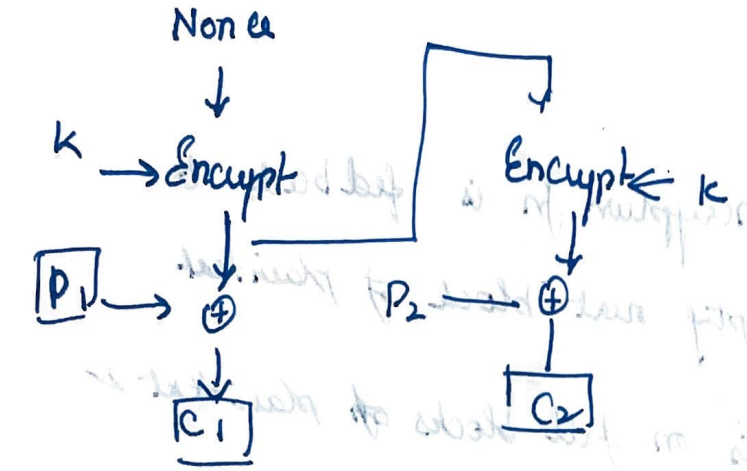
$C_N = P_N \oplus MSB(O_N)$

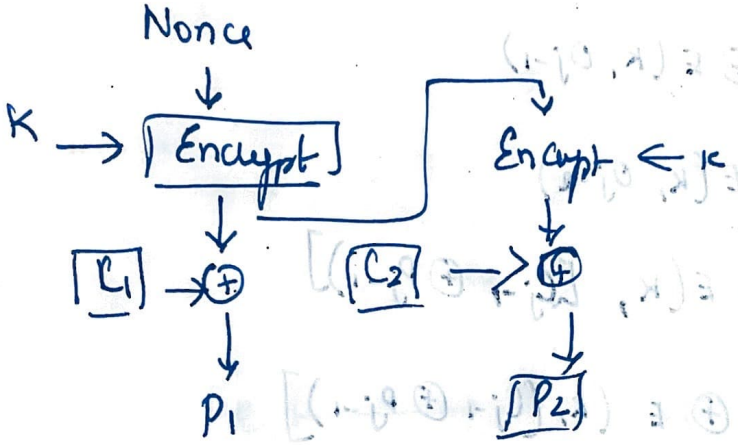
- Size of block → b

Last block of plaintext → u bits , u < b

MSB of u bits of the last O/p block O_N are used for Xor opr

IV is used → nonce





Encryption → Key, IV

→ O/p of encrypt to be xor to fixed

Bit errors do not propagate

Disadv : Msg stream modification

Counter mode

* ATM net security

Ctr → plaintext block size is used
diff counter value for each plaintext used

Counter initialized & incremented by 1

Encryption → Counter is encrypted ⊕ Plaintext

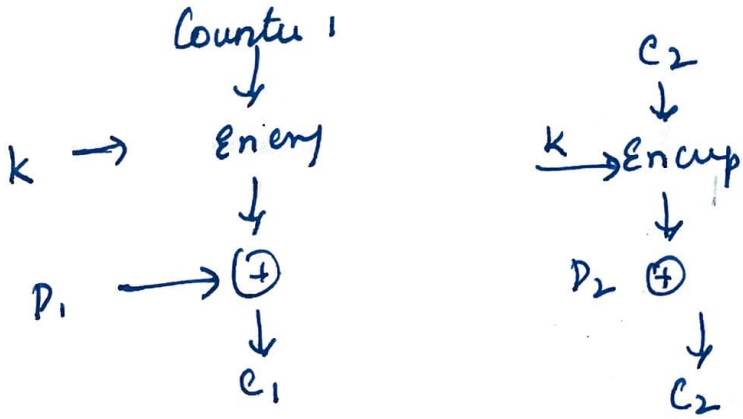
Decryption → Same counter values ⊕ ciphertext

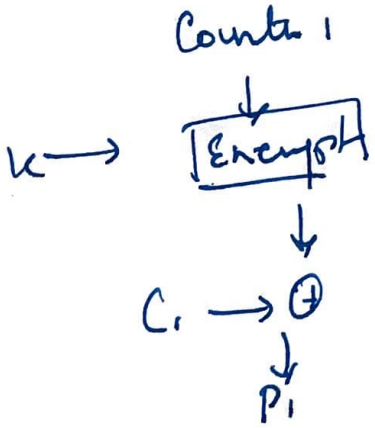
$C_j = P_j \oplus E(K, T_j)$ $\quad P_j = C_j \oplus E(k, T_j)$.

$C_N = P_N \oplus MSB(E, T_N)$

Last block MSB is used for EXOR opr.

if counter is repeated, confidentiality is used





* Increment by 1

Advan

* Hardware effi : Encryption (Dec) done parallely

↳ S/W eff : Parallel execution, processor support multiple features

↳ Pre processing

↳ Random access

↳ Provable Security

↳ Simplicity → ECB, CBC - Ctr requires only encryption

Paddy

↳ Simple 10*

↳ Multi rate 10*1