

# Operación XOR en dos mapas Renyi donde $i \neq j$ Y aplicación de pruebas NIST.

Marcos Daniel Calderón Calderón

---

## Abstract

En este reporte se presentan a pruebas NIST de dos Mapas Renyi con la restricción de que  $i \neq j$ , los resultados son combinados por medio de una operación XOR.

---

## 1. Ejemplo 1.

### 1.1. Procedimiento.

EL mapa caótico Renyi tiene la siguiente forma:

$$f(k) = \left( q2^{n-i}k + \left\lfloor \frac{k}{2^j} \right\rfloor \right) \text{ mód } 2^n \quad (1)$$

En este caso, se eligieron los siguientes parámetros:

- Mapa 1:  $i = 7, \quad j = 9$ .
- Mapa 2:  $i = 8, \quad j = 13$ .

Ahora, es necesario elegir un valor de  $q$  adecuado, en este ejemplo, utilizamos los siguientes:

- Mapa 1:  $q = 13$ .
- Mapa 2:  $q = 19$ .

Ahora, es necesario calcular el parámetro que se forma de la expresión:

$$\beta = q2^{n-i} \quad (2)$$

con base en lo anterior, se encontraron los siguientes parámetros para el mapa 1 y el mapa 2 respectivamente:

- $\beta_1 = 436207616$ .

---

*Email address:* `marcos.calderon@cimat.mx` (Marcos Daniel Calderón Calderón)

- $\beta_2 = 318767104$ .

También es importante elegir el tipo de dato que se va a utilizar para almacenar los valores generados por los mapas, en este caso se eligió el tipo de dato **unsigned long** (se utilizó un equipo con arquitectura de 32 bits, donde este tipo de dato tiene un tamaño de 32 bits).

Se generaron 80,000 valores a la hora de relacionar los valores de los dos mapas con la operación XOR. Como cada valor está formado de 32 bits, se obtiene un total de 2,560,000 bits para la aplicación de pruebas NIST.

### 1.2. Resultados.

Se utilizó el siguiente código para la ejecución de las pruebas NIST:

Se ejecutó el siguiente código:

- `./assess 2560000`
- User Prescribed Input File: **dosRenyisidifj1.dat**
- Enter 0 if you DO NOT want to apply all of the statistical tests to each sequence and 1 if you DO. Enter chice: **1**
- How many bitstreams? **1**
- Input File Format: [0] ASCII - A sequence of ASCII 0's and 1's [1] Binary - Each byte in data file contains 8 bits of data  
Select input mode: **1**

Con el procedimiento mencionado anteriormente, y con los parámetros especificados para este ejmplo 1, no se obtuvieron buenos resultados, de hecho, el archivo no pasó ninguna de las pruebas.

## 2. Ejemplo 2.

### 2.1. Procedimiento.

Para el ejemplo 2, se eligieron los siguientes parámetros:

- Mapa 1:  $i = 5$ ,  $j = 10$ .
- Mapa 2:  $i = 7$ ,  $j = 11$ .

Ahora, es necesario elegir un valor de  $q$  adecuado, en este ejemplo, utilizamos los siguientes:

- Mapa 1:  $q = 29$ .
- Mapa 2:  $q = 31$ .

Ahora, es necesario calcular el parámetro que se forma de la expresión:

$$\beta = q2^{n-i} \quad (3)$$

con base en lo anterior, se encontraron los siguientes parámetros para el mapa 1 y el mapa 2 respectivamente:

- $\beta_1 = 3892314112$ .
- $\beta_2 = 1040187392$ .

## 2.2. Resultados.

Se utilizó el siguiente código para la ejecución de las pruebas NIST:

Se ejecutó el siguiente código:

- `./assess 2560000`
  - User Prescribed Input File: **dosRenyisidifj2.dat**
  - Enter 0 if you DO NOT want to apply all of the statistical tests to each sequence and 1 if you DO. Enter chice: **1**
  - How many bitstreams? **1**
  - Input File Format: [0] ASCII - A sequence of ASCII 0's and 1's [1] Binary - Each byte in data file contains 8 bits of data
- Select input mode: **1**

Con el procedimiento mencionado en el ejemplo 2, y con los parámetros especificados, no se obtuvieron buenos resultados, de hecho, el archivo no pasó ninguna de las pruebas.

## 3. Conclusiones.

Cuando  $i \neq j$ , se obtienen ciclos muy cortos, esto provoca que las pruebas NIST fallen. Quizá se deba a que no se cumpla el criterio de invertibilidad. Cuando la muestra no es de calidad, las pruebas NIST fallan.

## 4. Anexos.

### 4.1. Código del ejemplo 1.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```

#define RENYI_MAP(var, parametro, j) ((var)*(parametro)+((var)>>(j)))
#define ITtotales 80000 /*Iteraciones totales para NIST.*/

/*
File:    main.c
Author: daniel
En este programa, se ejecutan de manera simultánea dos mapas de tipo Renyi,
estos no
son acoplados, las salidas de estos mapas se utilizan para una
operacion XOR.
*/

unsigned long calcular_parametro(unsigned long q, unsigned int n, unsigned int i){
    unsigned long potencia =1;
    unsigned int j;
    /*Calculo de la potencia (calculamos 2(a la)( n-i) ): */
    for(j =0; j<(n-i); j++){
        potencia*=2;
    }

    /*Ahora, lo que hacemos es la operacion q*2(a la )(n-i): */
    unsigned long parametro = q*(potencia);
    printf("  EL parametro es: %lu \n \n \n",parametro);
    return parametro;
}

int main(){

    /*Declaramos los arreglos que vamos a utilizar para guardar esto.
    EL tipo de dato será
    unsigned int, cuyo tamaño es de 34 bits (arquitectura de 64 bits).*/
    unsigned long Xtotal[ITtotales];
    FILE *  archivobin;
    unsigned long Xn1 = 7;
    unsigned long Xn2 = 9;
    unsigned int n = 32;

    /*Valores de i, j y q para el mapa 1.*/
    unsigned int i1=7;

```

```

unsigned int j1=9;
unsigned long q1=13;
/*Valores de i, j y q para el mapa 2.*/
unsigned int i2=8;
unsigned int j2=13;
unsigned long q2=19;

/*Calculo de los parametros para cada mapa: 1 y 2 respectivamente.*/
unsigned long param1 =calcular_parametro(q1, n, i1);
unsigned long param2 =calcular_parametro(q2, n, i2);

unsigned int iteraciones=0;
unsigned int IT = 80000;

/* Apertura del fichero de destino, para escritura en binario.*/
archivobin = fopen ("dosRenyisidifj1.dat", "wb");
if (archivobin==NULL)
{
perror("No se puede abrir dosRenyisidifj1.dat");
return -1;
}

printf("\n\n\n Operacion XOR en dos Renyis \n EL tamaño de unsigned long
    en maquina de 32 bits es:  %d", sizeof(long));

while (iteraciones < IT) {

    Xn1= RENYI_MAP(Xn1,param1,j1);
    Xn2= RENYI_MAP(Xn2,param2,j2);
    Xtotal[iteraciones++] = Xn1^Xn2;

    //printf("\nla iteracion %u para %u H es: \n",iteraciones-1, Xtotal[iteraciones-1]);

}

/*Escribimos la informacion.*/
fwrite(Xtotal,4,80000,archivobin);

if(!fclose(archivobin)){
    printf( "\nArchivo binario cerrado\n" );
}

```

```

    }
    else{
        printf( "\nError: Archivo binario no cerrado \n" );
        return 1;
    }

return 0;
}

```

#### 4.2. Código del ejemplo 2.

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

#define RENYI_MAP(var, parametro, j) ((var)*(parametro)+((var)>>(j)))
#define ITtotales 80000 /*Iteraciones totales para NIST.*/

```

```

/*
File:    main.c
Author: daniel
En este programa, se ejecutan de manera simultánea dos mapas de tipo Renyi,
estos no
son acoplados, las salidas de estos mapas se utilizan para una
operacion XOR.
*/

```

```

unsigned long calcular_parametro(unsigned long q, unsigned int n, unsigned int i){
    unsigned long potencia =1;
    unsigned int j;
    /*Calculo de la potencia (calculamos 2(a la)( n-i) ): */
    for(j =0; j<(n-i); j++){
        potencia*=2;
    }

    /*Ahora, lo que hacemos es la operacion q*2(a la )(n-i): */
    unsigned long parametro = q*(potencia);

```

```

    printf("  EL parametro es: %lu \n \n \n",parametro);
    return parametro;
}

int main(){

    /*Declaramos los arreglos que vamos a utilizar para guardar esto.
    EL tipo de dato será
    unsigned int, cuyo tamaño es de 34 bits (arquitectura de 64 bits).*/
    unsigned long Xtotal[ITtotales];
    FILE *  archivobin;
    unsigned long Xn1 = 7;
    unsigned long Xn2 = 9;
    unsigned int n = 32;

    /*Valores de i, j y q para el mapa 1.*/
    unsigned int i1=5;
    unsigned int j1=10;
    unsigned long q1=29;
    /*Valores de i, j y q para el mapa 2.*/
    unsigned int i2=7;
    unsigned int j2=11;
    unsigned long q2=31;

    /*Calculo de los parametros para cada mapa: 1 y 2 respectivamente.*/
    unsigned long param1 =calcular_parametro(q1, n, i1);
    unsigned long param2 =calcular_parametro(q2, n, i2);

    unsigned int iteraciones=0;
    unsigned int IT = 80000;

    /* Apertura del fichero de destino, para escritura en binario.*/
    archivobin = fopen ("dosRenyisidifj2.dat", "wb");
    if (archivobin==NULL)
    {
        perror("No se puede abrir dosRenyisidifj2.dat");
        return -1;
    }

    printf("\n\n\n Operacion XOR en dos Renyis \n EL tamaño de unsigned long en maquina

```

```

while (iteraciones < IT) {

    Xn1= RENYI_MAP(Xn1,param1,j1);
    Xn2= RENYI_MAP(Xn2,param2,j2);
    Xtotal[iteraciones++] = Xn1^Xn2;

    printf("\nla iteracion %u      result:  %lu  \n",iteraciones-1, Xtotal[it

}

/*Escribimos la informacion.*/
fwrite(Xtotal,4,80000,archivobin);

if(!fclose(archivobin)){
    printf( "\nArchivo binario cerrado\n" );
}
else{
    printf( "\nError: Archivo binario no cerrado \n" );
    return 1;
}

return 0;
}

```