Modes of Data Operation

Modes of Operation

- block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks with 56-bit key
- need some way to en/decrypt arbitrary amount of data in practice
- ANS X3.106-1983 Modes of Use (now FIPS 81) define 4 possible modes to cover all possible applications
- subsequently expanded by NIST & 5 modes are defined for symmetric block ciphers *(AES & DES)*
- have **block** and **stream** modes

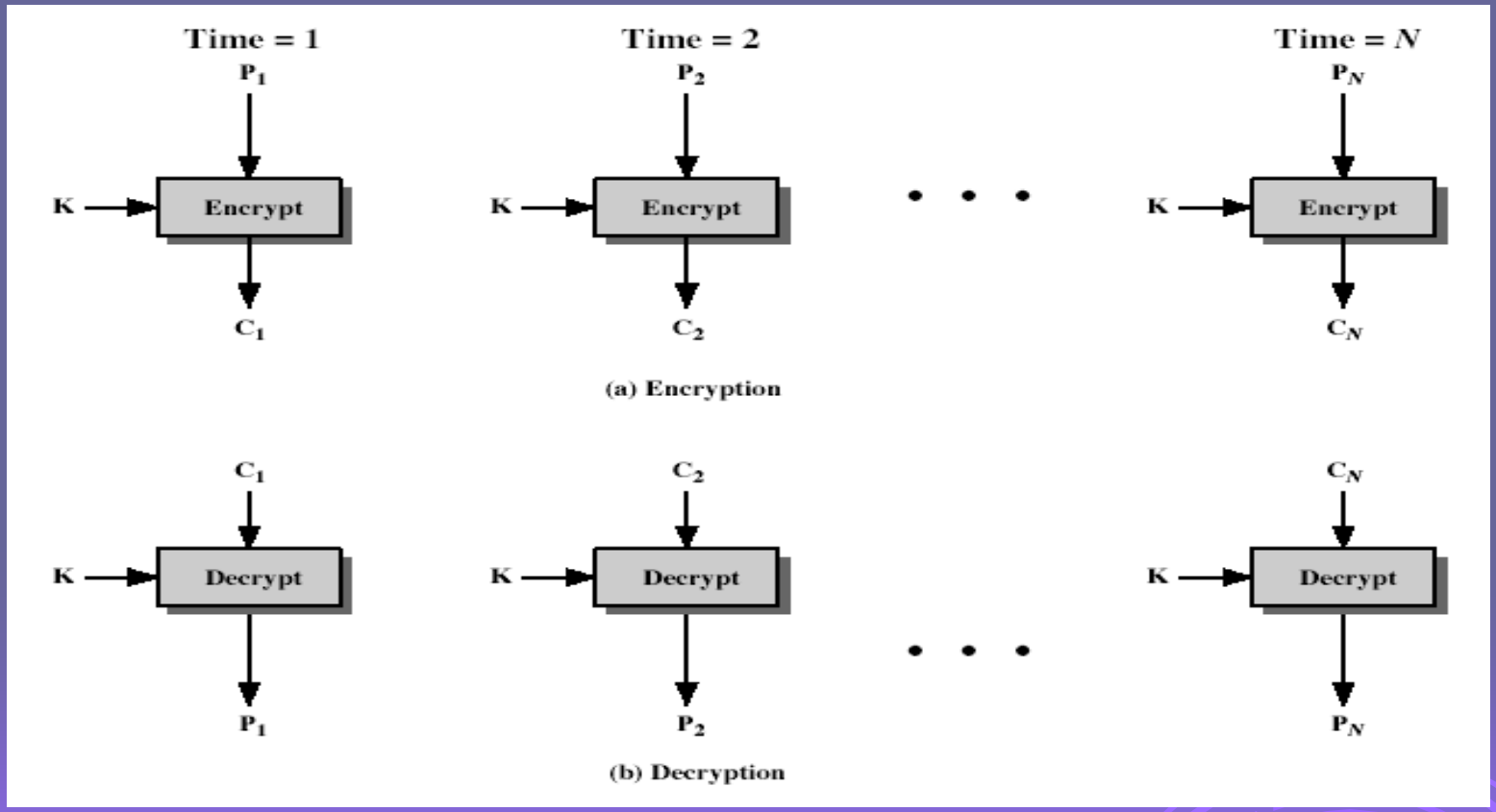
Electronic Codebook Book (ECB)

- 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

$$C_i = DES_{K1}(P_i)$$

- uses: secure transmission of single values

Electronic Codebook Book (ECB)



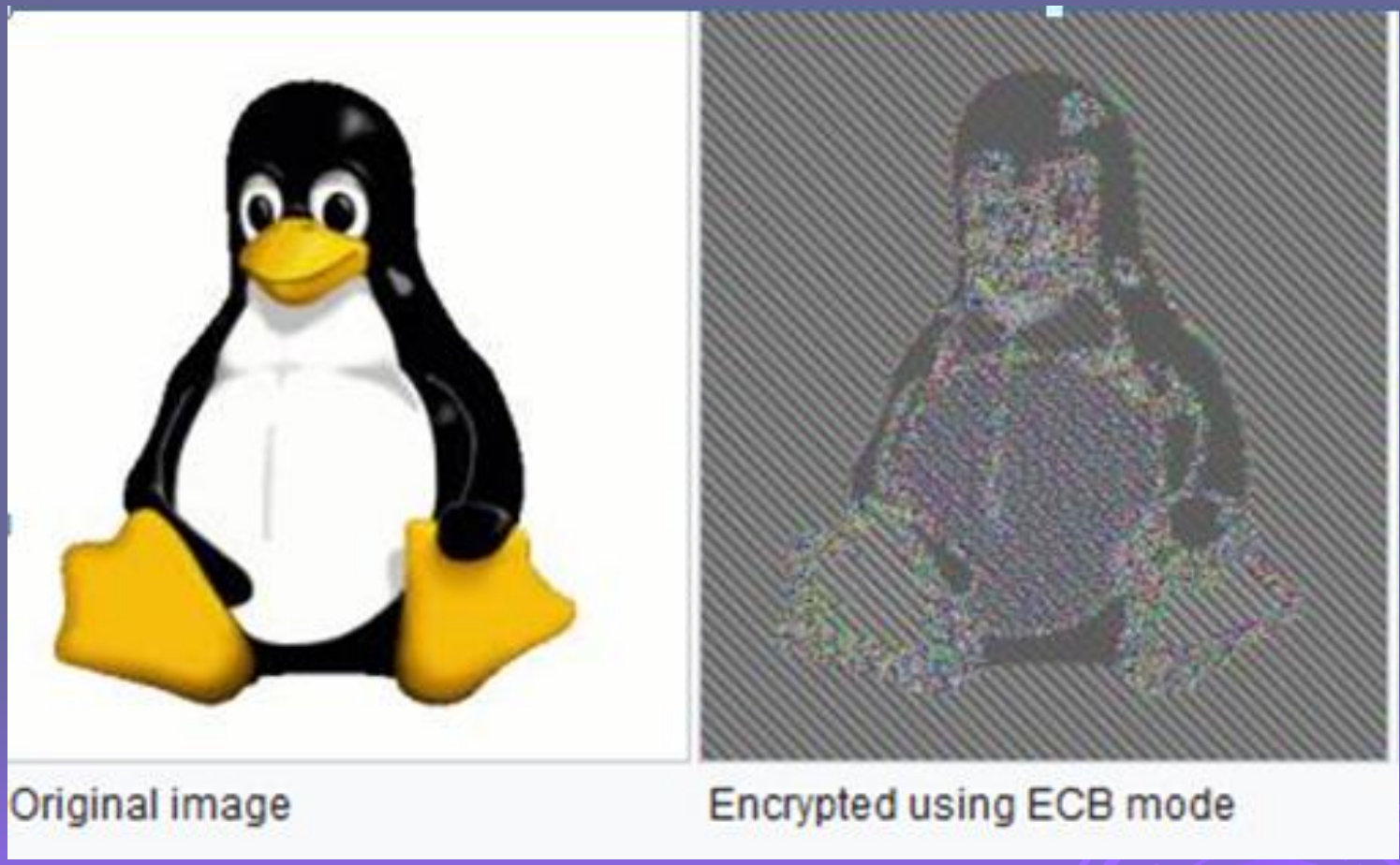
Advantages and Limitations of ECB

- message repetitions may show in ciphertext
  - if aligned with message block

  particularly with data such graphics

  or with messages that change very little, which become a **code-book analysis** problem
- weakness is due to the encrypted message blocks being independent
- main use is sending a few blocks of data

CASE.. ECB



Cipher Block Chaining (CBC)

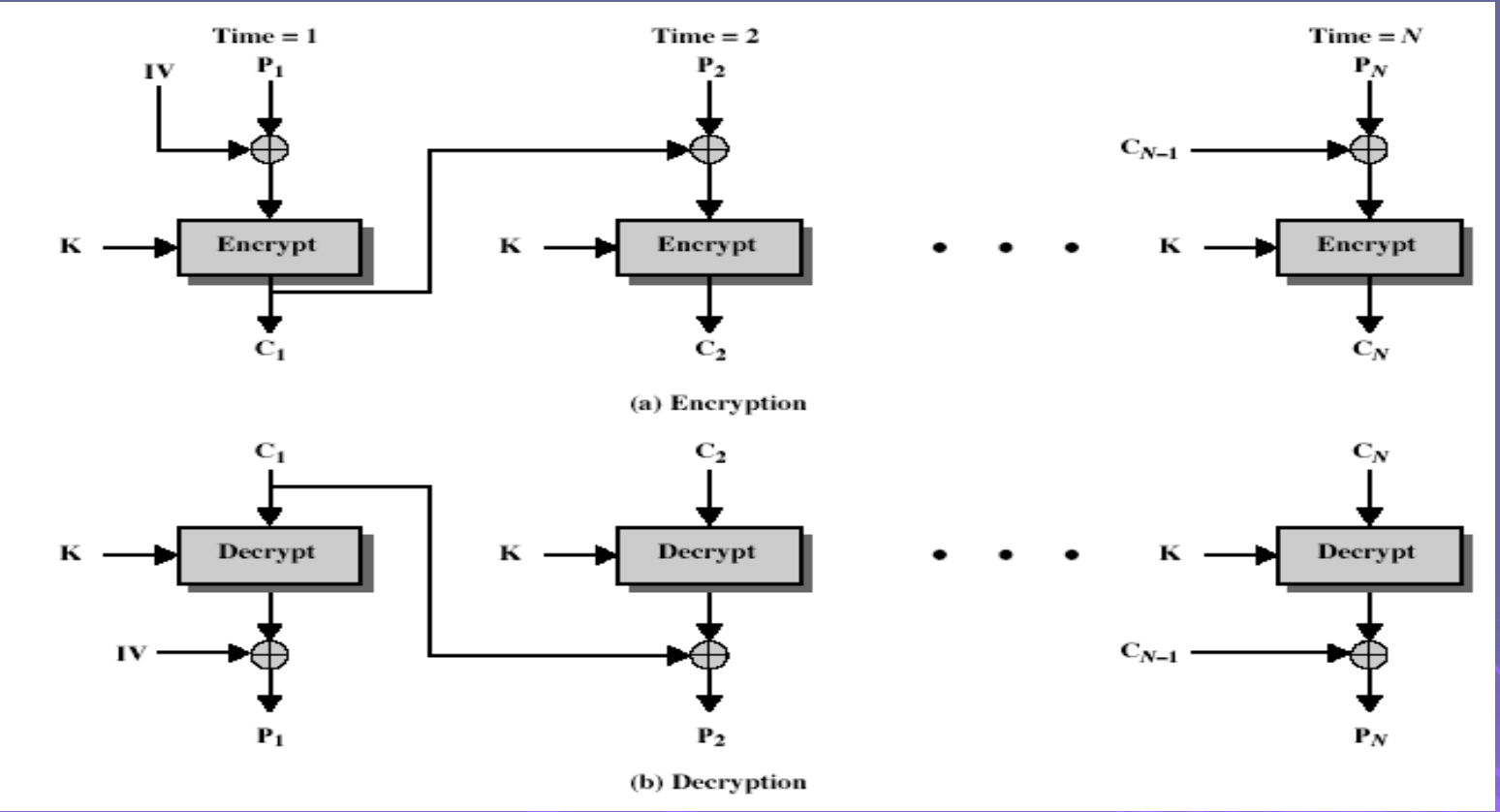
- message is broken into blocks
- linked together in encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process

$$C_i = DES_{K1}(P_i \text{ XOR } C_{i-1})$$

$$C_{-1} = IV$$

- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC)



Message Padding

- at the end of message; may handle a possible last short block
  - which is not as large as blocksize of cipher

  pad either with known non-data value (eg nulls)

  or pad last block along with count of pad size
    - eg. [b1 b2 b3 0 0 0 0 5]
    - means have 3 data bytes, then 5 bytes pad+count
  - this may require an extra entire block over those in message
- there are other, more esoteric modes, which avoid the need for an extra block

Advantages and Limitations of CBC

- a ciphertext block depends on **all preceding blocks** any change to a block affects all following ciphertext blocks
- need **Initialization Vector** (IV)
  - which must be known to sender & receiver
  - if sent in clear, attacker can change bits of first block, and change IV to compensate
  - hence IV must either be a fixed value
  - or must be sent encrypted in ECB mode before rest of message

Cipher FeedBack (CFB)

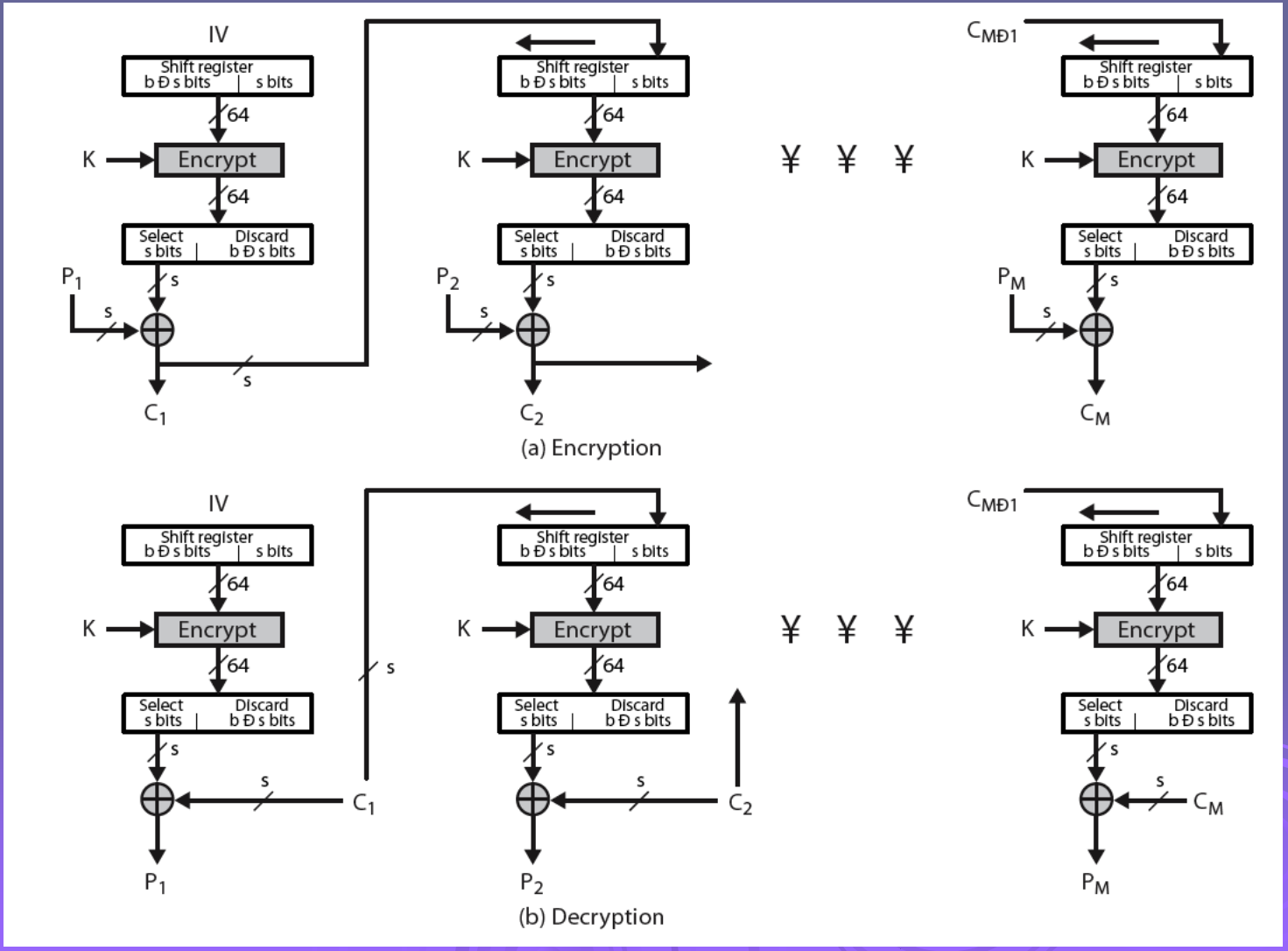
- message is treated as a **stream of bits**
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8, 64 or 128 etc) to be feed back
  - denoted CFB-1, CFB-8, CFB-64, CFB-128 etc
- most efficient to use all bits in block (64 or 128)

$$C_i = P_i \text{ XOR } DES_{K1}(C_{i-1})$$

$$C_{-1} = IV$$

- uses: stream data encryption, authentication

Cipher FeedBack (CFB)



Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- note that the block cipher is used in **encryption mode at both ends**
- **errors propagate for several blocks** after the error
- limitation is need to stall while do **block encryption after every n-bits**

Output FeedBack (OFB)

- message is treated as a **stream of bits**
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance

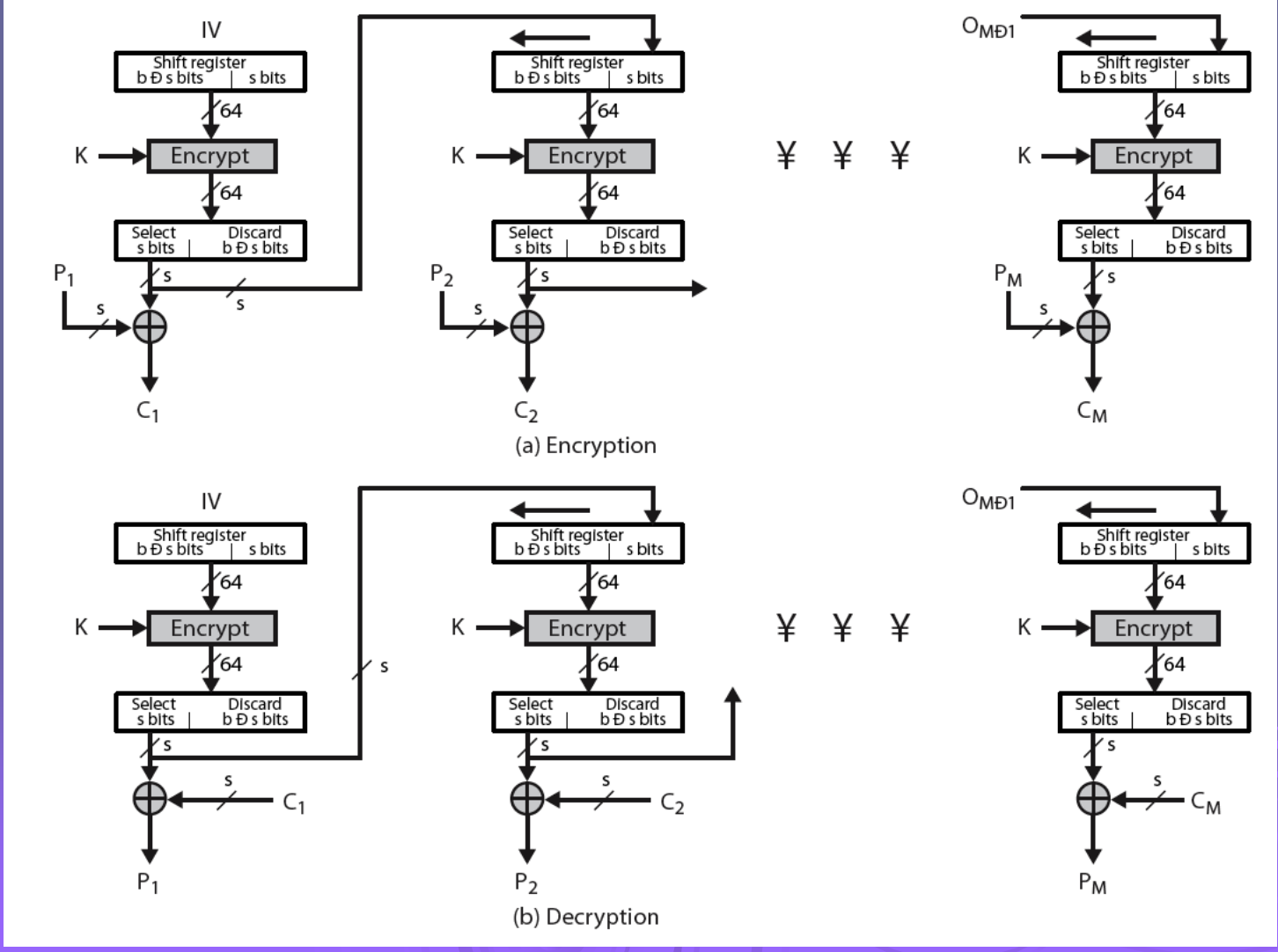
$$C_i = P_i \text{ XOR } O_i$$

$$O_i = DES_{K1}(O_{i-1})$$

$$O_{-1} = IV$$

- uses: stream encryption on noisy channels

Output FeedBack (OFB)



Advantages and Limitations of OFB

- **bit errors do not propagate**
- more **vulnerable to** message stream **modification** than CFB
- a variation of a Vernam cipher
  - hence must **never** reuse the same sequence (key+IV)
- **sender & receiver must remain in sync**
- originally specified with m-bit feedback
- subsequent research has shown that only **full block feedback** (ie CFB-64 or CFB-128) should ever be used

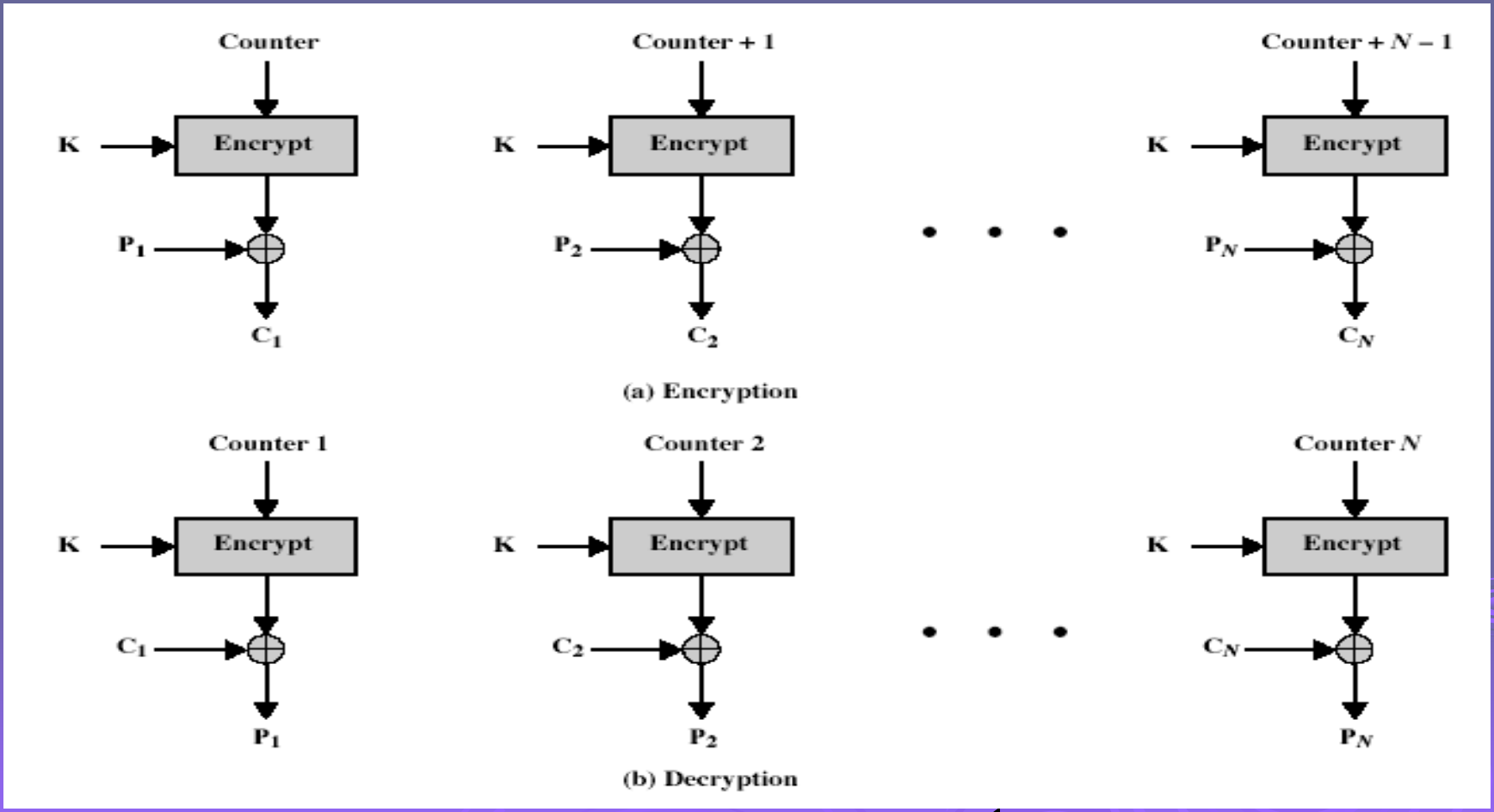
Counter (CTR)

- similar to OFB but encrypts counter value rather than any feedback value
- recently been standardized for use with AES along with the other existing 4 modes
- must have a different key & counter value for every plaintext block (never reused)

$C_i = P_i \text{ XOR } O_i \qquad O_i = DES_{K1}(i)$

- uses: high-speed network encryptions, applications in ATM (asynchronous transfer mode) network security

Counter (CTR)


Counter
K
Encrypt
P_1
C_1
Counter + 1
K
Encrypt
P_2
C_2
Counter + N – 1
K
Encrypt
P_N
C_N
(a) Encryption
Counter 1
K
Encrypt
C_1
P_1
Counter 2
K
Encrypt
C_2
P_2
Counter N
K
Encrypt
C_N
P_N
(b) Decryption

Advantages of CTR mode

✓**H/w Efficiency** – Unlike other chaining modes en/decryption can be done in parallel on multiple blocks of P or C. The throughput is only limited by amount of parallelism that is achieved.

✓**S/w Efficiency** – Due to parallel execution processors that support parallel features, such as aggressive pipelining, multiple instruction dispatch per block cycle, a large number of registers, and SIMD instructions can be effectively utilized.

contd..

✓**Preprocessing** – The execution of CTR mode does not depend on I/p of P or C, so with sufficient memory and its security, preprocessing can be used to prepare o/p of the encryption boxes that feed into the XOR functions. This strategy greatly enhances the output.

✓**Random Access** – The i^{th} block of P or C can be processed in random fashion.

✓**Provable security** – Proved that CTR method is as secure as other.

✓**Simplicity** – Requires only encryption algo. And not decryption algo.-> decryption key scheduling not required.

Limitations of CTR

- ✓ Must ensure and never reuse key/counter values, otherwise could break.

Tutorial (Solved)

Mode	Encrypt	Decrypt
ECB	$C_j=E_k[P_j]$ $j=1\ldots N$	$P_j = D_k[C_j]$ $j=1\ldots N$
CBC	$C_1=E_k[P_1 \text{ XOR } IV]$ $C_j=E_k[P_j \text{ XOR } C_{j-1}]$ $j=2\ldots N$	$P_1=D_k[C_1] \text{ XOR } IV$ $P_j=D_k[C_j] \text{ XOR } C_{j-1}$ $j=2\ldots N$
CFB	$C_1=P_1 \text{ XOR } E_k[IV]$ $C_j=P_j \text{ XOR } E_k[C_{j-1}]$ $j=2\ldots N$	$P_1=C_1 \text{ XOR } E_k[IV]$ $P_j=C_j \text{ XOR } E_k[C_{j-1}]$ $j=2\ldots N$
OFB	$C_i=P_i \text{ XOR } O_i$ $O_j=E_k[O_{j-1}] \quad i=1\ldots N$ $O_{i-1}=IV$	$P_i=C_i \text{ XOR } O_i$ $O_j=E_k[O_{j-1}] \quad i=1\ldots N$ $O_{i-1}=IV$
CTR	$C_i=P_i \text{ XOR } O_i$ $O_{i=}E_k(i) \quad i=1\ldots N$	$P_i=C_i \text{ XOR } O_i$ $O_{i=}E_k(i) \quad i=1\ldots N$

Tutorial_2/3

Complete the table---Comment on CIA priorities

Mode	*CIA?*	Encryption Parallelizable	Decryption Parallelizable	Random Read Access
ECB	*CIA/ CAI*	Yes	Yes	Yes
CBC	*CAI*	No	**Yes ??**	**Yes??**
CFB	*ACI*	No	**Yes??**	**Yes??**
OFB	*AIC/ ACI*	No	No	No
CTR	*CIA/ CAI*	Yes	Yes	Yes

Why CBC decryption can be parallel?

A major advantage of CBC mode is that, while encryption must be performed sequentially, decryption can be parallelized. The first IV is a public value and all other blocks use a ciphertext as an IV, which are public. This can make decryption faster than other block cipher modes of operation..

Tutorial_3/3

Complete the table---Comment on IV Security

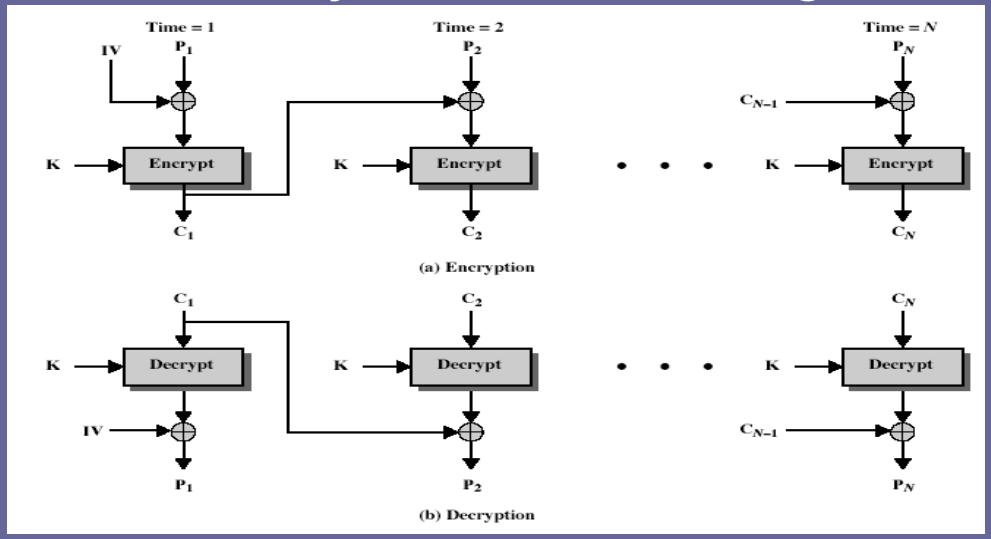
CBC	
CFB	
OFB	
CTR	

An initialization vector (IV) : To randomize encryption and hence to produce distinct ciphertexts even if the same plaintext is encrypted multiple times, without the need for a slower re-keying process.

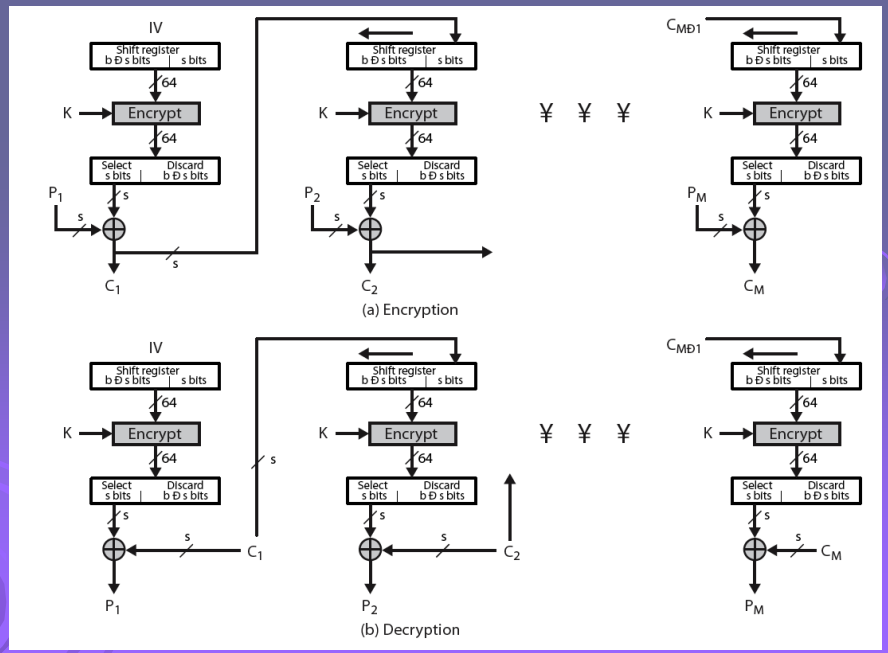
IV has different security requirements hence usually does not need to be secret. However, in most cases, it is important that IV is never reused under the same key.

An initialization vector (IV) :

For 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.

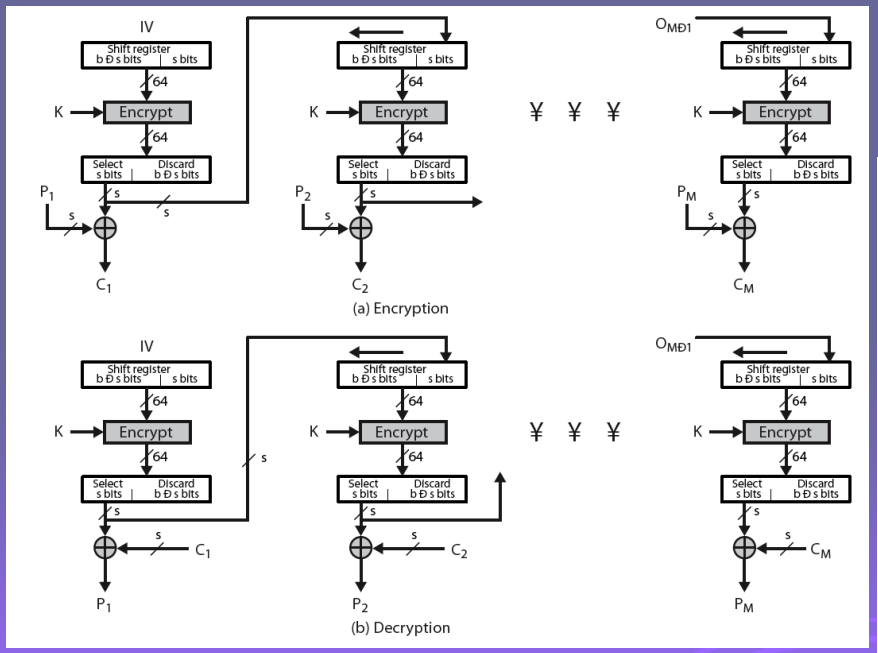


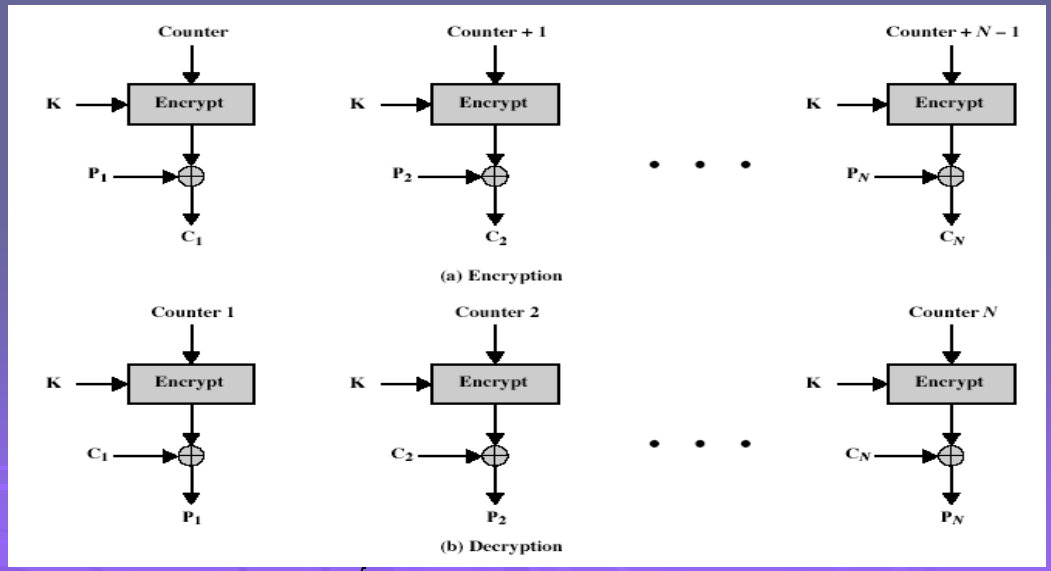
In CBC mode, IV must be unpredictable at encryption time; If an attacker knows the IV (or the previous block of ciphertext) before he specifies the next plaintext, he can check his guess about plaintext which is known as **TLS CBC IV attack**.



An initialization vector (IV) :

For OFB and CTR, reusing an IV completely destroys security. This can be seen because both modes effectively create a bitstream that is XORed with the plaintext, and this bitstream is dependent on the password and IV only. Reusing a bitstream destroys security.





	ECB	CBC	CFB	OFB	CTR
Security	Low	High	High	High	medium
Parallelism	Yes	No	No	No	Yes
Decrypting	Yes	Yes	No	No	No
random access	Yes	No	No	No	Yes
Speed	Yes	No	No	No	Yes
Complexity	No	Yes	Yes	Yes	No