

Assignment - 06  
Iftate Jahan Nilima

IT-21063

## Block cipher modes of operation:

A block cipher is an encryption algorithm that processes data in fixed-size blocks rather than one bit at a time. Block cipher modes of operation define how to securely encrypt and decrypt large amounts of data using a block cipher.

Here are a few common modes:

### Electronic code block (ECB):

The electronic codebook is the easiest block cipher mode of functioning. It is easier because of the direct

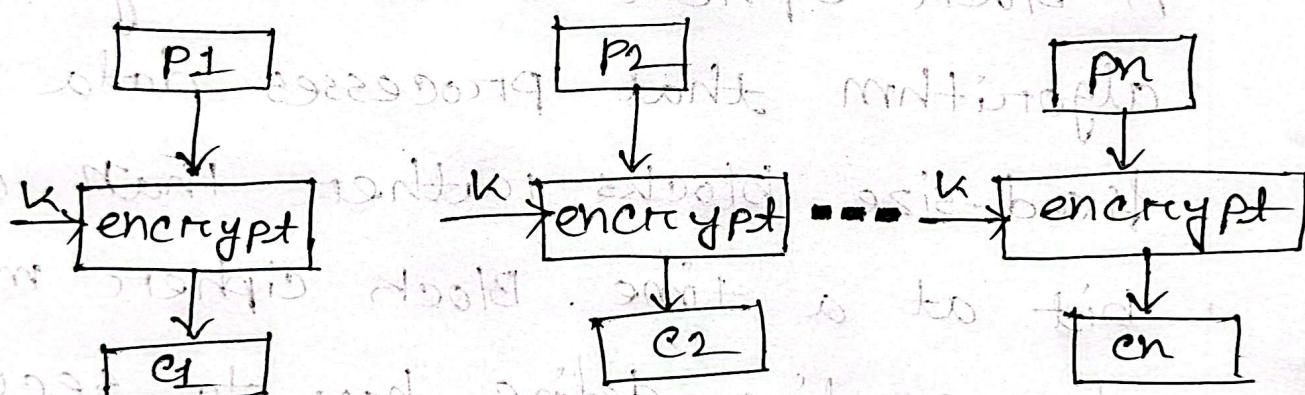
Nilima

IT-21063

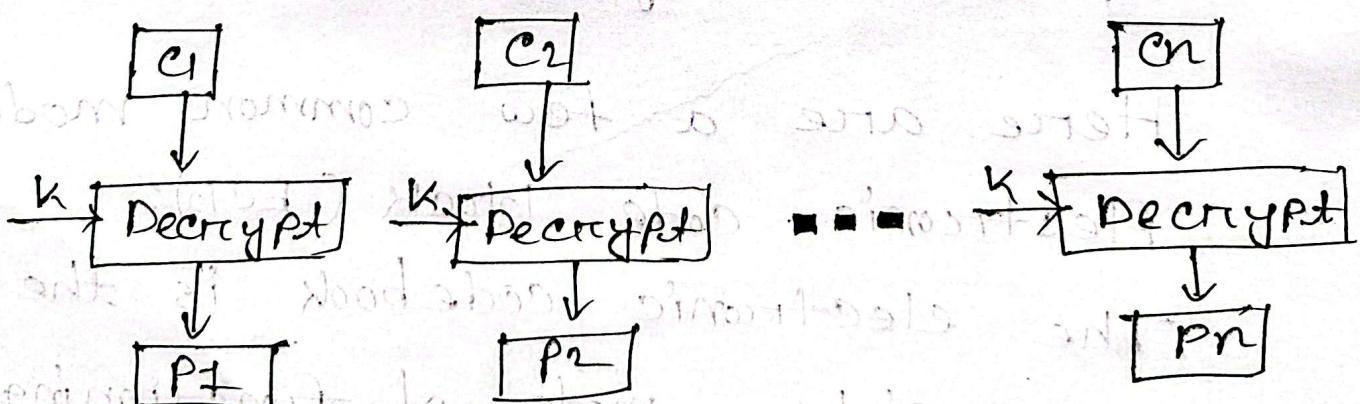
encryption of each block of input plaintext and output is in the form of blocks of encrypted ciphertext

Diagram:

Encryption:



Decryption:



Smith Nilima  
S2018545 IT-21063

## Advantages of ECB:

- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption
- simple way of the block cipher

## Disadvantages of ECB:

- Prone to decryption of blocks of bits is possible, thus it is a faster way of encryption.
- simple way of the block cipher

Nilema

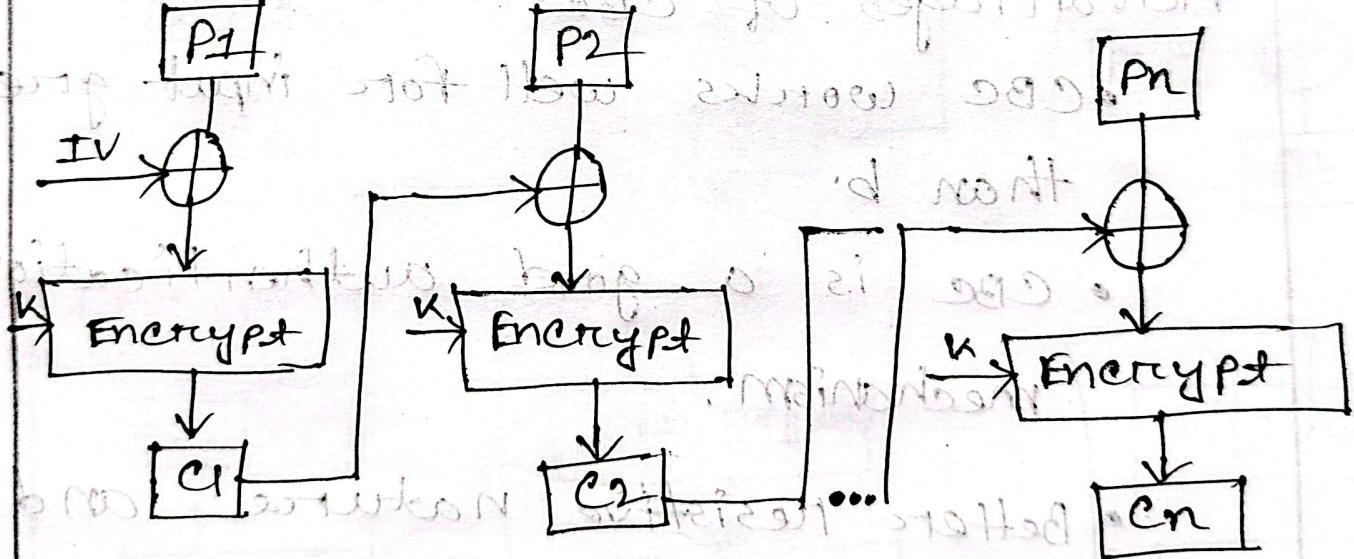
TT-21063

## Cipher Block Chaining (CBC)

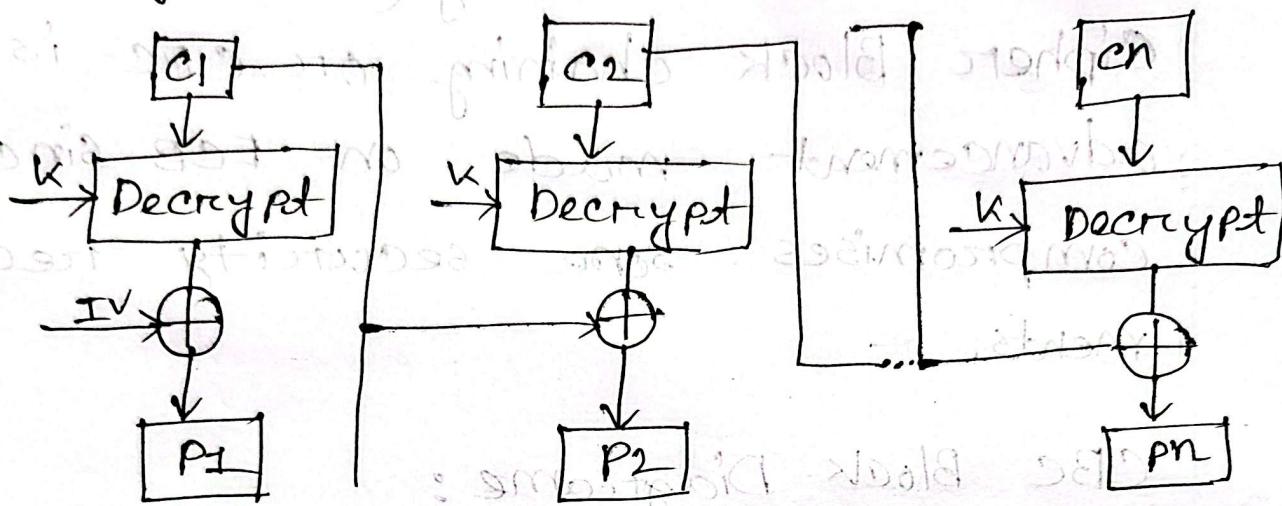
Cipher Block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements.

CBC Block Diagram:

Encryption



## Decryption:



## Advantages of CBC:

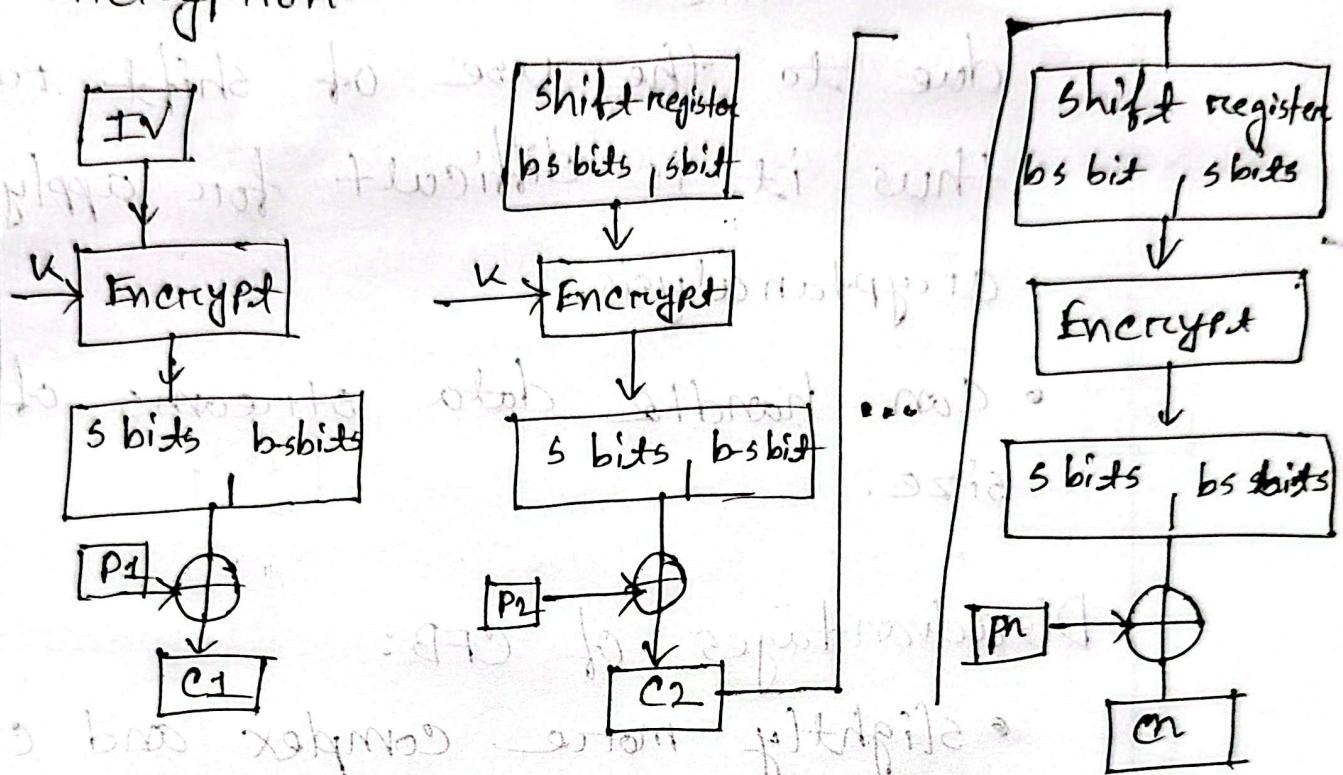
- CBC works well for input greater than b.
- CBC is a good authentication mechanism.
- Better resistive nature and more secure.

## Disadvantages:

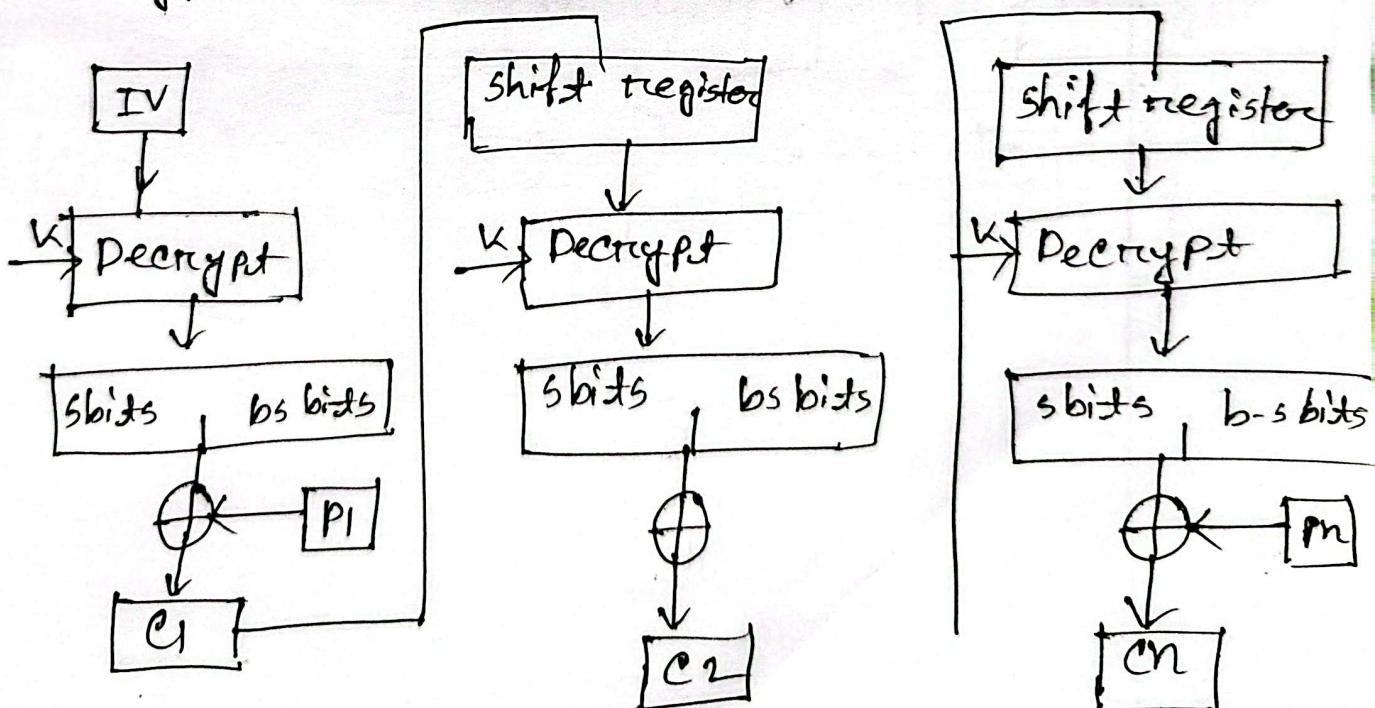
- Requires the previous ciphertext block for encryption and decryption.

## CFB Block Diagram:

Encryption



Decryption:

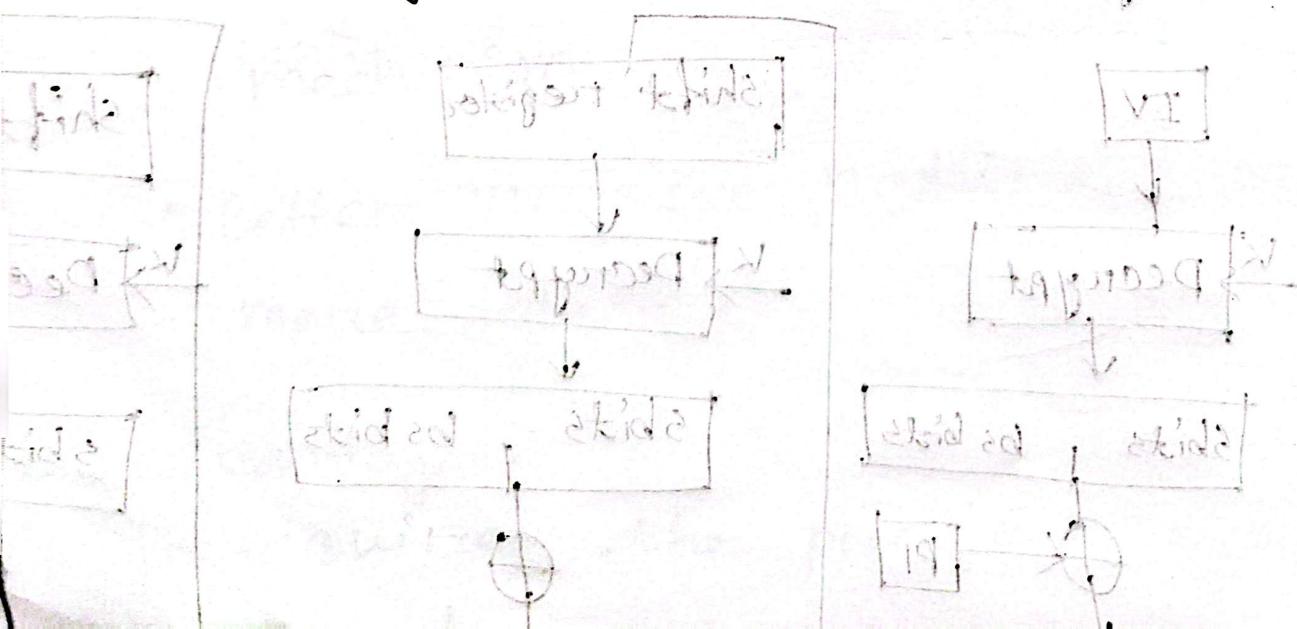


## Advantages of CFB:

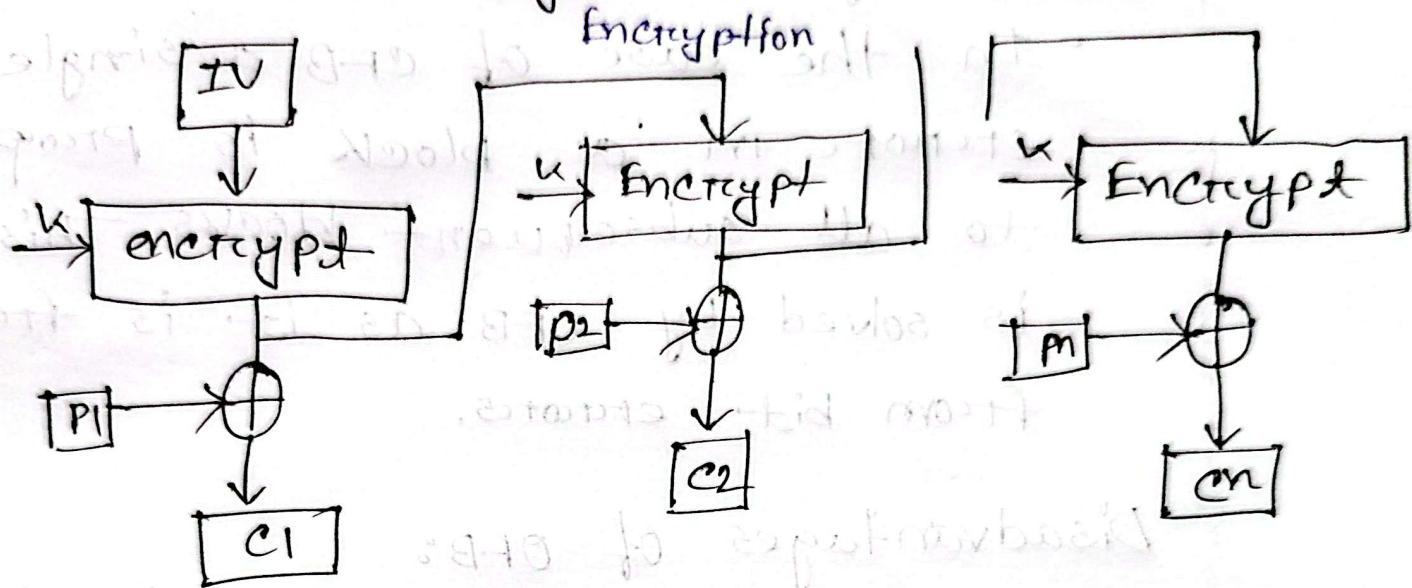
- Since, there is some data loss due to the use of shift register thus it is difficult for applying cryptanalysis.
- Can handle data streams of any size.

## Disadvantages of CFB:

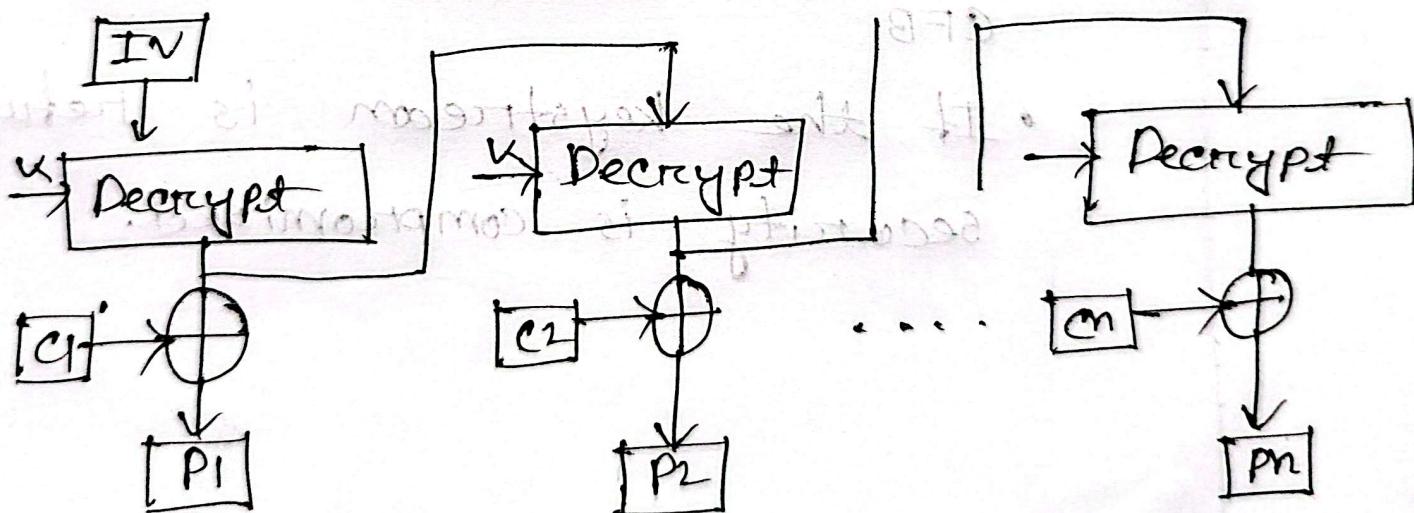
- slightly more complex and can propagates errors.



## OFB Block Diagram:



## Decryption:



## Advantages of OFB

- In the case of CFB a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as it is free from bit errors.

## Disadvantages of OFB:

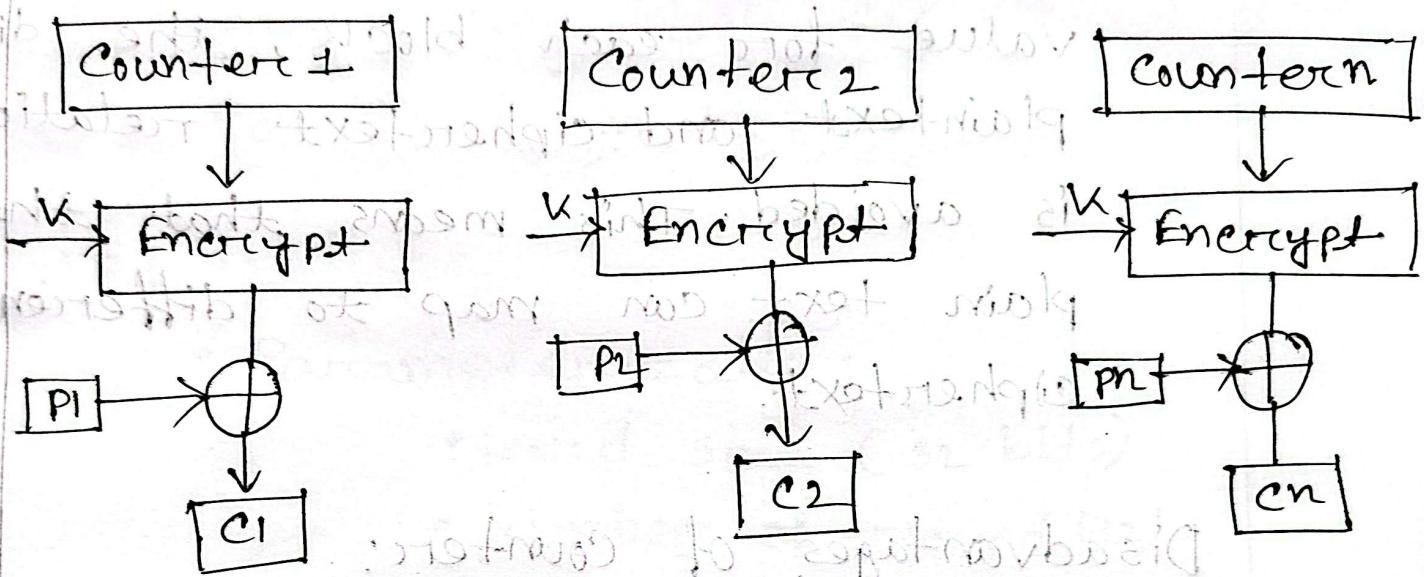
- It is more susceptible to message stream modification attack than CFB.

- If the keystream is refined, security is compromised.

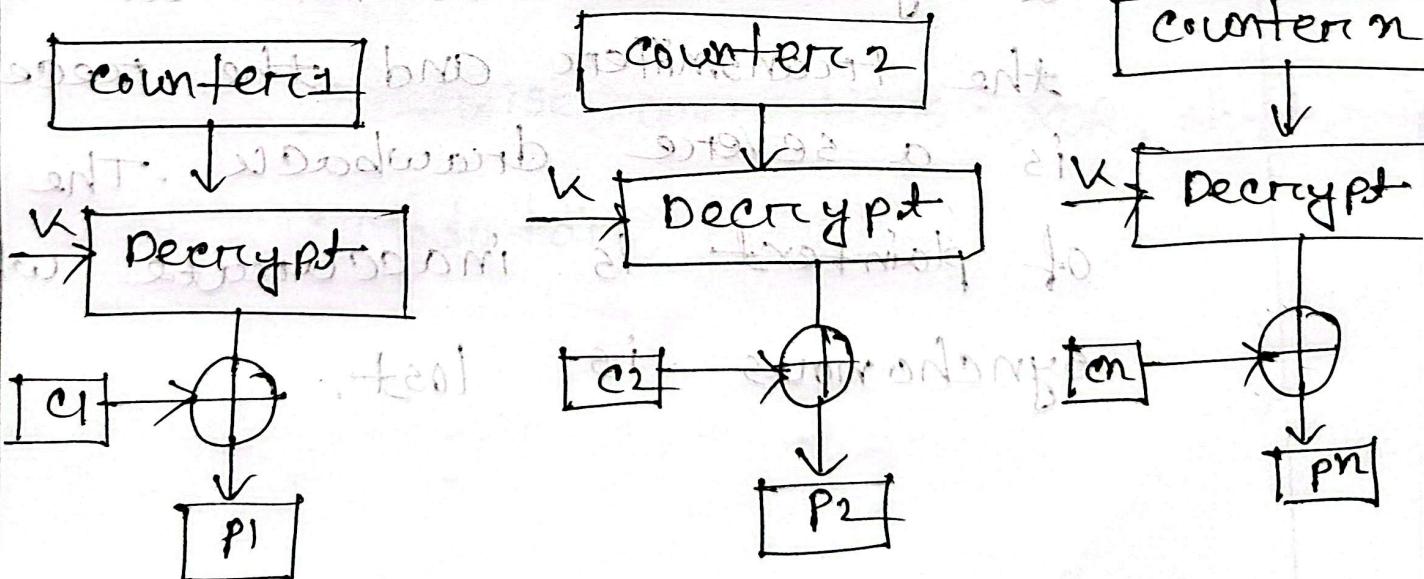
Nilema

IT-21063

### Block Diagram of CTR:



### Decryption



## Advantages of counter (CTR)

- Since there is a different counter value for each block, the direct plaintext and ciphertext relationship is avoided. This means that the same plain text can map to different ciphertext.

## Disadvantages of counter:

- The fact that CTR mode requires a synchronous counter at both the transmitter and the receiver is a severe drawback. The recovery of plaintext is inaccurate when synchronous is lost.

## Introduction of RC5:

RC5 is a fast, simple and secure symmetric key block cipher designed by Ron Rivest in 1994.

### Key features:

- Parameterizable:
  - Word size (32 bits)
  - Number of round (12)
  - Key length (8 bytes)
- Uses:
  - Bitwise operations: XOR, shift, rotate
  - Modular addition

## Java Implementation:

```
public class RC5 {
```

```
    private static final int WORD_SIZE = 32;
```

```
    private static final int R = 12;
```

```
    private static final int B = 8;
```

```
    private static final int C = 34;
```

```
    private static final int T = 2 * (R + 1);
```

(Add 16 words below)

```
    private int[] s = new int[T];
```

```
    private static final int P = 0xB7E15163;
```

```
    private static final int Q = 0x9E3779B9;
```

```
    public RC5 (byte[] key) {
```

```
        keySchedule(key);
```

```
}
```

```
private void keySchedule (byte[] key)
```

```
{
```

```
    int[] t = new int[C];
```

```
    for (int i = 0; i < B, i++) {
```

$L[i/A] = (L[i/A] \ll B) + (\text{key}[i] \text{ } 2:0 \times FF);$

}

$S[0] = P;$

for (int i=1; i < T; i++) {

$S[i] = S[i-1] + Q;$

}

int A=0, B=0, i=0, j=0;

for (int k=0; k < 3\*T; k++) {

$A = S[i] = \text{Integer.rotateleft}((S[i] +$   
 $0 \times B), 3);$

$B = L[j] = \text{integer.rotateleft}((L[j] +$   
 $A), (A+B));$

$i = (i+1) \% T;$

$j = (j+1) \% C;$

}

}

public int[] encrypt(int[] pt) {

$\text{int } A = pt[0] + S[0];$

SCHOOL-FILE

```
public int[] decrypt (int [] ct) {
    int B = ct [1];
    int A = ct [0];
    for (int i = R; i >= 1; i++) {
        A = s [0];
        B = s [1];
        return new int [] {A, B};
    }
}

System.out.print ("Encrypted : y.08x
y.08x\n", cf [0], ct [1]);
int [] dt = rc5.decrypt (ct);
System.out.print ("Decrypted : y.08x
y.08x\n", dt [0], dt [1]);
}
```

Output: Encrypted : 7f93d8c2 1a23ba29

Decrypted : 12345678 9abedef0

Code for ECB mode Example

```
import javax.crypto.cipher;
import java.security.KeyGenerator;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.security.Key;
```

public class ECBModeExample {

```
public static void main (String [] args)
    throws Exception {
    KeyGenerator keyGeneration = KeyGenerator.getInstance ("AES");
    SecretKey secretKey = keyGeneration.generateKey ();
    Cipher cipher = Cipher.getInstance ("AES/ECB/PKCS5Padding");
    test (cipher);
}
```

```
cipher.init(cipher.ENCRYPT_MODE, secretkey)
byte[] encrypted = cipher.doFinal
```

```
(("This is a test"), getbytes());
```

```
System.out.println("Encrypted : " + new
String(encrypted));
```

```
cipher.init(cipher.DECRYPT_MODE,
secretkey);
```

```
byte[] decrypted = cipher.doFinal
(encrypted);
```

```
System.out.println("Decrypted : " + new
String(decrypted));
```

```
}
```

Output: Encrypted: ??6.??JD?Vx?·??

Decrypted: This is a test.