Sagor Roy
IT-21044

Modes of operation and RC5-Block Diagram, Java implementation and output:-

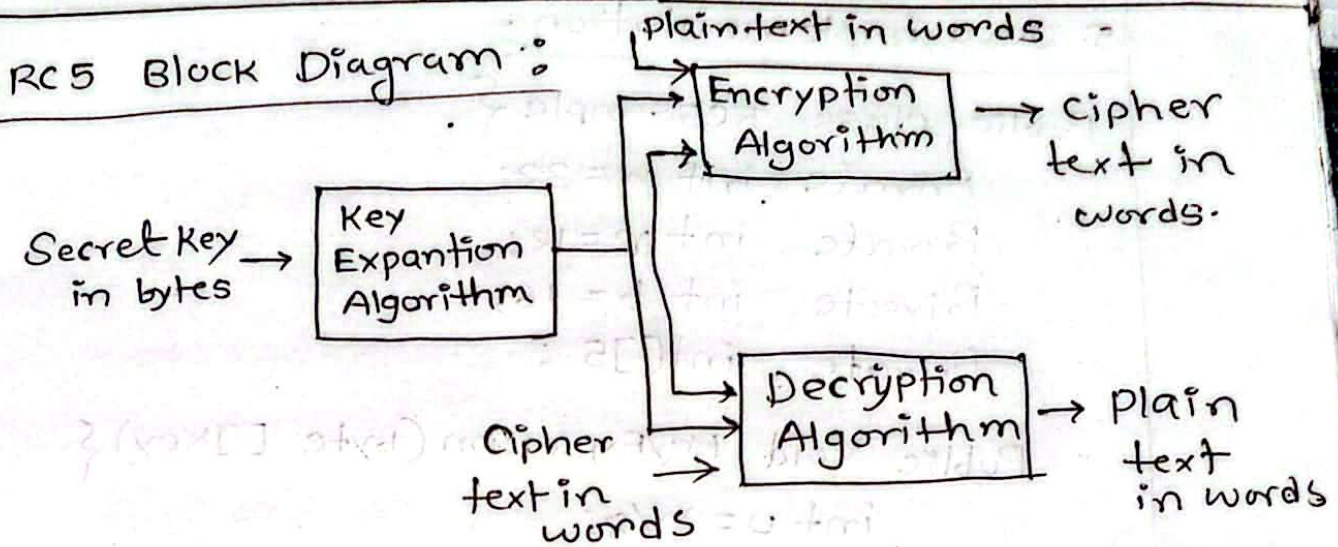
Block ciphers are used in many cryptographic systems.
RC5 is a symmetric key block encryption algorithm designed by Ronald Rivest in 1994.
Modes of operation define how block ciphers process large amounts of data beyond one block.

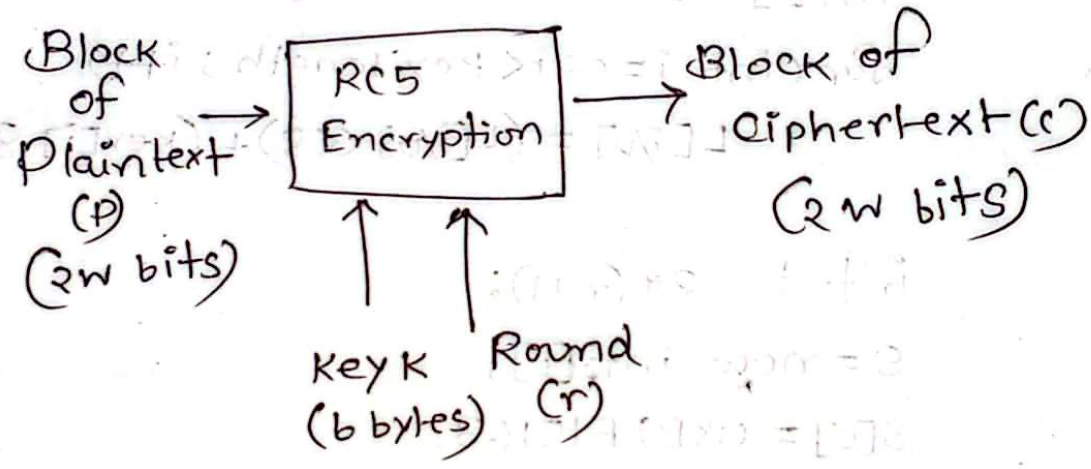
Modes of operation :

Mode	Description	Advantage	Disadvantage
ECB (Electronic codebook)	Each block is encrypted independently	Fast and Simple	Not secure for patterns
CBC (cipher Block chaining)	Each block is XORed with the Previous ciphertext block	More Secure	Needs IV, Slower
CFB (cipher Feedback)	Converts block cipher to self Synchronizing stream cipher	Can encrypt Partial blocks	Error Propagation
CFB (output FBedback)	Stream cipher using output of the block cipher	Resistant to error Propagation	Vulnerable to reus of IV

Sagor Roy
IT-21044

RC5 Block Diagram:





Block Diagram of RC5
(full and Encryption Block)

Sagor Roy
IT-21044

5. Java implementation :

```
Public class RC5Simple {
    Private int w=32;
    Private int r=12;
    Private int b=16;
    Private int[]S ;
  Public void KeyExpansion (byte []key) {
        int u= w/8;
        int c=key.length/u;
        int[] L=new int[c];
        for(int i=0; i<key.length ; i++)
              L[i/u] = (L[i/u]<<8)+(key[i] & 0xFF);

        int t= 2*(r+1);
        S=new int[t];
        S[0]=0xB7E15163;
        for (int i=1; i<t; i++)
            S[i]= S[i-1]+ 0x9E3779B9;

        int A=0, B=0, i=0, j=0;
        for(int k=0; k<3* Math.max(t,c); 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;
```

Sagor Roy
IT-21044

```
        j = (j+1)%c;
      }
    }
Public int[] encrypt(int[] plaintext) {
    int A = plaintext[0] + S[0];
    int B = plaintext[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+1]
    }
  return new int[](A, B);
  }

Public int[] decrypt(int[] ciphertext) {
    int A = ciphertext[0];
    int B = ciphertext[1];
  for(int i=r; 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];
  return new int[]{A, B};
}
```

Sagor Roy
IT-21044

```
Public static void main (string[] args) {
    RC5simple rc5 = new RC5simple();
    byte[] key = "Examplekey123456".getBytes();
    rc5.keyExpansion(key);
    int[] plaintext = {0x12345678, 0x9ABCDEF0};
    int[] ciphertext = rc5.encrypt(plaintext);
    int[] decrypted = rc5.decrypt(ciphertext);

    System.out.printf("plaintext : %08x %08x \n",
            Plaintext[0], Plaintext[1]);

    System.out.printf("ciphertext : %08x %08x\n,
            ciphertext[0], ciphertext[1]);

    System.out.printf("Decrypted : %08x %08x\n",
            decrypt[0], decrypt[1]);

}
}
```

Sagor Roy
IT-21044

Output (Sample):

Plaintext : 12345678 9ABCDEF0
Ciphertext : 1D8B5E74 29A17C40
Decrypted : 12345678 9ABCDEF0

This assignment provided a clear understandir of RC5 encryption and various block cipher modes of operation. By implementing RC5 in Java, we demonstated how encryption and decryption work in practice. It alsc highlighted the importance of choosing the right mode for secure and efficient data protection.