Cifrador de Flujo Basado en un Generador Binario Pseudoaleatorio, con Clave de 256 Bits

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 presente documento expone el procedimiento de construcción de un cifrador de flujo basado en un generador binario pseudoaleatorio conformado por la combinación no lineal de registros de desplazamiento con retroalimentación lineal con funciones no lineales de filtrado. Para verificar la aleatoriedad de las secuencias obtenidas, se aplican a las mismas un conjunto de pruebas estadísticas de aleatoriedad.

Keywords: NLFSR, LSFR, Clave, Período, Polinomios Primitivos, Pruebas de Aleatoriedad, XOR, Bits aleatorios

1 Introducción

Se trata de un dispositivo conformado por un generador de números binarios pseudoaleatorios, con una clave de 256 bits, basado en el uso de distintos registros de desplazamiento de reatroalimentación lineal (LFSR, sigla en inglés), combinados mediante funciones booleanas balanceadas y de alta no linealidad.

La secuencia cifrante binaria pseudoaleatoria entregada por este generador es sometida a una operación XOR, con la secuencia binaria de los carácteres del texto plano a cifrar en código ASCII binario, de esto se obtiene una nueva secuencia binaria, que es el texto cifrado en código ASCII binario [1], [2].

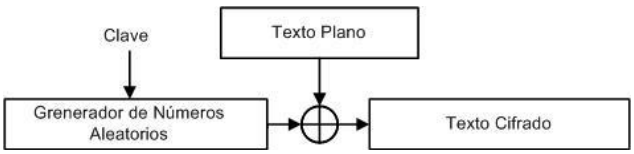

Fig. 1. Cifrador de Flujo

Para desencriptar se realizar una operación XOR entre el texto cifrado en código ASCII binario y la misma secuencia pseudoaleatoria binaria producida por el generador de números binarios pseudoaleatorios, con la que se realizó el cifrado. Que

da como resultado el texto plano en código ASCII binario.

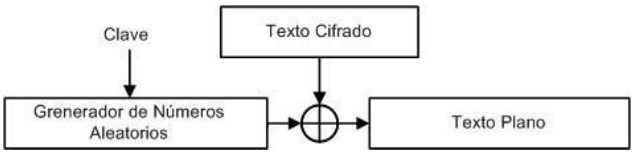

Fig. 2. Descifrador de Flujo

2 Definición del modelo para el generador pseudoaleatorio

Los componentes principales, son LFSR y las funciones de filtrado no lineal que son funciones booleanas de cuatro variables. Para el generador en estudio se dispone de tres de estos LFSR con dos funciones de filtrado no lineal, que entrega dos secuencias, se tiene que el conjunto produce seis secuencias aleatorias. La combinación se producen según el criterio parada y arranque; en la figura 3.

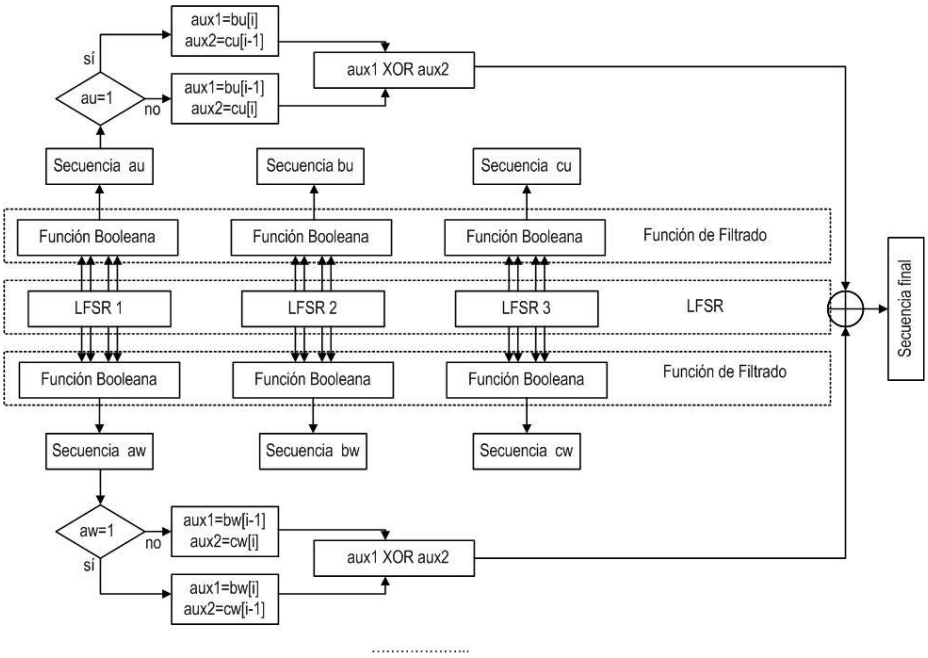

Fig. 3. Esquema generador aleatorio

3 Elección de los 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	47	$P(x) = x^{47} + x^{32} + x^{24} + x^{11} + 1$
2	61	$P(x) = x^{61} + x^{57} + x^{26} + x^{3} + 1$
3	59	$P(x) = x^{59} + x^{54} + x^{46} + x^{26} + 1$

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

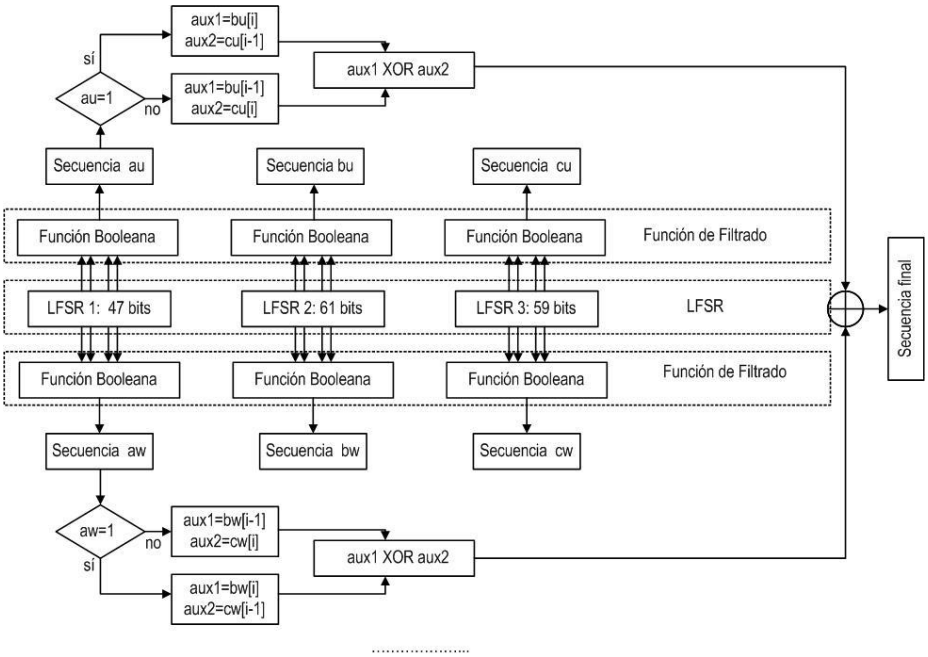

Fig. 4. Esquema generador aleatorio

4 Selección de las funciones booleanas

4.2 Propiedades criptográficas deseables

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

- **Función balanceada:** 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: $F(0) = 0$
- **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:

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

- **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, es balanceada para toda u con w(u)=1, para: $f(x \oplus u)$

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

Tabla 2**.** Funciones de cuatro variables adoptadas

f_{NAF}
$f_{5775} = a \oplus b \oplus a \cdot b \oplus a \cdot c \oplus a \cdot d$
$f_{4722} = a \oplus b \oplus a \cdot c \oplus b \cdot c \oplus c \cdot d$
$f_{4529} = a \oplus c \oplus a \cdot c \oplus b \cdot c \oplus c \cdot d$
$f_{4393} = a \oplus c \oplus a \cdot d \oplus b \cdot d \oplus c \cdot d$
$f_{2402} = b \oplus a \cdot c \oplus b \cdot c \oplus d \oplus c \cdot d$
$f_{3981} = a \oplus a \cdot c \oplus b \cdot c \oplus d \oplus c \cdot d$

5 Conformación del generador combinacional

El generador combinacional queda de la siguiente manera:.

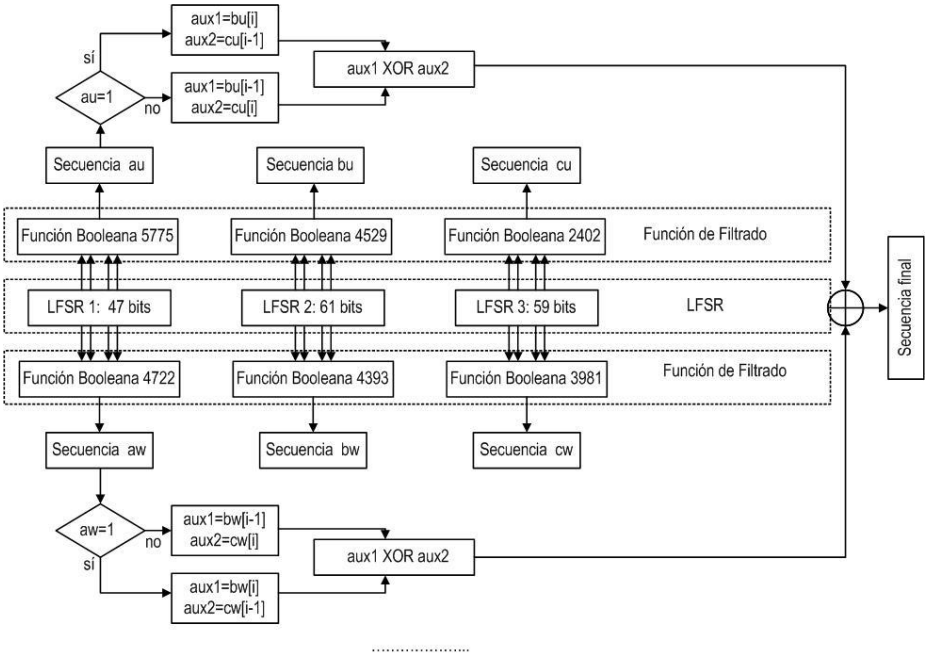

Fig. 5. Generador Combinacional

6 Clave

Para originar 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 el procedimiento de 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 6.

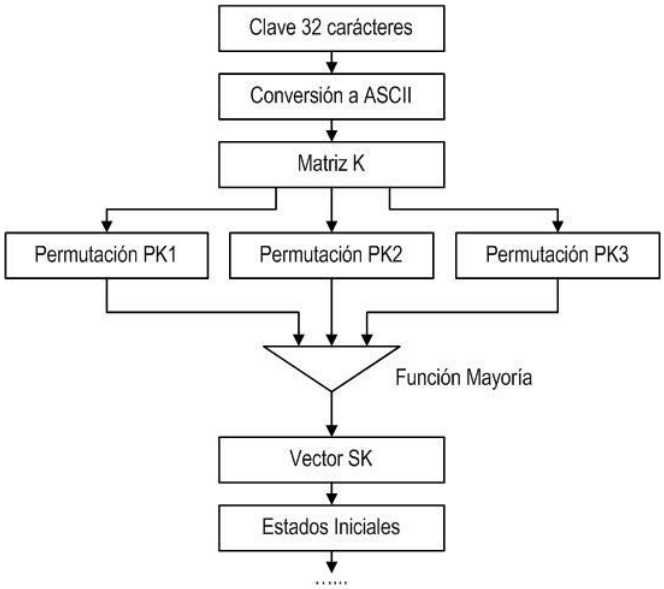

Fig. 6.. Clave del generador

7 Elección de las pruebas estadísticas

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

7.1 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.

7.2 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.

7.3 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.

7.4 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.

7.5 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.

7.6 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.

8 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.

9 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:

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

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

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.

9.1 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.

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

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 3. Pruebas

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

Las secuencias que produce el cifrador superan las pruebas de aleatoriedad.

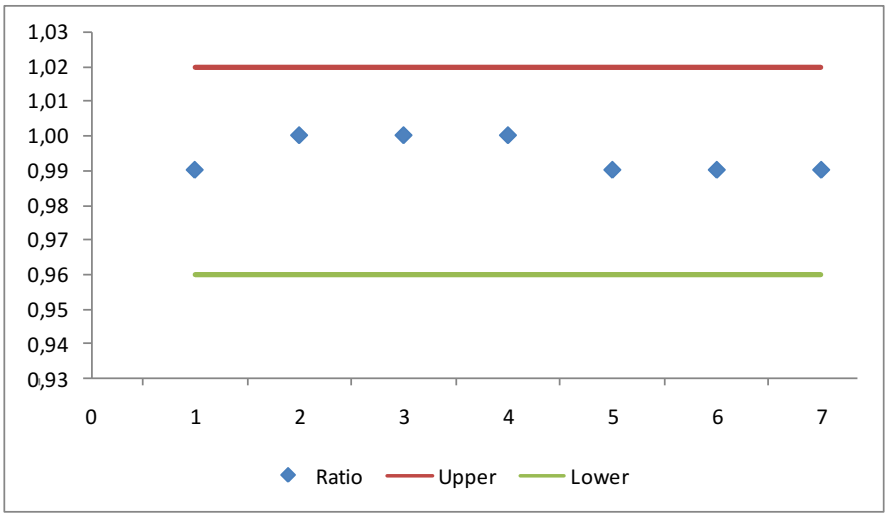

Fig. 7. Gráfico de puntos

9.2 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 4. Pruebas χ^2

Pruebas	χ^2	$\chi^2 ref$	Pasa
Frecuencias	0,225	0,0001	Sí
Frecuencias en un Bloque	0,936	0,0001	Sí
Rachas	0,720	0,0001	Sí
Rachas de Unos en un Bloque	0,335	0,0001	Sí
Sumas Acumuladas Adelante	0,262	0,0001	Sí
Sumas Acumuladas Atrás	0,798	0,0001	Sí
Entropía Aproximada	0,658	0,0001	Sí

9.3 Análisis final

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

Tabla 5. 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.

10 Comparación

10.1 Comparación de frecuencias

Diferencias entre texto plano y texto cifrado.

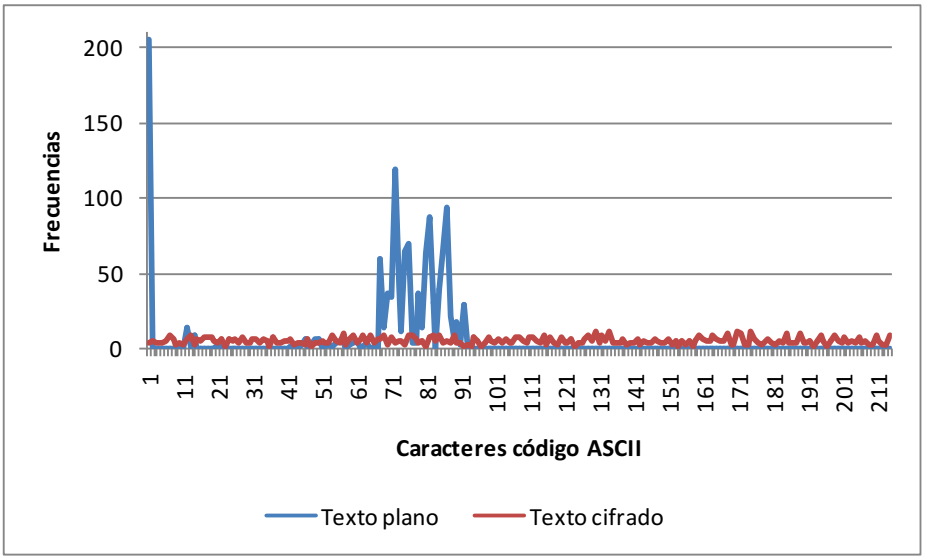

Fig. 8. Frecuencias de caracteres del texto plano y cifrado

11 Conclusiones

Se ha presentado un cifrador de flujo con algunas características interesantes tales como clave de mayor longitud y la incorporación de un generador basado en la combinación de LFSR.

Sobre este diseño se pueden implementar otras variantes para lograr futuras versiones que contemplen entre otras cosas: claves más largas y nuevos generadores binaries pseudoaleatorios.

La respuesta de esta versión fue buena y entregó un texto cifrado con una frecuencia de carácteres con cierta uniformidad, lo que hace difícil un criptoanálisis basado en la estadística de aparición de carácteres.

Finalmente se realizaron pruebas estadísticas de aleatoriedad sobre cien secuencias obtenidas del mismo texto plano con cien claves distintas, las que dieron resultados positivos.

12 Referencias

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

[2] Canteaut, A. and Filio, E., "Ciphertext only reconstruction of stream ciphers based on combination generators. Fast Software Encryption 2000", Lecture Notes in Computer Science, 1978, pp. 165–180, 2001.

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

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

[5] Constantinescu, N., "Combining Linear Feedback Shift Registers", in Annals of University of Craiova, Math. Comp. Sci. Ser, 2009, 36 (2), pp. 42–46.

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

[7] 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)

[8] 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)

[9] 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).