Unit 2

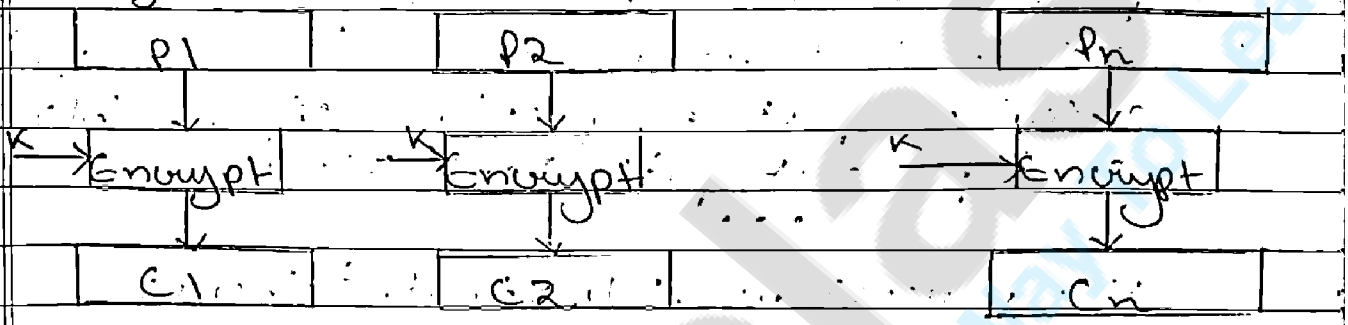
Q) Block cipher modes of Operation / Algorithm modes used for secret key cryptography. Encryption algorithms are divided into two categories based on input type, as block cipher and stream cipher. Block cipher is an encryption algo algorithm which takes fixed size of input say b bits and produces a ciphertext of b bits again. If input is larger than b bits it can be divided further. For different applications and uses, there are several modes of operations for a block cipher.

Electronic Code Book (ECB) -
Electronic code book is the easiest block cipher mode of functioning. It is easier because of direct encryption of each block of input plain text and output is in form of blocks of encrypted cipher text. Generally,

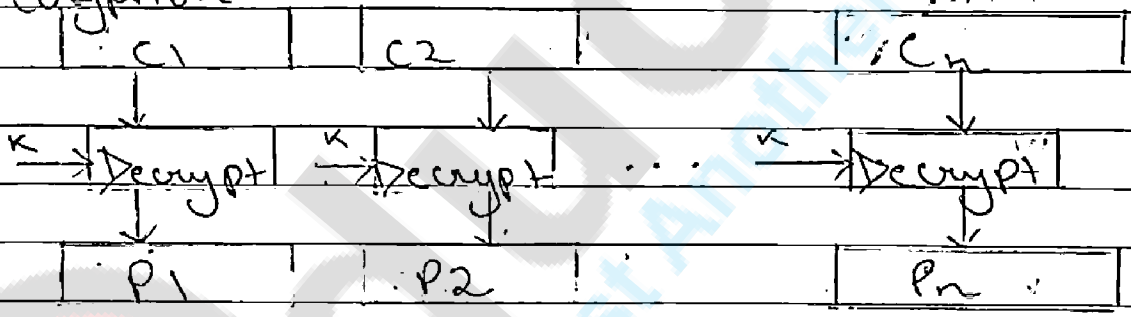
If a message is larger than b bits in size, it can be broken down into bunch of blocks and the procedure is repeated.

Procedure of ECB is illustrated below.

Encryption



Decryption



Advantages of using ECB -

1) Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
2) Simple way of block cipher.

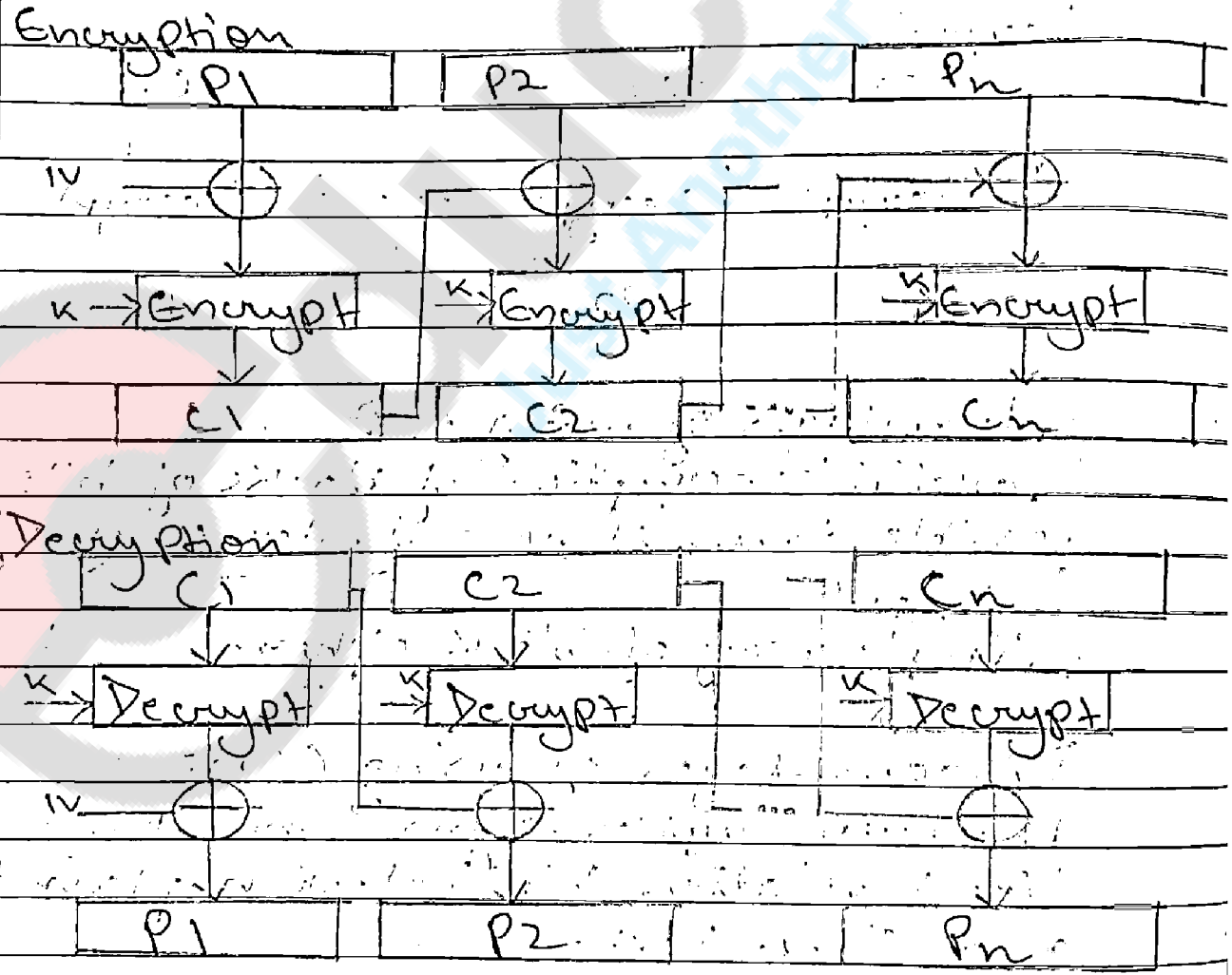
Disadvantages of using ECB -

Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

Cipher Block Chaining:-

Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements. In CBC, previous cipher block is given as input to next encryption algorithm after XOR with original plaintext block. In a nutshell here, a cipher block is produced by encrypting a XOR output of previous cipher block and present plaintext block.

The process is illustrated here:



Advantages of CBC-

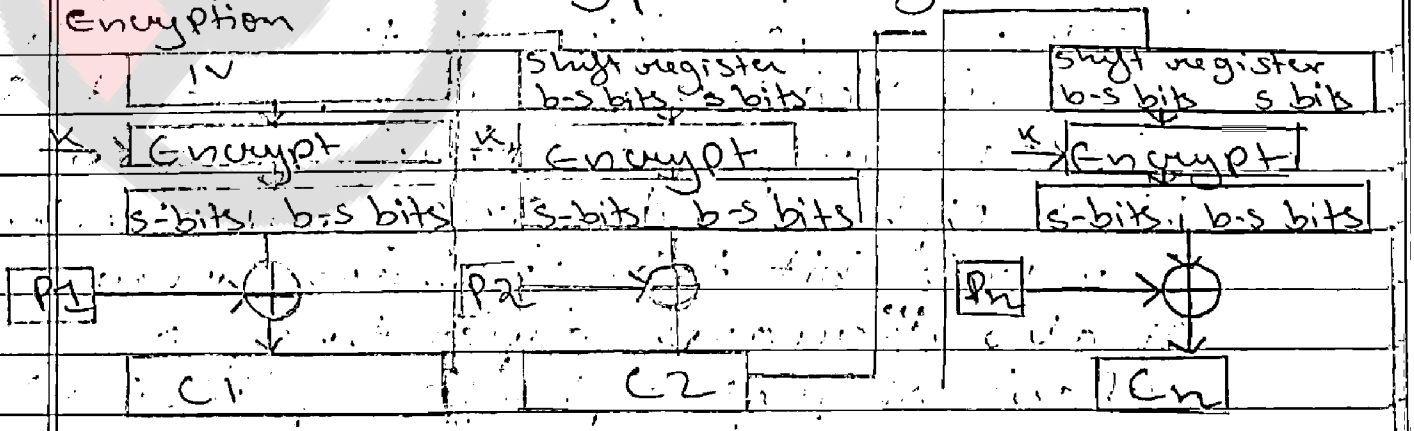
1) CBC works well for input greater than b bits.
2) CBC is a good authentication mechanism.
3) Better resistive nature towards cryptanalysis than ECB.

Disadvantages of CBC:-

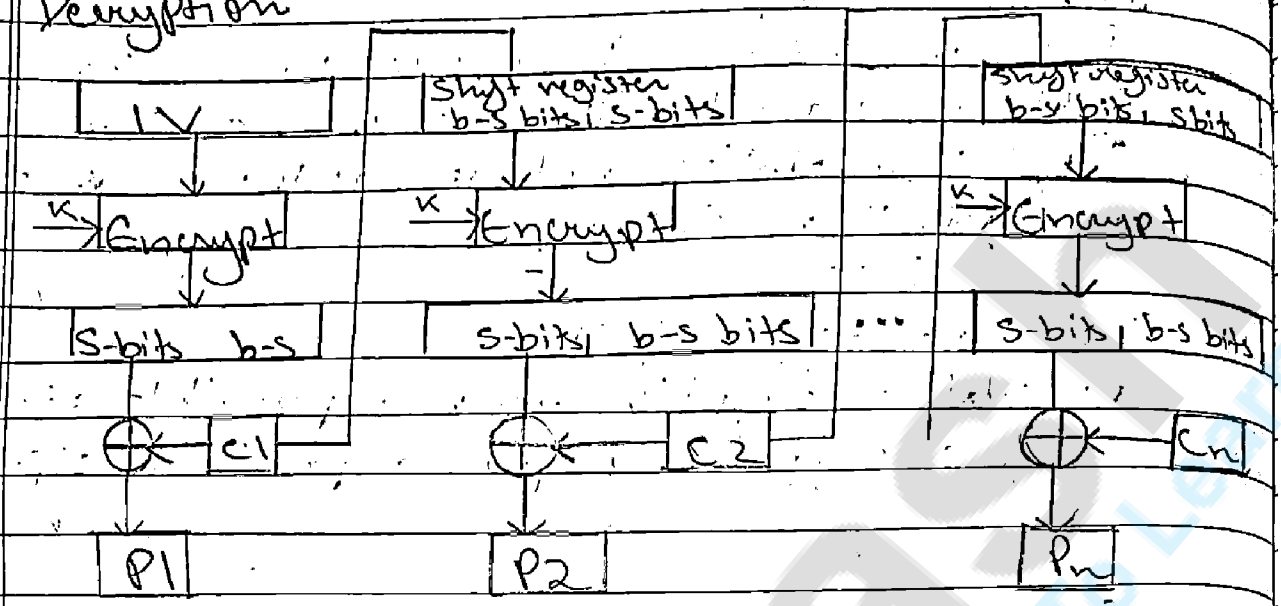
Parallel encryption is not possible since every encryption requires previous cipher.

Cipher Feedback Mode (CFB):-

In this mode the cipher is given as feedback to the next block of encryption with some new specifications. first an initial vector IV is used for first encryption and output bits are divided as set of s and b-s bits, The left hand side bits are selected and are applied an XOR operation with plain text bits. The result given as input to a shift register and the process continues. The encryption and decryption process for the same is shown in the diagram, both of them use encryption algorithm.



Decryption



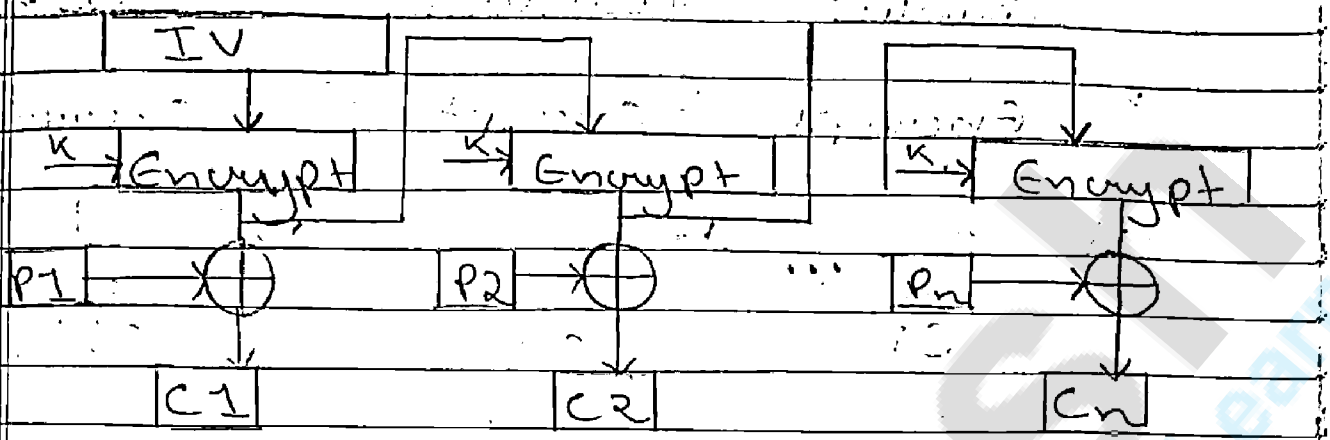
Advantages of CFB

Since, there is some data loss due to use of shift register, thus it is difficult for applying eg cryptanalysis.

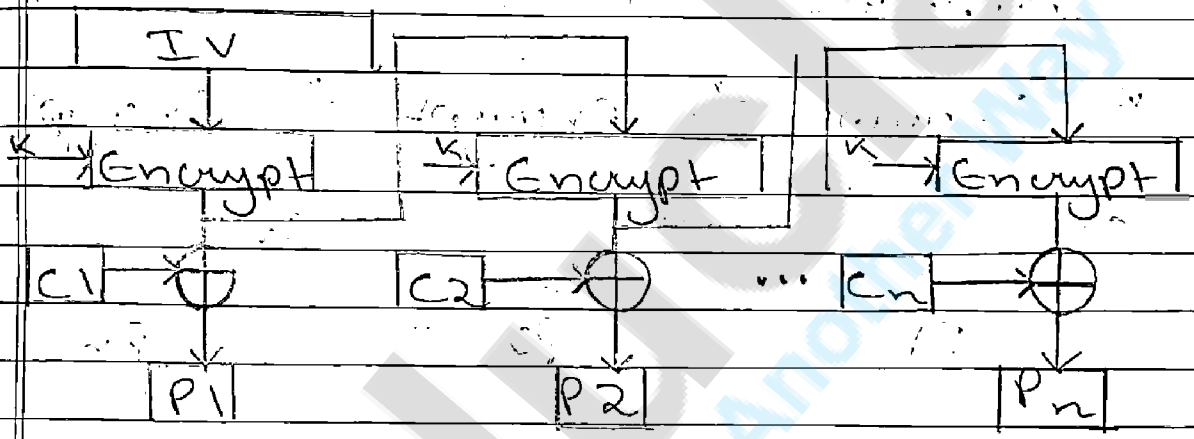
Output feedback Mode:-

The output feedback mode follows nearly same process as the Cipher feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output. In this output feedback mode, all bits of the block are send instead of sending selected S bits. The Output feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases dependency or relationship of cipher on plain text.

Encryption



Decryption

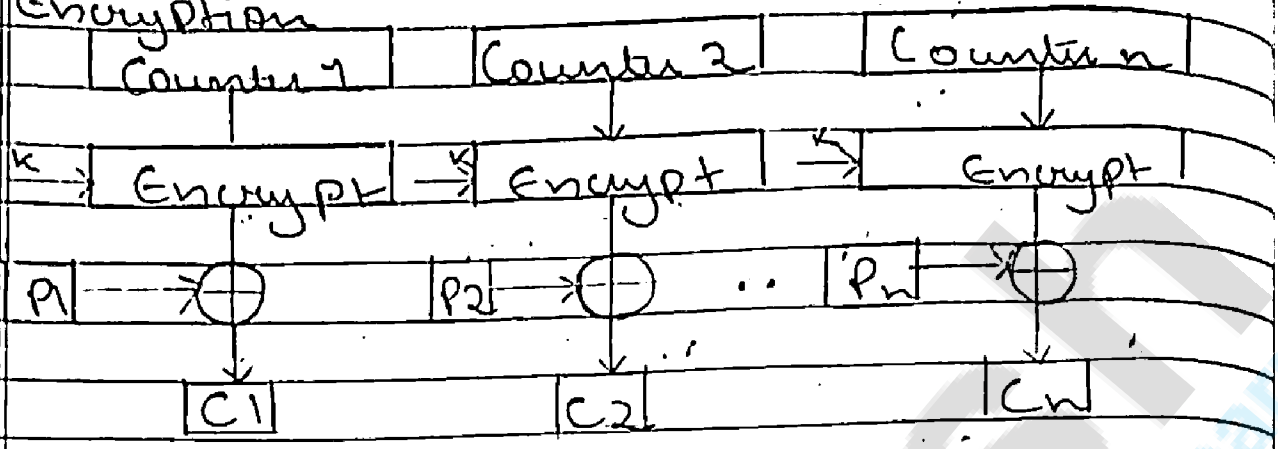


Counter Mode -

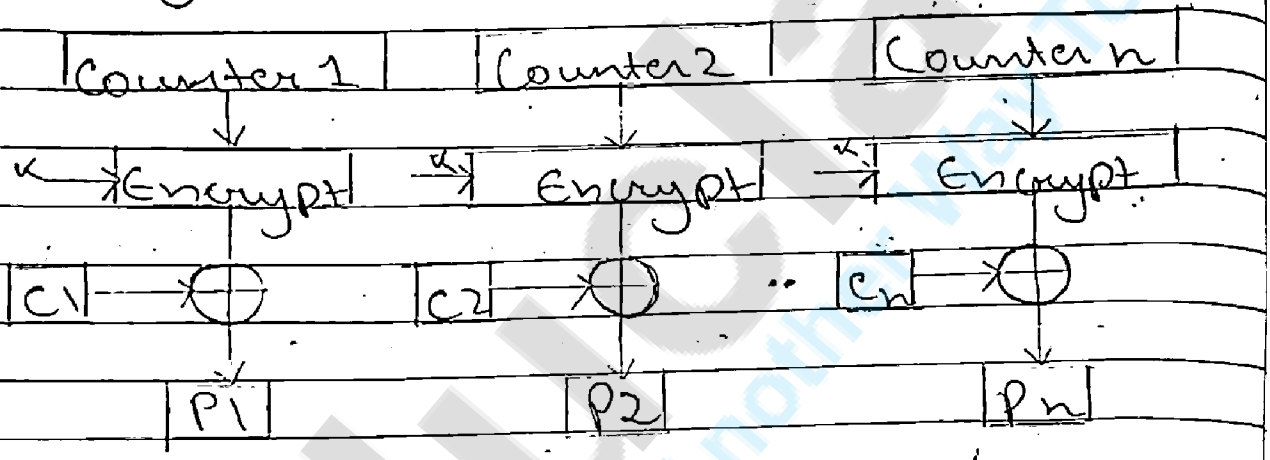
The Counter Mode or CTR is a simple counter based block cipher implementation. Every time a counter ~~its~~ initiated value is encrypted and given as input to XOR with plaintext which results in cipher-text block. The CTR mode is independent of feedback use and thus can be implemented in parallel.

Its simple implementation is shown in the diagram

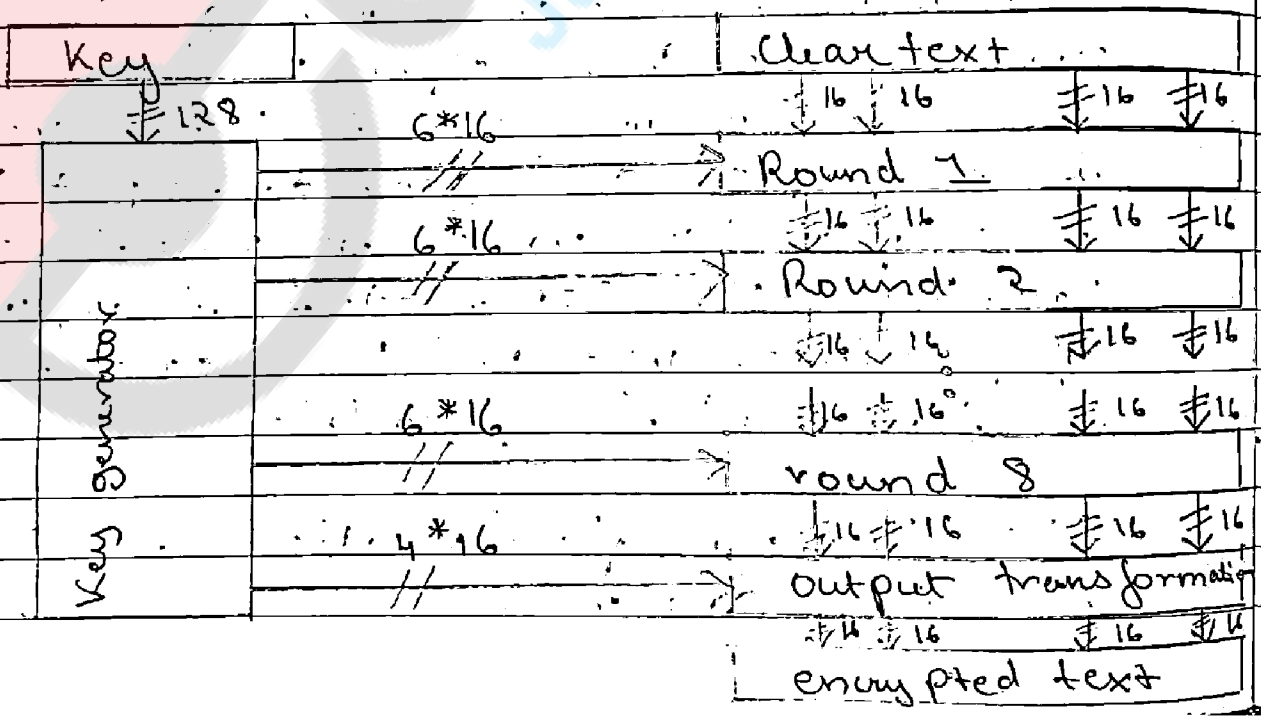
Encryption



Decryption



Q] IDEA Algorithm



IDEA - International Data Encryption Algorithm. It is patent protected to prevent fraud and privacy. It was meant to be replacement for the Data Encryption Standard. It is considered among the best known publicly algorithms. It is a block cipher that takes input of 64 bit and key used is of 128 bit from which we derive 52 subkeys that is used in the algorithm.
Idea perform 8 identical round for encryption in which 6 different subkeys are used and at last four keys are used for output transformation. IDEA was used in Pretty Good Privacy.

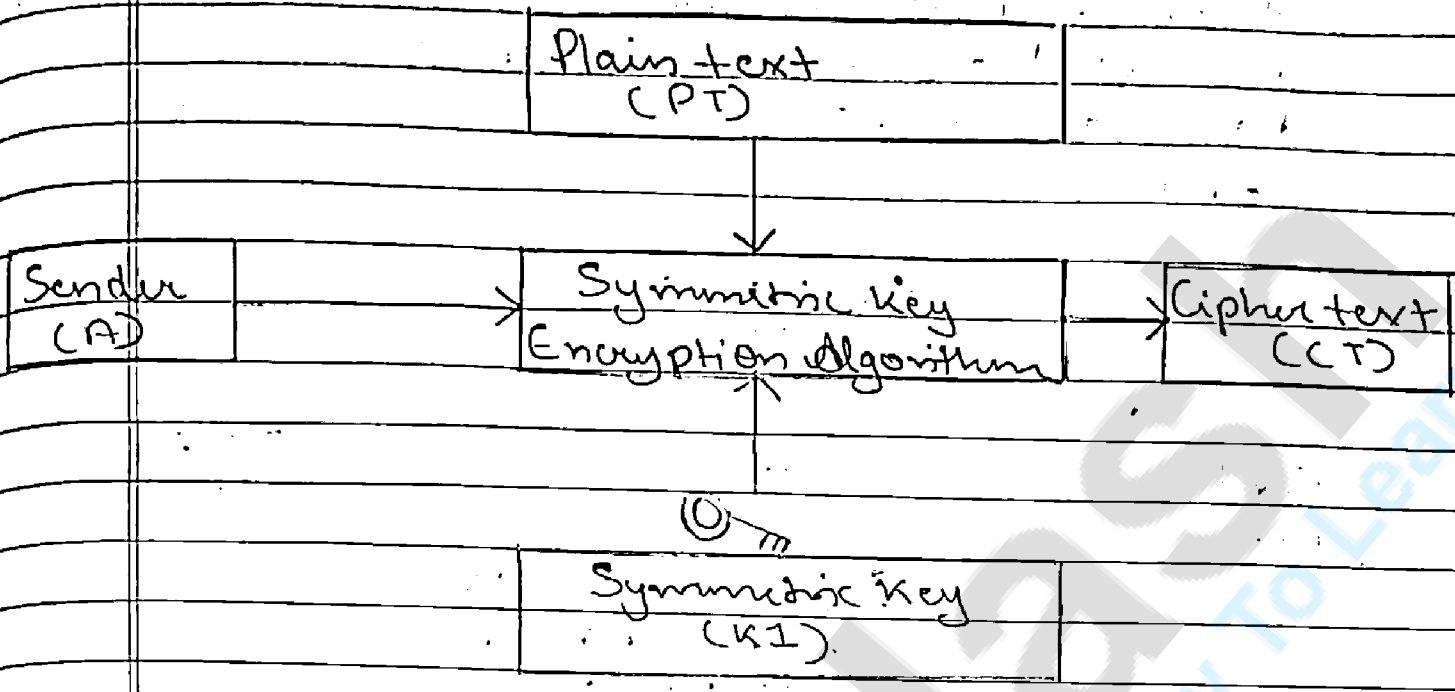
The process performed in each rounds are:
(PT1, PT2, PT3 and PT4 are the four parts (block) of plain text that is an input for the rounds of IDEA.
Each PT is of 16-bit that in total 64-bit of plain text).
1) Multiply PT1 and Key1.
2) Add PT2 and Key2.
3) Add PT3 and Key3.
4) Multiply PT4 and Key K4
5) result of Step1 XOR result of Step 3
6) result of Step2 XOR result of Step4
7) Multiply step 5 with Key5
8) Add result of step 6 and step 7
9) Multiply result of step 8 with Key 6.
10) Add result of step 7, and step 9.
11) result of step 1 XOR result of Step 9.
12) result of step 3 XOR result of Step 9.

13) result of step 2 XOR result of step 10.
14) result of step 4 XOR result of step 10.

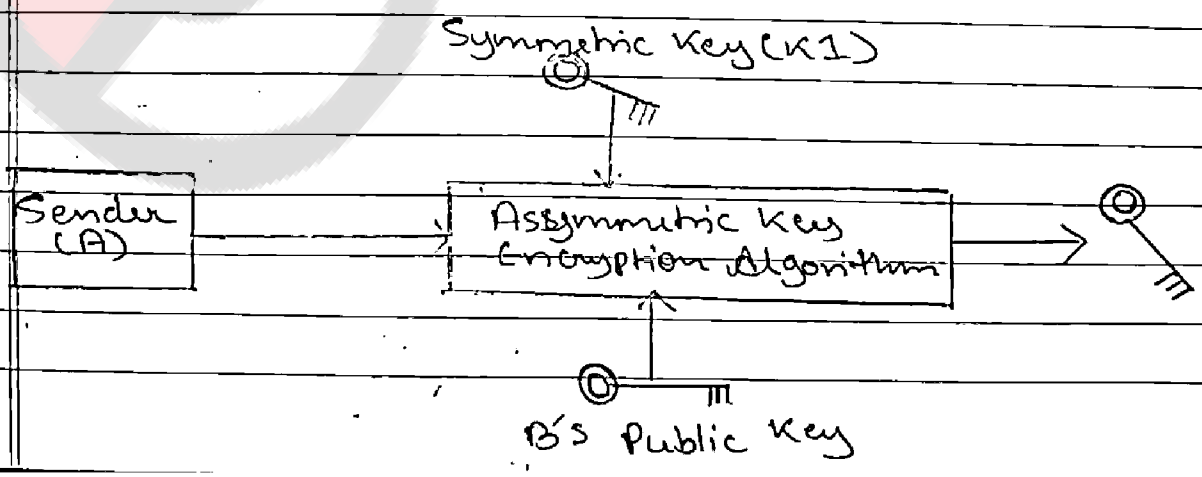
Output transformation:
1) Multiply R1 with Key 49.
2) Add R2 and Key 50.
3) Add R3 and Key 51.
4) Multiply R4 and Key 52.
(R1, R2, R3 and R4 are the four blocks (part) plain text after the complete 8 rounds, which are the inputs for the output transformation resulting in cipher text.)

Q) Symmetric and Asymmetric Key cryptography together.
Indeed in practice, symmetric-key cryptography and asymmetric-key cryptography are combined to have a very efficient security solution. The way it works is as follows, assuming that A is the sender of message and B is its receiver.

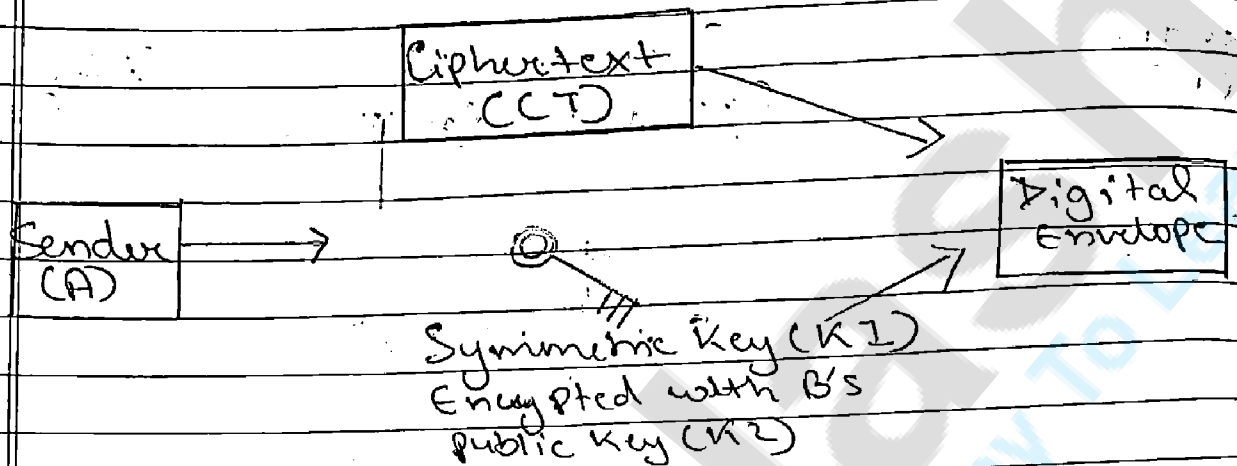
1) A's computer encrypts the original plain-text message (PT) with the help of a standard symmetric key cryptography algorithm, such as DES, IDEA etc. This produces a cipher-text message (CT) as shown in diagram. The key used in this operation (K1) is called one-time symmetric key, as it is used once and then discarded.



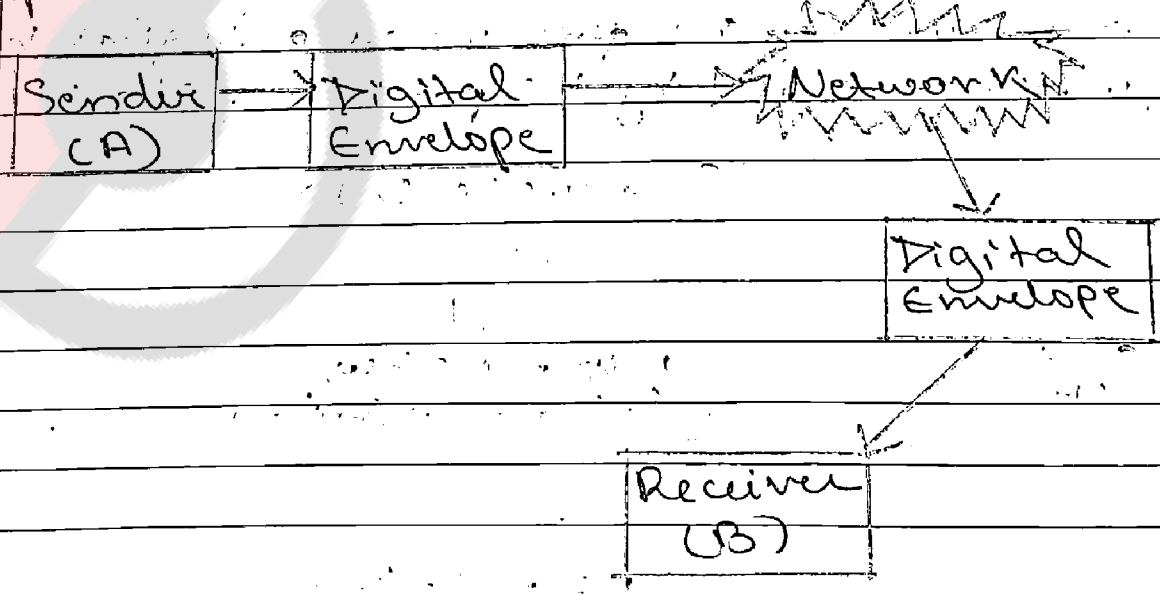
2) We have encrypted the plain text (PT) with a symmetric-key operation. We must now transport this one-time symmetric key (K1) to the server so that the server can decrypt the ciphertext (CT) to get back the original plain-text message (PT). It now takes the one-time symmetric key of step 1 (i.e K1), and encrypts K1 with B's public key (K2). This process is called key-wrapping of the symmetric key, and is shown in diagram below. We have shown that the symmetric key K1 goes inside a logical box, which is sealed by B's public key (i.e K2)



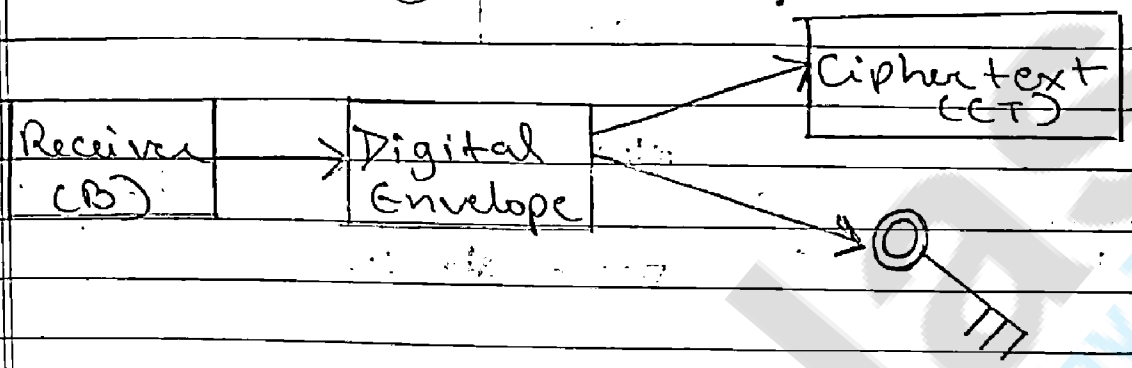
3) Now, A puts the cipher text (CT) and the encrypted symmetric key together inside a digital envelope. This is shown in diagram below.



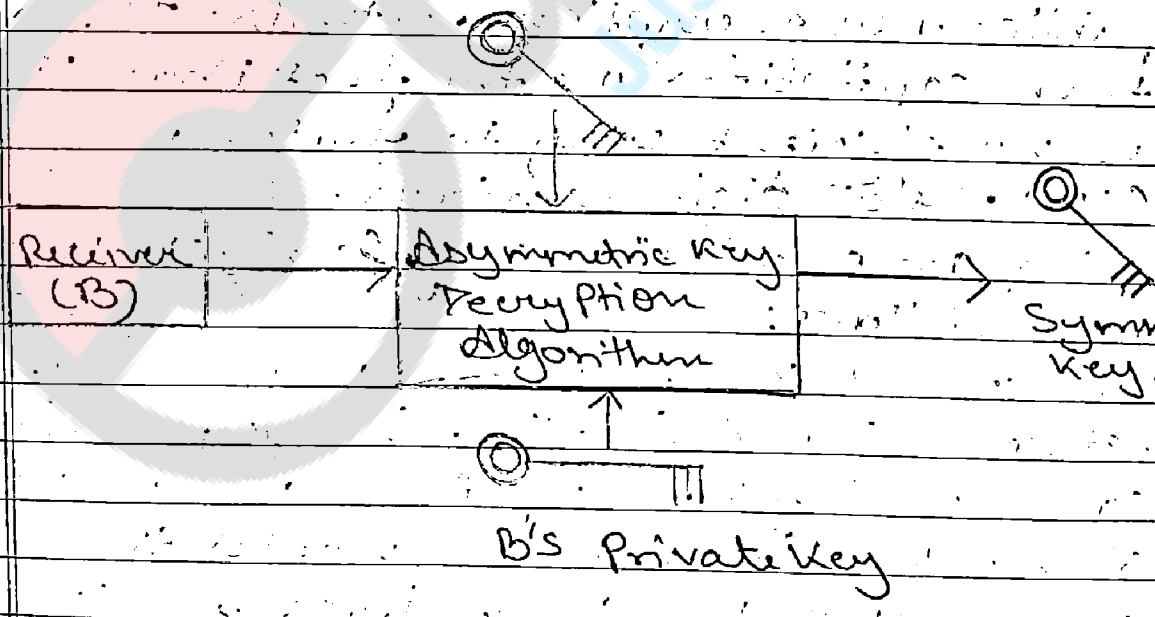
4) The sender(A) now sends the digital envelope [which contains the cipher text (CT) and the ~~and~~ onetime symmetric key (K1) encrypted with B's public key, (K2)] to B using the underlying transport mechanism (network). This is shown in diagram. We do not show the contents of the envelope, and assume that the envelope contains the two entities.



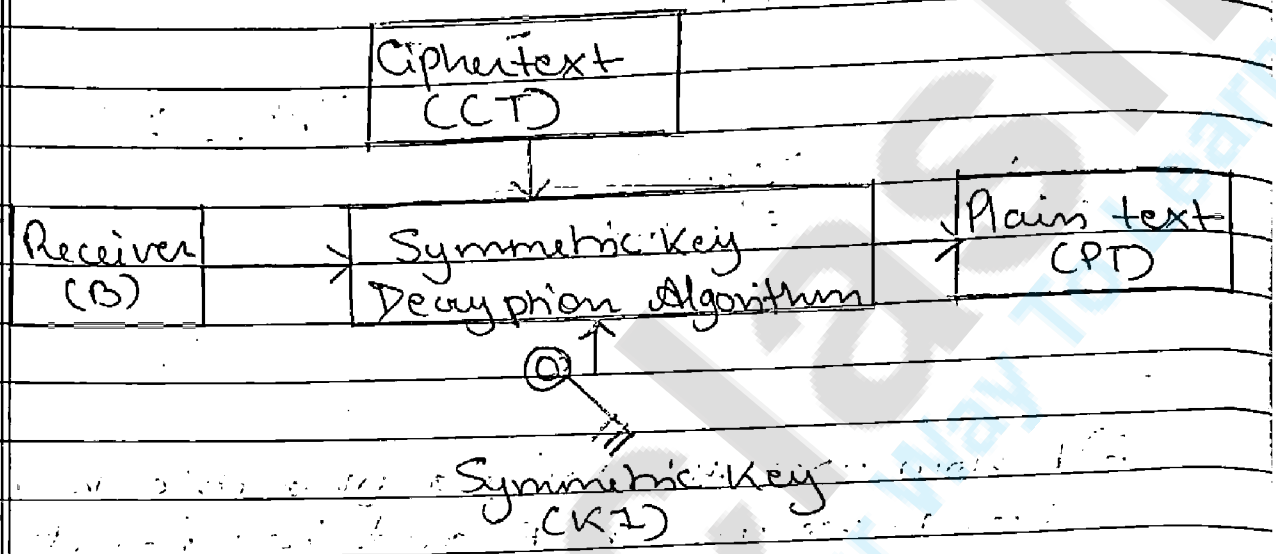
5) B receives digital envelope and opens it. After B opens this digital envelope, he gets 2 things, first is cipher text (CT) and another one is the one-time session key (K1) which is encrypted using B's public key (K2). This is shown in diagram below.



6) B now uses the same asymmetric-key algorithm as we used by A and her private key (K3) to decrypt (i.e open up) the logical box that contains the symmetric key (K1), which was encrypted with B's public key (K2). This is shown in diagram below. This the output of the process is the one-time symmetric key K1.



4) Finally, B applies the same symmetric-key algorithm as was used by A, and the symmetric key KI to decrypt the cipher text (CT). This process yields the original plain text (PT) as shown in diagram



Q] RSA

The Rivest-Shamir-Adleman (RSA) algorithm is one of the most popular and secure public-key encryption methods. The algorithm capitalizes on the fact that there is no efficient way to factor very large (100-200) digit numbers.

Using an encryption key (e, n), the algorithm is as follows:

i) Represent the message as an integer between 0 and (n-1). Large messages can be broken up into a number of blocks. Each block would then be represented by an integer in the same range.

2) Encrypt the message by raising it to the eth power modulo n. The result is a ciphertext message C.
3) To decrypt ciphertext message C, raise it to another power d modulo n.
The encryption key (e, n) is made public.
The decryption key (d, n) is kept private by the user.

How to determine appropriate values for e, d and n
1) Choose two very large (100+ digit) prime numbers. Denote these numbers as p and q.
2) Set n equal to p * q.
3) Choose any large integer, d, such that GCD (d, ((p-1) * (q-1))) = 1
4) Find e such that e * d = 1 (mod ((p-1) * (q-1)))

Advantages of RSA
1) Very fast, very simple encryption and verification.
2) Easier to implement than Elliptical curve cryptography (ECC)
3) Easier to understand
4) Widely deployed, better industry support.

Disadvantages of RSA
1) Very slow key generation.
2) Slow decryption, which is slightly tricky to implement securely.

3) Two-part key is vulnerable to CCA attack if poorly implemented.

To this day the RSA together with the AES algorithm is the mostly used algorithm in commercial systems.
It is used:
- to protect web traffic, in the SSL protocol (Security Socket layer).
- to guarantee email privacy and authenticity in PGP (Pretty Good Privacy)
- to guarantee remote connection in SSH (Secure Shell).
- Furthermore it plays an important role in the modern payment systems through SET protocol (Secure Electronic Transaction).
RSA has been used in most digital data, information and telephone security applications.

Security
Possible approaches to attacking RSA are:
- brute force key search - infeasible given size of numbers
- mathematical attacks - based on difficulty of computing $\phi(n)$, by factoring modulus n
- timing attacks - on running of decryption
- chosen ciphertext attacks - given properties of RSA

classmate
Date
Page

1) Secret key cryptography

With secret key cryptography, a single key is used for both encryption and decryption.

The sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver.

The receiver applies the same key (or ruleset) to decrypt the message and recover the plaintext.

Because a single key is used for both functions, secret key cryptography is also called symmetric encryption.

A symmetric encryption scheme has five ingredients

1) Plaintext :- This is the original intelligible message or data that is fed into the algorithm as input.

2) Encryption algorithm :- The encryption algorithm performs various substitutions and transformations on the plaintext.

3) Secret key :- The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm

depend on the key.

4) Ciphertext - This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.

5) Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that in fact, is the secret. The biggest difficulty with this approach, of course, is the distribution of the key.

Requirements

Two requirements for secure use of symmetric encryption:

- a strong encryption algorithm
- a secret key known only to sender/receiver

Mathematically have:

$$Y = E(K, X)$$

$$X = D(K, Y)$$

Implies a secure channel to distribute

Key

Secret Key cryptography schemes

Secret key cryptography schemes are generally categorized as being either stream ciphers or block ciphers.

Stream ciphers operate on a single bit at a time and implement some form of feedback mechanism so that the key is constantly changing.

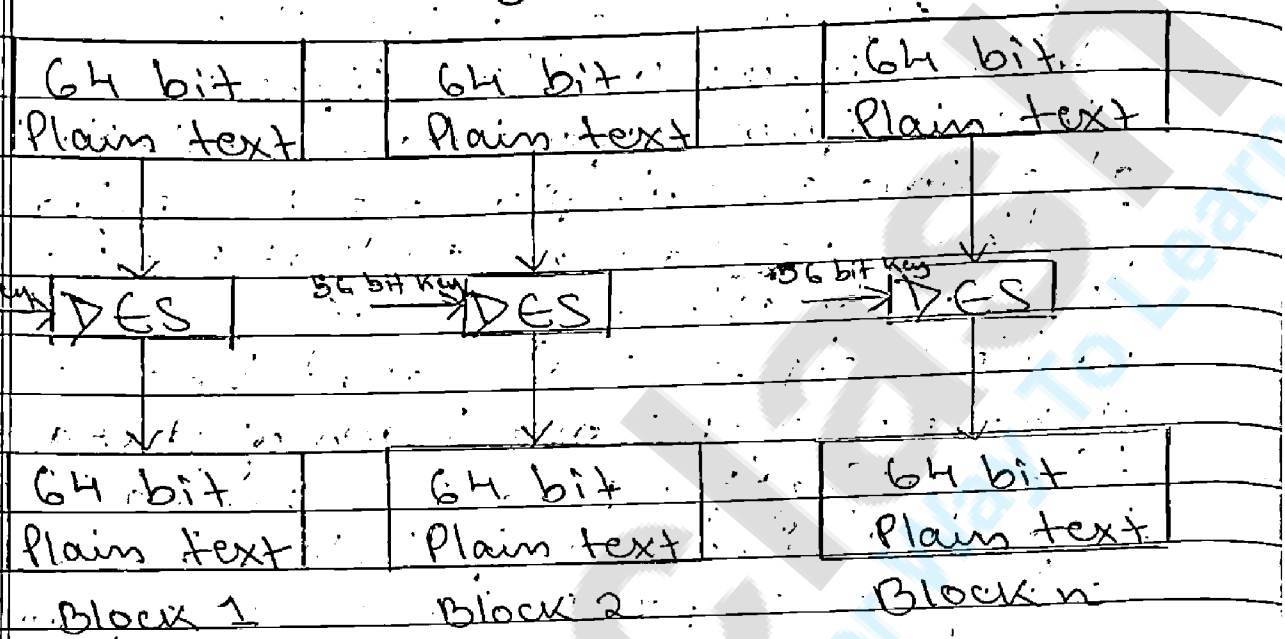
A block cipher is so-called because the scheme encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

(J) Data Encryption Standard (DES)

Data encryption standard (DES) has been found vulnerable against very powerful attacks and therefore, the popularity of DES has been found slightly on decline.

DES is a block cipher and encrypts data in blocks of size of 64 bit each, means 64 bits of plain text goes as the input to DES, which produces 64 bits of cipher text. The same algorithm and

Key are used for encryption and decryption, with minor differences. The key length is 56 bits. The basic idea is shown in diagram.



We have mentioned that DES uses a 56 bit key. Actually, the initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56 bit key. That is, bit position 8, 16, 24, 32, 40, 48, 56 and 64 are discarded.

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

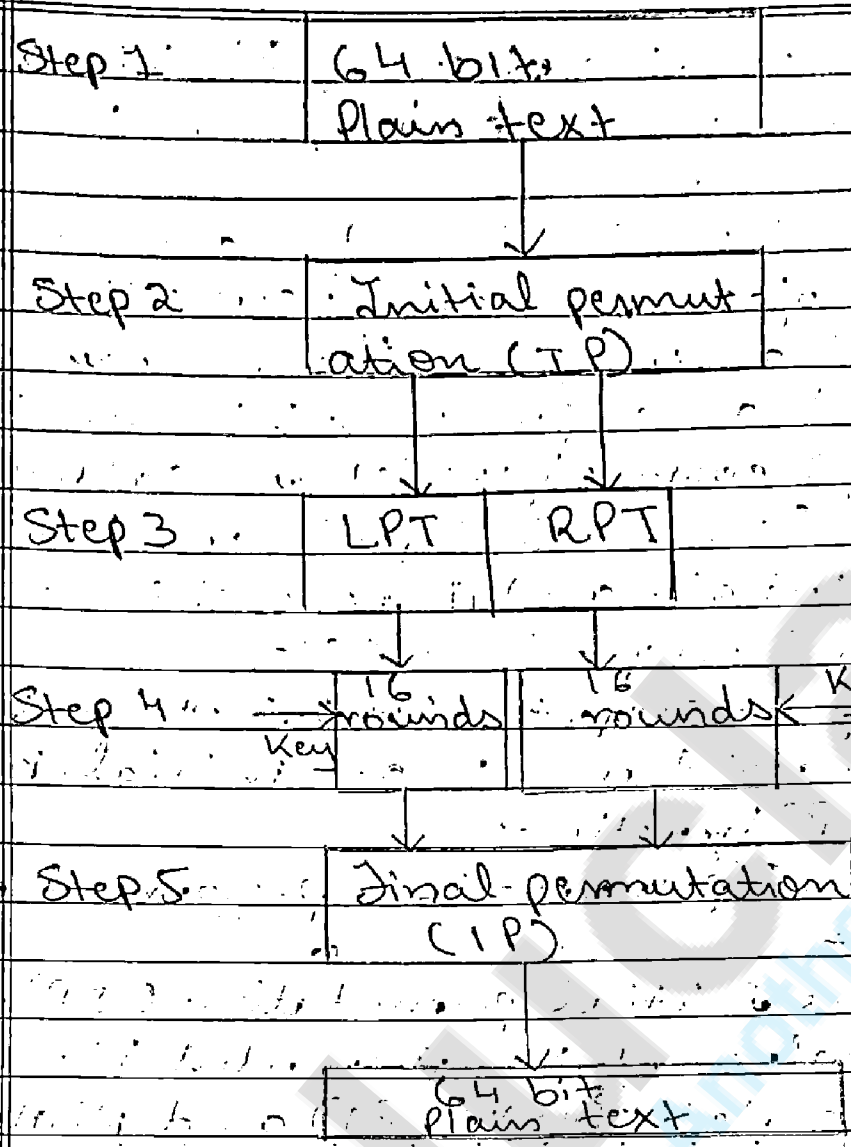
discarding of every 8th bit of original key

Thus, the discarding of every 8th bit of the key produces a 56-bit key from the original 64-bit key.

DES is based on the two fundamental attributes of cryptography: substitution (also called as confusion) and transposition (also called as diffusion). DES consists of 16 steps, each of which is called as a round. Each round performs the steps of substitution and transposition.

Broad-level steps in DES:-

1) In the first step, the 64 bit plain text block is handed over to an initial Permutation (IP) function.
2) The initial permutation performed on plain text.
3) Next the initial permutation (IP) produces two halves of the permuted block; says left Plain Text (LPT) and Right Plain Text (RPT).
4) Now each LPT and RPT to go through 16 rounds of encryption process.
5) In the end, LPT and RPT are rejoined and a final Permutation (FP) is performed on the combined block.
6) The result of this process produces 64 bit cipher text.



Initial Permutation (IP) –

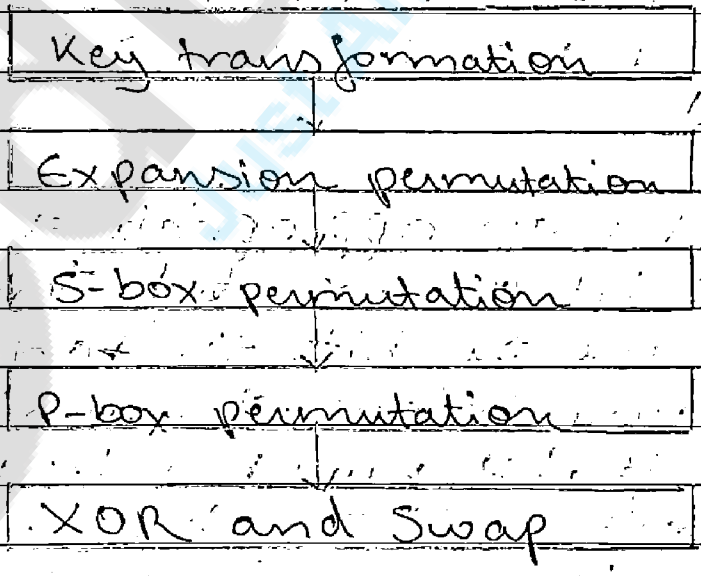
The initial permutation (IP) happens only once and it happens before the first round. It suggests how the transposition in IP should proceed. For example, it says that the IP replaces the first bit of the original plain text block with the 58th bit of the original plain text, the second bit with the 50th bit of the original plain text block and so on.. This is nothing but jugglery of

bit positions of the original plain text block. The same rule applies for all the other bit positions which shows in the diagram.

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	33	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Initial permutation table

As we have noted after IP done, the resulting 64-bit permuted text block is divided into two half blocks. Each half block consists of 32 bits, and each of the 16 rounds in turn consists of the broad level steps outlined in diagram.



Step 1: Key transformation -
We have noted initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the

initial key. Thus, for each a 56-bit key is available. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called as key transformation. For this the 56 bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round.

For example, if the round number 1, 2, 9 or 16 the shift is done by only position for other rounds, the circular shift is done by two positions. The number of key bits shifted per round is shown in figure.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# Key bits shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

After an appropriate shift 48 of the 56 bit are selected. For selecting 48 of the 56 bits the table shown in figure below. For instance, after the shift, bit number 14 moves on the first position; bit number 17 moves on the second position and so on. If we observe the table carefully, we will realize that it contains only 48 bit positions. Bit number 18 is discarded, like 7 others, to

reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as selection of a 48 bit subset of the original 56-bit key it is called Compression Permutation.

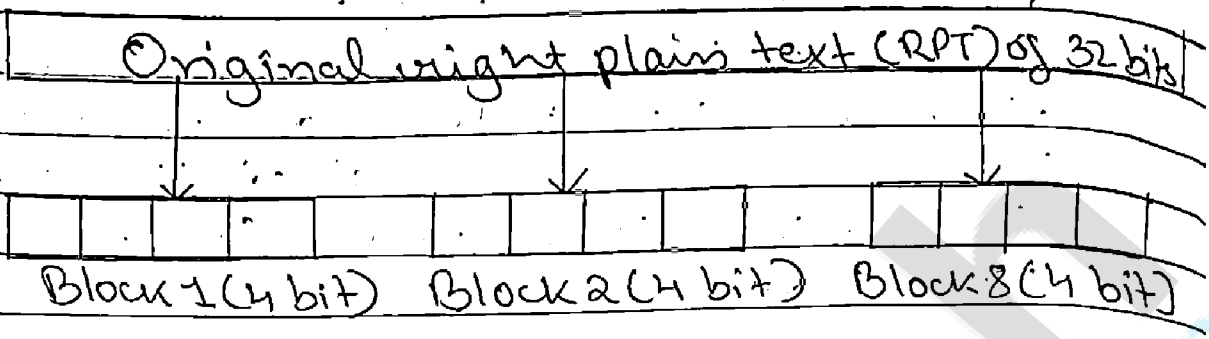
14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

compression permutation

Because of this compression permutation technique, a different subset of key bits is used in each round. That's make DES not easy to crack.

Step 2: Expansion Permutation:-

Recall that after initial permutation, we had two 32-bit plain text areas called as left Plain Text (LPT) and Right Plain Text (RPT). During the expansion permutation, the RPT is expanded from 32 bits to 48 bits. Bits are permuted as well hence called as expansion permutation. This happens as the 32 bit RPT is divided into 8 blocks, with each block consisting of 4 bits. Then, each 4 bit block of the previous step is then expanded to a corresponding 6 bit block, i.e. per 4 bit block, 2 more bits are added.



- division of 32 bit RPT into 8 bit blocks

This process results into expansion as well as permutation of the input bit while creating output. Key transformation process compresses the 56-bit key to 48 bits. Then the expansion permutation process expands the 32-bit RPT to 48-bits. Now the 48-bit key is XOR with 48-bit RPT and resulting output is given to the next step, which is the S-box substitution.

Problems with DES

1) Two chosen input to an S-box can create the same output.
2) The purpose of initial and final permutation is not clear.
3) DES fails in front of linear cryptanalysis, because during its design this attack wasn't invented.
4) Now in the age of parallel computing breaking DES has become

easy with the help of brute force attack which was impossible during that time.

Q) Variations of DES

Double DES

In this approach, we use two instances of DES ciphers for encryption and two instances of reverse ciphers for decryption.

Each instances use a different key.

- The size of the key is doubled.

There are issues of reduction to single stage.

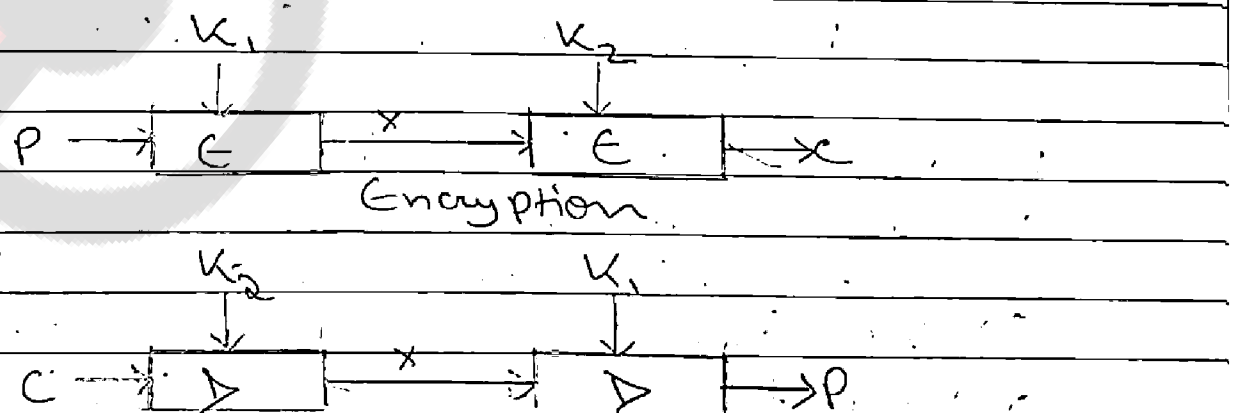
However, double DES is vulnerable to meet-in-the-middle attack

Given a plaintext P and two encryption keys K_1 and K_2, a ciphr text can be generated as,

$$C = E(K_2, E(K_1, P))$$

Decryption requires that the keys be applied in reverse order,

$$P = D(K_1, D(K_2, C))$$



Double encryption

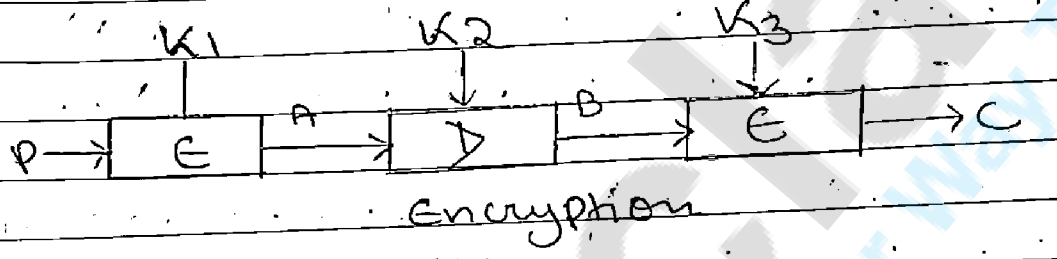
2) Triple DES with 2-key

Use three stages of DES for encryption and decryption.

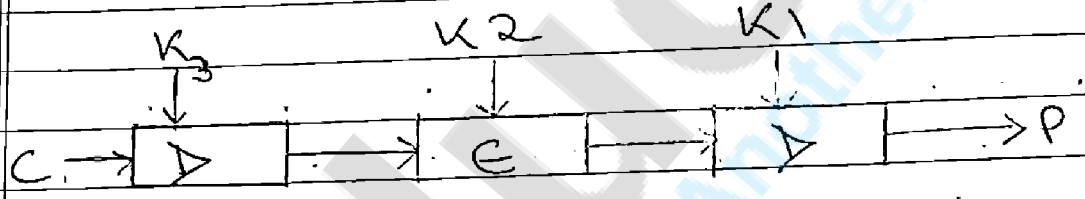
The 1st, 3rd stage use K_1 key and 2nd stage use K_2 key.

To make triple DES compatible with single DES, the middle stage uses decryption in the encryption side and encryption in the decryption side.

It's much stronger than double DES



Encryption



Decryption

The function follows an encrypt-decrypt ~~E D E~~ (E D E) sequence

$$C = E(K_1, D(K_2, E(K_1, P)))$$
$$P = D(K_1, E(K_2, D(K_1, C)))$$

By the use of triple DES with 2-key encryption, it raises the cost of meet-in-the-middle attack to 2^{112}

It has the drawback of requiring a key length of $56 \times 3 = 168$ bits

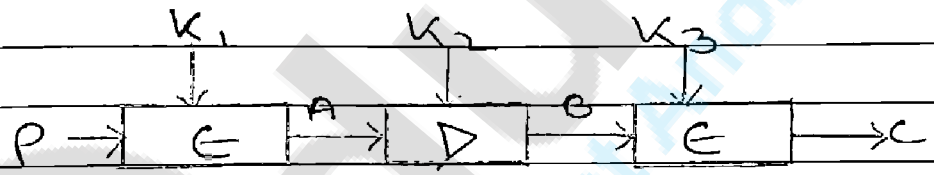
which may be somewhat unwieldy.

3) Triple DES with 3 key

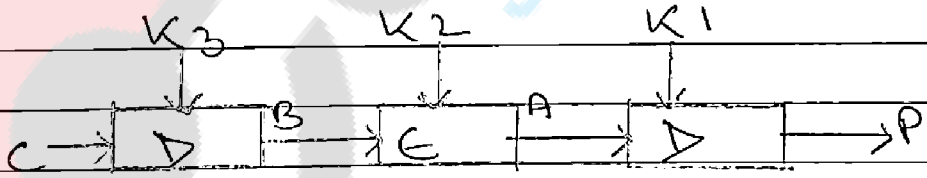
Although the attacks just described appear impractical, anyone using two-key 3DES may feel some concern. Thus, many researches now feel that 3-key 3DES is the preferred alternative. Use three stages of DES for encryption and decryption with three different keys.

3-key 3DES has an effective key length of 168 bits and is defined as,

$$C = E(K_3, D(K_2, E(K_1, P)))$$
$$P = D(K_1, E(K_2, D(K_3, C)))$$



Encryption



Decryption