other key is generated.

Ex: PT = SECURITY and Key = ACMTKYIV

PT → SECURITY →	18	4	2	20	17	8	19	24
Key → ACMTKYIV →	0	2	12	19	10	24	8	21
	18	6	14	39	27	32	27	45 → Add

After getting set of no's greater than 26, we should subtract with (−26) to the greater no's.

After performing addition we get set of no's. The set of no's are greater than 26 we have to identify and subtract with (−26) to the greater number.

	18	6	14	39 (−26)	27	32	27	45
Subtract	18	6	14	13	1	6	1	19 → CT
	S	G	O	N	B	G	B	T

These are the Cipher Text numbers. Corresponding to this number we should find correspond alphabets.

The Sender Sends SECURITY to word to receives by using One time pad Cipher it will be Sent in SGONBGBT the receiver attacker cannot understand. So, after reaching on receiver side the receives will decrypt it. Decryption means going reverse (Bottom Down to Top) order. again he will read the word SECURITY.

RAILFENCE TRANSPOSITION

Rearranging the existing words.

↳ Here plain text and depth are given. No need of key here. We have to take plain text and arrange diagonally.

Ex:

T		O		B		E		R		E
	R		U		L		F		E	

We are rearranging the words diagonally. When you are writing in CT we should write in row wise. like "TOBERERULFE". We didnot include any new alphabet in this. Already existing alphabets are rearranged. If depth is 2, then we have to write 2 rows.

↳ According to depth the no of rows will be increased.

After writing diagnally, you have to write the CT in row wise. You are writing PT in diagnally.

↳ It is used for short messages and it is not so efficient.

COLUMNAR TRANSPOSITION:

We have to rearrange.

First, we have to take plain text msg and arrange it in a matrix. It is not mandatory that we have to take always take square matrix only. We should be careful that how many Columns we are taking. We can take any no of rows. We should keep in mind that how many Columns we took.

We should fill all the alphabets in plain Text into this matrix row wise.

For Ex: Information Security, we have to fill this word into this matrix in row wise.

After arranging all this words into matrix then we have to generate a key. Key is having only numbers all the numbers it will have. All those no's should be less than the no. of columns or equal to the no. of Columns.

1	2	3	4	5
I	N	F	O	R
M	A	T	I	O
N	S	E	C	U
R	I	T	Y	

Key = 32514

↳ We can take our wish. the key value should be less than "5". or equal to '5'. Each and every Element should be less than or 5 or equal to 5.

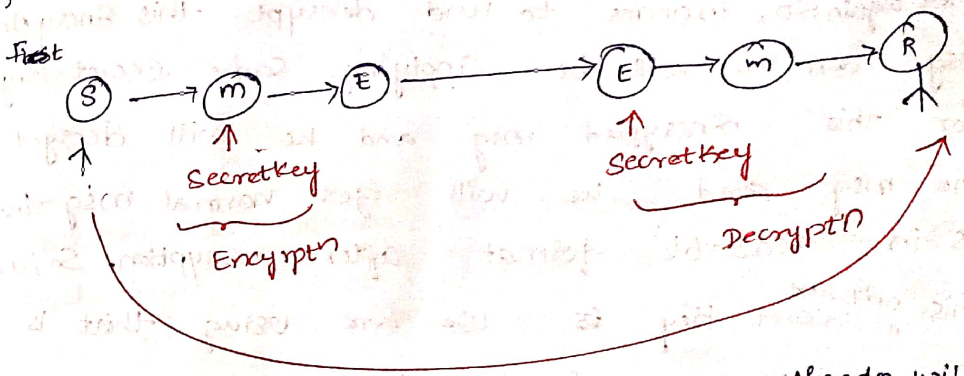
CT = FTET NASI ROUIMNROICY

∴ This is our Encrypted msg the Ciphers Text msg.

* In substitution we do replacement
In Transposition we do rearrangement

Symmetric key :-

↳ The main keyword we use is One key. Both sender side and receiver side. That means we will use Only one key for encryption process and decryptn process.



First Sender want to Send msg to Receiver. Sender will generate a msg as HELLO after generating this msg, this msg has to be encrypted. How this msg will be encrypted by using a Secret key the msg will be encrypted. He will apply secret key on this msg and it converts into Unreadable format which is nothing but our cipher Text and this process is called Encryption. What secret key we are using for Encryption we are using same?

Secret key for Decryption also. So, the Encrypted msg will be Enter into a network this is any transmission media it may be internet or text msg it will Enter into a transmission media. After entering the transmission media the Encrypted msg will be reached to the Reciever side. The receiver side Cannot understand the encrypted msg, he has to decrypt the msg into plain Text again So, inorder to decrypt this Encrypted msg he will be applying same Secret key for this Encrypted msg and he will decrypt the msg and he will get normal msg that is in readable format after decryption. So, in this only one Secret key we are using that is

Symmetric key Cryptography.

Adv:- It Can be easily implemented because, there is only one key. we need not generate two keys.

Disadv:- As, we are having only one key It can be easily hacked. It can be Easily known to 3rd person that means we are having 3rd party who is observing our Communica For ex: So when this 3rd person if he tries to get the key of Sender he will the key of receiver also. Because both of them are same. If it is having both of them are different If he get this only he can see what is the msg. he cannot of Receiver. It is not so effecient compare to Assymetric key.

Assymetric Keys:- In this we are having different keys on sender and Receiver side. Not same keys. There are 2 types of Keys 1 is public key 2 is Private Key.

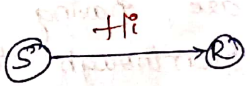
* Publickey is known to Everybody
* Private key is not known to Everbody. Only that person will be knowning about private key.

→ In Assymetic key what happens & how the process will be going on;

→ First, the Sender will generate the msg that he want to send to Receiver. After generating the msg so, the sender has to generate encrypt the msg inorder to encrypt a msg we need a key. we will use public key of 'R'. 'R' means Receivers Public key is used the msg is encrypted. That encrypted msg is generated. This Encrypted msg will be transmitted to the receiver side through network. On reaching receiver side this Encrypt msg is should be Converted into normal msg that

into PT again. This is done by Decryption. Inorder to decrypt the msg we need a Key. Here, we are using private key of 'R'. On using Private key of 'R' on this 'E' encrypted msg and it will be converted into normal msg 'm' that is in readable format. then receiver can easily read the msg. In this process we are having some more security when compare to Symmetric key Cryptography. It is not some Secure but little bit more than Symmetric key Cryptography.

Steganography:- Steganography means hiding information with in another msg. Whenever we are sending Information from Sender to Receiver third party person will be observing. In Steganography what we do is we will be Embedding our msg in a image, vedio or in file.. then we will be sending the msg instead of sending msg directly our msg is embedded, our msg is rolled in a different type of file like Image file or vedio file or a PDF file then the msg is sent. So, that the third person who is observing that he feel like the image is being transferred. But, actually what is being transferred is present inside the image is being transferred. So, after transfering the msg from sender to Receiver then later msg is extracted from Embedded devices by Receiver. Embedded device means image file, vedio file or whatever the device is used which are using externally. Internally some msg is present. Suppose you, want to send "Hi". Here, Sender and Receiver you want to send msg



"Hi". Instead of sending directly msg "Hi" you, will be embedding "Hi" in a image. In a image inside the image the msg will be these. Whenever we are sending this msg from Sender to Receiver the third person will feel like this is an image, the image being transferred. transmitted. But actually what is being transmitted the msg which is present inside the image is being transmitted. On reaching receiver side receiver will extract the msg. he will take the msg from image and then he will read the msg.

We have several techniques performed by Steganography "LSB" which stands for Least Significant Bit. and audio / vedio Steganography, character marking we have 'n' no. of steganography techniques through which we will be doing steganography

Difference between Cryptography and Steganography

↳ In cryptography we will be hiding the msgs from attackers.

↳ In steganography we will be learning how to hide the msg by embedding that msg by inserting that msg in another msg that is the main difference between cryptography and steganography.

↳ Even in Steganography we are having some possible type of attacks. Even though the msgs is embedded in another msg also we will have some type of attacks. We have some attacks in cryptography like active & passive attacks in the same way we are having some attacks in steganography also. like 3rd person will try to modify the data all those type of attacks are possible in steganography.

Key Range:-

Key Range is total number of keys from smallest to largest available key

So, Ex:- Keys : TOM
JERRY
DOREMON
SPIDERMAN
} Key Range

This above, all are bunch of keys.

UNIT-II

1) BLOCK CIPHER PRINCIPLES AND ALGORITHMS :-

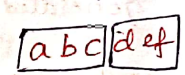
Block Cipher :-

↳ Our main moto is to convert PT in to CT.

So, the third person cannot understand what we are sending.

↳ We are dividing our plain text, is divided into no. of blocks.

↳ Suppose we have abcdef as a plain Text. we will be dividing it into no. of blocks.



2 Blocks

↳ After dividing the plain text into no. of blocks we will be converting each individual blocks into a cipher Text block.

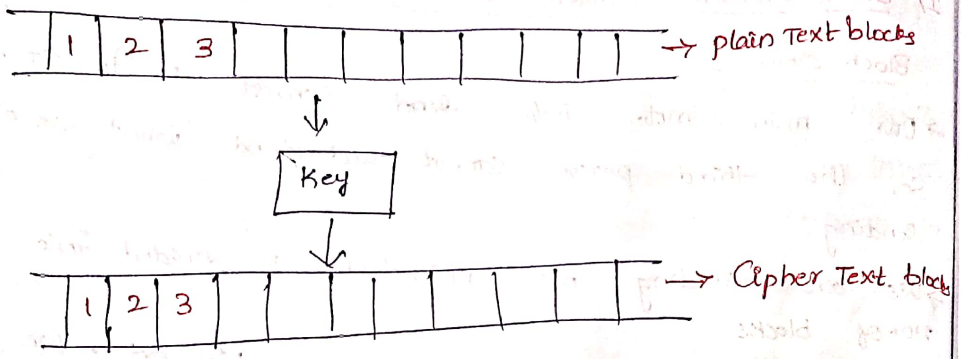
Block size (40, 56, 64, 128, 256 bits)

Whenever, we are dividing the blocks then, we are converting the PT to CT. Always, make sure that PT block size should be equal to CT block size.

PT block size = CT block size

Ex:- Block size of PT = 40 bit
The corresponding CT = 40 bit
} It should be in same size.

↳ The conversion of PT to CT is happen through Key.



↳ For each and every single blocks of plain text the individual CT block is generated.

↳ Whatever PT you have you will be divided into no. of blocks. After dividing PT into n blocks you will be using the Key, with that Key you will convert into CT.

↳ After generating the corresponding CT blocks now we have to combine all CT blocks.

abcdef

PT [ab] [cd] [ef] — For each & every text by using a Key we generate CT.

↓K ↓K ↓K

CT [xy] [kl] [mo]

xykImo

This xykImo is corresponding CT for abcdef Corresponding PT.

We have several algorithms to generate CT. After generating CT, this CT will send to receiver side, On reaching receiver use the key and he will generate corresponding PT. Then he will combine all the blocks & he will read actual msg.

Block Ciphers principles:-

Three design principles are there. They are:

1. Numbers of Rounds → 10R, 16R, 20R.

↳ Each & every algorithm has several Rounds.

↳ How many higher no. of rounds will have that much tough the algorithm to the third party person to break it.

↳ How many rounds will be more that much hard will become to the hacker.

No. of Rounds should be more.

2. Design of function F:- We will have a function based on that function only means like f(x) = ax+b

↳ You must design a function 'f' which will be very much complicated to understand. How much harder the function is that much more time the hackers takes to Decode or to break the algorithm.

We Should take **Non-linear functions**. bcz they are complicated.

3. Key Schedule Algorithm: You should be careful while generating a Key because Key is very important. If we are having minor change in Key lot of

changes will be there.

Block Ciphers modes of operations:-

ECB – Electronic Code Book
CBC – Ciphes Block Chaining
CFB → Cipher Feed Back mode
OFB → Output Feed Back mode
CTR → Counter modes.

Block Ciphes algorithms:-

1. Data Encryption Standard
2. Advanced Encryption Standard
3. Blowfish algorithm

DES algorithm:- (Data Encryption standard)

↳ Converting PT to CT.
↳ It is Block Ciphes algorithm.
It has total of 16 Rounds.
Text size = 64 bits → PT.
Key size = 48 bits.

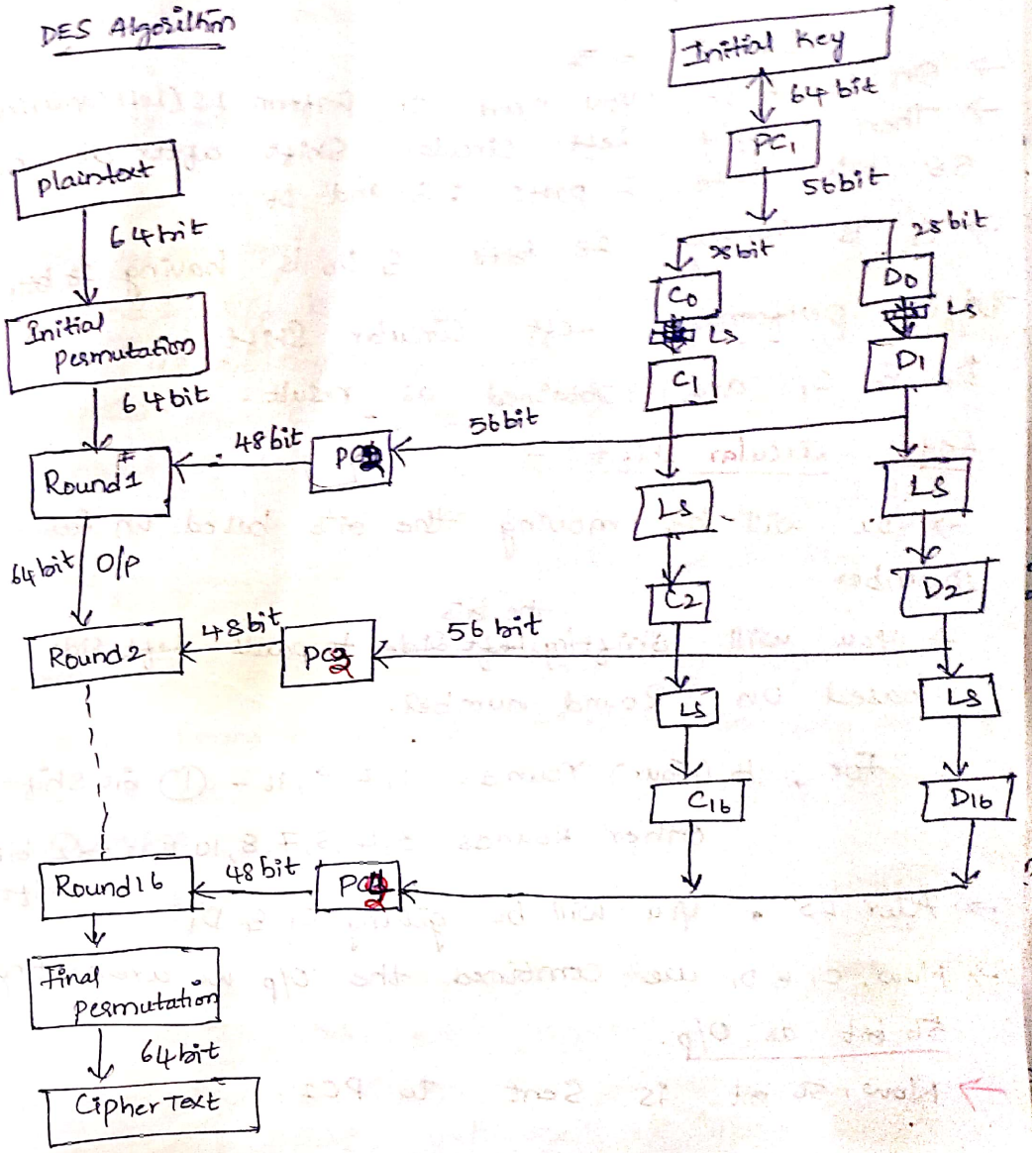
Why we got 48 bits as Key size means 16 bits are gone. 8 bits are removed for parity and 8 bits for rearrangement.

In each round 4 steps are performed.
Total 16 Rounds will be there. In each & every round 4 steps will be there inside.

4 steps:

1. Dividing Bits into 2 parts of 32-bit each.
2. Bit shuffling.
3. Non linear Substitutions.
4. Exclusive OR Operations.

DES Algorithm



→ First we have Initial Key which is of 64 bit

→ 64 bit will be sent as input for PC1.

Inside PC1 (Permuted choice 1)

→ From, 64 bits, 8 parity bits are to be removed from every 8th position.

In 64 → 8×8

∴ 64 − 8 = 56.

→ On C_0 & D_0 you need to perform LS (Left Circular shift)

→ Then apply Left Circular Shift after dividing 56 bits into 2 parts i.e C_0 and D_0.

→ C_0 is having 28 bits & D_0 is having 28 bits

→ After performing Left Circular Shift D_1 & C_1 are obtained as result.

Left circular shift :—

→ You will be moving the bits based on Round number.

→ You will Shifting the bits towards Left side based on Round number.

For, 4 (Four) rounds 1, 2, 9, 16 — (1) bit shift

Other Rounds 3, 4, 5, 7, 8, 10, 11, 12 → (2) bit shift

→ After LS, you will be getting C_1 & D_1.

→ Now, C_1 & D_1 are combined the o/p we are getting 56 bit as o/p.

→ Now, 56 bit is sent to PC2.

In PC2 :—

↳ C_1 and D_1 are combined to form 56 bits again. Permuted choice 2 is applied.

↳ 56 bits are rearranged, they are permuted and 48 bits are selected.

↳ We remove 8 bits from 56 bits then it became 48 bits.

↳ This 48 bits will send for key for round 1

→ I/p for round 1 is 48 bits.

→ This 48 bit key i/p is given for Round 1.

→ Round 1 o/p is 64 bit. is generated.

→ Next how 48 bit key is generated, on C_1 & D_1 LS is applied then C_2 & D_2 will get.

→ combine C_2 & D_2 then it give to PC2

→ 56 bit is given to PC2 will do permutation, rearrangement & selection & generate 48 bit key for Round 2.

→ This same process will be repeating upto 16

→ After Round 16 you get 64 bit key then you will be applying final permutation

→ After final permutation you get Cipher Text.

↳ This is the Algorithm for to convert PT to CT. In this we perform 16 rounds.

↳ for each & every round we have 4 substeps in that, and each & every round

What is the i/p we are giving the o/p of the previous round plus 48 bit key.

→ for all the rounds the 48 bit key will not be given. It will have different key for diff rounds.

→ Because, inorder to secure the algorithm.
→ The third party get the key corrupt the algorithm. So, we use different keys for different rounds.

BLOCK CIPHER MODES OF OPERATION:

Dividing PT into diff no. of blocks is Block Cipher.

And when we are doing operation on that 1 block cipher we have diff modes. Totally we have 5 modes.

1) Electronic Code Book (ECB):-

1. First we will do PT into no. of blocks.
2. We will be encrypting the PT with the help of a key.
3. In order to Encrypt we need PT & Key.
4. In order to decrypt we need CT & Key.
5. PT & key is given as input

DES :- (Data Encryption Standard)

The DES is a symmetric key algorithm for the encryption of digital data.

Developed in the early 1970's by IBM and was published as an official Federal Information Processing standard in 1976.

→ In our algorithm we take 64 bit key.

Key = 0001001100110100 0101011101111001 1011110111011000

This is 64 bit key.

↳ The 64-bit key is permuted accordingly to the following table, PC-1.

↳ In this as per rule only 56 bits of the original key appear in the permuted key.

PC-1 Table.

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

As per table we should arrange the key values.

→ 57th bit in key will be our 1st element.

56 bit key = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

→ In table 7 rows and 8 columns are there.

→ 7×8 = 56 bit ,

The no. of bits in o/p that is equal to the no. of digits in the pc1 table.

→ After getting 56 bit key then Next, Split this key into left and right halves, C_0 and D_0, where each half has 28 bits.

from the permuted key, we get.

C_0 = 1111000 0110011 0010101 0101111 Added

D_0 = 1010101 1011001 1001111 0001111 Added

Iteration Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of Left shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2

16
1

By Applying this Left shift on C_0, D_0 it becomes C_1 & D_1 as the Left shift will be done.

C_1 = 1110000 1100110 0101010 1111

D_1 = 1010101 0110011 0011110 0011110

→ After getting upto C_{16} to D_{16}.

Next, Combining. Of C_1 & D_1 we get again 56 bit. That 56 bit i/p is given to pc-2 table. After taking pc-2 table then according to pc-2 table only 48 bit output will be come.

→ Each pair has 56 bits, but pc-2 only uses 48 of these.

C_1 = 1110000 1100110 0101010 1111

D_1 = 1010101 0110011 0011110 0011110

Together, Combinely we get 56 bit as.

C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

pc-2 table gives 48 bits.

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

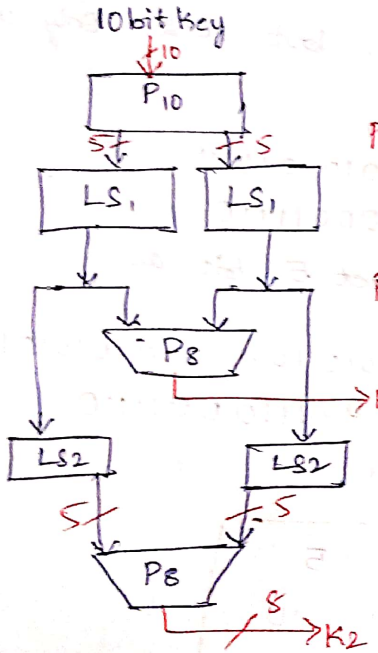
8×6 = 48

After Applying pc2 on C_1, D_1 we get o/p

K_1 = 0001101100000 001011 101111 000111 000001110010

SDES:-

Key Generation:-



IP →

1	2	3	4	5	6	7	8
2	6	3	1	4	8	5	7

P_{10}

1	2	3	4	5	6	7	8	9	10
3	5	2	7	4	10	1	9	8	6

P_8

1	2	3	4	5	6	7	8
6	3	7	4	8	5	10	9

IP^{-1}

1	2	3	4	5	6	7	8
4	1	3	5	7	2	8	6

E/P

1	2	3	4	5	6	7	8
4	1	2	3	2	3	4	1

P_4

1	2	3	4
2	4	3	1

In binary forms,

$$S_0 = \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}$$

Key Explanation:-

Key

1010 010 0010 → 10 bit key

01000 00100 → LS1 we divide 5 bits & 5 bits. We will apply One numbers to Left Shift.

K_1 → 00001000 → After applying P_8 10 becomes key will become '8' bits. This is Key 'K_1' = 00001000

After generating Key 'K_1' we will do Left Shift with two numbers. Again we will take 10 bit key and do Left shift again with 2 numbers

00000 10000

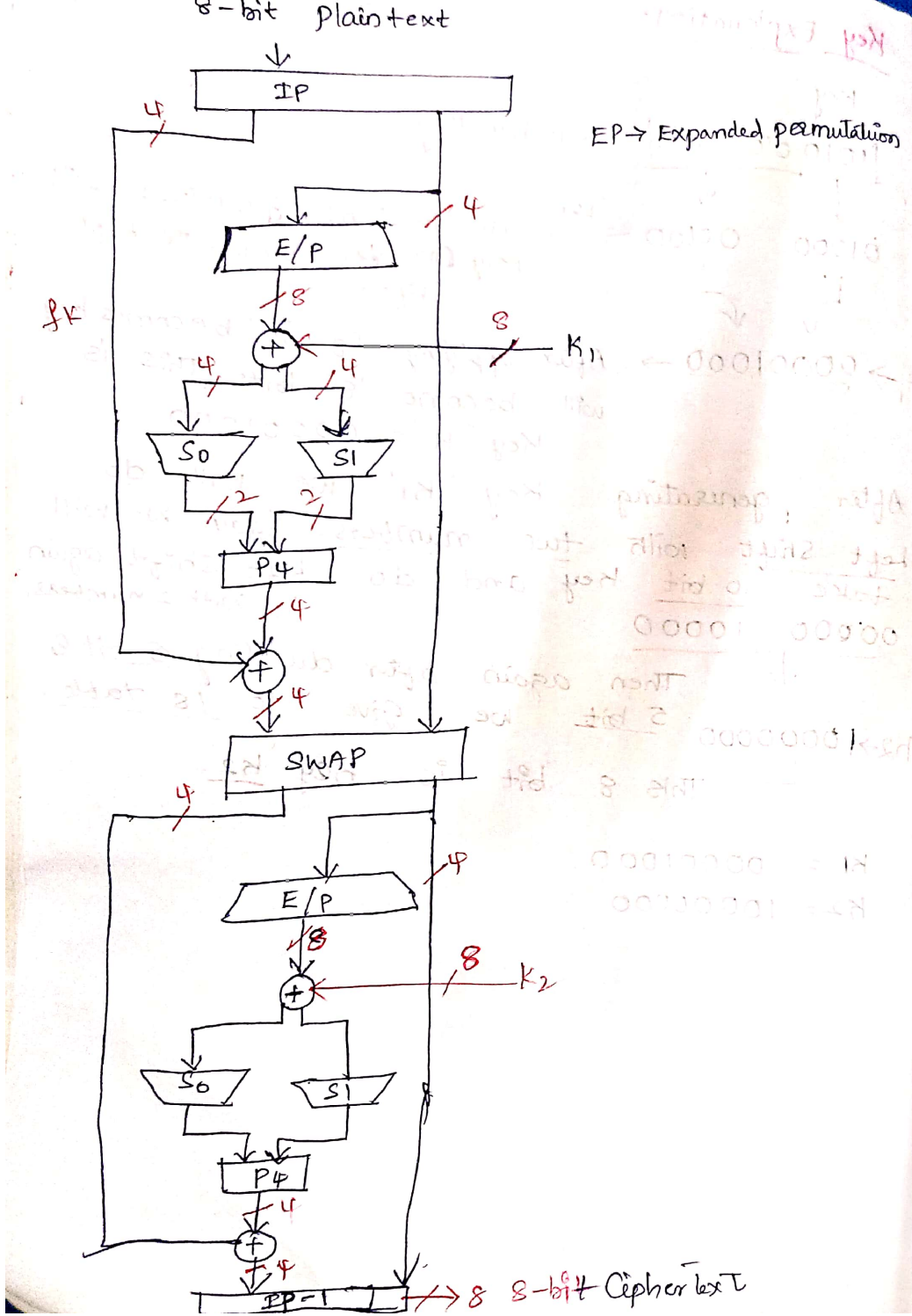
Then again after dividing 5 bit & 5 bit we give to P_8 table.

K_2 → 10000000

This 8 bit is key 'K_2'.

K_1 = 00001000

K_2 = 10000000



EP → Expanded permutation

8-bit plain text = 11001000 we divide ... 4 bits.

This 8 bit gives to IP table. After giving 8 bit to IP table.

L R
10010010 We will take right 4 bits and we will give to EP table.

0010 will be applied → E/P table → After giving right 4 bits to E/P table then we get,

00010100 → E/P table after giving 4 bits to E/P then
⊕ 00001000 → K1 we take K1 to do XOR operation
00011100
S0 S1

S0 → 0001 → 1st & last bit is for row.
S1 → 1100

2^2 2^1 2^0
4 2 1
0 1 → 1

S0 → row → 01 = 1st row } 3 = 11
S0 → Column → 00 = 0th column } After doing decimal to binary conversion we get. 4 2 1, 0 0 → 0

S1 → row → 10 = 2 } 1 → 01
S1 → Column → 10 = 2

1101 → This 4 bits we will give to P4 table

1101 → After giving to P4 table then is from left part

1101
⊕ 1001 → This is 4 bits after giving to IP table.
0100 0010 — These are right bits of after IP.

then we should swap right to left, left to right the bits will be swapped.

R
0010 0100 → we should give this to E/-

00101000

~~00010100~~ → After giving to E/p table.

XOR ⊕ 10000000 — ~~(K2)~~ Only right bits will give to E/p

00101000 → After E/p

⊕ 10000000 ⇒ K2

00101000

10101000

S0 S1

S0 S1

S_0 = 1010

S_1 ⇒ 1000

3	2	1	0
2	2	2	2
8	4	2	1
		1	1

8 4 2 1

0 0

0 0

S_0 = row = 10 ⇒ 2^{row} } 2^{col} 2 = 10

S_0 = Column ⇒ 01 ⇒ 1

S_1 = row = 10 ⇒ 2^{row} } 2^{column} 3 = 11

S_1 = Column = 00 ⇒ 0

1011 → This will give to P4 table.

0 1 1 1

⊕ 0 0 1 0

0 1 0 1 0 1 0 0

L R

These are right bits from after swapping

After we should do IP^{-1}

This is 10000101 the 8-bit Ciphertext

CT = 10000101