

Cryptography and Network Security

Chapter 6

Fifth Edition

by William Stallings

Lecture slides by Lawrie Brown

Chapter 6 – Block Cipher Operation

Many savages at the present day regard their names as vital parts of themselves, and therefore take great pains to conceal their real names, lest these should give to evil-disposed persons a handle by which to injure their owners.

— *The Golden Bough, Sir James George Frazer*

Multiple Encryption & DES

- clear a replacement for DES was needed
 - theoretical attacks that can break it
 - demonstrated exhaustive key search attacks
- AES is a new cipher alternative
- prior to this alternative was to use multiple encryption with DES implementations
- Triple-DES is the chosen form

Why not Double-DES?

- could use 2 DES encrypts on each block
 - $C = E_{K_2}(E_{K_1}(P))$
- concern at time of reduction to single stage
- “meet-in-the-middle” attack
 - works whenever use a cipher twice
 - since $X = E_{K_1}(P) = D_{K_2}(C)$
 - attack by encrypting P with all keys and store
 - then decrypt C with keys and match X value
 - can show takes $O(2^{56})$ steps
 - Requires... known plaintext

Triple-DES with Two-Keys

- hence must use 3 encryptions
 - would seem to need 3 distinct keys
- but can use 2 keys with E-D-E sequence
 - $C = E_{K1}(D_{K2}(E_{K1}(P)))$
 - n.b. encrypt & decrypt equivalent in security
 - if $K1=K2$ then can work with single DES
- standardized in ANSI X9.17 & ISO8732
- no current known practical attacks
 - several proposed impractical attacks might become basis of future attacks

Triple-DES with Three-Keys

- although there are no practical attacks on two-key Triple-DES, there have been some indications
- can use Triple-DES with Three-Keys to avoid even these
 - $C = E_{K3}(D_{K2}(E_{K1}(P)))$
- has been adopted by some Internet applications, e.g., PGP, S/MIME



Modes of Operation

- block ciphers encrypt fixed size blocks
 - e.g., DES encrypts 64-bit blocks
- need some way to en/decrypt arbitrary amounts of data in practice
- NIST SP 800-38A defines 5 modes
- have **block** and **stream** modes
- to cover a wide variety of applications
- can be used with any block cipher

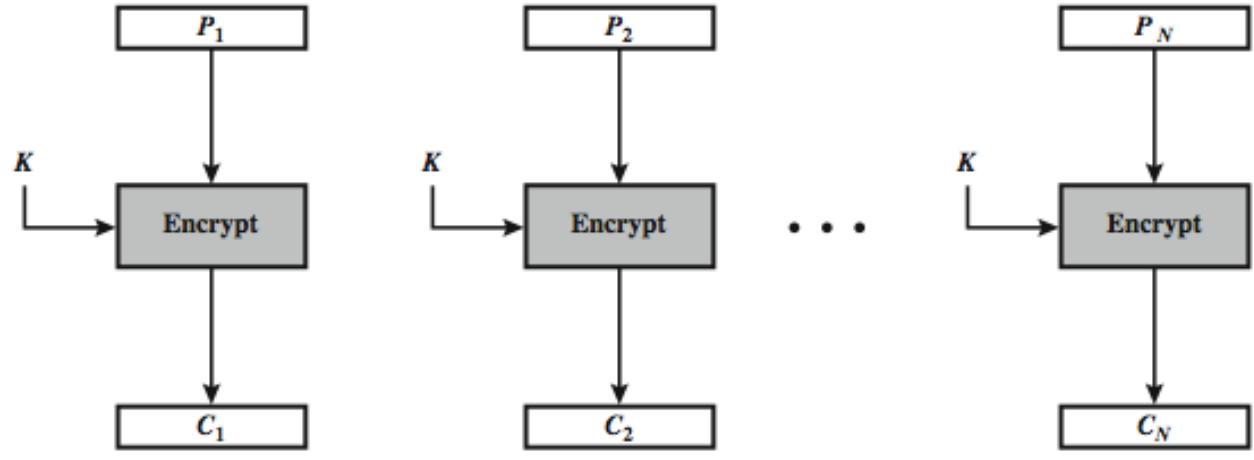
Electronic Codebook Book (ECB)

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

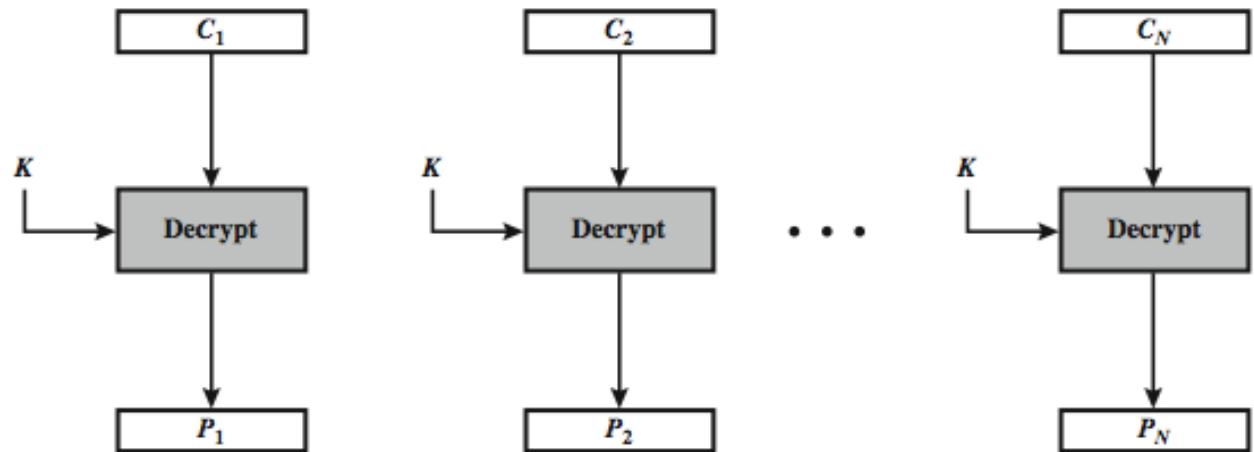
$$C_i = E_K(P_i)$$

- uses: secure transmission of single values

Electronic Codebook Book (ECB)



(a) Encryption



(b) Decryption

Advantages and Limitations of ECB

- message repetitions may show in ciphertext
 - if aligned with message block
 - particularly with data such as 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
- vulnerable to cut-and-paste attacks
- main use is sending a few blocks of data

Cipher Block Chaining (CBC)

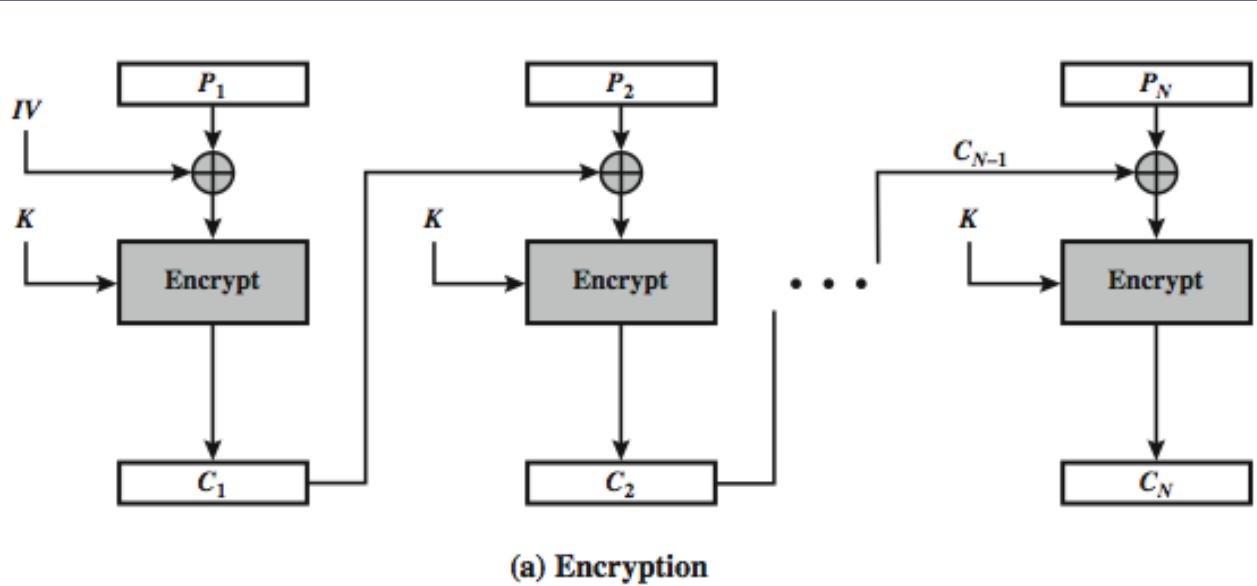
- message is broken into blocks
- linked together in encryption operation
- each previous cipher block is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process

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

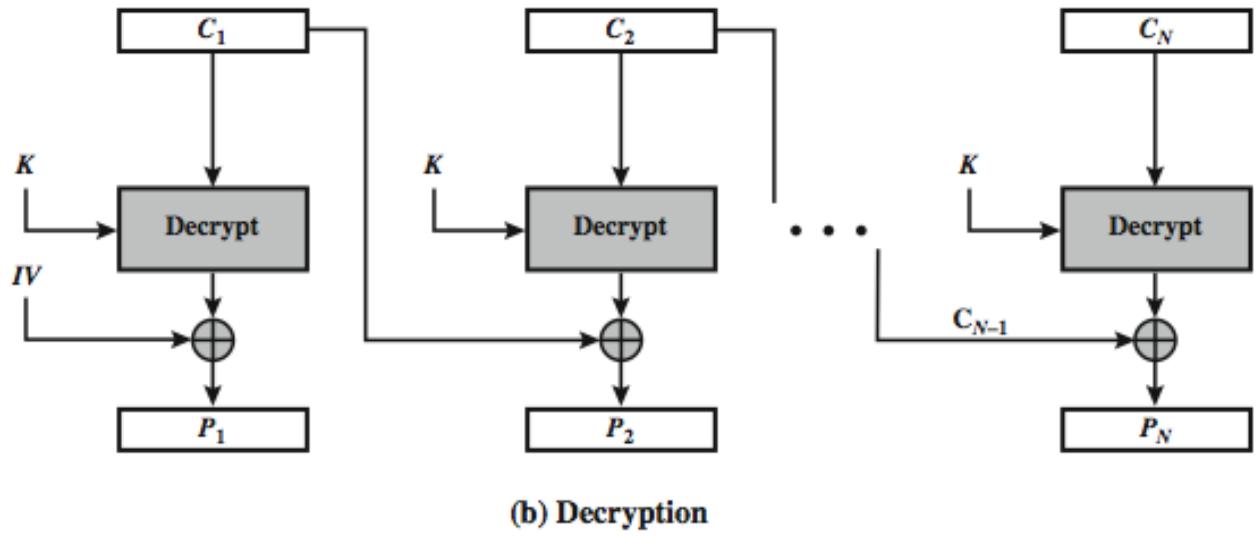
$$C_{-1} = IV$$

- IV prevents same P from making same C
- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC)



(a) Encryption



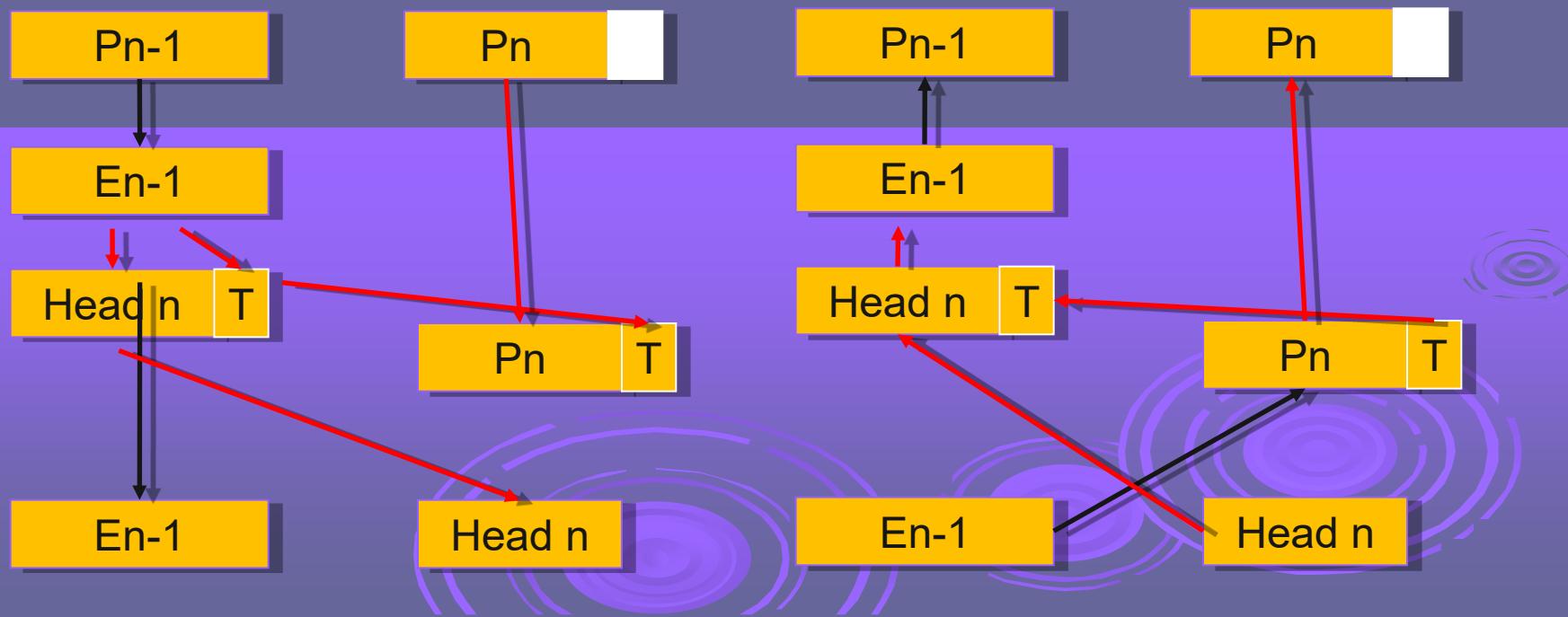
(b) Decryption

Message Padding

- at end of message must handle a possible last short block
 - which is not as large as blocksize of cipher
 - pad either with known non-data value
 - e.g., nulls
 - or pad last block along with count of pad size
 - e.g., [b1 b2 b3 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

Ciphertext Stealing

- Use to make ciphertext length same as plaintext length
- Requires more than one block of ptxt



Advantages and Limitations of CBC

- a ciphertext block depends on **all** blocks before it
- any change to a block affects all following ciphertext blocks... avalanche effect
- need **Initialization Vector (IV)**
 - which must be known to sender & receiver
 - if sent in clear, attacker can change bits of first block, by changing corresponding bits of IV
 - hence IV must either be a fixed value (as in EFTPOS)
 - or derived in way hard to manipulate
 - or sent encrypted in ECB mode before rest of message
 - or message integrity must be checked otherwise

Stream Modes of Operation

- block modes encrypt entire block
- may need to operate on smaller units
 - real time data
- convert block cipher into stream cipher
 - cipher feedback (CFB) mode
 - output feedback (OFB) mode
 - counter (CTR) mode
- use block cipher as some form of **pseudo-random number** generator... Vernam cipher

Cipher FeedBack (CFB)

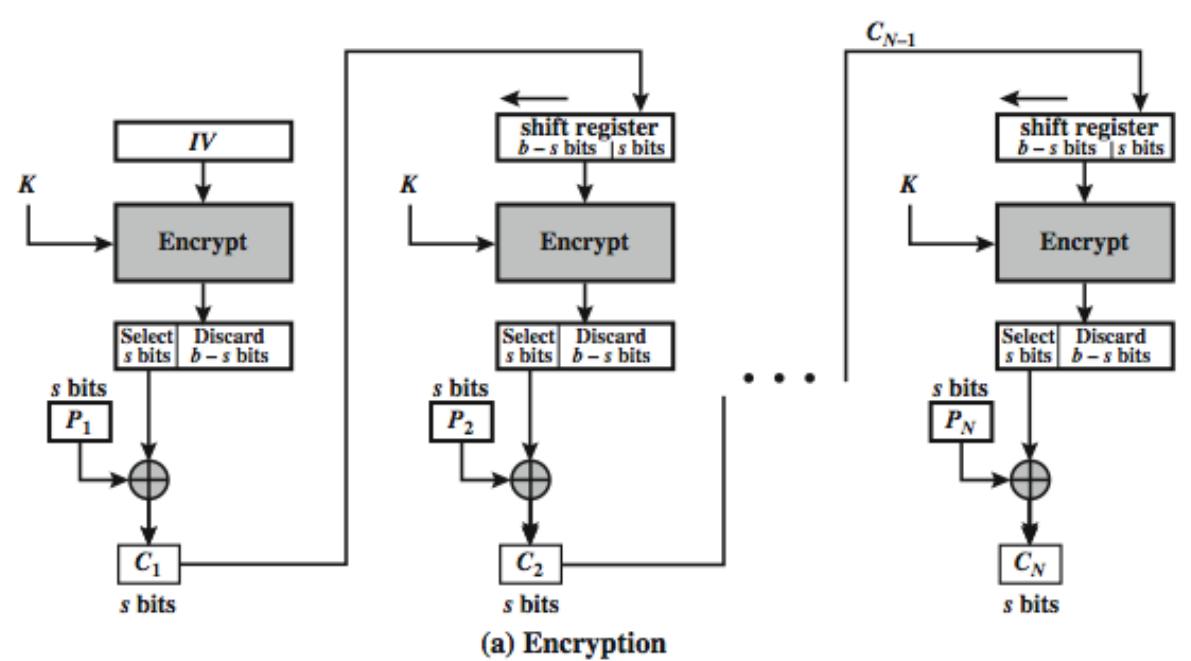
- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bits (1,8, 64 or 128 etc) to be feed back
 - denoted CFB-1, CFB-8, CFB-64, CFB-128, etc.
- most efficient to use all bits in block (64 or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

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

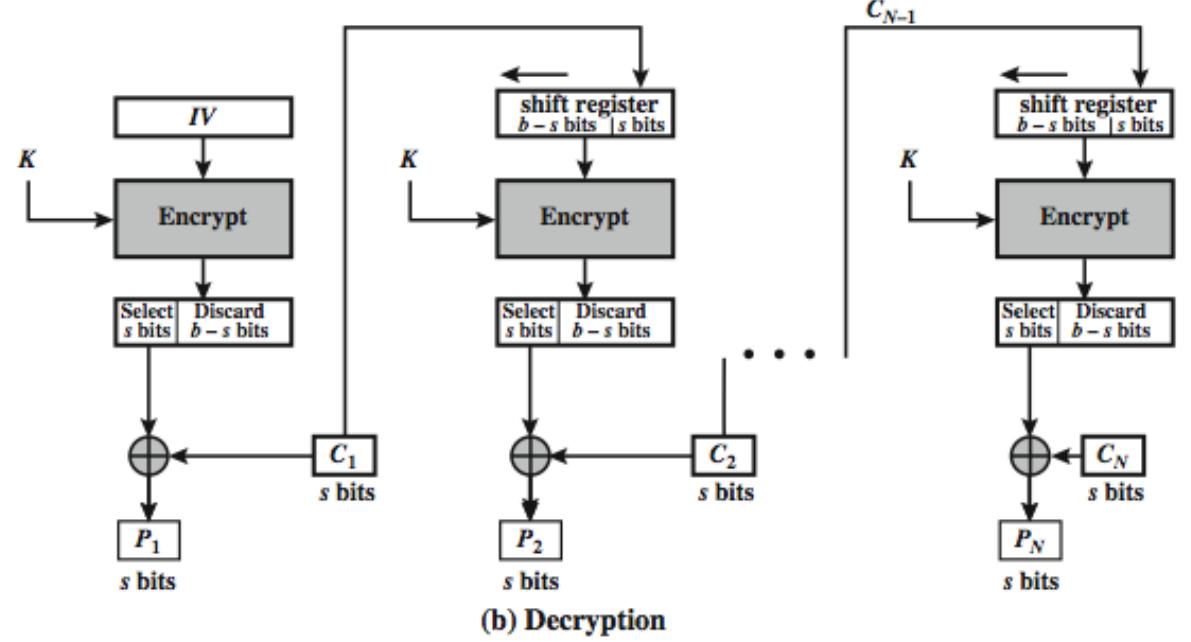
$$C_{-1} = \text{IV}$$

- uses: stream data encryption, authentication

s-bit Cipher FeedBack (CFB-s)



(a) Encryption



(b) Decryption

Advantages and Limitations of CFB

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

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)

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

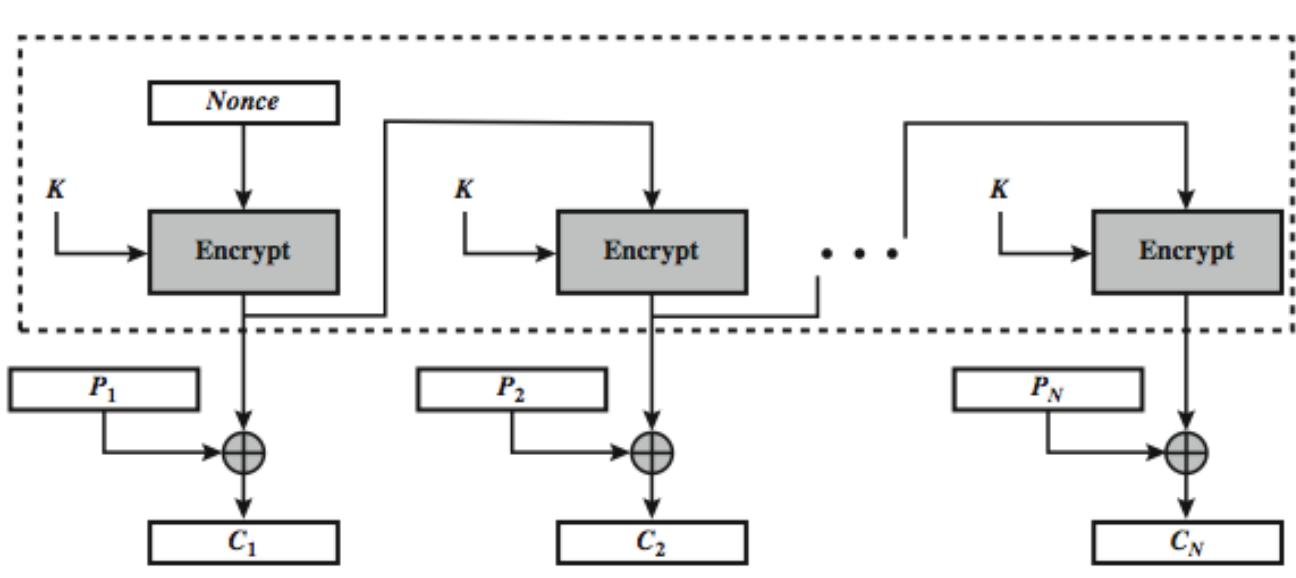
$$C_i = P_i \text{ XOR } O_i$$

$$O_{-1} = \text{IV}$$

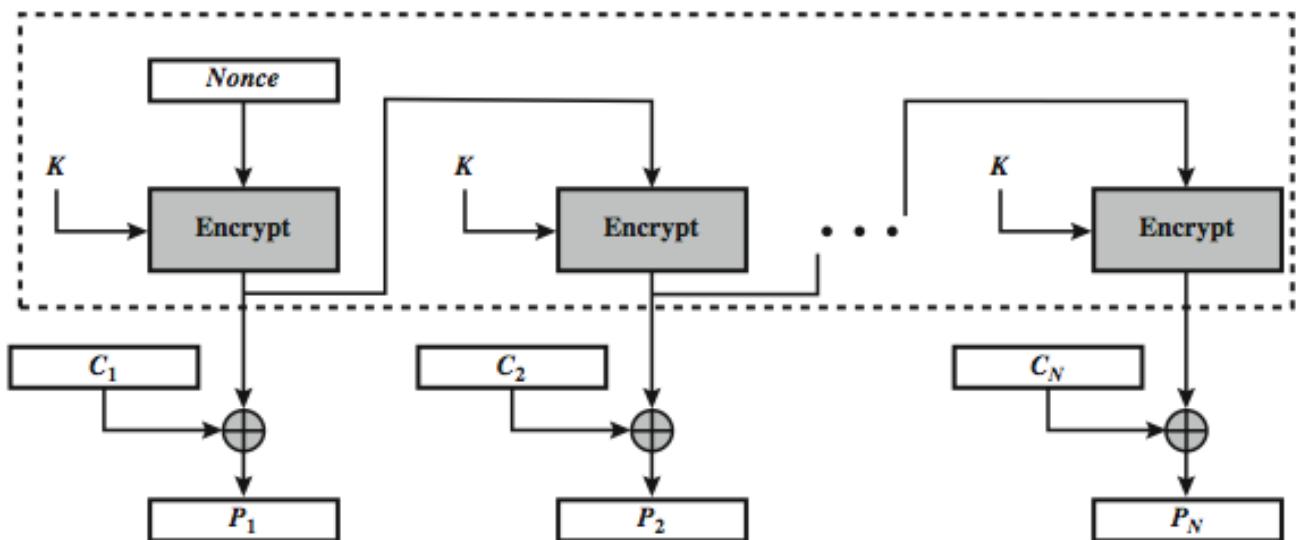
- feedback is independent of message
- can be computed in advance
- uses: stream encryption on noisy channels

Why noisy channels?

Output FeedBack (OFB)



(a) Encryption



(b) Decryption

Advantages and Limitations of OFB

- needs an IV which is unique for each use
 - if ever reuse attacker can recover outputs...
 - OTP
- can pre-compute
- bit errors do not propagate
- more vulnerable to message stream modification...
 - change arbitrary bits by changing ciphertext
- sender & receiver must remain in sync
- only use with full block feedback
 - subsequent research has shown that only **full block feedback** (ie CFB-64 or CFB-128) should ever be used

Counter (CTR)

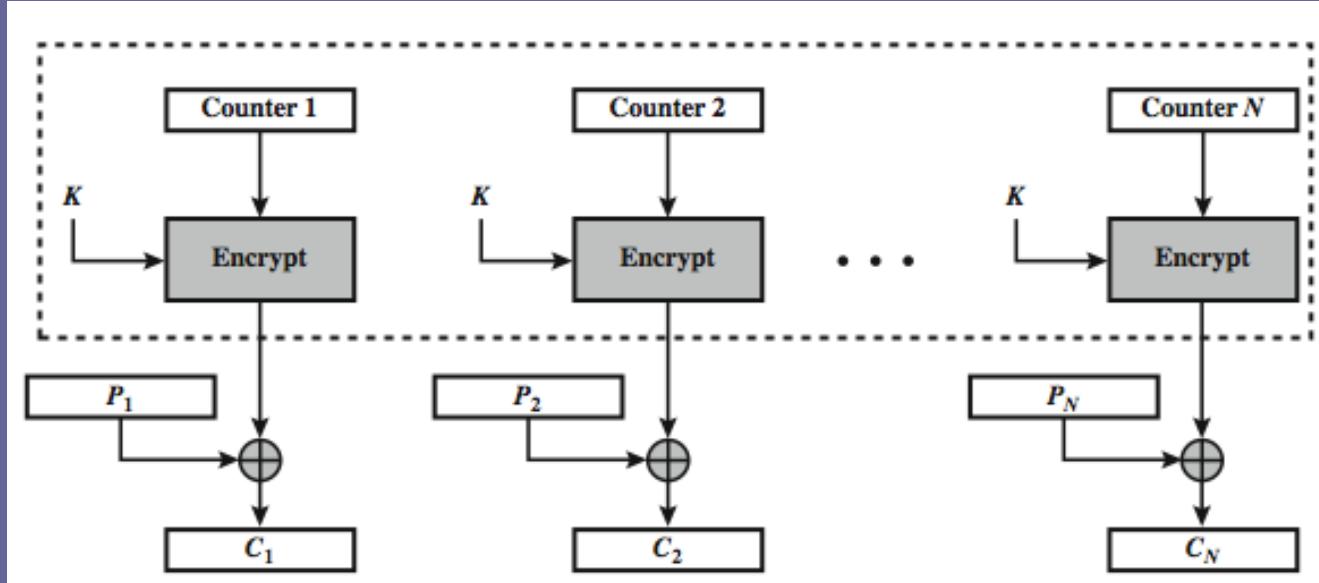
- a “new” mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value

$$O_i = E_K(i)$$

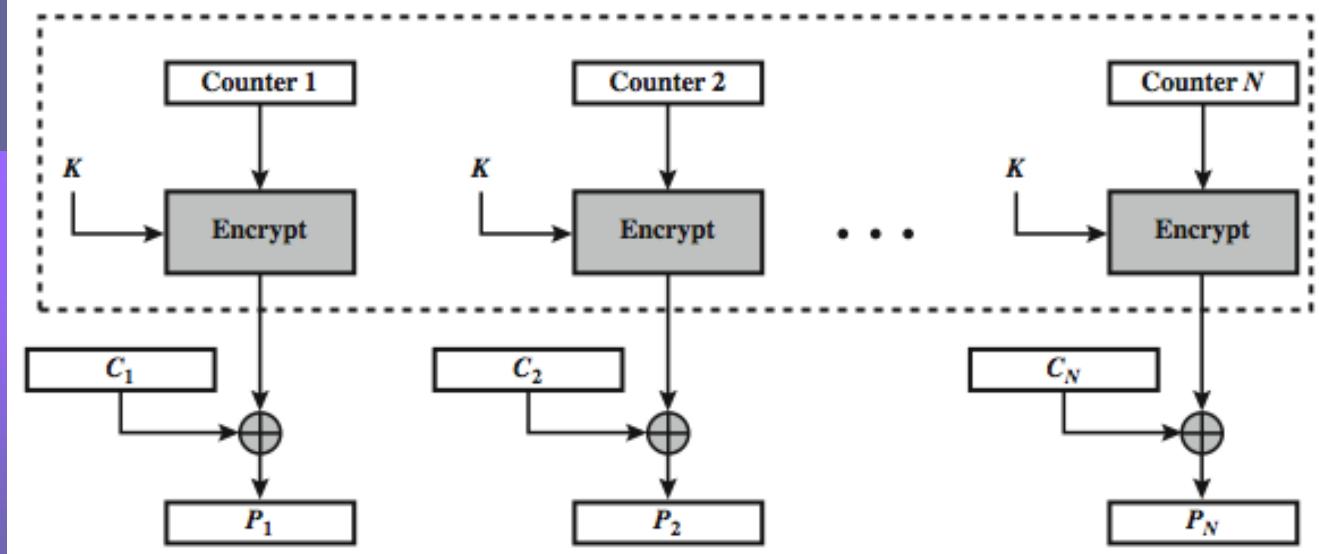
$$C_i = P_i \text{ XOR } O_i$$

- must have a different key & counter value for every plaintext block (never reused)
 - again, OTP issue
- uses: high-speed network encryptions

Counter (CTR)



(a) Encryption

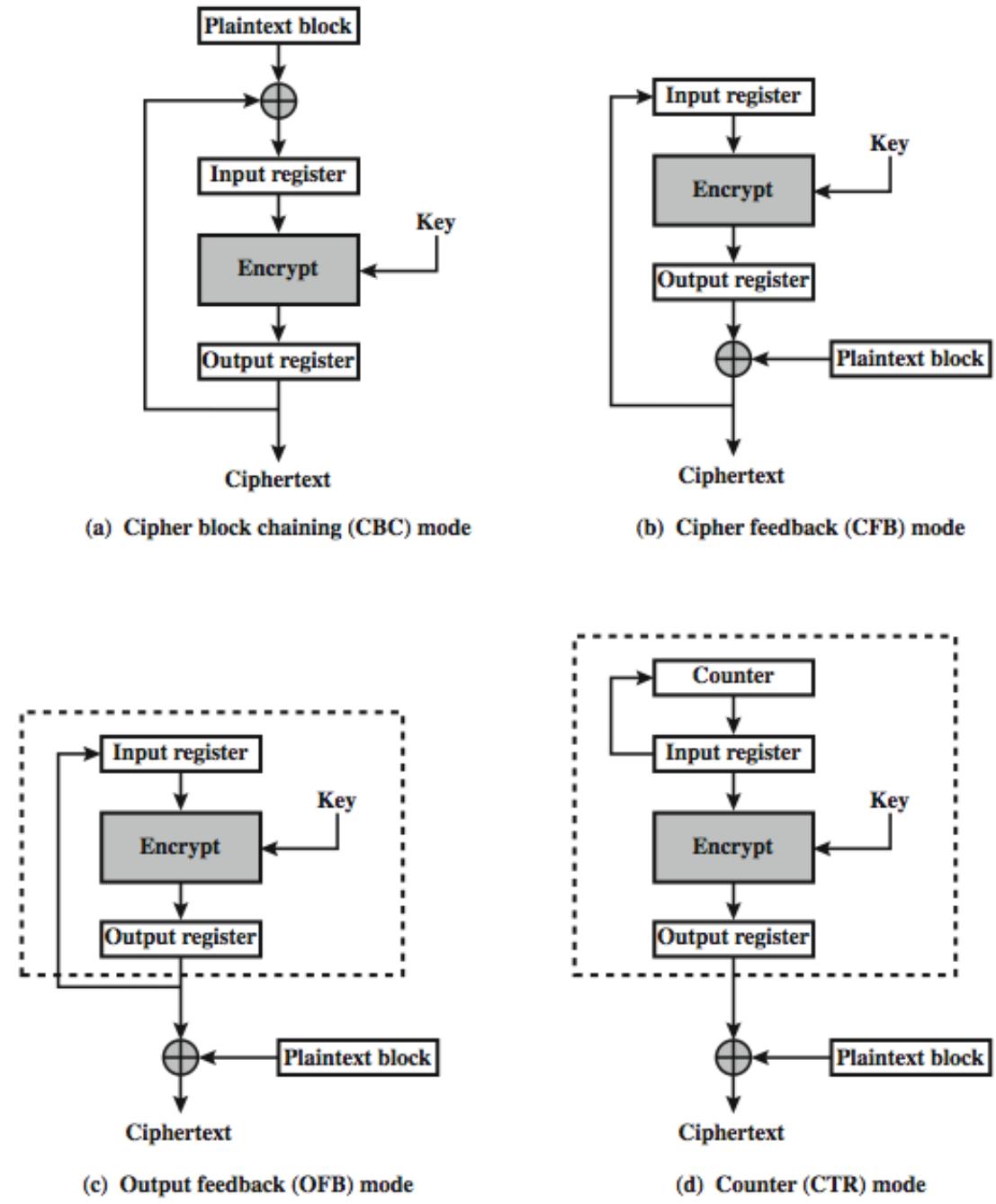


(b) Decryption

Advantages and Limitations of CTR

- efficiency
 - can do **parallel** encryptions in h/w or s/w
 - can preprocess in advance of need
 - good for bursty high speed links
- random access to encrypted data blocks
- provable security (good as other modes)
- never have cycle less than 2^b
- but must ensure never reuse key/counter values, otherwise could break (cf OFB)

Feedback Character- istics



XTS-AES Mode

- need mode for block oriented storage
 - No extra room in sector – data only
 - Disk addressed by sector number
 - Encryption can only take key externally
 - Encryption can also use sector#, block#
- Access to any sector should be independent of other sectors
- Must prevent attack that copies sector to unused sector, then requests decryption

XTS-AES Mode

- new mode, for block oriented storage use
 - in IEEE Std 1619-2007
- concept of tweakable block cipher
- different requirements to transmitted data
- uses AES twice for each block

$$T_j = E_{K2}(i) \text{ XOR } \alpha^j$$

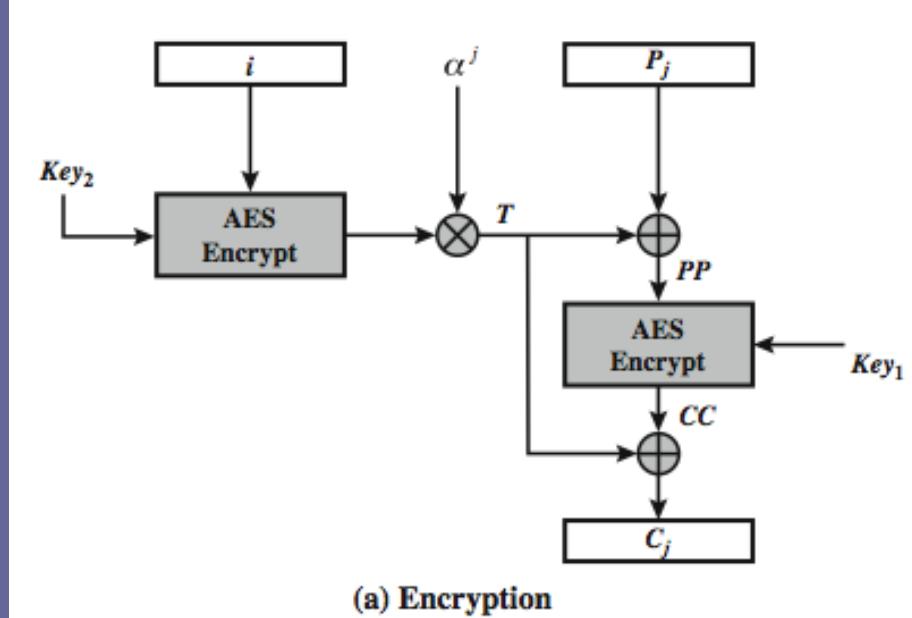
$$C_j = E_{K1}(P_j \text{ XOR } T_j) \text{ XOR } T_j$$

where i is tweak & j is sector no

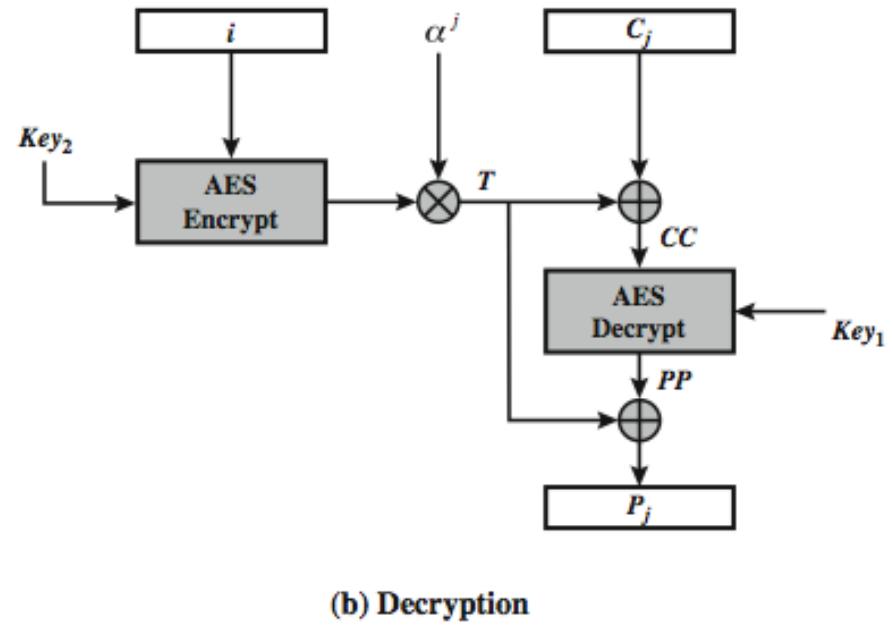
- each sector may have multiple blocks

XTS-AES Mode per block

Key whitening applied by XOR
With “tweak” that depends on
- sector
- block
- second key
Makes attacks more difficult
Makes operations depend on
data location

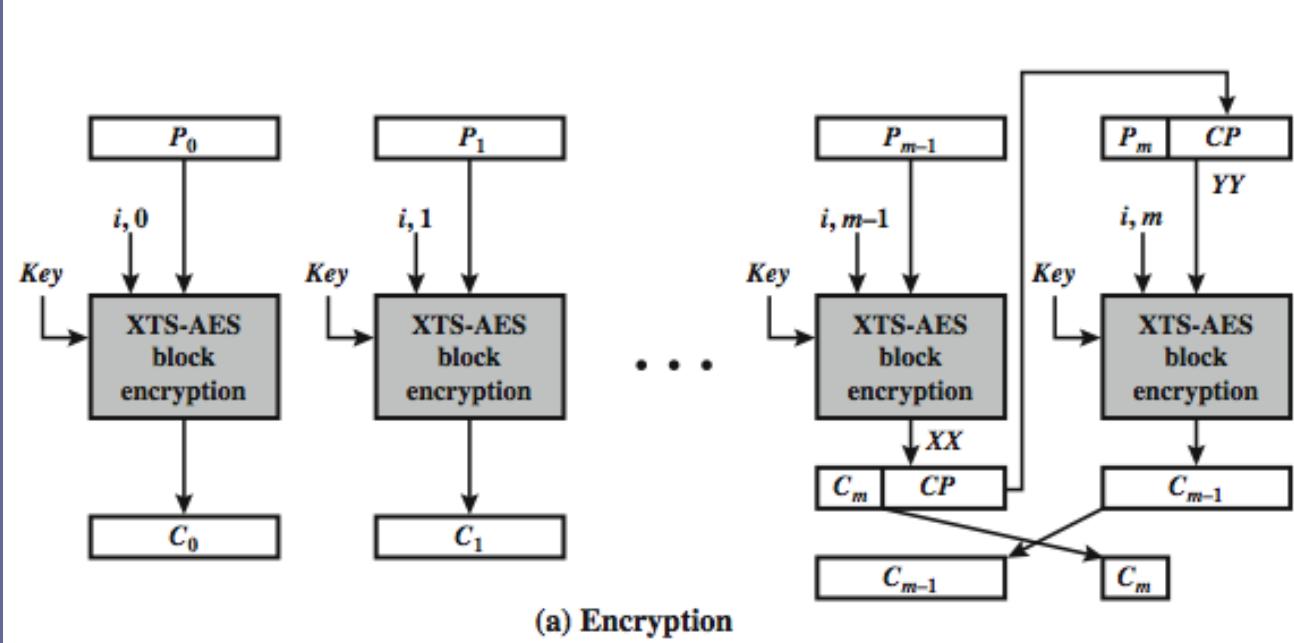


(a) Encryption

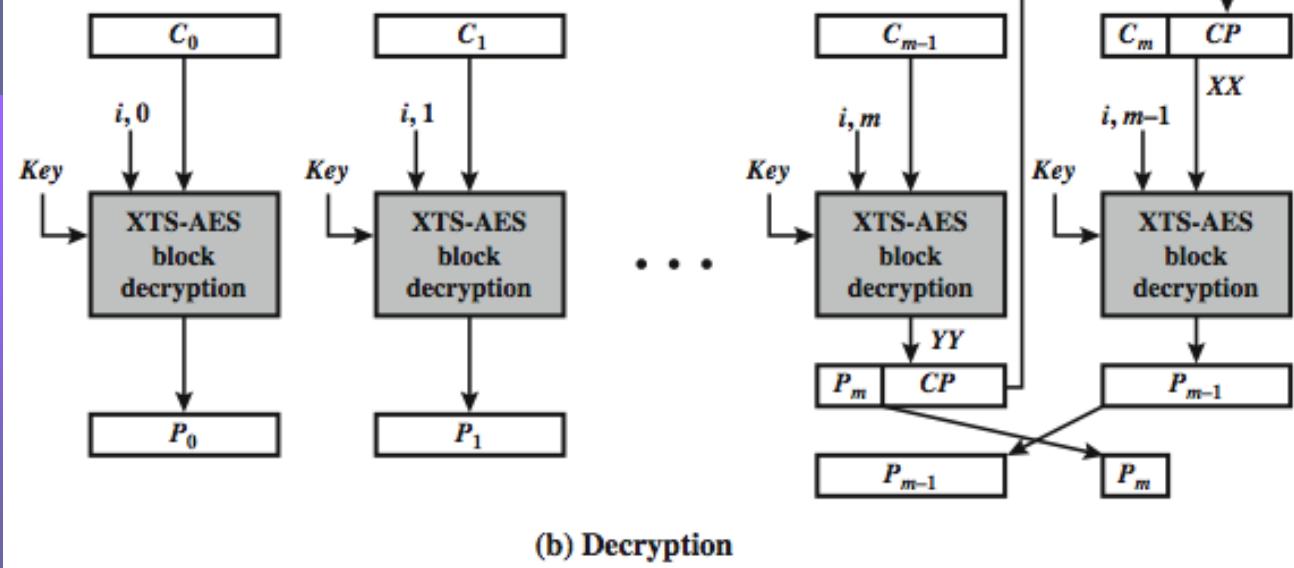


(b) Decryption

XTS-AES Mode Overview



(a) Encryption



(b) Decryption

Advantages and Limitations of XTS-AES

- efficiency
 - can do parallel encryptions in h/w or s/w
 - random access to encrypted data blocks
- has both nonce & counter
- addresses security concerns related to stored data

Summary

- Multiple Encryption & Triple-DES
- Modes of Operation
 - ECB, CBC, CFB, OFB, CTR, XTS-AES
- Next – Stream ciphers (Ch 7), then hash functions (Ch 11)

