Cryptography and Network Security Chapter 6

Fifth Edition

by William Stallings

Lecture slides by Lawrie Brown

(with edits by RHB)

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***

Outline

- Multiple Encryption & Triple-DES
- Modes of Operation
  - ECB, CBC, CFB, OFB, CTR, XTS-AES

Multiple Encryption & DES

- clear a replacement for DES was needed
  - theoretical attacks that can break it
  - demonstrated exhaustive key search attacks
- AES was a new alternative cipher
- but an alternative was needed before AES
- the alternative was to use multiple DES encryptions, making for a more complex cipher
- Triple-DES was the chosen form

Double-DES?

- could use 2 DES encrypts on each block
  - $C = E_{K2}(E_{K1}(P))$
- issue of reduction to single stage
- and have “meet-in-the-middle” attack
  - works whenever use a cipher twice
  - Since there is an $X = E_{K1}(P) = D_{K2}(C)$
  - attack by encrypting P with all keys and store
  - and decrypt C with all keys and match X value
  - can show takes $O(2^{56})$ steps

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 are no practical attacks on two-key Triple-DES, have some doubts
- 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, eg PGP, S/MIME

Cryptographic algorithms and the GROUP PROPERTY

- A block cipher is a just big substitution (on 64-bit data blocks in the case of DES).
- The product of two substitutions is just another substitution.
- So, is the product of two DES en/de-cryptions just another DES en/de-cryption?
- NO! A DES en/de-cryption is specified by a (56-bit) **KEY**.
- For a product of two arbitrary DES en/de-cryptions, there is **NO KEY** that specifies the substitution.
- So, multiple DES en/de-cryptions define a bigger class of substitutions on 64-bit data blocks.
- Therefore multiple DES is strictly more powerful.

Modes of Operation

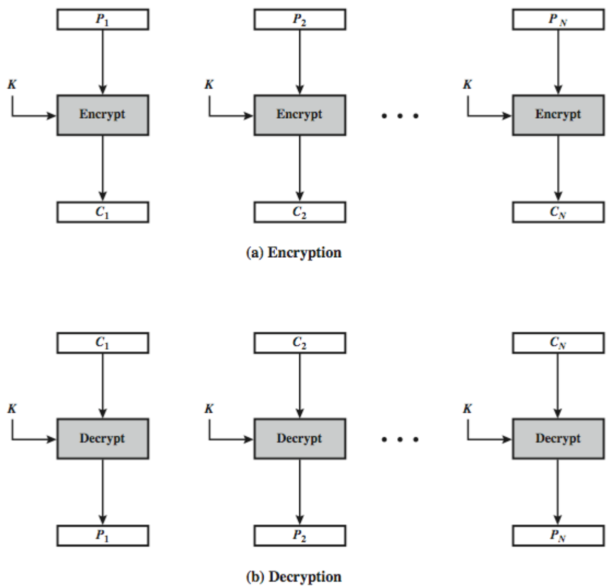
- block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks with 56-bit key
- need some way to en/de-crypt arbitrary amounts of data in practice
- NIST SP 800-38A defines 5 modes
- have **block** and **stream** modes
- they cover a wide variety of applications
- can be used with any block cipher

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

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 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

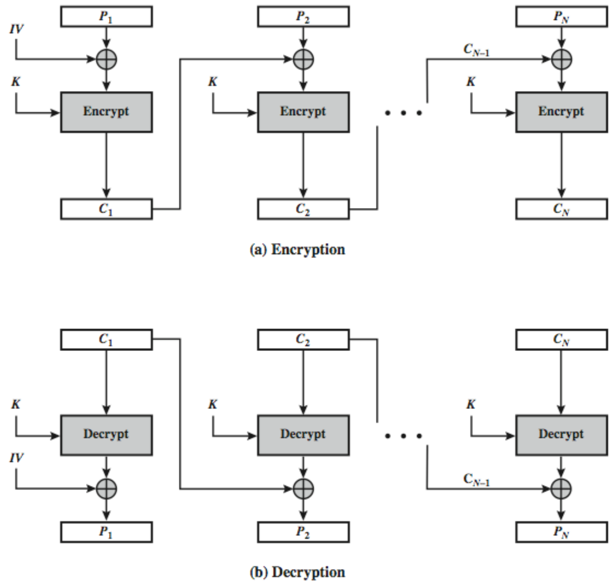
Cipher Block Chaining (CBC)

- 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

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

  $C_{-1} = IV$
- 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 (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** blocks before it
- 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 (as in EFTPOS)
  - or must be sent encrypted in ECB mode before rest of message

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

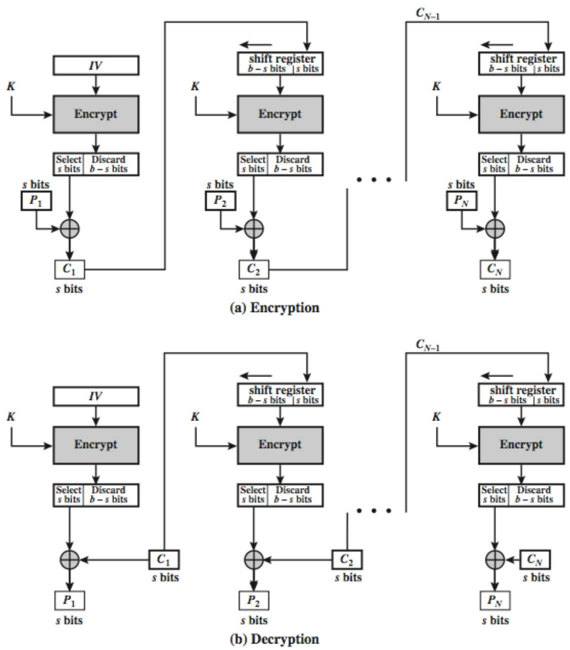
Cipher FeedBack (CFB)

- message is treated as a stream of bits
- added to the output of the block cipher
- result is fed 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)

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

  $C_{-1} = IV$
- uses: stream data encryption, authentication

s-bit Cipher FeedBack (CFB-s)



(a) Encryption

(b) Decryption

Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- limitation is the need to stall while do block encryption after every n-bits
- block cipher is used in **encryption** mode at **both** ends to yield ps-random bitstream
- errors propagate for several blocks after the error (but not indefinitely)

Output FeedBack (OFB)

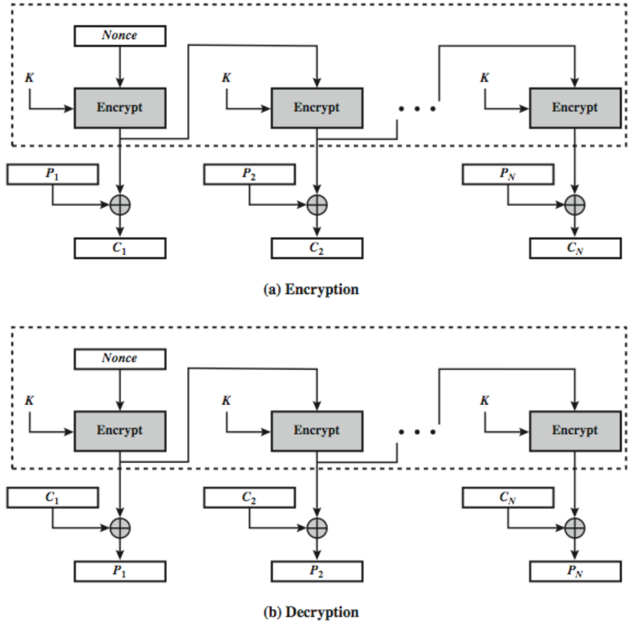
- message is treated as a stream of bits
- output of cipher is added to message
- output is then fed back (hence name)
- feedback is independent of message
- can be computed in advance

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

  $C_i = P_i \text{ XOR } O_i$

  $O_{-1} = IV$ (Nonce)
- uses: stream encryption on 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
- bit errors do not propagate
- more vulnerable to message stream modification
- 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

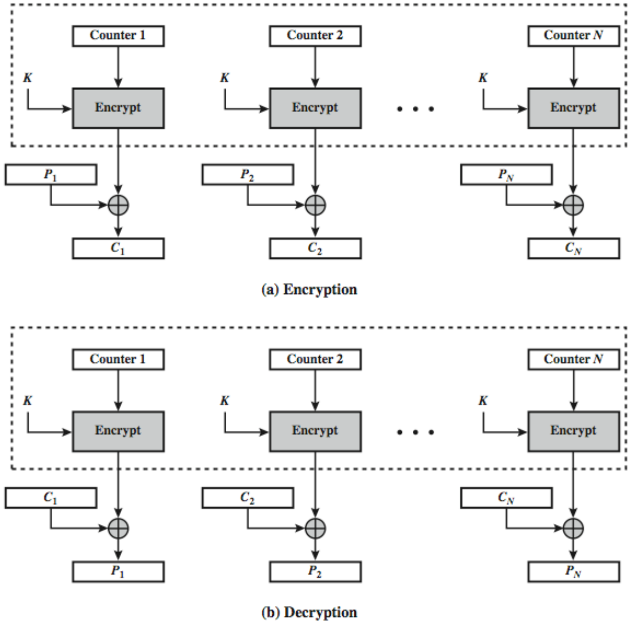
Counter (CTR)

- a newer mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

  $O_i = E_K(i)$

  $C_i = P_i \text{ XOR } O_i$
- uses: high-speed network encryptions

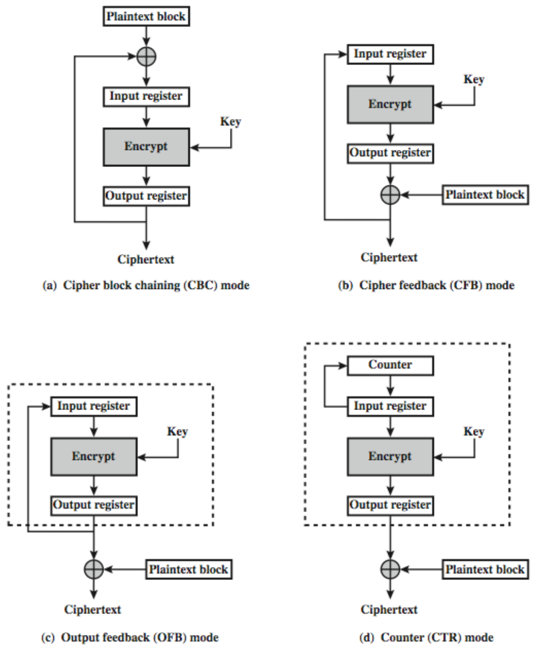
Counter (CTR)



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)
- but must ensure never reuse key/counter values, otherwise could break (cf. OFB)

Feedback Character-istics



XTS-AES Mode

- new mode, for block oriented storage use
  - in IEEE Std 1619-2007
- concept of tweakable block cipher
- basic idea:

  `T` is tweak, `H` is a hash function

  `C = H(T) XOR E(K, H(T) XOR P)`
- different requirements to transmitted data

XTS-AES Mode

- in XTS-AES, the tweak T_j is

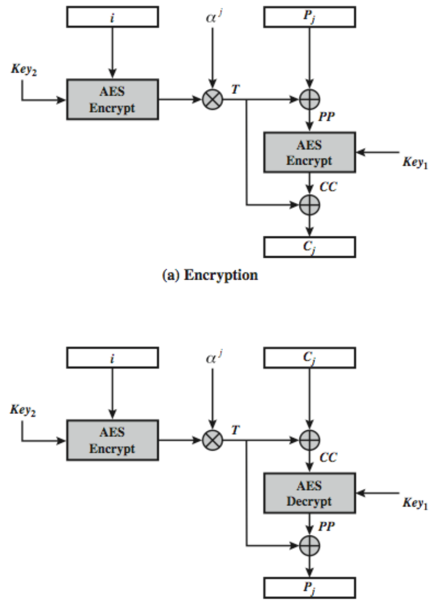
  $T_j = E_{K2}(i) \times \alpha^j$ (× and exp. in GF(2^{128}))

  where i is (relatively arbitrary, non-secret) tweak base of data sector, and j is block no.
- now, use AES twice for each block

  $T_j = E_{K2}(i) \times \alpha^j$ (× and exp. in GF(2^{128}))

  $C_j = E_{K1}(P_j \text{ XOR } T_j) \text{ XOR } T_j$
- each sector will usually have many blocks

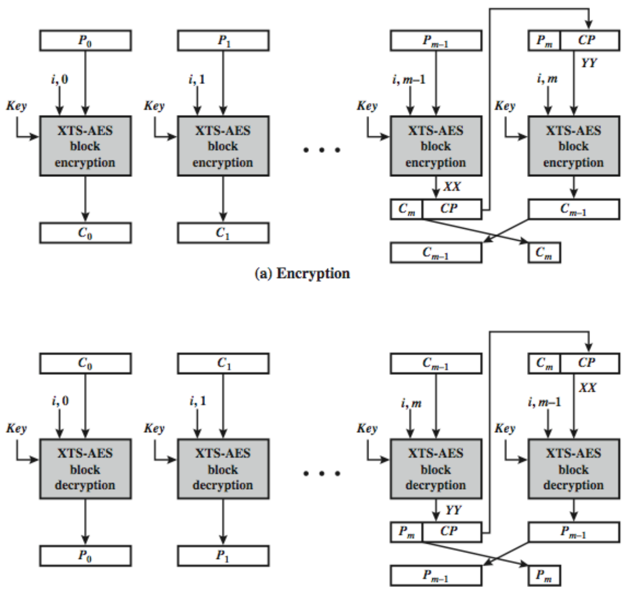
XTS-AES Mode per block



(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 concerned related to stored data