COMP3028 Computer Security
Lecture 5 - Cryptography II

Block Ciphers, DES

Block Cipher

- One of the most widely used types of cryptographic algorithms
- Provide secrecy and/or authentication services
- A block of plaintext is treated as a whole and used to produce a ciphertext block of equal length
- Typically a block size of 64 or 128 bits is used
- The majority of network-based symmetric cryptographic applications make use of block ciphers

Block Cipher Principles

- Most symmetric block ciphers are based on a **feistel cipher structure**
- Needed since must be able to **decrypt** ciphertext to recover messages efficiently
- Block ciphers look like an extremely large substitution
- Would need table of 2^{64} entries for a 64-bit block
- Instead create from smaller building blocks

Substitution-Permutation Ciphers

- In 1949 shannon introduced idea of substitution-permutation (S-P) networks
- These form the basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations we have seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)

Confusion and Diffusion

- Cipher needs to completely obscure statistical properties of original message
- A one-time pad does this
- More practically shannon suggested combining elements to obtain:
- **Confusion** – makes relationship between ciphertext and key as complex as possible
- **Diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext

Substitution Operation

- A binary word is replaced by some other binary word
- the whole substitution function forms the key
- if use n bit words, the key is $2^n!$ bits, which grows rapidly as n increases
- can also think of this as a large lookup table
- with n address lines (hence 2^n addresses)
- each n bits wide, being the output value
- will call them **S-boxes**

Permutation Operation

- a binary word has its bits reordered (permuted)
- the re-ordering forms the key
- if use n bit words, the key is n! bits
- which grows more slowly, and hence is less secure than substitution
- this is equivalent to a wire-crossing in practice (though is much harder to do in software)
- will call these **P-boxes**

Substitution-Permutation Network

- Shannon combined these two primitives
- he called these **mixing transformations**
- special form of product ciphers where

  S-Boxes

  – provide **confusion** of input bits

  P-Boxes

  – provide **diffusion** across S-box inputs

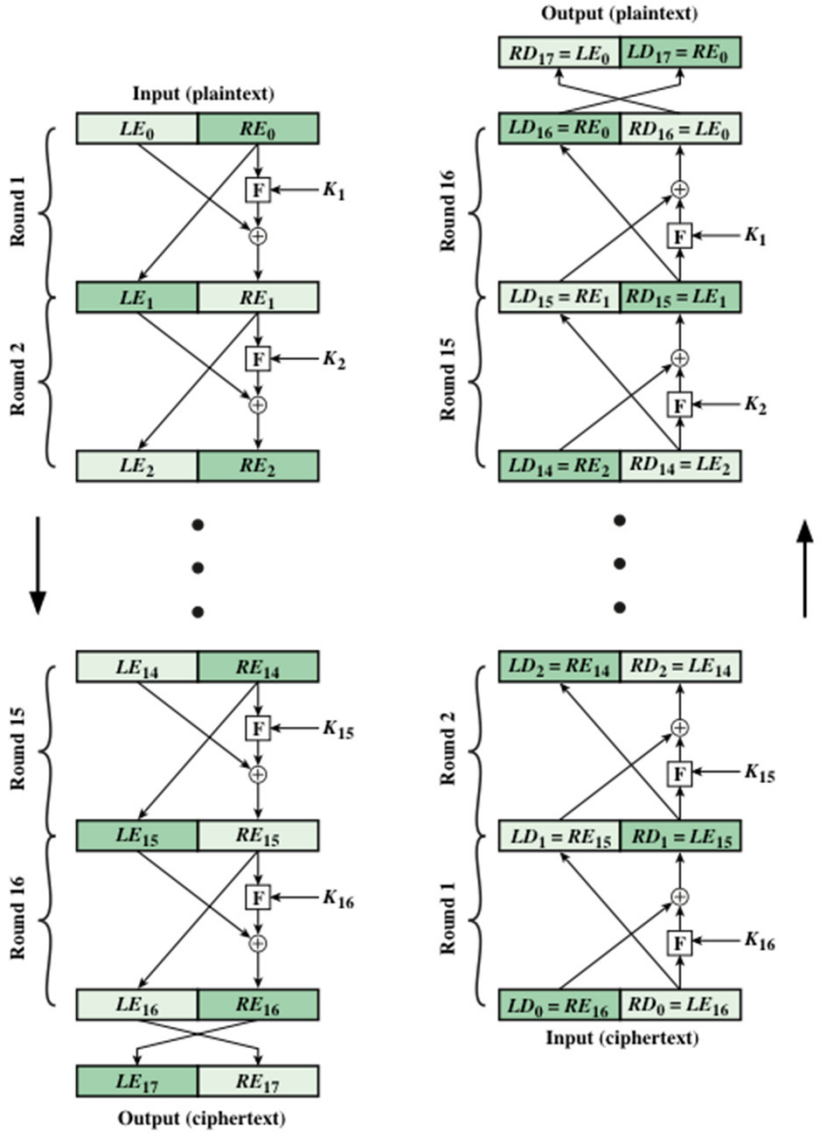
Practical Substitution-Permutation Networks

- in practice we need to be able to decrypt messages, as well as to encrypt them, hence either:
  - have to define inverses for each of our S & P-boxes, but this doubles the code/hardware needed, or
  - define a structure that is easy to reverse, so can use basically the same code or hardware for both encryption and decryption

Feistel Cipher Structure

- Horst Feistel devised such a structure called the **Feistel Cipher**
- partitions input block into two halves
  - process through multiple rounds:
    - perform a substitution on left data half by applying a round function of right half & a subkey
    - then have permutation swapping the two halves

Feistel Encryption & Decryption



Feistel Cipher Design Principles

- **block size**
  - increasing size improves security, but slows cipher
- **key size**
  - increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- **number of rounds**
  - increasing number improves security, but slows cipher
- **subkey generation**
  - greater complexity can make analysis harder, but slows cipher
- **round function**
  - mixing substitution and permutation increases complexity which can make analysis harder, but slows cipher

Feistel Cipher Design Principles

- **fast software en/decryption**
  - In many cases, encrypting is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the speed of execution of the algorithm becomes a concern
- **ease of analysis**
  - If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

Data Encryption Standard (DES)

- most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security

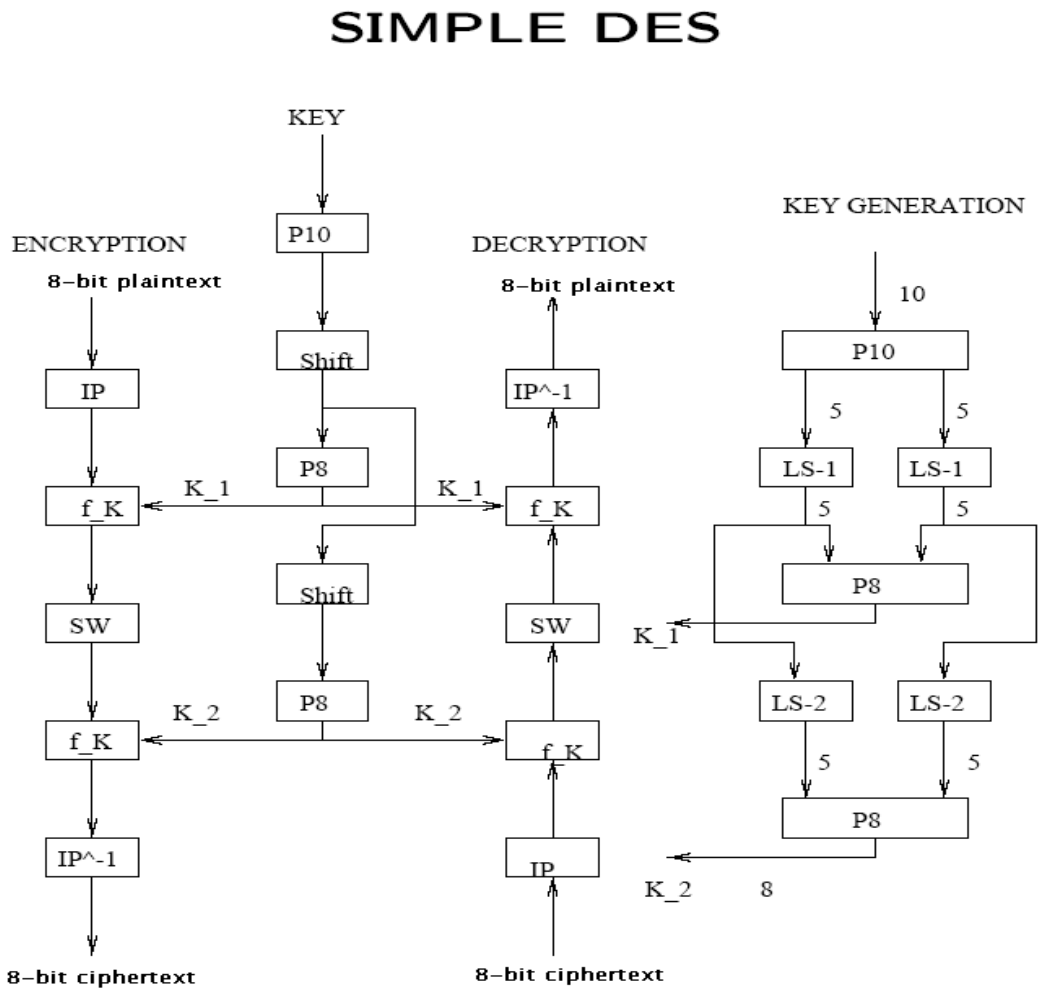
DES History

- IBM developed Lucifer cipher
  - by team led by Feistel
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES

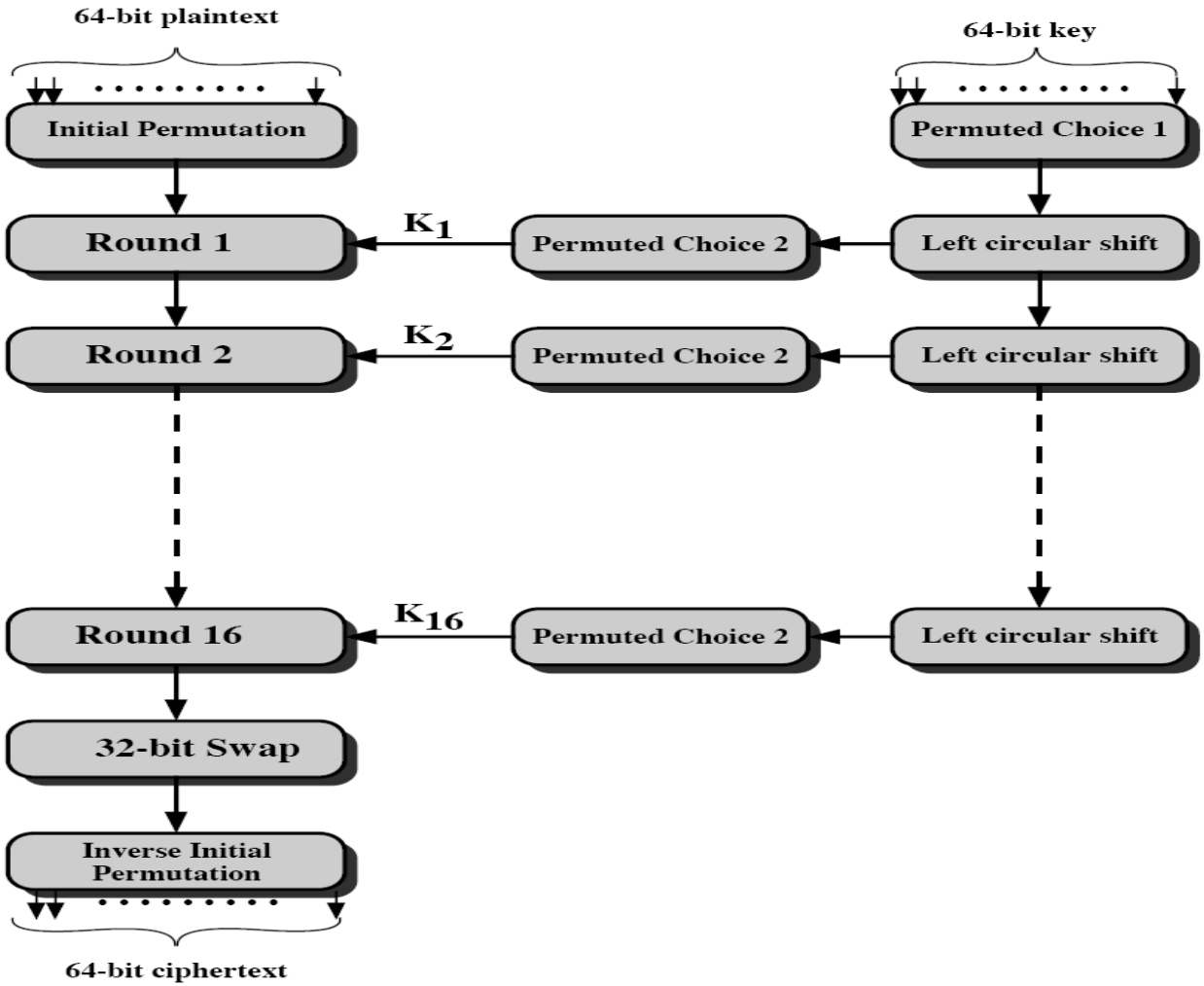
DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (vs. Lucifer 128-bit)
  - and because design criteria were classified
- subsequent events and public analysis show in fact design was appropriate
- DES has become widely used, esp. in financial applications
- still standardised for legacy application use

Simplified DES (SDES)



DES Encryption



Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Inverse Initial Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Expansion Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- even bits to LH half, odd bits to RH half
- quite regular in structure (easy in h/w)
- example:

```
IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)
```

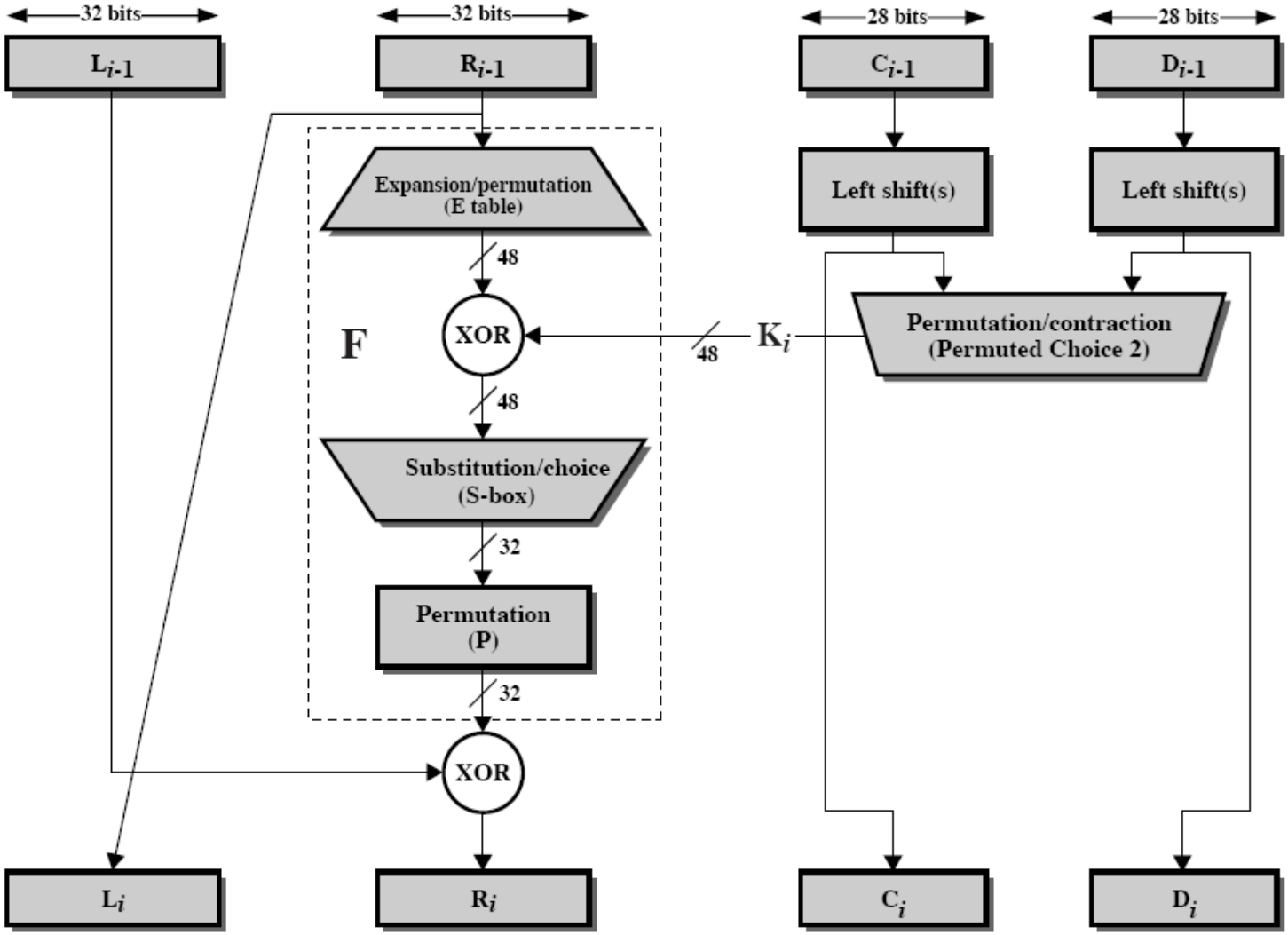
DES Round Structure

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i)$
- F takes 32-bit R half and 48-bit subkey and:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes this using 32-bit perm P

Single Round of DES Algorithm



Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) select one rows
  - inner bits 2-5 (**col** bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:

```
S(18 09 12 3d 11 17 38 39) = 5fd25e03
```

Definition of DES S-Boxes

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

DES Key Schedule

- forms subkeys used in each round
- consists of:
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K
    - selecting 24-bits from each half and permuting them by PC2 for use in function F
- can be described functionality by

  K_i = PC2 (K (PC1 (key)), i)

DES Key Schedule (cont.)

Permuted Choice One (PC-1)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Schedule of Left Shifts

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again using subkeys in reverse order (SK_{15} … SK_0)
- IP undoes final FP step of encryption
- 1st round with SK_{15} undoes 16th encrypt round
- etc. until 16th round with SK_0 undoes 1st encrypt round
- then final FP undoes initial encryption IP
- thus recovering original data value

Avalanche Effect

- key desirable property of encryption algorithm
- where a change of one input or key bit results in changing approx half output bits
- making attempts to "home-in" by guessing keys impossible
- DES exhibits strong avalanche

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - in 1999 above combined in 22hrs!
- still must be able to recognize plaintext
- now considering alternatives to DES

Strength of DES – Analytic Attacks

- now have several analytic attacks on DES
- these utilise some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks include
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Strength of DES – Timing Attacks

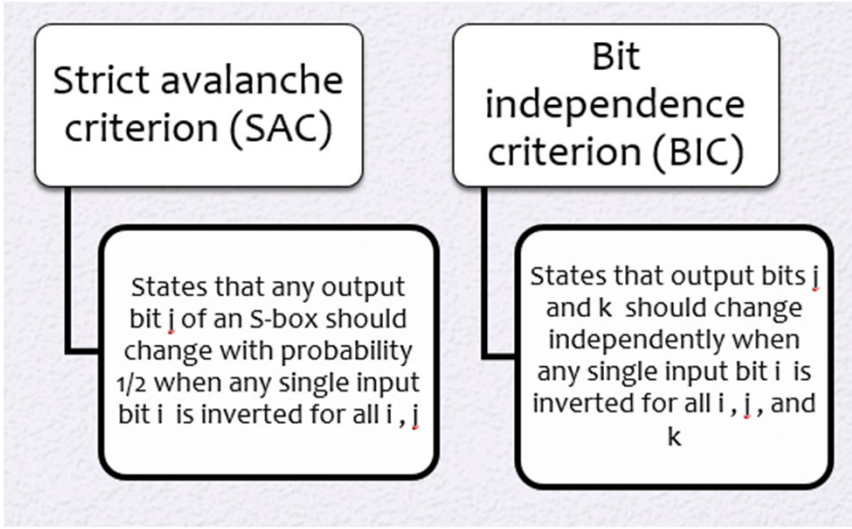
- attacks actual implementation of cipher
- use knowledge of consequences of implementation to derive knowledge of some/all subkey bits
- specifically use fact that calculations can take varying times depending on the value of the inputs to it
- particularly problematic on smartcards

Block Cipher Design Principles

- Number of rounds
  - The greater the number of rounds, the more difficult it is to perform cryptanalysis
  - In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack
  - If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search

Block Cipher Design Principles

- Design of function F
  - The heart of a Feistel block cipher is the function F
  - The more nonlinear F, the more difficult any type of cryptanalysis will be
  - The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function
  - The algorithm should have good avalanche properties



Block Cipher Design Principles

- Key schedule algorithm
  - With any Feistel block cipher, the key is used to generate one subkey for each round
  - In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key
  - It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext SAC and BIC

Multiple Encryption & DES

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

Modes of Operation

- block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks, with 56-bit key
- need ways to use in practise, given usually have arbitrary amount of data to encrypt/decrypt
- four modes of operations were defined in **ANSI X3.106-1983 Modes of Use** (now FIPS 81)
- subsequently 5 defined for symmetric block ciphers

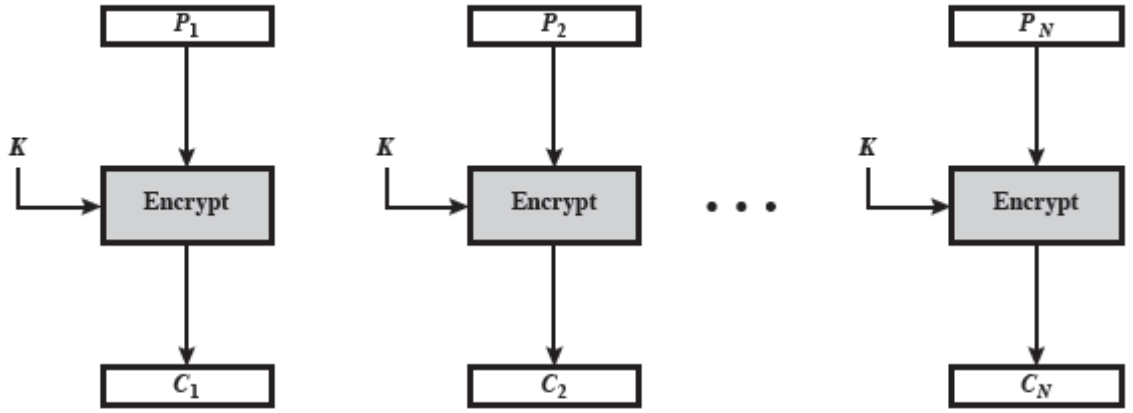
Electronic Code Book (ECB)

- 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

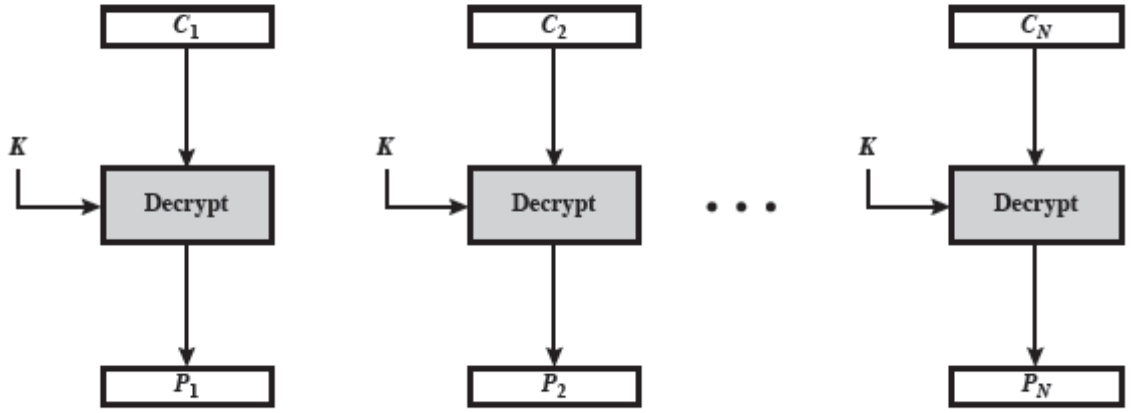
$$C_i = E_K(P_i)$$

- uses: secure transmission of single values

Electronic Codebook Book (ECB)



(a) Encryption



(b) Decryption

Advantages and Limitations of ECB

- repetitions in message 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 due to encrypted message blocks being independent
- main use is sending a few blocks of data

Cipher Block Chaining (CBC)

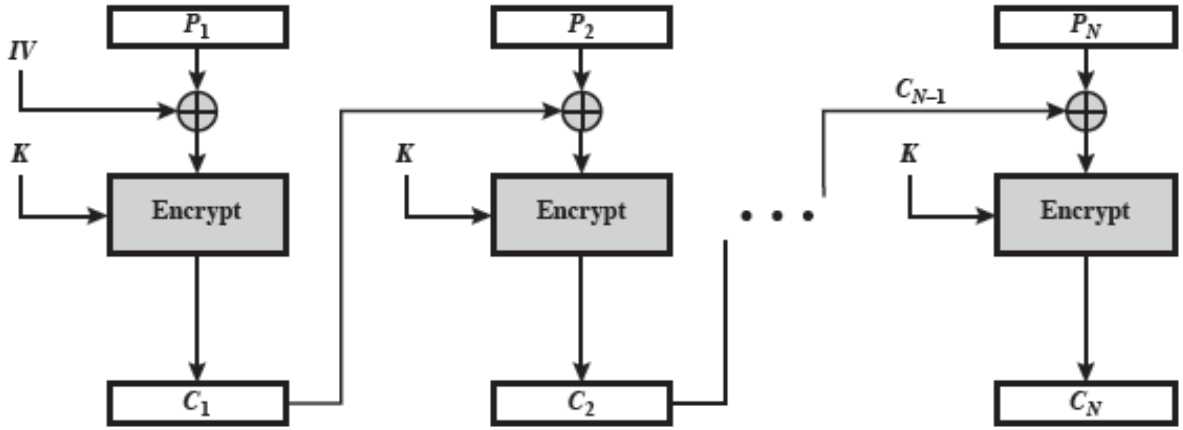
- message is broken into blocks
- but these are linked together in the encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initialisation Vector (IV) to start process

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

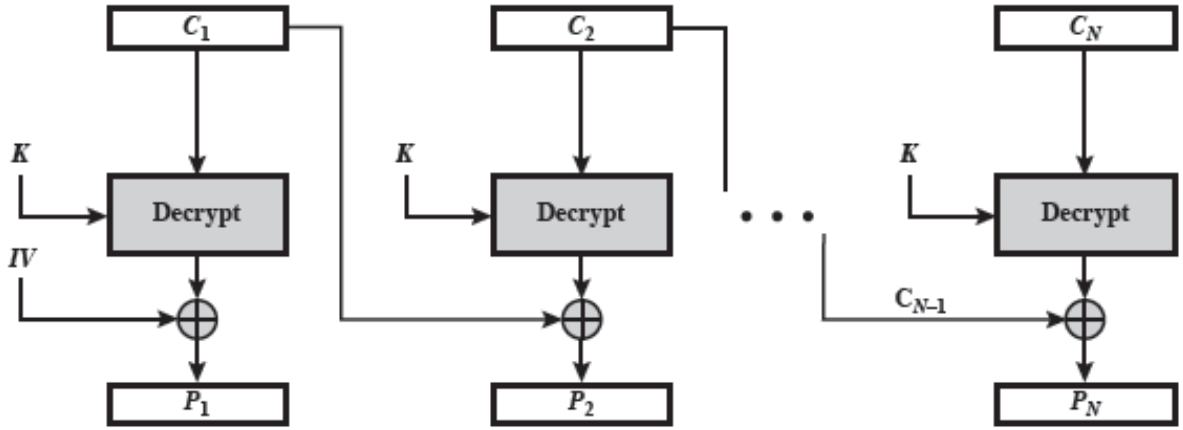
$$C_0 = IV$$

- 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 **Initialisation Vector** (IV)
  - known to sender & receiver
  - if IV is sent in the clear, an attacker can change bits of the first block, and change IV to compensate
  - hence either IV must be a fixed value (as in EFTPOS) or must be sent encrypted in ECB mode before rest of message

Cipher FeedBack (CFB)

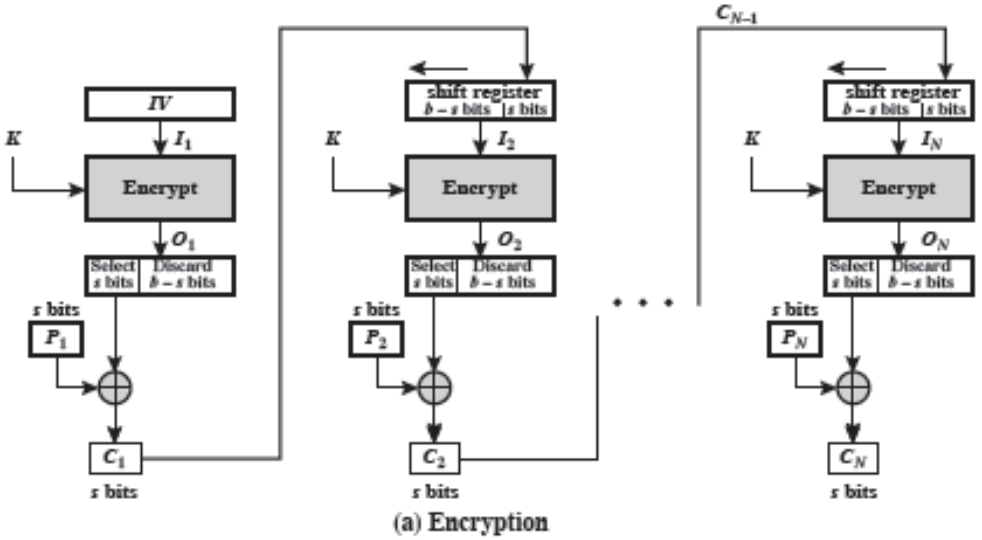
- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8 or 64 or 128 etc.) to be feed back
  - denoted CFB-1, CFB-8, CFB-64, CFB-128 etc.
- is most efficient to use all bits in block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

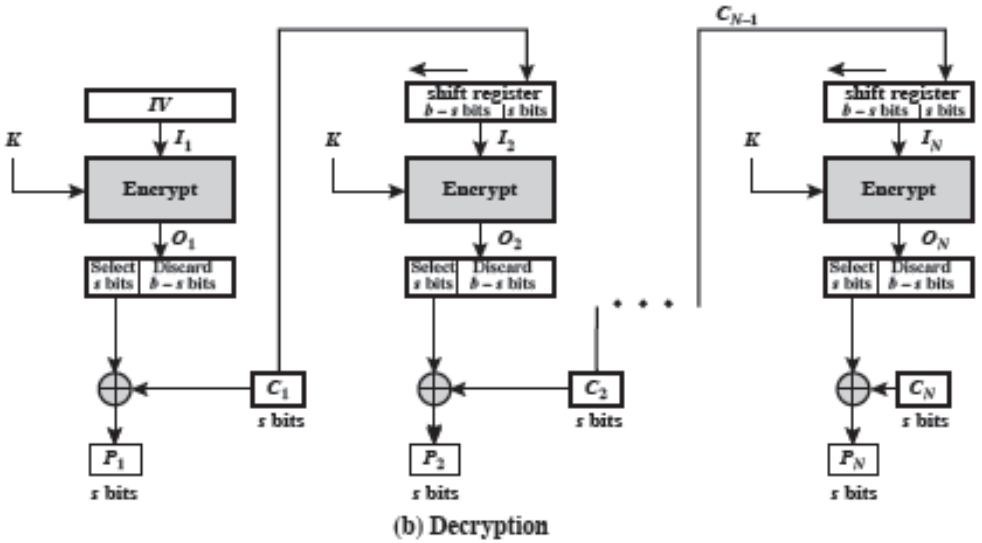
$$C_i = P_i \text{ XOR } E_K(C_{i-1})$$

$$C_0 = IV$$

- 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

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

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

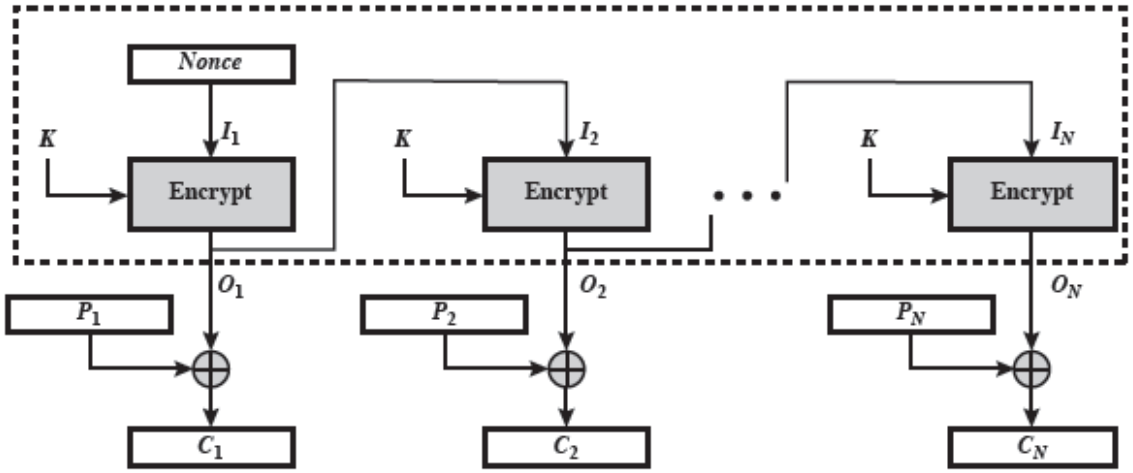
$$O_i = E_K(O_{i-1})$$

$$O_0 = IV$$

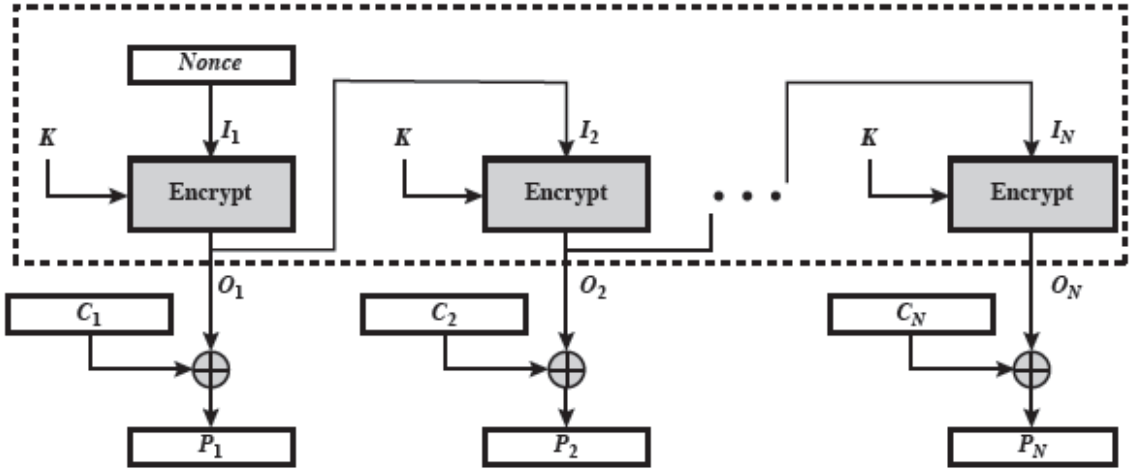
- uses: stream encryption over noisy channels

Output FeedBack (OFB)

Nonce is an arbitrary number used ONLY ONCE. It may be an IV.



(a) Encryption



(b) Decryption

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)

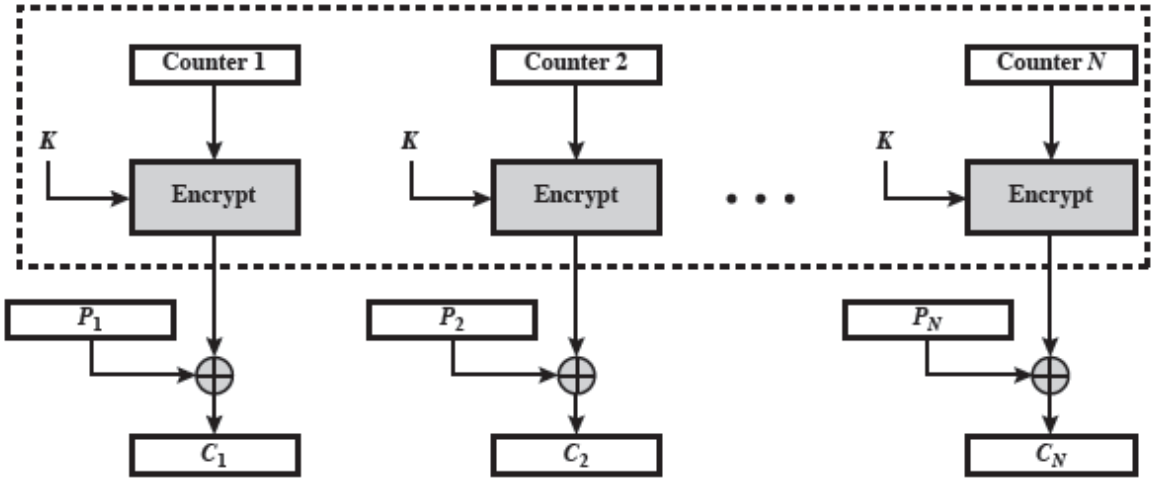
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

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

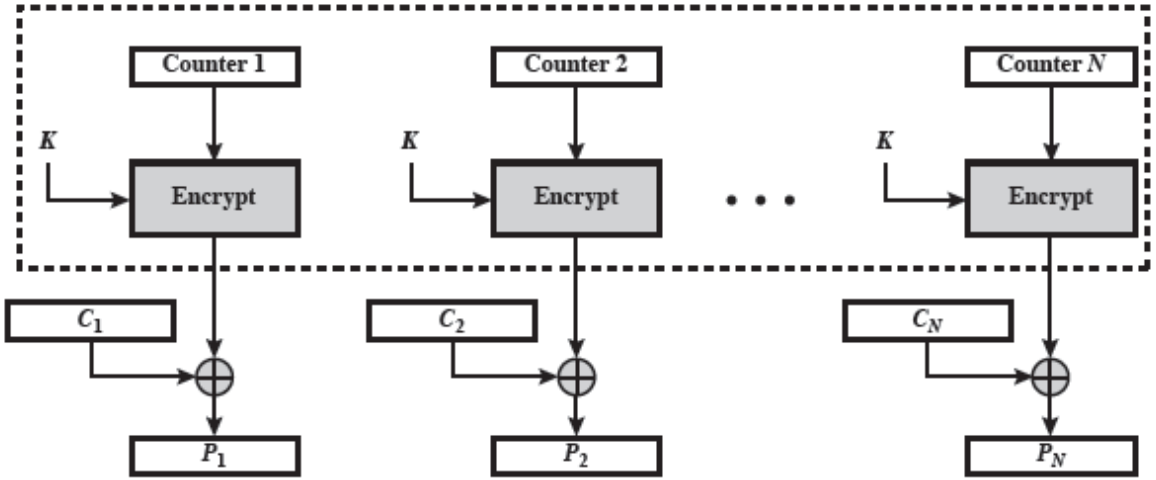
$$O_i = E_K(\text{Counter}_i)$$

- uses: high-speed network encryptions

Counter (CTR)



(a) Encryption



(b) Decryption

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - 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)

Block Cipher Modes of Operation

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of plaintext bits is encoded independently using the same key.	•Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next block of plaintext and the preceding block of ciphertext.	•General-purpose block-oriented transmission •Authentication
Cipher Feedback (CFB)	Input is processed *s* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	•General-purpose stream-oriented transmission •Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	•Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	•General-purpose block-oriented transmission •Useful for high-speed requirements

Summary

- Block Cipher
- SP-Network
- Feistel Structure
- DES
- Modes of Operation
  - ECB, CBC, CFB, OFB, CTR
- Further reading: Anderson Chapter 5, Gollmann Chapter 14