Cryptography and Network Security Chapter 3

- Fifth Edition by William Stallings
- Lecture slides by Lawrie Brown

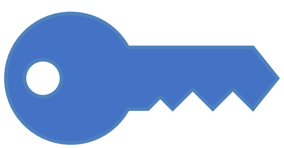
Modern Block Ciphers & Data Encryption Standard (DES)

Arabic Language Audio Comments by Prof. Mohamed Ashraf Madkour

Modern Block Ciphers

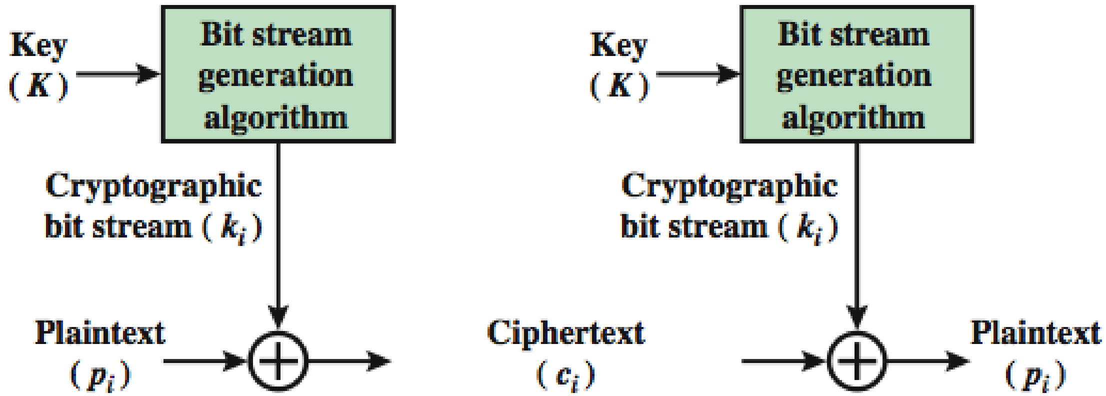
- now look at modern block ciphers
- one of the most widely used types of cryptographic algorithms
- provide secrecy /authentication services
- focus on DES (Data Encryption Standard)
- to illustrate block cipher design principles

Block vs Stream Ciphers

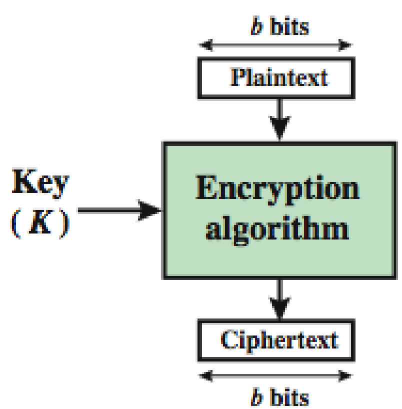


- block ciphers process messages in blocks, each of which is then encrypted / decrypted
- block ciphers look like a substitution on very big bit patterns (characters) of 64-bits or more
- stream ciphers process messages a bit or byte at a time when encrypting or decrypting
- many current ciphers are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

Block vs Stream Ciphers



(a) Stream Cipher Using Algorithmic Bit Stream Generator

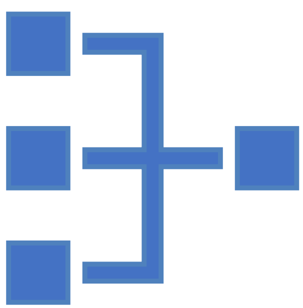


(b) Block Cipher

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
- using idea of a product cipher

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

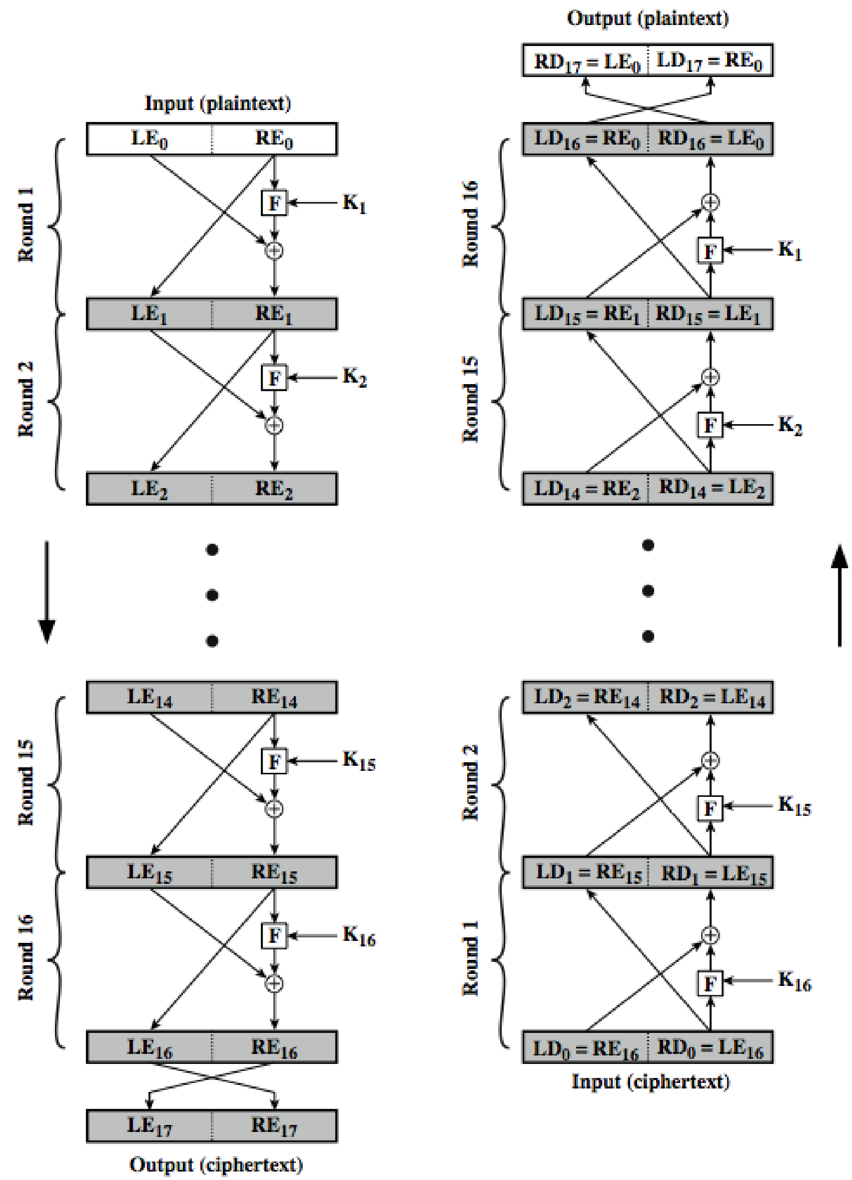
Confusion and Diffusion

- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S & P elements to obtain:
- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as possible

Feistel Cipher Structure

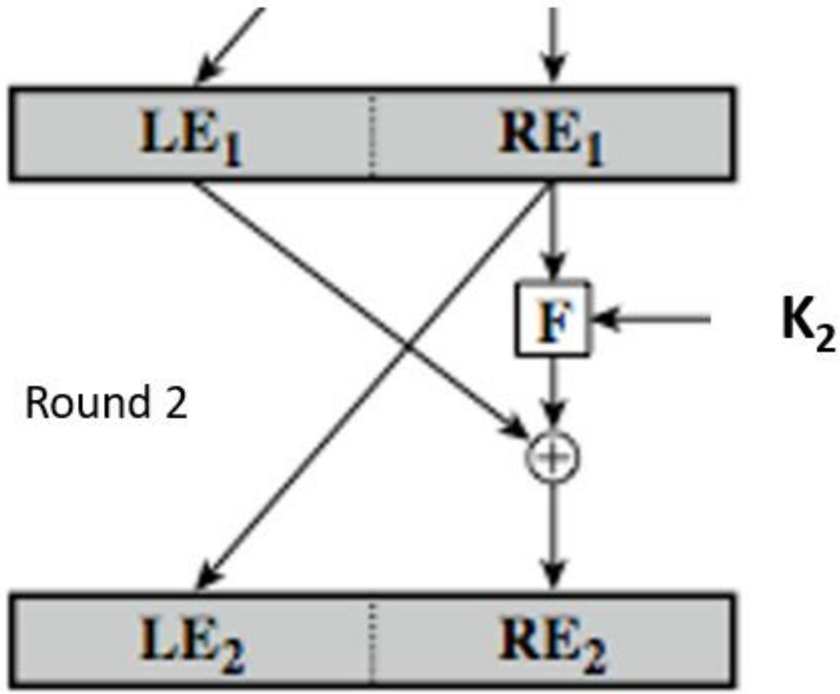
- Horst Feistel devised the **feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves (the next diagram describes the following encryption/decryption operations)
  - process through multiple rounds which perform a substitution on left data half based on round function of right half & subkey then have permutation swapping halves
- implements Shannon's S-P net concept

Feistel Cipher Structure



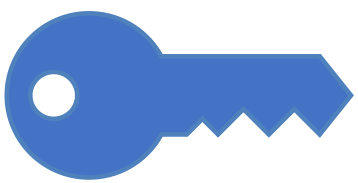
Detailed Description of Round 2 during Encryption

- LE_1: left-half input obtained from the output of round 1
- RE_1: right-half input obtained from the output of round 1
- F: cryptographic function containing substitution and permutation (this function is the same for all rounds)
- K_2: subkey for round 2 (each round has its own different subkey)
- LE_2 & RE_2: left and right-half outputs of round 3 (this is the input for round 3, and so on)



Feistel Cipher Design Elements

(Example: DES Cipher)

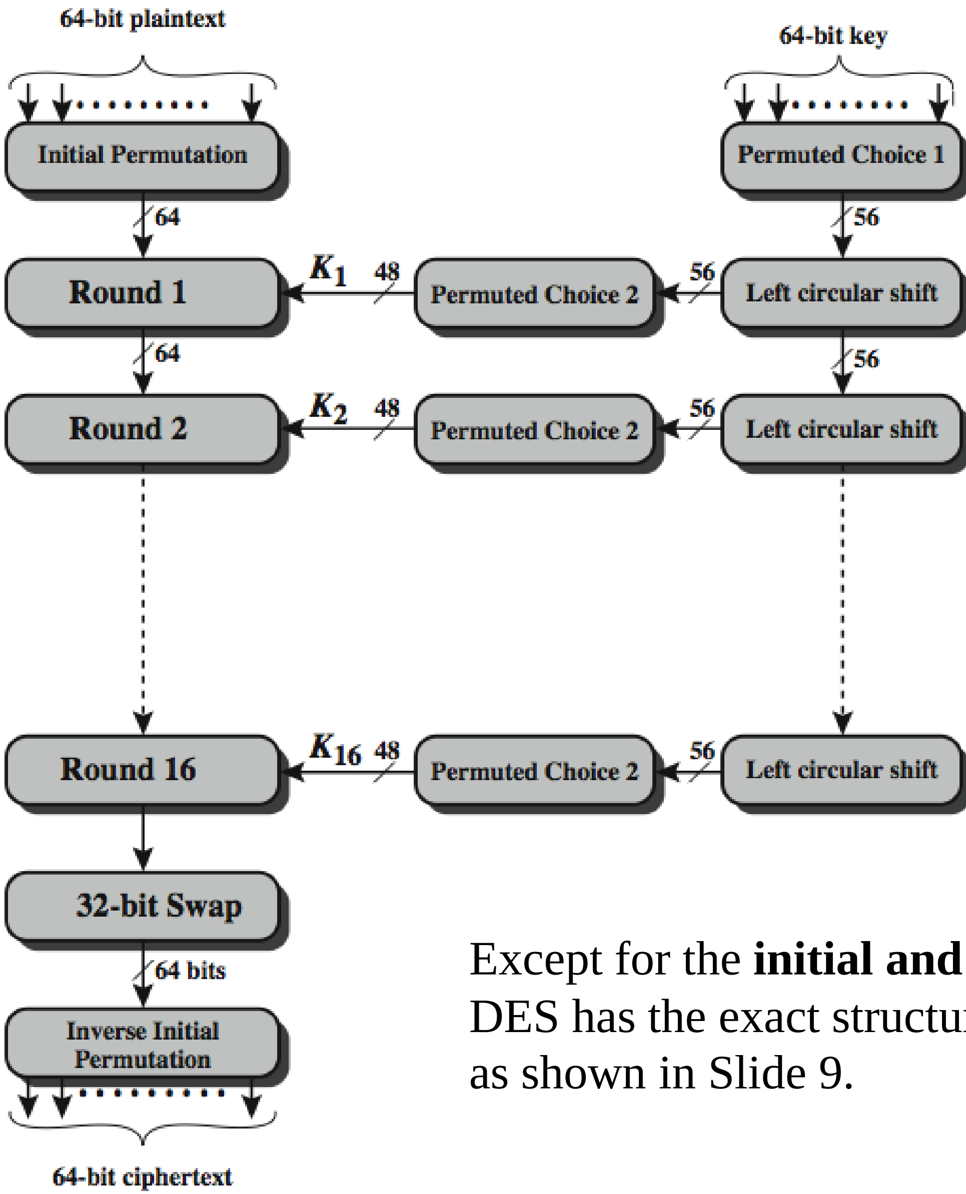


- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software encryption/decryption
- ease of analysis

Data Encryption Standard (DES)

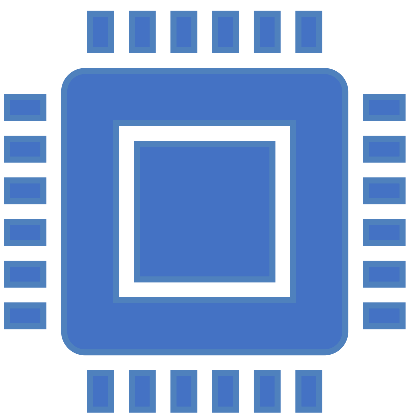
- most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
- broken in 1997 by a brute force attack; now replaced by AES in 2001.
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security
- Use of DES has flourished
  - especially in financial applications
  - still standardised for legacy application use

DES Encryption Overview



Except for the **initial and final permutations**, DES has the exact structure of a Feistel cipher, as shown in Slide 9.

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 implementation)
- example:

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

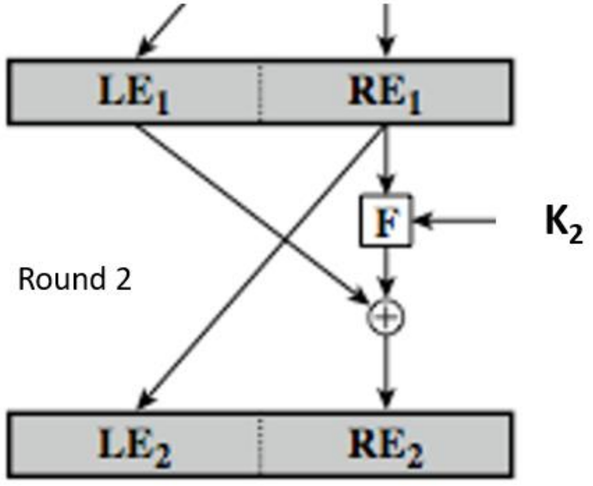
DES Round Structure

(refer to the next diagram)

- 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 R to 48-bits using permutation E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit permutation P



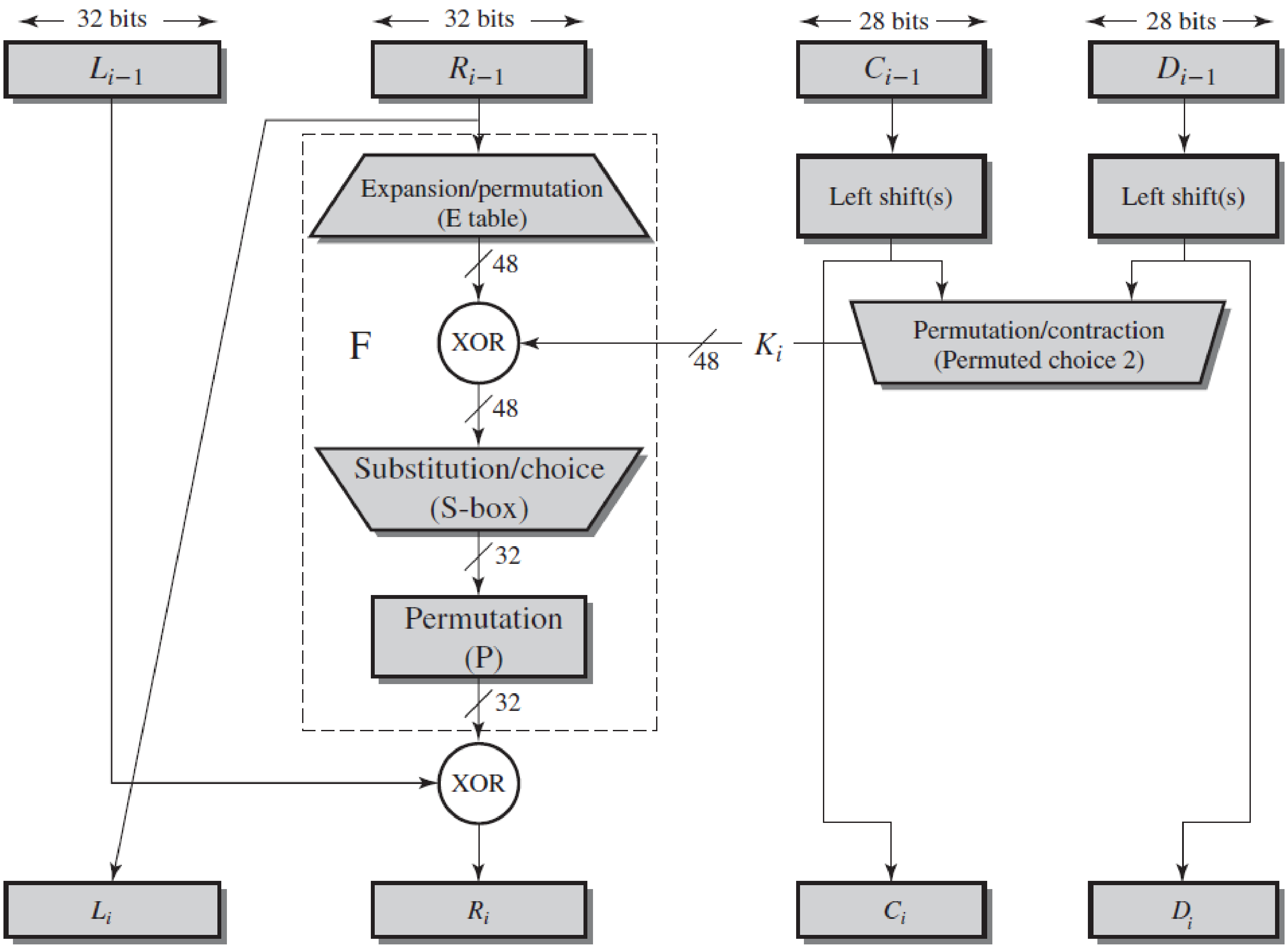


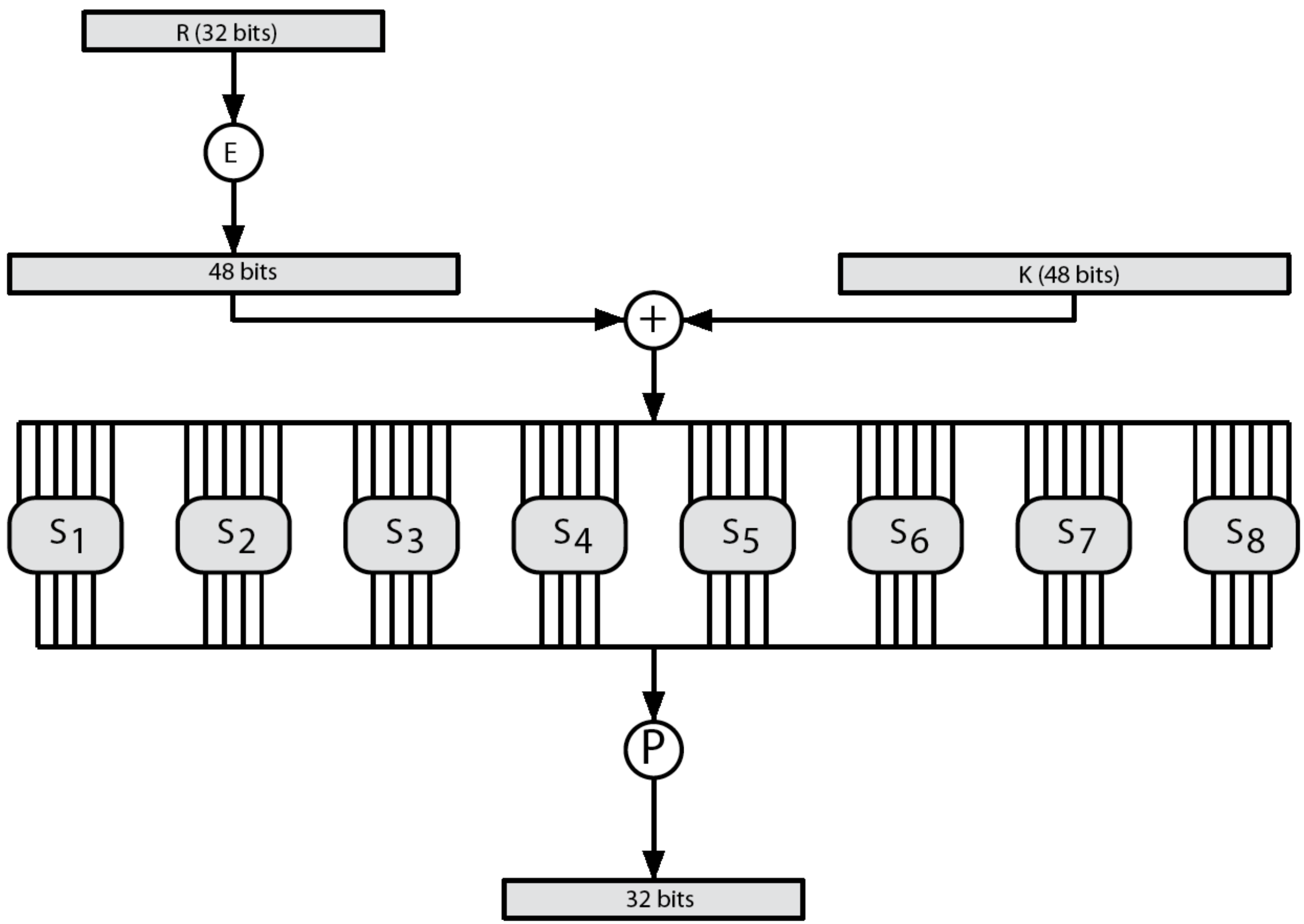
Figure 3.6 Single Round of DES Algorithm

Table 3.3 Definition of DES S-Boxes

	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
S_1	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
	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
S_2	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
	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
S_3	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
	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
S_4	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 Round Structure

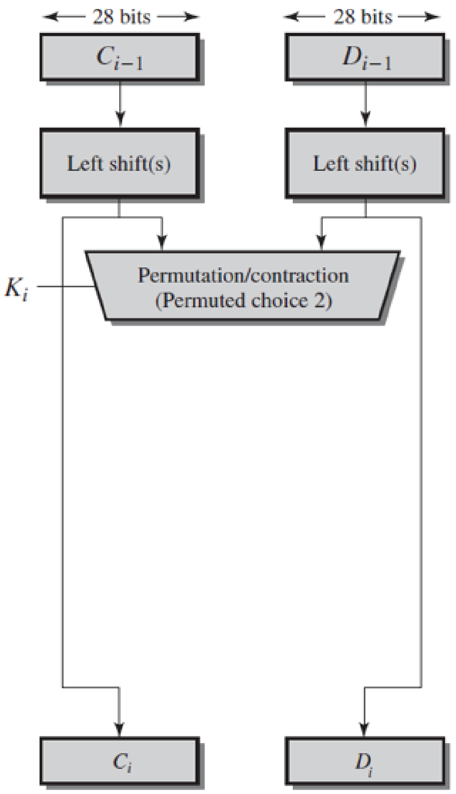


Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is containing four 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
- row selection depends on both data & key
- example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

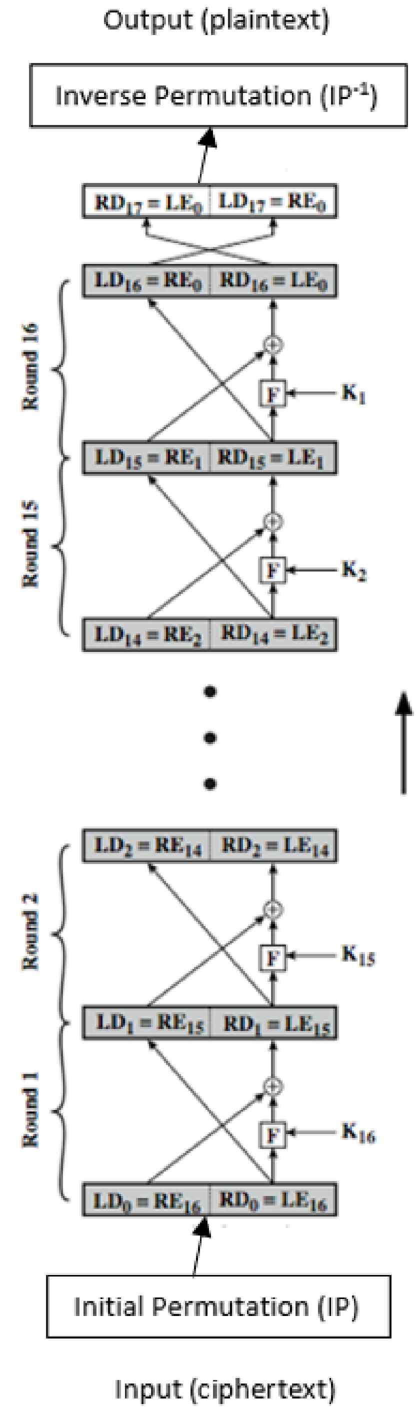
DES Key Schedule

- 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
- note practical use issues in h/w vs s/w



DES Decryption

- decryption must unwind steps of data computation during encryption
- with Feistel design, do encryption steps again using subkeys in reverse order (K_{16} … K_1)
  - IP undoes final permutation step of encryption
  - 1^{st} round with K_{16} undoes 16^{th} encrypt round
  - ….
  - 16th round with K_1 undoes 1st encrypt round
  - then final inverse permutation (IP^{-1}) undoes initial encryption IP
  - thus recovering original data value



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

Avalanche Effect

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

Avalanche in DES

- This table considers encrypting two separate blocks that differ only in one bit (the MSB), i.e. $\delta = 1$.
- As the encryption process proceeds, we notice that δ increases in each successive round.
- At last, we get $\delta = 32$
- This means that 32 bits have changed after encrypting the two blocks.
- Having about half the bits changed (32 bits of a 64-bit block) is the best avalanche.

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
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

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- recent 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
- must now consider 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
  - timing attacks
  - differential cryptanalysis
  - linear cryptanalysis

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

Differential and Linear Cryptanalysis

- Differential Cryptanalysis
  - a statistical attack against Feistel ciphers
  - differential cryptanalysis compares two related pairs of encryptions
- Linear Cryptanalysis
  - another recent development
  - also a statistical method
  - developed by Matsui in early 90's
  - can attack DES with 2^{43} known plaintexts, easier but still in practice infeasible

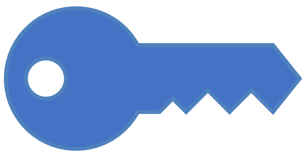
DES Design Criteria

- 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

- number of rounds:
  - more rounds is better
  - exhaustive search is the best attack
- function $F(R_{i-1},K_i)$ provides:
  - confusion
  - nonlinearity
  - strong avalanche
- key schedule provides:
  - 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