Cryptography

Dr. S.R. Shinde

Block vs Stream Ciphers

- block ciphers process messages into 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
- hence are focus of course

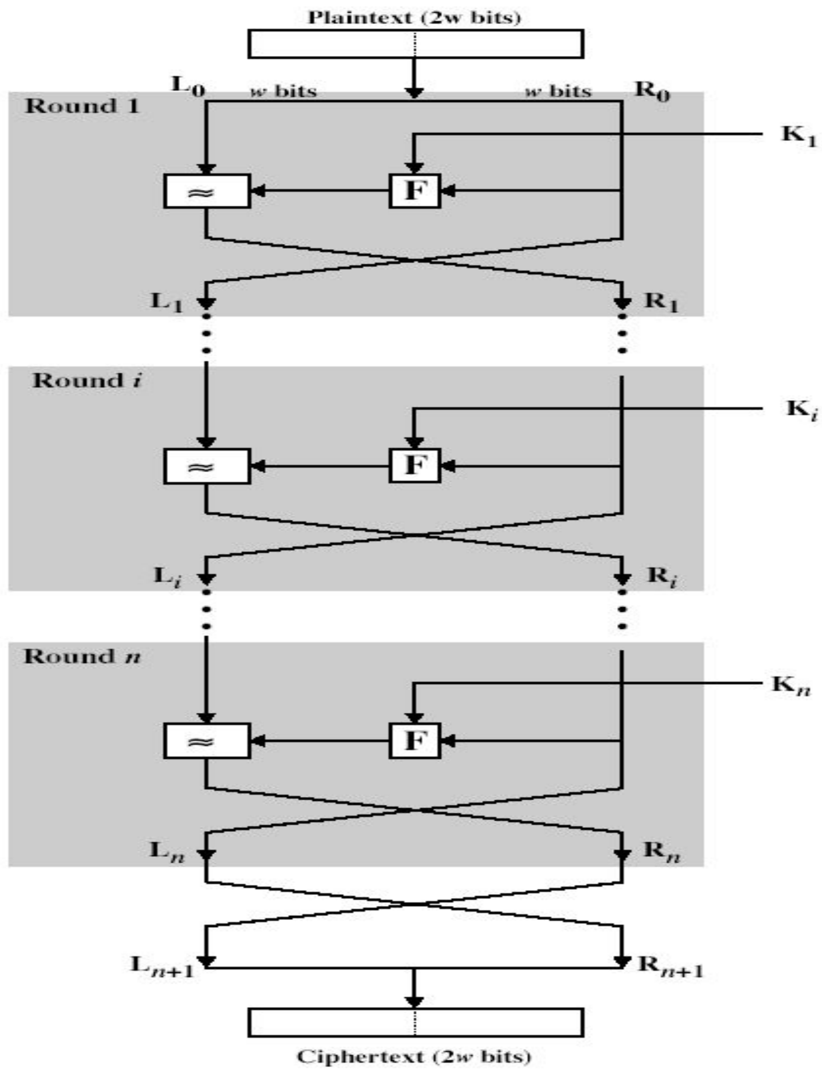
Block Cipher Principles

- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
- arbitrary reversible substitution cipher for a large block size is not practical
  - 64-bit general substitution block cipher, key size 2^{64}!
- most symmetric block ciphers are based on a **Feistel Cipher Structure**
- needed since must be able to **decrypt** ciphertext to recover messages efficiently

Feistel Cipher Structure

- Horst Feistel devised the **feistel cipher**
  - implements Shannon's substitution-permutation network concept
- partitions input block into two halves
  - 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

Feistel Cipher Structure



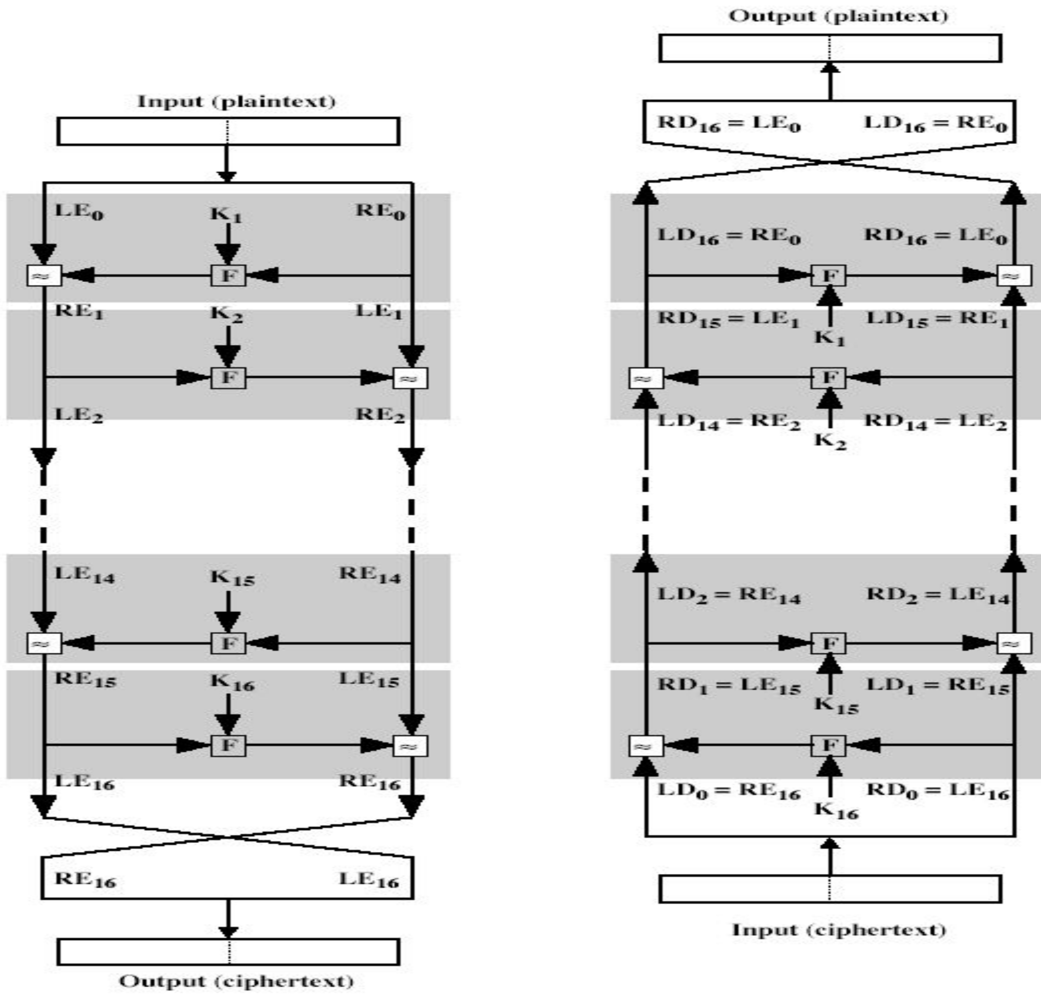
Feistel Cipher

- n sequential rounds
- A substitution on the left half L_i
  - 1. Apply a round function F to the right half R_i and
  - 2. Take XOR of the output of (1) and L_i
- The round function is parameterized by the subkey K_i
  - K_i are derived from the overall key K

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**
  - greater complexity can make analysis harder, but slows cipher
- **fast software en/decryption & ease of analysis**
  - are more recent concerns for practical use and testing

Feistel Cipher Decryption



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

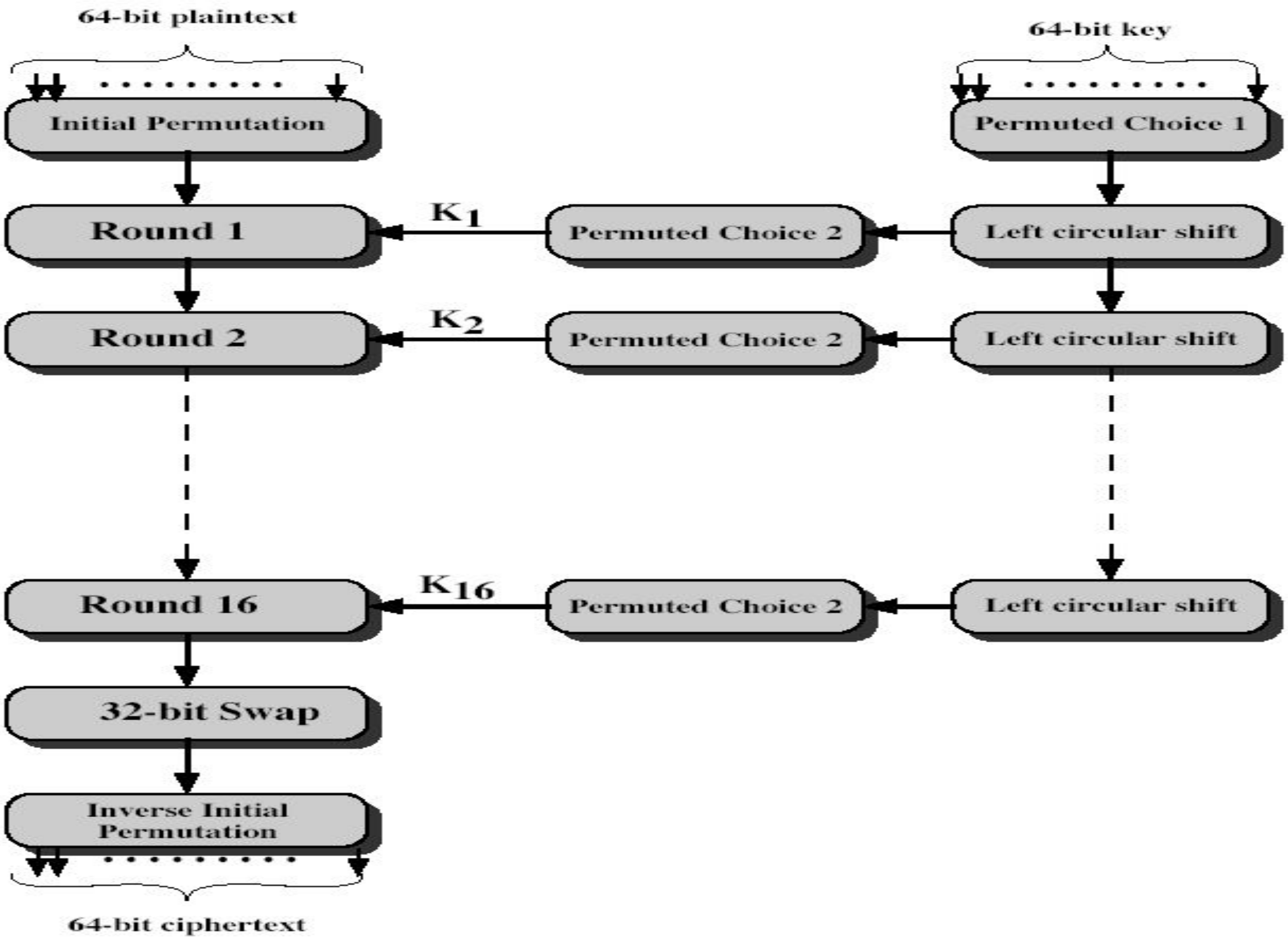
DES History

- 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)
- subsequent events and public analysis show in fact design was appropriate
- DES has become widely used, especially in financial applications

DES Encryption



Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- quite regular in structure
- example:

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

Initial Permutation								*Final Permutation*							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25

DES Round Structure

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

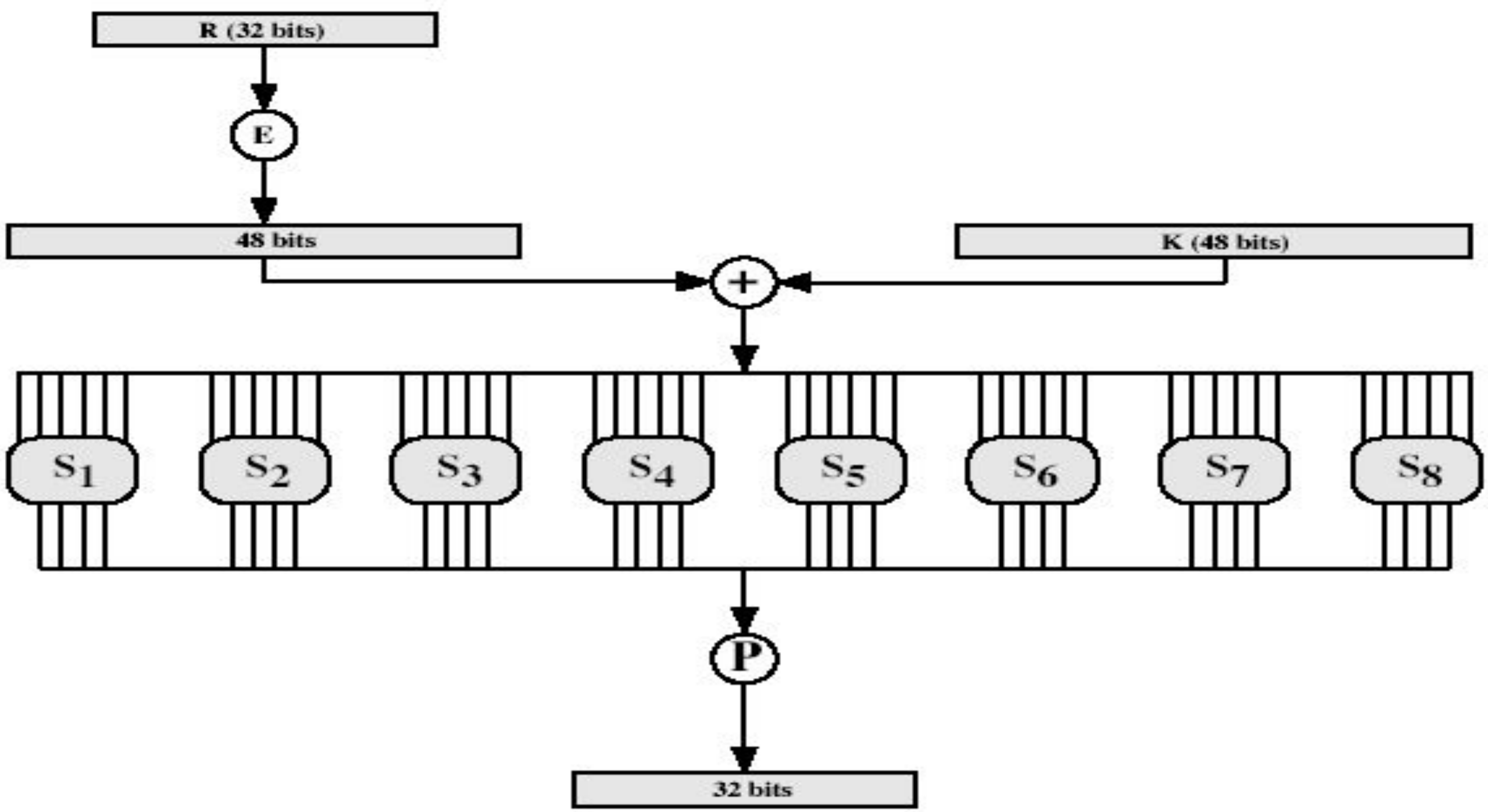
  $L_i = R_{i-1}$

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

Expansion Permutation table

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	31	31	32	01

The round function F(R,K)



Substitution Boxes S

- 8 S-boxes
- Each S-Box mapps 6 to 4 bits
  - outer bits 1 & 6 (**row** bits) select the row
  - inner bits 2-5 (**col** bits) select the column
  - For example, in S1, for input 011001,
    - the row is 01 (row 1)
    - the column is 1100 (column 12).
    - The value in row 1, column 12 is 9
    - The output is 1001.
- result is 8 X 4 bits, or 32 bits

S-Box 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

There are eight S-boxes

DES Key Schedule

- forms subkeys used in each round
- 1. initial permutation of the key PC1
- 2. divide the 56-bits in two 28-bit halves
- 3. at each round
  - 3.1. Left shift each half (28bits) separately either 1 or 2 places based on the left shift schedule
    - Shifted values will be input for next round
  - 3.2. Combine two halfs to 56 bits, permuting them by PC2 for use in function f
    - PC2 takes 56-bit input, outputs 48 bits

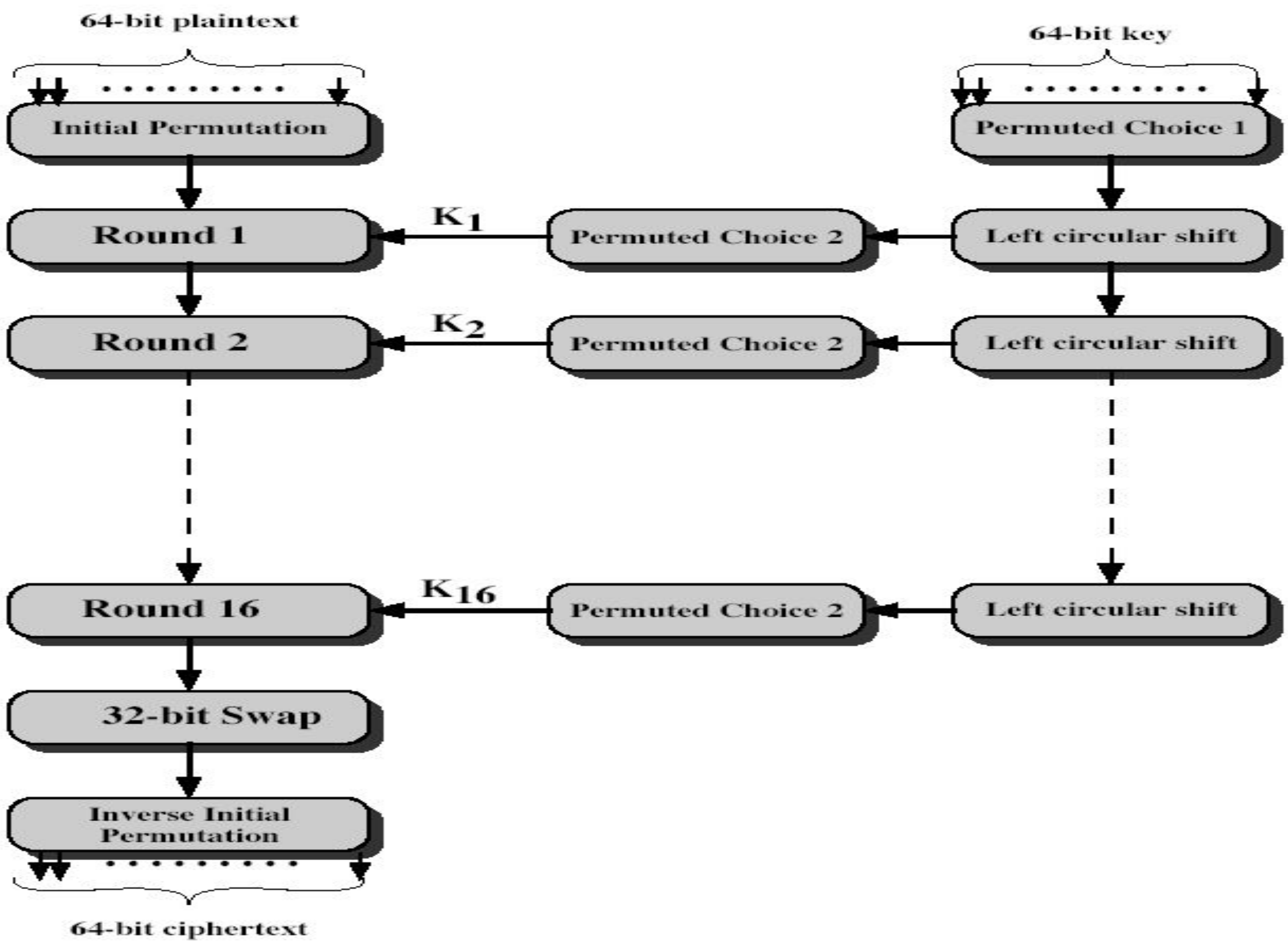
Permutation table (32-bit)

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again
- using subkeys in reverse order (SK16 … SK1)
- note that IP undoes final FP step of encryption
- 1st round with SK16 undoes 16th encrypt round
- ….
- 16th round with SK1 undoes 1st encrypt round
- then final FP undoes initial encryption IP
- thus recovering original data value

DES Decryption (reverse encryption)



Avalanche Effect

- key desirable property of encryption algorithm
- DES exhibits strong avalanche
- where a change of **one** input or key bit results in changing approx **half** output bits
- In cryptography, the **avalanche effect** is the desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions, wherein if an input is changed slightly (for example, flipping a single bit), the output changes significantly

Strength of DES (cont.)

- Avalanche effect in DES
  - If a small change in either the plaintext or the key, the ciphertext should change markedly.
- DES exhibits a strong avalanche effect.

(a) Change in Plaintext		(b) Change in Key	
Round	Number of bits that differ	Round	Number of bits that differ
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

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 hardware (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 – 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

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