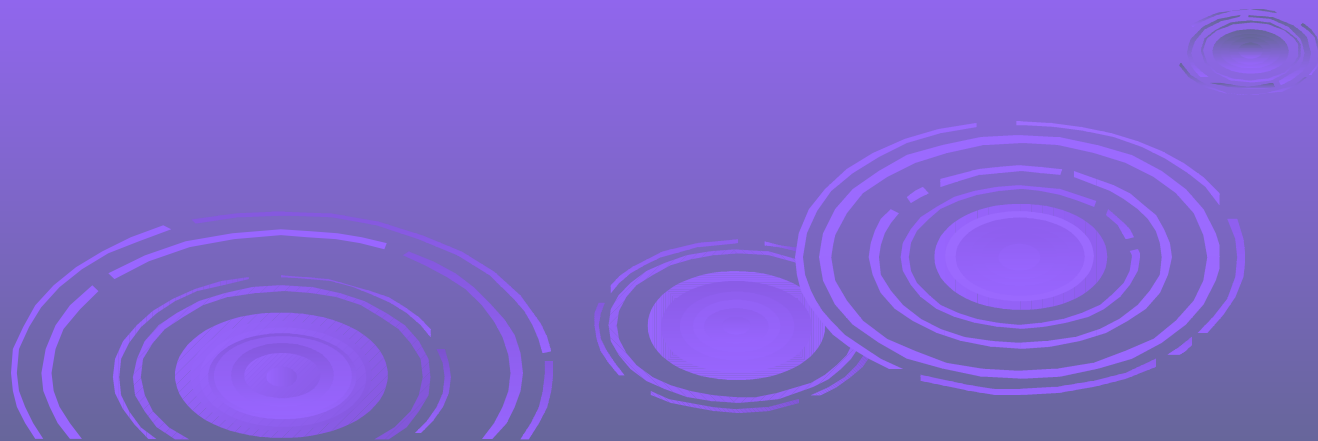


Cryptography and Network Security

Chapter 6



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



Double-DES?

- could use 2 DES encrypts on each block
 - $C = E_{K_2} (E_{K_1} (P))$
- issue of reduction to single stage
- and have “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

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
 - $C = E_{K1}(D_{K2}(E_{K1}(P)))$
 - nb 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

Triple-DES with Three-Keys

- although there are no practical attacks on two-key Triple-DES there are 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, eg PGP, S/MIME



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 amounts of data in practise
- **ANSI X3.106-1983 Modes of Use** (now FIPS 81) defines 4 possible modes
- subsequently 5 defined for AES & DES
- have **block** and **stream** modes

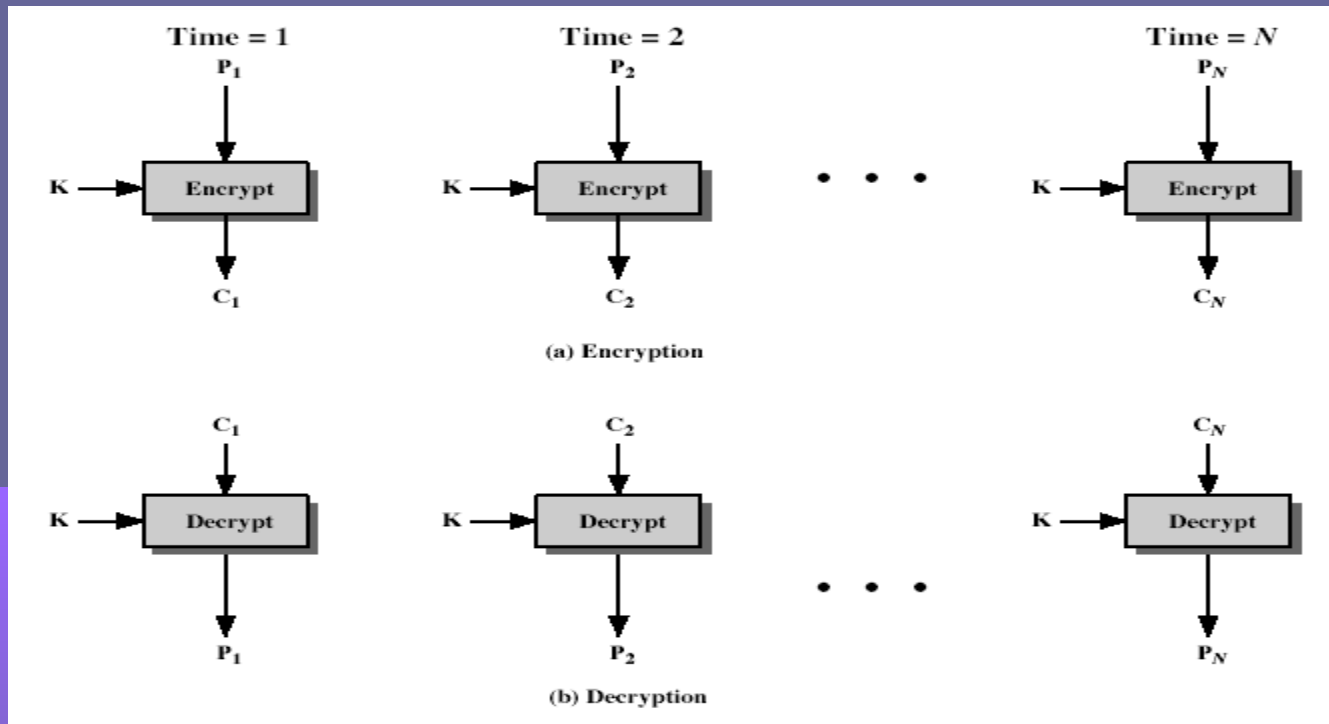
Electronic Codebook Book (ECB)

- 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

$$C_i = \text{DES}_{K1}(P_i)$$

- 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

Cipher Block Chaining (CBC)

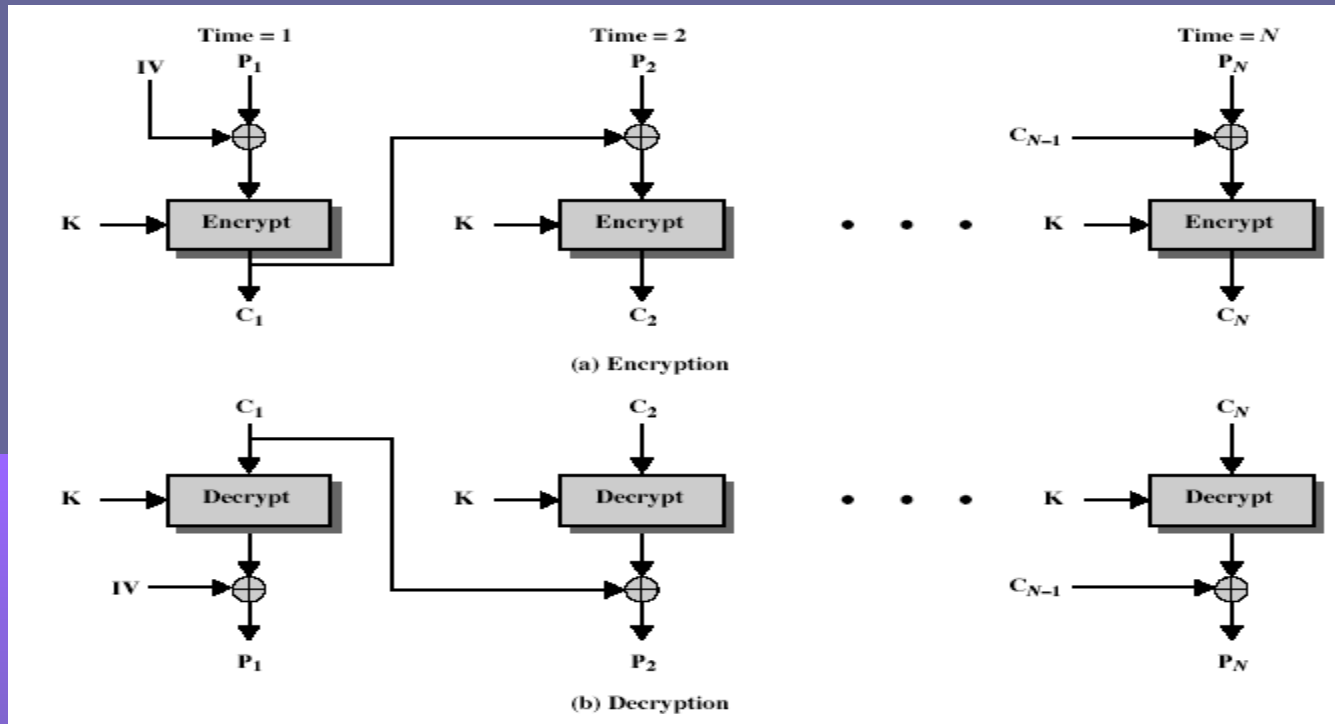
- message is broken into blocks
- linked together in encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process

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

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

- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC)



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

Cipher FeedBack (CFB)

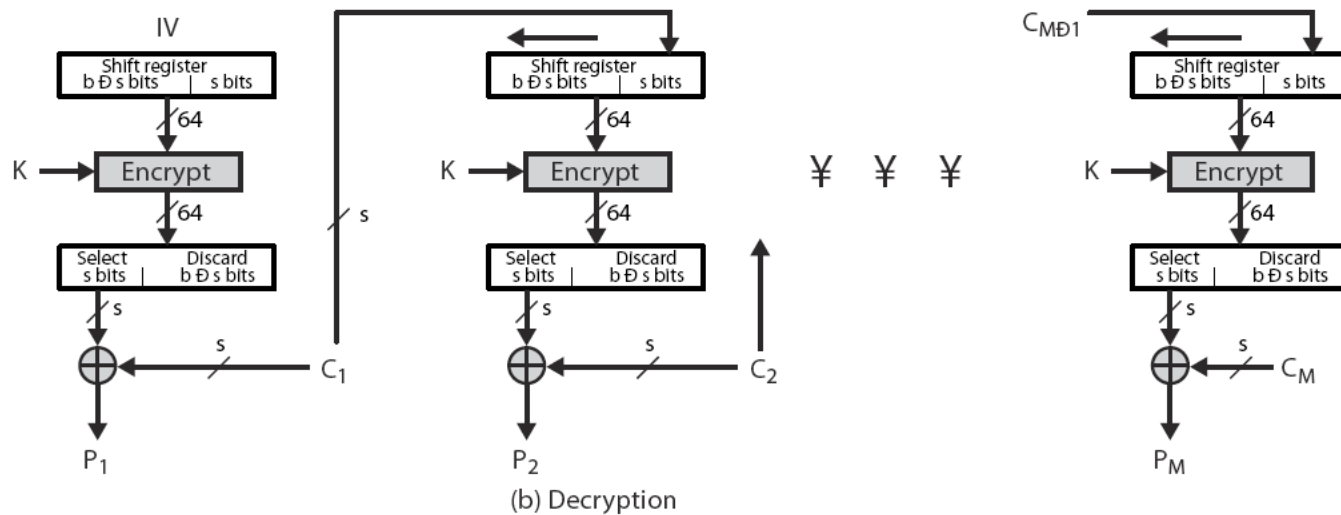
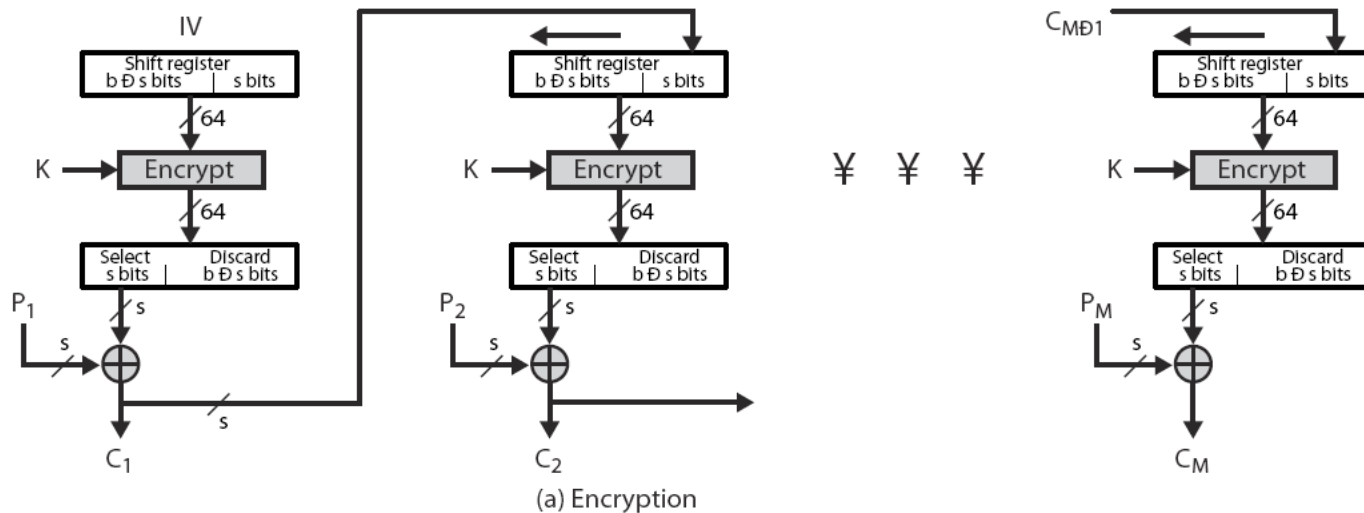
- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8, 64 or 128 etc) to be feed back
 - denoted CFB-1, CFB-8, CFB-64, CFB-128 etc
- most efficient to use all bits in block (64 or 128)

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

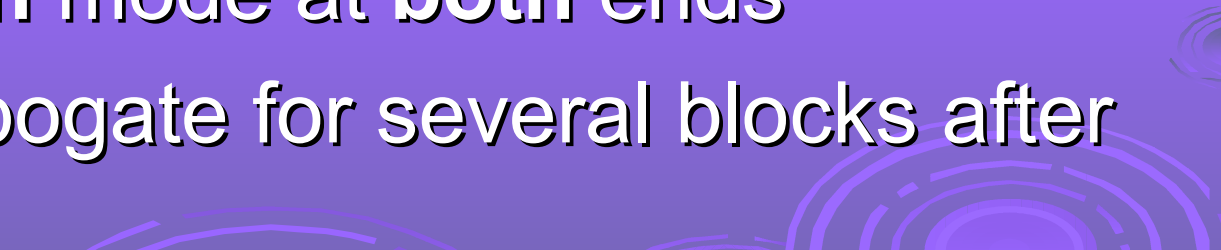
$$C_{-1} = \text{IV}$$

- uses: stream data encryption, authentication

Cipher FeedBack (CFB)



Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
 - most common stream mode
 - limitation is need to stall while do block encryption after every n-bits
 - note that the block cipher is used in **encryption** mode at **both** ends
 - errors propagate for several blocks after the error
- 
- A decorative graphic consisting of several concentric circles in a light blue color, located in the bottom right corner of the slide.

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

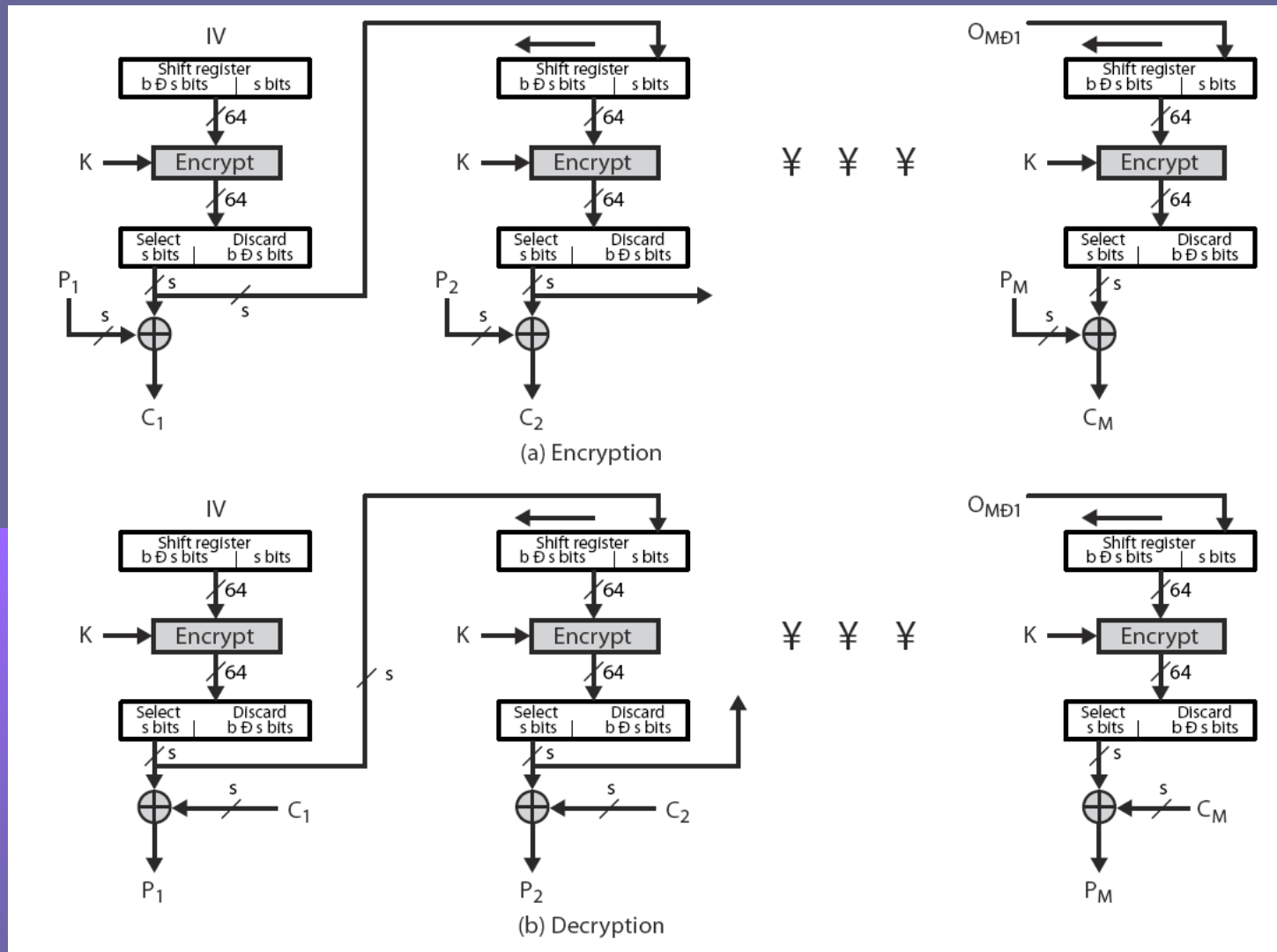
$$C_i = P_i \text{ XOR } O_i$$

$$O_i = \text{DES}_K(O_{i-1})$$

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

- 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
- 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)

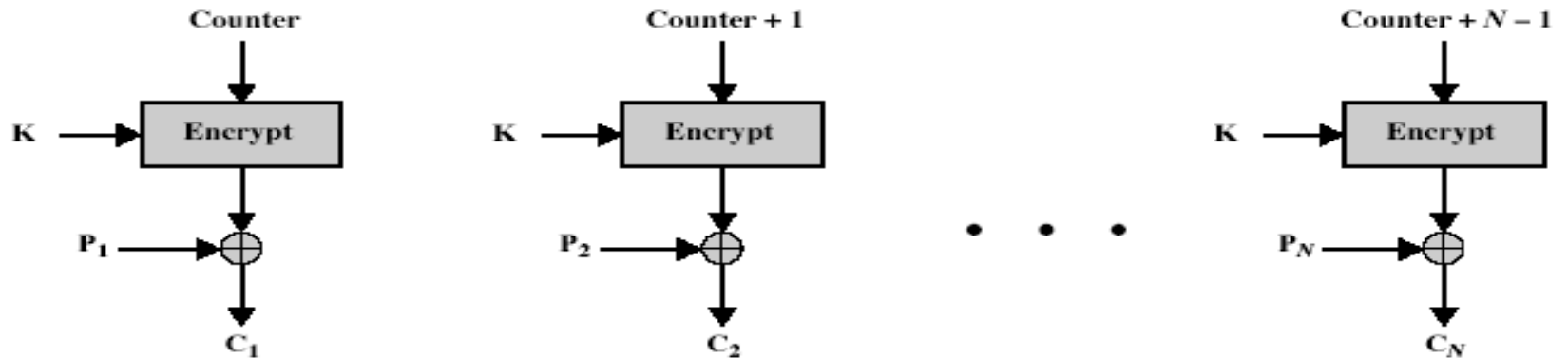
- a “new” mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

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

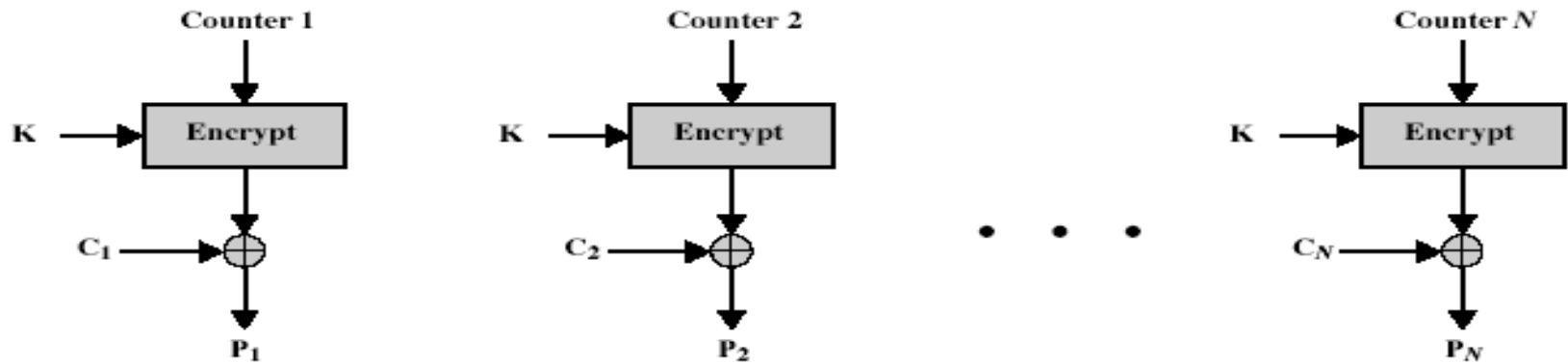
$$O_i = \text{DES}_{K1}(i)$$

- uses: high-speed network encryptions

Counter (CTR)



(a) Encryption



(b) Decryption

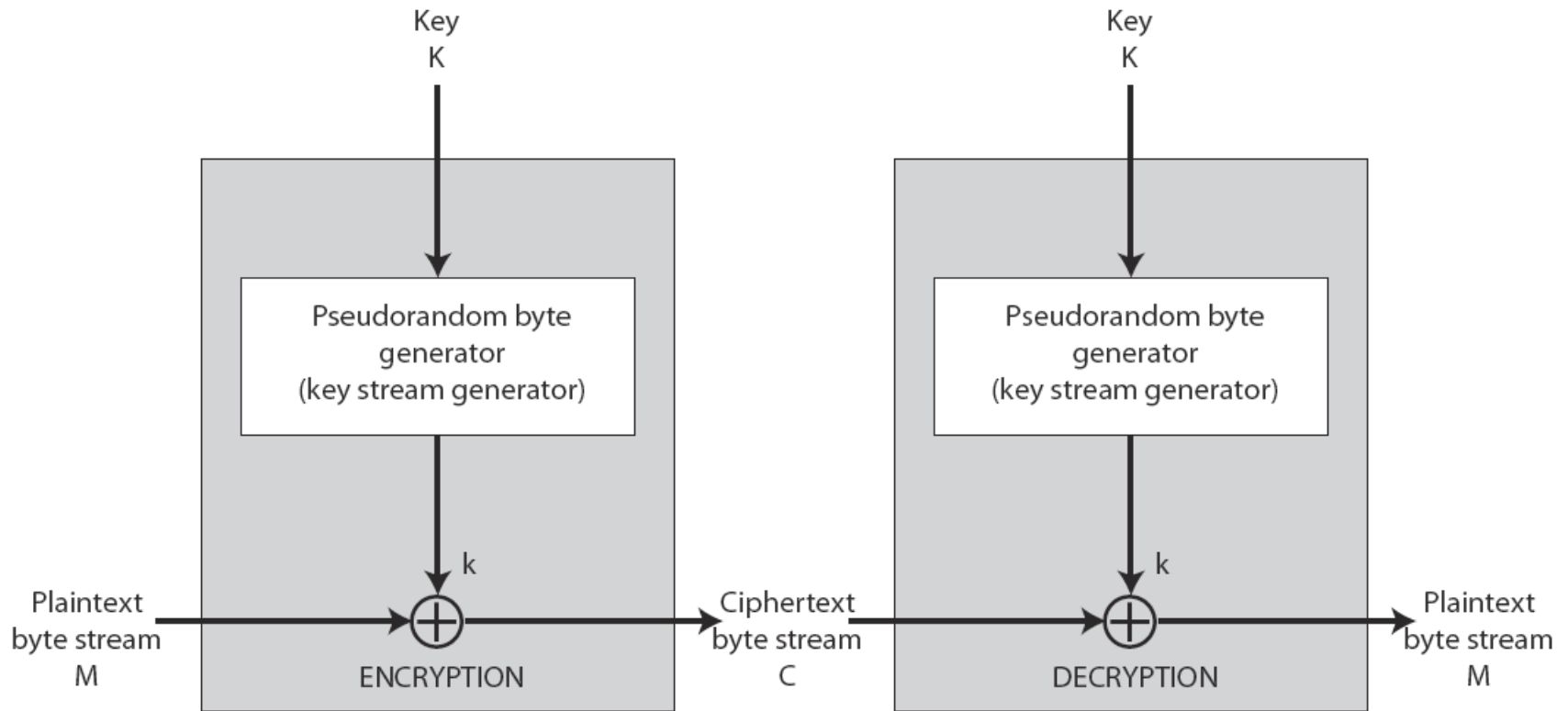
Advantages and Limitations of CTR

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

Stream Ciphers

- process message bit by bit (as a stream)
- have a pseudo random **keystream**
- combined (XOR) with plaintext bit by bit
- randomness of **stream key** completely destroys statistically properties in message
 - $C_i = M_i \text{ XOR } \text{StreamKey}_i$
- but must never reuse stream key
 - otherwise can recover messages (cf book cipher)

Stream Cipher Structure



Stream Cipher Properties

- some design considerations are:
 - long period with no repetitions
 - statistically random
 - depends on large enough key
 - large linear complexity
- properly designed, can be as secure as a block cipher with same size key
- but usually simpler & faster

RC4

- a proprietary cipher owned by RSA DSI
- another Ron Rivest design, simple but effective
- variable key size, byte-oriented stream cipher
- widely used (web SSL/TLS, wireless WEP)
- key forms random permutation of all 8-bit values
- uses that permutation to scramble input info processed a byte at a time



RC4 Key Schedule

- starts with an array S of numbers: 0..255
- use key to well and truly shuffle
- S forms **internal state** of the cipher

```
for i = 0 to 255 do
    S[i] = i
    T[i] = K[i mod keylen]
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) (mod 256)
    swap (S[i], S[j])
```

RC4 Encryption

- encryption continues shuffling array values
- sum of shuffled pair selects "stream key" value from permutation
- XOR $S[t]$ with next byte of message to en/decrypt

$i = j = 0$

for each message byte M_i

$i = (i + 1) \pmod{256}$

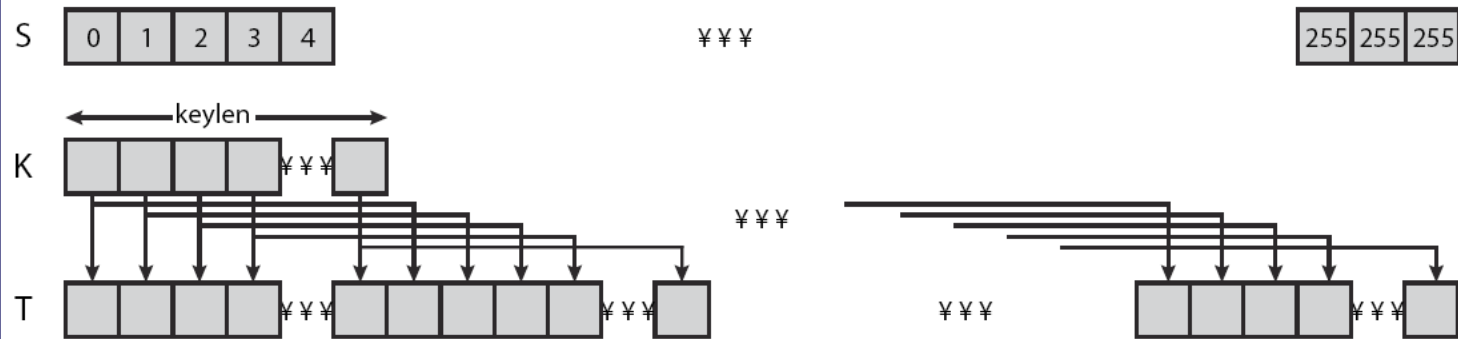
$j = (j + S[i]) \pmod{256}$

swap($S[i]$, $S[j]$)

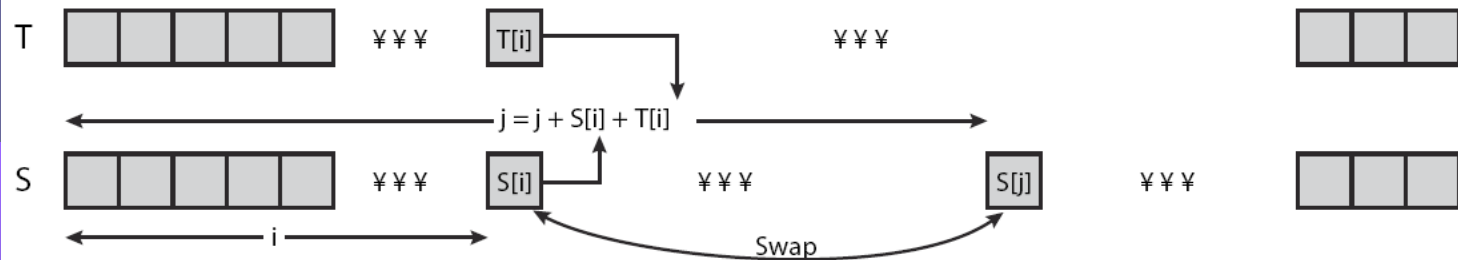
$t = (S[i] + S[j]) \pmod{256}$

$C_i = M_i \text{ XOR } S[t]$

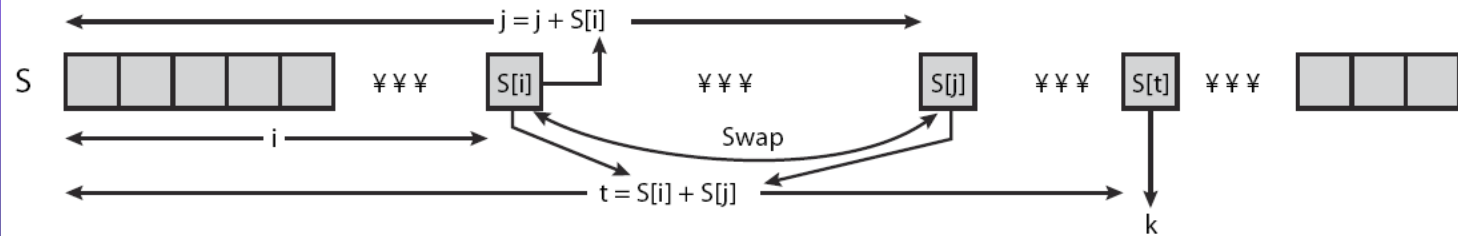
RC4 Overview



(a) Initial state of S and T



(b) Initial permutation of S



(c) Stream Generation

RC4 Security

- claimed secure against known attacks
 - have some analyses, none practical
- result is very non-linear
- since RC4 is a stream cipher, must **never reuse a key**
- have a concern with WEP, but due to key handling rather than RC4 itself

Summary

- Triple-DES
- Modes of Operation
 - ECB, CBC, CFB, OFB, CTR
- stream ciphers
- RC4

