**Lab Practical #08:**

1. Implementation of Block Cipher mode of Operation - Cipher Block Chaining Mode (CBC).

**Program:**

import java.util.Scanner;

public class CBC {

    public static void main(String[] args) {

        String cipherTexString = "";

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Plain Text(Length : 8) : ");

        String plaintTexString = sc.nextLine();

        System.out.println("Enter Key :");

        int keyString = sc.nextInt();

        int xorPlainTextInt = 0;

        int initalVector = 2526;

        int initalVectorforDec = 2526;

        char[] ptArray = plaintTexString.toCharArray();

        for (int i = 0; i < ptArray.length; i = i+4) {

            String plaintSubString = plaintTexString.substring(i, i+4);

            if(i==0){

               int ans = initalVector ^ (Integer.parseInt(plaintSubString));

                if(ans == 0){

                    String binaryString = String.format("%04d", Integer.parseInt(Integer.toBinaryString(ans)));

                    xorPlainTextInt = Integer.parseInt(binaryString);

                }

                else{

                        xorPlainTextInt = ans;

                }

                int finaalans = xorPlainTextInt ^ (keyString);

                initalVector = xorPlainTextInt ^ (keyString);

                cipherTexString += initalVector;

            }

            else{

                int ans = initalVector ^ (Integer.parseInt(plaintSubString));

                if(ans == 0){

                    String binaryString = String.format("%04d", Integer.parseInt(Integer.toBinaryString(ans)));

                    xorPlainTextInt = Integer.parseInt(binaryString);

                    System.out.println(xorPlainTextInt);

                }

                else{

                    xorPlainTextInt = ans;

                }

                // System.out.println("Apply Encryption for ans and the key ::"+(xorPlainTextInt ^ Integer.parseInt(keyString)));

                // System.out.println("Apply Encryption for ans and the key ::"+finaalans);

                initalVector = xorPlainTextInt ^ (keyString);

                // System.out.println(initalVector);

                cipherTexString += initalVector;

            }

        }

        System.out.println("Cipher Text    :  " + cipherTexString);

        //Decryption Algorithm

        String decTextString = "";

        char[] cipherArray = cipherTexString.toCharArray();

        for (int i = 0; i < cipherArray.length; i=i+4) {

            String cipherSubString = cipherTexString.substring(i, i+4);

            if(i==0){

                int decAns = ~(Integer.parseInt(cipherSubString)^(keyString));

                decTextString += Math.abs(decAns+1)^initalVectorforDec;

                // System.out.println(Math.abs(decAns+1)^initalVectorforDec);

                initalVectorforDec = Integer.parseInt(cipherSubString);

            }

            else{

                int decAns = ~(Integer.parseInt(cipherSubString)^(keyString));

                decTextString += Math.abs(decAns+1)^initalVectorforDec;

                // System.out.println(Math.abs(decAns+1)^initalVectorforDec);

                initalVectorforDec = Integer.parseInt(cipherSubString);

            }

        }

        System.out.println("Decrypted Text :  " + decTextString);

    }

}

                initalVector = xorPlainTextInt ^ (keyString);

                cipherTexString += initalVector;

            }

        }

        System.out.println("Cipher Text    :  " + cipherTexString);

        //Decryption Algorithm

        String decTextString = "";

        char[] cipherArray = cipherTexString.toCharArray();

        for (int i = 0; i < cipherArray.length; i=i+4) {

            String cipherSubString = cipherTexString.substring(i, i+4);

            if(i==0){

                int decAns = ~(Integer.parseInt(cipherSubString)^(keyString));

                decTextString += Math.abs(decAns+1)^initalVectorforDec;

                initalVectorforDec = Integer.parseInt(cipherSubString);

            }

            else{

                int decAns = ~(Integer.parseInt(cipherSubString)^(keyString));

                decTextString += Math.abs(decAns+1)^initalVectorforDec;

                initalVectorforDec = Integer.parseInt(cipherSubString);

            }

        }

        System.out.println("Decrypted Text :  " + decTextString);

    }

}

**Output:**

PS F:\Sem-6\ins> java CBC

Enter Plain Text(Length : 8) :

12345678

Enter Key :

12345678

Cipher Text : 123484826946

Decrypted Text : 12345678