3 Block Ciphers and the Data Encryption Standard

ch4 in textbook

Yanwei Yu

E-mail: ywyu@ustc.edu.cn

Review of Classical Encryption

- transposition：
  - No key -> key
- substitution:
  - Monalphabetic(单表替换) -> Polyalphabetic Substitution (多表替换)
  - No key -> key
- product：rotor

Learning Objectives

- Understand the distinction between stream ciphers and block ciphers.
- Present an overview of the Feistel cipher and explain how decryption is the inverse of encryption.
- Present an overview of Data Encryption Standard (DES).
- Explain the concept of the avalanche effect(雪崩效应).
- Discuss the cryptographic strength of DES.
- Summarize the principal block cipher design principles.

Outline

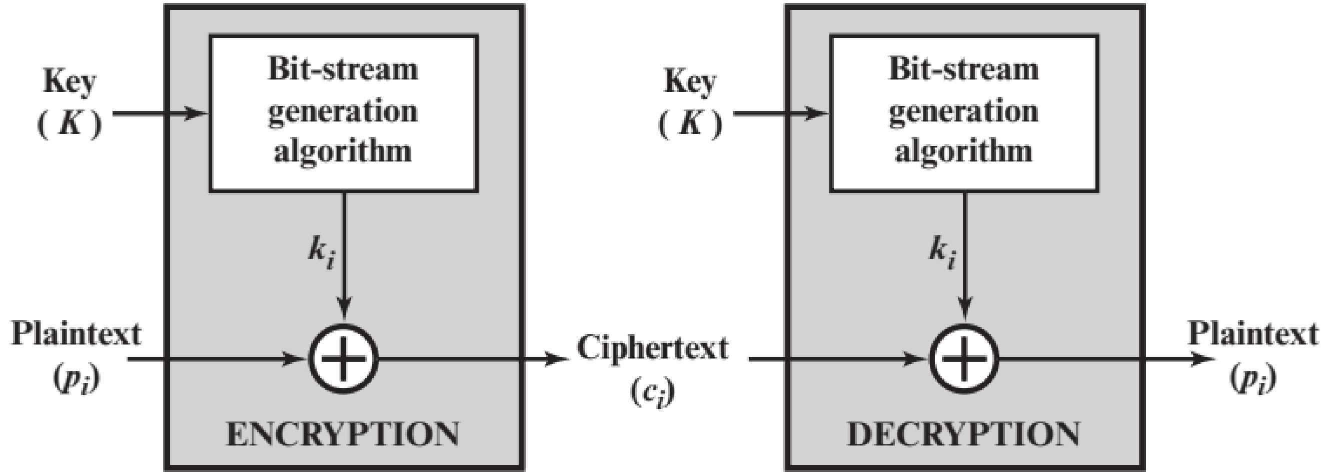
- Block vs. Stream Ciphers
- Ideal Block Cipher
- Feistel Cipher Structure
- DES

Modern Block Ciphers

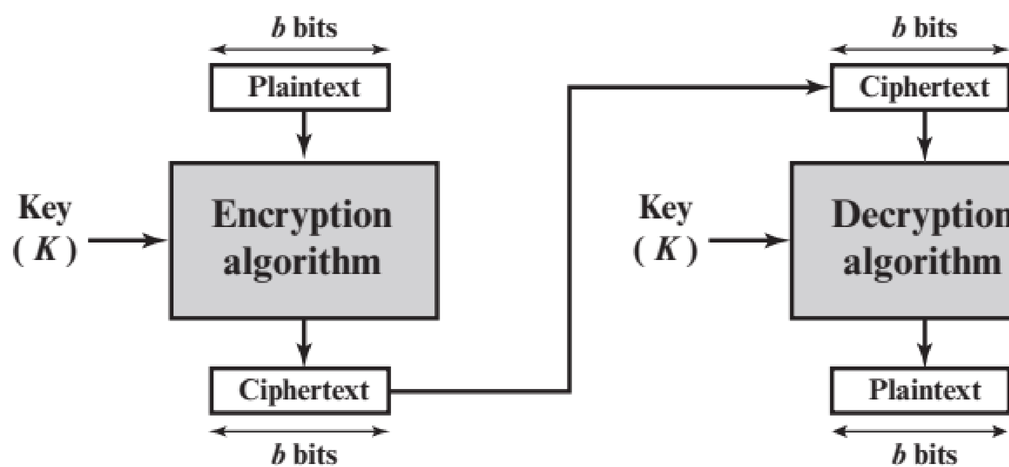
- one of the most widely used types of cryptographic algorithms
- provide secrecy / authentication services
- focus on DES (Data Encryption Standard)
  - Based on feistel structure
- to illustrate block cipher design principles

Block vs. Stream Ciphers

- block ciphers process messages in blocks, each of which is then en/decrypted
- like a substitution on very big characters
  - 64-bits or more
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers
- broader range of applications



(a) Stream cipher using algorithmic bit-stream generator

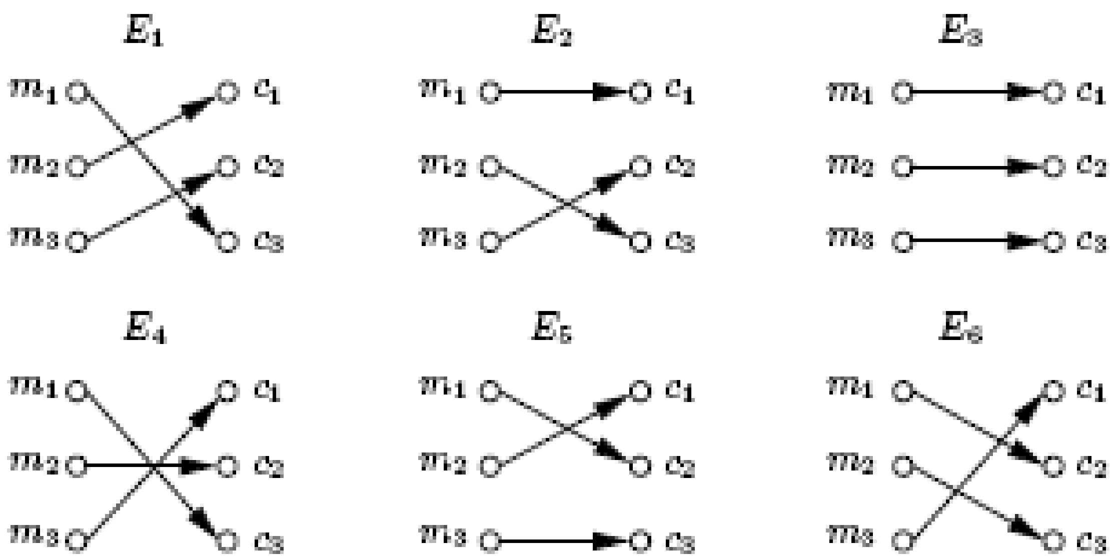


(b) Block cipher

Figure 4.1 Stream Cipher and Block Cipher

Example of Symmetric Block Cipher

- **Assume M={m1,m2,m3}, C={c1,c2,c3}, K={1,2,3,4,5,6} and C=E_k(M), M=D_k(C)**



Number of substitution tables: 3!= 6

number of keys: 6

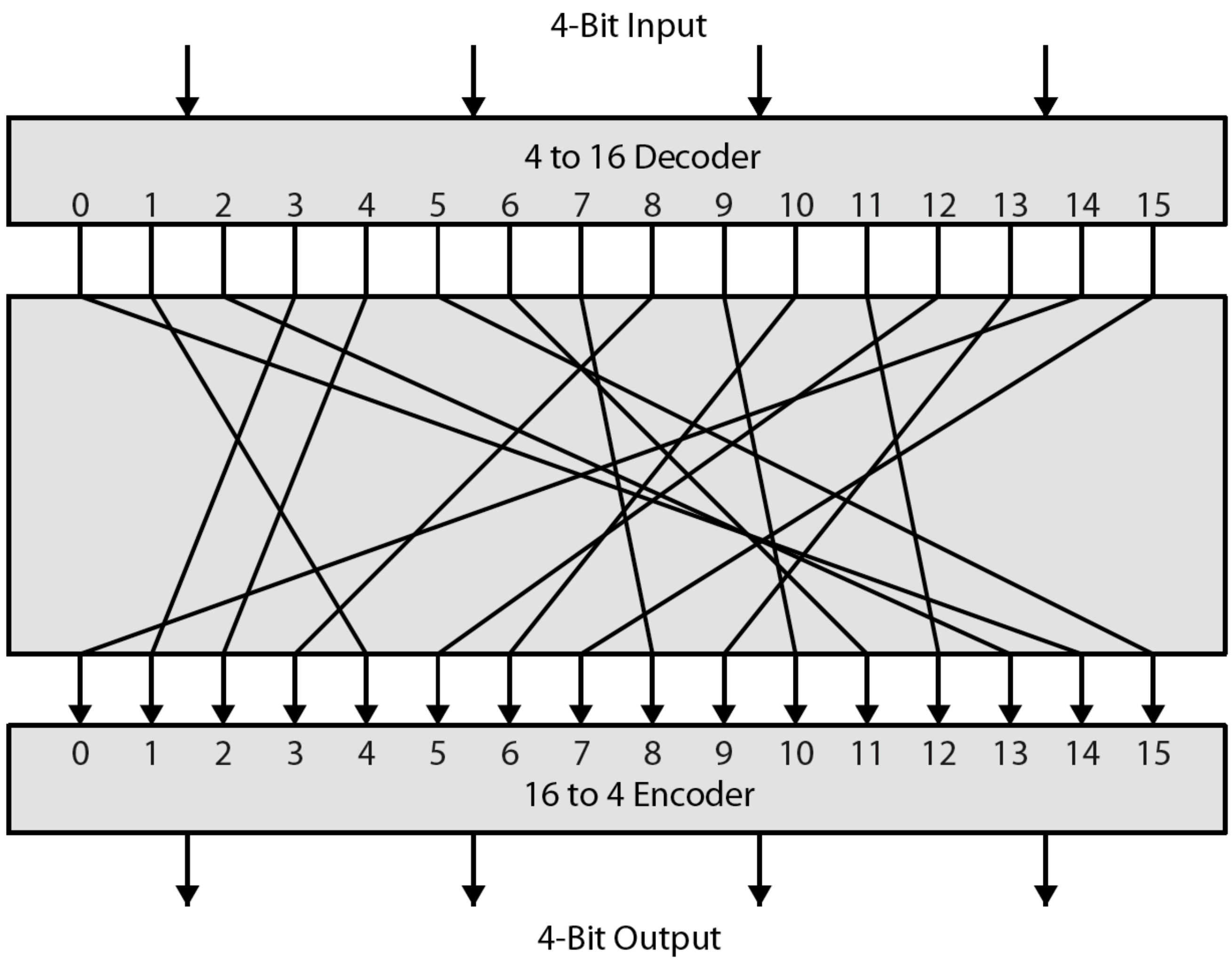
Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

Ideal Block Cipher



Block Cipher Principles

- **most symmetric block ciphers are based on a** Feistel Cipher Structure
  - using **substitution, transposition, product**
- Feistel Cipher Structure: based on **Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper**

Claude Shannon and Substitution-Permutation Ciphers

- **Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper**
- **form basis of modern block ciphers**
- **S-P nets are based on the two primitive cryptographic operations seen before:**
  - ***substitution* (S-box)**
  - ***permutation* (P-box)**
- **provide *confusion* & *diffusion* of message & key**

Confusion and Diffusion

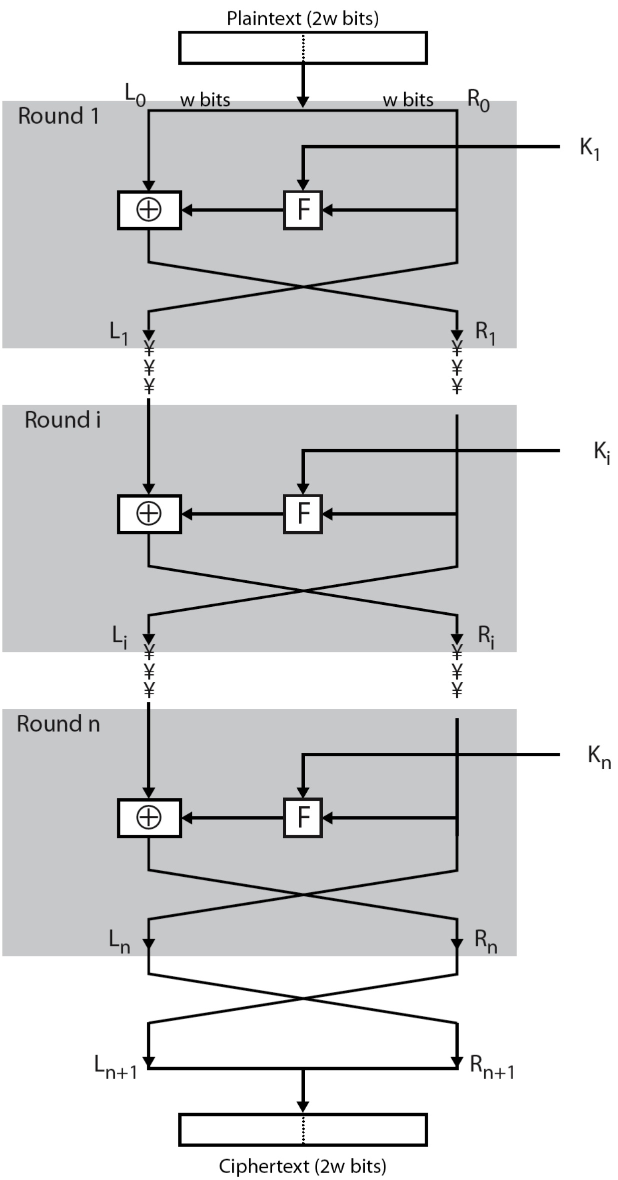
- Computationally secure ciphers based on the idea of confusion and diffusion
- Diffusion(扩散) – spreading influence of one plaintext letter to many ciphertext letters
  - E.g. through the use of permutations and linear substitutions
- Confusion(混淆) – makes relationship between ciphertext and key as complex as possible
  - E.g. through the use of non-linear substitutions

Feistel Cipher Structure

- **Horst Feistel devised the** feistel cipher
  - **based on concept of invertible product cipher**
  - **implements Shannon's S-P net concept**

- **process through multiple rounds**
- **For each round,**
  - **partitions input block into two halves**
  - **perform a substitution on left data half based on the round function of last right half & subkey**
  - **then have permutation swapping halves**

Feistel Cipher Structure

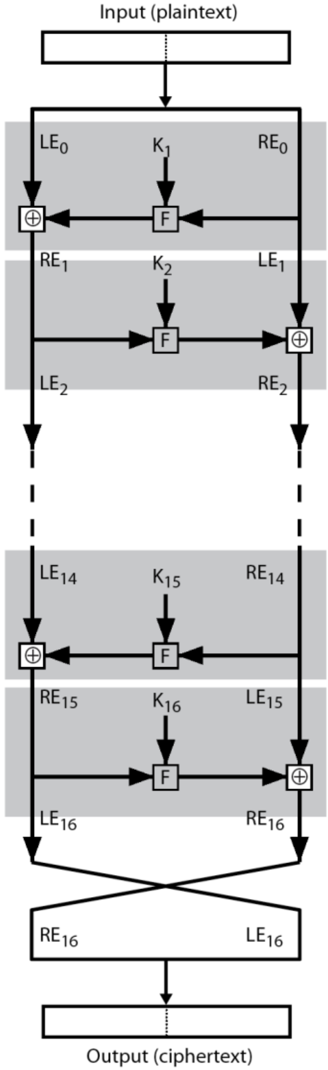


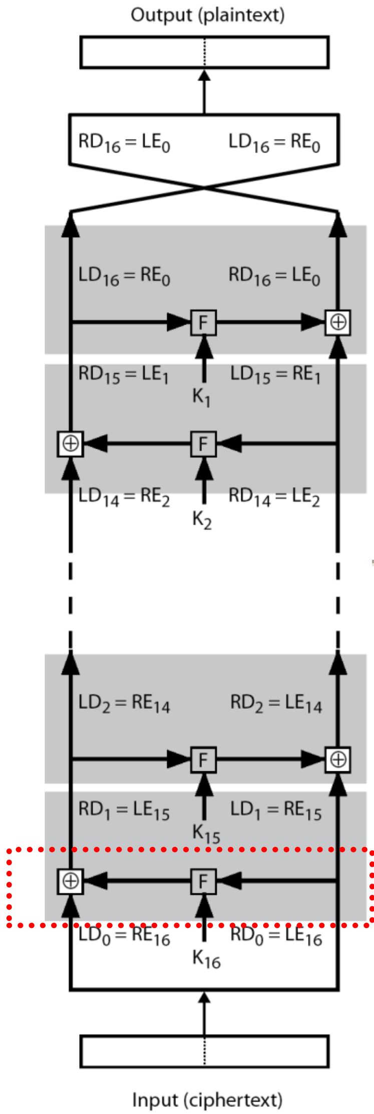
Feistel Cipher Design Elements

- block size
- key size
- number of rounds
- subkey generation algorithm
- round function (no invertible requirements)

- fast software en/decryption
- ease of analysis

Feistel Cipher Decryption





Q:If $LD_0 = RE_{16}$, $RD_0 = LE_{16}$
Then $LD_1 = RE_{15}$, $RD_1 = LE_{15}$?

First, consider the encryption process. We see that

$LE_{16} = RE_{15}$

$RE_{16} = LE_{15} \times F(RE_{15}, K_{16})$

On the decryption side,

$LD_1 = RD_0 = LE_{16} = RE_{15}$

$RD_1 = LD_0 \times F(RD_0, K_{16})$

$= RE_{16} \times F(RE_{15}, K_{16})$

$= [LE_{15} \times F(RE_{15}, K_{16})] \times F(RE_{15}, K_{16})$

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$

Feistel Example

Figure 4.4 Feistel Example

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

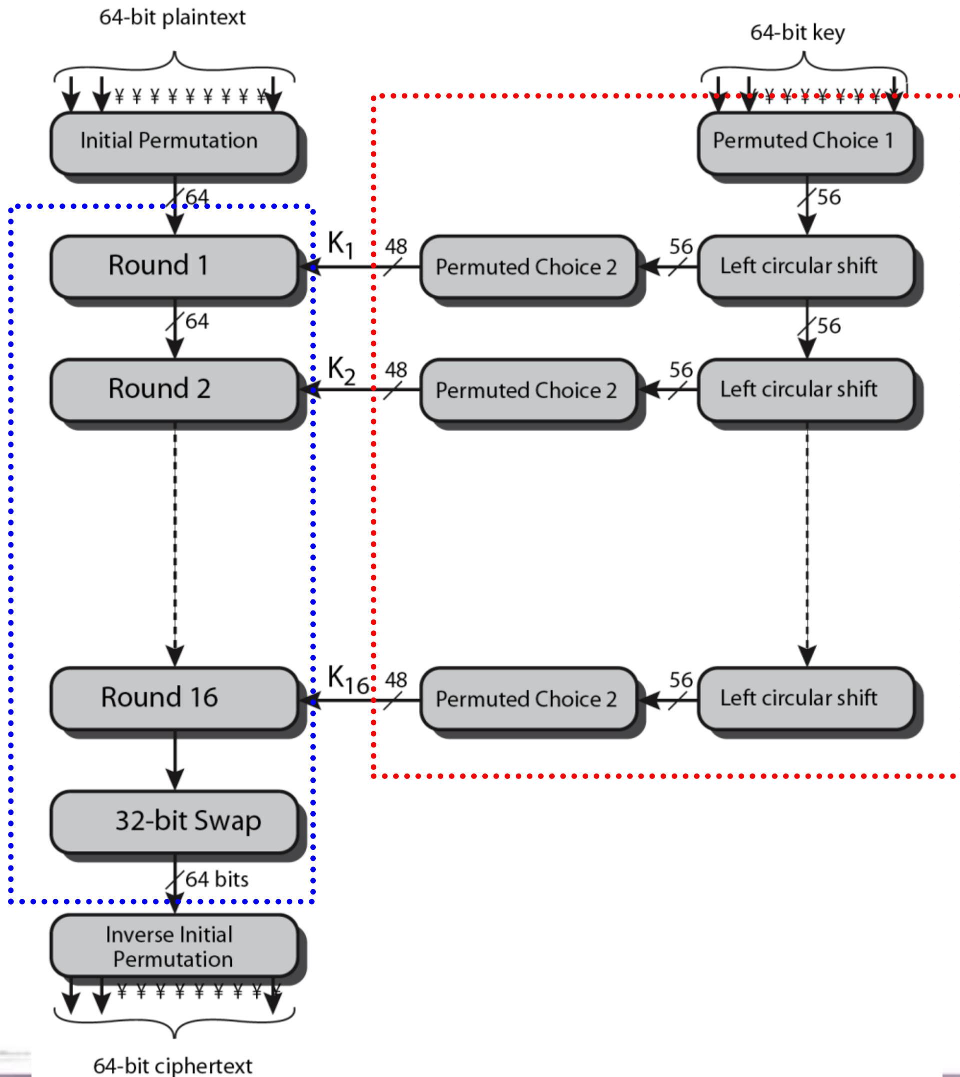
DES History

- IBM developed Lucifer cipher
  - by team led by Feistel in late 60's
  - 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(争议) over design
  - in choice of 56-bit key (vs Lucifer 128-bit)
  - and because design criteria were classified(机密的)
- subsequent events and public analysis show in fact design was appropriate
- use of DES has flourished(风行世界)
  - especially in financial applications
  - still standardised for legacy application use

DES Encryption Overview



A DES Example

Plaintext:	02468aceeca86420
Key:	0f1571c947d9e859
Ciphertext:	da02ce3a89ecac3b

Table 4.2 DES Example

Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

Note: DES subkeys are shown as eight 6-bit values in hex format

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)
```

(a) 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

(b) 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

M:

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

X=IP(M)

M=IP^{-1}(X)

X:

M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_2
M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_4
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_6
M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_8
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_9	M_1
M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_3
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_5
M_{63}	M_{55}	M_{47}	M_{39}	M_{31}	M_{23}	M_{15}	M_7

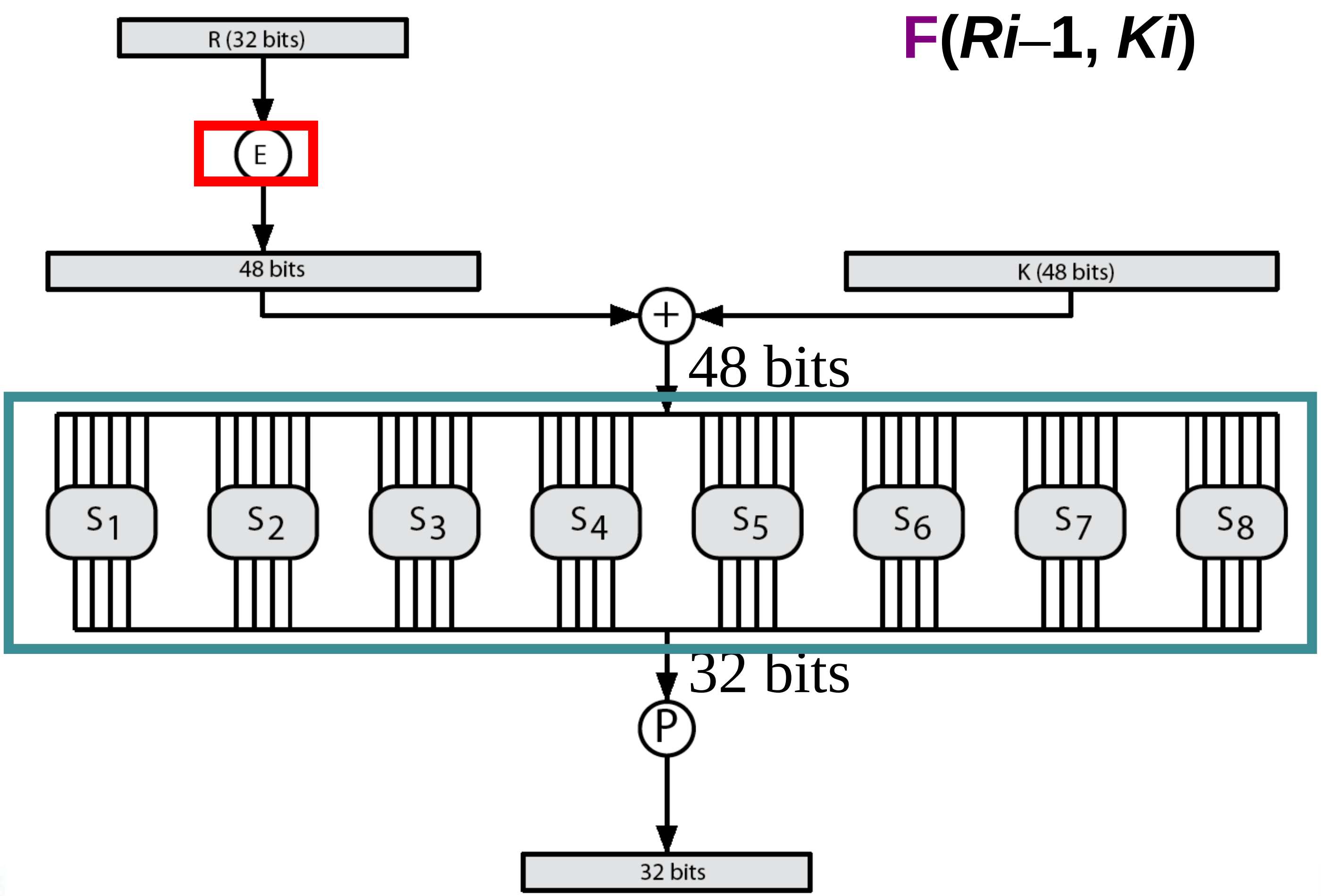
DES Round Structure

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

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

- F takes 32-bit R half and 48-bit subkey:
  - expands 32-bit R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P

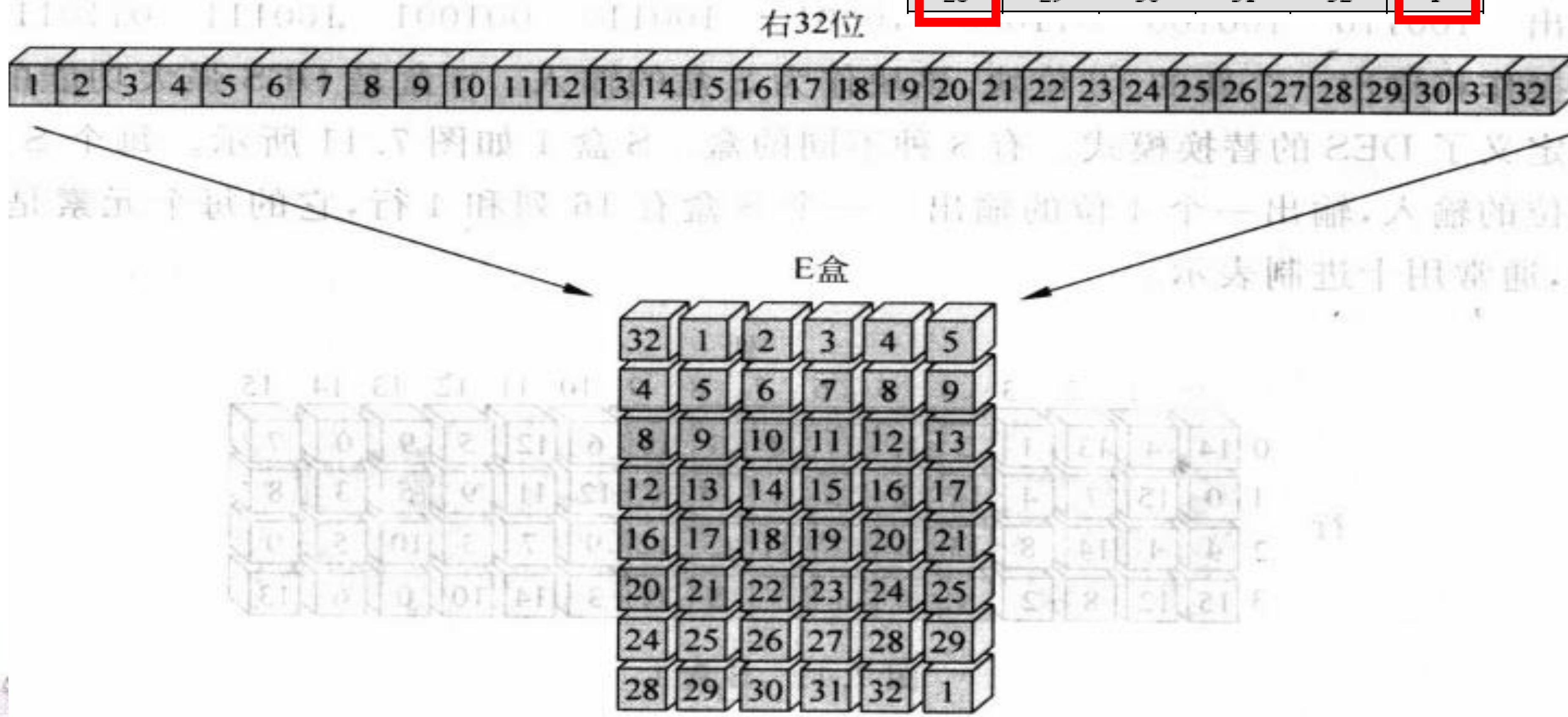
DES Round Structure



Perm E

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



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 row of 4
  - inner bits 2-5 (col bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

S-Box Example

- $x_5 \; x_0$
  $\underbrace{1 \quad 0}$

$x_5 \quad x_4 \quad x_3 \quad x_2 \quad x_1 \quad x_0$

$1 \quad \underbrace{0 \quad 1 \quad 1 \quad 0} \quad 0$

(y_3, y_2, y_1, y_0)
$=(0,0,1,0)$

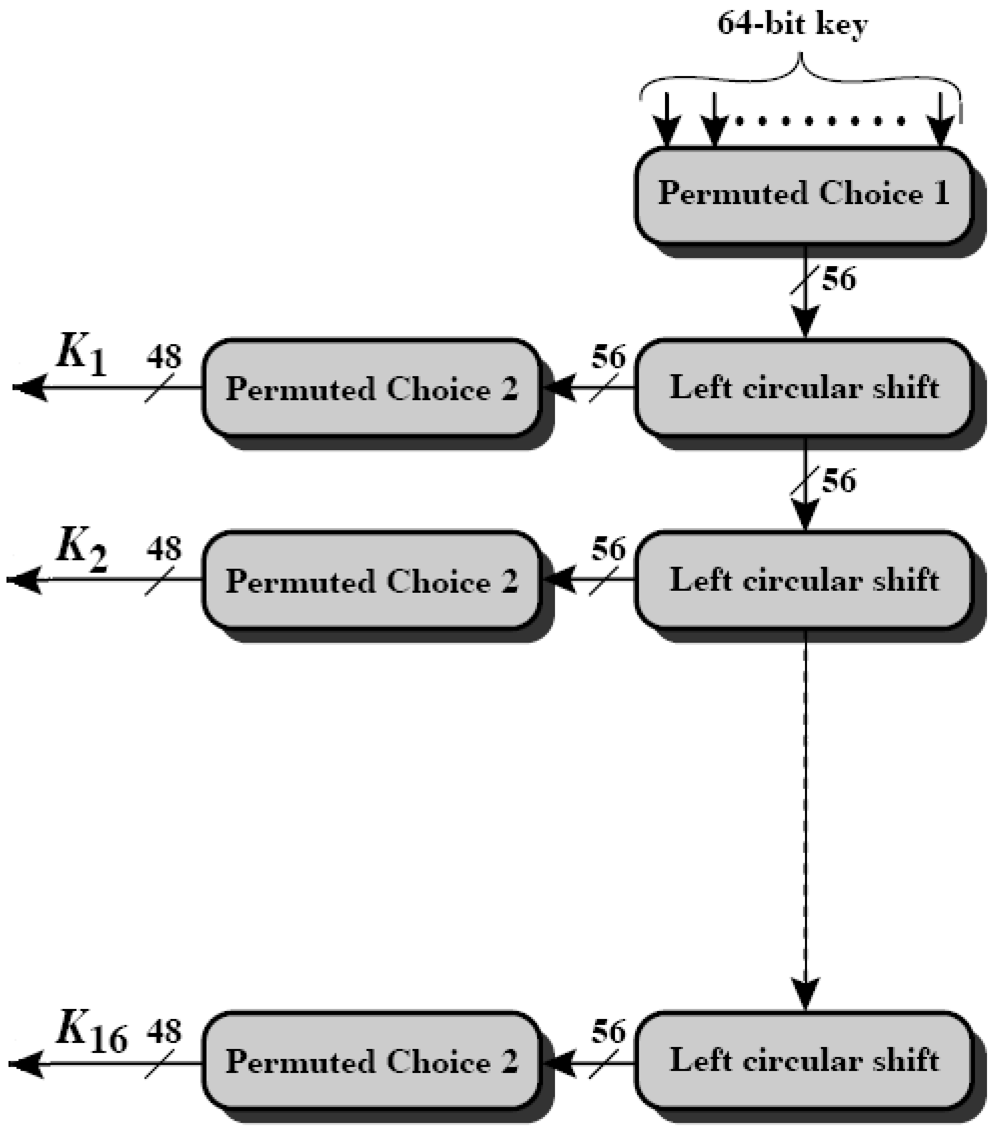
列号 / 行号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	2	14	10	0	6	13

Perm P

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

DES Key Schedule



DES Key Schedule

- forms subkeys used in each round
  - 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 & permuting them by** PC2 **for use in round function F**

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_{16} … SK_1)
  - IP undoes final FP step of encryption
  - 1st round with SK_{16} undoes 16th encrypt round
  - ….
  - 16th round with SK_1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

Avalanche(雪崩) 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(雪崩)**

Table 4.3 Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

the original key, **0f1571c947d9e859**
the altered key, **1f1571c947d9e859**

Table 4.4 Avalanche Effect in DES: Change in Key

Round		δ	Round		δ
	02468aceeca86420 02468aceeca86420	0	**9**	c11bfc09887fbc6c 548f1de471f64dfd	34
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3	**10**	887fbc6c600f7e8b 71f64dfd4279876c	36
2	bad2284599e9b723 9ad628c59939136b	11	**11**	600f7e8bf596506e 4279876c399fdc0d	32
3	99e9b7230bae3b9e 9939136b768067b7	25	**12**	f596506e738538b8 399fdc0d6d208dbb	28
4	0bae3b9e42415649 768067b75a8807c5	29	**13**	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
5	4241564918b3fa41 5a8807c5488dbe94	26	**14**	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
6	18b3fa419616fe23 488dbe94aba7fe53	26	**15**	56b0bd7575e8fd8f d2c3a56f2765c1fb	27
7	9616fe2367117cf2 aba7fe53177d21e4	27	**16**	75e8fd8f25896490 2765c1fb01263dc4	30
8	67117cf2c11bfc09 177d21e4548f1de4	32	**IP^{-1}**	da02ce3a89ecac3b ee92b50606b62b0b	30

Complementation property

- **Let E denote DES, and x the bitwise complement of x. Then**
  - **$y = E_K(x)$ => $\overline{y} = E_{\overline{K}}(\overline{x})$.**
  - **bitwise complementing both the key K and the plaintext x results in complemented DES ciphertext.**

Strength of DES – Key Size

- **56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values**
  - **brute-force attack: on average, half the key space has to be searched**
  - **a brute-force attack to DES: $O(2^{55})$**
- **brute force search looks hard**
- **recent advances have shown is possible**
- **still must be able to recognize plaintext**
- **must now consider alternatives to DES**
  - **3DES, AES**

Table 4.5 Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns $= 1.125$ years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns $= 5.3 \times 10^{21}$ years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns $= 5.8 \times 10^{33}$ years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns $= 9.8 \times 10^{40}$ years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns $= 1.8 \times 10^{60}$ years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns $= 6.3 \times 10^{9}$ years	6.3×10^{6} years

Reducing exhaustive key search

- Use complementation property(互补性)
  - if $C=E_K(P)$,then $\overline{C}=E_{\overline{K}}(\overline{P})$
- reduce the expected number of keys required before success from 2^{55} to 2^{54} to a cryptanalyst in chosen-plaintext exhaustive key search.
- This is not a practical concern.
- If some attacker knows (M,C_1) and ($\overline{M}$,C_2) where C_1=E(K,M) and C_2=E(K,$\overline{M}$).
- Now the attacker need to guess value of K by brute-force attack.
- So, the attacker try all possible key values k1, k2, ..., untill he finds the correct key K.

- **For eack possible key ki,**
  - **Step1: Compute E(ki,M).**
  - **Step2: Judge whether ki is correct key K.**
    - **if E(ki,M)==C_1, then K=ki**
  - **Step3: Judge whether $\overline{k}_i$ is correct key K.**
    - if $E(k_i, M) == \overline{C}_2$, then $E(\overline{k}_i, \overline{M}) = C_2$ that can deduce $\overline{k}_i$ is the correct key K.

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 including
  - 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 information about 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

Differential Cryptanalysis

- **one of the most significant recent (public) advances in cryptanalysis**
- **known by NSA in 70's cf DES design**
- **Murphy, Biham & Shamir published in 90's**
- **powerful method to analyse block ciphers**
- **used to analyse most current block ciphers with varying degrees of success**

- **DES reasonably resistant to it, cf(compare) Lucifer**
- **can attack DES with 2^{47} chosen plaintexts($\approx O(2^{55.1})$), easier but still in practise infeasible**

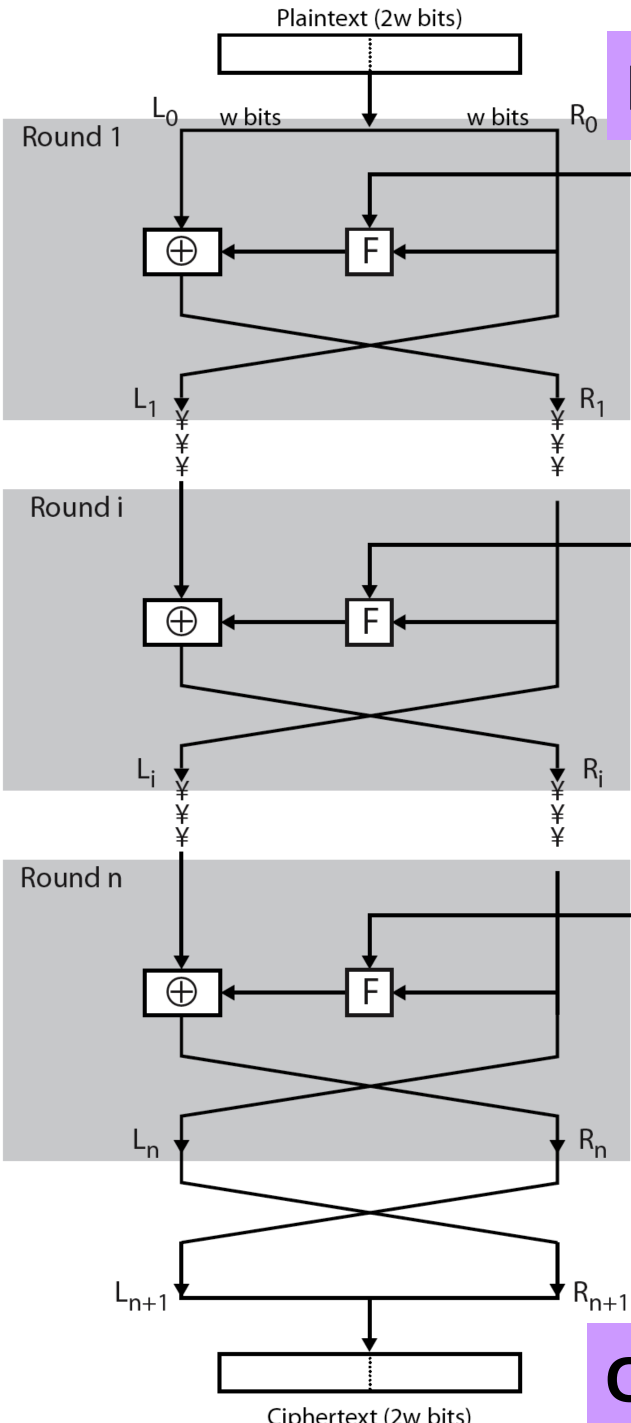
Differential Cryptanalysis

- a statistical attack against Feistel ciphers
- uses cipher structure not previously used
- design of S-P networks has output of function *f* influenced by both input & key
- hence cannot trace values back through cipher without knowing value of the key
- differential cryptanalysis compares two related pairs of encryptions

Differential Cryptanalysis Compares Pairs of Encryptions

- with a known difference in the input
- searching for a known difference in output
- when same subkeys are used

$$\begin{aligned}\Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\ &= [m_{i-1} \oplus f(m_i, K_i)] \oplus [m'_{i-1} \oplus f(m'_i, K_i)] \\ &= \Delta m_{i-1} \oplus [f(m_i, K_i) \oplus f(m'_i, K_i)]\end{aligned}$$



$M=m_0 \,||\, m_1$

$E(M,K)=C$

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

$m_1 \,||\, m_2$

$m_{i+1} = m_{i-1} \oplus F(m_i, k_i)$

$m_i \,||\, m_{i+1}$

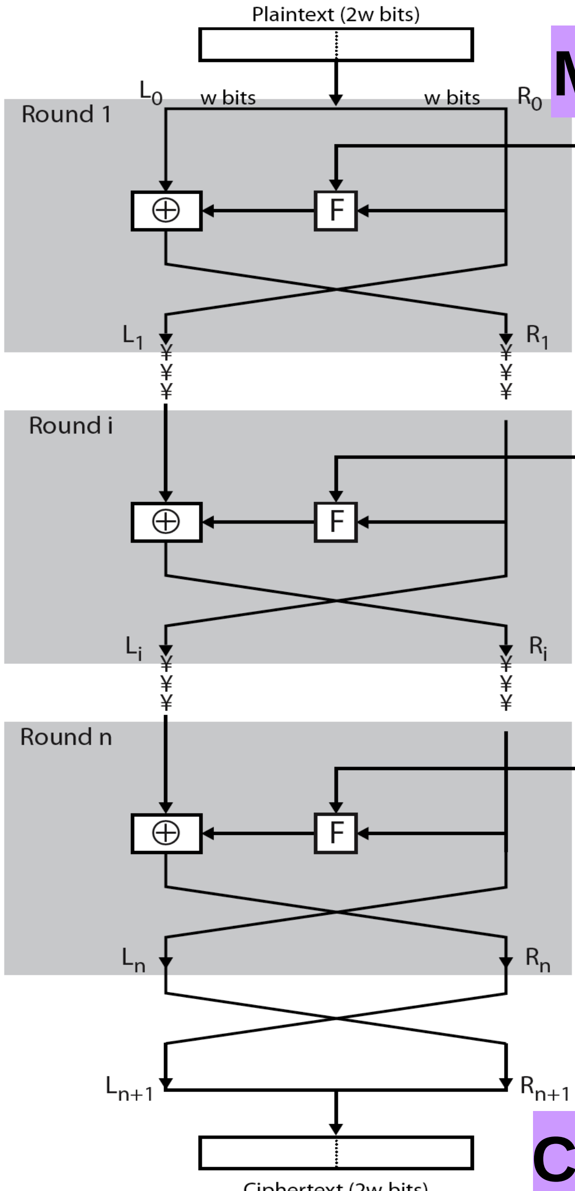
$m_n \,||\, m_{n+1}$

$C=m_{n+1} \,||\, m_n$

$$m_{i+1} = m_{i-1} \oplus F(m_i, k_i),$$
$$\Delta m_i = m_i \oplus m'_i$$

$$\Delta m_{i+1} = m_{i+1} \oplus m'_{i+1}$$
$$= \Delta m_{i-1} \oplus [F(m_i, k_i) \oplus F(m'_i, k_i)]$$

$M = m_0 \,||\, m_1$



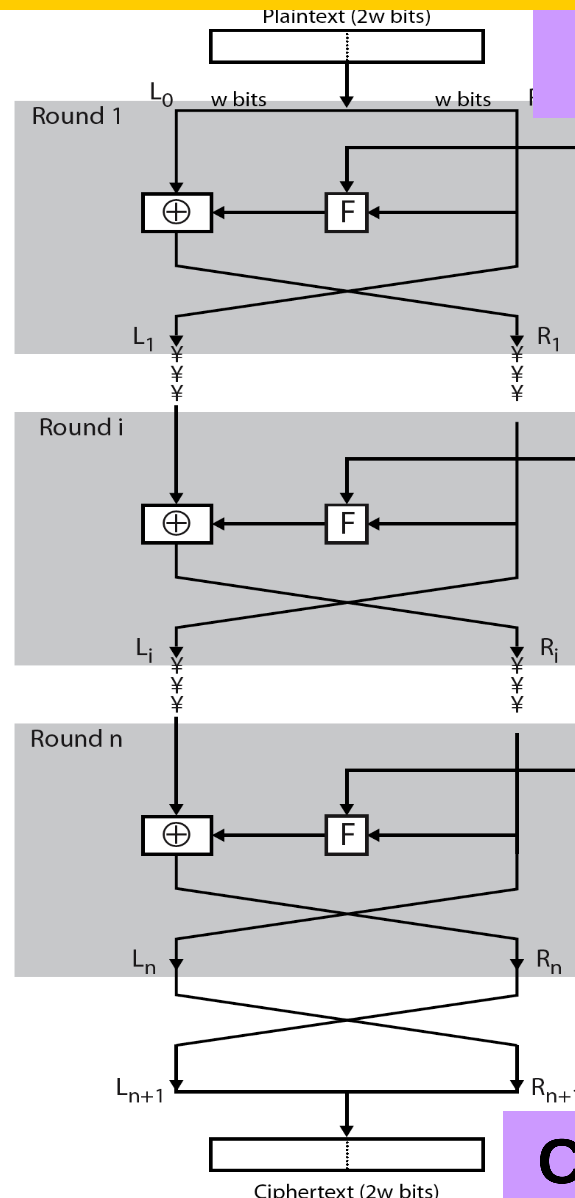
$m_1 \,||\, m_2$

$m_i \,||\, m_{i+1}$

$m_n \,||\, m_{n+1}$

$C = m_{n+1} \,||\, m_n$

$M' = m'_0 \,||\, m'_1$



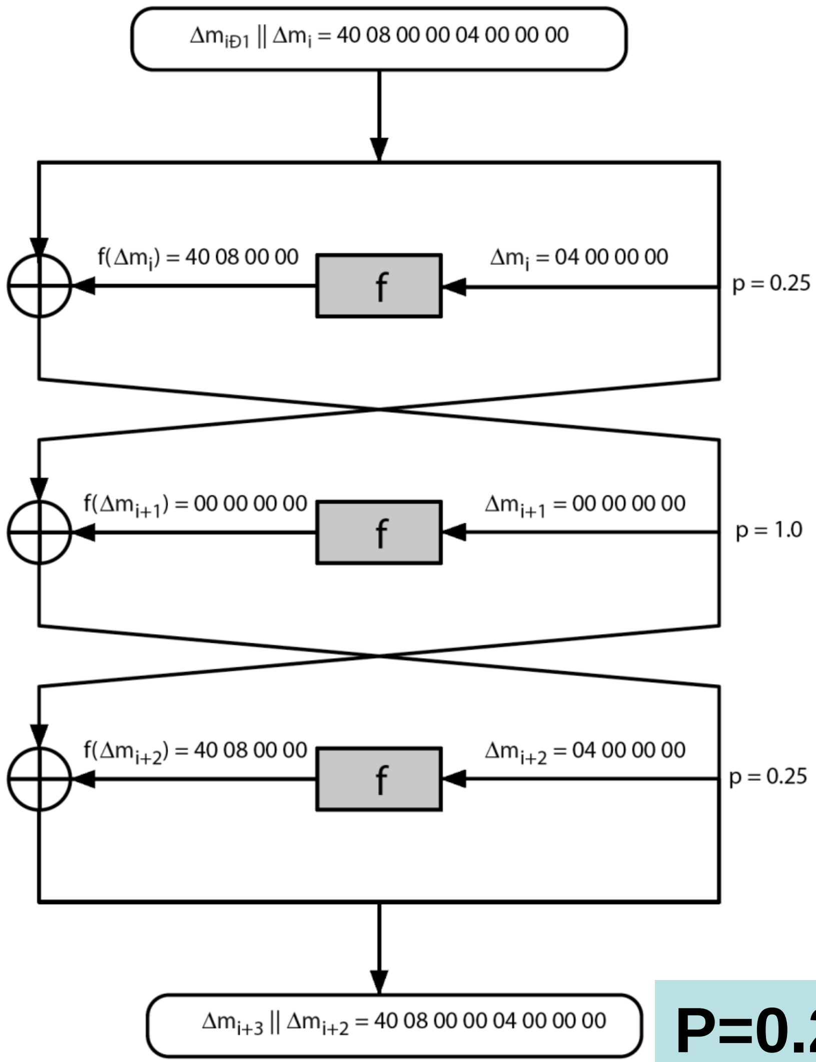
$m'_1 \,||\, m'_2$

$m'_i \,||\, m'_{i+1}$

$m'_n \,||\, m'_{n+1}$

$C' = m'_{n+1} \,||\, m'_n$

Differential Cryptanalysis



P=0.25*1*0.25=0.0625

Differential Cryptanalysis

- have some input difference giving some output difference with probability p
- if find instances of some higher probability input / output difference pairs occurring
- can infer subkey that was used in round
- then must iterate process over many rounds (with decreasing probabilities)

Differential Cryptanalysis

- **perform attack by repeatedly encrypting plaintext pairs with known input XOR until obtain desired output XOR**
- **when found**
  - **if intermediate rounds match required XOR, have a** right pair
  - **if not, then have a** wrong pair**, relative ratio is S/N for attack**
- **can then deduce keys values for the rounds**
  - **right pairs suggest same key bits**
  - **wrong pairs give random values**
- **for large numbers of rounds, probability is so low that more pairs are required than exist with 64-bit inputs**
- **Biham and Shamir have shown how a 13-round iterated characteristic can break the full 16-round DES**

Linear Cryptanalysis

- another recent development
- also a statistical method
- must be iterated over rounds, with decreasing probabilities
- developed by Matsui et al in early 90's
- based on finding linear approximations
- can attack DES with 2^{43} known plaintexts, easier but still in practise infeasible

Linear Cryptanalysis

- **find linear approximations with prob p≠0.5**

  $P[i_1,i_2,...,i_a] \oplus C[j_1,j_2,...,j_b] = K[k_1,k_2,...,k_c]$

  where i_a, j_b, k_c are bit locations in P,C,K

- **gives linear equation for key bits**
- **using a large number of trial encryptions**

DES Design Criteria

- as reported by Coppersmith in [COPP94]
- 7 criteria for S-boxes provide for
  - non-linearity
  - resistance to differential cryptanalysis
  - good confusion
- 3 criteria for permutation P provide for
  - increased diffusion

Block Cipher Design

- basic principles still like Feistel's in 1970's
- number of rounds
  - more is better, exhaustive search best attack
  - For 16 rounds, efficiency of differential cryptanalysis is worse than exhaustive search attack.
- function f:
  - provides "confusion", is nonlinear, avalanche
  - have issues of how S-boxes are selected
- key schedule
  - complex subkey creation, key avalanche

Summary

- have considered:
  - block vs stream ciphers
  - Feistel cipher design & structure
  - DES
    - details
    - strength
  - Differential & Linear Cryptanalysis
  - block cipher design principles

Key Terms

avalanche effect	Feistel cipher	round
block cipher	irreversible mapping	round function
confusion	key	subkey
Data Encryption Standard (DES)	permutation	substitution
diffusion	product cipher	
	reversible mapping	

Review Questions

- **4.3** Why is it not practical to use an arbitrary reversible substitution cipher of the kind shown in Table 4.1?
- **4.5** What is the difference between diffusion and confusion?
- **4.6** Which parameters and design choices determine the actual algorithm of a Feistel cipher?
- **Problem 4.1, see P141.**

Thanks!