Wednesday, December 2, 2020 10:41 AM

Decryption of Hill Cipher – we'll do it later

Monoalphabetic Cipher
- rather shifting we shuffle the letters arbitrarily.
- each PT letter gets mapped to a different random cipher letter.

- So the Key length = 26 letter long.

Polyalphabetic Cipher
- each letter can be encoded by any letter in cipher alphabet & each letter represents different letter from PT each time it appears.

ATBASH Cipher (monoalphabetic)

PT	A-M	N-Z
CT	Z-N	M-A

a b c d e . . . . z
z y x w . . . . . a

Using ATBASH Cipher, encode your name.

Plaintext: B I K A S H
1 8 10 0 18 7

Ciphertext: 24 17 15 25 7 18
Y Q O Z H S

Affine Ciphers:
Encryption: $E(x) = (ax+b) \bmod m$

$ax+b$

Chapter # 4 MODERN SYMMETRIC CIPHERS

Binary Block Substitution

Plaintext b_0 b_1 b_2 b_3

Converting incoming bits to one of 16 integers

Random 1 to 1 mapping of i/p integers to 16 o/p integers

Convert integers to 4 bit patterns

Ciphertext t_0 t_1 t_2 t_3

N

(for n=4)

In a modern block cipher, we replace a block of N-bits from plaintext with a block of N-bits from a ciphertext. This general idea is illustrated in the figure alongside (above) (for n=4) (although N is set to 64 or its multiples

Flash

Procedure: ⇒ firstly, the 4 bit patterns are converted into 24 different possible patterns which we can represent these patterns by an integer bet^n 0 & 15 (2^4=16).

0000 - 0
0001 - 1
⋮
1111 - 15

} total of 16 integers

⇒ In an ideal cipher, the relationship bet^n i/p & o/p block is completely random. But it must reversible for decryption work. ∴ it has to be 1 to 1 mapping

why ??
each i/p block is uniquely mapped to each o/p.

SECURITY IS INSECURITY.

⇒ The mapping bet^n i/p & o/p blocks can be constructed as a mapping from integers corresponding to i/p block to the integers corresponding to o/p block.

⇒ The encryption key for the ideal block cipher is the code book itself meaning the table shows relationship bet^n i/p & o/p blocks.

⇒ in the fig. above, an ideal block cipher is shown that uses block size of 4. Each block of 4 bits in the p.T. is converted 4 bits of ciphertext.

Size of Encryption for binary block substitution

Consider a 64 bit block encryption with a 64 bit Key, we can consider each i/p block as 2^{64} integers and for each such integer we specify o/p block of 64 bits. We can construct the code block by displaying just the o/p blocks in order of corresponding i/p blocks such a code block will be size $64 * 2^{64}$ i.e 10^{21} (approx.)

This implies that the encryption key for an ideal block cipher using 64 bit blocks will be of size 10^{21} (size in KB ⇒ ??).

So, in practical idea thinking of logical issues related to transmission storage and processing such large keys, this is quite a hefty load.

Shannon's Theory of Confusion & Diffusion

→ Block cipher looks extremely large substitution as we need table of 2^{64} entries for 64 bit blocks.

→ Using product cipher idea 1949, Claude Shannon introduced idea of substitution Permutation (S-P) network called modern substitution transpose product cipher. These form the basics for modern block ciphers.

→ S-P network based on two primitive cryptographic operations we've seen for → substitutions (S-box)
→ permutation (p-box)

→ provides confusion & diffusion in messages.

Diffusion:
→ dissipates the statistical structure of plaintext over block of ciphertext.
This mechanism of diffusion seeks to make the statistical relationship betⁿ the plaintext ciphertext as complex as possible in order to deduce the key.

Confusion:
– makes the relationship betⁿ ciphertext & key as complex as possible to make thwart attempts to discover the key very tedious task.
This mechanism of confusion seeks to make the statistical relationship betⁿ the key & ciphertext as complex as possible.

Fiestel Cipher Structure

– implements shannon's diffusion & ~~comp~~ confusion theory (S-P Network Concept) based on an invertible product cipher.

S-Substitution P-Permutation

P.T. → C.T. → P.T.

– Horst Fiestel (inventor) developed Fiestel Cipher in early 1970 at IBM and it was first implemented by Fiestel & Don Coppersmith in Lucifer Cipher.

— Feistel Cipher is based on same basic algorithm for encryption and decryption. The feistel Cipher structure consists of multiple round of processing of plain text with each round consisting of substitution round followed by permutation round as shown in the figure below:

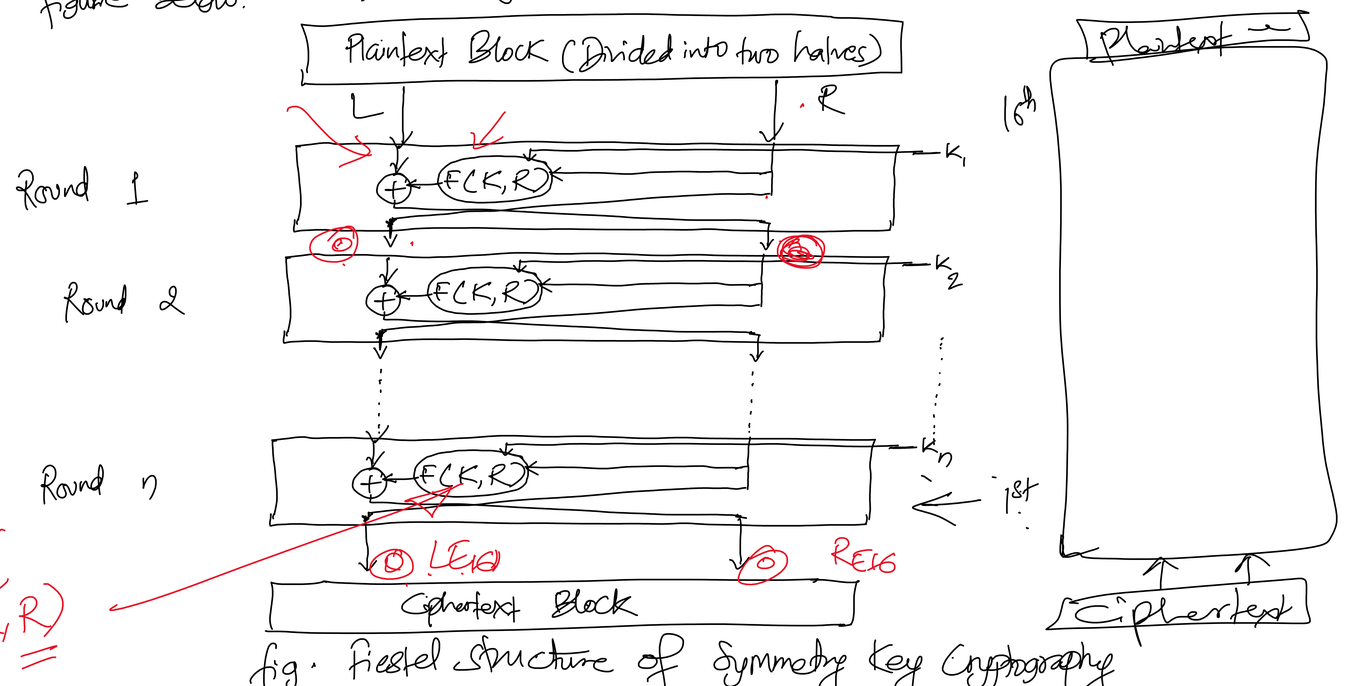


fig. Feistel Structure of Symmetry key Cryptography

→ The plaintext block is divided into two halves at each round & we denote these halves with L & R
(L: left, R: right)

→ In each round, the right half of the block (i.e. R) goes through an operation that depends upon R and the encryption key K.

→ The permutation step at each round consists of swapping the modified L & R. Hence, the current L for the next round would be R and the current R would be L for the next round.
This process continues for 'n' rounds.

(E - encryption, D - decryption)

Mathematical Description:

Let RE_i & LE_i denote the O/p half block at the end of i^{th} round of processing. The (after E denotes encryption in Feistel structure, the relationship b/w i/p of i^{th} round and O/p of previous round (i.e $(i-1)^{th}$ round) is given by.

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$

Where, $\oplus$: bitwise XOR operation
F : some function / operation that scrambles the previous RE_{i-1} with the round key K_i.

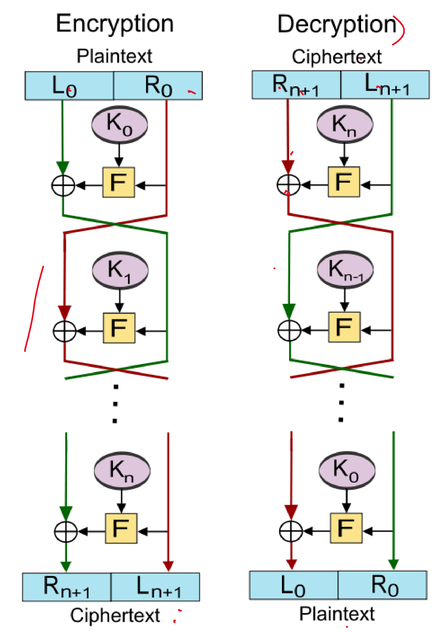
— The round key is derived from the main encryption key.
— The function F is referred to as Feistel Function after Horst Feistel. If we assume 16 rounds of processing the last round, the processing is given by:

$$LE_{16} = RE_{15}$$
$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$

$LD_1 = LE_{16}$

Decryption of Ciphertext based on Feistel Structure

The decryption algorithm is exactly the same as encryption algorithm with the only difference that the round keys are used in reverse order. The o/p of each round during decryption is the input to corresponding round during encryption except for left right switch bet^n the two halves. This property hold true regardless of the choice of Feistel function.



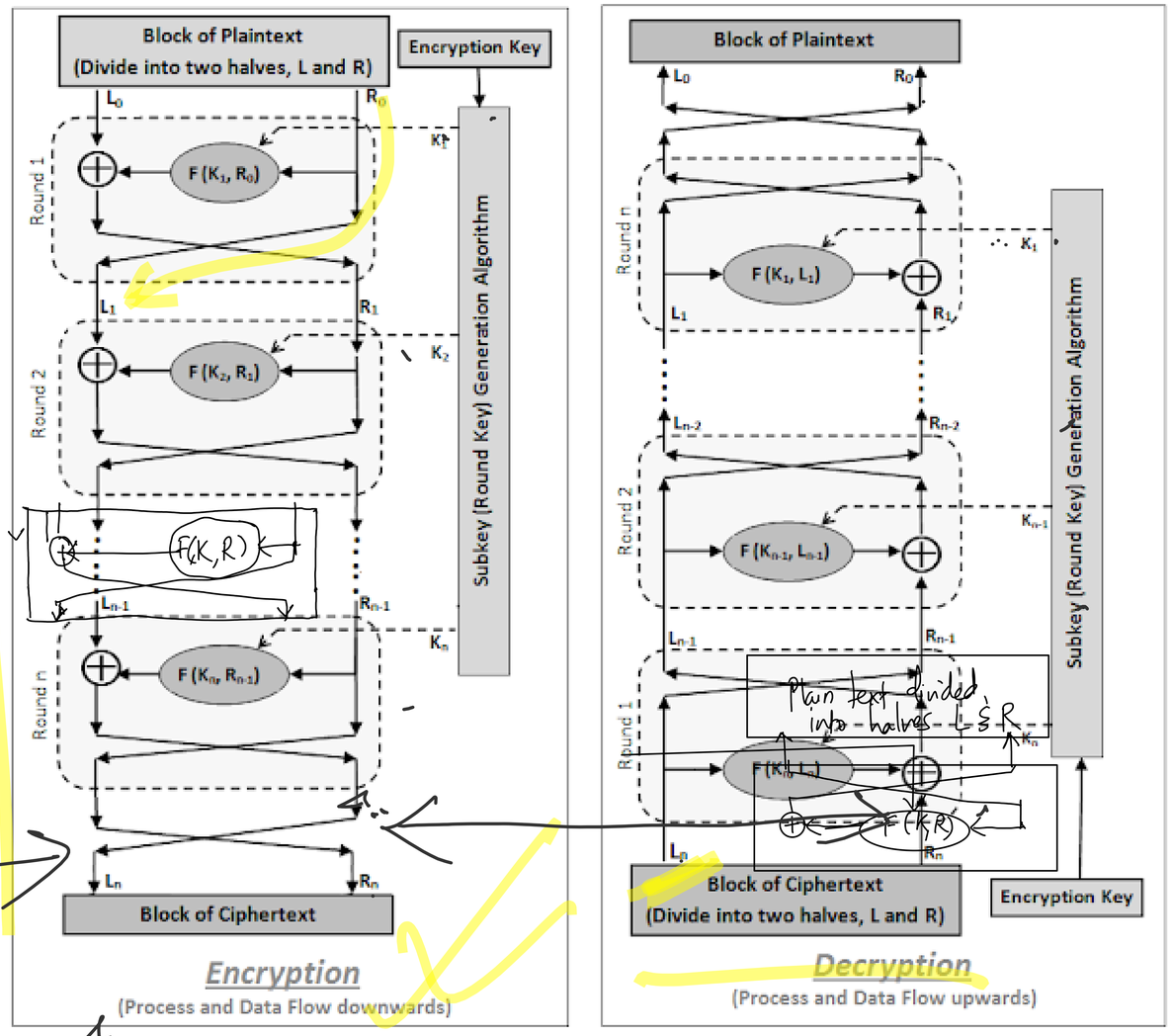
$LD_1 = LE_{16}$

Lucifer Cipher
1970 Feistel
DES - 1974
TDES
AES / RSA
IDEA

Data Encryption Standard (DES)
— In 1974 IBM developed DES based on their Lucifer Cipher Algorithm.
— Symmetric key block Cipher.
— DES is a remarkably engineered algorithm with powerful influence in cryptography.
— DES has a key length of 56 bits (+ 8 parity bits : total 64 bits) and block length of n = 64 bits. It consists of 16 rounds of Feistel Network.
— A group called Electronic Frontier Foundation broke DES in 22 hrs and 15 mins.

10^{17}

- NIST issued an update ... to be triple DES (3-DES) i.e three consecutive application of DES.
- Using permutations, the round keys are generated from the main key.
- The key length is of 48 bits.



$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$

$$LE_{16} = RE_{15}$$
$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$

Decryption

round number

left half

LD_0

Decryption

$$LD_0 = RE_{16}$$
$$RD_0 = LE_{16}$$

Now for 1st decryption Round, we write the eqn as:

$$LD_1 = RD_0$$
$$\therefore LD_1 = LE_{16} = RE_{15}$$

$$\therefore RD_0 = LD_1 = RE_{15}$$

XOR

Again, $RD = LD_0 \oplus F(RD_0, K_{16})$

$= RE_{16} \oplus F(RE_{15}, K_{16})$

$= LE_{15} \oplus F(RE_{15}, K_{16}) \oplus F(RE_{15}, K_{16})$

$= LE_{15}$

$A \oplus A = 0$

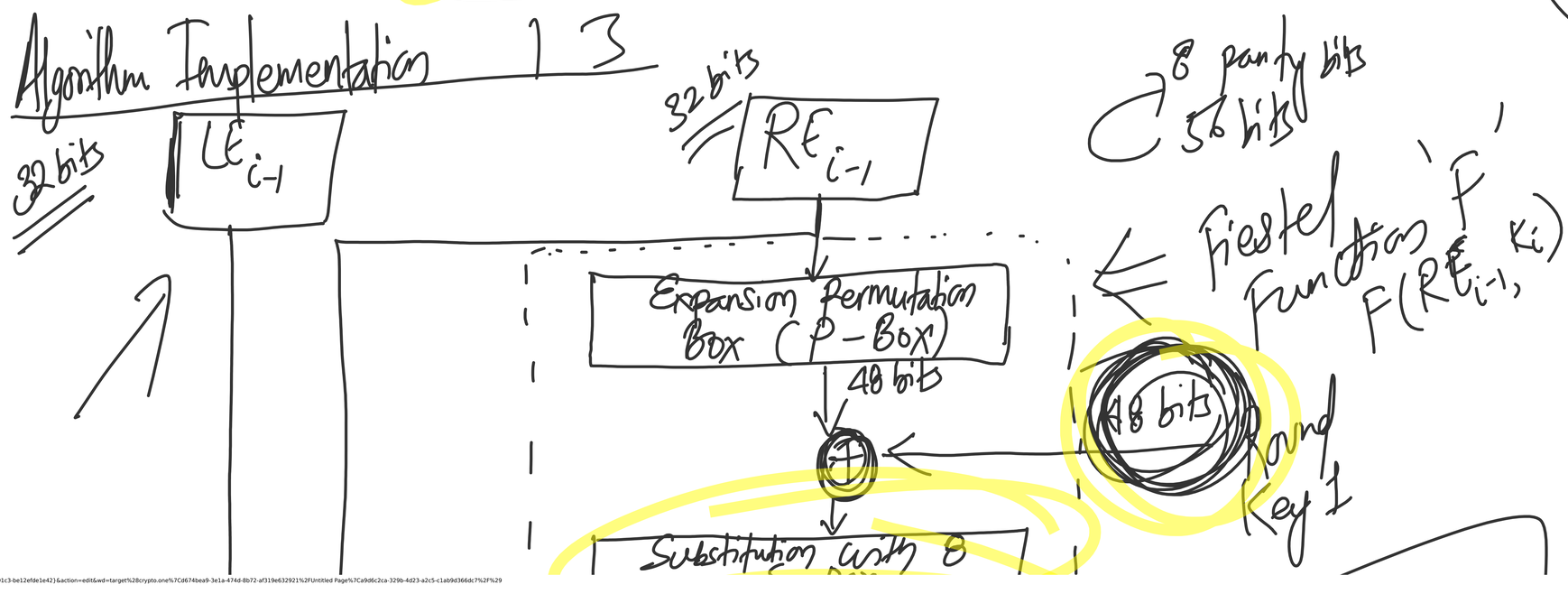
$A \oplus 0 = A$

This shows that except for the left right switch, the O/p of the first round of decryption is the same as the i/p of the last round of encryption.

$LD_0 = RE_{16}$	$RD_1 = LE_{15}$
$RD_0 = LE_{16}$	$LD_1 = RE_{15}$

$(A \oplus B) \oplus C = A \oplus (B \oplus C)$

Algorithm Implementation 13



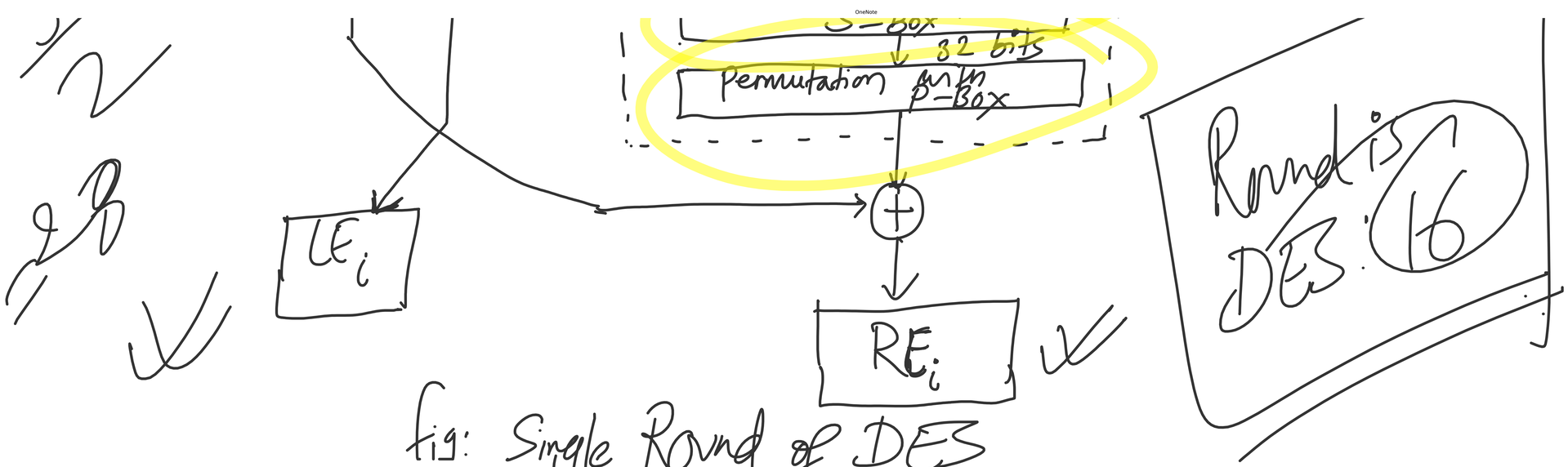


fig: Single Round of DES

Algorithm Implementation of DES is called DEA (Data Encryption Algorithm)

1. $64 \longrightarrow 32$ bits $\longrightarrow 48$ bits.
2. Expansion Permutation $\rightarrow$ E-setup.
3. E setup steps:
   (i) Divide the 32 bit block in Eight 4 bit words.
   (ii) Attach an additional bit to the left of each bit word i.e. the last bit of previous word.
   (iii) this attached word is the beginning of the right of each 4 bit word (i.e the beginning of next 4 bit word).
4. Divide the 56 bit key into two halves. Each of the half

is permuted to yield a 32 bit

(5) Round Key is XORed with the 48 bit of expanded o/p of the E-step. This step is known as key mixing.

(6) Divide the o/p generated in previous step into eight 6-bit words. Each of these 6 bit words go through substitution step. It's replacement is a 4 bit words.

An S-box is used for the substitution purpose.

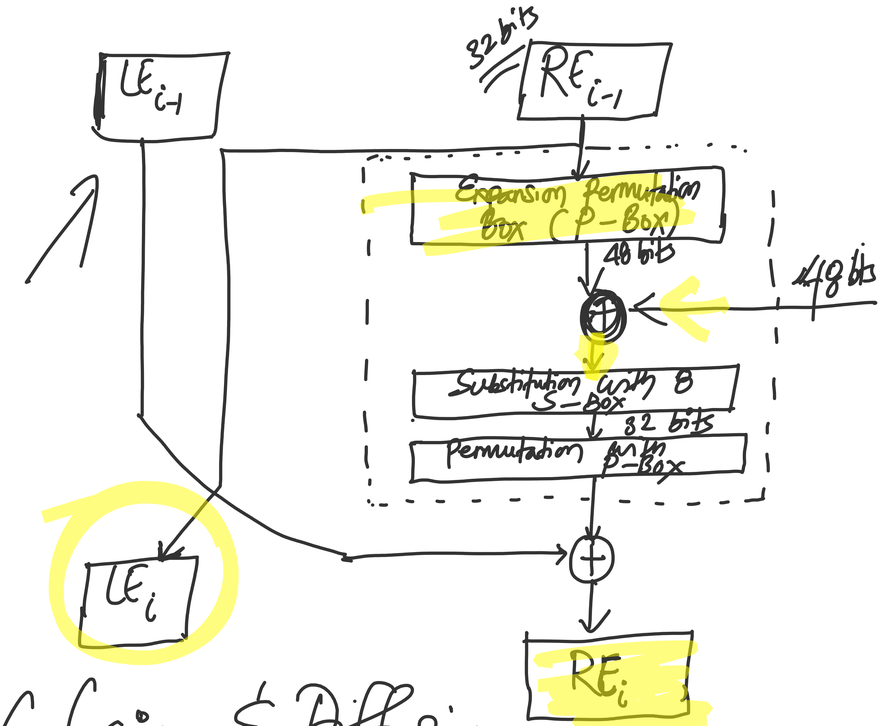
$6 \times 8 \Rightarrow 4 \times 8$

48 bits 32 bits

(7) # After all substitution, we end up with 32 bits which goes through a P-box permutation.

(8) The o/p of P-box is XORed with the left half of 64 bit block that we started with.

This o/p will be right half for the next round.



Confusion & Diffusion

(Statistical relation as complex as possible.)

→ Substitution step (main step) Using S-box

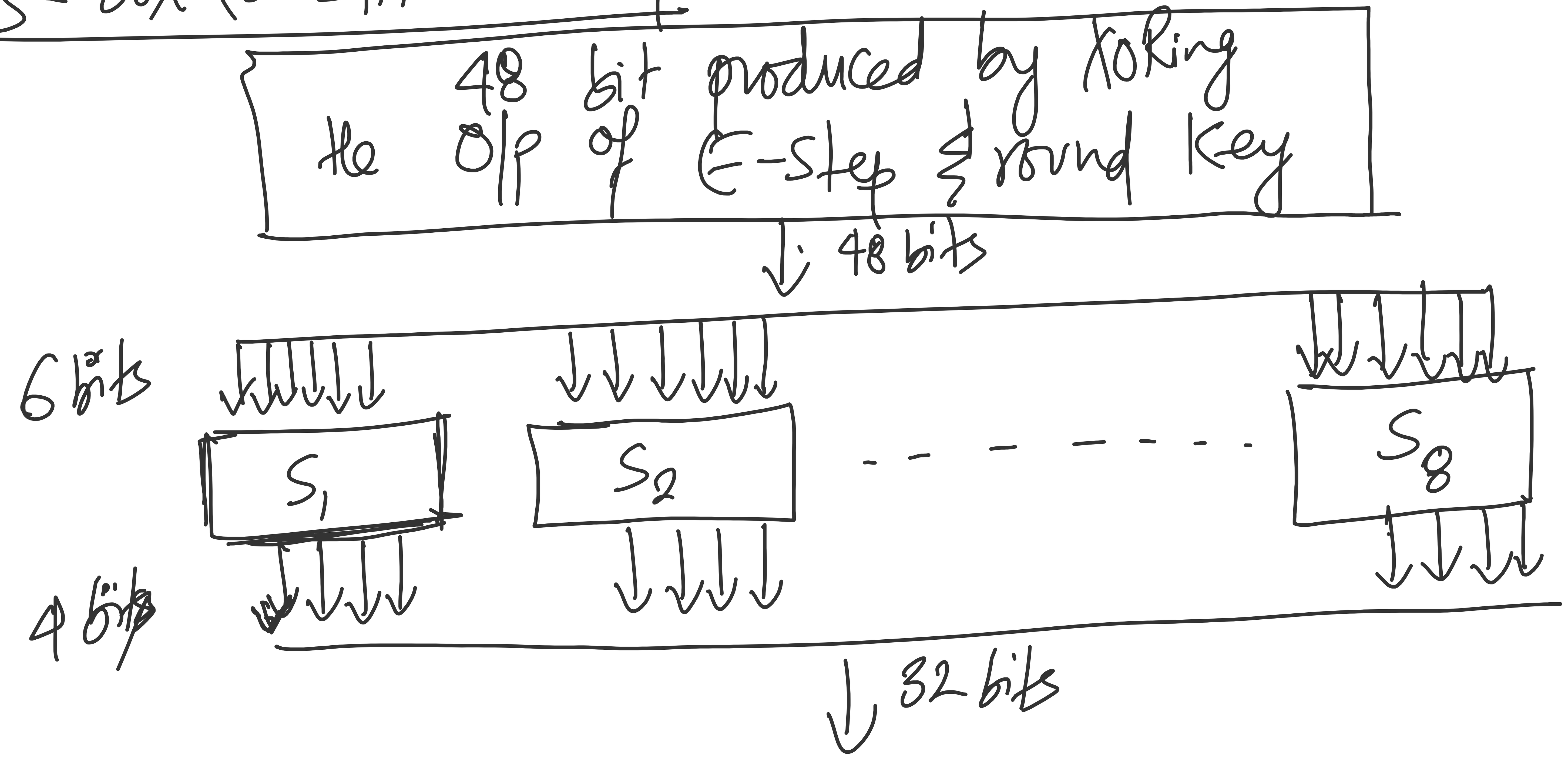
→ Substitution Step (main goal)
– to introduce diffusion (P.T, C.T)

→ multiple round Keys : to introduce Confusion. (P.T., Key)

S-box (Substitution Step) / P-box (Permutation Step)

Generation of Round Key

S-box (Substitution Step)



48 → 32 bits Converted O/p.

Each S-box Consists of a 4X16 look up table for a 4-bit word O/p. The first & last bit of 6 bit i/p is decoded into one of 4 rows & middle 4 bits decoded 16 Columns of the look up table.

	0	1	2	3	4	. . . . .	15.
0	14	4	13	1	2		
1	0	15	7	4	14		
2	4	1	14	3	13		
3	15	12	8	2	4		

fig: 4X16 Sbox look up table

different

P-box Permutation

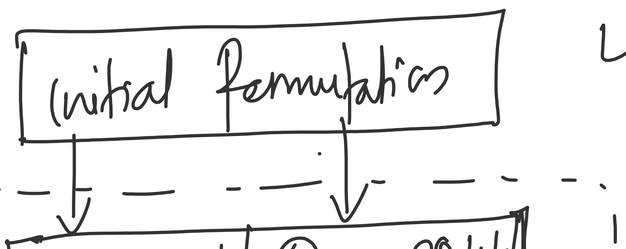
12	6	19	20	27	11	27	16
0	14	22	25	4	17	30	9
1	7	23	13	31	26	2	8
18	12	25	5	21	10	3	24

← place values of i/p.

This permutation table says that the 0th o/p bit will be 12th i/p bit, 1st o/p bit will be the 6th i/p bit and so on for all 32 bit o/p.

This yeilds 32 bits o/p.

DES Key Generation



16
32 bit
48 bit
24 bit

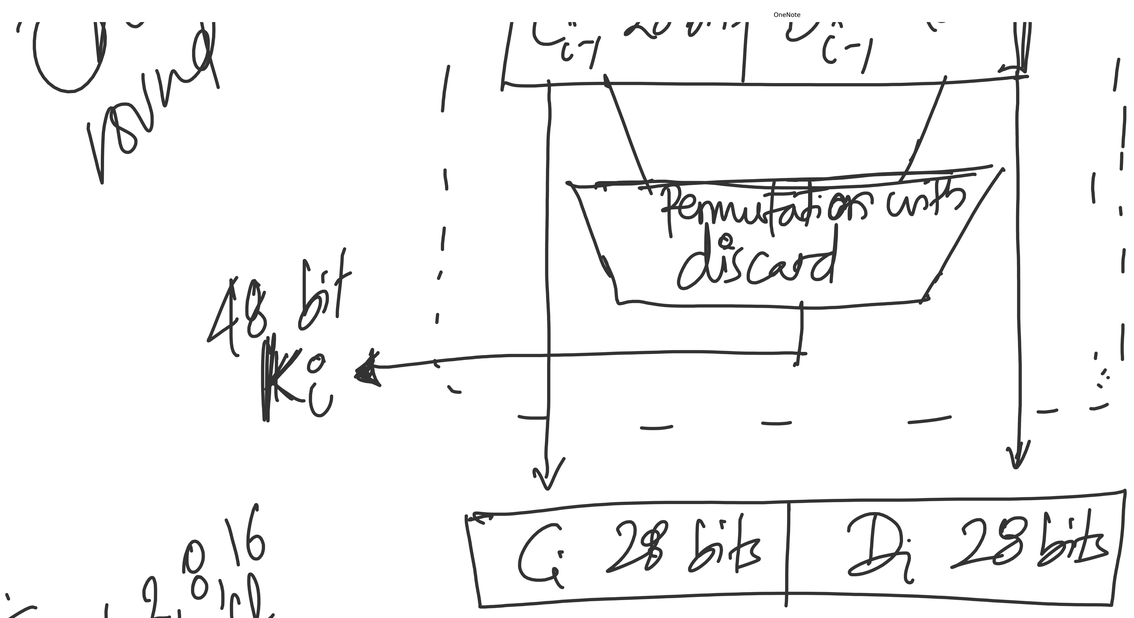
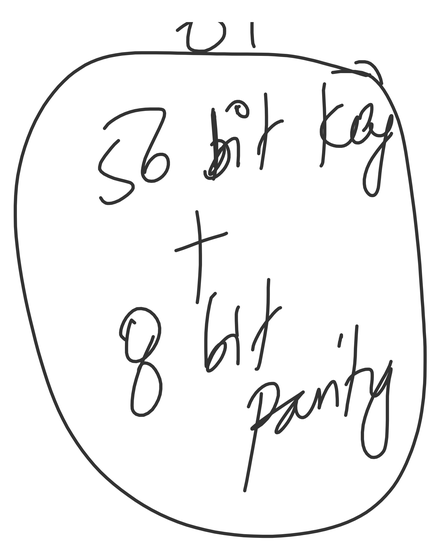


Fig. DES Key Generation



Note:
Round 1, 2, 9, 16 single shift
Others 2-bit shift

11111001
← 1 bit Shift
11110010
← 2 bit Shift
11100100

Single round of DES

Round key

S-box

DES -

DEA -

AES -