Generador Binario Pseudoaleatorio Basado en la Combinación de Registros de Desplazamiento con Retroalimentación Lineal, mediante Suma Real con Acarreo

Andrés Francisco Farías – Andrés Alejandro Farías

Departamento Académico de Ciencias Físicas, Matemáticas y Naturales
Universidad Nacional de La Rioja, La Rioja. Argentina
(afarias665@yahoo.com.ar, andres_af86@hotmail.com)

Abstract
El trabajo consiste en el desarrollo de un generador binario pseudoaleatorio basado en la combinación de registros de desplazamiento con retroalimentación lineal (Linear Feedback Shift Register, LFSR), mediante la suma real con acarreo de las secuencias de salida de las funciones de filtrado no lineal, que se alimentan de los registros de los LFSR. Se incluye la descripción del modelo, la estructura de cada generador, selección de las funciones booleanas que cuenten con las mejores propiedades criptográficas, la definición de la combinación final. Por último, para verificar la aleatoriedad de las secuencias obtenidas, se aplican a las mismas un conjunto de pruebas estadísticas de aleatoriedad.

Keywords: LFSR, cipher, key, Boolean function, non-linearity

1 Introducción

El generador se basa en LFSR [1], [2],de distintas longitudes, que tienen, cada uno, dos funciones de filtrado no lineal que se abastecen de las secuencias producidas por los mismos LFSR, después en grupo de a cuatro, esos resultados se combinan mediante un proceso de suma real con acarreo. De esto se obtienen dos secuencias que se someten a una operación de XOR, para obtener un resultado final, que es sometido luego a pruebas de aleatoriedad.

El desarrollo de un generador pseudoaleatorio de estas características requiere de varias etapas:

- Presentación del modelo.
- Selección de los distintos LFSR.
- Búsqueda de funciones booleanas de cuatro variables en base a sus propiedades criptográficas.
- Composición del generador con los componentes ya seleccionados.
- Clave y el procedimiento para generar los estados iniciales de los LFSR.

- Elección de las pruebas estadísticas a utilizar y los criterios de análisis de los resultados.
- Puesta en funcionamiento y realización de las pruebas de aleatoriedad necesarias sobre las secuencias obtenidas.

2 Definición esquemática del modelo

El generador propuesto en este trabajo, está conformado por cuatro LFSR, que tienen cada uno, dos funciones booleanas de filtrado no lineal, que producen secuencias binarias, las que luego son combinadas mediante un procedimiento de suma real con acarreo.. Las secuencias obtenidas son sometidas a una operación XOR, según la figura:

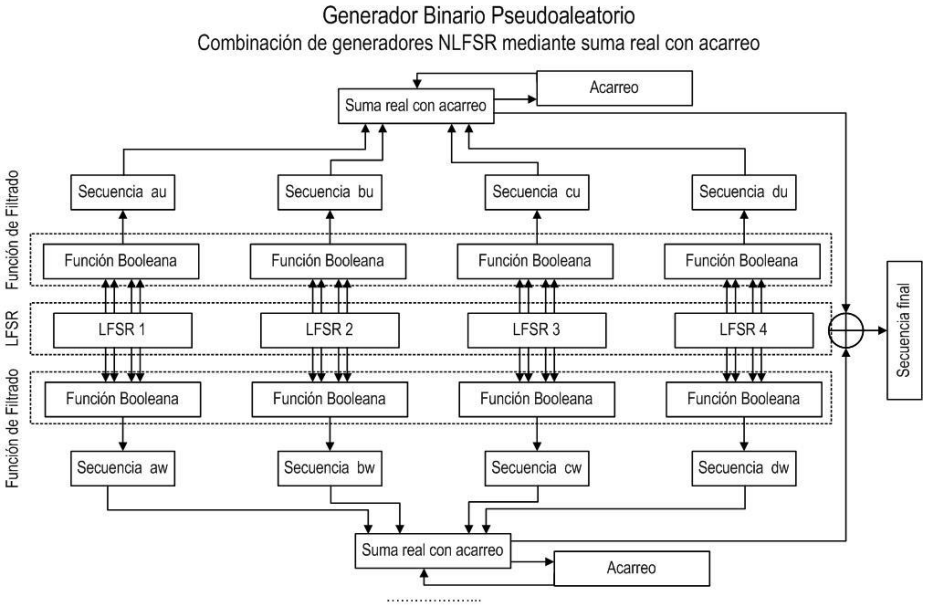

Fig. 1. Esquema generador binario pseudoaleatorio

3 Elección de los distintos LFSR

Las longitudes y polinomios primitivos de cada LFSR, que componen el generador, son las siguientes [3], [4], [5].

Tabla 1. LFSR, longitudes y polinomios primitivos del Generador

LFSR	Longitud	Polinomios primitivos
1	31	$P(x) = x^{31} + x^{25} + x^{23} + x^{8} + 1$
2	37	$P(x) = x^{37} + x^{22} + x^{14} + x^{2} + 1$

3	41	$P(x) = x^{41} + x^{32} + x^{31} + x^{27} + 1$
4	43	$P(x) = x^{43} + x^{27} + x^{22} + x^{5} + 1$

El generador propuesto en este trabajo, está conformado según la figura:

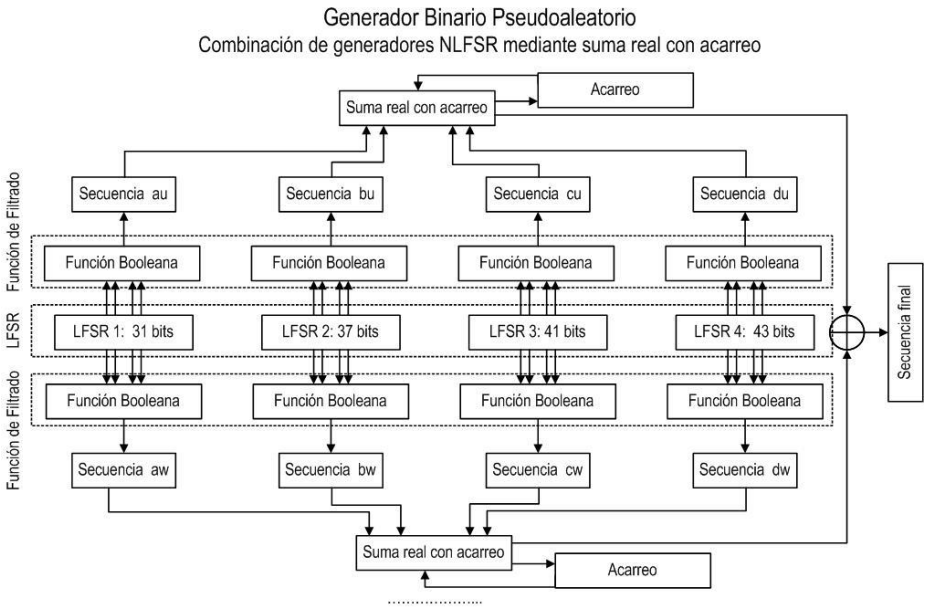

Fig. 2. Generador binario pseudoaleatorio

4 Selección de las funciones booleanas

4.1 Propiedades criptográficas deseables .

A continuación se indican algunas de las propiedades criptográficamente más significativas, adoptadas para este trabajo [6], [7], [8].

- **Función Balanceada**: Una función booleana de n-variables $\boldsymbol{f}$ es balanceada si $\boldsymbol{w(f) = 2n - 1}$. Esta propiedad es deseable para evitar ataques criptodiferenciales. La función es balanceada cuando el primer coeficiente del espectro de Walsh-Hadamard, es igual a cero: $\boldsymbol{F(0) = 0}$.
- **No Linealidad**: Valores altos de esta propiedad reducen el efecto de los ataques por criptoanálisis lineal. La No Linealidad de una función booleana puede ser calculada directamente de la transformada de Walsh-Hadamard, (Ecuación 2):

$$NL_f = \frac{1}{2} \cdot (2^n - |WH_{max}(f)|) \qquad (1)$$

- **Grado Algebraico**: El grado algebraico de una función, es el número de entradas más grande que aparece en cualquier producto de la Forma Normal Algebraica. Es deseable que sean valores altos.

- **SAC**: El Criterio de Avalancha Estricto requiere los efectos avalancha de todos los bits de entrada. Una función booleana se dice que satisface SAC sí y solo sí, la Ecuación 3, es balanceada para toda u con w(u)=1. $f(x) \oplus f(x \oplus u)$

4.2 Tabla de resultados.

Siguiendo los criterios arriba indicados las funciones booleanas aceptadas, son:

Tabla 2. Funciones de cuatro variables adoptadas

f_{NAF}
$f_{84} = a \cdot c \oplus b \cdot c \oplus a \cdot d \oplus b \cdot d \oplus c \cdot d$
$f_{89} = a \cdot c \oplus b \cdot c \oplus d \oplus a \cdot d \oplus b \cdot d$
$f_{100} = a \cdot c \oplus b \cdot c \oplus d \oplus a \cdot b \cdot d \oplus c \cdot d$
$f_{176} = c \oplus a \cdot c \oplus b \cdot c \oplus a \cdot d \oplus b \cdot d$
$f_{199} = c \oplus a \cdot c \oplus b \cdot c \oplus d \oplus c \cdot d$
$f_{381} = c \oplus a \cdot b \cdot c \oplus a \cdot d \oplus b \cdot d \oplus c \cdot d$
$f_{468} = c \oplus d \oplus a \cdot d \oplus b \cdot d \oplus c \cdot d$
$f_{536} = a \cdot b \oplus b \cdot c \oplus a \cdot d \oplus b \cdot d \oplus c \cdot d$

5 Conformación del generador combinacional

El generador combinacional queda de la siguiente manera:.

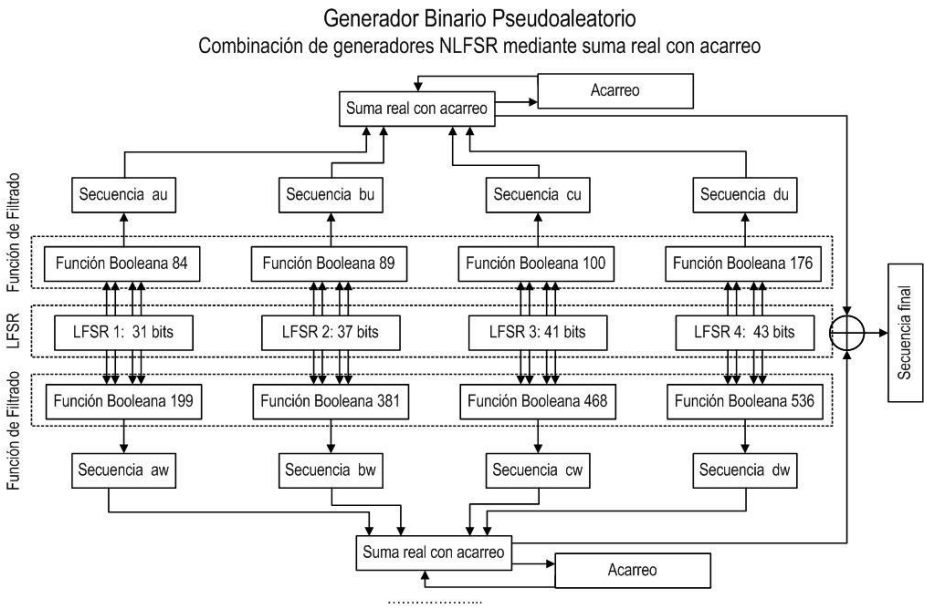

Fig. 3. Generador Combinacional

6 Clave

Para obetener los estados iniciales de los distintos LFSR se realiza un proceso que utiliza una clave de una longitud de 32 caracteres, que expresada en código ASCII (American Standard Code for Information Interchange), tiene longitud de 256 bits.

Para simplificar la introducción de la clave, se aceptan solamente las letras del alfabeto inglés (minúsculas y mayúsculas) y los números del sistema de numeración decimal, es decir un total de 62 caracteres.

La clave es sometida a un proceso criptográfico, que se indica en la Figura 4.

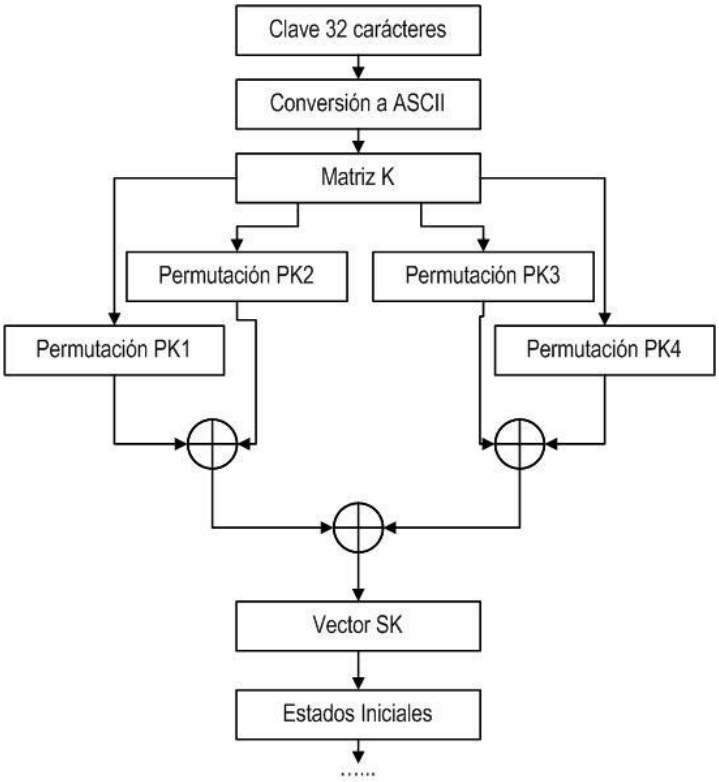

Fig. 4.. Generador para estados iniciales

7 Permutaciones

7.1 Generador congruencial multiplicativo

El generador tiene la siguiente expresión: [9]

$$x_{i+1} = (a_x \cdot x_i)mod\ m_x \qquad (2)$$

Donde: $a_x = multiplicador, m_x = módulo, x_0 = semilla$

Tabla 3. Vectores, módulos, multiplicadores y semillas

Vector	módulo	multiplicador	semilla
PK1	1048576	2741	3249
PK2	1048576	2749	3271
PK3	1048576	2753	3301
PK4	1048576	2767	3347

7.2 Generación de los estados iniciales

De la operación resulta un vector SK[j] de 256 bits, que es el que proveerá los estados iniciales de los LFSR, en forma secuencial.

8 Elección de las pruebas estadísticas

Fueron seleccionadas algunas pruebas de la Norma NIST Special Publication 800-22, del trabajo de Rukhin (et al.) [9].

8.1 Prueba de frecuencia

El propósito de esta prueba es determinar si el número de unos y ceros en una secuencia es aproximadamente el mismo que se espera de una secuencia verdaderamente aleatoria. La prueba evalúa la cercanía de la fracción de unos a ½, que es decir, el número de unos y ceros en una secuencia debe ser aproximadamente el mismo. Todas las pruebas posteriores dependen de la aprobación de esta prueba.

8.2 Prueba de frecuencia en un bloque

La meta de esta prueba es determinar si la frecuencia de unos en un bloque de M bits es aproximadamente $M / 2$, como se esperaría bajo un supuesto de aleatoriedad.

8.3 Prueba de rachas

Una racha de longitud k consta de exactamente k bits idénticos y está acotada antes y después con un poco del valor opuesto. El propósito de la prueba de rachas es determinar si el número de rachas unos y ceros de varias longitudes es lo esperado para una secuencia aleatoria.

8.4 Prueba de rachas de unos en un bloque

El fin de esta prueba es determinar si la longitud de la ejecución más larga de las dentro de la secuencia probada es consistente con la longitud de la serie más larga de las que cabría esperar en una secuencia aleatoria. Tenga en cuenta que una irregularidad en la longitud esperada de la serie más larga implica que también hay una irregularidad en la longitud de la serie más larga de ceros.

8.5 Prueba de sumas acumuladas

Determina si la suma acumulativa de las secuencias parciales que ocurren en la secuencia probada es demasiado grande o demasiado pequeña en relación con el comportamiento esperado de esa suma acumulada para secuencias aleatorias.

8.6 Prueba de entropía aproximada

El enfoque de esta prueba es la frecuencia de todas las posibles superposiciones patrones de m bits en toda la secuencia. El propósito de la prueba es comparar la frecuencia de bloques superpuestos de dos longitudes consecutivas / adyacentes (m, m + 1) contra el resultado esperado para un secuencia aleatoria.

9 Pruebas sobre el generador

Se analizaron cien secuencias binarias, obtenidas del generador a partir de cien claves distintas.

El nivel de significancia adoptado para las pruebas estadísticas es de $\alpha = 0{,}01$. La hipótesis nula es:

$$H_0 \rightarrow \text{p\_valor} > 0{,}01$$

Debido al gran volumen de procesamiento requerido, se desarrolló un programa escrito en lenguaje C++, con los algoritmos correspondientes al generador y a las pruebas estadísticas. Es decir que el software calculó las secuencias binarias y simultáneamente realizó las pruebas sobre las mismas.

10 Interpretación de los resultados

Teniendo los resultados se realizan dos procesos para la interpretación de los mismos:

- Proporción de muestras que pasan las pruebas.
- Prueba de Uniformidad de los p-valor
    - Tabla de frecuencia e histograma
    - Prueba de Bondad de Ajuste

Se aplica la prueba de Bondad de Ajuste χ^2aplicando la siguiente expresión:

$$\chi^2 = \sum_{i=1}^{10} \frac{\left(F_i - \frac{s}{10}\right)^2}{\frac{s}{10}} \qquad (3)$$

Donde: $F_i = \text{Frecuencia de la clase i} \quad s = \text{Cantidad de muestras}$

El primer procedimiento se realiza considerando los resultados de todas las pruebas y el segundo se realiza en forma individual. En todos los casos se deben superar todas las pruebas para aceptar los resultados.

10.1 Proporción de muestras que pasan las pruebas

Para el análisis de los resultados, se determina la proporción de muestras que superan las pruebas, y con esos datos se construye un gráfico de puntos, luego se verifica si los mismos caen dentro de los límites superior e inferior, donde k es el número de muestras.

$$LS, LI = (1 - \alpha) \pm 3 \cdot \sqrt{\alpha \cdot (1 - \alpha)/k} \qquad (4)$$

En nuestro caso $k = 100$ y el nivel de significancia elegido es: $\alpha = 0.01$, los límites quedan:. $LS = 1{,}02$ y $LI = 0{,}96$

Se consideran todas pruebas, los resultados se indican en la tabla

Tabla 4. Pruebas

Pruebas	Proporción	Superior	Inferior
Frecuencias	1,00	1,02	0,96
Frecuencias en un Bloque	0,97	1,02	0,96
Rachas	1,00	1,02	0,96
Rachas de Unos en un Bloque	0,97	1,02	0,96
Sumas Acumuladas Adelante	0,99	1,02	0,96
Sumas Acumuladas Atrás	1,00	1,02	0,96
Entropía Aproximada	1,00	1,02	0,96

En el gráfico se aprecia el resultado, en definitiva la secuencia que entrega el generador supera las pruebas de aleatoriedad.

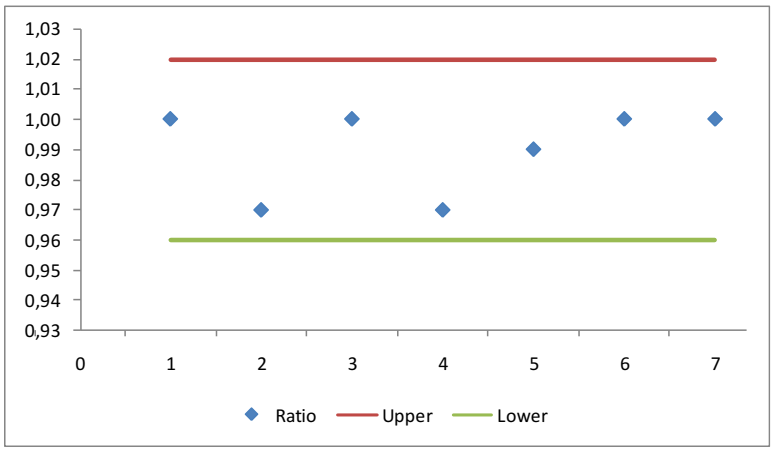

Fig. 5. Gráfico de puntos

10.2 Prueba de bondad de ajuste

Este control se ejecuta para cada prueba sobre las cien muestras, con los resultados de las frecuencias de p-valor obtenidos.

Tabla 5. Pruebas χ^2

Pruebas	χ^2	$\chi^2 ref$	Pasa
Frecuencias	0,936	0,0001	Sí
Frecuencias en un Bloque	0,319	0,0001	Sí
Rachas	0,924	0,0001	Sí
Rachas de Unos en un Bloque	0,384	0,0001	Sí
Sumas Acumuladas Adelante	0,401	0,0001	Sí
Sumas Acumuladas Atrás	0,456	0,0001	Sí
Entropía Aproximada	0,163	0,0001	Sí

10.3 Análisis final

En base a los resultados de la pruebas se realiza una tabla resumen.

Table 6. Análisis final

Análisis	Pruebas	Resultados
Proporción de secuencias que pasan las pruebas	Todas	Supera
Distribución uniforme de p-valor	Frecuencias	Supera
	Frecuencias dentro de un bloque	Supera
	Rachas	Supera
	La más larga racha de unos en un bloque	Supera
	Sumas acumuladas adelante	Supera
	Sumas acumuladas atrás	Supera
	Entropía estimada	Supera

En definitiva las secuencias que entrega el generador son pseudoaleatorias.

11 Conclusiones

El generador obtenido entrega secuencias binarias pseudoaleatorias de elevado período y complejidad lineal. Para ello se diseñó un dispositivo que combina en forma no lineal las secuencias producidas por cuatro LFSR, que cuentan con ocho funciones de filtrado no lineal, que luego se combinan mediante dos sumas reaesl con acarreo.

Los LFSR tienen polinomios de conexión primitivos, lo que asegura un elevado período en la secuencia resultante.

Las funciones booleanas de filtrado no lineal, son las responsables del proceso de entregar secuencias no lineales y aseguran las mejores prestaciones criptográficas, si cumplen determinados criterios. Realizado el proceso de selección, las funciones fueron incorporadas al generador y luego puestas a funcionar para generar las secuencias respectivas con distintos valores de claves y ser sometidas a las pruebas de aleatoriedad respectivas.

Las secuencias obtenidas, superaron todas las pruebas lo que demuestra que el generador funciona de acuerdo a lo previsto.

12 Referencias

[1] Massodi, F., Alam, S. and Bokhari, M., "A Analysis of Linear Feedback Shift Registers in Stream Ciphers", International Journal of Computer Application, 16 (17), pp. 0975 – 887, 2012.

[2] Menezes, A.,Van Oorschot, P. and Vanstone, S., "Handbook of Applied Cryptography", Massachusetts Institute of Technology, 1996.

[3] Parr, C. and Pelzl, L., Understanding Cryptography, Springer, 2010.

[4] Stahnke, W., "Primitive Binary Polynomials", Mathematics of Computation, 27. 124, pp. 977-980, 1973.

[5] Seroussi, G., "Table of Low-Weight Binary Irreducible Polynomials", Computer Systems Laboratory, 1998.

[6] Clark, J., Jacob, J., Maitra, S., Stanica, P.: Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion. Computational intelligence. 20. (3), 450—462 (2004)

[7] Braeken, A.: Cryptographic Properties of Boolean Functions and S-Boxes. Faculteit Ingenieurswetenschappen. Katholieke Universiteit Leuven (2003)

[8] Elhosary, A., Hamdy, N., Farag, I., Rohiem, I.: State of the Art in Boolean Functions Cryptographic Assessment. International Journal of Computer Networks and Communications Security.1. (3), 88--94 (2013)

[9] Fishman, G.: Multiplicative Congruential Random Number Generators with Modulus 2ß : An Exhaustive Analysis for ß = 32 and a Partial Analysis for ß = 48. Mathematics of Computation. 54. (189), 33--344 (1990)

[10] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., and Vo, S., "A Statistical Prueba Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", National Institute of Standards and Technology, (2000).