Advantages:

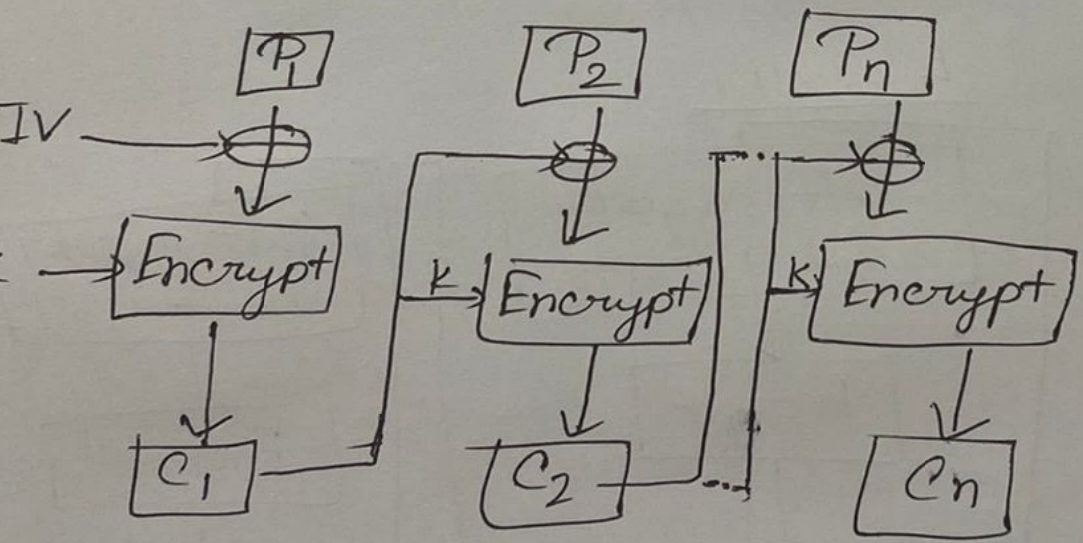
- Parallel encryption of blocks of bits is possible, thus it is a faster using of encryption.
- Simple way of the block cipher.

Disadvantages of ECB:

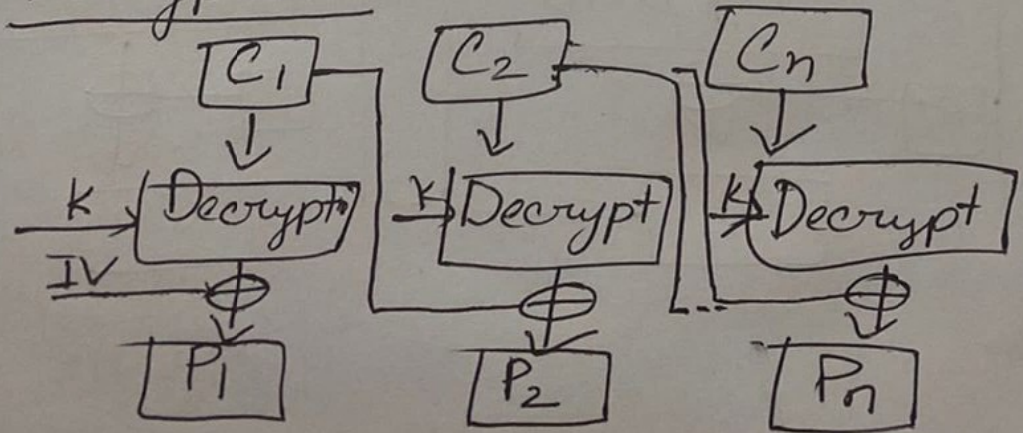
- prone to cryption of blocks of bits is possible, thus it is a faster way of encryption.
- Simple way of the block cipher.

CBC block Diagram:

Encryption:



Decryption:



ID-21052

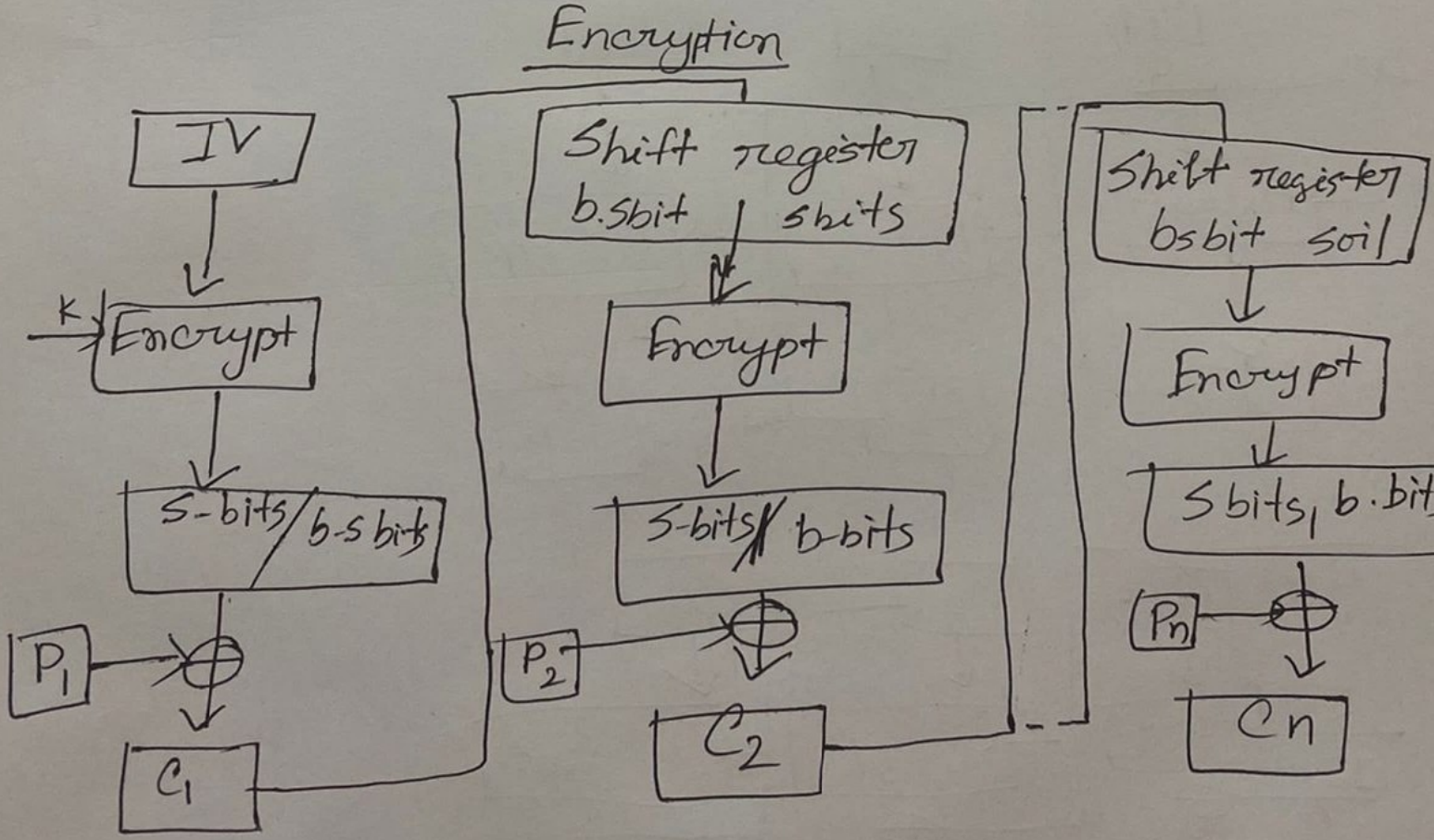
Advantages of CBC:

- 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:

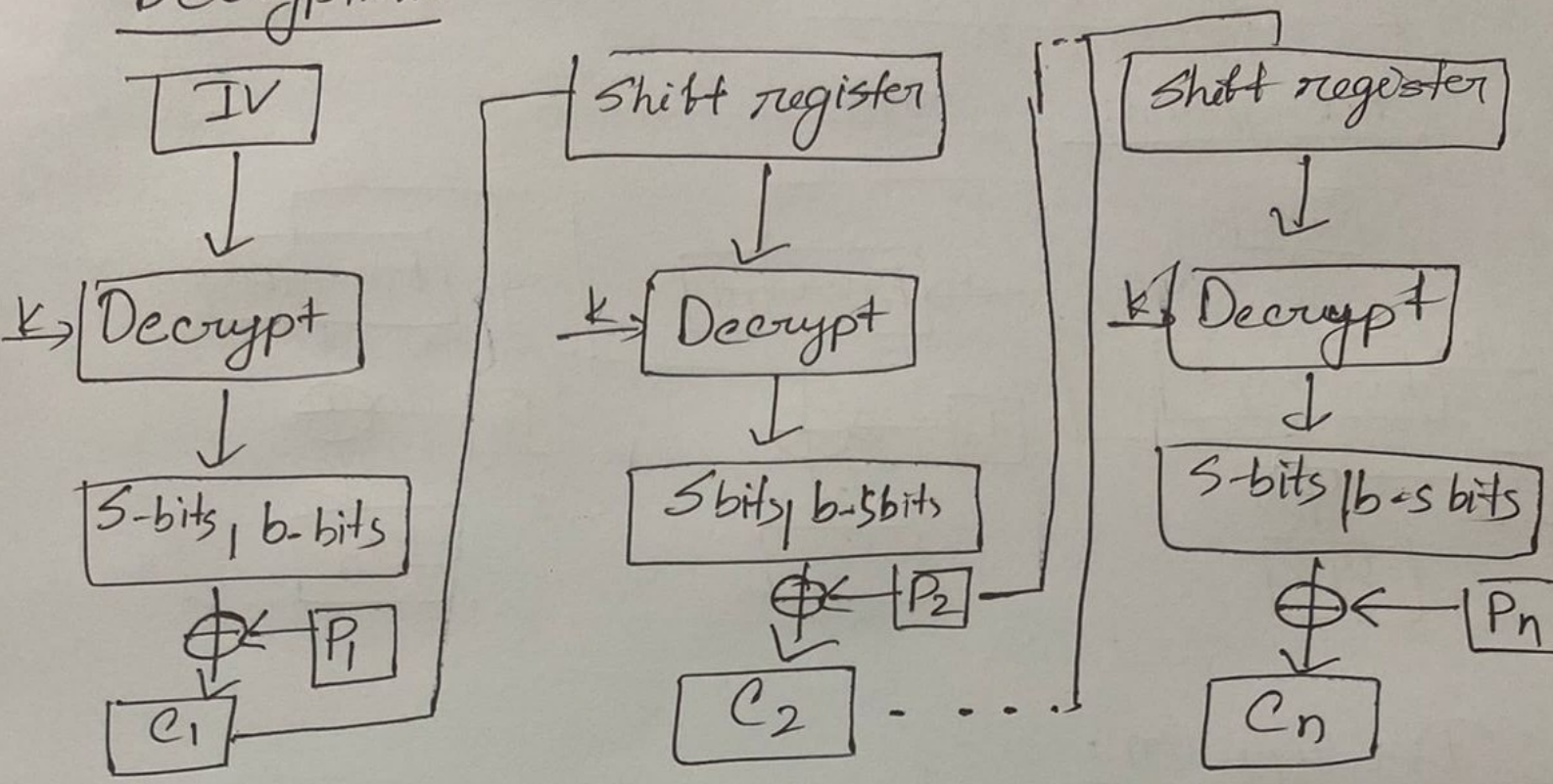
- Requires the previous cipher text book for encryption and decryption.

CFB Block diagram:



ID-21052

Decryption:



Advantages of CFB:

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

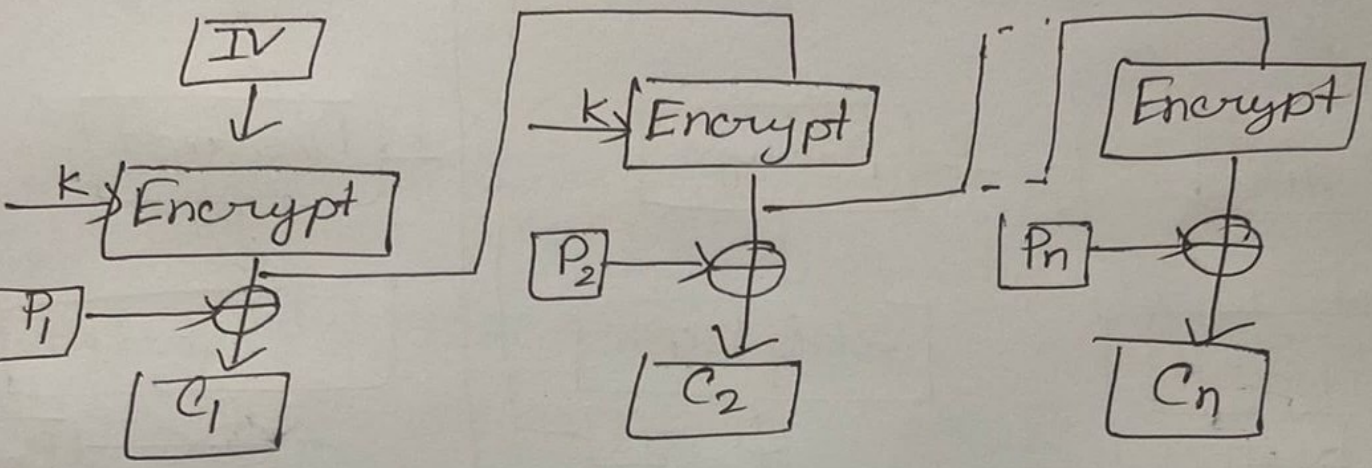
Disadvantages:

- Slightly more complex and propagate errors.

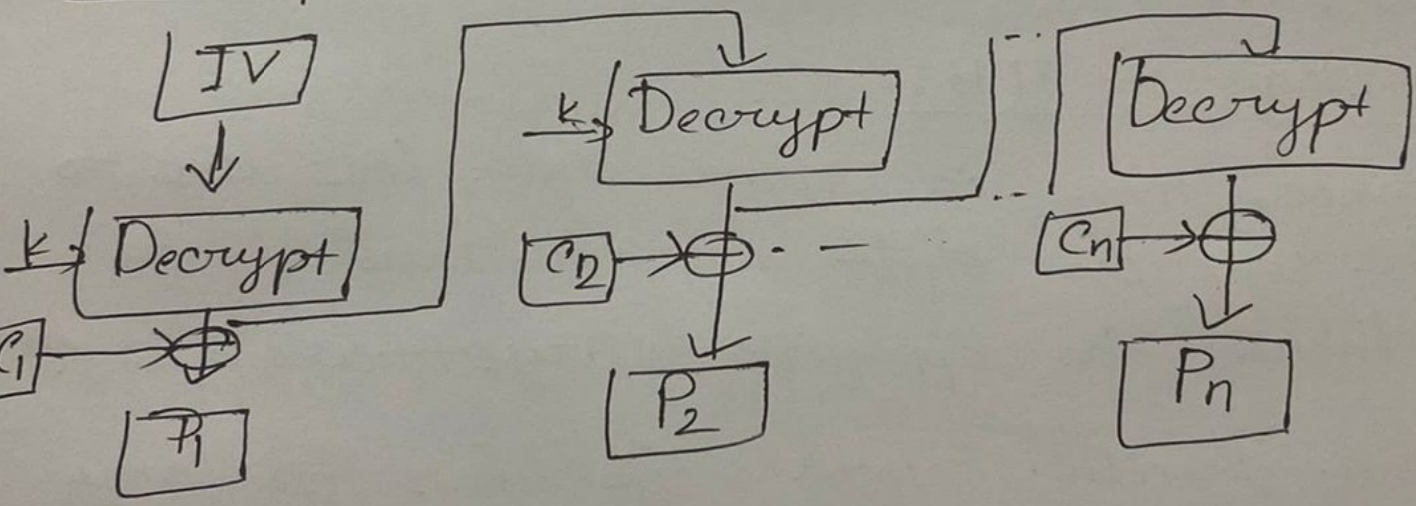
ID-21052

OFB Block Diagram:

Encryption:



Decryption:



Advantages of OCB:

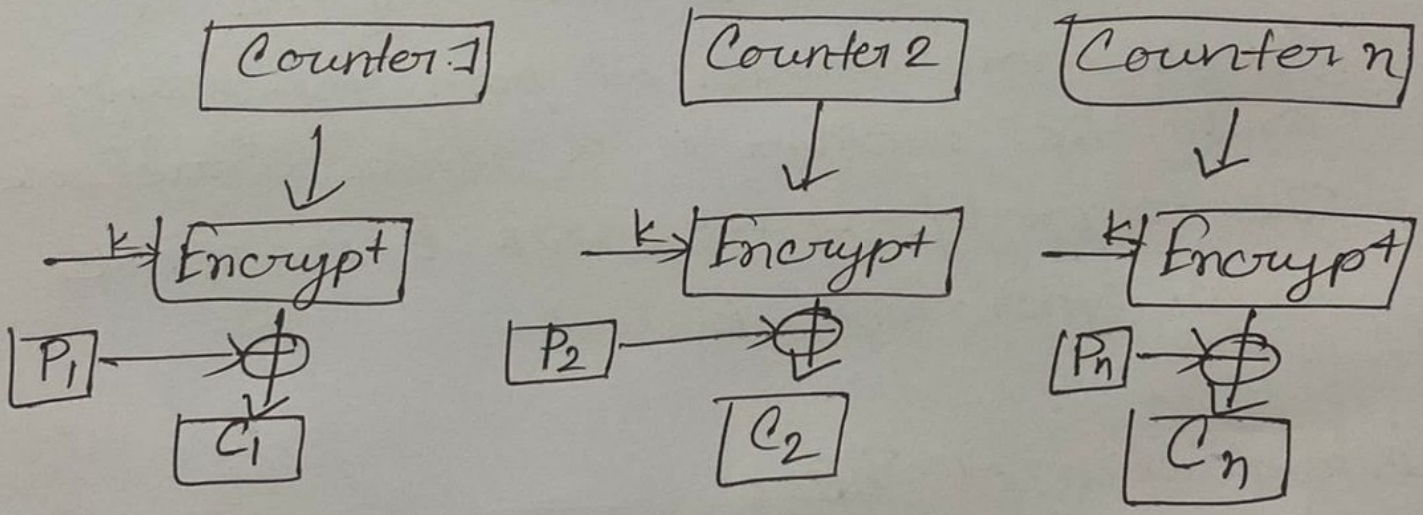
- In the case of CFB, a single bit error in a block is propageted to all subsequ blocks. This problem is solved by OFB as it is free from bit errors.

Disadvantages of OFB:

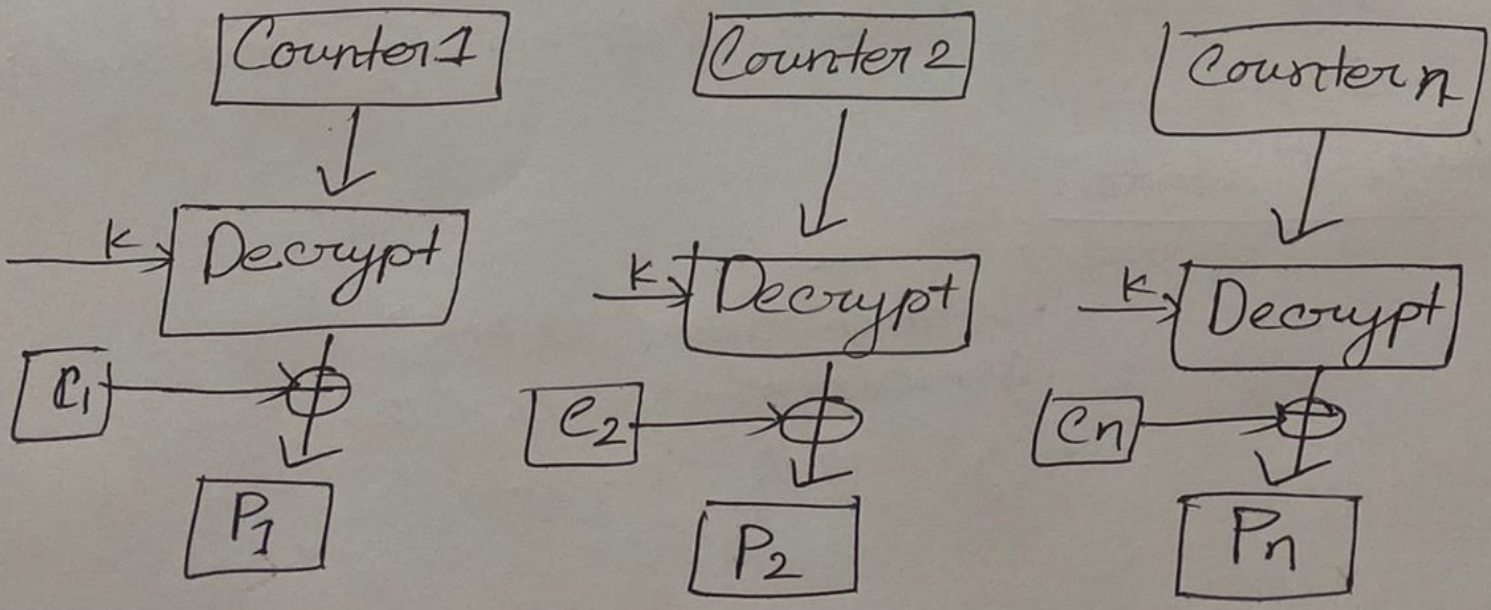
- It is more susceptible to a message stream modification attack than CFB.
- If the key stream is reused, security is compromised.

Block Diagram of CTR:

Encryption:



Decryption:



ID-21052

Advantages of ~~CTR~~ counter CTR:

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

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 addiction

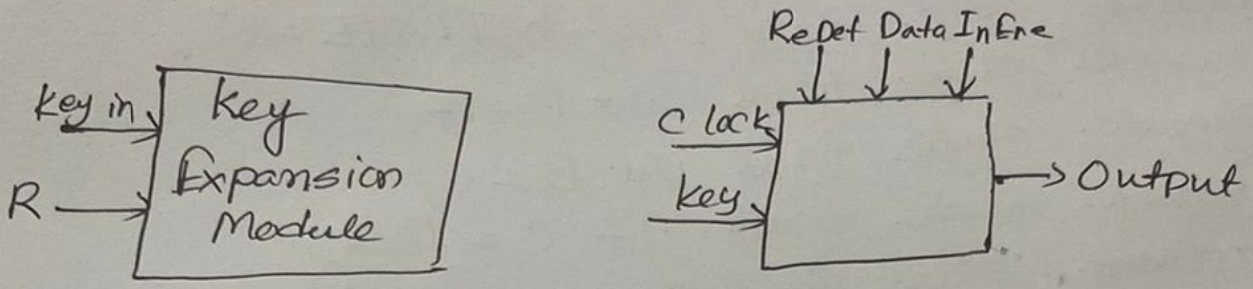
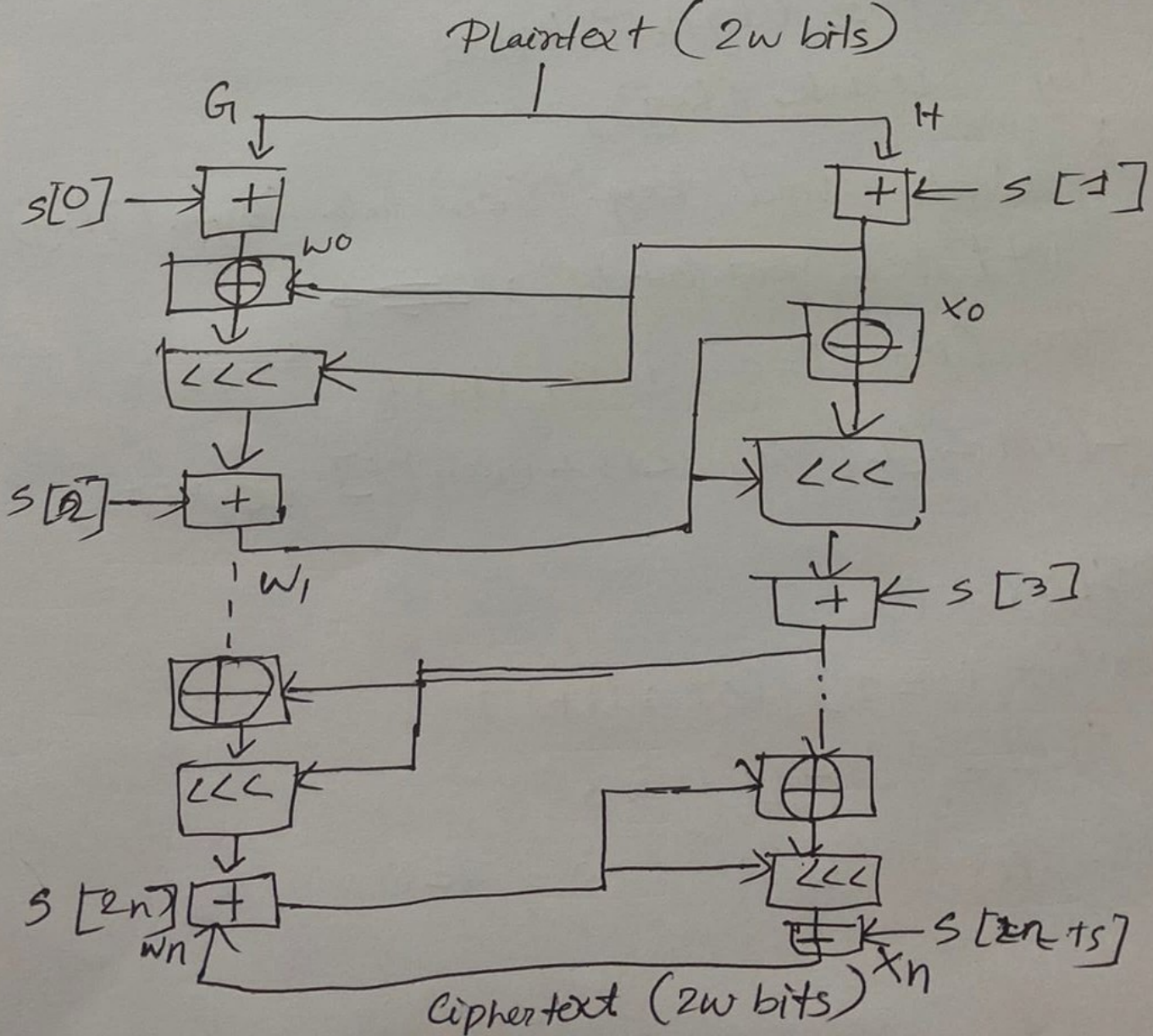


Fig : RC_5 Encryption Block Diagram

RC_5 Block Diagram:



ID-21052

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/a;
private static final Int T= 2*(R+1)n
private int [] S = new int [];
private static final int P= 0XB7E15163;
private static final int Q = 0X9E377939
Public RC5 (byte [] key){
key schedule (key),
}
private void key Schedule (byte [] key){
  int [] L = new int [c]; ~~{~~
  for (int i=0 ; i<B, i++){
L(i/A = (L[i/A] <<B + (key [i] &0 XFF);
}

  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,
```

ID-21052

```
for (int k=0, k<3*T; k++) {
A= S[i] = Integer, rotate left ((S[i]+A+B), 3);
B =L[j] = Integer, rotate left ((L(j*A+B);
                                (A+B);
  i =(i+1)% T,
  J =(J+1)%C,
}
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. rotate left (A^B.B) +5[2* i),
} B = Integer. rotate left (B^A.A) + s [2* i]
        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, rotate Right (B-S[2* i+1].A)^ A
A = Interger, rotate Right (A-S[2* i], (B)^
```

ID-21052

```
}A = S[0];
B = S[1];
public static void main [string [] args ]){
byte [] key = "password". glt Bytes (),
RC5. rc5 = new RC5 (Key);
int [] pt = {0x12345678, 0x9abcdef0},
int [] ct = RC5, encrypt (pt);
System. out. printf ("Encrypted : %.08x %.08x\n",
                                ct[0] ct[1]),
int [] dt = rc5. decrypt (ct),
System. out. print ("Decrypted : %. 08x %.08x\n"
                                dt[0]. dt[1]),
}}
```

Simple output:

Encrypted : 7193d8c2 1923ba29

Decrypted : 12345678 9abcdf0

Explanation:

- The output plaintext is : 0x12345678 0x9abcdef0
- The encrypted ciphertext looks random.
- After decryption, we get black the exact original plaintext.

Cryptography and Cyber Law

Assignment - 06

Rocktim Chakma

IT-21052

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 bit). 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.

Common modes:

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

Description:

ECB: Each block is encrypted independently No Secure for patterns.

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

ID-21052

CFB: Converts block cipher into a self Synchronizing stream cipher.

OFB: Turns block cipher into a synchoron Stream cipher.

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

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

The procedure of ECB is illustrated below: