# Operación XOR de manera cruzada en mapas Renyi donde $i = j$, aplicación de pruebas NIST.

Marcos Daniel Calderón Calderón
Maestría en Ciencias de la Computación
Centro de Investigación en Matemáticas (CIMAT)
Guanajuato , Gto.
marcos.calderon@cimat.mx

***Resumen*—En este reporte se explica de manera detallada el funcionamiento de una operación XOR de manera cruzada entre la parte inferior y superior de dos números por mapas caóticos Renyi.**

## I. Introducción.

EL mapa caótico Renyi tiene la siguiente forma:

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

.

Ahora, para facilitar la explicación, supongamos que estamos trabajando con datos de 8 bits. Esto significa que cada número se puede dividir en dos partes de 4 bits, la parte izquierda es la más significativa, la parte derecha es la menos significativa. Supongamos que vamos a trabajar con los siguientes datos:

$$x_1 = 103 \quad (01100111) \qquad x_2 = 89 \quad (01011001) \tag{2}$$

También, necesitamos un valor auxiliar:

$$a = 15 \qquad (00001111) \tag{3}$$

El esquema que se manejará es el siguiente:

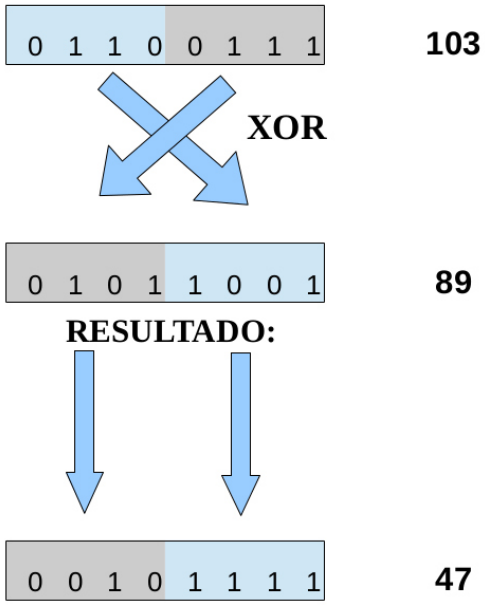


Figura 1. Esquema de intercambio.

Un código simplificado (para ocho bits) que hace la operación anterior es el siguiente:

```
char a = 15;
char temp;
char temp1;
char temp2;
char Xn1;
char Xn2;
char Xn3;
Xn1=103;
Xn2= 89;
        temp1 = Xn2 & a;
        temp2 = Xn1 >> 4;
        temp = temp1^temp2;
        temp1 = Xn2 >> 4;
        temp2 = Xn1 & a;
```

```
Xn3=(temp1^temp2)<<4;
Xn3|=temp;
```

Ahora, para los ejemplos que se muestran aquí se utilizan 32 bits, esto significa que se van a dividir los datos generados por los mapas caóticos en dos partes: cada una de 16 bits. También, en este caso, necesitamos un nuevo valor para a: $(a = 2^{16} - 1 = 65,535)$

## II. Ejemplo 1.

### *II-A. Procedimiento.*

Para que el mapa sea invertible, es necesario que se cumpla la siguiente condición:

$$0 < i = j \leq n \tag{4}$$

por lo tanto, hemos elegido los siguientes parámetros:

- Mapa 1: $i = j = 9$.
- Mapa 2: $i = j = 13$.

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:

$$\beta = q2^{n-i} \tag{5}$$

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

- $\beta_1 = 109051904$.
- $\beta_2 = 9961472$.

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.

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

Se ejecutó el siguiente código:

- **./assess 2560000**
- User Prescribed Input File: **dosRenyisINtercambio1.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**

### *II-B. Resultados.*

Los resultados obtenidos son los siguientes:

Cuadro I. Primeros 3 datos obtenidos en el ejemplo 1.

| Primeros tres valores generados con el esquema propuesto. | | |
|---|---|---|
| Primer valor generado | | |
| Especificación | Base 10 | Representación binaria |
| Valor mapa 1 | 1417674752 | 0101010010000000 0000000000000000 |
| Valor mapa 2 | 189267968 | 0000101101001000 0000000000000000 |
| Resultado | 189289600 | 0000101101001000 0101010010000000 |
| Segundo valor generado | | |
| Especificación | Base 10 | Representación binaria |
| Valor mapa 1 | 2768896 | 0000000000101010 0100000000000000 |
| Valor mapa 2 | 23104 | 0000000000000000 0101101001000000 |
| Resultado | 1073764970 | 0100000000000000 0101101001101010 |
| Tercer valor generado | | |
| Especificación | Base 10 | Representación binaria |
| Valor mapa 1 | 5408 | 0000000000000000 0001010100100000 |
| Valor mapa 2 | 2516582402 | 1001011000000000 0000000000000010 |
| Resultado | 2199912450 | 1000001100100000 0000000000000010 |

Cuadro II. Resultados de las pruebas de aleatoriedad NIST a los datos dosRenyisIntercambio1.dat .

| Prueba Aplicada | P-Valor | Exito? |
|---|---|---|
| Aproximate Entropy | 0.128865 | √ |
| Block Frecuency | 0.404371 | √ |
| Cumulative Sums | F: 0.503743, R: 0.542327 | √ |
| FFT | 0.679644 | √ |
| Frecuency | 0.963111 | √ |
| Linear Complexity | 0.778769 | √ |
| Longest Run | 0.455532 | √ |
| Non Overlapping Template | 146 de 148 | √ |
| Overlapping Template | 0.089385 | √ |
| Random Excursions | 7 de 8 | √ |
| Random Excursions Variant | 18 de 18 | √ |
| Rank | 0.699856 | √ |
| Runs | 0.950166 | √ |
| Serial | 2 de 2 | √ |
| Universal | 0.086113 | √ |

## III. Ejemplo 2.

### *III-A. Procedimiento.*

Para el ejemplo 2, se eligieron los siguientes parámetros:

- Mapa 1: $i = j = 5$.
- Mapa 2: $i = j = 14$.

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:

$$\beta = q2^{n-i} \tag{6}$$

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

- $\beta_1 = 3892314112$.
- $\beta_2 = 8126464$.

### *III-B. Resultados.*

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

- **./assess 2560000**
- User Prescribed Input File: **dosRenyisIntercambio2.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**

Los resultados obtenidos son los siguientes:

Cuadro III. PRIMEROS 3 DATOS OBTENIDOS EN EL EJEMPLO 2.

| Primeros tres valores generados con el esquema propuesto. | | |
|---|---|---|
| Primer valor generado | | |
| Especificación | Base 10 | Representación binaria |
| Valor mapa 1 | 3355443200 | 1100 1000 0000 0000 0000 0000 0000 0000 |
| Valor mapa 2 | 154402816 | 0000 1001 0011 0100 0000 0000 0000 0000 |
| Resultado | 154454016 | 0000 1001 0011 0100 1100 1000 0000 0000 |
| Segundo valor generado | | |
| Especificación | Base 10 | Representación binaria |
| Valor mapa 1 | 104857600 | 0000 0110 0100 0000 0000 0000 0000 0000 |
| Valor mapa 2 | 9424 | 0000 0000 0000 0000 0010 0100 1101 0000 |
| Resultado | 8848 | 0000 0000 0000 0000 0010 0010 1001 0000 |
| Tercer valor generado | | |
| Especificación | Base 10 | Representación binaria |
| Valor mapa 1 | 3276800 | 0000 0000 0011 0010 0000 0000 0000 0000 |
| Valor mapa 2 | 3569352704 | 1101 0100 1100 0000 0000 0000 0000 0000 |
| Resultado | 3569352754 | 1101 0100 1100 0000 0000 0000 0011 0010 |

Cuadro IV. RESULTADOS DE LAS PRUEBAS DE ALEATORIEDAD NIST A LOS DATOS DOSRENYISINTERCAMBIO2.DAT .

| PRUEBA APLICADA | P-VALOR | EXITO? |
|---|---|---|
| APROXIMATE ENTROPY | 0.847752 | $\surd$ |
| BLOCK FRECUENCY | 0.330180 | $\surd$ |
| CUMULATIVE SUMS | F:0.170864, R: 0.109874 | $\surd$ |
| FFT | 0.774289 | $\surd$ |
| FRECUENCY | 0.167972 | $\surd$ |
| LINEAR COMPLEXITY | 0.712980 | $\surd$ |
| LONGEST RUN | 0.516759 | $\surd$ |
| NON OVERLAPPING TEMPLATE | 146 DE 148 | $\surd$ |
| OVERLAPPING TEMPLATE | 0.563560 | $\surd$ |
| RANDOM EXCURSIONS | 8 DE 8 | $\surd$ |
| RANDOM EXCURSIONS VARIANT | 18 DE 18 | $\surd$ |
| RANK | 0.648994 | $\surd$ |
| RUNS | 0.618793 | $\surd$ |
| SERIAL | 2 DE 2 | $\surd$ |
| UNIVERSAL | 0.227311 | $\surd$ |

## IV. CONCLUSIONES.

Se obtuvieron resultados satisfactorios a la hora de aplicar las pruebas NIST a los datos generados por el esquema ropuesto. En este caso, se supuso que $i = j$.

Normalmente ocurre que si se obtienen trayectorias aceptables de los mapas utilizados al aplicar el esquema que se menciona en este documento, se obtendrán valores que aprobarán las pruebas NIST.

# V. Anexos.

## *V-A. Codigo 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, hacemoxs un esquema cruzado para la generacion de numeros
aleatorios.
*/

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(){


  /*En este primer ejemplo, se hace la prueba para i = j en cada uno de
   los dos mapas.*/
   unsigned long Xtotal[ITtotales];
   FILE *  archivobin;
   unsigned long Xn1 = 13;
   unsigned long Xn2 = 19;
   unsigned int n = 32;

   /*Valores de i, j y q para el mapa 1.*/
   unsigned int i1=9;
   unsigned int j1=9;
   unsigned long q1=13;
   /*Valores de i, j y q para el mapa 2.*/
   unsigned int i2=13;
   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 ("dosRenyisIntercambio1.dat", "wb");
 if (archivobin==NULL)
 {
 perror("No se puede abrir dosRenyisIntercambio1.dat");
 return -1;
 }

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


/*Sea a el valor de 2^16 -1 (el maximo valor en 16 bits de 32.)*/
 unsigned long a = 65535;
 /*Declaramos valores temporales.*/
 unsigned long temp;
 unsigned long temp1;
 unsigned long temp2;
 unsigned long Xn3;

 while (iteraciones < IT) {

         /*Primero, creamos los valores por medio del mapa renyi.*/
          Xn1= RENYI_MAP(Xn1,param1,j1);
          Xn2= RENYI_MAP(Xn2,param2,j2);

          /*obtenemos la parte derecha del valor de Xn2.*/
          temp1 = Xn2 & a;
          /*Obtenemos la parte izquierda del valor de Xn1.*/
          temp2 = Xn1 >> 16;
          /*Hacemos la combinacion.*/
          temp = temp1^temp2;

          /*Obtenemos la parte izquierda del valor de x2.*/
          temp1 = Xn2 >> 16;
          /*Obtenemos la parte derecha del valor de x1.*/
          temp2 = Xn1 & a;

          /*Ahora, formamos el nuevo valor para x3.*/
          Xn3=(temp1^temp2)<<16;
          Xn3|=temp;


          Xtotal[iteraciones++] = Xn3;



 }


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

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

```
      return 1;
   }



return 0;
}
```

*V-B. Codigo 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, hacemoxs un esquema cruzado para la generacion de numeros
aleatorios.
*/

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 = 13;
   unsigned long Xn2 = 19;
   unsigned int n = 32;

   /*Valores de i, j y q para el mapa 1.*/
   unsigned int i1=5;
   unsigned int j1=5;
   unsigned long q1=29;
   /*Valores de i, j y q para el mapa 2.*/
   unsigned int i2=14;
   unsigned int j2=14;
   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 ("dosRenyisIntercambio2.dat", "wb");
 if (archivobin==NULL)
 {
 perror("No se puede abrir dosRenyisIntercambio2.dat");
 return -1;
 }

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

/*Sea a el valor de 2^16 -1 (el maximo valor en 16 bits de 32.)*/
 unsigned long a = 65535;
 /*Declaramos valores temporales.*/
 unsigned long temp;
 unsigned long temp1;
 unsigned long temp2;
 unsigned long Xn3;


 while (iteraciones < IT) {

          /*Primero, creamos los valores por medio del mapa renyi.*/
          Xn1= RENYI_MAP(Xn1,param1,j1);
          Xn2= RENYI_MAP(Xn2,param2,j2);

          /*obtenemos la parte derecha del valor de Xn2.*/
          temp1 = Xn2 & a;
          /*Obtenemos la parte izquierda del valor de Xn1.*/
          temp2 = Xn1 >> 16;
          /*Hacemos la combinacion.*/
          temp = temp1^temp2;

          /*Obtenemos la parte izquierda del valor de x2.*/
          temp1 = Xn2 >> 16;
          /*Obtenemos la parte derecha del valor de x1.*/
          temp2 = Xn1 & a;

          /*Ahora, formamos el nuevo valor para x3.*/
          Xn3=(temp1^temp2)<<16;
          Xn3|=temp;


          Xtotal[iteraciones++] = Xn3;


 }


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

```
   if(!fclose(archivobin)){
      printf( "\nArchivo binario intercambio 2 cerrado\n" );
   }
   else{
      printf( "\nError: Archivo binario  intercambio 2 no cerrado \n" );
      return 1;
   }



return 0;
}
```