Hack It Out

Turing, TechTatva 2016

Basics

Ingredients

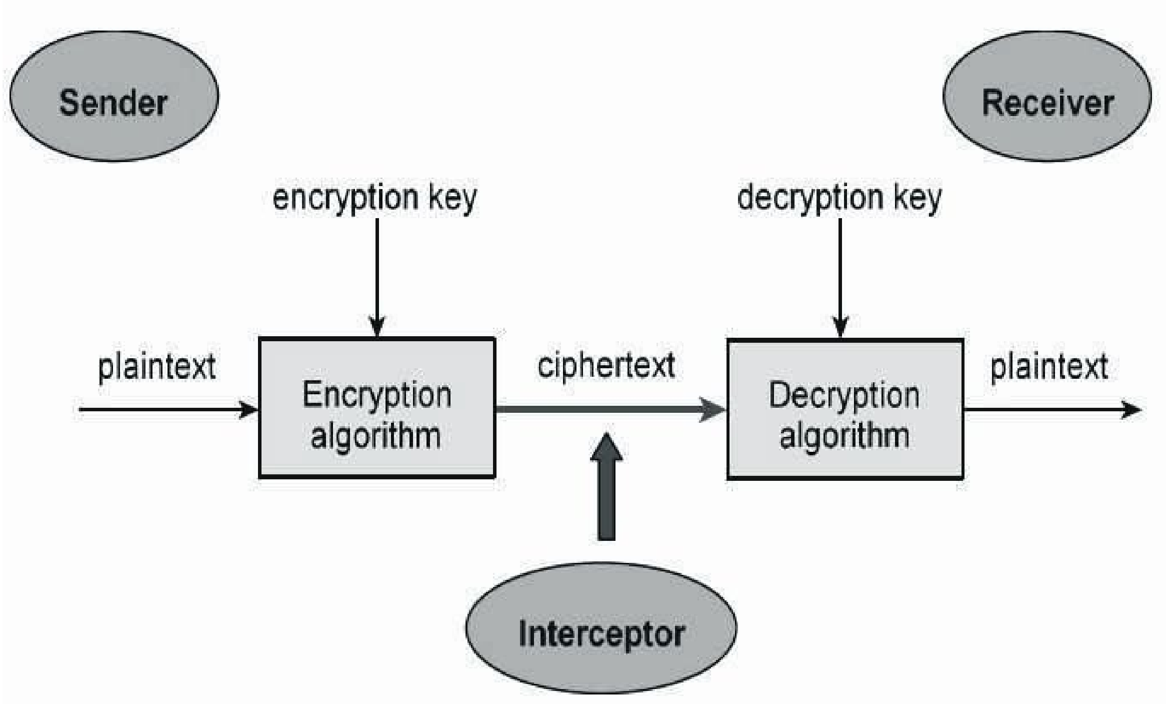
- **Plaintext:** The message
- **Encryption Algorithm:** Used to perform substitutions and manipulations on plaintext
- **Secret Key:** Input to EA. Algorithm produces different outputs based on key
- **Ciphertext:** The encrypted message as output by EA
-
- **Decryption Algorithm:** Reverse of EA

Requirements for Encryption Algorithm

EA should be such that attacker who knows the algorithm and has access to few ciphertexts should not be able to decipher the ciphertexts or guess the key

Sender and receiver must have copies of secret key in a secure fashion and must keep this key secure

Cryptosystem



Dimensions

Systems characterized along 3 independent dimensions:

- **Types of Operations:** Substitution and Transposition
- **Number of Keys:** One or more
- **Processing:** Block Cipher or Stream Cipher

Cryptanalysis and Brute Force

Cryptanalysis: Attack based on nature of algorithm or knowledge of general characteristics of plaintext or plaintext-ciphertext pairs.

Brute Force: The attacker tries every possible key on a piece of ciphertext

Encryption schemes should be:

Unconditionally Secure
Computationally Secure

Types of cryptanalysis attacks

- **ciphertext only**
  - only know ciphertext / algorithm, statistical, can identify plaintext
- **known plaintext**
  - attack cipher by knowing / suspecting plaintext & ciphertext
- **chosen plaintext**
  - attack cipher by selecting plaintext and obtaining ciphertext, useful if limited set of messages
- **chosen ciphertext**
  - attack cipher by selecting ciphertext and obtaining plaintext
- **chosen text**
  - attack cipher by selecting either plaintext or ciphertext to en/decrypt

Ciphers

What’s a cipher?

A cipher is a secret or disguised way of writing a message so as to hide it from prying eyes.

Examples:

- Caesar Cipher
- Hill Cipher
- Vigenere Cipher
- Playfair Cipher
- One Time Pad
- Rail Fence Cipher...etc

Block Ciphers

Divides the message into blocks of fixed size and encrypt one block at a time

Example

DES
AES

Stream Ciphers

Encrypts messages as a stream of characters, one character at a time.

Example

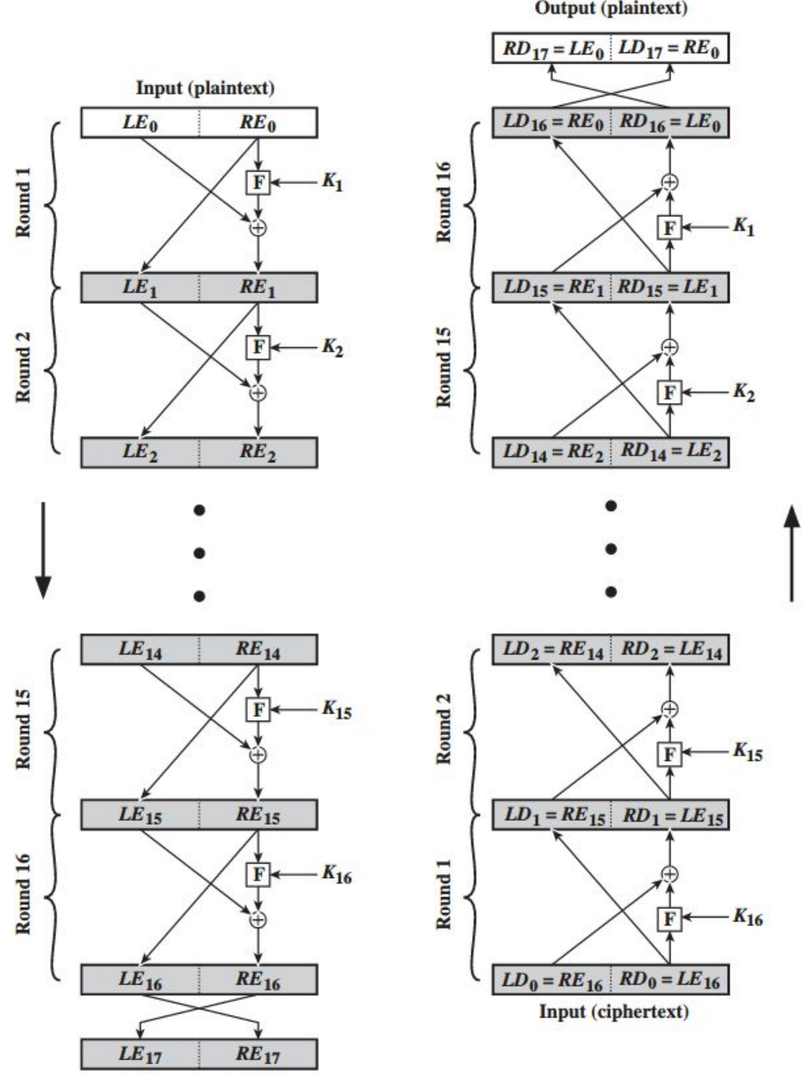
RC4
Salsa20

Feistel Cipher

Not a type of cipher, rather a model from which many block ciphers have been derived.

- The input block to each round is divided into two halves that can be denoted as L and R for the left half and the right half.
- In each round, the right half of the block, R, goes through unchanged. But the left half, L, goes through an operation that depends on R and the encryption key. First, we apply an encrypting function ‘f’ that takes two input − the key K and R. The function produces the output f(R,K). Then, we XOR the output of the mathematical function with L.

- The permutation step at the end of each round swaps the modified L and unmodified R. Therefore, the L for the next round would be R of the current round. And R for the next round be the output L of the current round.
- Above substitution and permutation steps form a ‘round’. The number of rounds are specified by the algorithm design.
- Once the last round is completed then the two sub blocks, ‘R’ and ‘L’ are concatenated in this order to form the ciphertext block.


Output (plaintext)
$RD_{17} = LE_0$
$LD_{17} = RE_0$
Input (plaintext)
LE_0
RE_0
Round 1
F
K_1
LE_1
RE_1
Round 2
F
K_2
LE_2
RE_2
LE_{14}
RE_{14}
Round 15
F
K_{15}
LE_{15}
RE_{15}
Round 16
F
K_{16}
LE_{16}
RE_{16}
LE_{17}
RE_{17}
$LD_{16} = RE_0$
$RD_{16} = LE_0$
Round 16
$LD_{15} = RE_1$
$RD_{15} = LE_1$
Round 15
$LD_{14} = RE_2$
$RD_{14} = LE_2$
$LD_2 = RE_{14}$
$RD_2 = LE_{14}$
Round 2
$LD_1 = RE_{15}$
$RD_1 = LE_{15}$
Round 1
$LD_0 = RE_{16}$
$RD_0 = LE_{16}$
Input (ciphertext)

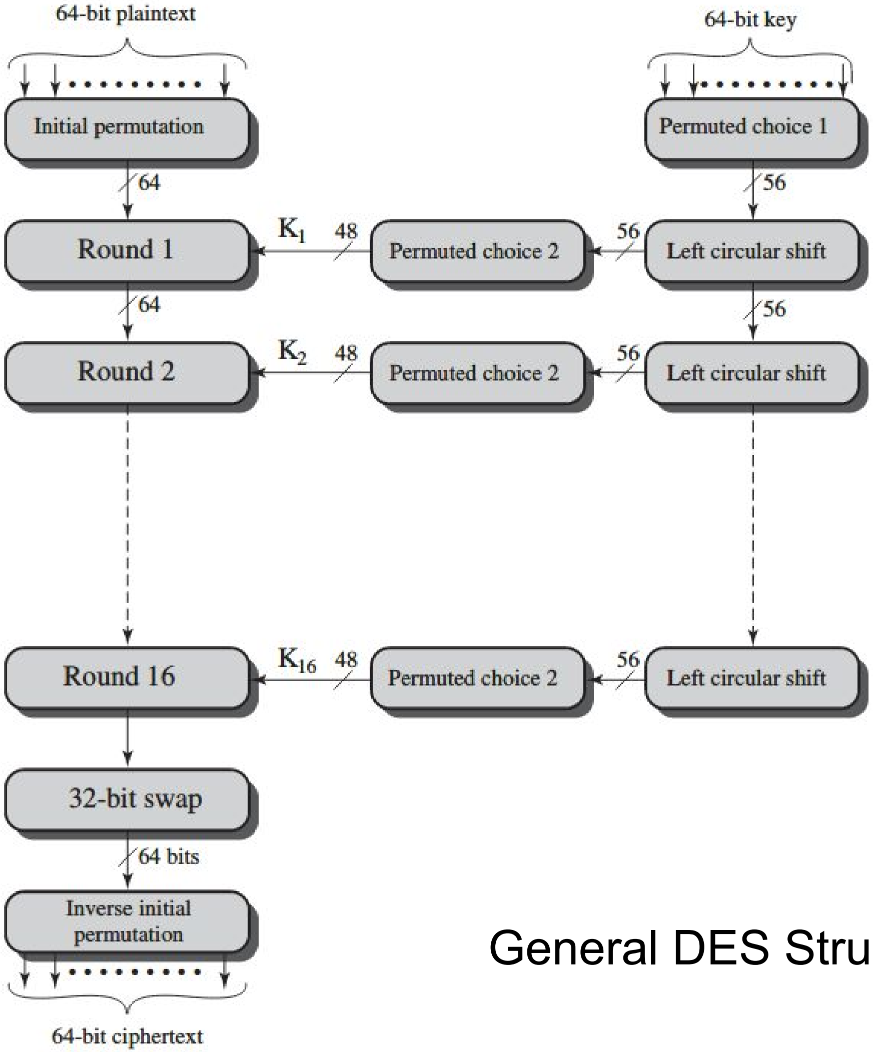
Data Encryption Standard

Confusion

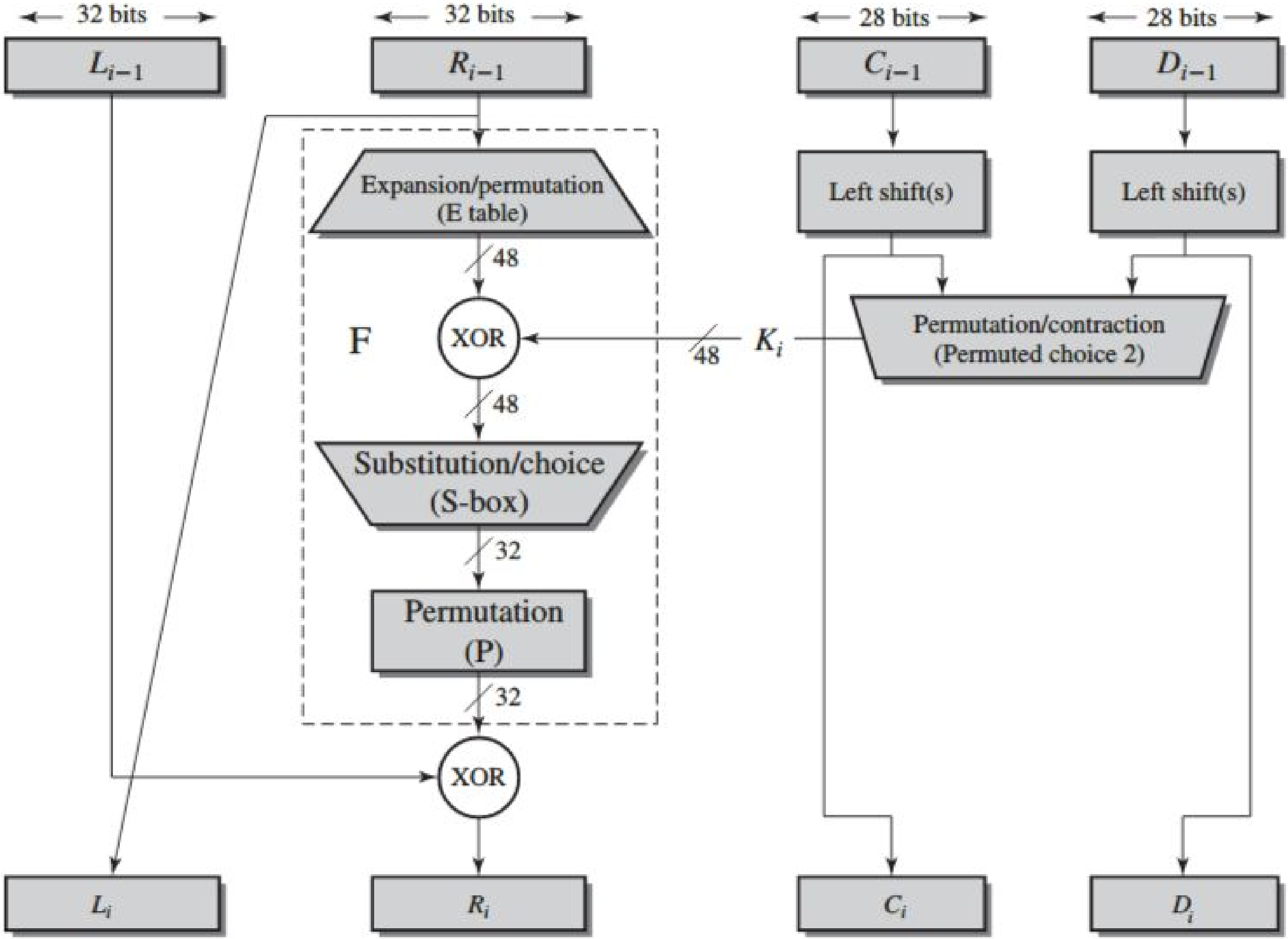
Confusion means that each binary digit (bit) of the ciphertext should depend on several parts of the key.

Diffusion

Diffusion means that if we change a single bit of the plaintext, then many of the bits in the ciphertext should change, and vice versa

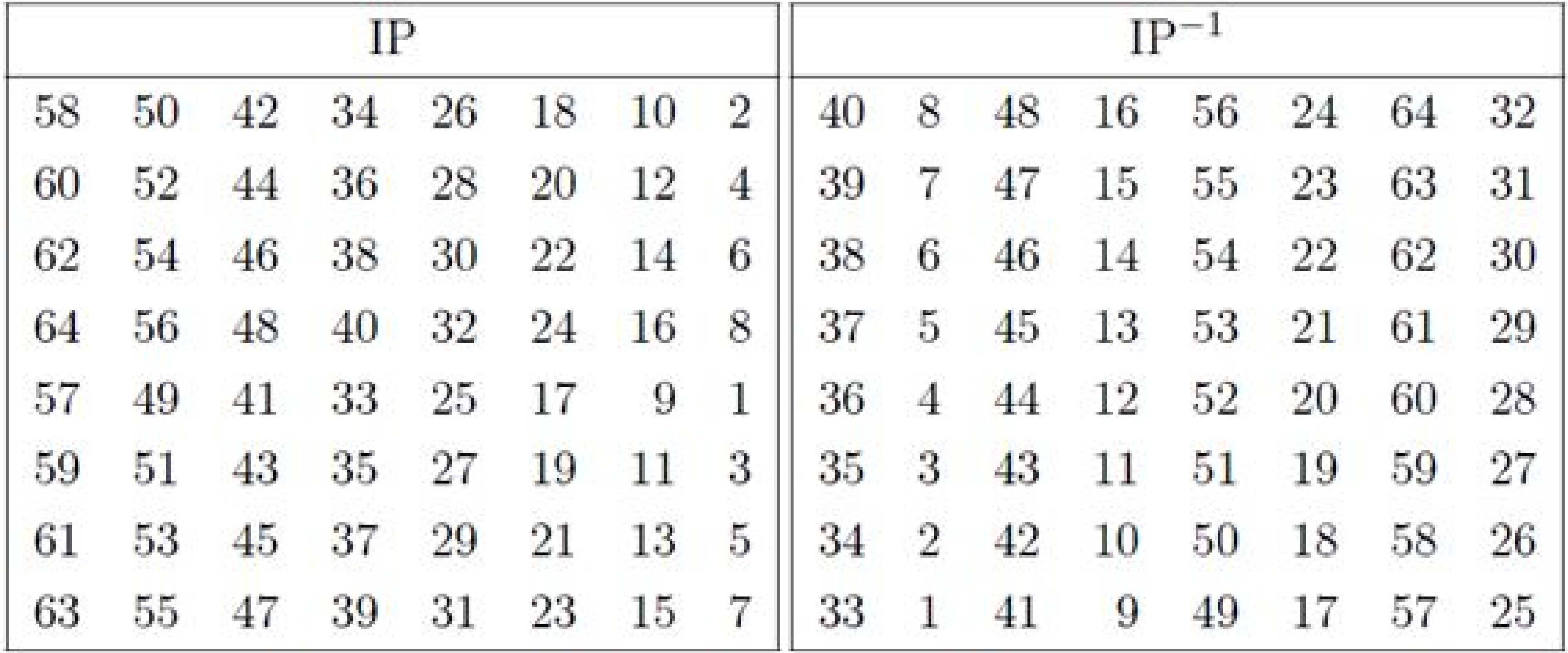


General DES Structure



Single Round of DES

Initial Permutation

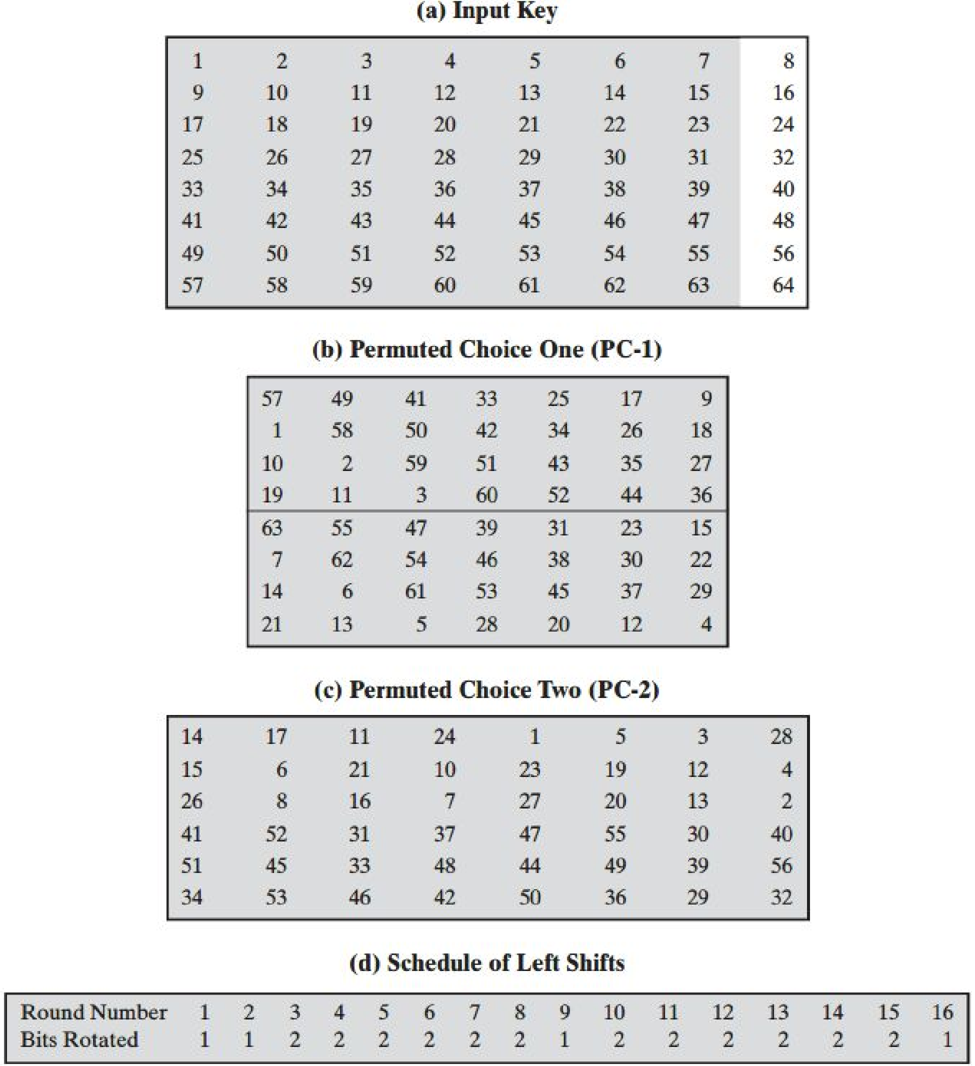


IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

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

Key Generation

- Key length = 64 bits
- Every 8th bit is ignored, hence key size = 56 bits
- 56 bit key subjected to permutation (Permuted Choice 1)
- 56 bit divided into 2 28 bit halves
- Each half subjected to circular left shift individually. Number of shifts varies by round of encryption
- The halves serve as input to Permuted Choice 2, where 48 bit is generated



(a) Input Key

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(b) 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

(c) 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

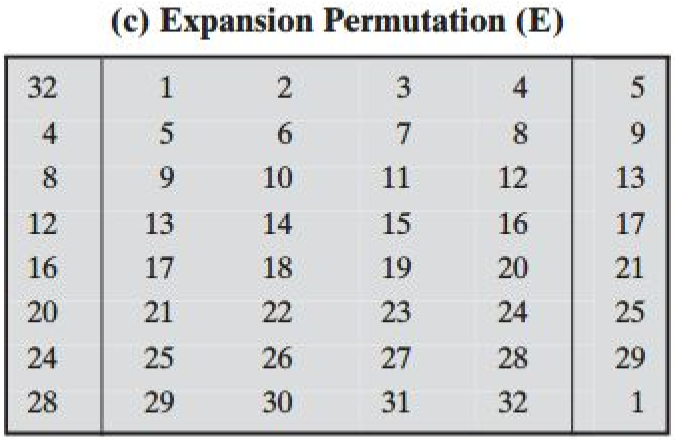
(d) Schedule of Left Shifts

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

Expansion Permutation

The 32 bit right half is permuted and expanded to 48 bits, i.e 16 bits of the 32 bit right half are duplicated

The output of the expansion permutation is XORed with the 48 bit key obtained and passed as an input to the Substitution/Choice (S-Box)



(c) 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

S Boxes

Accepts 6 bits as input and returns 4 bits as output.

The first and last of the 6 bit input is used to select the row and the middle four bits are used to select the column

8 S-Boxes for 48 bits

First 6 bits send to S-Box 1, next 6 to S-Box 2 and so on

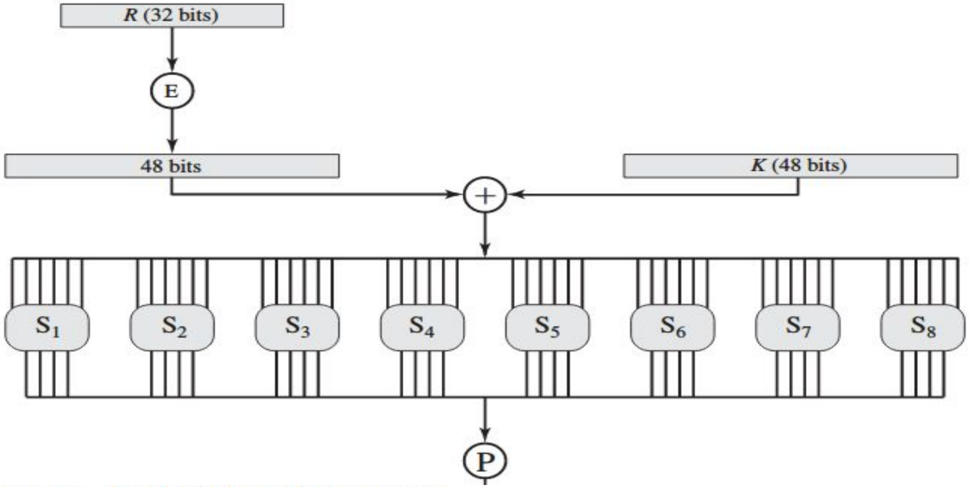
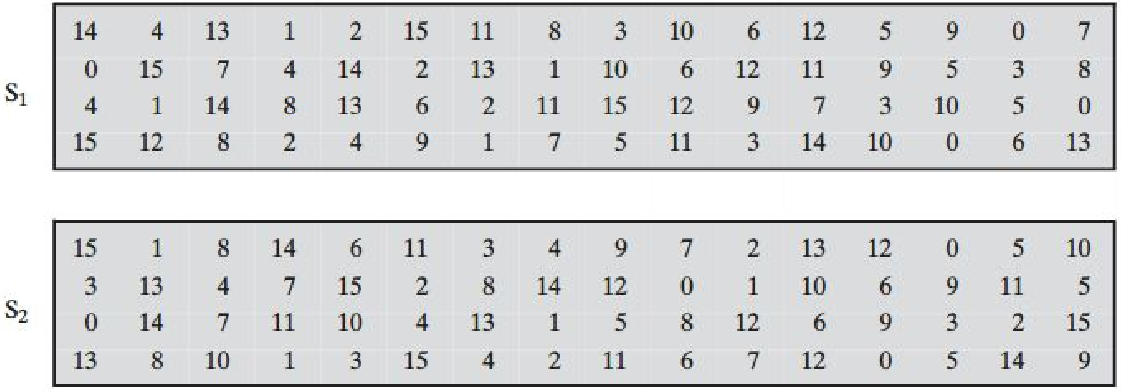


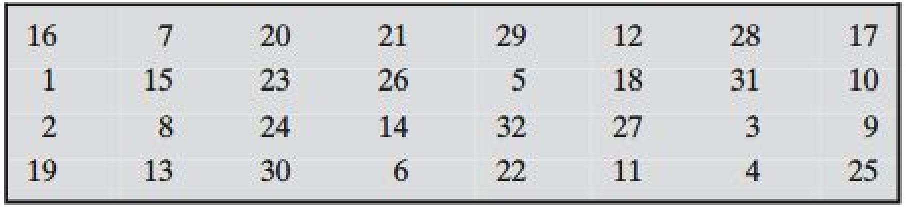
Table 3.3 Definition of DES S-Boxes



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

Permutation

32 bit output of S Boxes is permuted and XORed with 32 bits of the left half. This now becomes the right half for the next round of DES.



(d) 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

Final Operations

After the 16th round, the 32 bit swap functions swaps the left and right halves of the output to produce the preoutput

The 64 bit result is then permuted to produce the 64 bit ciphertext. This permutation is the inverse of the Initial Permutation

DES Decryption

Des decryption follows the same steps as the encryption, except the subkeys are applied in reverse, i.e. SubKey 16 is applied first and SubKey 1 is applied last.

Strength of DES

DES has 2^{56} possible keys, i.e. 7.2 x 10^{16} keys, hence brute-force seems impractical. However, a rate of 10^{9} decryptions/sec is reasonable for today’s multicore computers. At this rate it is possible to break a DES encryption in 1.125 years. Contemporary supercomputers can perform 10^{13} decryptions/sec which can break a DES encryption in 1 hour.

Another concern is that the design criteria for the 8 S-Boxes were not made public, hence leading to the speculation that cryptanalysis is possible for an opponent who knows the weakness of the S-Boxes.

DES Attack Examples

The DESCHALL project was the first to break a message encrypted by DES in 1997

EFF’s Deep Crack broke a DES key in 56 hours in 1998

Together with distributed.net (A worlwide distributed computing effort), Deep Crack broke a DES key in 22 hours and 15 minutes in 1999

COPACOBANA (Cost Oprimized Parallel Code Breaker) developed in 2006 in under $10,000 which could break DES keys in 6.4 days on an average.

COPACOBANA RIVYERA developed in 2008 can break DES in less than a day