More on Symmetric Ciphers

CSS 322 – Security and Cryptography

Contents

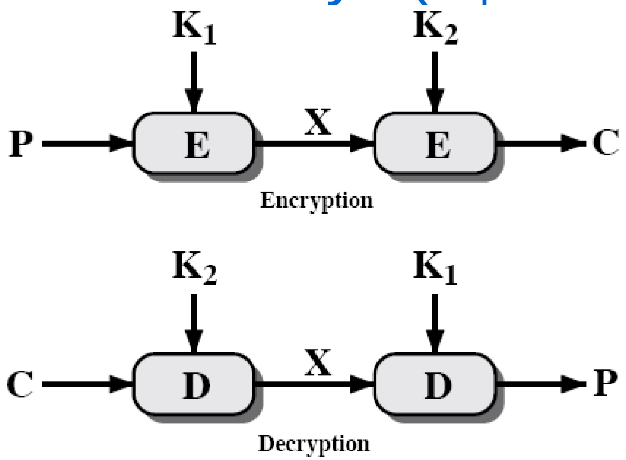
- Multiple Encryption and Triple DES
- Block Cipher Modes of Operation
- Stream Ciphers

Multiple Encryption and DES

- 56-bit key size of DES is a vulnerability to brute force attack
- AES is one alternative to DES
- Another alternative is to use multiple DES encryptions
  - Advantage: make use of existing software/hardware/expertise in DES

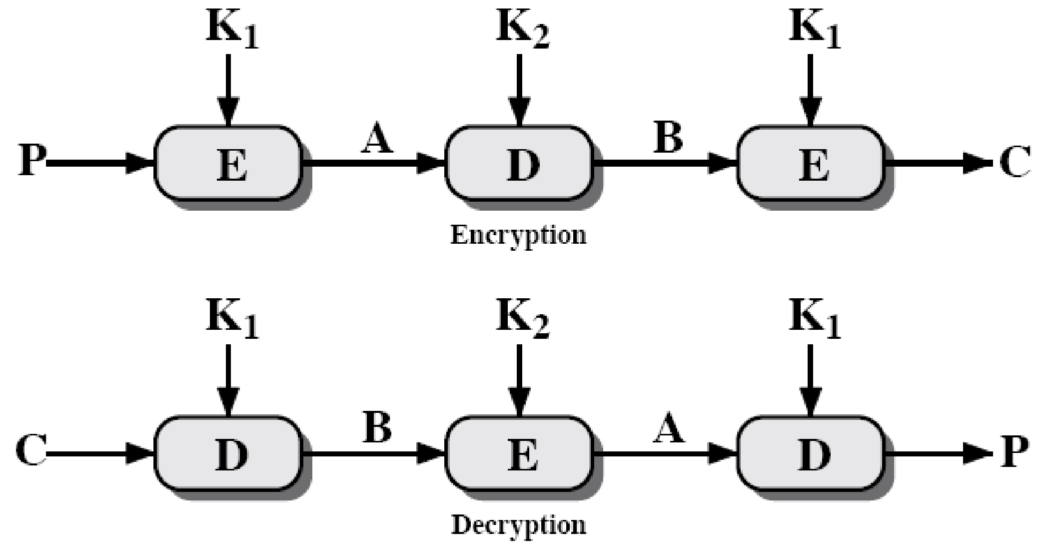
Double DES

- Encrypt with two different keys (K_1 and K_2)



- Effective key length: 2 x 56 bits = 112 bits???
- Meet-in-the-middle attack
  - $X = E(K_1,P) = D(K_2,P)$
  - Given known pair of (P,C), calculate X for all possible K_1
  - Decrypt C using all possible K_2, and if matching X, check keys
  - Result: Double DES can be cracked with almost same effort as single DES

Triple DES



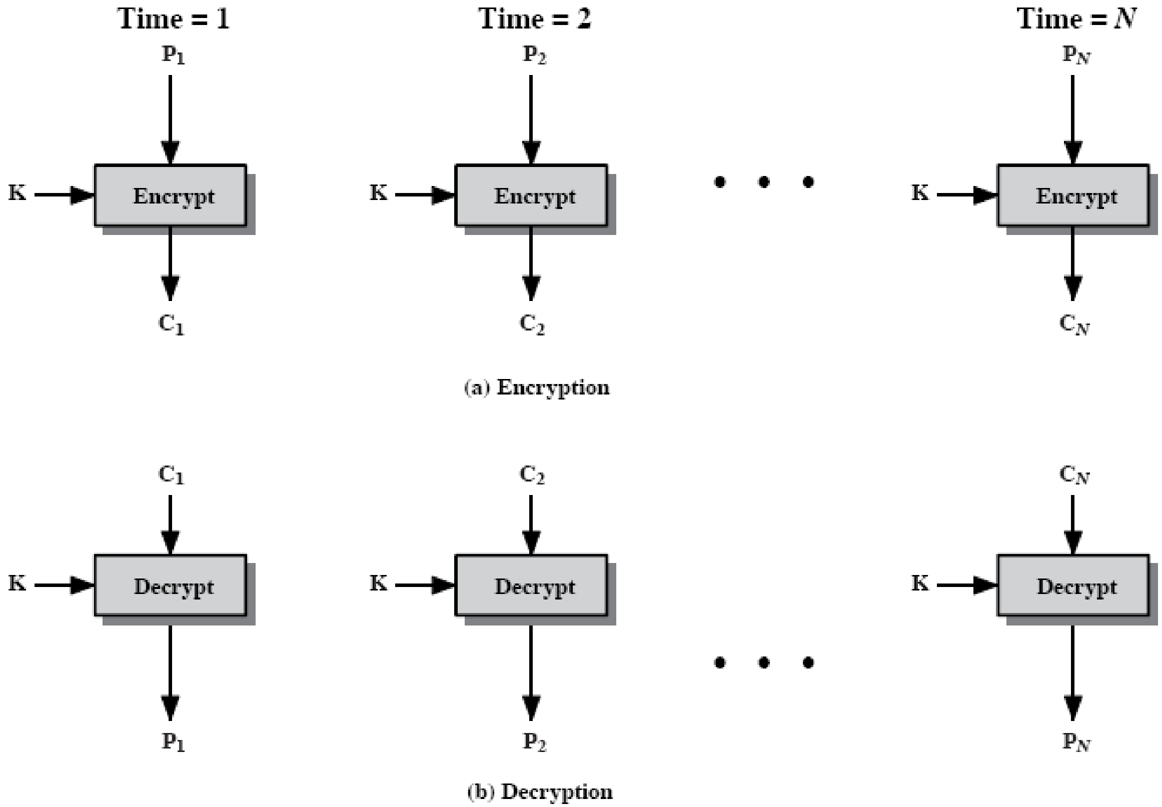
- Counters the meet-in-the-middle attack
- Some standards adopted 2 key triple DES (112 bit)
- Some Internet standards use 3 key triple DES (168 bits)
- Brute force is order of 2^{112}; no known practical attacks
- 3 times slower than single DES

Block Cipher Modes of Operation

- Block cipher operates on fixed sized block (e.g. DES 64 bits)
- Various "modes of operation" defined to apply block cipher algorithms to larger blocks or streams of data
- Can be applied to any block cipher, e.g. DES, AES

Electronic Codebook (ECB)

- Break the message into 64-bit blocks and encrypt each block with same key
  - Padding is used to expand last block to 64-bits
  - Useful for encrypting short messages, such as keys



Problem:

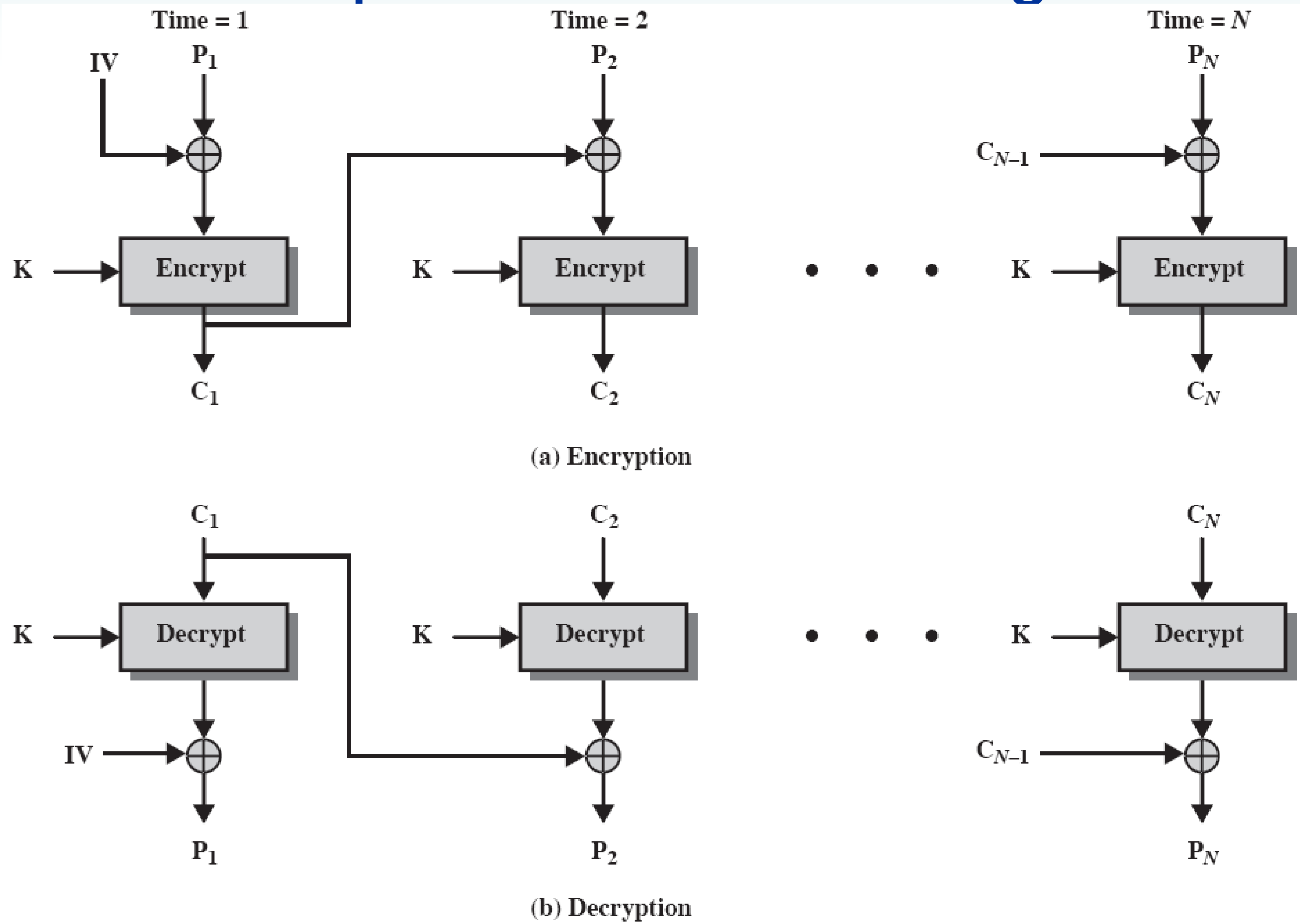
With long message, repetitions in plaintext may produce repetitions in ciphertext.

Attacker may exploit this if they know patterns or part of plaintext

Cipher Block Chaining (CBC)

- Aim: if same plaintext block is repeated, then different ciphertext block is produced
- Approach: XOR ciphertext from previous block with plaintext of current block before encryption
- Requires Initialisation Vector (IV) for first block
  - Should keep IV secret (only known to sender and receiver)
- Good for most block cipher applications with large input blocks
- Can be used for authentication

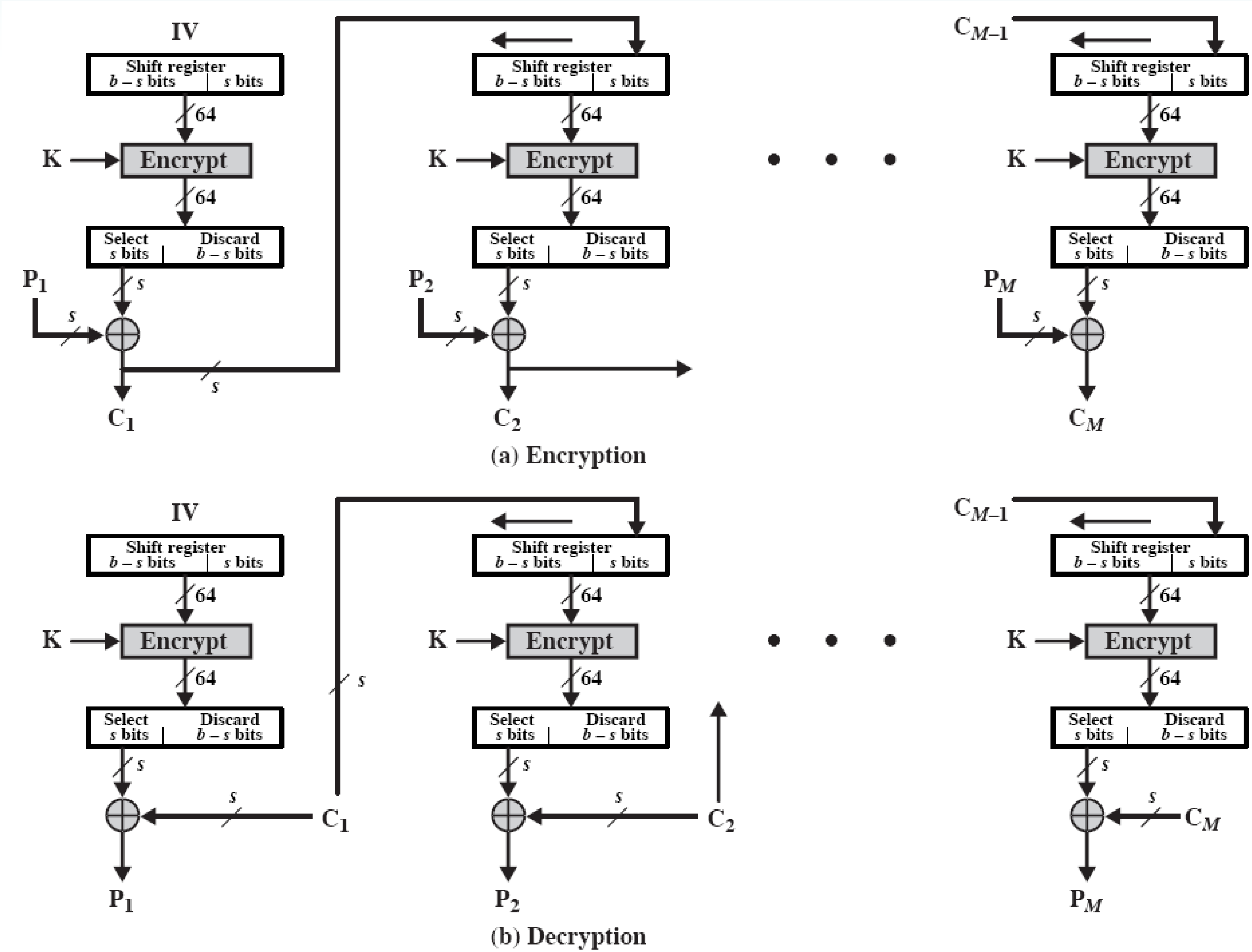
Cipher Block Chaining



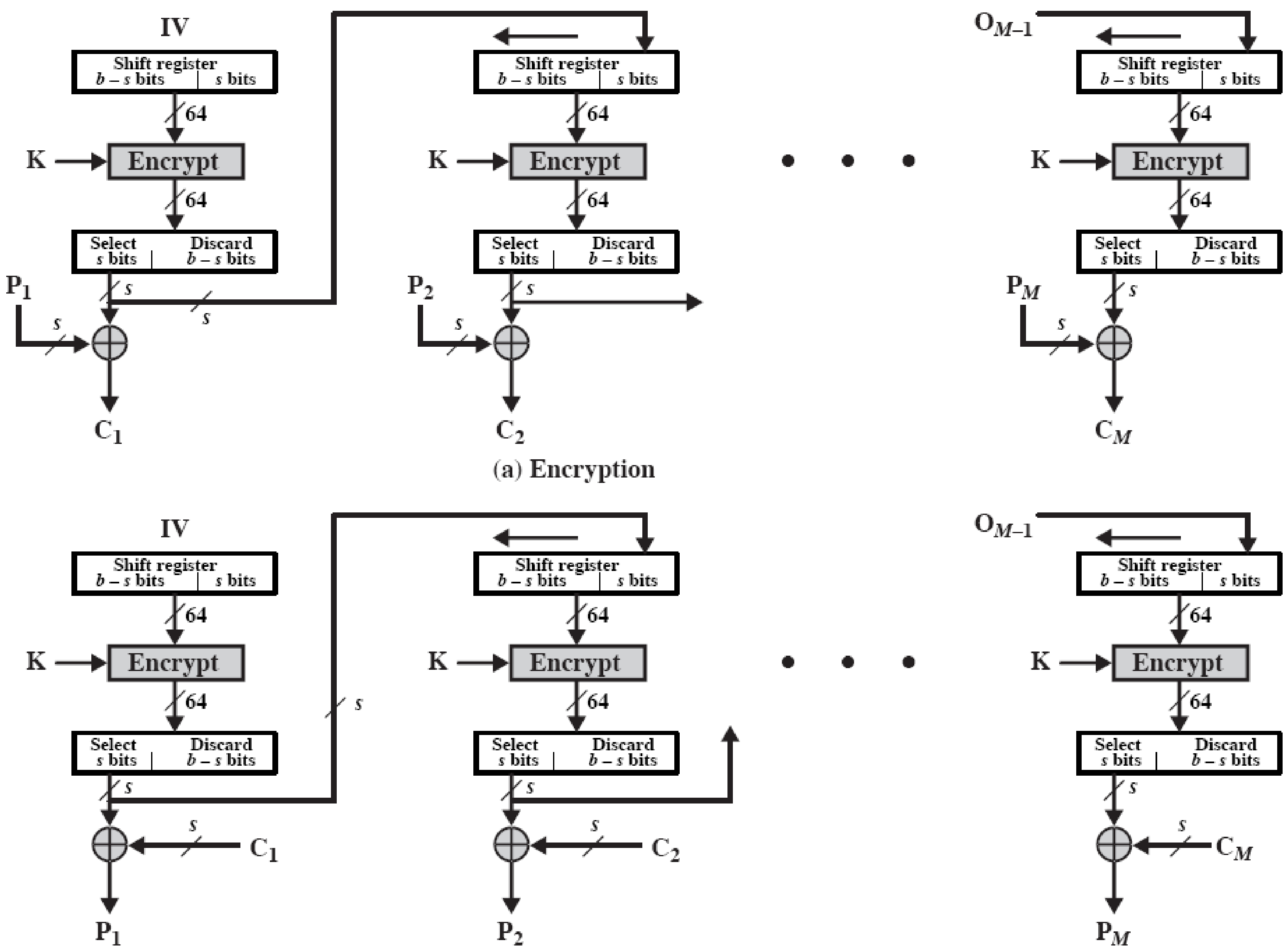
Cipher Feedback Mode (CFB)

- Used to make DES a stream cipher
  - Stream cipher allows message to be transmitted immediately
  - Operates on segments (usually 8 bits) instead of blocks
- Uses chaining (like CBC) so ciphertext is function of all previous plaintext
- Requires Initialisation Vector (IV)
- Variant: Output Feedback Mode (OFB)
  - Output of encryption (instead of ciphertext) is fed into next stage
  - Errors in transmission are not propagated through stages
  - Useful for error-prone channels, such as satellite communications

Cipher Feedback Mode



Output Feedback Mode



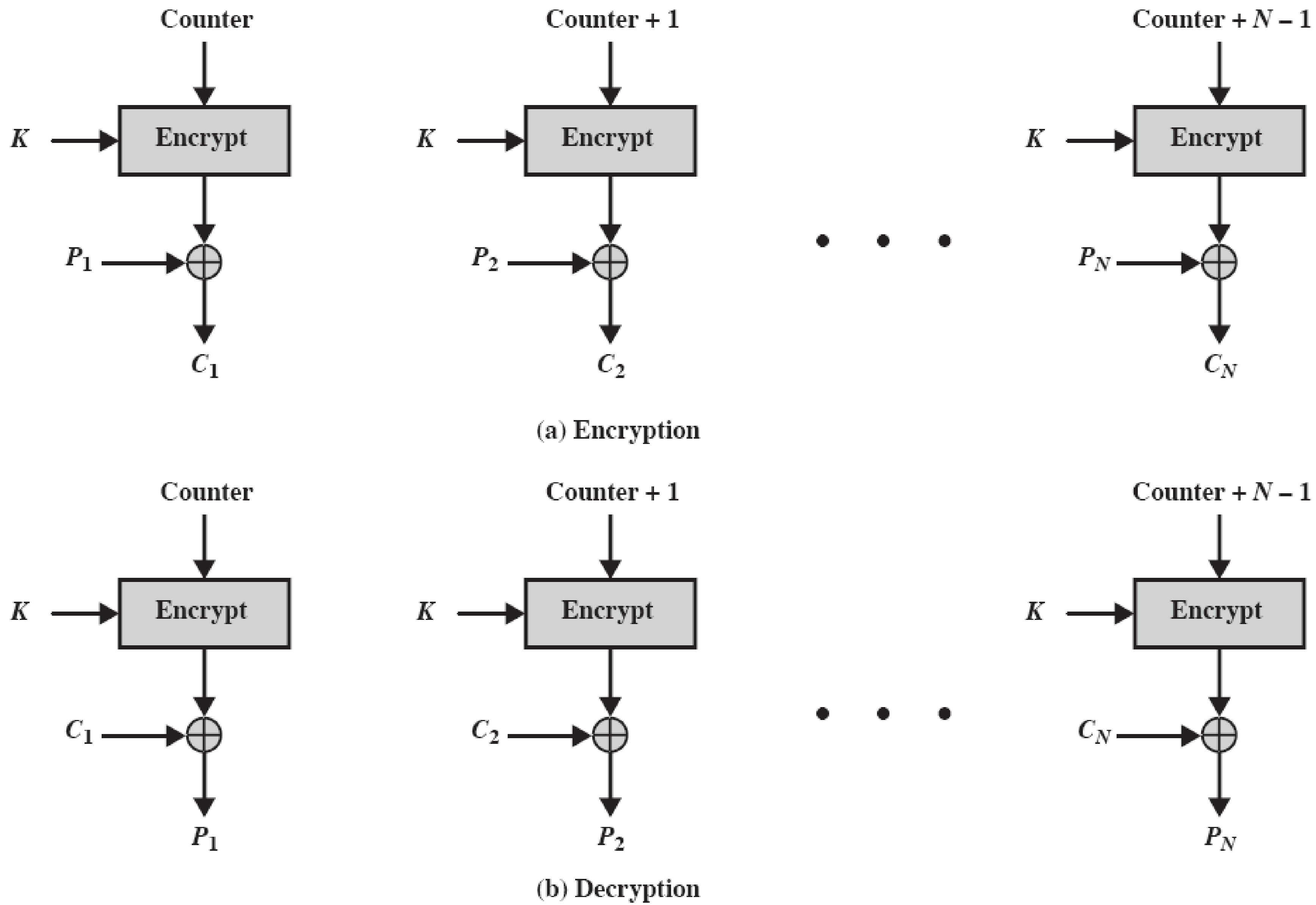
(a) Encryption

(b) Decryption

Counter Mode

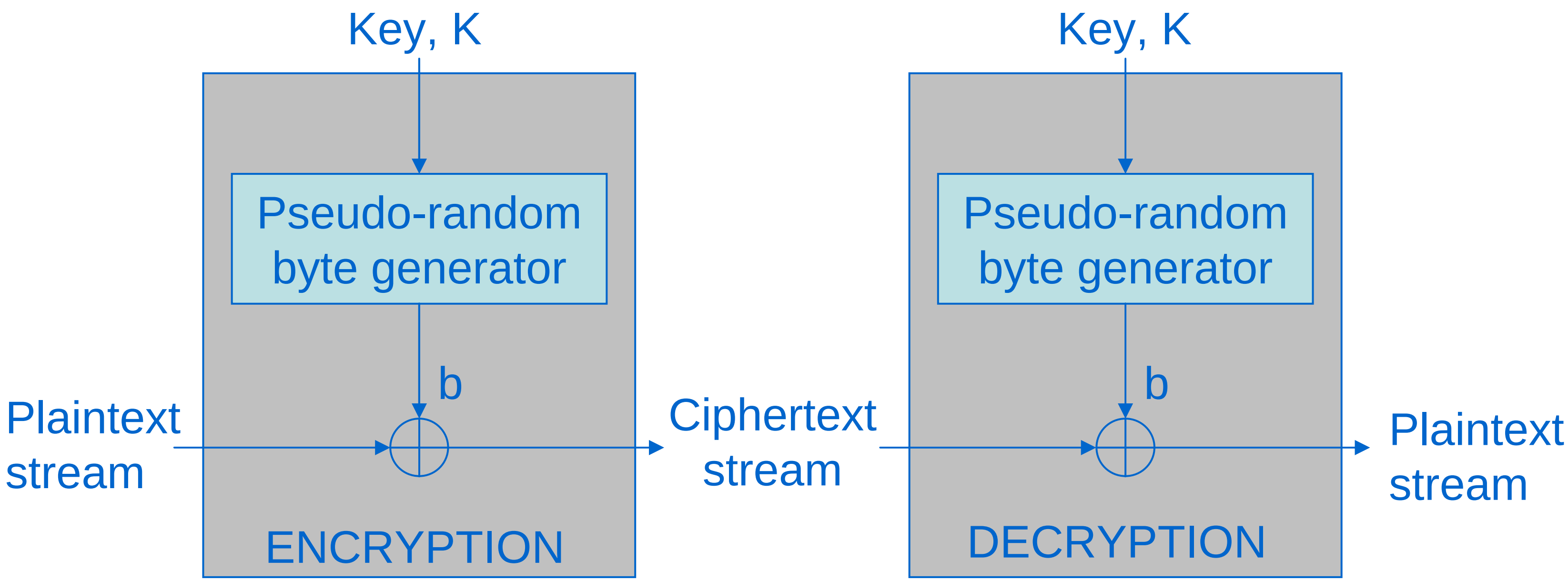
- Input a counter (usually incremented by 1 for each stage) into encrypt; XOR result with plaintext
- No chaining between stages (input of one stage does not need output of previous stage)
- Used in ATM and IPsec
- Advantages
  - Efficient hardware/software implementations – make use of parallel processing abilities of CPU
  - Only requires implementation of encryption algorithm (for example AES encrypt – no need for AES decrypt)
  - At least as secure as other modes

Counter Mode



Stream Ciphers

- Usually encrypt 1 byte at a time
  - Much faster and easier to implement than block ciphers
- Basic encryption method:
  1. Generate pseudo-random number, using key as input
  2. XOR pseudo-random number with byte of plaintext



Stream Cipher Design

- Randomness of input from PRNG hides the structure of plaintext
- Desired properties of stream cipher:
  - Encryption sequence has long period, that is many different random numbers produced before repeating
    - Same as with Vigenère cipher: long keyword makes it stronger
  - The random numbers produced ‘appear’ random (see topic on random numbers)
  - Input keys are long to prevent brute-force attacks
    - 128 bits is sufficient currently
- Do not re-use same key with different plaintext
  - Attacks are relatively easy if have two different ciphertexts produced with same key

RC4 Stream Cipher

- Proprietary design invented by Ron Rivest in 1987
- Algorithm published anonymously on Internet in 1994
- Used in SSL/TLS (secure Internet sockets), WEP/WPA (wireless LAN)
- Very simple and efficient implementation
- Can use variable size key: 8 to 2048 bits
- Several theoretical limitations of RC4
  - No known attacks if use 128-bit key and discard initial values of stream
  - RC4 is used in WEP (shown to be weak security for wireless LANs) – problem with how keys are used, not RC4 algorithm

RC4 Algorithm

- Initialise 256-byte state vector S:
  - S[0]=0, S[1]=1, S[2]=2, …, S[255]=255

```
for i=0 to 255 do {
      S[i] = i;
      T[i]=K[i mod keylen]; }
```

- Initialise 256-byte temporary vector T to the key (or if key is less than 256-bytes, repeat the key)
- Perform initial permutation on S:
  - Go through S, swapping bytes according to T

```
j = 0;
for i=0 to 255 do {
      j = (j + S[i] + T[i]) mod 256;
      Swap (S[i], S[j]); }
```

RC4 Algorithm

- Generate the stream
  - Cycle through S, swapping S[i] with another byte, and repeat once have reached end of S

```
i, j = 0;
while (true) {
      i = (i + 1) mod 256;
      j = (j + S[i]) mod 256;
      Swap (S[i], S[j]);
      t = (S[i] + S[j]) mod 256;
      k = S[t]; }
```

- To encrypt, XOR k with byte of plaintext
- To decrypt, XOR k with byte of ciphertext