Cryptography and cyber law

Assignment - 06

Murraduzzaman Asha
IT - 21014

Assignment topic:

Modes of Operation and RC5- Block Diagram and java Implementation and output.

Modes of Operation:

Block ciphers encrypt data in fixed-size blocks (64 or 128 bits). But in real applications, data is often much longer. So, we use modes of operations to securely process large data by using block cipher repeatedly.

Murcaduzzaman Asha
IT-21014

Common modes:

- ECB - Electronic codebook
- CBC - Cipher Block chaining
- CFB - Cipher Feedback
- OFB - Output Feedback
- CTR - Counter

Description:

ECB: Each block is encrypted independently Not secure for patterns.

CBC: XORs each plaintext block with previous ciphertext block before encryption.

CFB: Converts block cipher into a self synchronizing stream cipher

IT-21014

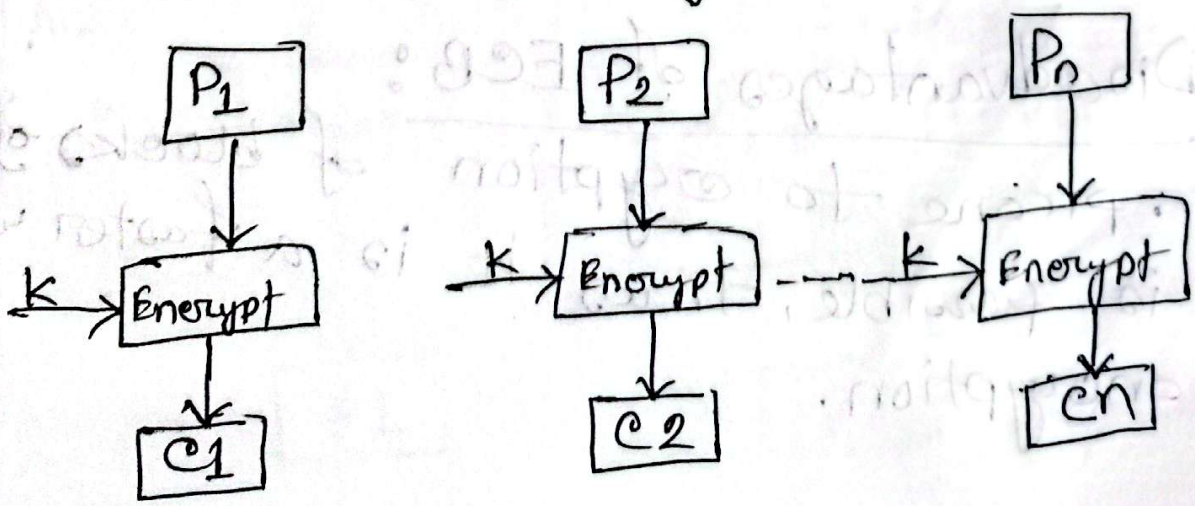
OFB: Turns block cipher into a Synchoronous stream cipher.

CTR: Use a counter that gets encrypted and XORed with plaintext. Fast and parallelizable.

Note: Among them, CBC and CTR are the most widely used due to their security and efficiency.

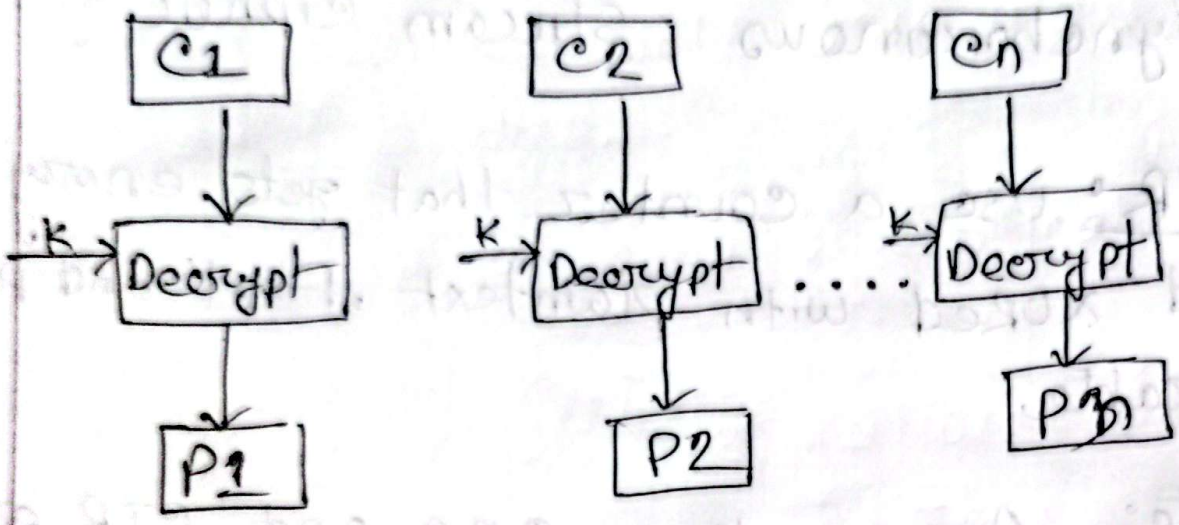
The procedure of ECB is illustrated below:

Encryption



IT-21014

Decryption



Advantages:

- 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:

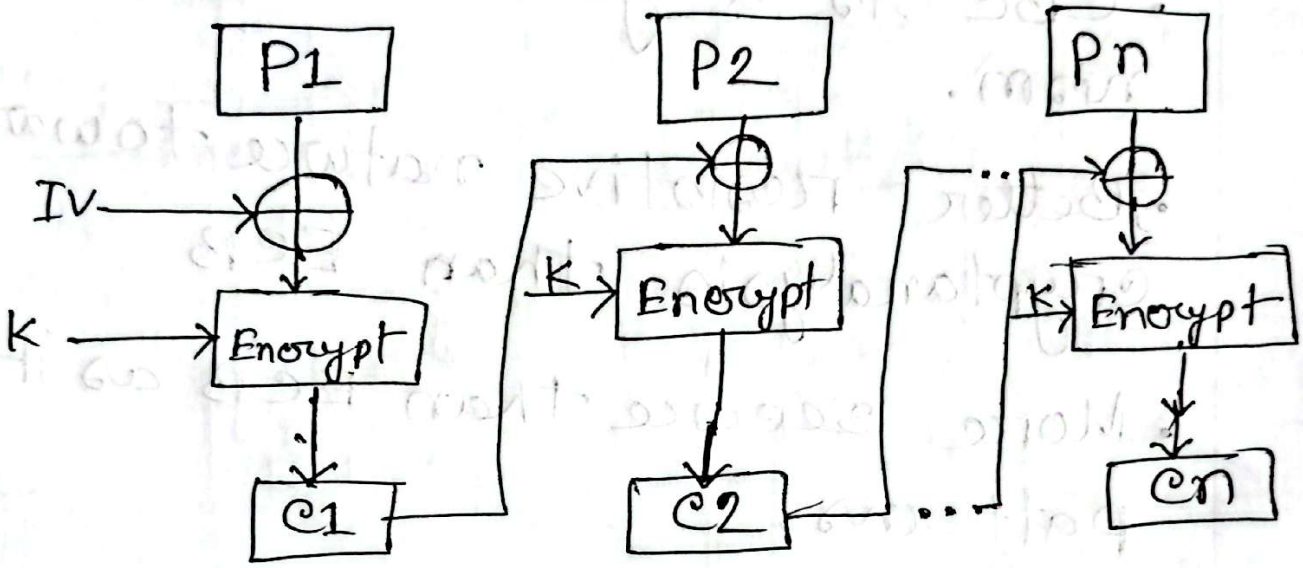
- prone to cryption of blocks of bits is possible, thus it is a faster way of encryption.

IT-21014

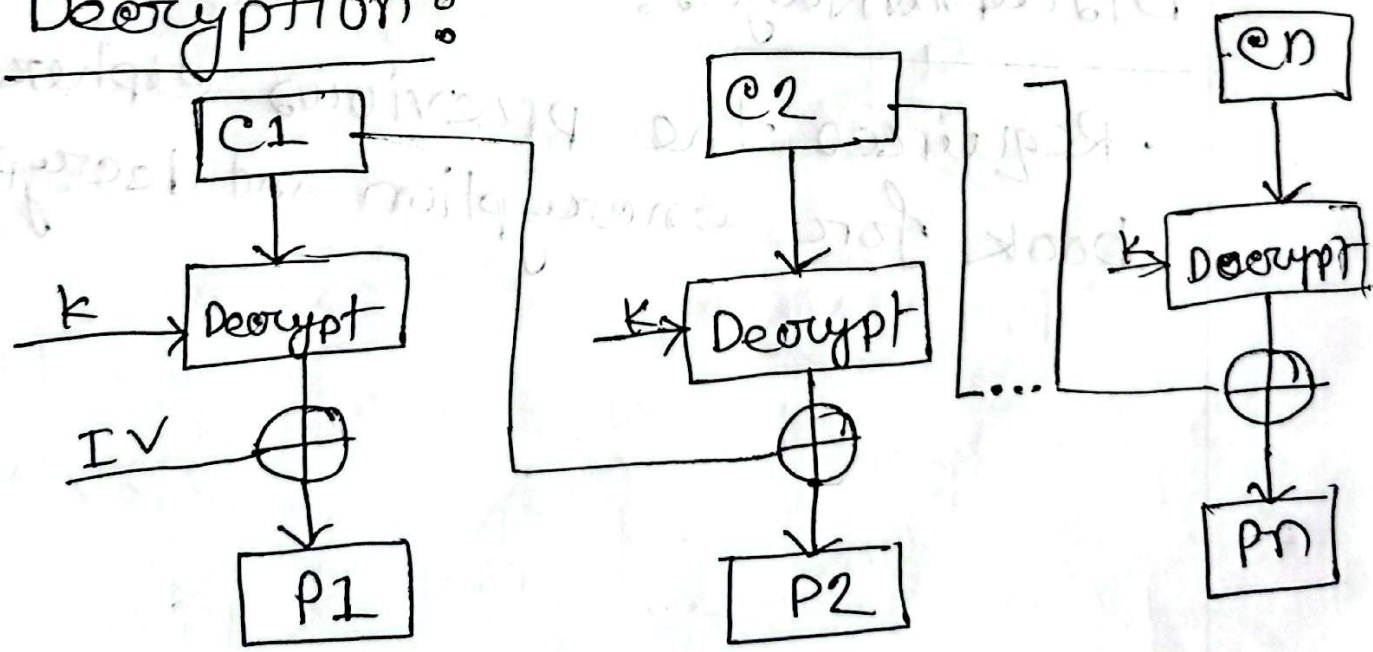
- Simple way of the block cipher.

CBC Block Diagram:

Encryption:



Decryption:



IT-21014

Advantages of CBC:

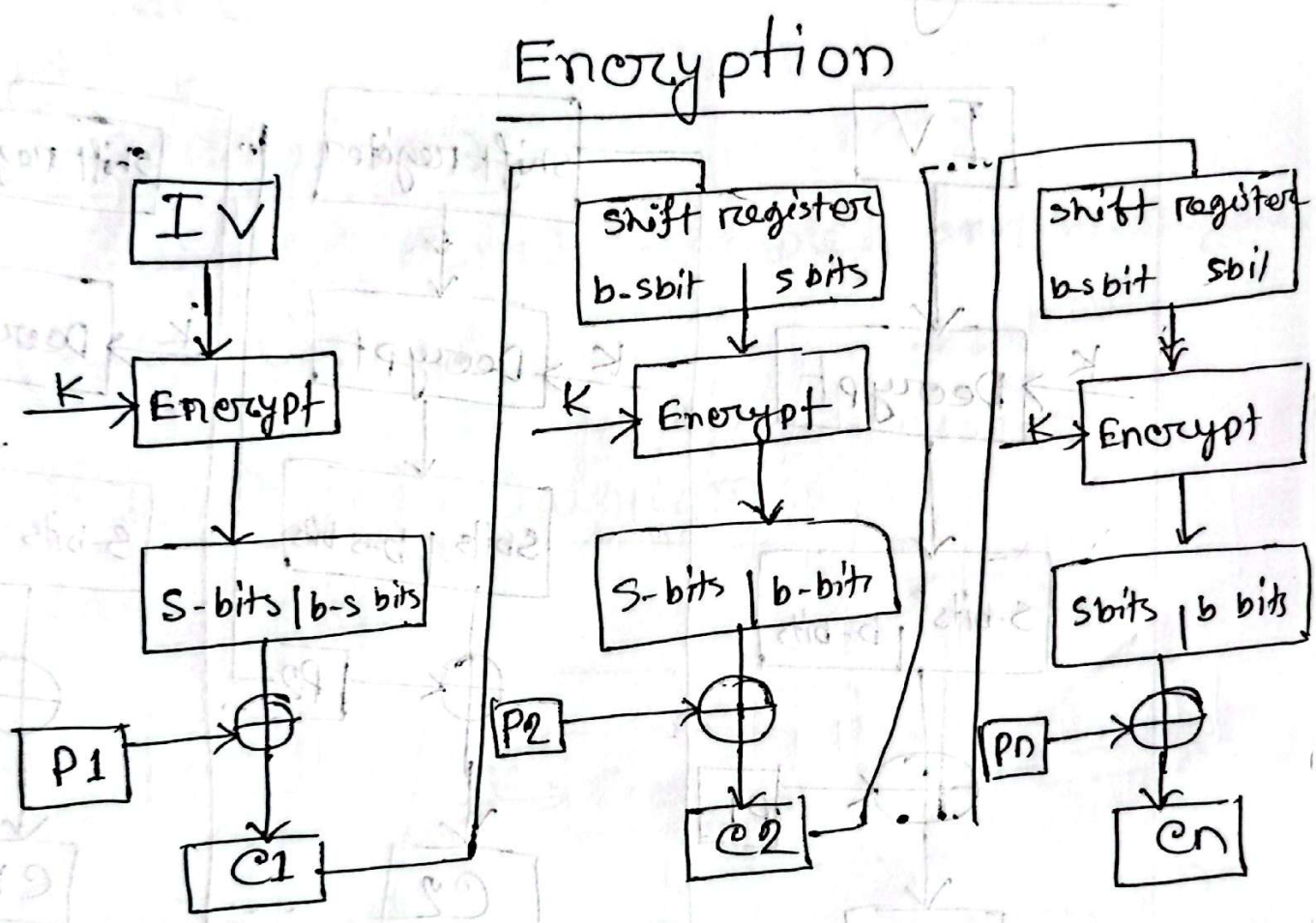
- CBC works well for input greater than b bits.
- CBC is a good authentication mechanism.
- Better resistive nature towards cryptanalysis than ECB
- More secure than ECB as it hides patterns.

Disadvantages:

- Requires the previous cipher text book for encryption and decryption.

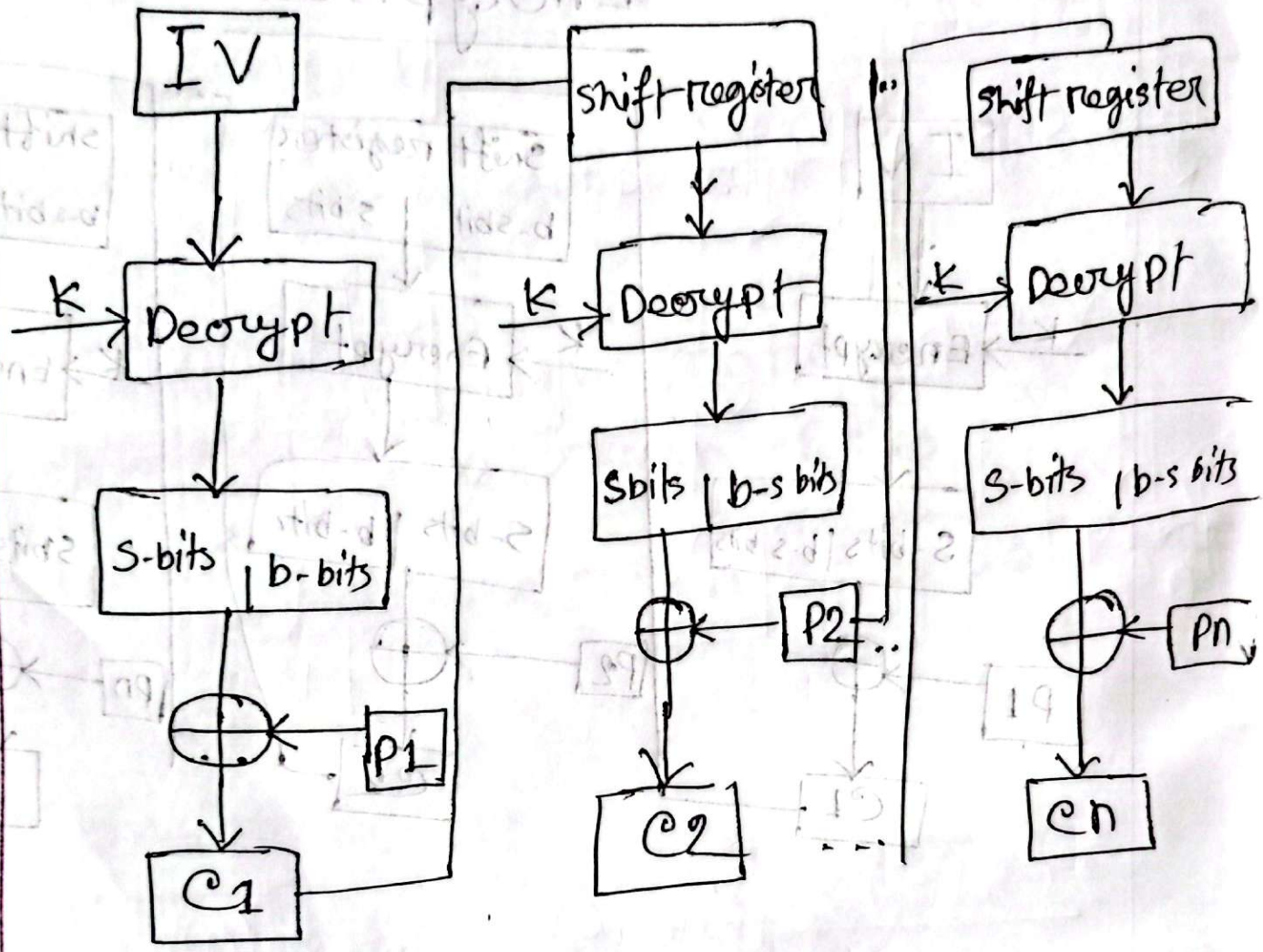
IT-21014

CFB Block Diagram:



IT 21014

Decryption:



Advantages of CFB:

- Since there is some data loss due to the use of shift register, thus it is difficult for applying crypto-analysis.

IT-21014

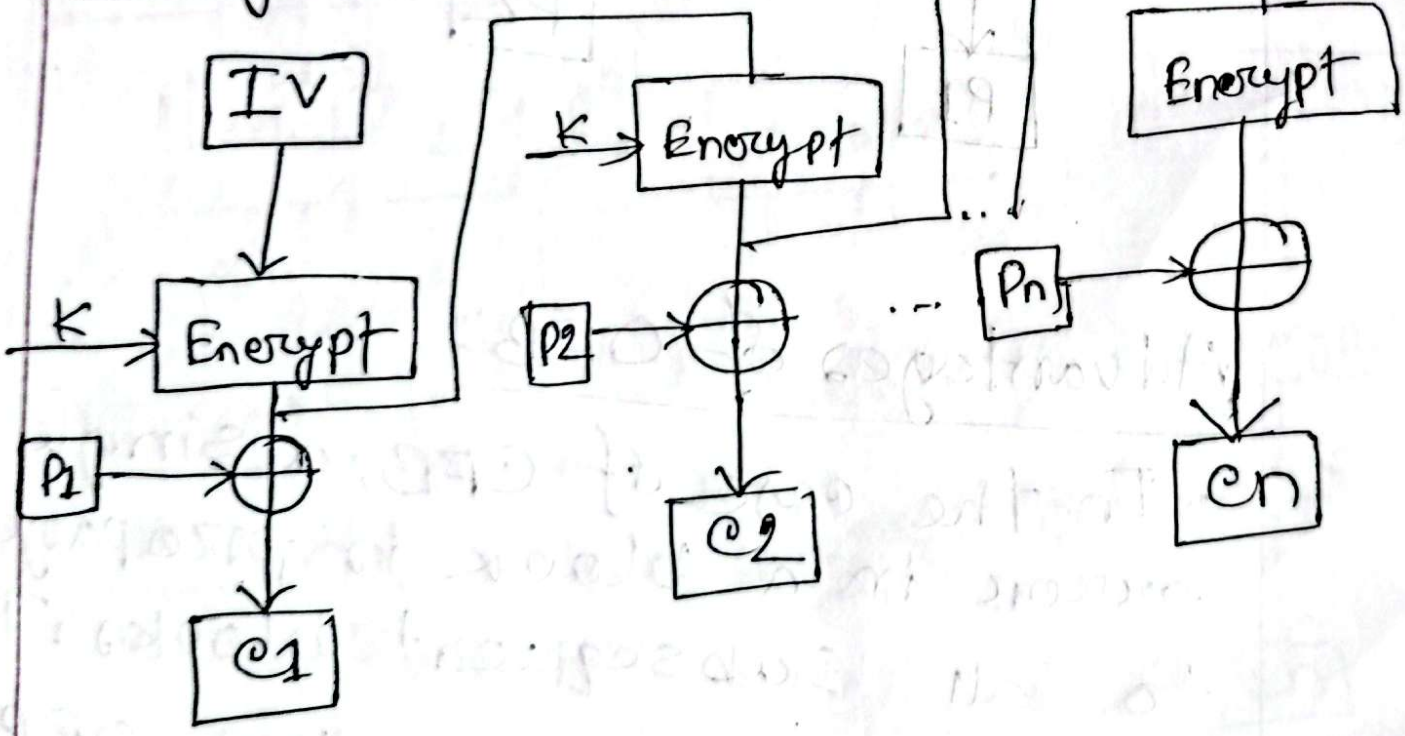
- Can handle data streams of any size.

Disadvantages:

- Slightly more complex and propagates errors.

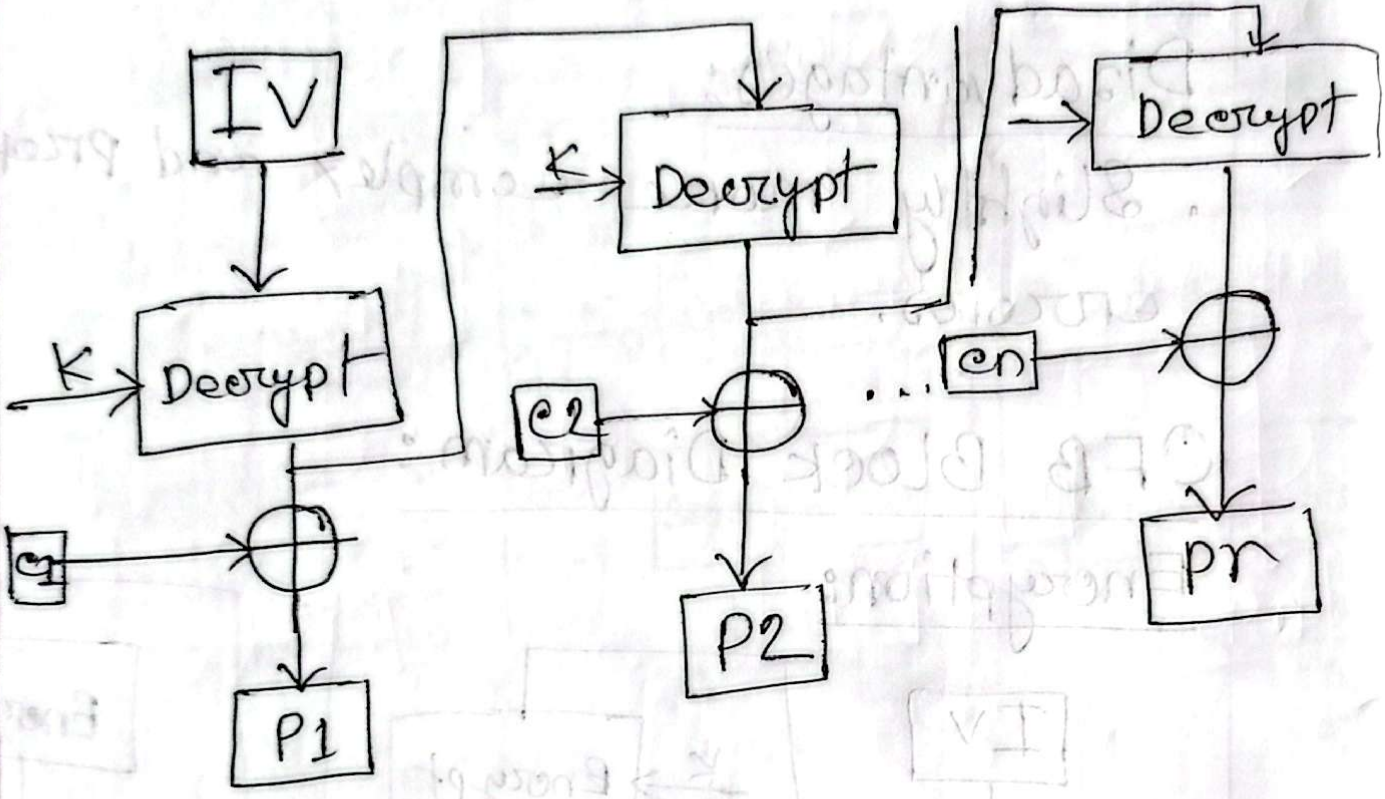
OFB Block Diagram:

Encryption:



IT-21014

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.

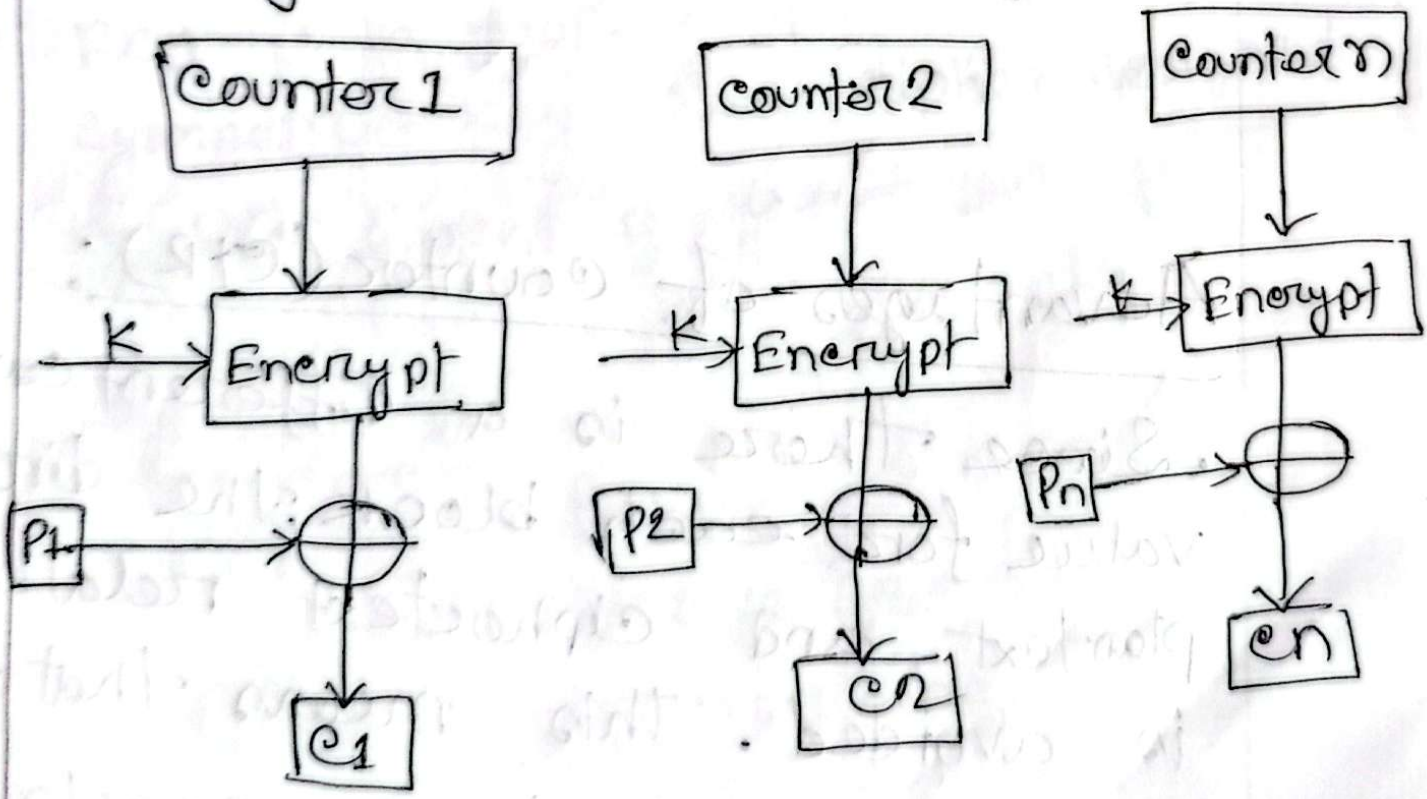
IT 21014

Disadvantages of OFB:

- It is more susceptible to a message stream modification attack than CFB.
- If the key stream is refused, security is compromised.

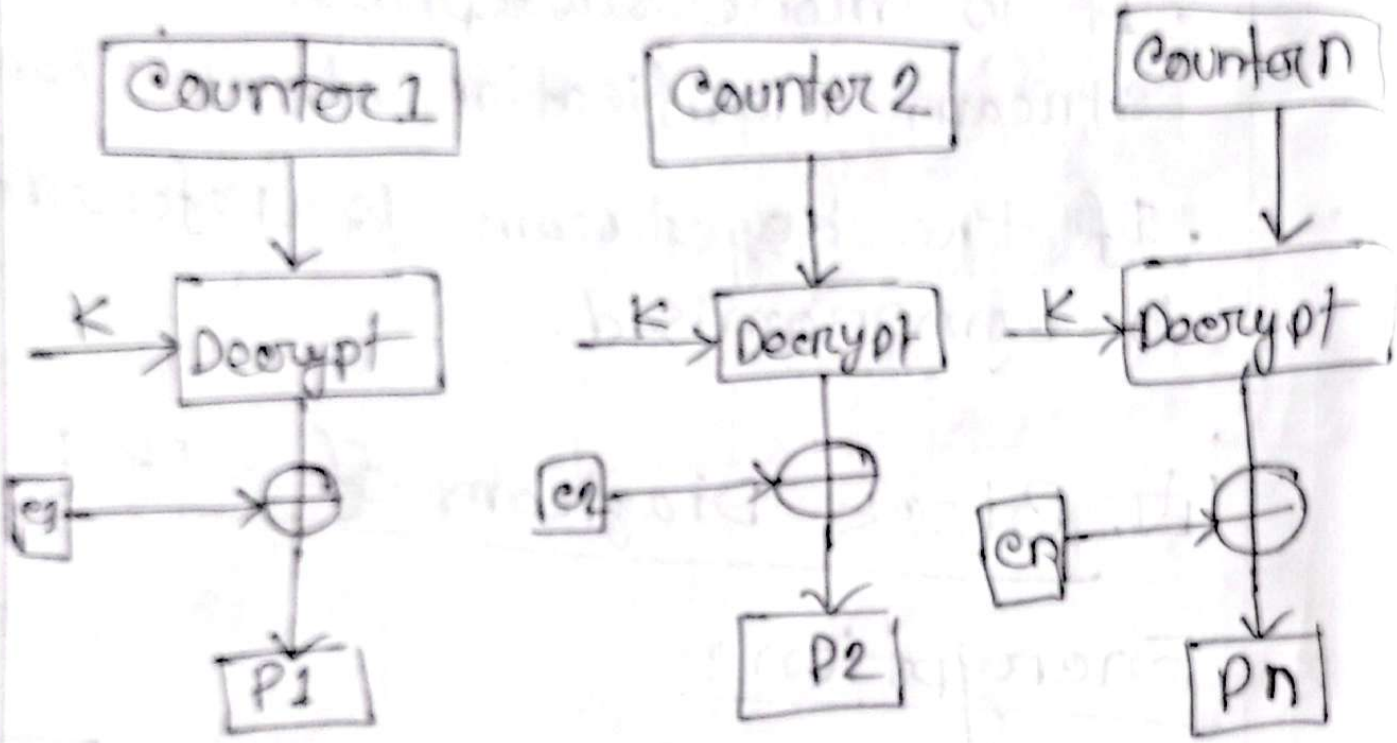
4. Block Diagram of CTR:

Encryption:



IT-210 14

Decryption:



Advantages of counter (CTR):

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

IT-21014

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 synchoronous 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)

IT-21014

Uses:

- Bitwise operations: XOR, shift rotate.
- Modular addiction

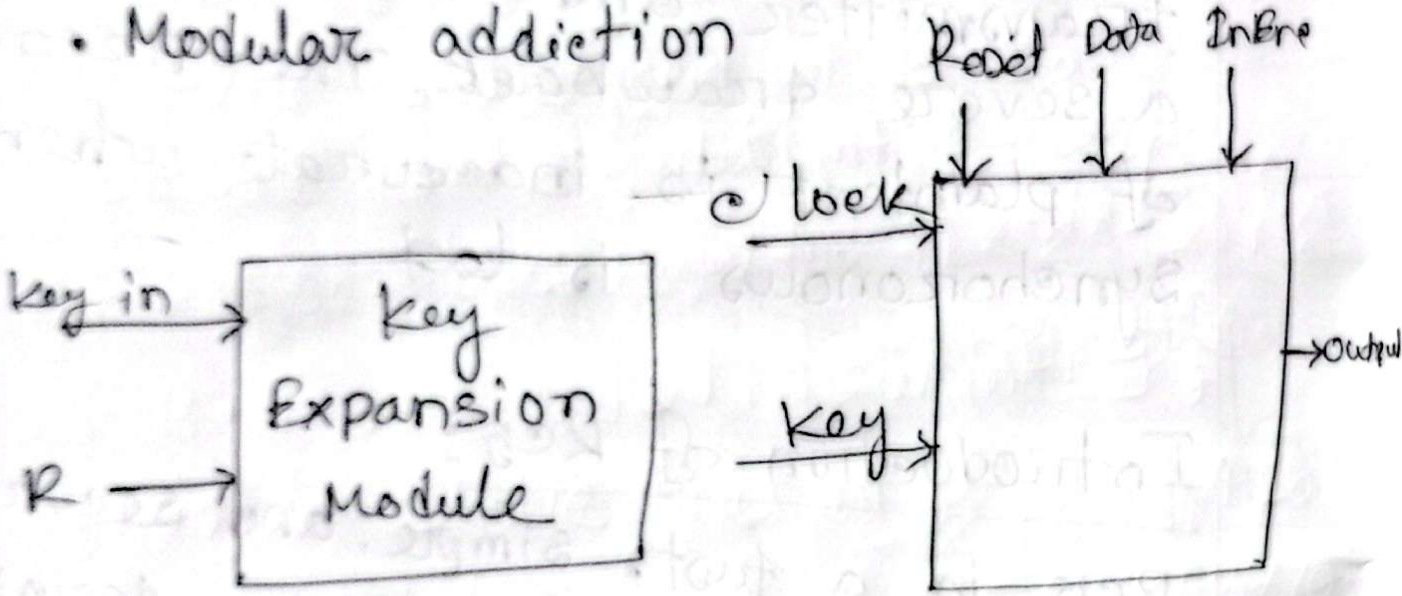
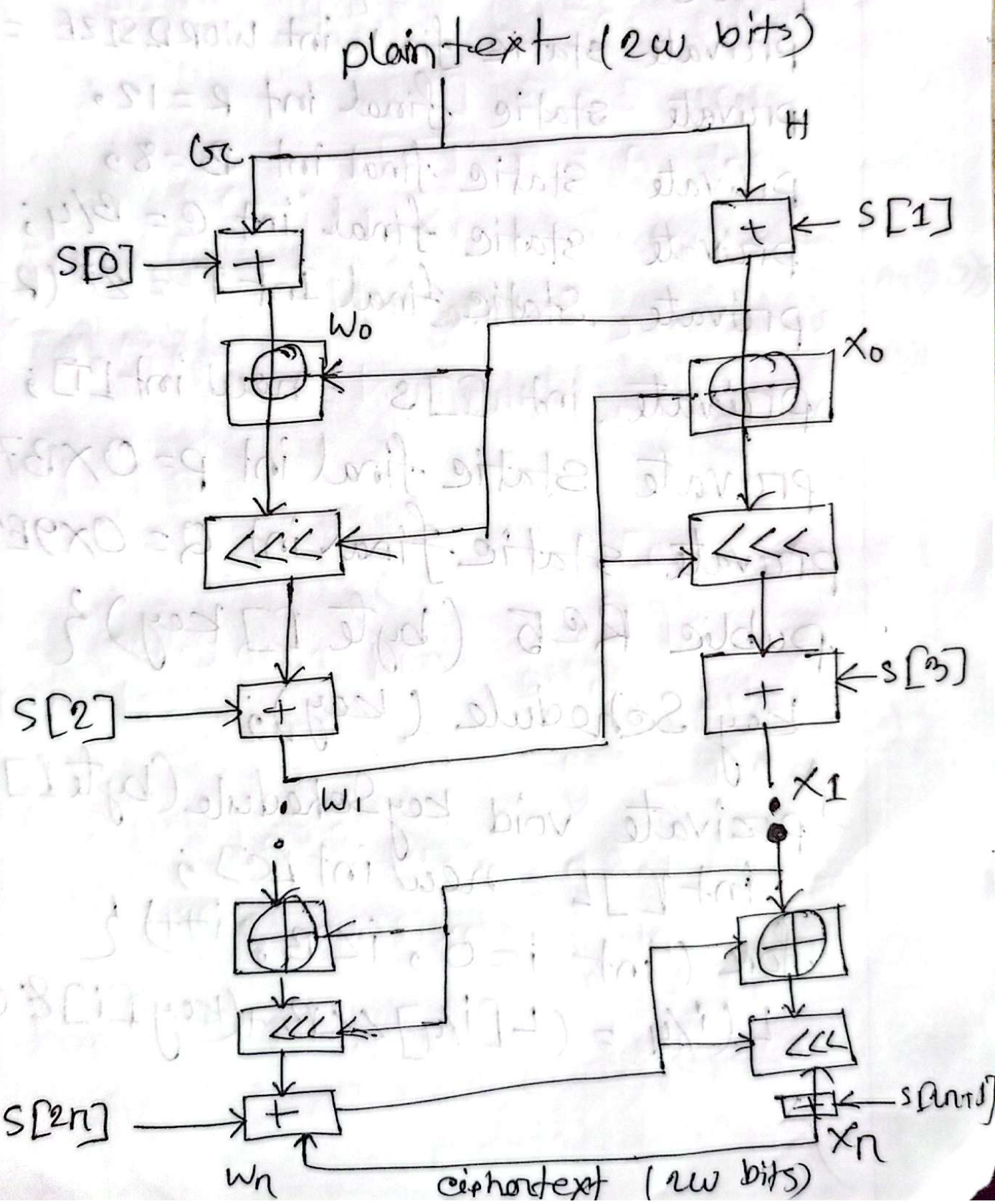


Fig: RC5 Encryption Block Diagram

IT-21014

RC5 Block Diagram:



IT-21014

Java Implementation:

```
public class RC5 {
 private static final int WORDSIZE =32;
 private static final int R=12;
 private static final int B=8;
 private static final int C= B/4;
 private static final int T = 2*(R+1);
 private int [] S = new int[T];
 private static final int P= 0XB7E15163;
 private static final int Q= 0X9E3779B9;
 public RC5 (byte [] key) {
  Key Schedule (key);
 }
 private void keySchedule (byte [] key) {
  int [] L = new int [C];
  for (int i=0; i<B, i++) {
   L(i/4 = (L[i/4] << B +(key [i] & 0XFF);
 }
```

IT-21014

```
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] = Integer.rotateLeft((S[i]+A+B),3);
B = L[j] = Integer.rotateLeft((L[j]+A+B);
                                (A+B);

    i = (i+1)%T;
    j = (j+1)%C;
}
```

IT-21014

```
public int[] encrypt (int[] pt){
    int A = pt[0]+S[0],
    int B = pt[1]+S[1];
for (int i=1, i≤R, i++){
    A= Integer.rotateLeft(A^B,B)+S[2*i],
    B= Integer.rotateLeft(B^A,A)+S[2*i]
```

IT-21014

```
    return new int [] {A,B};
}
public int [] decrypt (int [] ct){
    int B = ct [1];
    int A = ct [0];
for (int i=k , i>=1 ; i--) {
B = Integer. rotateRight (B-S[2*i+1], A) ^ A;
A = Integer. rotateRight (A-S[2*i], B) ^ B;
}
A-= S[0];
B-= S[1];
public static void main (String [] args){
byte [] key = "password". getBytes();
RC5 rc5 = new RC5 (key);
int [] pt = {0x12345678, 0x9abcdef0};
int [] ct = rc5. encrypt (pt);
```

IT-21014

IT- 21014

```
System.out.printf("Encrypted : %08X
        %08X\n", ct[0] ct[1]);

Int[] dt = rcb.decrypt(ct);
System.out.print("Decrypted: %08X
        %08X\n", dt[0], dt[1]);

}}
```

IT-21014

Sample output:

Encrypted : 7f93d8c2 1423ba29

Decrypted : 12345678 9abcdef0

Explanation:

- The output plantext is : 0X12345678 0X9abcdef0
- The encrypted ciphertext looks random.
- After decryption, we get back the exact original plaintext