

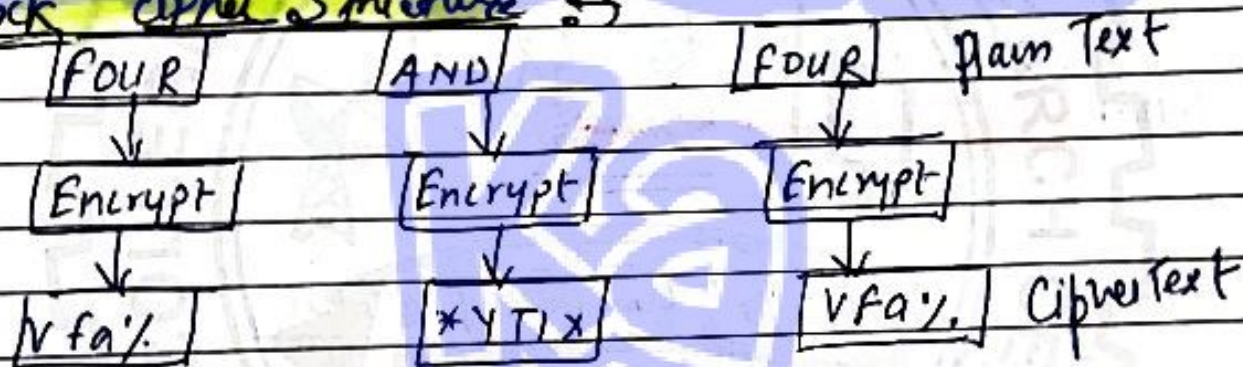
## Unit-2 . Block Cipher

Rather than encrypting one byte at a time, a block of bytes is encrypted at one go.

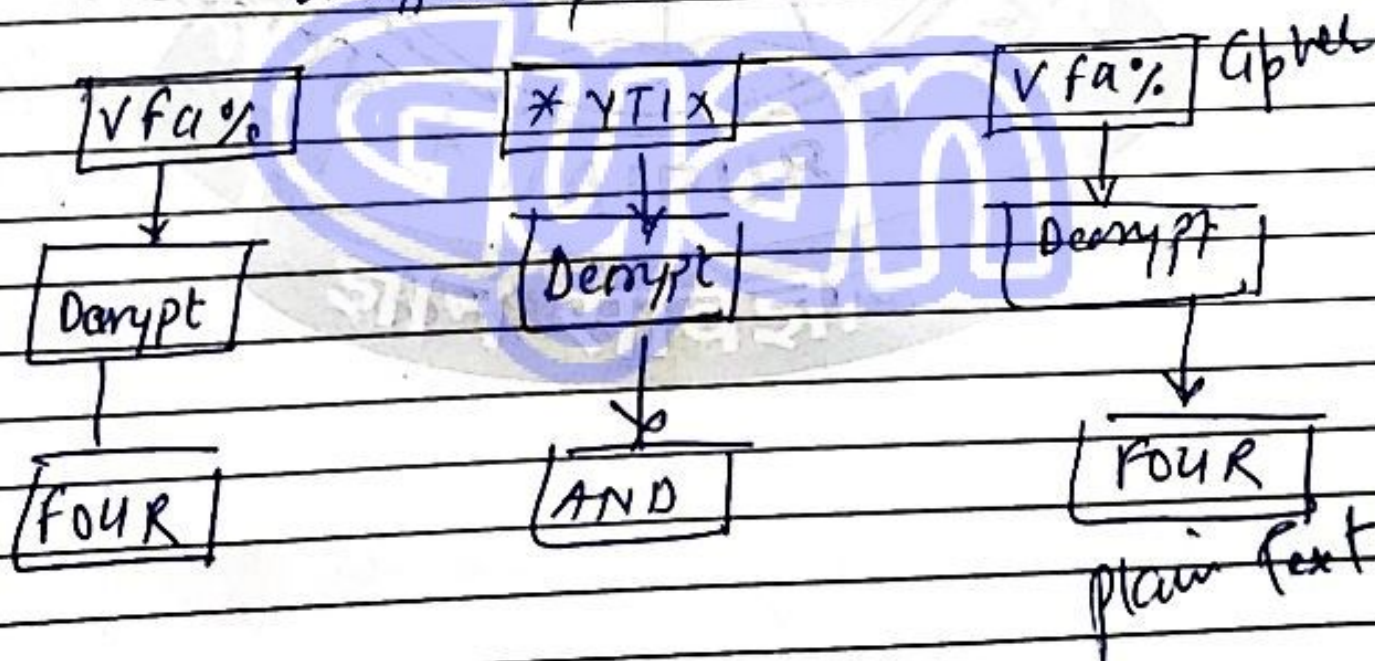
Suppose we have a plain text FOUR AND FOUR that need to be encrypted.

"Block Cipher techniques involves encryption of one block of text at a time. Decryption also takes one block of encrypted text at a time"

➤ Block cipher Structure ⇒



Encryption process





## 2.1 Data Encryption standard (DES)

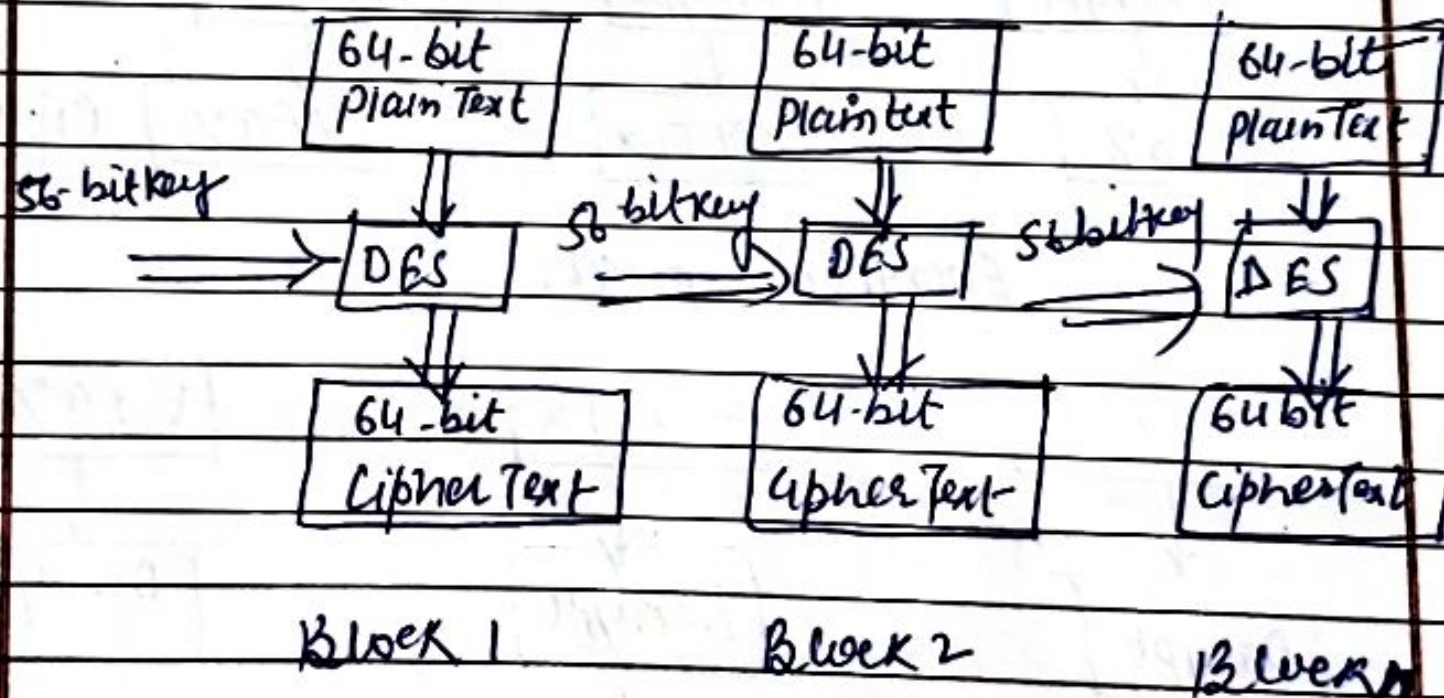
Data Encryption standard (DES) is also called as the data encryption algorithm.

DES is generally used in the ECB, CBC or the CFB mode.

The Origin of DES go back to 1972 when in the US the National bureau of standard now known as the National Institute of standard and Technology.

## 2.2 Structure of (DES) :-

### How Does it works





## 2.4 Advanced Encryption Standard (AES)

### 2.4.1 Definition →

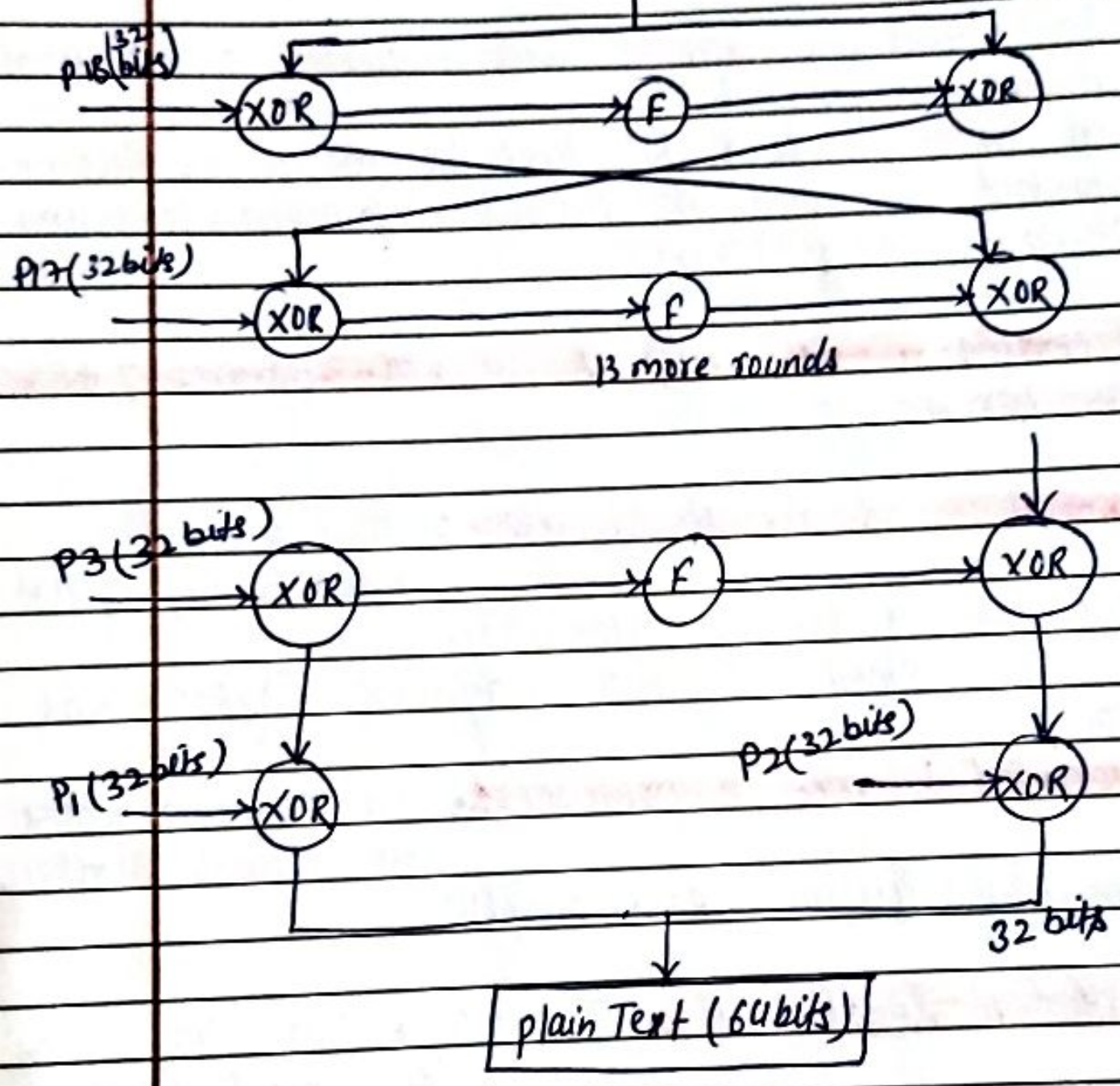
In October 2000 Rijndael was announced as the final selection for AES.

In November 2001 Rijndael became a US Government Standard published as Federal Information Processing Standard 197 (FIPS 197).

### 2.4.2 According to its design, the main features of AES are as follows.

- (i) Symmetric and Parallel structure → This gives the implementers of the algorithm a lot of flexibility. It also stands up well against cryptanalysis attacks.
- (ii) Adapted to modern processors → The algorithm works well with modern processors (Pentium, RISC, parallel).
- (iii) Suited to Smart Card → The algorithm can work well with smart card.

Cipher text Y (64 bits)





operation go The keys of Rijndael are in  
Mathematical Concept called as Galois  
field theory

Steps involved in, Rijndael at high level

- (i) Do the following one time initialization processes
  - (a) Expand the 16 byte key to get the actual key block to be used
  - (b) Do one time initialization of the 16 byte plain text block called as state.
  - (c) XOR the state with the key block.

- (ii) For each round, do the following
  - (a) Apply S-box to each of the plain text bytes.
  - (b) Rotate row R of the plain text block (i.e. state) by R bytes.
  - (c) Perform a Mix Column operation
  - (d) XOR the state with key block.

#### 2.4.4 AES Transformation :-

AES has many transformation function  
including :-

4.4.1 S-box (Substitute Byte Transformation)



## 9.4.4 S-box

### Definition →

In cryptography, an S-box (Substitution box) is a basic component of symmetric key algorithm which perform substitution.

In block cipher they are typically used to obscure the relationship between the key and ciphertext — shanon property of confusion.

In general an S-box takes ~~from~~ some number of input bits,  $m$ , and transform them into some number of outputs bits,  $n$ , where  $n$  is not necessarily equal to  $m$ .

An  $m \times n$  S-box can be implemented as a look up table with  $2^m$  words of  $n$  bits each.

Fixed table are normally used as in the data encryption standard. but in some cipher the tables are generated dynamical from the key



## 2.1 S-Box Design Criteria

Description  $\Rightarrow$  S-boxes are boolean mapping from  $\{0,1\}^m \rightarrow \{0,1\}^n$ ,  $m \times n$  mappings. Thus there are  $n$  component functions each being a map from  $m$  bits to  $1$  bits. In other words each component's function is boolean function.

### 2.4.4.3 Types of S-box

- $\Rightarrow$  From the cryptographic point of view even good S-boxes when implemented in hardware design are found to leak information.
- $\Rightarrow$  Hardware version of S-box may have power consumption.
- $\Rightarrow$  S-box are prone to attack side channel attacks causing frequent breakage of the cipher.
- $\Rightarrow$  Algebraic attacks are also evident on S/w implementation of S-Box.



## ~~2.7~~ Bent function

2.7.1 Definition 80. A bent function is a special types of boolean function. Bent functions are named so because they present maximum possible distance from all linear and affine function.

### 2.7.2 Properties

⇒ A bent function can be defined as a boolean function

$$f: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$$

⇒  $f(x_1, x_2) = x_1 x_2$  and  $G(x_1, x_2, x_3, x_4) = x_1 x_2 \oplus x_3 x_4$  are the simplest example of bent function.

⇒ The sequence of value  $(-1)^{f(x)}$  with  $x \in \mathbb{Z}_2^n$  is called a bent sequence.

⇒ Bent function and bent sequence have equivalent properties.



## 2.4.4 S-Box Design

### (i) Balanced Component function

The component function of substitution box must be balanced in the manner to ensure a message bits.

### (ii) Non-linearity of Component function

High level of non linearity must be ensured during the design of the component function.

### (iii) Non-Zero Linear Combination

Non zero linear combination of components function must be balanced and highly non-linear.



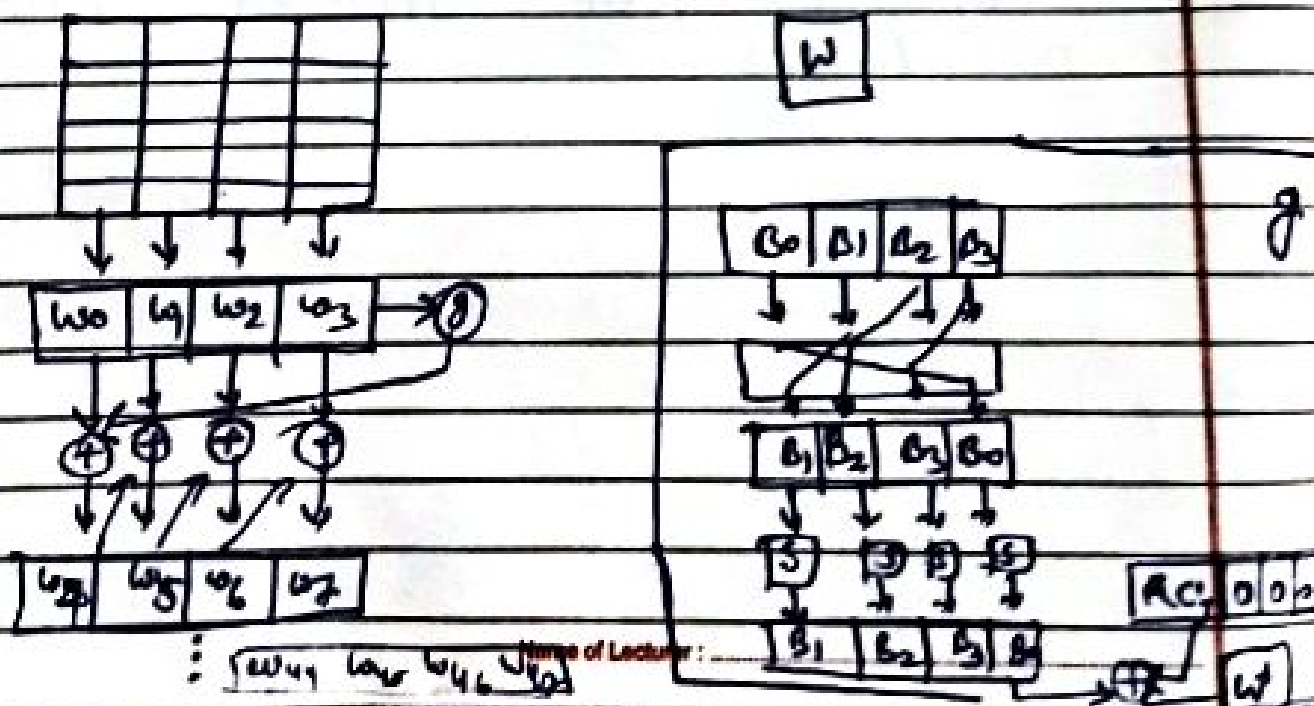
## 2.5 Key Expansion

The AES key expansion algo takes as Input a four-word (16 byte) and produces a linear array of 44 words (176 bytes).

This is sufficient to provide a four-word round key for the initial AddRoundKey stage and each of 10 rounds of the cipher.

→ The key is copied into four words of the expanded key. The remainder of the expanded key is filled in four words at a time. Each new word  $w[i]$  depends on the immediately preceding word,  $w[i-1]$ , and added word  $w[i-4]$  four positions back,  $w[i-4]$ . An irreducible polynomial, a simple XOR is used.

### 2.5.1 Diagramatic Implementation ⇒





## 2.6 Multiple Encryption

⇒ Multiple Encryption is the process of encrypting an already encrypted message one or more times, either using the same or a different algorithm. It is also known as cascade encryption, cascade ciphering, multiple encryption & superciphertext.

2.6.1 ⇒ Independent Keys :- Picking two ciphers, if the key used is the same for both, the second cipher could possibly undo the first cipher, partly or entirely. If an attacker could possibly decrypt all the remaining layers, assuming the same key is used for all layers.

2.6.2 ⇒ Independent Initialization Vector :- Unlike the exception of the one-time pad, no cipher has been theoretically proven to be unbreakable. Furthermore, some recurring prop. may be found in the ciphertexts generated by the first cipher. Since these ciphertexts are the plaintexts used by the second cipher, the second cipher may be rendered vulnerable to attack based on known plaintext properties.



To prevent this kind of attack, one can use the method provided by Bruce Schneier.

- Generate a random pad  $R$  on the same size as the plaintext.
- Encrypt  $R$  using the first cipher and key.

- XOR the plaintext with the pad, then encrypt the result using the second cipher.
- Concatenate both ciphertexts in order to build the final ciphertext.

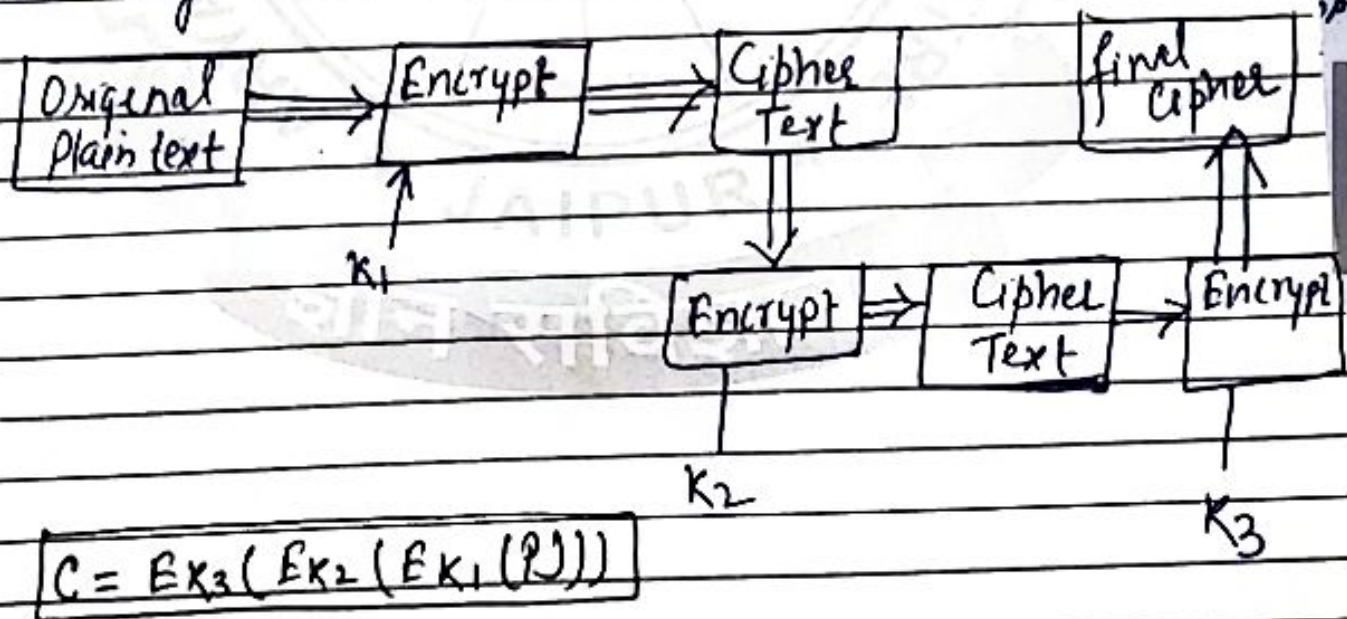
## 2.4 Triple DES

Although the meet in middle attack on Double DES is not quite practical yet, in cryptography it is always better to take the minimum possible chances. Consequently Double DES seemed inadequate, paving the way for Triple DES.

As we can imagine Triple DES is DES three times.

It comes in two flavours. one that uses three keys and other that uses two keys. We will study both one by one.

Triple DES with three keys So the idea of Triple DES with three keys



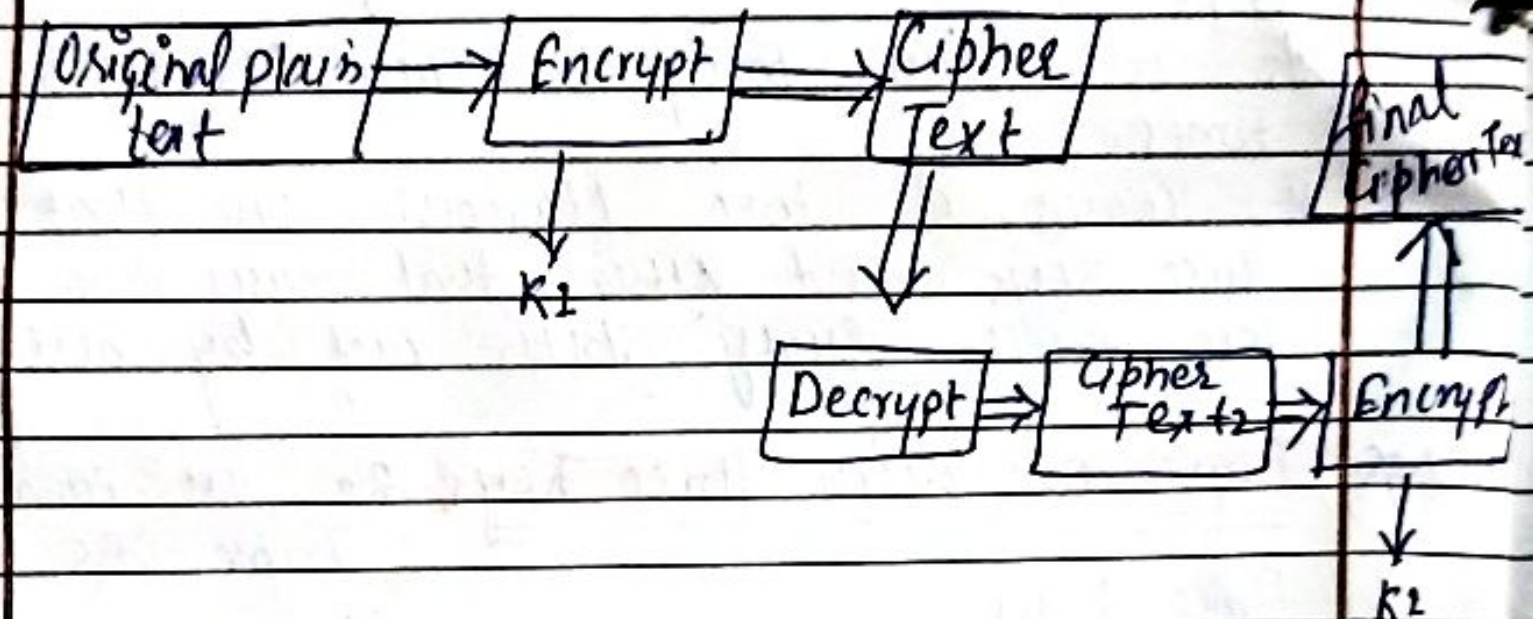


2

## Triple DES With two keys

Triple DES with three keys is highly secure. It can be denoted in the form of equation  $C = EK_3(EK_2(EK_1(P)))$ .

$$C = EK_3(EK_2(EK_1(P)))$$





## Block Cipher modes of operation

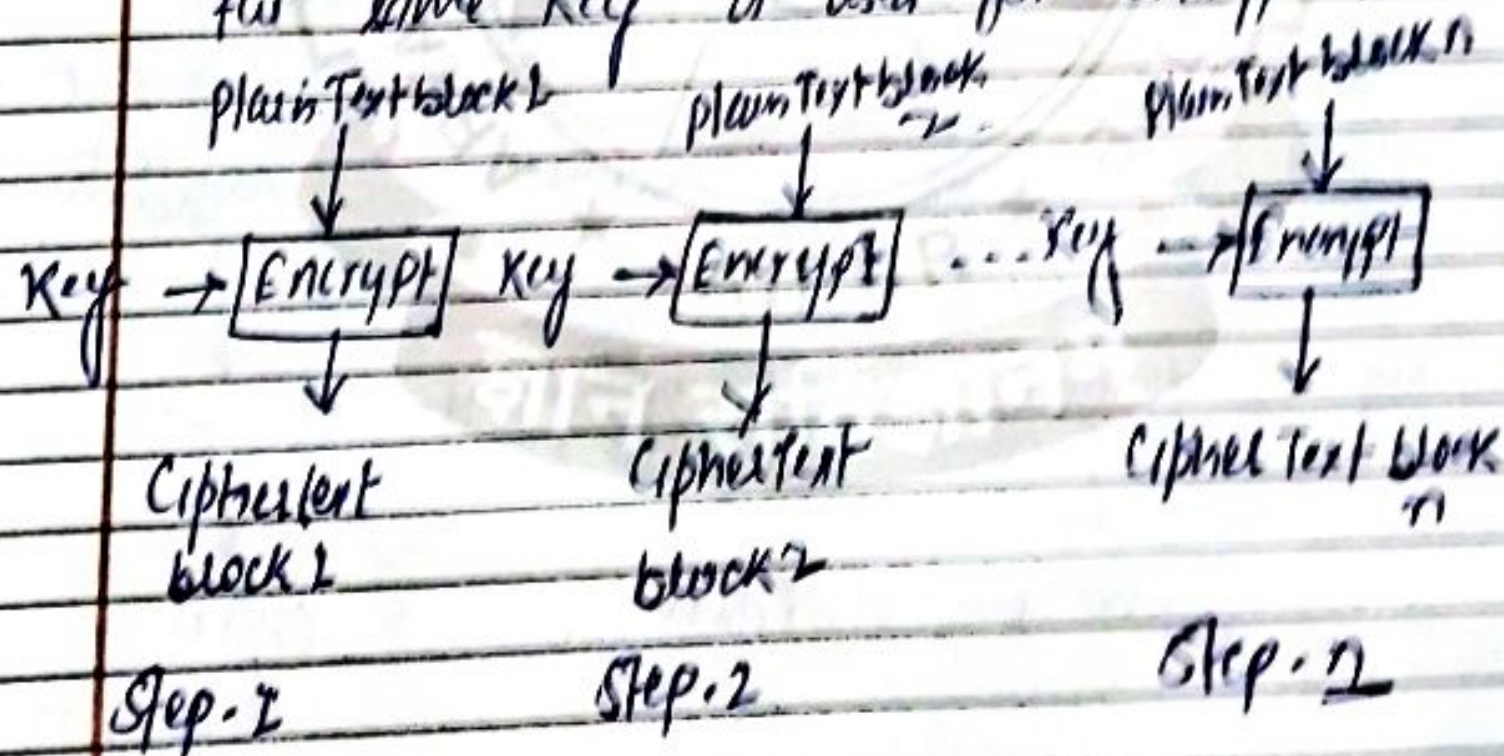
There are four important operation modes

- ① Electronic Code Book (ECB)
- ② Cipher Block Chaining (CBC)
- ③ Cipher feedback (CFB)
- ④ Output feedback (OFB)

### 2.3 ① Electronic Code Book

ECB is the simplest mode of operation. Here the incoming plain text message is divided into blocks of 64 bit each.

Each such block is then encrypted independently of the other blocks. For all blocks in message the same key is used for encryption.



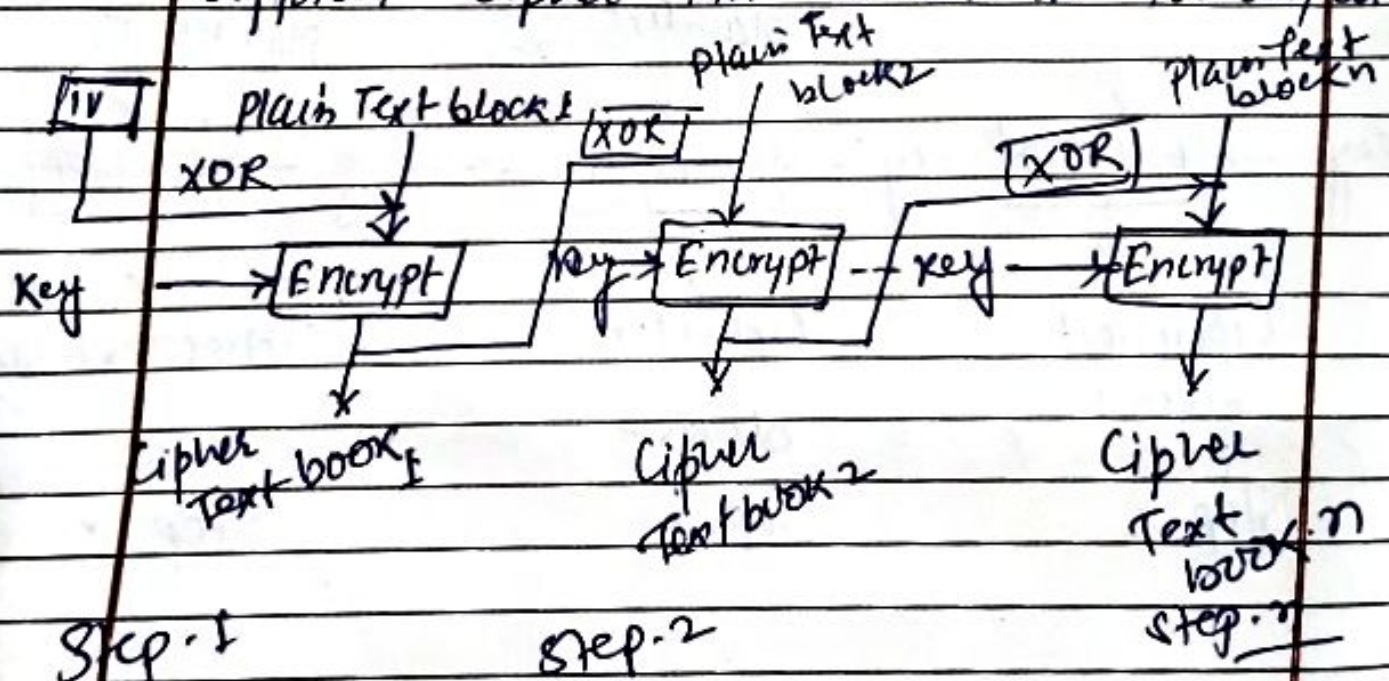


## Cipher Block chaining mode (CBC)

We saw that in case of ECB within a given message / for a given key, a plain text block always produces same cipher text block. Thus if a block of plain text occurs more than one in the input the corresponding cipher text block will also occur more than one in output.

This provides some clues to Cryptanalyst. To overcome this problem the Cipher block chaining mode ensures that even if a block of plain text repeats in the input.

These two plain text blocks yield totally different cipher text blocks in the output.

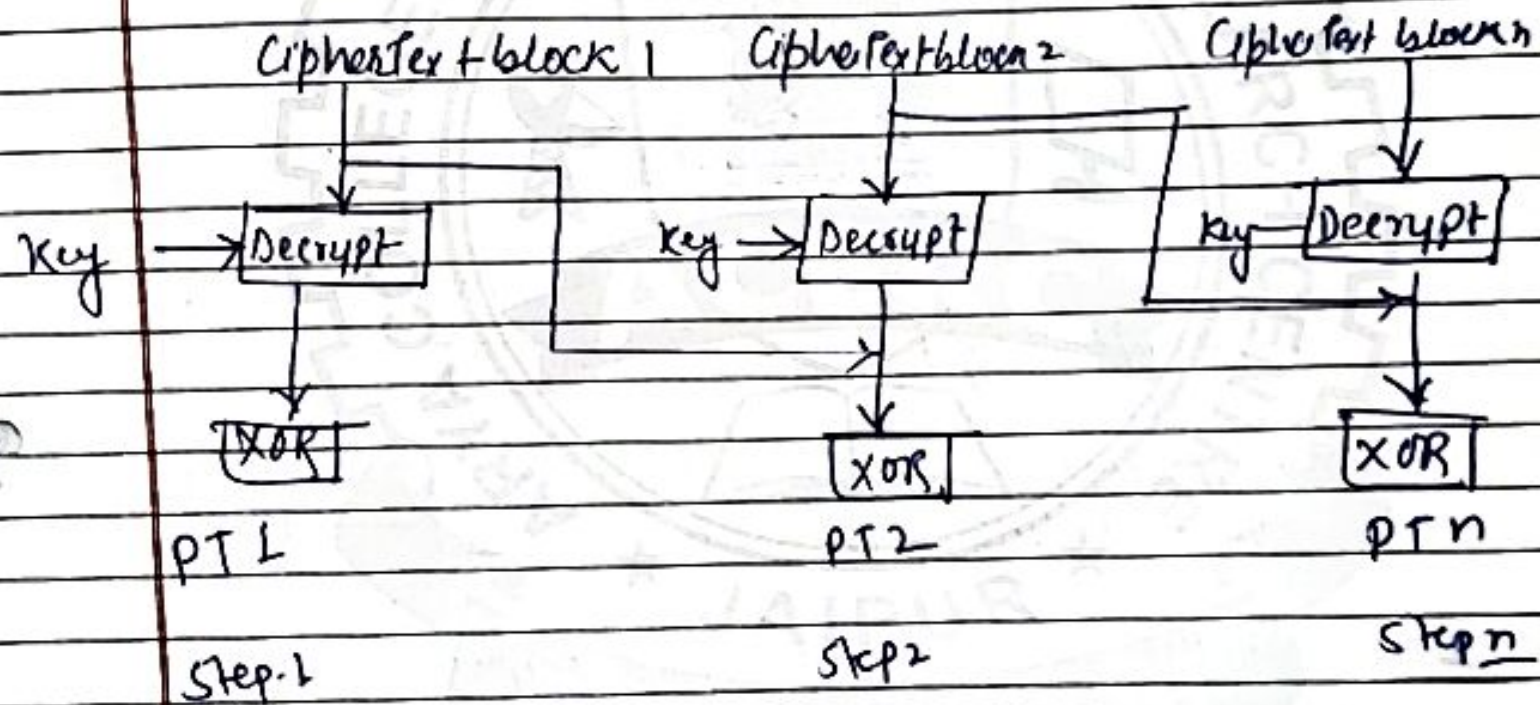




### Ex 10: ③ Cipher Feedback mode

Not all application can work with blocks of data.  
Security is also required in application that are character oriented.

The Cipher feedback mode is useful in such cases. In this mode data is encrypted in units that are smaller (that it could be size of 8 bits).

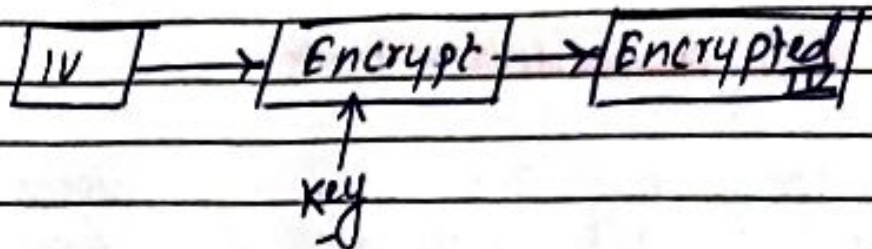


CBC mode- Full Decryption process

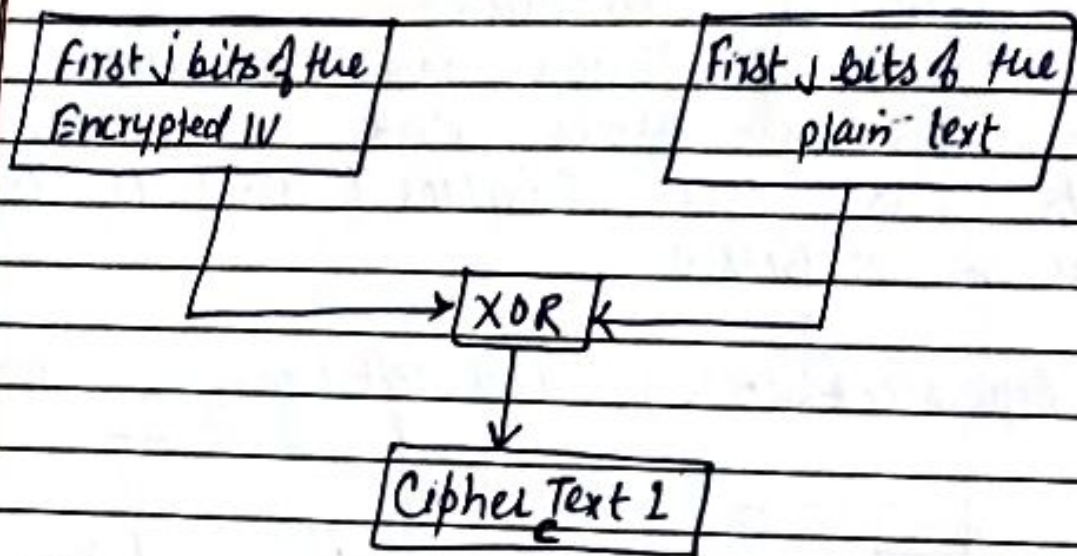


## Working of CFB

Step-1



Step-2



Left shift IV by j position



Moves bits of C into the rightmost side of IV



Step 3

Now the leftmost j bits of the encrypted IV are XORed with first j bits of plain Text. This produces the 1st position of Cipher Text say (C) as shown

Step-3

Now the bits IV are shifted left by j position. Then the rightmost j position of the shift register now contain unpredictable data. These rightmost j position are now filled with C.



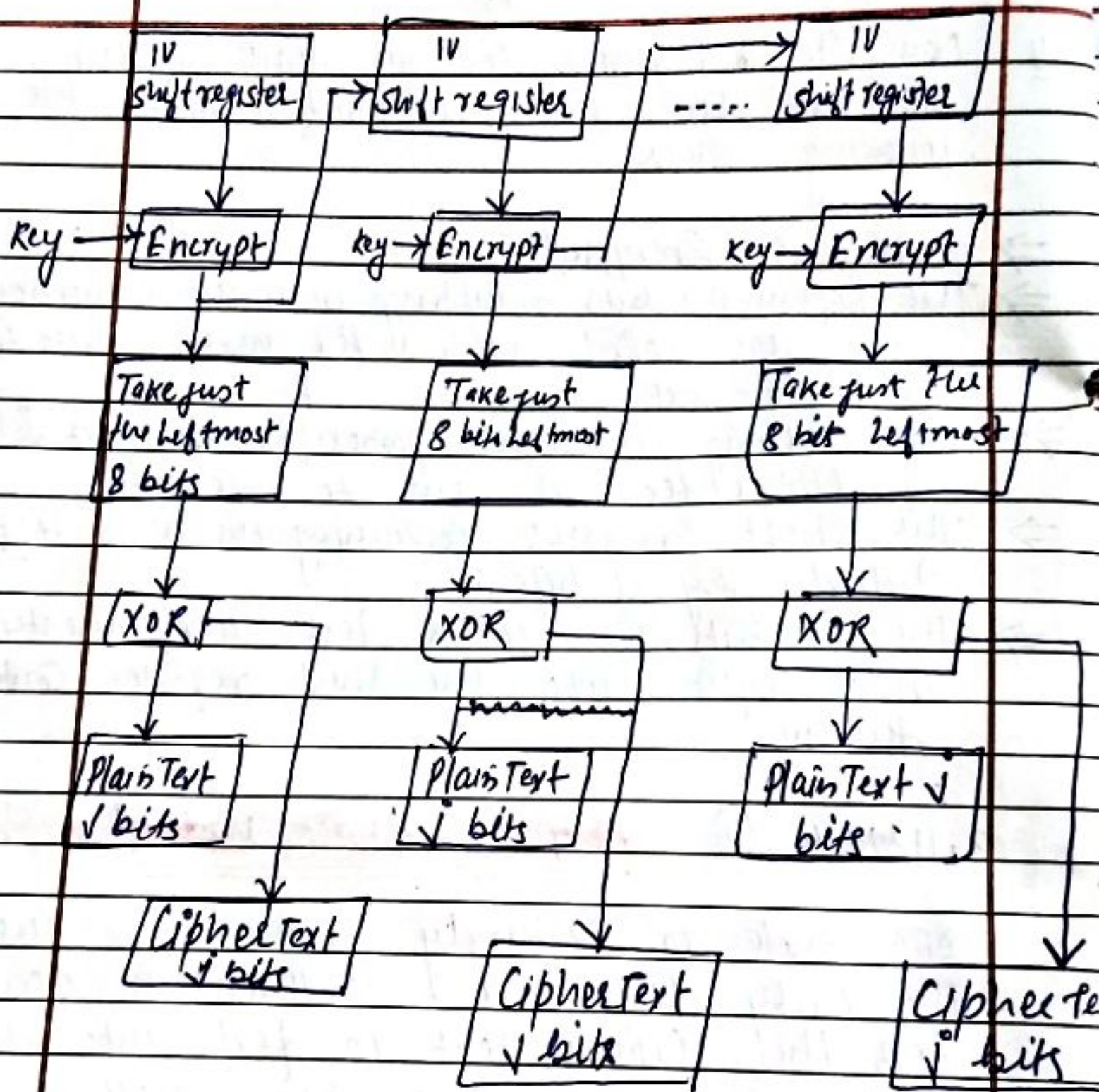
Step 4 Now Step 1 and 3 continue until all the plain text units are encrypted that is the following steps.

- ⇒ IV is Encrypted
- ⇒ The leftmost  $j$  bits resulting from this encryption process are XORed with the next  $j$  bits of the plain text.
- ⇒ The resulting cipher text portion (i.e. the next  $j$  bits of cipher text) is sent to receiver
- ⇒ The shift register containing the IV is left shifted by  $j$  bits.
- ⇒ The  $j$  bits of cipher text are inserted from right into the shift register containing the IV.

#### 2.11. (4) Output Feedback Mode (OFB)

OFB mode is extremely similar to the CFB. The only difference is that in case of CFB the cipher text is fed into the next stage of encryption process. But in case of OFB the output of the IV shift register encryption process is fed into next stage of encryption process.





CFB - The overall encryption process



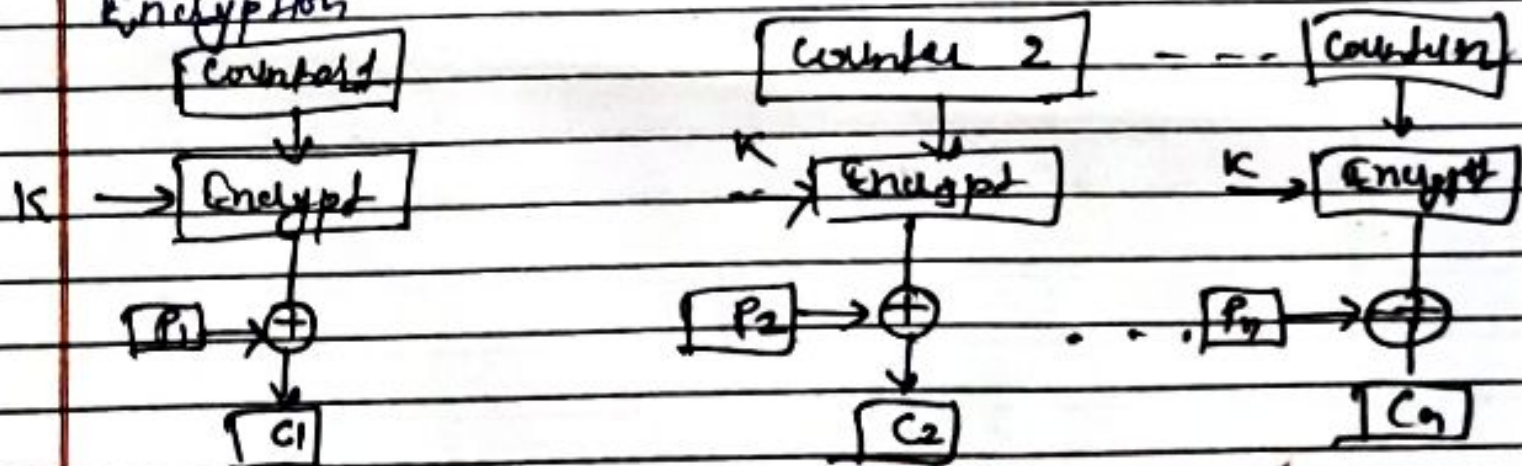
## 2.12. Counter Mode

The Counter Mode or (CTR) is a simple counter based block cipher implementation in cryptography. Each or every time a counter initiated value is encrypted and given as input to XOR with plaintext or ciphertext which results in ciphertext block. The CTR Mode is independent of feedback use and thus can be implemented in parallel in this mode. It generates the next keystream block by encrypting successive values as named as "Counter".

⇒ This counter can be any purpose or function which generates a sequence that is guaranteed not to call for a long time, although on actual increment by one counter is the simplest or easiest & most popular or famous.

Simple implementation is shown as :

Encryption

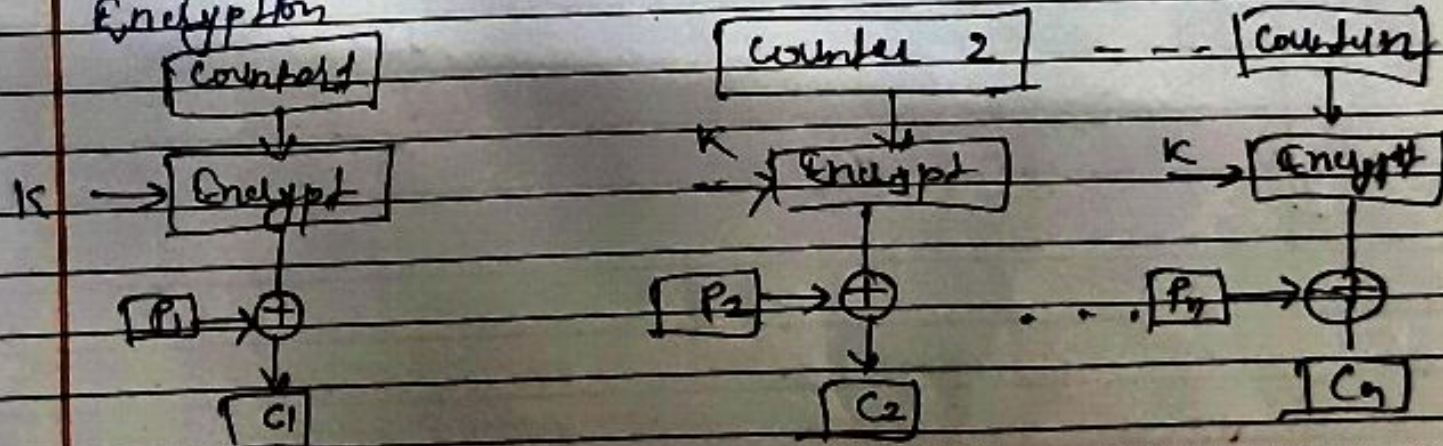


Name of Lecturer : .....



Simple implementation is shown as a

Encryption



Name of Lecturer: .....

Decryption

