Afsana Akter Mitu
ID: IT-21053

Assignment – 6

Q: 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 ~~obtain~~ often much longer. So we use modes of operations to securely process large data by using a block cipher repeatedly.

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.

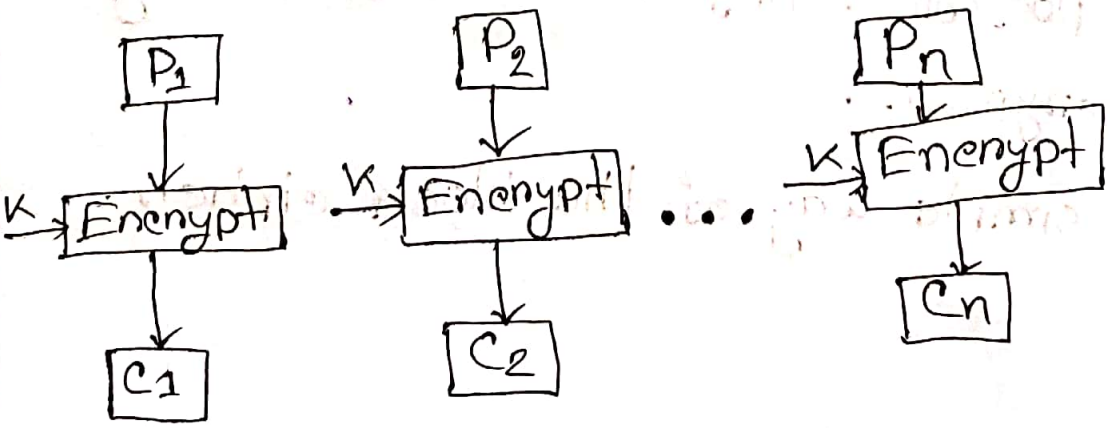
OFB: Turns block cipher into a synchronous stream cipher.

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

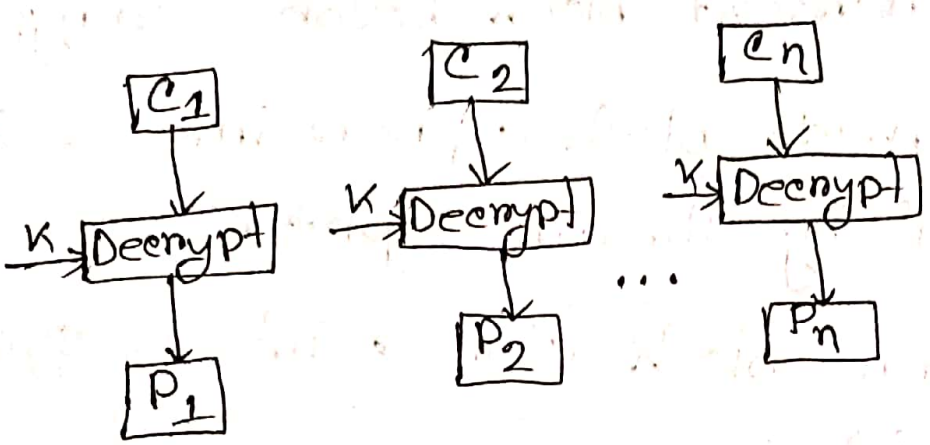
Note: Among them, CBC and CTR are the most widely used due to their security and efficiency.

The pocedure of ECB is illustrated below:

Encryption



Decryption



Advantage of ECB:

→ Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.

→ simple way of the block cipher.

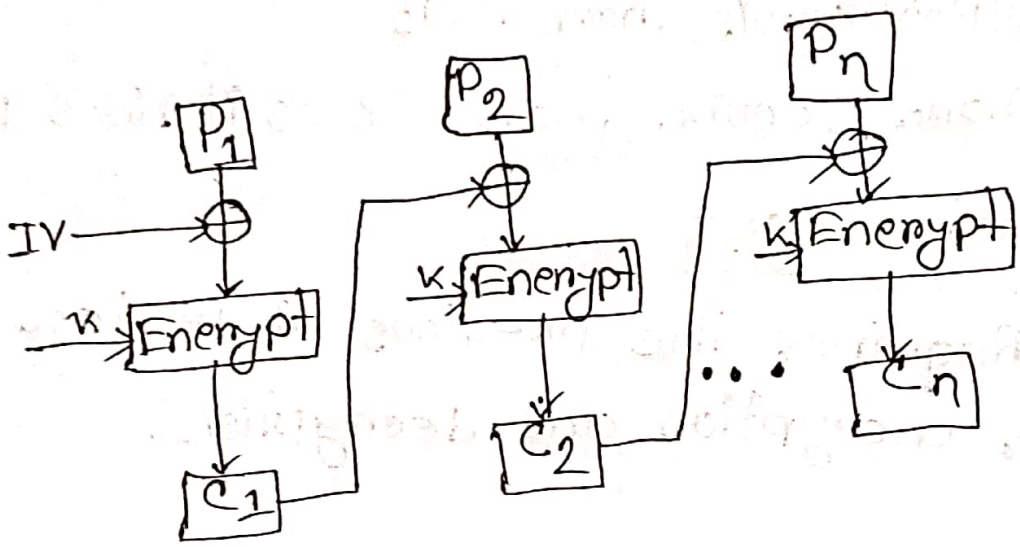
Disadvantages of ECB:

→ Prone to cryption of blocks of bits is possible, thus it is a faster way of encryption.

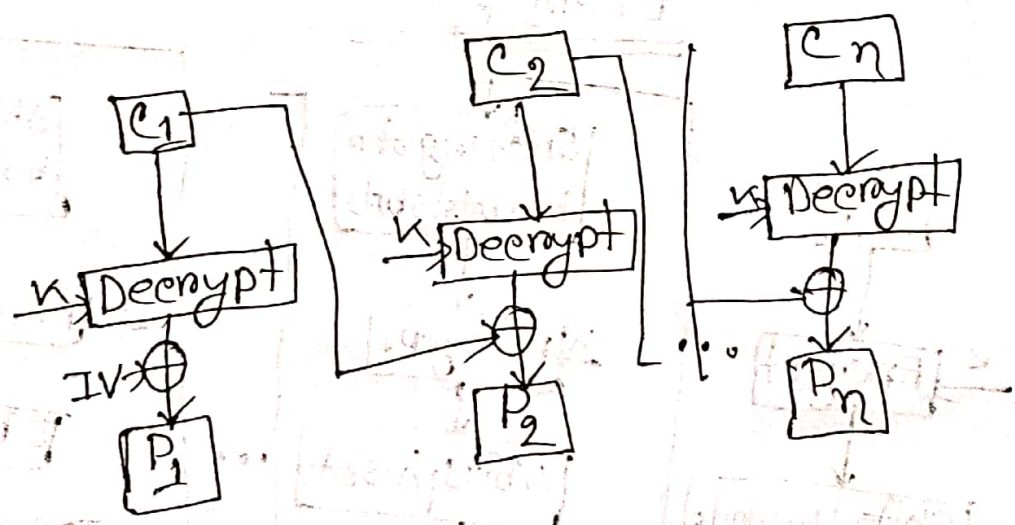
→ simple way of the block cipher.

CBC Block Diagram:

Encryption



Decryption



Advantage:

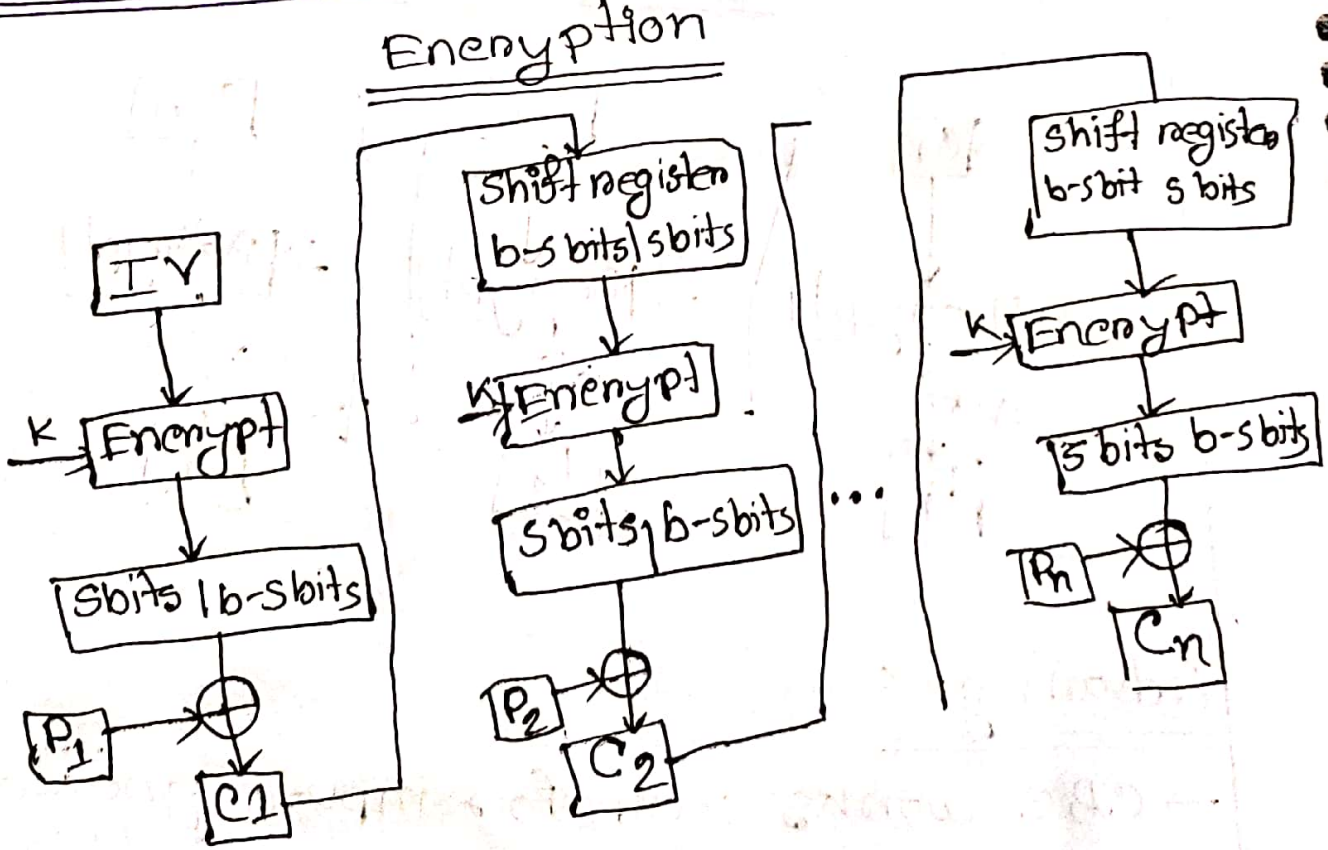
→ CBC works well for input greater than b bits.

→ CBC is a good authentication mechanism
→ Better resistive nature towards cryptanalysis than ECB
→ More secure than ECB as it hides patterns.

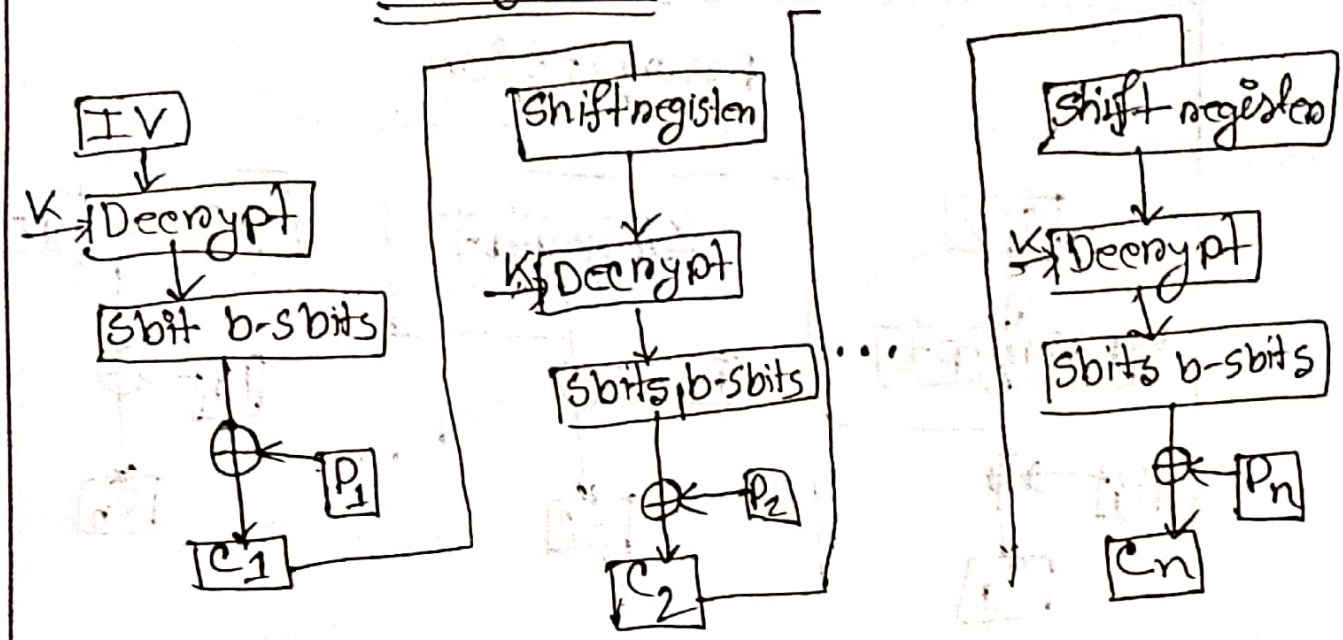
Disadvantage:

→ Requires the previous ciphertext book for encryption and decryption.

CFB Block Diagram:



Decryption



Advantage of CFB:

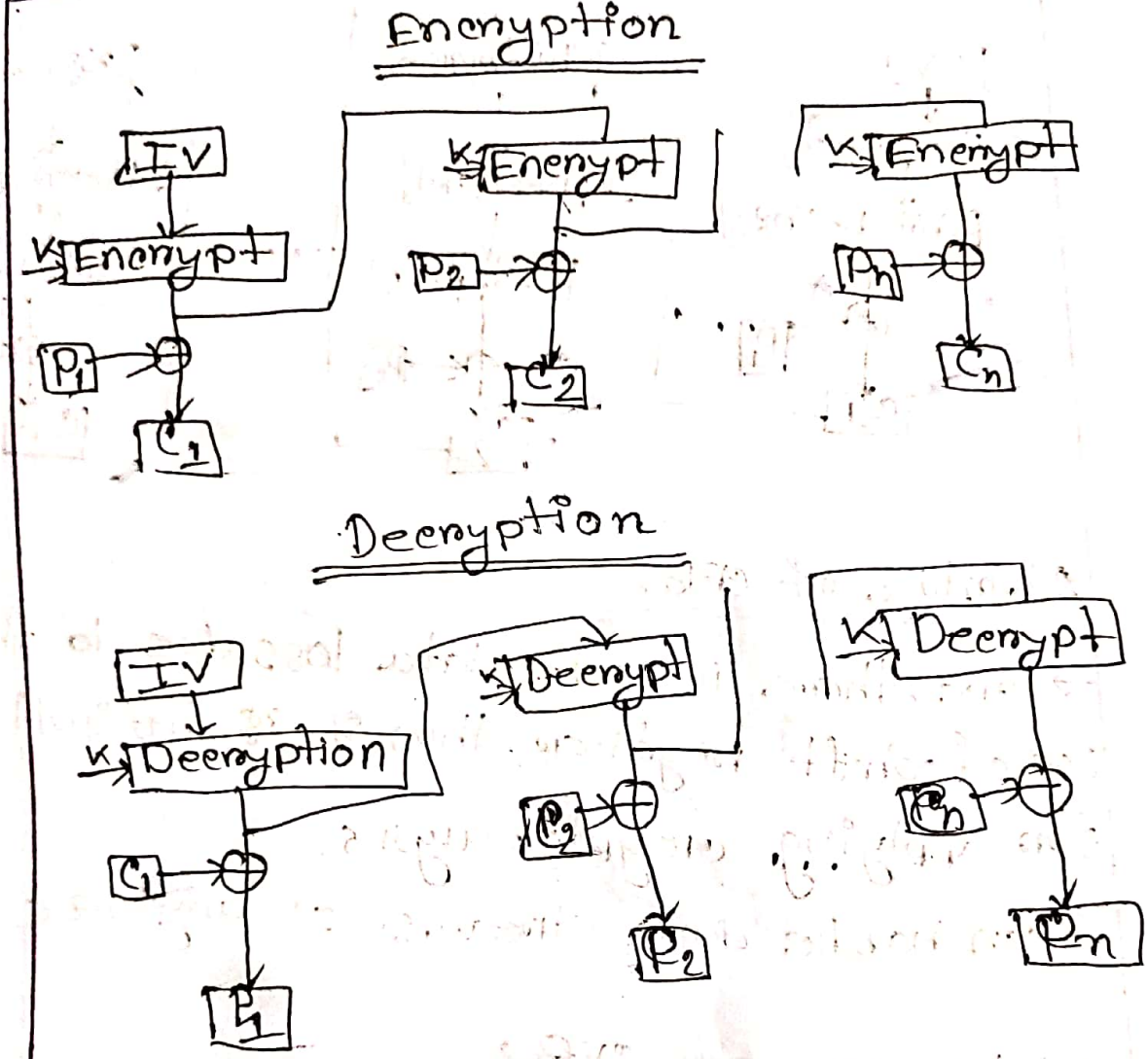
→ Since, there is some data loss due to the use of shift negester, thus it is difficult for applying cryptanalysis.

→ Can handle data streams of any size

Disadvantage of CFB:

→ Slightly more complex and can propagate errors.

OFB Block Diagram :



Advantages of OFB :

→ In the case of OFB a single bit error in a block is propagated to all subsequent blocks this problem is

solved by OFB as it is free bit error.

Disadvantages of OFB:

→ It is more susceptible to a message stream modification attack than CFB
→ If the keystream is refused security is compromised.

Introduction of RC5:

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

Key Features:

→ Parametenizable:
- word size (32 bits)
- Number of round (12)
- key length (8 bytes)

→ Uses:
- Bitwise operations: XOR, shift, rotate
- modulare addiction

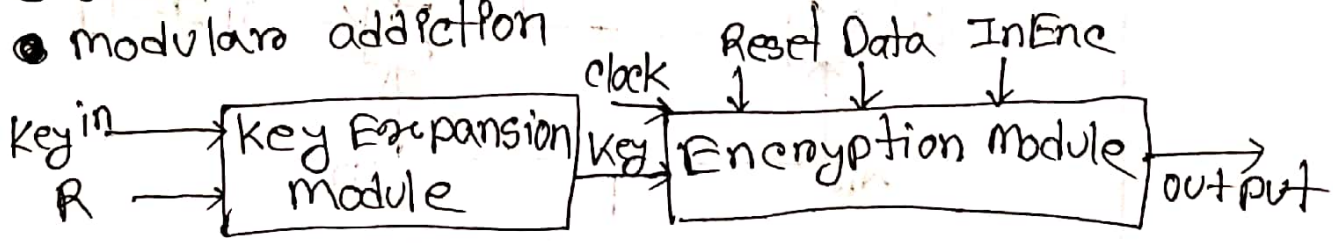
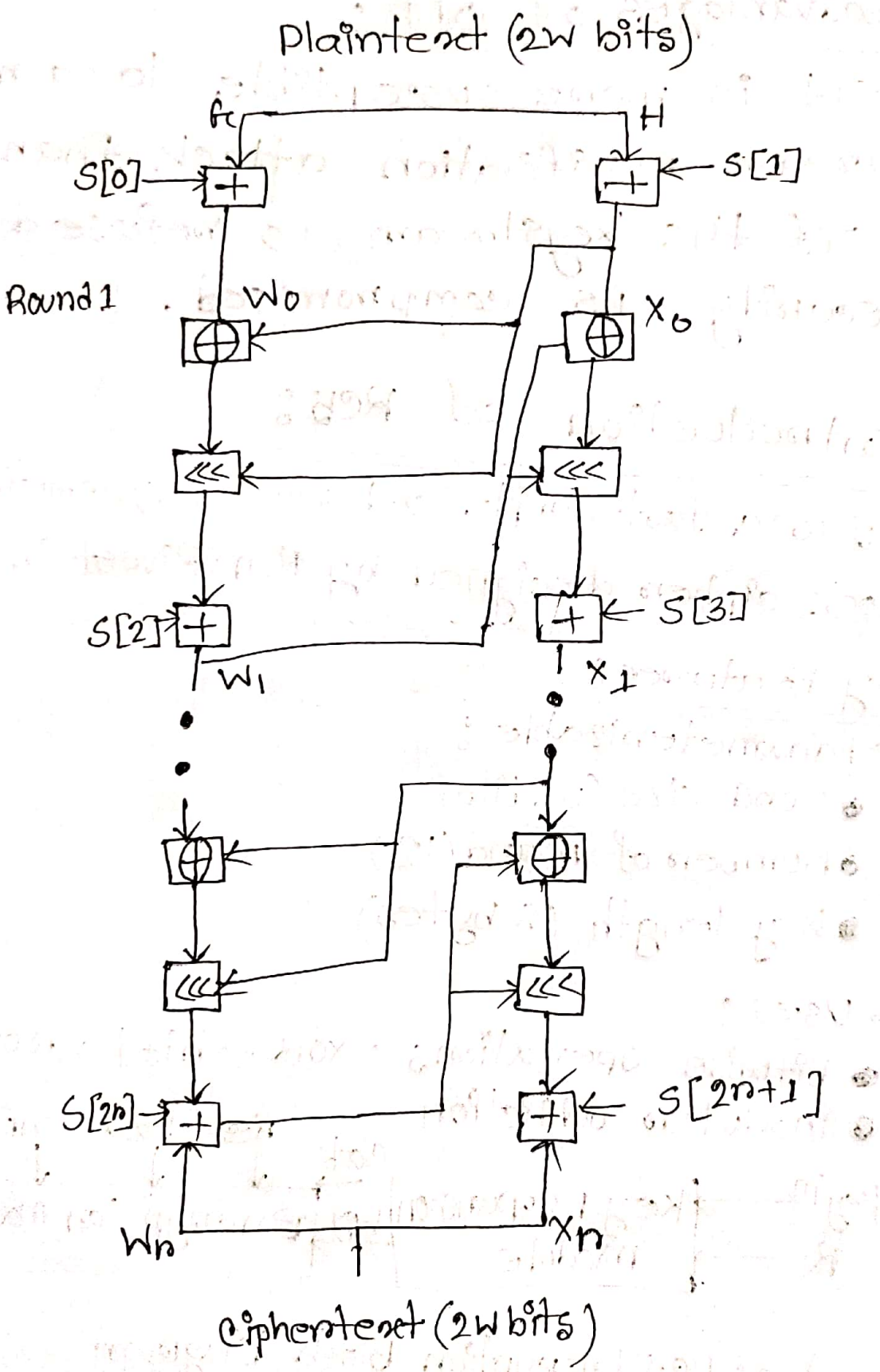


Fig: RC5 Encryption Block Diagram

RC5 Block Diagram :



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 = 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] & 0XOFF);
       }

    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*r+1; k++){
A=S[i]=Integer.rotateleft((S[i]+A+B),3);
B=L[j]=Integer.rotateleft(cL(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.rotateLeft(A^B,B)+S[2*i];
B=Integer.rotateLeft(B^A,A)+S[2*i];
}
return new int[]{A;B};
}
public int[] decrypt(int[].ct){
int B=c+[1];
int A=ct[0];
for(int i=R; i>=1; i--){
B=Integer.rotateRight.(B-S[2*i+1],A)^A;
```

```
    A =Intlegen.rotateRight (A-S[2*i], B)^B;
  }
  A = S[0];
  B = S[1];
return new int [] {A,B};
  }
public static void main (String [] args){
  byte [] key = "password".getBytes();
  Rc5 rc5 = new Rc5 (key);
  int[] pt = {0x12345978, 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]);
}
```

Sample output :

Encrypted : 7f93d8c2 1423ba29

Decrypted : 12345678 9abcdef0