



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



CRYPTOGRAPHY

Alumnos: Caballero Huesca Carlos Eduardo

Martínez García Brando Josué

Grupo: 3CV1

Profesora: Díaz Santiago Sandra

Mixcolumn

Fecha: 23 de Octubre del 2016

1.- Su información personal, la fecha de la sesión de laboratorio y el tema que estamos estudiando en esta sesión de laboratorio.

Caballero Huesca Carlos Eduardo

Martínez García Brando Josué

18 de octubre del 2016

Sesión 8

Mixcolumn

2.- Sólo las funciones más importantes de su código fuente, explicando lo que hacen.

Pasa un arreglo binario a su equivalente en decimal. Cada que encuentra un

```
40 int BinarioDecimal(int polinomio[N]){
41
42     int j, decimal=0;
43
44     for(j=0; j<N; j++){
45
46         if(polinomio[j]==0 || polinomio[j]==1){
47
48             decimal = polinomio[7-j]*pow(2,j)+decimal;
49         }
50         else{
51             printf("Los coeficientes ingresados estan fuera del espacio GF(2)\n");
52         }
53     }
54
55     return decimal;
56 }
```

Pasa un número decimal a su equivalente en binario.

```
77 void DecimalBinario(int decimal, int binario[N]){
78
79     int aux, i, arrAux[N];
80
81     for(i=0; i<N; i++){
82         arrAux[i]=0;
83         binario[i]=0;
84     }
85
86     aux = decimal;
87
88     for(i=0; decimal!=0; i++){
89         arrAux[i] = decimal%2; //RESIDUO
90         decimal = decimal/2;
91     }
92
93     for(i=0; i<N; i++){
94         binario[7-i]=arrAux[i]; //INVIERTE EL ARREGLO
95     }
96
97     for(i=0; i<N; i++){
98         printf("%d\t", binario[i]); //IMPRIME
99     }
100     printf("\n");
101
102
103 }
```

Realiza la multiplicación del polinomio .

Primero recorreremos el numero binario de cualquiera de los dos polinomios, cada que encuentre un **1** lo recorreremos n bits (dependiendo de la posición en que se encuentre el **1**) y vamos haciendo **XOR** con estos resultados.

```
105 int multiplicacionPolinomio(int polinomio[N],int polinomio2){
106
107     int aux=0,i;
108
109     for(i=0;i<N;i++){
110
111         if(polinomio[7-i]==1){
112
113             aux^=(polinomio2<<i)|
114
115         }
116     }
117
118     return aux;
119
120 }
```

Indica la posición en la que se encuentra el primer 1 al recorrer un arreglo.

Recorreremos el arreglo, cuando encuentre un **1** rompe el ciclo y retorna el valor que tiene **i** .

```
149
150 int posicionPrimerUno(int polinomio[15]){
151
152     int i;
153
154     for(i=0;i<15;i++){
155         if(polinomio[i] == 1){
156             break;
157         }
158     }
159
160     return i;
161 }
```

3.- Impresión de pantallas mostrando cómo sus programas trabajan.

```
C:\Users\Carlos\Desktop\practica06>mixcolumn.cmd
C:\Users\Carlos\Desktop\practica06>gcc mixcolumn.c fun.c -o a
C:\Users\Carlos\Desktop\practica06>a
Introduce el segundo polinomio con base al siguiente orden: (presionar TAB despues de colocar cada bit)
x7    x6    x5    x4    x3    x2    x    1:
0      0      0      0      0      1      0      1
Introduce el segundo polinomio con base al siguiente orden: (presionar TAB despues de colocar cada bit)
x7    x6    x5    x4    x3    x2    x    1:
0      0      0      0      0      1      0      1

x7    x6    x5    x4    x3    x2    x    1:
0      0      0      1      0      0      0      1
```

```
C:\Users\Carlos\Desktop\practica06>a
Introduce el segundo polinomio con base al siguiente orden: (presionar TAB despues de colocar cada bit)
x7    x6    x5    x4    x3    x2    x    1:
0      0      0      0      1      1      1      1
Introduce el segundo polinomio con base al siguiente orden: (presionar TAB despues de colocar cada bit)
x7    x6    x5    x4    x3    x2    x    1:
0      0      0      0      0      0      1      1

x7    x6    x5    x4    x3    x2    x    1:
0      0      0      1      0      0      0      1
```

```
Introduce el segundo polinomio con base al siguiente orden: (presionar TAB despues de colocar cada bit)
x7    x6    x5    x4    x3    x2    x    1:
0      1      0      0      1      0      0      1
Introduce el segundo polinomio con base al siguiente orden: (presionar TAB despues de colocar cada bit)
7     x6    x5    x4    x3    x2    x    1:
0      0      0      0      0      0      0      0

7     x6    x5    x4    x3    x2    x    1:
0      0      0      0      0      0      0      0
```

```
x7      x6      x5      x4      x3      x2      x      1:
0       0       0       0       0       0       0       0

C:\Users\Carlos\Desktop\practica06>mixcolumn.cmd

C:\Users\Carlos\Desktop\practica06>gcc mixcolumn.c fun.c -o a

C:\Users\Carlos\Desktop\practica06>a
Introduce el segundo polinomio con base al siguiente orden (presionar TAB despues de colocar cada bit)
x7      x6      x5      x4      x3      x2      x      1:
0       0       0       0       0       0       0       1
Introduce el segundo polinomio con base al siguiente orden (presionar TAB despues de colocar cada bit)
x7      x6      x5      x4      x3      x2      x      1:
0       1       1       1       0       1       1       1

x7      x6      x5      x4      x3      x2      x      1:
0       1       1       1       0       1       1       1
```