SYMMETRIC KEY CRYPTO & Block Ciphers

Falguni Roy

Last Chapter

- have considered:
  - terminology
  - classical cipher techniques
  - substitution ciphers
    - cryptanalysis using letter frequencies
  - transposition ciphers

Modern Block Ciphers

- Will now look at modern block ciphers
- One of the most widely used types of cryptography algorithms
- Provide strong secrecy and/or authentication services
- In particular will introduce DES (data encryption standard)

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

C. Shannon and Substitution-Permutation Ciphers

- In 1949 Shannon introduced idea of substitution-permutation (S-P) networks
  - Modern substitution-transposition product cipher
- 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) (transposition)
- Provide *confusion* and *diffusion* of message

Diffusion and Confusion

- Introduced by Claude Shannon to thwart cryptanalysis based on statistical analysis
  - confusion and diffusion are two properties of the operation of a secure cipher
  - Assume the attacker has some knowledge of the statistical characteristics of the plaintext
- Cipher needs to completely obscure statistical properties of original message
- A one-time pad does this

Diffusion and Confusion

- More practically shannon suggested combining elements to obtain:
- **Diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
  - Diffusion means that if we change a single bit of the plaintext, then (statistically) half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change.
- **Confusion** – makes relationship between ciphertext and key as complex as possible
  - Confusion means that each binary digit (bit) of the ciphertext should depend on several parts of the key, obscuring the connections between the two

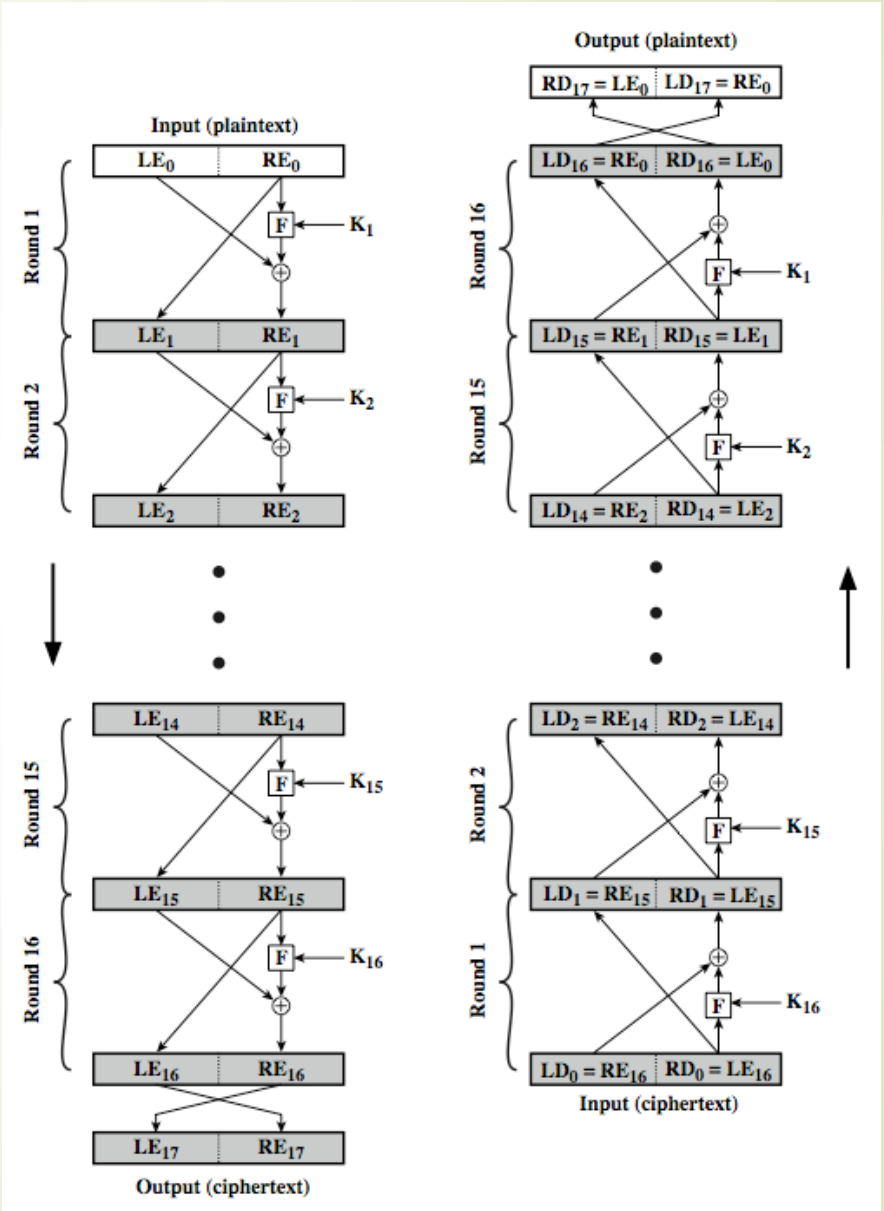
Feistel Cipher Structure

- 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

$$L_i = R_{i-1}$$

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



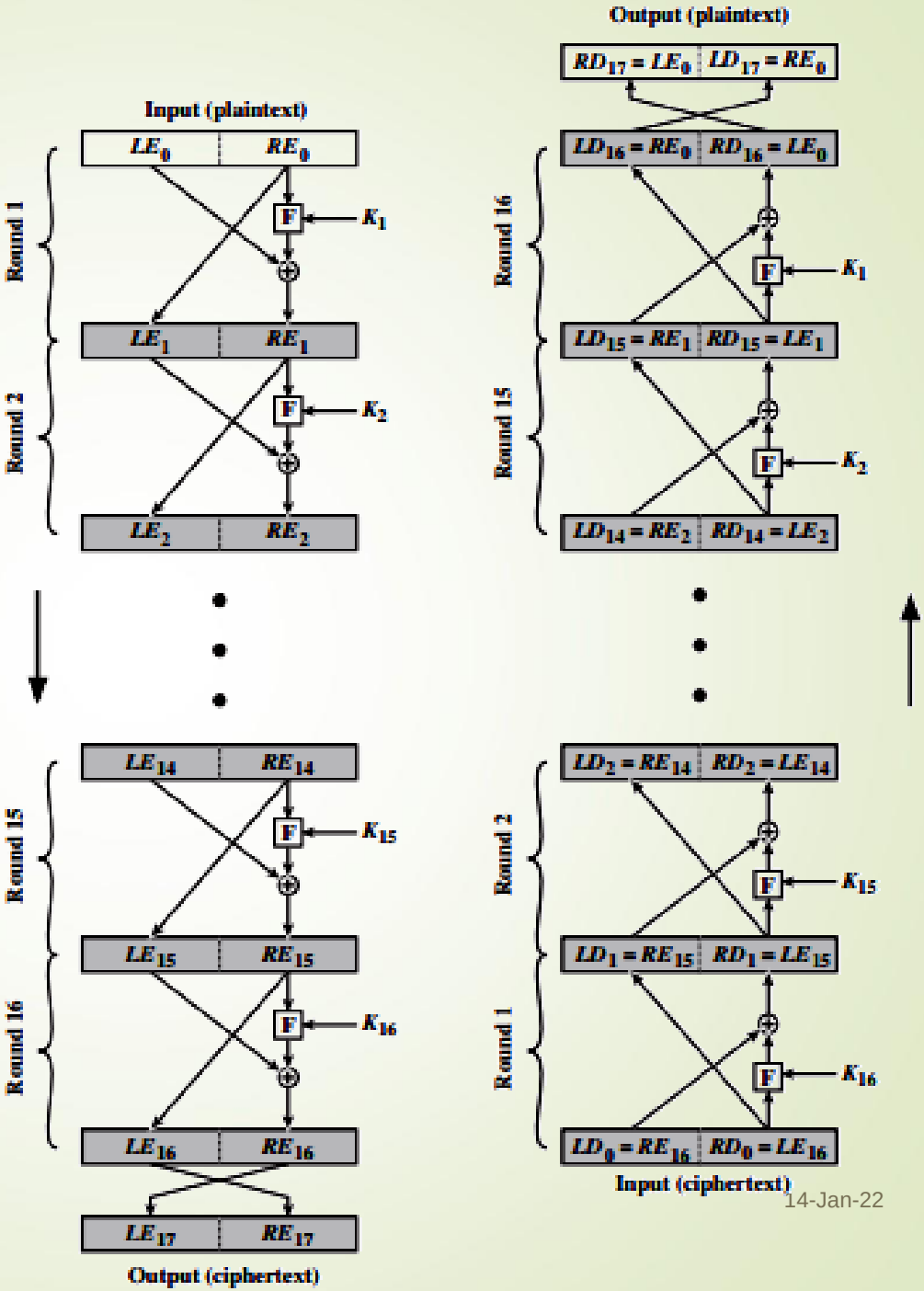
Feistel Cipher

- N sequential rounds
- A substitution on the left half l_i
  - Apply a round function F to the right half r_i and
  - 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



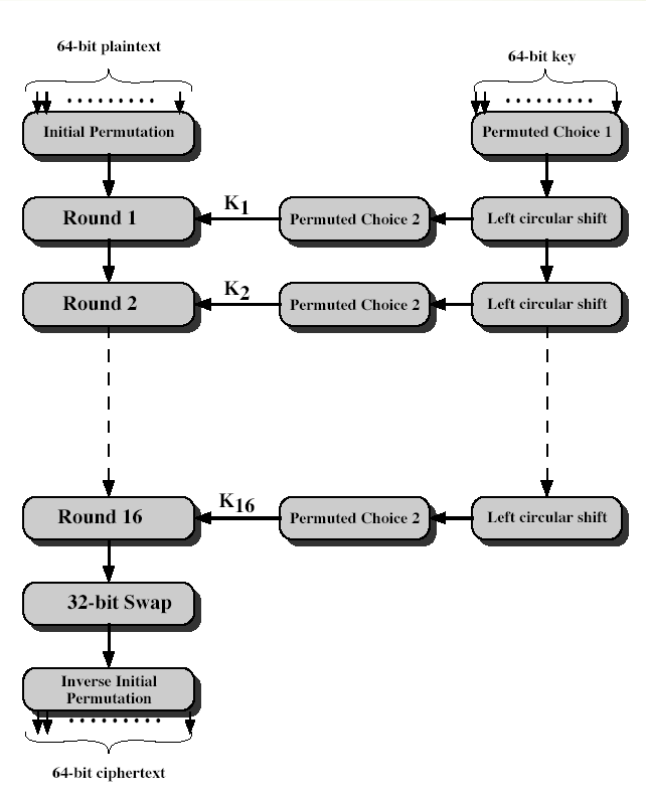
Data Encryption Standard (DES)

- Most widely used block cipher in world
- Adopted in 1977 by NBS (now NIST)
- DES is a Feistel cipher with 16 rounds;
- DES has a 64-bit block length;
- DES uses a 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 Encryption

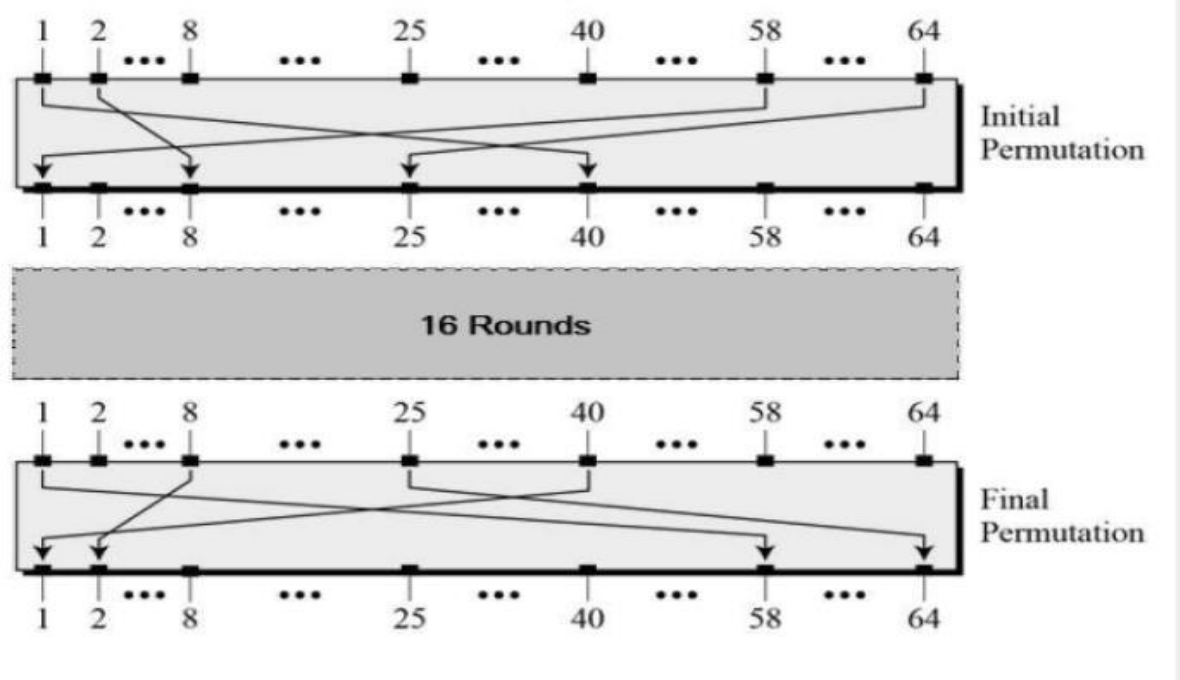


required to specify DES

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

Initial and Final Permutation (IP & FP)

- IP: first step of the data computation
- IP reorders the input data bits
- The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other.
- They have no cryptography significance in DES.



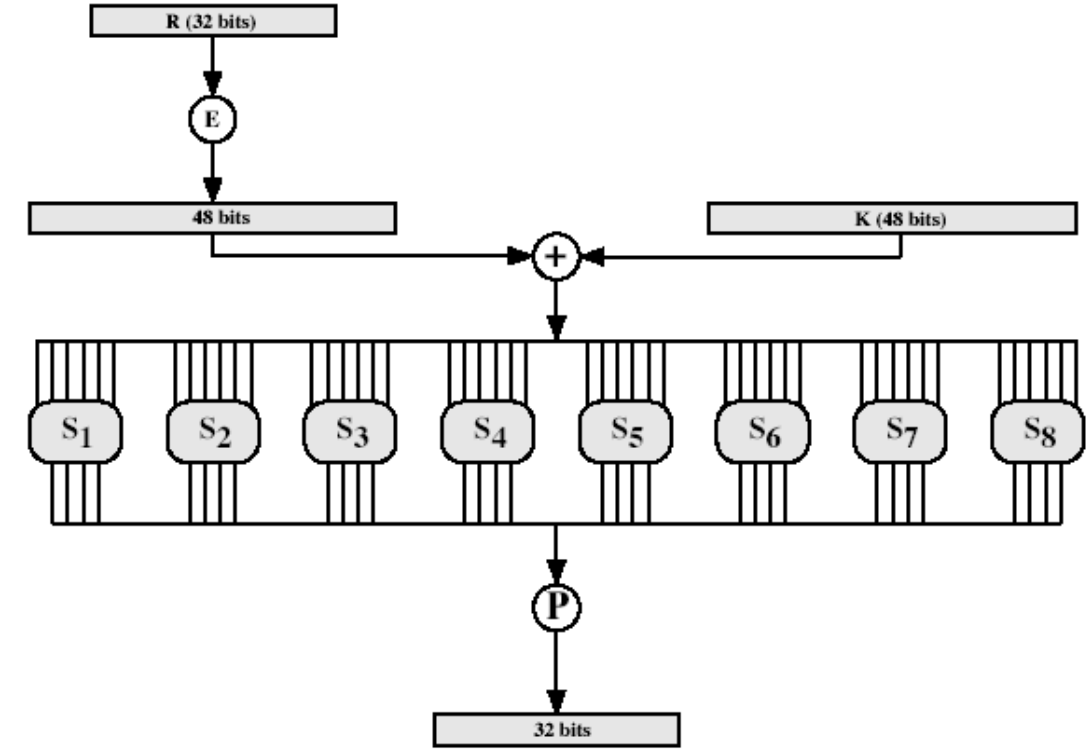
DES Round Structure

- Uses two 32-bit L & R halves
- As for any feistel cipher can describe as:

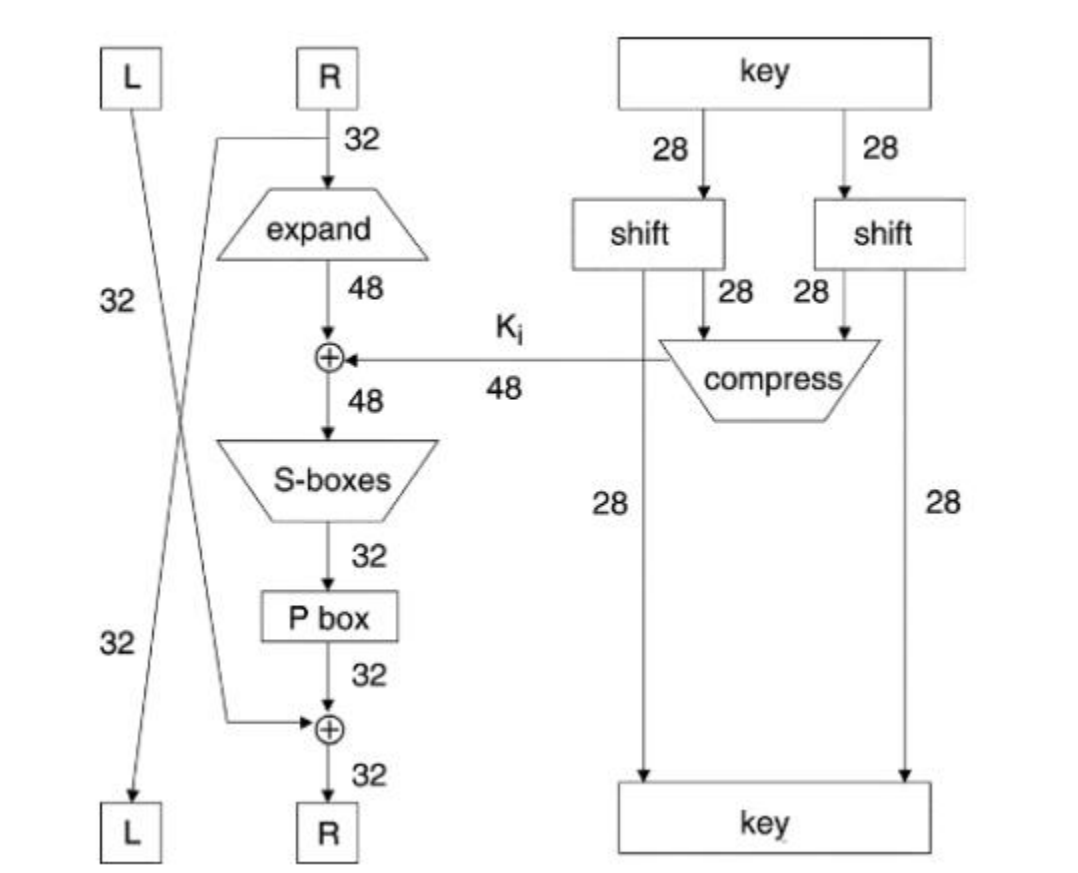
  $L_i = r_{i-1}$

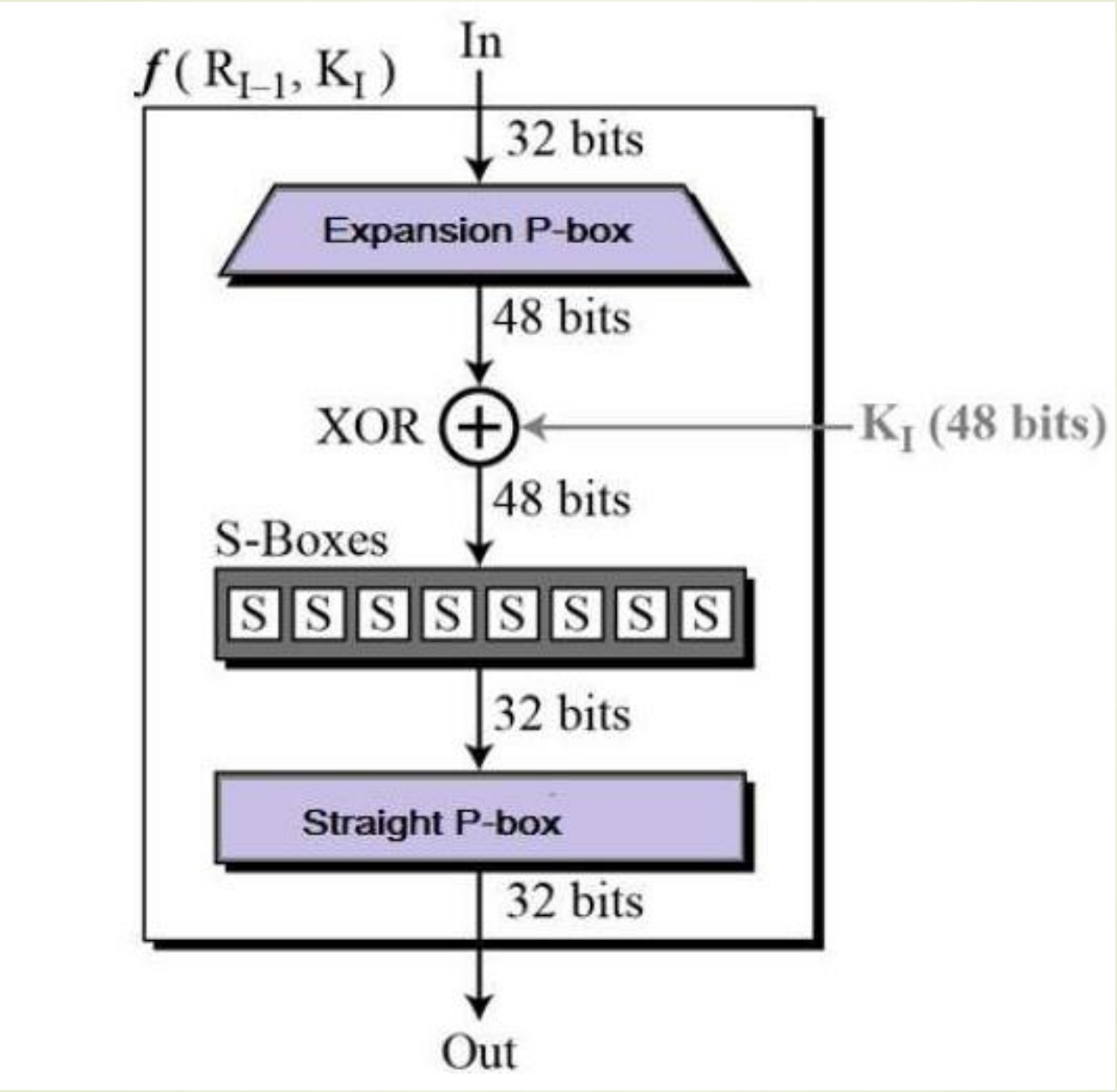
  $R_i = l_{i-1}$ xor f(r_{i-1}, k_i)
- Takes 32-bit R half and 48-bit subkey and:
  - Expands R to 48-bits using expansion permutation E (table 3.2 c.)
  - Adds to subkey
  - Passes through 8 s-boxes to get 32-bit result
  - Finally permutes this using 32-bit permutation function P (table 3.2 d)
  - s-boxes provide the "confusion" of data and key values whilst the permutation P then spreads this as widely as possible, so each S-box output affects as many S-box inputs in the next round as possible, giving "diffusion".

The round function F(R,K)



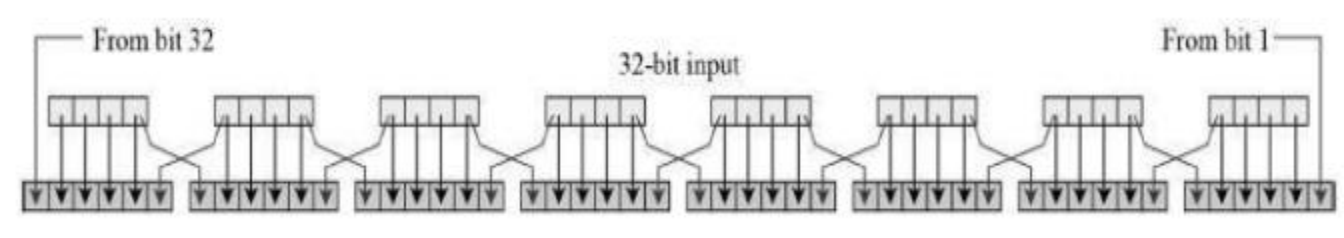
One round of DES




$f(R_{I-1}, K_I)$
In
32 bits
Expansion P-box
48 bits
XOR
K_I (48 bits)
48 bits
S-Boxes
S S S S S S S S
32 bits
Straight P-box
32 bits
Out

Expansion Permutation Box

- Expansion Permutation Box – Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits.



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

Substitution Boxes S

- 8 s-boxes (table 3.3)
- 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

	$b_1b_2b_3b_4$															
b_0b_5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
1	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
2	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
3	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D

DES S-box 1

Straight Permutation

- **Straight Permutation** – The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

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

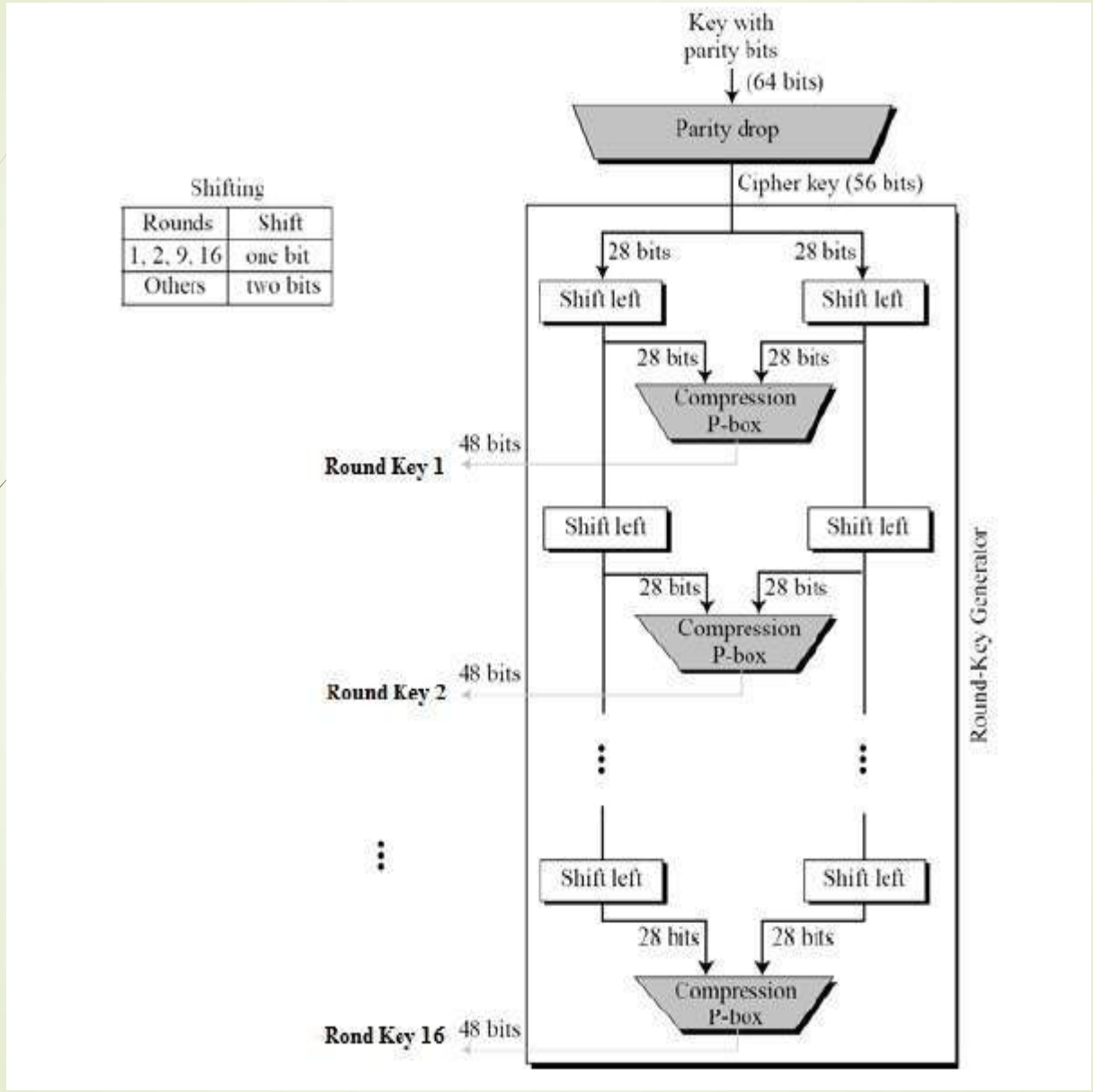
DES Key Schedule

Forms subkeys used in each round

- Initial permutation of the key PC1 (table 3.4b)
- Divide the 56-bits in two 28-bit halves
- At each round
  - Left shift each half (28bits) separately either 1 or 2 places based on the left shift schedule (table 3.4d)
    - Shifted values will be input for next round
  - Combine two halfs to 56 bits, permuting them by PC2 (table 3.4c) for use in function f
    - PC2 takes 56-bit input, outputs 48 bits

Shifting

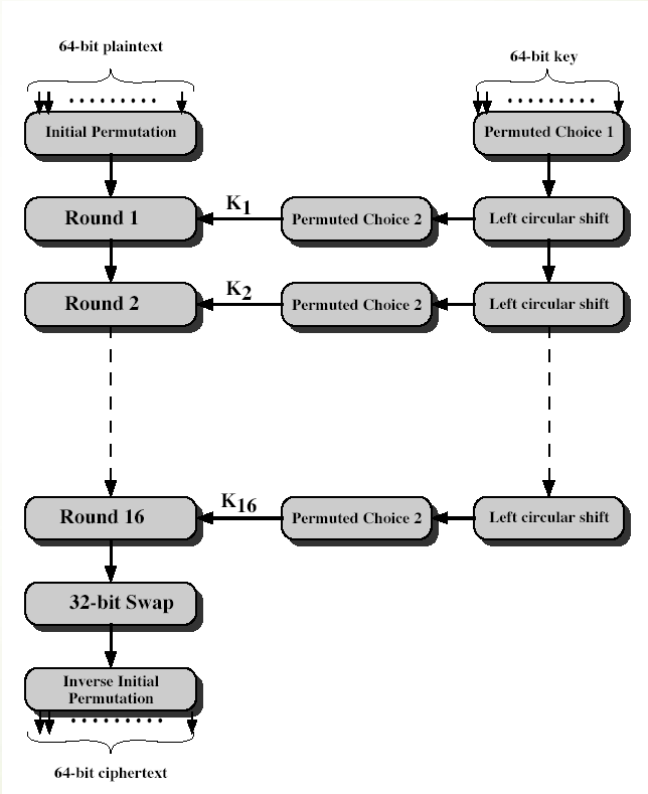
Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



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)



- These two properties make cipher very strong.
- Avalanche effect − A small change in plaintext results in the very great change in the ciphertext.
- Completeness − Each bit of ciphertext depends on many bits of plaintext.

Avalanche Effect

- key desirable property of encryption alg
- 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
- 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

Differential Cryptanalysis

- One of the most significant recent (public) advances in cryptanalysis
- Known in 70's with DES design
- Murphy, biham & shamir published 1990
- Powerful method to analyse block ciphers
- Used to analyse most current block ciphers with varying degrees of success
- DES reasonably resistant to it

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 values of the key
- Differential cryptanalysis compares two related pairs of encryptions

Differential Cryptanalysis Compares Pairs of Encryptions

- Differential cryptanalysis is complex
- with a known difference in the input
- searching for a known difference in output

$$\Delta m_{i+1} = m_{i+1} \oplus m'_{i+1}$$

$$= \left[m_{i-1} \oplus \mathrm{f}(m_i, K_i)\right] \oplus \left[m'_{i-1} \oplus \mathrm{f}(m'_i, K_i)\right]$$

$$= \Delta m_{i-1} \oplus \left[\mathrm{f}(m_i, K_i) \oplus \mathrm{f}(m'_i, K_i)\right]$$

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

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**
- Can then deduce keys values for the rounds
  - Right pairs suggest same key bits
  - Wrong pairs give random values
- Larger numbers of rounds makes it more difficult
- Attack on full DES requires an effort on the order of 2^{47}, requiring 2^{47} chosen plaintexts to be encrypted

Linear Cryptanalysis

- Another recent development
- Also a statistical method
- Based on finding linear approximations to model the transformation of DES
- Can attack DES with 2^{47} known plaintexts, still in practise infeasible

Presentation Topics

- DSE advantages & attacks : **Sultana Marjan, Sourav Barman**