Chapter 3 – Block Ciphers and the Data Encryption Standard

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
  - 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

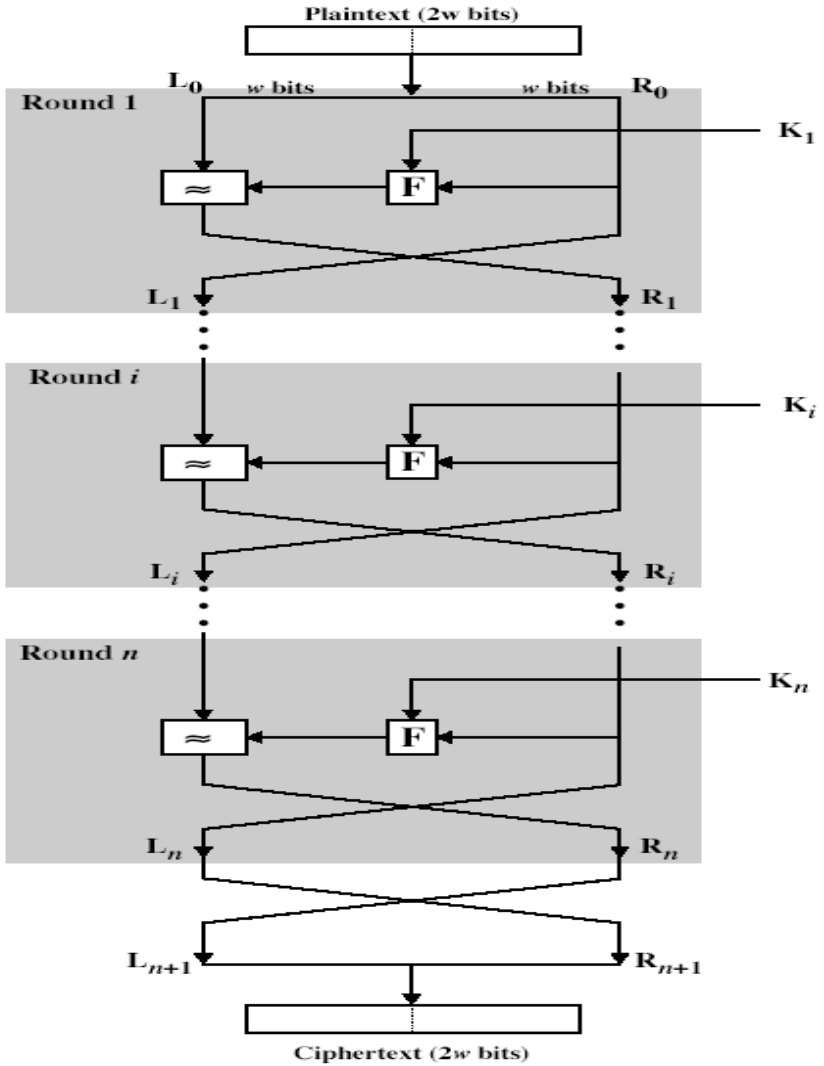
Diffusion and Confusion

- more practically Shannon suggested combining elements to obtain:
- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as possible

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



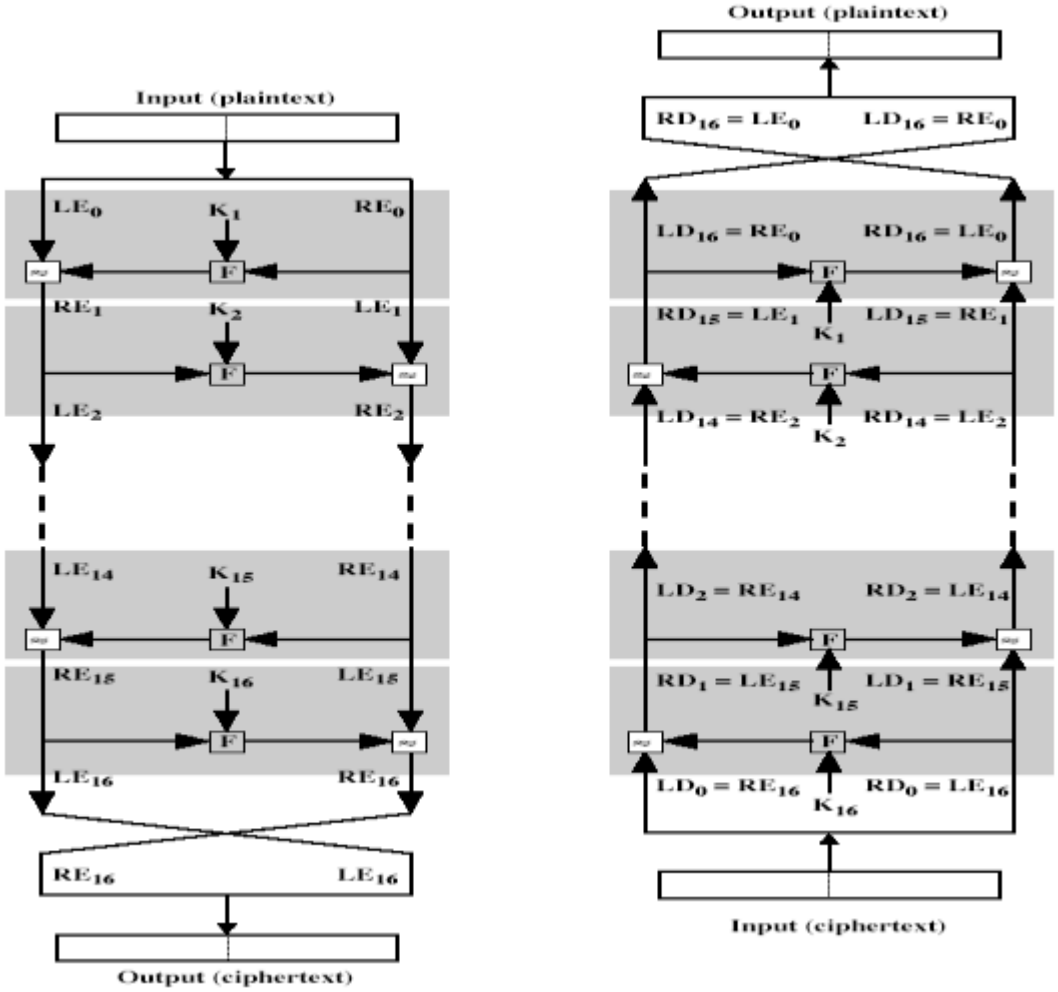
Feistel Cipher

- 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

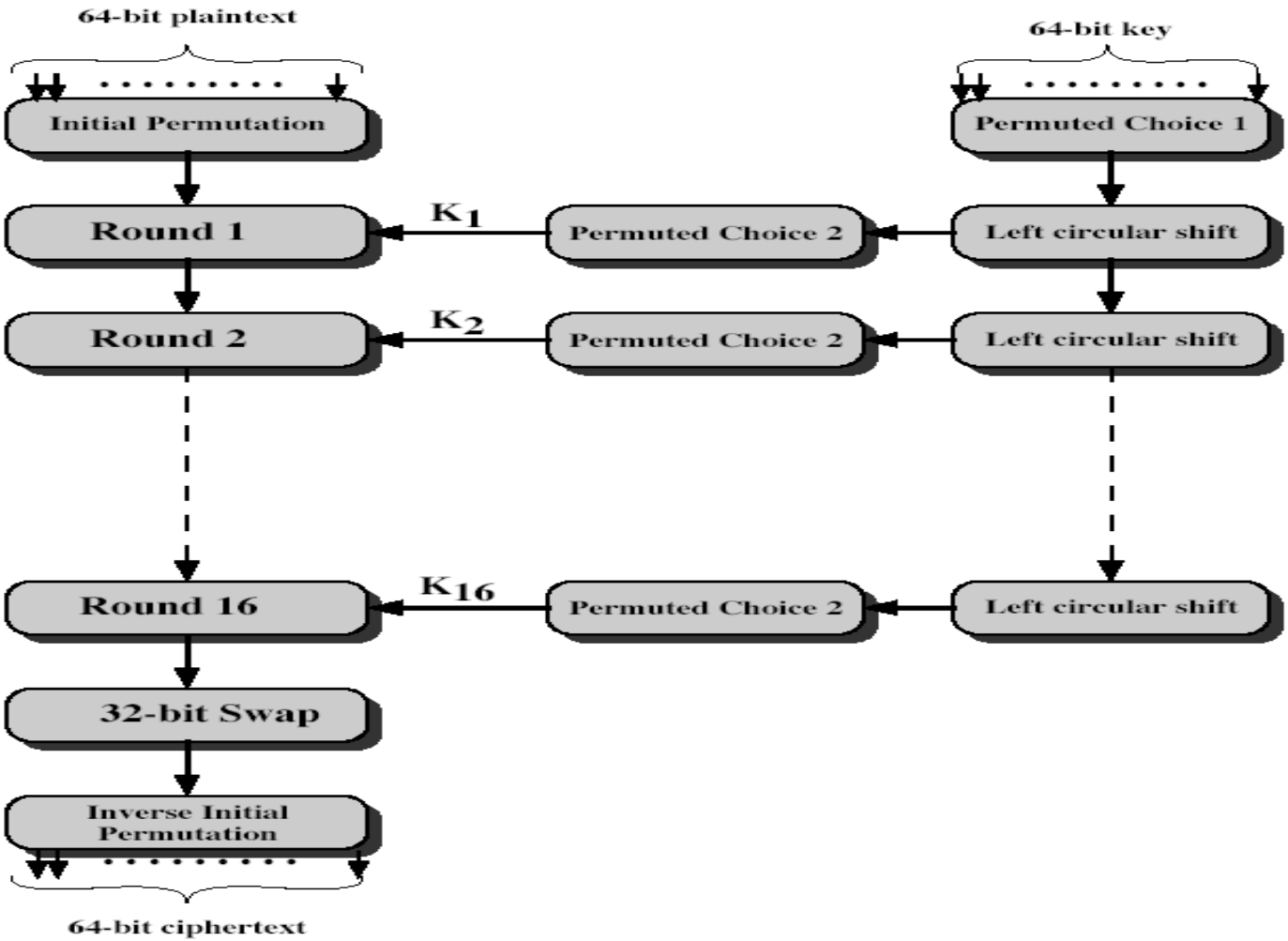
DES History

- 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
  - see text Table 3.2
- example:

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

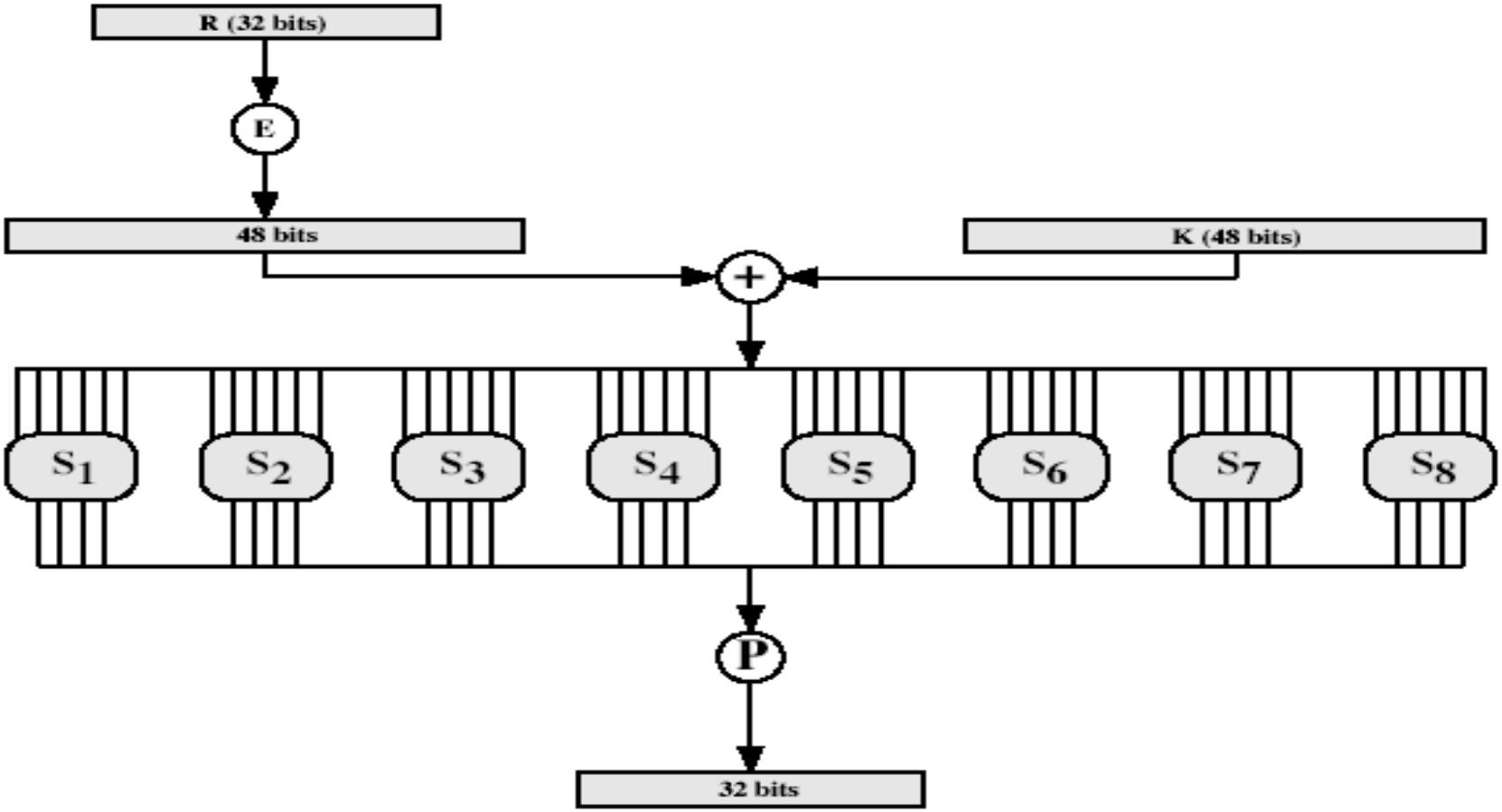
DES Round Structure

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

  $L_i = R_{i-1}$

  $R_i = L_{i-1}$ 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)

The round function F(R,K)



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

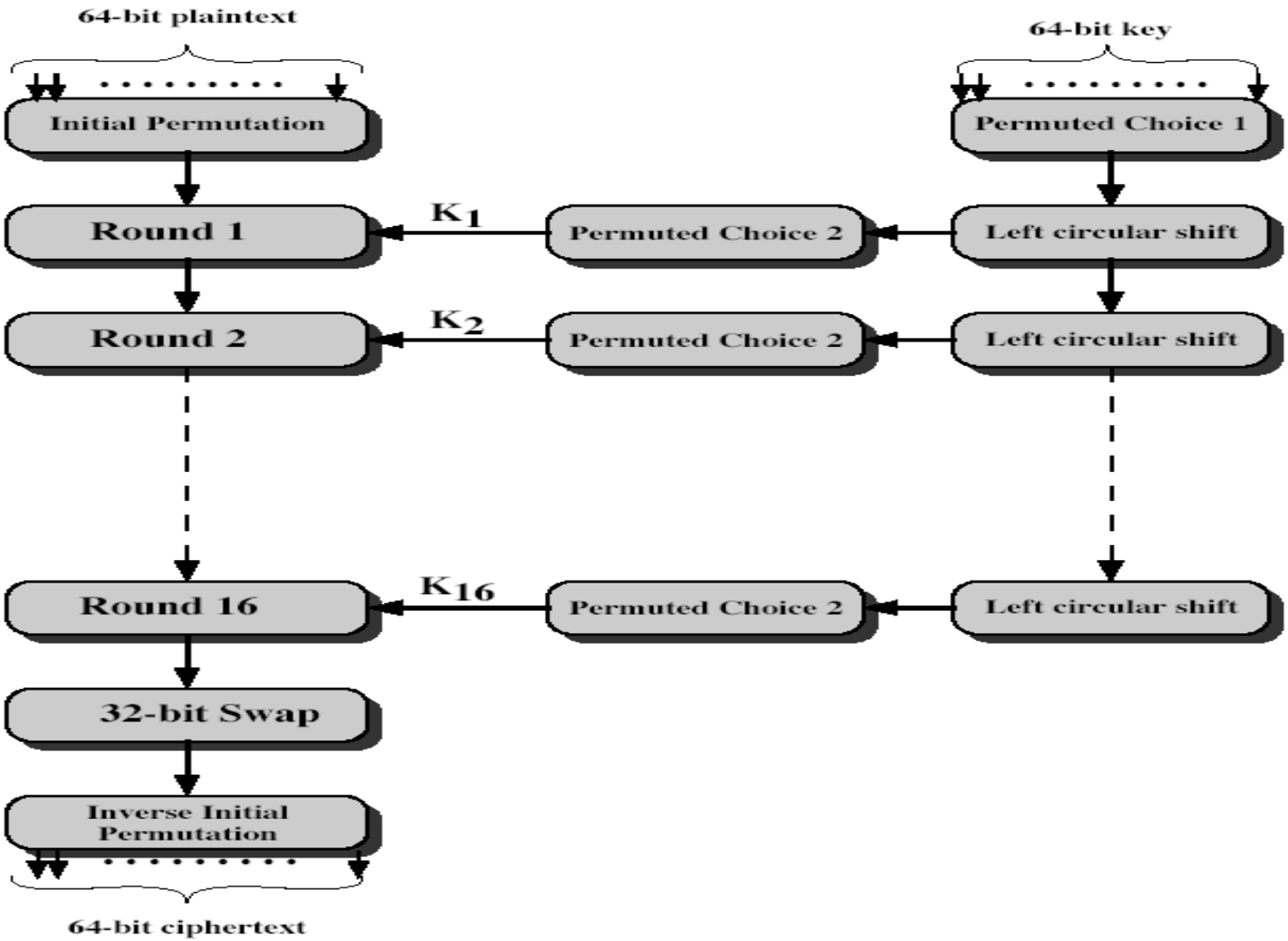
DES Key Schedule

- forms subkeys used in each round
- 1. initial permutation of the key PC1 (Table 3.4b)
- 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 (Table 3.4d)
    - Shifted values will be input for next round
  - 3.2. 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

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 alg
- DES exhibits strong avalanche
- where a change of **one** input or key bit results in changing approx **half** output bits

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

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

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

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

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