D.E.S

Data Encryption Standard

(i) block cipher
(ii) symmetric cipher (same key for encryption + decryption)
(iii) 64 bit plain text block.
It encrypts the data in blocks of size 64 bit each
(iv) 16 rounds. each round is a fiestel round.

Steps

(i) Initial permutation
(ii) 16 fiestel rounds
(iii) swapping / left right swap
(iv) Final permutation / Inverse initial permutation

Basic structure

64 bit plain text

↓

Initial permutation

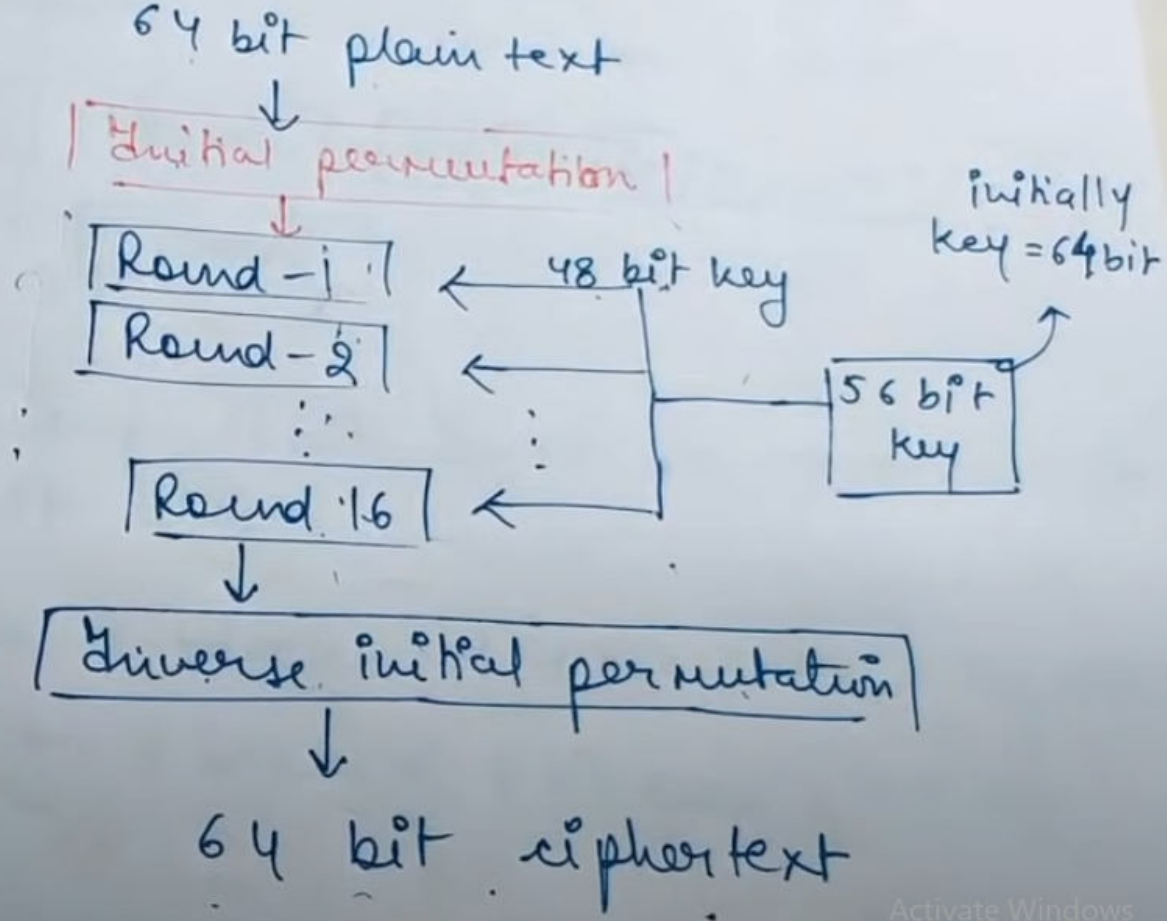
Round - 1

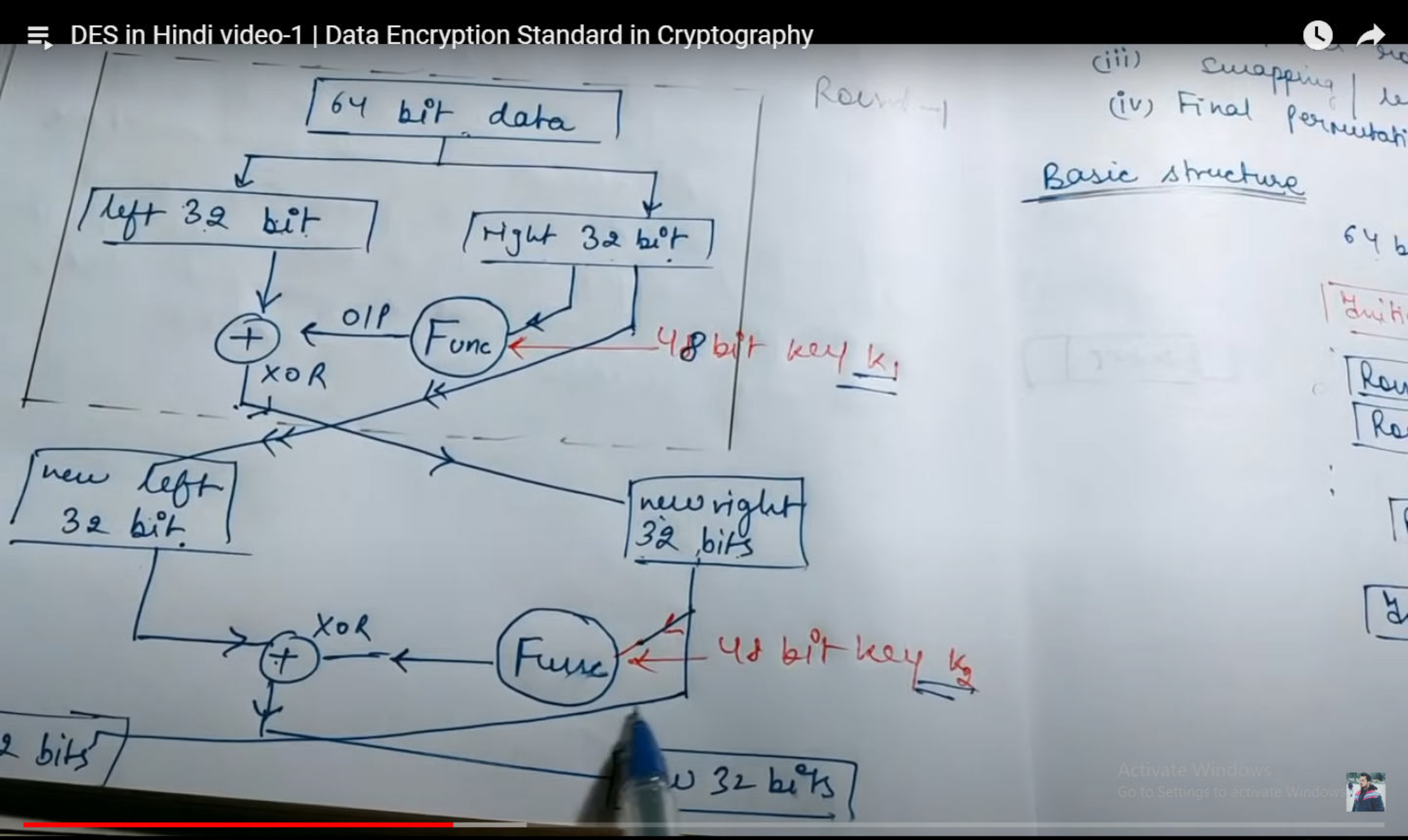
Round – 1

(iv) Final permutation | Inverse initial permutation

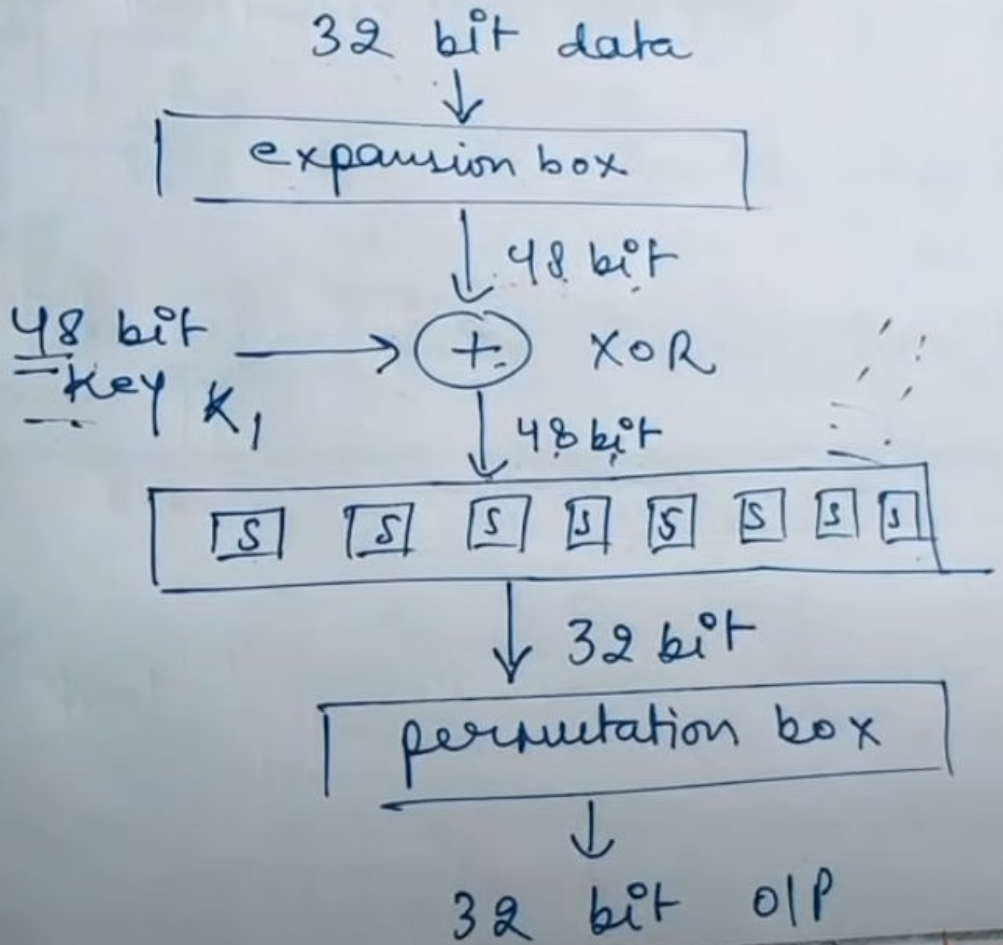
Swap / left right swap

Basic structure

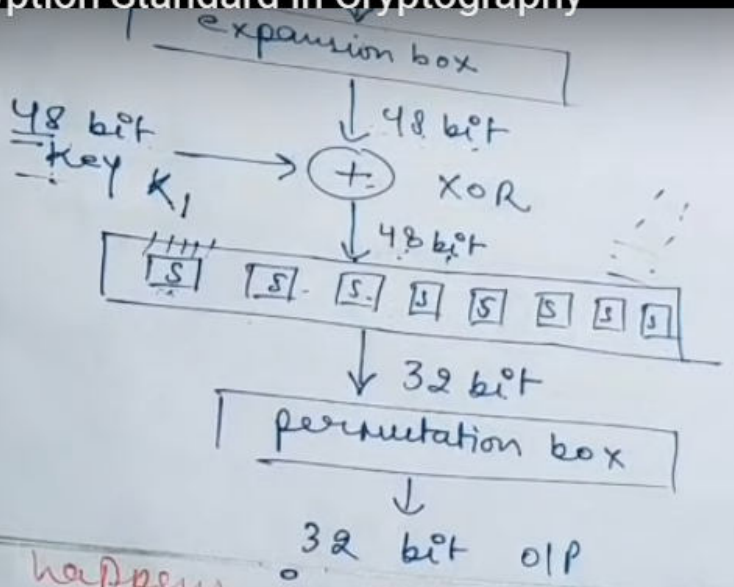



64 bit data
left 32 bit
Right 32 bit
O/P
Func
48 bit key K1
XOR
new left 32 bit
new right 32 bits
XOR
Func
48 bit key K2
Round-1
(iii) Swapping
(iv) Final permutati
Basic structure
64 b
Activate Windows
Go to Settings to activate Windows.

Function definition



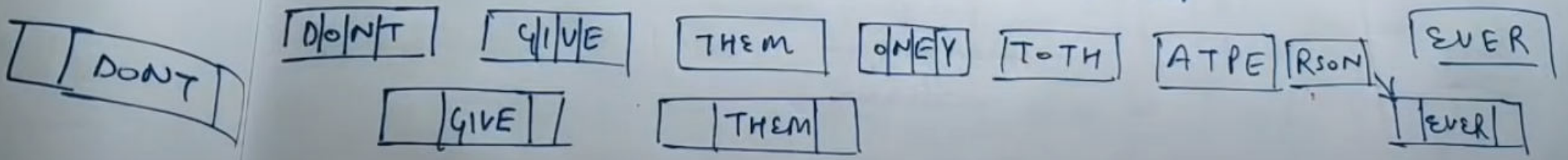
What happens in expansion box?



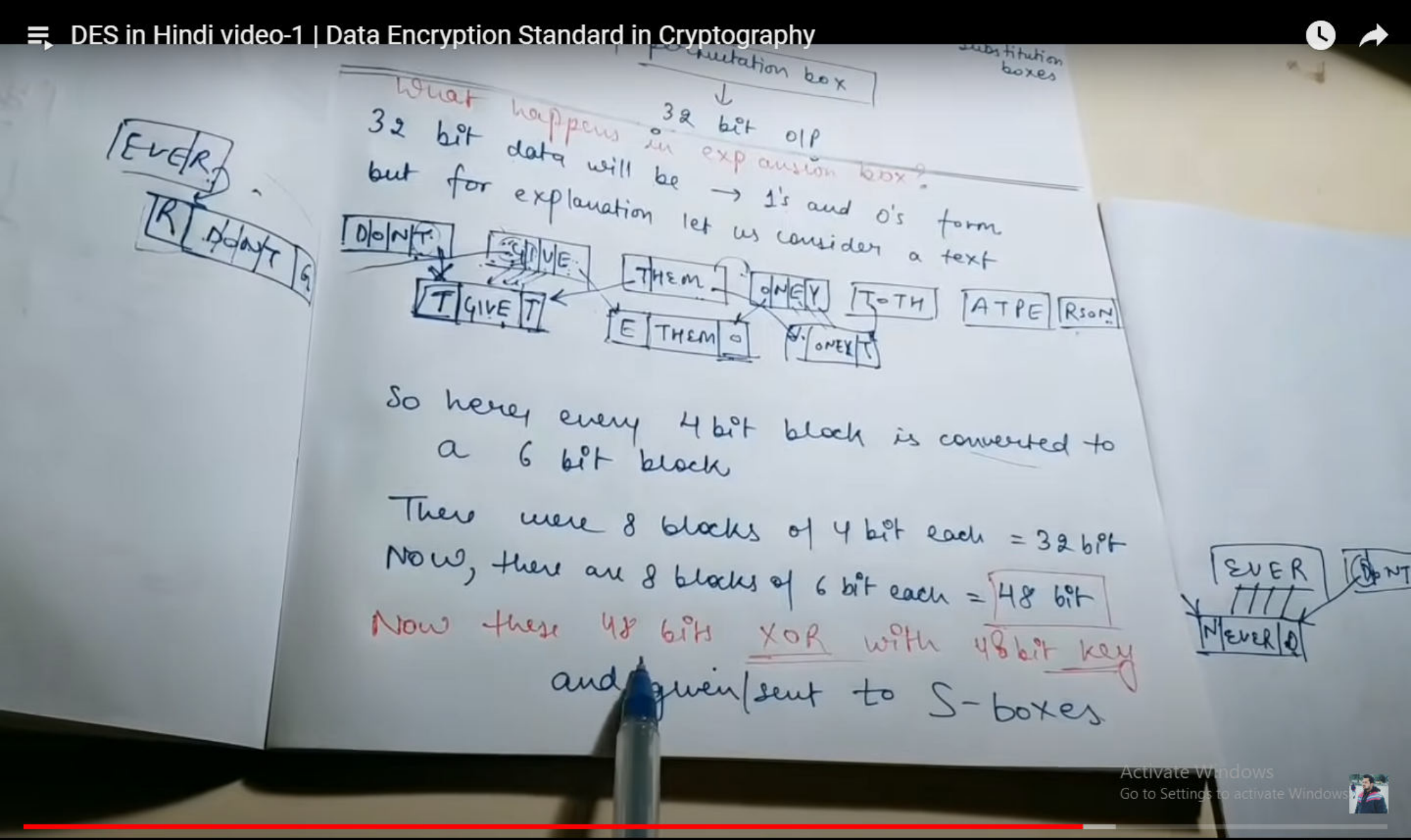
What happens in expansion box?

32 bit data will be → 1's and 0's form

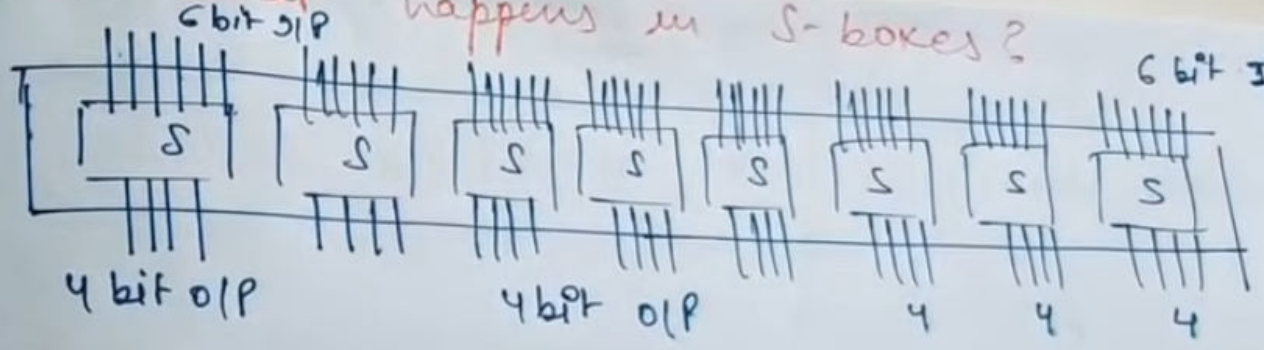
but for explanation let us consider a text



So here every 4 bit block is converted to a 6 bit block



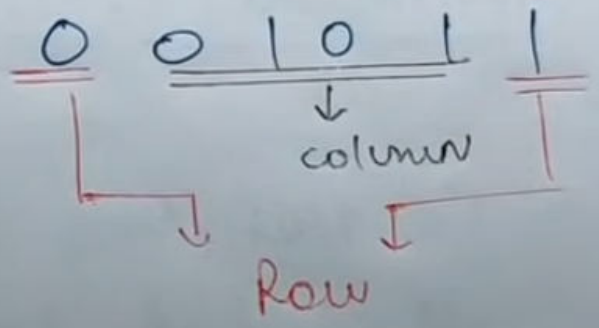
Ques What happens in S-boxes?



o/p → 4×8 = 32 bits These 32 bits will go into permutation box.

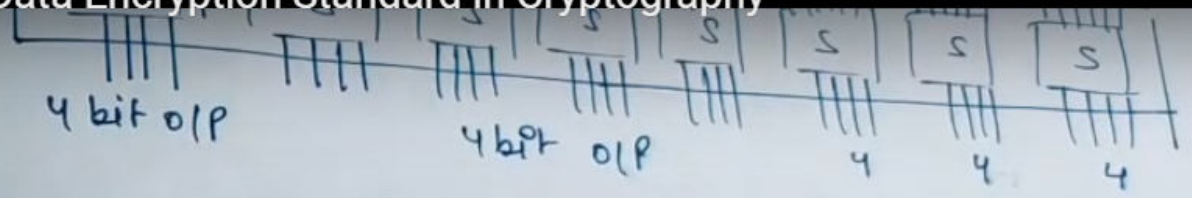
how 6 bit converted to 4 bit?

→ eg



S,box - table 1

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



o/p → 4×8 = 32 bits These 32 bits will go into permutation box.

how 6 bit converted to 4 bit ?

→ eg 0 0 1 0 1 1

o/ → 1

o1o/ →

column

Row

S,box - table 1

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

numbers will be filled.

will have a dif table.

How 16 subkeys are generated?

→ actually, we have 64 bit key which go as a i/p to PC-1 (permuted choice-1) and we get o/p as 56 bit key.

Inside PC-1 (permuted choice-1)

64 bit key divided into → 8 parts each of 8bit

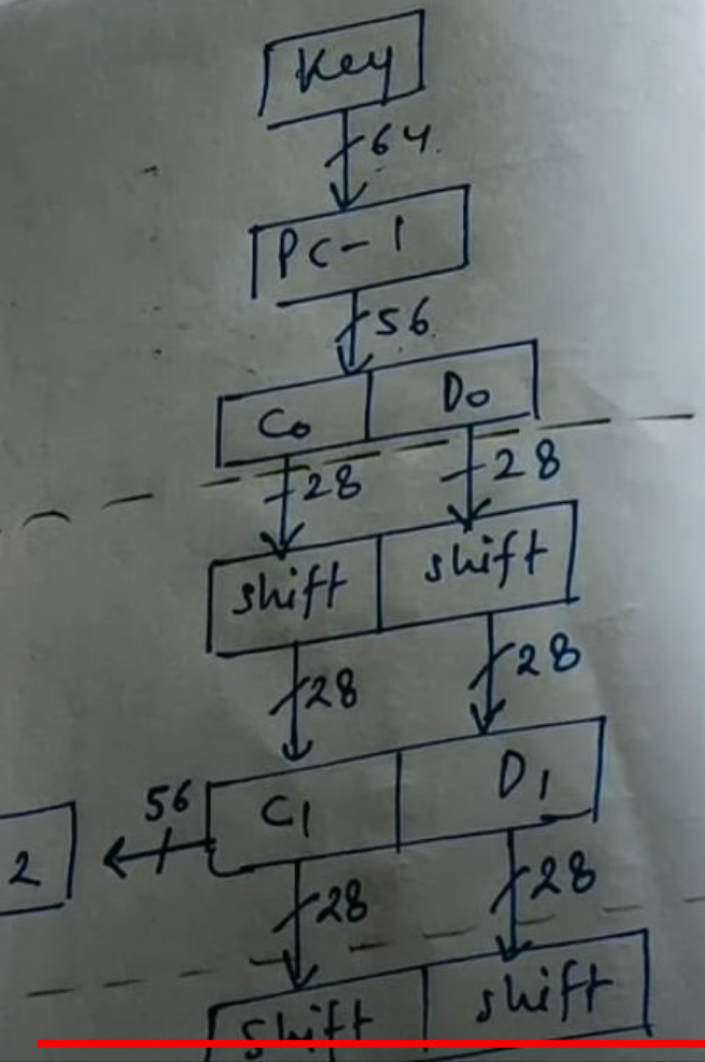
8 * 8 = 64 bit

1 2 3 4 5 6 7 8	9,10,11,12,13,14,15 16	...	58,59...64
1st part	2nd part/block		8th part

From each part, last bit → discarded

ie bit → 8, 16, 24, 32, ... 64 discarded.

Hence, ... have 8 parts of 7 bits each



How 16 subkeys are generated ?

→ actually, we have (64) bit key which go as a i/p to PC-1 (permuted choice-1) and we get o/p as 56 bit key.

Inside PC-1 (permuted choice-1)

64 bit key divided into → 8 parts each of 8bit

8 * 8 = 64 bit

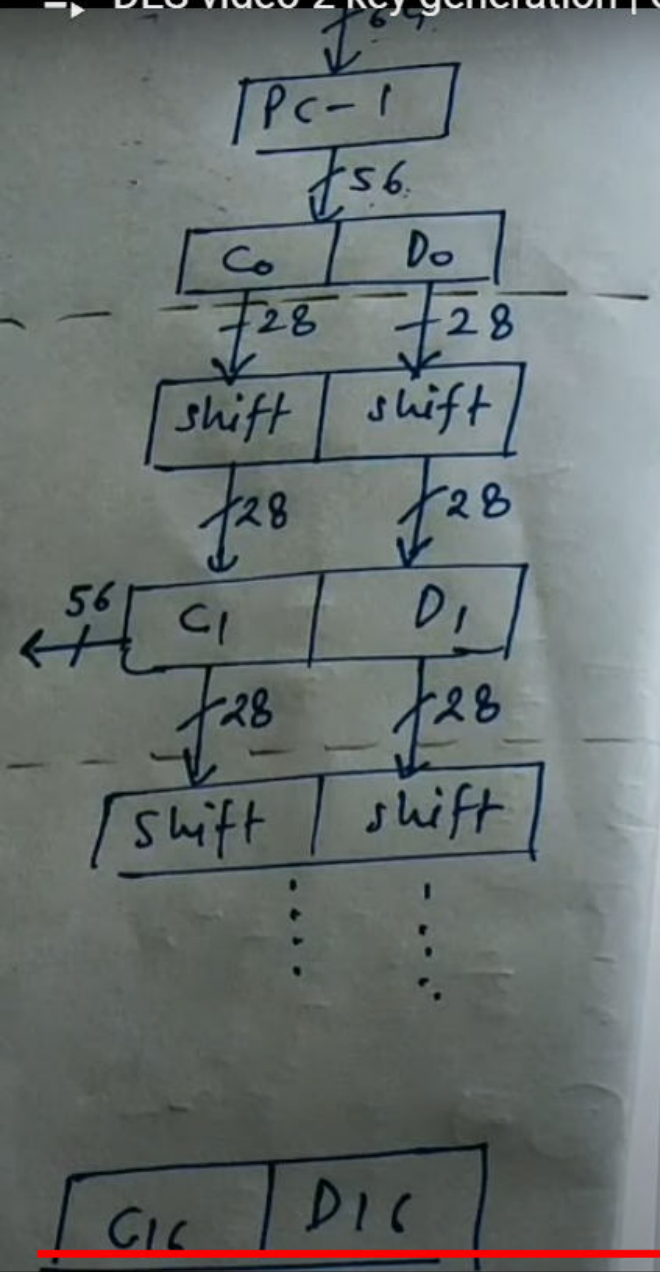
1 2 3 4 5 6 7 (8)	9,10,11,12,13,14,15,(16)	...	58,59...(64)
1st part	2nd part/block		8th part

From each part, last bit → discarded

ie bit → 8, 16, 24, 32, ... 64 discarded.

Hence, we have 8 parts of 7 bits each

= 8 * 7 = 56 bit



1st part

9,10,11,12,13,14,15,16

2nd part/block

8 * 8 = 64 bit

58,59...64

8th part

From each part, last bit → discarded

i.e bit → 8, 16, 24, 32, ... 64 discarded.

Hence, we have 8 parts of 7 bits each

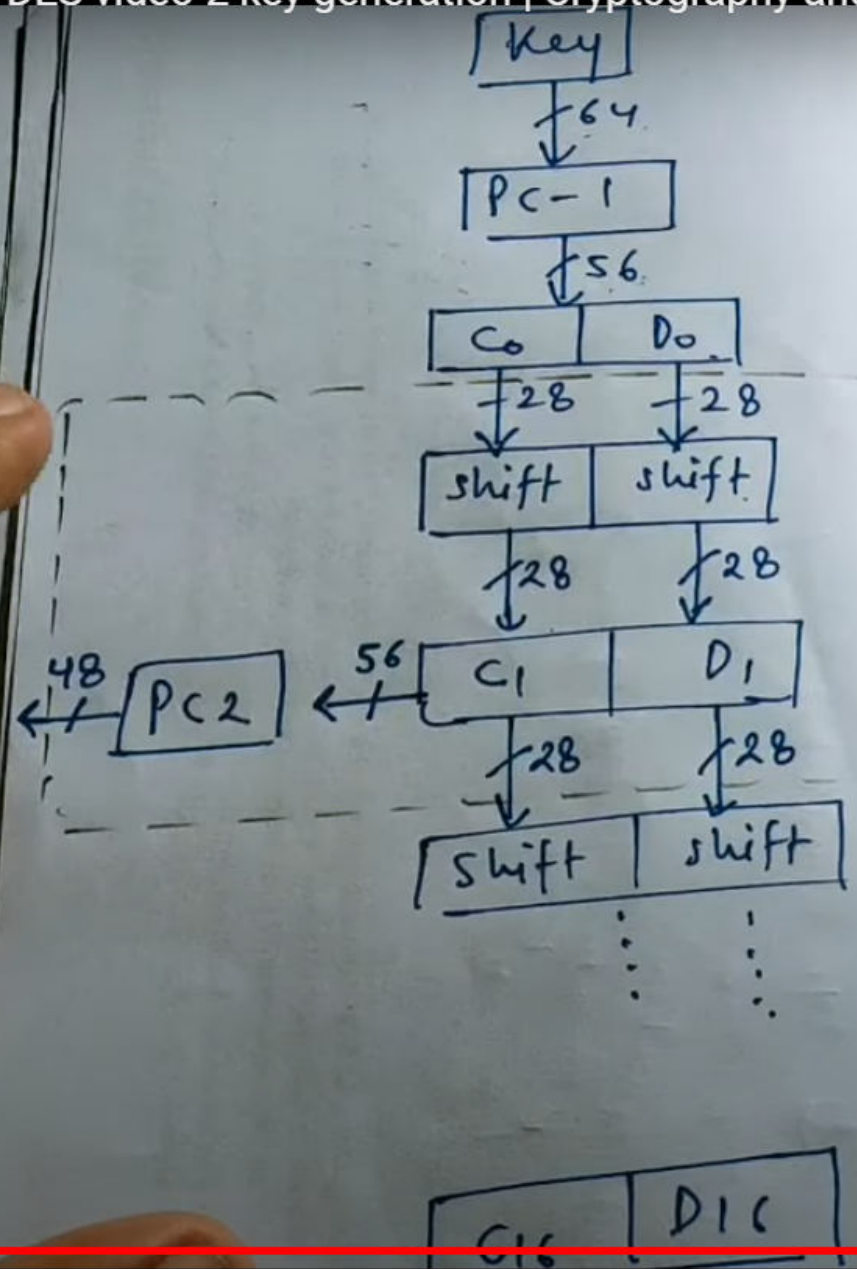
= 8 * 7 = 56 bits

→ O/P of PC-1 is 56 bits which is then divided into 2 parts of 28 bits each → C_0, D_0

Now, these bits are shifted with left shift in each round.

in Rounds $i = 1, 2, 9, 16$ → 1 shift i.e rotated left by 1 bit

in other rounds, 3, 4, 5, 6, 7, 8, 10, 11, ... 15 two halves rotated left by 2 bits.



2nd part/block

58, 59 ... 64
8th part

each part, last bit → discarded
bit → 8, 16, 24, 32, ... 64 discarded.

we have 8 parts of 7 bits each
= 8 * 7 = 56 bits

PC-1 is 56 bits which is then divided
2 parts of 28 bits each → C0, D0
these bits are shifted with left shift in
each round.

rnds i = 1, 2, 9, 16 → 1 shift i.e
rotated left by 1 bit
3, 4, 5, 6, 7, 8, 10, 11, ... 15
er rounds,
two halves rotated left by
2 bits.

shifting, we get (C1, D1) which goes
up to PC-2

Inside PC-2 56 bit → 48 bit
using a predefined table
Then we get our 1st key
for Round 1.

In C_1 → 28 bit → (1-28)
D_1 → 28 bits → (29-56)

Now 56 bit ⇄ how 48 selected?

Left half C_1 (9, 18, 22, 25, position bits are missing).
ie 24 left

Right half D_1 (35, 38 , 54 position bits are missing)
ie removed.
ie 24 left

10 → 12
1 → 6.

DES ANALYSIS

Properties

(i) Avalanche effect → It means a small change in plaintext (or key) should create a significant change in the ciphertext.

DES has been proved to be strong with regard to this property.

eg. Plain → 0000000000000000
cipher → 4789FD476E82A5F1

plain → 0000000000000001
cipher → 0A4ED5C15A63FEA3

key used is same say
key = 22234512987ABB23

plain → 0000000000000000 1
Cipher → 0A4ED5C15A63FEA3

Key used is same say
key = 22234512987ABB23

Although, the two plaintexts differ only in 1 bit, ciphertext block differs a lot/significantly.

2. Completeness effect → It means that each bit of the ciphertext needs to depend on many bits on the plaintext.

The confusion and diffusion produced by D-boxes and S-boxes in DES, show a very strong completeness effect.

DES Weaknesses

- **Key size** — 2^{56}

Critics believe that the most serious weakness of DES is its key size of 56 bits.

B/c, with today's technology (like parallel processing and v. powerful processors) it can easily be cracked.

↓

2^{56} keys (brute force attack)

→ ∴ we use triple DES (3DES) with two keys (112 bits) or triple DES with 3 keys (168 bits).

(ii) **Weak keys** → four out of 2^{56} keys are called weak keys. after the parity drop operation is the one which consists of

(ii) Weak keys → four out of 2^{56} keys are called weak keys.

A weak key is the one which, after the parity drop operation consists of all 0s, all 1's or half 0's and half 1's.

The disadvantage of using a weak key is:
If we encrypt a block with a weak key and subsequently encrypt the result with the same weak key, we get the original block.

The process creates the same original block if we decrypt the block twice.

So, if after 2 decryptions, if the result is the same then, attacker is successful.

(iii) Semi weak keys → Six key pairs are called semi-weak keys. (refer book pg 154).

A semi-weak key creates only two different round keys, and thus each of them is repeated 8 times. (show in book pg 155).

key → K_1 but
key → K_2

(iv) Possible weak keys → 48

There are 48 keys that are called possible weak keys.

A possible weak key is a key that creates only 4 distinct round keys, in other words, the 16 round keys are divided into 4 groups and each group is made of 4 equal keys.

(iv) Possible weak keys → 48

There are (48) keys that are called possible weak keys.

A possible weak key is a key that creates only 4 distinct round keys, in other words, the 16 round keys are divided into 4 groups and each group is made of 4 equal keys.

(v) Key clustering

means 2 or more dif keys can create the same ciphertext from the plain text.

Weakness in cipher Design

(i) Two specifically chosen i/p's to S-box array can create the same o/p.

DIFFERENCE BTW AES and DES

AES

(i) AES stands for Advanced Encryption Standard

(ii) Key length can be 128 bits, 192 bits or 256 bits.

(iii) no. of rounds depends on the key length

Round	bits
10	→ 128
12	→ 192
14	→ 256

(iv) The structure is based on the substitution-permutation network.

(v) AES is more secure than DES and is the de-facto world standard.

(vi) Rounds in AES are: byte substitution, shift Row, Mixcolumn and key addition.

DES

(i) DES stands for Data Encryption Standard

(ii) key length is 64 bits (56 bits in each round)

(iii) DES involves 16 rounds of identical operations.

(iv) The structure is based on Fiestel network.

(v) It is less secure. It can be broken down (i.e, it is weak). 3DES more secure than DES

(vi) Rounds in DES are: Expansion, XOR operation with round key, substitution and permutation

MODULUS OF -VE NUMBER

$n \pmod m$

$n = qm + R$

$51 \equiv 1 \pmod{10}$

$-51 \equiv 9 \pmod{10}$

eg 1. $-51 = (-6)10 + R$ $\quad -51 = -60 + R$ $\quad R = 9$

we have to choose q such that we get a more −ve no. than 51 or the same −ve no.

$\therefore$ we chose $q = -6$

$\therefore$ Remainder $= 9$

$10 \pmod 7$

$10 = 7 * 1 + 3$

eg 2. $-50 \pmod{10} = ?$

$-50 = (-5)10 + R$

we can choose q such that the no. becomes the same as -50

$R = 0$

eg 3. $-37 \equiv ? \pmod 5$

$-37 = (-8)5 + R$

$-40 + R = -37$

$R = 3$

Method 2

$(x \bmod y)$ if $|x| \bmod y \neq 0$