INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

CRYPTOGRAPHY

Alumnos: Caballero Huesca Carlos Eduardo

Martínez García Brando Josué

Grupo: 3CV1

Profesora: Díaz Santiago Sandra

Modos de operación: CBC, CTR

Fecha: 20 de Septiembre del 2016

Introducción

1.- Explique con sus propias palabras cómo funcionan las permutaciones.

Las permutaciones consisten en cambiar letra a letra o bit a bit de lugar en la misma cadena, así el orden de las letras cambia.

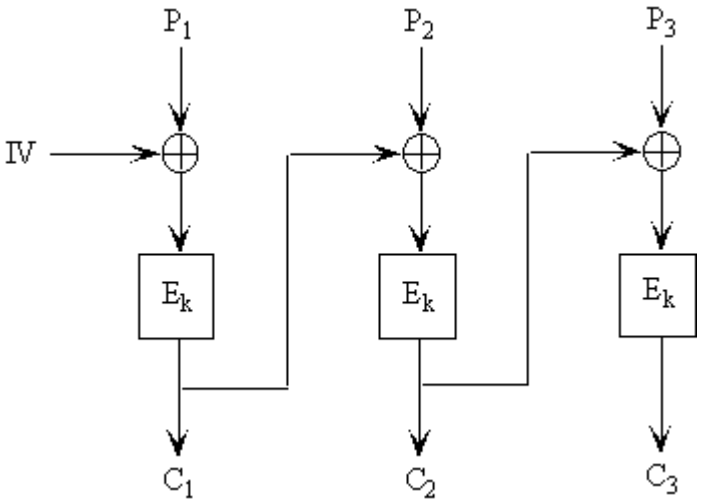
2.- Explique con sus propias palabras, cómo funciona la CBC para cifrar y descifrar.

Para cifrar.

Es necesario tener un vector de inicialización, el cual es pseudorandom y solo puede ser utilizado una sola vez.

El vector de inicialización (iv) se hará xor con el texto en plano, posteriormente el resultado entrará en un bloque de cifrado en el cual se harán normalmente permutaciones, pero no siempre. Posteriormente el resultado será el texto cifrado.

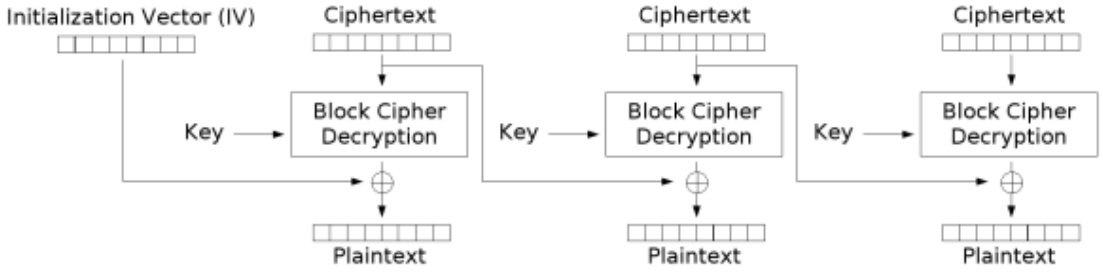
El texto cifrado del primer bloque, ocupará el lugar del vector de inicialización y se hará xor con el segundo bloque del texto plano y entrará en el bloque de cifrado y nos dará el texto cifrado y se hará así con todos los bloques restantes. [1]



Para decifrar.

Para poder descifrar el texto cifrado entrara en el bloque de cifrado juntamente con la llave el resultado se hará xor con el vector iv y el resultado será el texto claro.

Para el siguiente bloque se tomará el texto cifrado del primer bloque en lugar del vector iv y se hará xor con el resultado de meter el segundo bloque del texto cifrado y nos dará el segundo bloque del texto en plano y así se hará con todos los bloques. [2]



Cipher Block Chaining (CBC) mode decryption

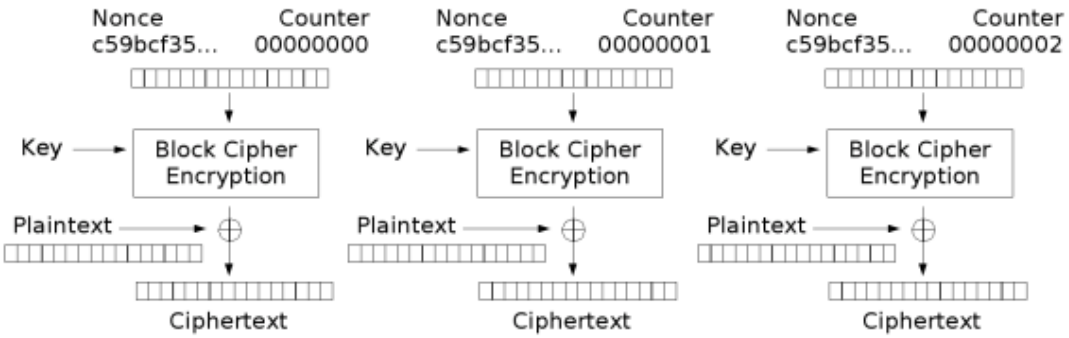
3.- Explique con sus propias palabras, cómo funciona el CTR para cifrar y descifrar.

Cifrado.

Para el cifrado con CTR es necesario un vector de inicialización el cual se tomará de manera pseudoaleatoria y solo se utiliza una vez.

Se introducirá en el bloque de cifrado, el resultado se hará xor con el texto plano y nos dará el texto cifrado.

No existe la retroalimentación y lo único que se hace es que el vector de inicialización se incrementara para poder cambiar el texto cifrado.



Counter (CTR) mode encryption

Descifrado.

Para poder descifrar es necesario el vector iv el cual entrará en el bloque de descifrado y se hará un xor con el texto cifrado y nos dará el texto plano.

Y así se hace con todos los bloques de manera sucesiva.

Funciones más importantes.

```c
void generarLLave(int longitud, char archivoLlave[]){   /*Parametros para generar la llave, Longitud y el archivo
                                                          donde se va guardar la llave*/
    FILE *llave;
    int permutacion[longitud],i,j,aux,d;

    srand(time(NULL));

    for (i=0;i<longitud;i++){                           //Permutacion de la llave
        aux=1 + rand() % longitud;
        d= 0;
        for (j=0; j<=i;j++){
            if (aux == permutacion[j]){                 //Comparamos si aux es igual a la posicion en j, si es así
                d=1;                                    //es porque exiten duplicados
            }
        }
        if (d==1){                                      //Si tenemos duplicados decrementamos i en 1
            i--;
        }
        else{
            permutacion[i] = aux;                       //A la permutacion en la posición i le asignamos
        }                                               //el valor de aux
    }

    llave = fopen(archivoLlave,"w");                    //Escribimos en el archivo el resultado
    for(i=0;i<longitud;i++){
        fprintf(llave,"%d ",permutacion[i]);
```

```c
    printf("\nSe genero con EXITO la llave en el archivo: %s\n\n",archivoLlave);
}

void leerTextoClaro(char textoClaro[]){ //GUARDA contenido en un ARREGLO y CUENTA sus caracteres

    FILE *texto;
    int i=0,j;
    char arreglo[500],lee;

    texto = fopen(textoClaro, "r");     //Abrimos el archivo de textoClaro y lo leemos

    while( fscanf(texto, "%c", &lee) != EOF ){  //Guardamos en un arreglos su contenido y contamos los caracteres
    arreglo[i] = lee;
    i++;
    }

    fclose( texto );

    printf("El mensaje del texto en plano es:\n");
    for(j=0;j<i;j++) {
    printf("%c",arreglo[j]);
    }

    }

int totalCaracteres(char textoClaro[]){
    FILE *texto;
```

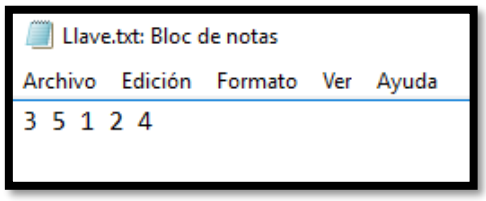
```c
int numeroBloques(int i, int longitud){
    int NoBloques;

    NoBloques = i/longitud;             //Determinamos el número de bloques a usar.

    if(i%longitud!=0){                  //Si el total de caracteres modulo longitud es diferente de 0
        NoBloques++;                    // Incremento en 1 a NoBloques
    }

    return NoBloques;
}
```

```
89
90  void divisionPorBloques(int longitud,int i,char textoClaro[]){
91
92      FILE *Bloques;
93      FILE *texto;
94      Bloques = fopen("Bloques.txt","w");
95      int huecos =0,j,k,b=0;
96      char arreglo[500],aux[500],lee;
97
98      texto = fopen(textoClaro, "r");
99      while( fscanf(texto, "%c", &lee) != EOF ){
100     aux[b] = lee;
101     b++;
102     }
103
104
105
106     huecos=longitud-(i%longitud);           //Determinamos los espacios en blanco que se tienen despues de dividir
107                                             //el mensaje en bloques
108     for(j=0;j<i;j++){
109         if(j%longitud==0){
110             printf("\n");
111         }
112         printf("%c",aux[j]);                //Separamos por bloques dependiendo de la cantidad de caracteres
113
114
115     }
```

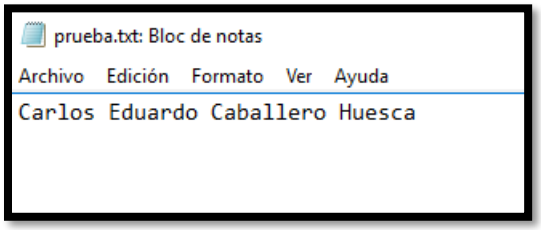
```
101     b++;
102     }
103
104
105
106     huecos=longitud-(i%longitud);           //Determinamos los espacios en blanco que se tienen despues de dividir
107                                             //el mensaje en bloques
108     for(j=0;j<i;j++){
109         if(j%longitud==0){
110             printf("\n");
111         }
112         printf("%c",aux[j]);                //Separamos por bloques dependiendo de la cantidad de caracteres
113
114
115     }
116         if(j%longitud!=0){  //2**n
117         for(k=0;k<huecos;k++){              //En los espacios en blanco restante colocamos una X
118         printf("x");
119         }
120         }
121
122
123     fclose(Bloques);
124     fclose( texto );
125
126 }
```

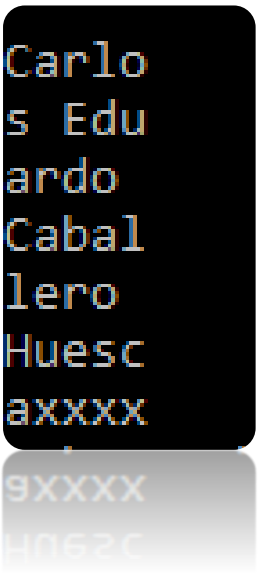
Capturas de pantalla

```
C:\Users\Carlos\Desktop\Practica 3>a Llave.txt prueba.txt
Ingresa la longitud de los bloques:5

Se genero con EXITO la llave en el archivo: Llave.txt
```




```
El mensaje del texto en plano es:
Carlos Eduardo Caballero Huesca

Total de caracteres:31
```





Bibliografía

[1] Spi1.nisu.org. (2016). Fundamentos de criptografía.. [online] Disponible en: http://spi1.nisu.org/recop/al01/skar/tema2.html [Visto 26 Sep. 2016].

[2] Modos de cifrado: ECB, O. and Lerch, D. (2007). Modos de cifrado: ECB, CBC, CTR, OFB y CFB.. [online] Dlerch.blogspot.mx. Disponible en: http://dlerch.blogspot.mx/2007/07/modos-de-cifrado-ecb-cbc-ctr-ofb-y-cfb.html [Visto 26 Sep. 2016].