Cifrador de flujo basado en un generador de secuencias pseudoaleatorias

Andrés Francisco Farías[1] – Andrés Alejandro Farías[2]

[1] DACEFyN – UNLaR (afarias665@yahoo.com.ar)
[2] DACEFyN – UNLaR (andres_af86@hotmail.com)

Resumen. Un cifrador de flujo, donde el generador de números binarios pseudoaleatorios está conformado por la combinación de funciones booleanas de tres generadores que a su vez están compuestos por tres registros de desplazamiento con retroalimentación lineal (LFSR, sigla en inglés) con polinomios primitivos de conexión. Los estados iniciales son aportados por una clave de 256 bits, que sufre un proceso de transformación mediante permutaciones que finalmente procesa una función booleana de cuatro variables. Las secuencias binarias obtenidas del cifrador son sometidas posteriormente a pruebas estadísticas de aleatoriedad

Palabras Clave: Cifrador, clave, LSFR, pruebas de aleatoriedad, bits aleatorios.

Key Words: Cipher, key, LSFR, randomness tests, random bits.

1 Introducción

Actualmente resulta de mucha importancia la criptografía como un elemento que hace a la seguridad informática, dado que además de proporcionar protección, nos permite custodiar la confidencialidad e integridad de la información que es un aspecto muy necesario para todas las organizaciones públicas y privadas.

La criptografía es una herramienta fundamental en las instituciones debido a que en el flujo de información o en su almacenamiento el riesgo está presente, y pueden suceder robos, adulteración o borrado de información, violación de contraseñas y accesos no autorizados entre otros. Hoy se cuenta con sofisticados algoritmos de cifrado cada uno con características particulares, pero todos con un fin en común, el de proteger la información y evitar que algún desconocido use en forma maliciosa los datos de la empresa.

El motivo de este trabajo,es presentar un cifrador de flujo con clave de 256 bits, que en principio, por su longitud de clave, desalienta un posible ataque por fuerza bruta, y está basado en un generador de secuencias binarias pseudoaleatorias, que superó holgadamente las pruebas estadísticas de aleatoriedad, y es el encargado de proveer la secuencia cifrante en estos sistemas.

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].

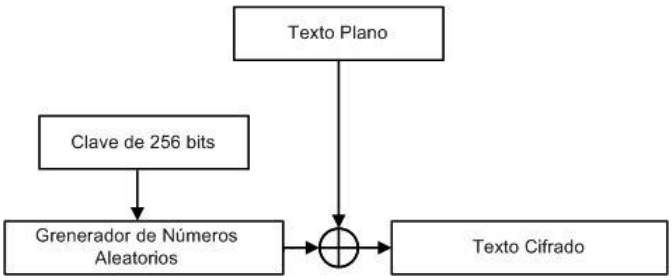

Fig. 1. Cifrador de Flujo

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.

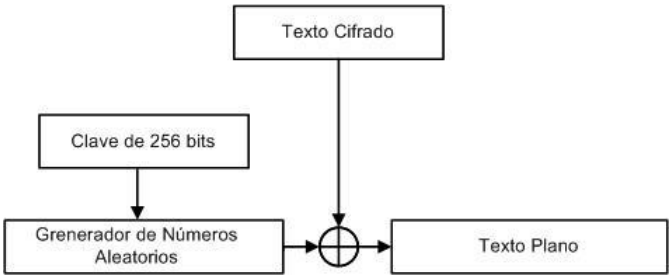

Fig. 2. Descifrador de Flujo

2 Definición del modelo para el generador pseudoaleatorio

Se propone un modelo conformado por tres generadores constituídos, cada uno, por tres Linear Feedback Shift Register (LFSR), cuyas secuencias se combinan mediante una función booleana. Finalmente la salidas de estos generadores se combinan mediante funciones booleanas de tres y cuatro variables.

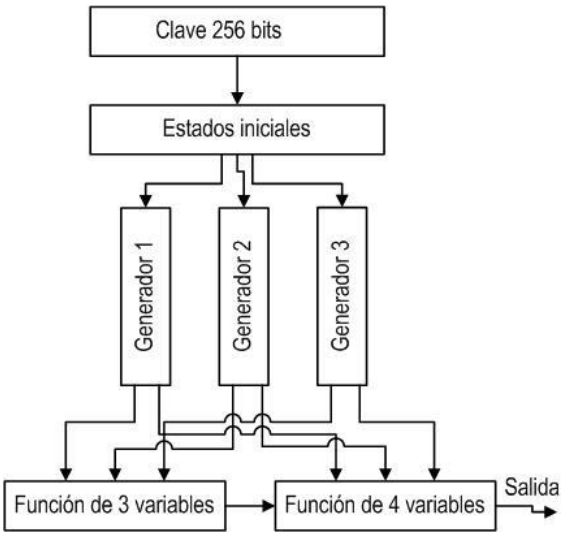

Fig. 3. Esquema del generador de secuencias binarias pseudoaleatorias

Los LFSR, que componen cada generador de secuencia combinada, debe cumplir con un requisito indispensable: los polinomios de conexión son de tipo primitivo [3], [4] para obtener períodos de máxima longitud.

3 Propiedades criptográficas de las funciones booleanas

A continuación se indican algunas de las propiedades criptográficamente más significativas, adoptadas para este trabajo [5] y [6].

- **Función Balanceada:** Una función booleana de n-variables f es balanceada si $\mathrm{w(f)} = \mathrm{2n} - 1$. 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: $\mathrm{F}(0) = 0$.
- **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: $NL_f = \frac{1}{2} \cdot (2^n - |WH_{max}(f)|)$
- **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, $f(x \oplus u)$, es balanceada para toda u con w(u)=1:.

3.1 Tabla de resultados

Siguiendo los criterios establecidos en los párrafos anteriores, adoptamos las funciones, indicadas en la Tabla 1.

Tabla 1. Funciones de tres y cuatro variables aceptadas para el generador

f_{NAF}
$f_{34} = x \oplus y \oplus x \cdot y \oplus y \cdot z$
$f_{84} = x \cdot z \oplus y \cdot z \oplus x \cdot w \oplus y \cdot w \oplus z \cdot w$
$f_{19} = a \oplus (a \cdot b) \oplus c \oplus (a \cdot c)$
$f_{34} = d \oplus e \oplus (d \cdot e) \oplus (e \cdot f)$
$f_9 = h \oplus (g \cdot h) \oplus i \oplus (h \cdot i)$

4 Generador 1

Conformado por tres LFSR, con sus respectivos polinomios primitivos, cuyas secuencias de salida se combinan mediante una función booleana de tres variables.

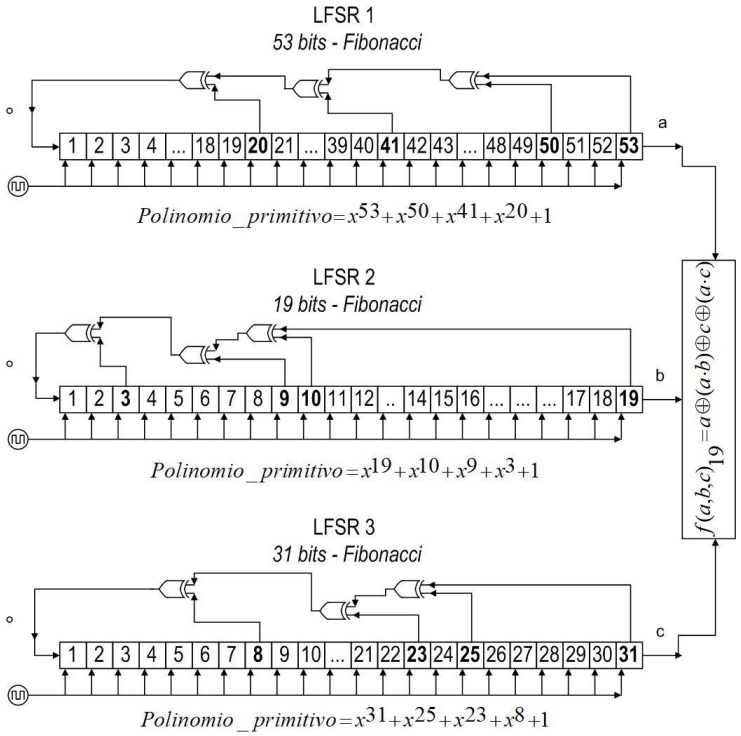

Fig. 4. Generador 1 conformado por tres LFSR tipo Fibonacci

5 Generador 2

Conformado por tres LFSR, con sus respectivos polinomios primitivos, cuyas secuencias de salida se combinan mediante una función booleana de tres variables.

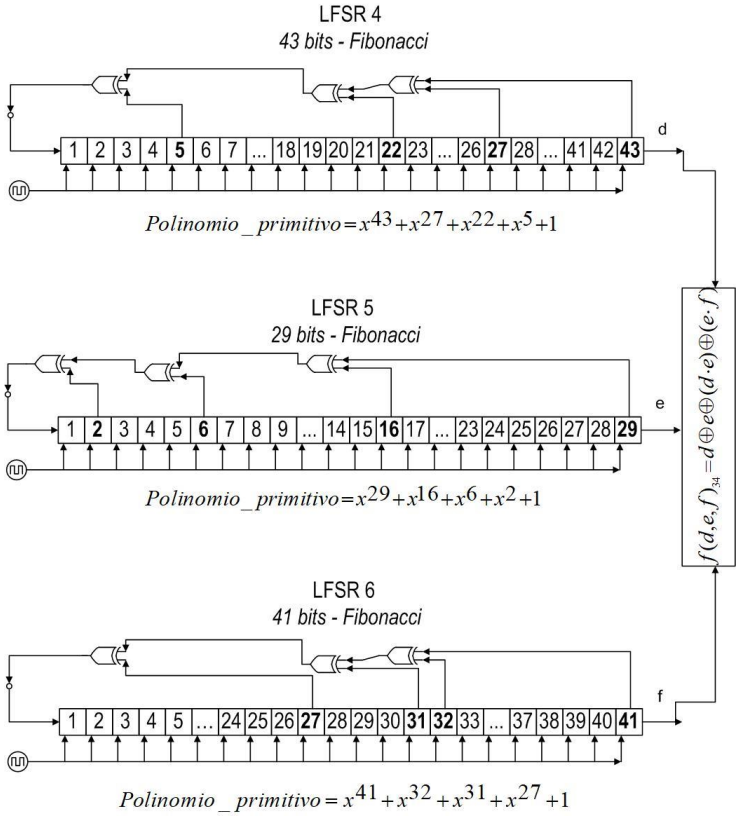

Fig. 5. Generador 2 conformado por tres LFSR tipo Fibonacci

6 Generador 3

Conformado por tres LFSR, con sus respectivos polinomios primitivos, cuyas secuencias de salida se combinan mediante una función booleana de tres variables.

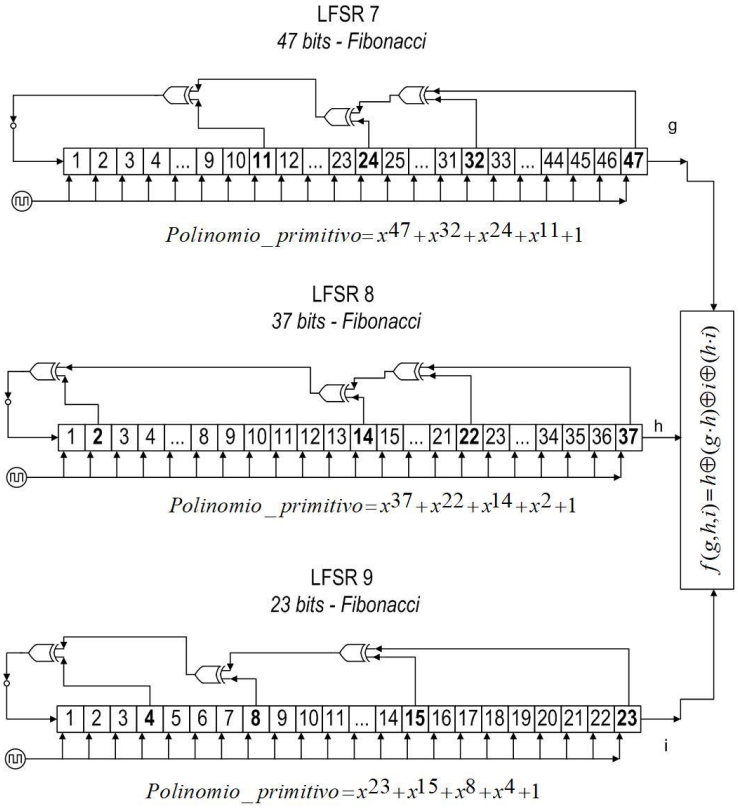

Fig. 6. Generador 3 conformado por tres LFSR tipo Fibonacci

7 Generador Combinacional

El Generador Combinacional usa dos funciones booleanas como funciones de combinación no lineal. Una de tres y otra de cuatro variables, que cumplen con el criterio de ser balanceadas y alta no linealidad

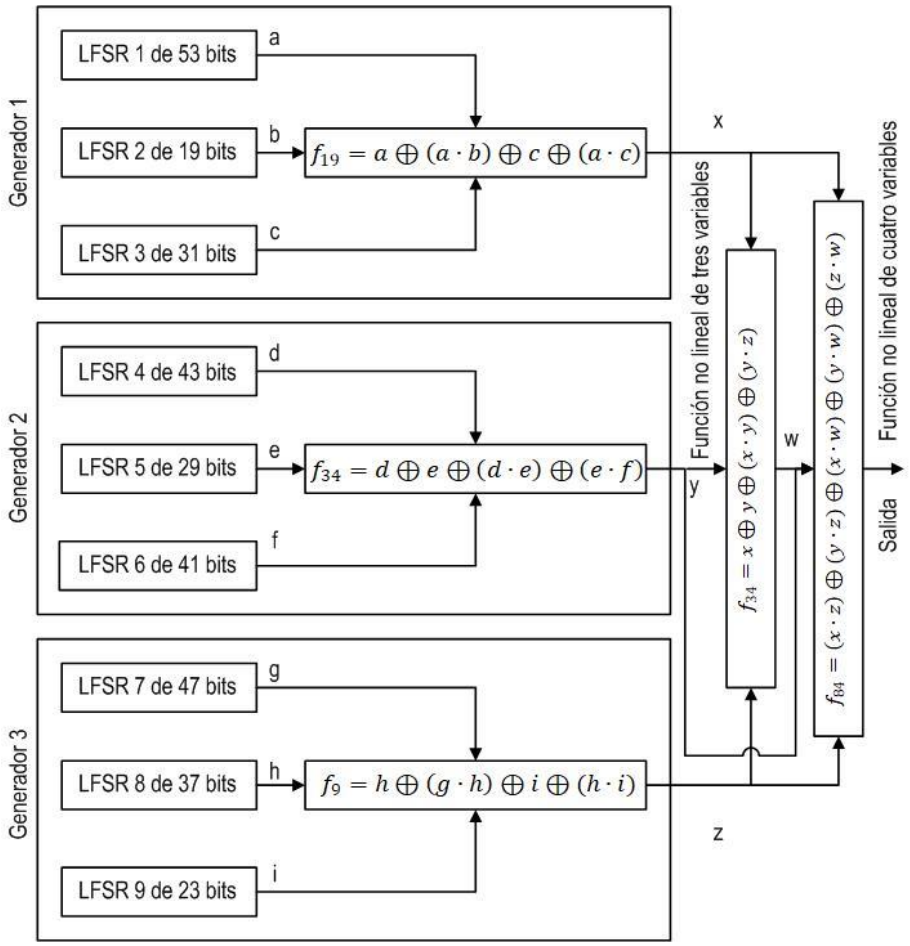

Fig. 7. Generador Combinacional

8 Clave

Para originar los estados iniciales (semillas) de los distintos LFSR se realiza un proceso que utiliza una clave de una longitud de 32 caracteres, que expresados 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 serán permitidos.

Se recurre a vectores con una distribución aleatoria de las posiciones, para obtenerla se utiliza a un generador de números aleatorios, en esta ocasión se adopta un generador congruencial multiplicativo.

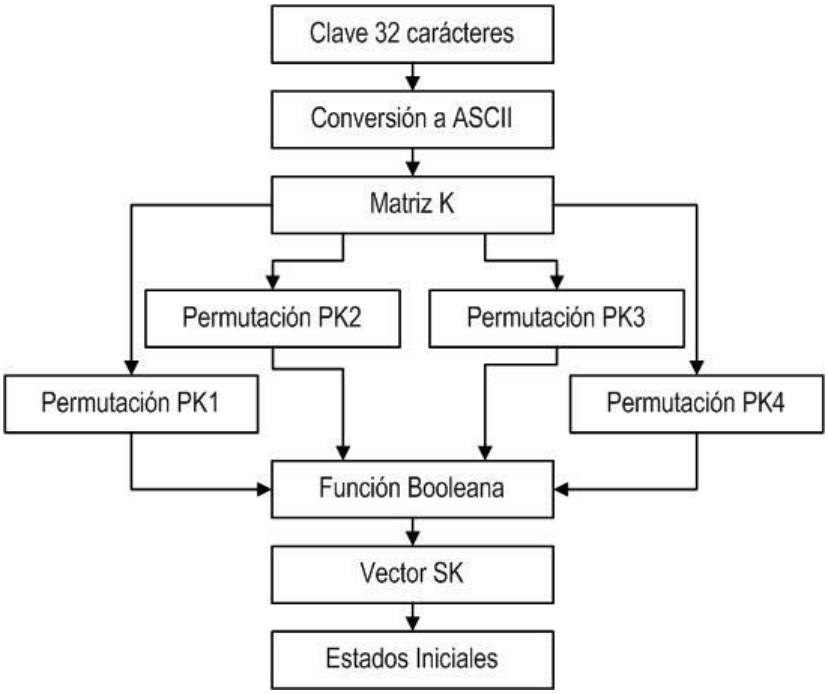

Fig. 8. Clave

8.1 Permutación 1

Para realizar la permutación de la clave se utiliza un vector que modifica las posiciones en forma aleatoria, esto se calcula con un generador congruencial multiplicativo, con semilla: $x_0 = 3249$

$$x_{i+1} = (1747 \cdot x_i) mod\ 1048576$$

8.2 Permutación 2

Para realizar la permutación de la clave se utiliza un vector que modifica las posiciones en forma aleatotia, esto se calcula con un generador congruencial multiplicativo, con semilla: $x_0 = 3271$

$$x_{i+1} = (1753 \cdot x_i) mod\ 1048576$$

8.3 Permutación 3

Para realizar la permutación de la clave se utiliza un vector que modifica las

posiciones en forma aleatotia, esto se calcula con un generador congruencial multiplicativo, con semilla: $x_0 = 3301$

$$x_{i+1} = (1759 \cdot x_i) mod\ 1048576$$

8.4 Permutación 4

Para realizar la permutación de la clave se utiliza un vector que modifica las posiciones en forma aleatotia, esto se calcula con un generador congruencial multiplicativo, con semilla: $x_0 = 3347$

$$x_{i+1} = (1777 \cdot x_i) mod\ 1048576$$

8.5 Función booleana

La función booleana que procesa los cuatro vectores K1, K2, K3 y K4 es la siguiente:

$$MK = K2 \oplus (K1 \cdot K3) \oplus (K2 \cdot K3) \oplus (K1 \cdot K4) \oplus (K2 \cdot K4)$$

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

9 Pruebas Básicas Estadísticas

A las secuencias producidas por este generador le haremos un control sobre su aleatoriedad y para ello utilizamos cinco pruebas estadísticas básicas [7]:

Prueba de Frecuencia (Monobit). El propósito de esta prueba es determinar si el número de 0 y 1 en s es aproximadamente el mismo, como se esperaría para una secuencia aleatoria, el estadístico χ_1 se determina con la expresión: $\chi_1 = \frac{(n_0 - n_1)^2}{n}$. El valor calculado se compara con la distribución χ^2con un grado de libertad $\nu = 1$.

Prueba de Series. El propósito de esta prueba es determinar si el número de ocurrencias de las cuatro formas (00, 01, 10, 11) de dos bits es aproximadamente el mismo número esperado para una secuencia aleatoria.
Las secuencias aleatorias son uniformes, por lo que cada forma de 2 bits tiene la misma probabilidad de ocurrencia, el estadístico χ_2 se determina con la expresión: $\chi_2 = \frac{4}{n-1} \cdot (n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n} \cdot (n_0^2 + n_1^2) + 1$. El valor calculado se compara con la distribución χ^2con un grado de libertad $\nu = 2$.

Prueba de Póker. Esta prueba se usa para probar la aleatoriedad de un patrón de 3 bits en una secuencia, el estadístico χ_3 se determina con la expresión $\chi_3 =$

$\sum_{\mathrm{i}}^{\mathrm{d}} \frac{(\mathrm{A(d)_i}-\mathrm{E_i})^2}{\mathrm{E_i}}$, donde $\mathrm{E_i} = \mathrm{C_i^d} \cdot \left(\frac{\mathrm{L}}{2^{\mathrm{d}}.\mathrm{d}}\right)$. El valor calculado se compara con la distribución χ^2con un grado de libertad $\nu = 2^d - 1$.

Prueba de Autocorrelación. El propósito de esta prueba es verificar las correlaciones entre la secuencia s y las versiones desplazadas (no cíclicas) de la misma. Sea d un entero fijo $1 \leq d \leq \left\lfloor \frac{n}{2} \right\rfloor$. El número de bits en s que no es igual a los desplazados d es $A_{(d)} = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}$.

Prueba de Rachas. El objetivo de esta prueba es calcular el número total de secuencias ininterrumpidas de bits igual a cero o uno. Una secuencia ininterrumpida de bits idénticos de longitud k es una subsecuencia compuesta por k bits del mismo valor (cero o uno) y que está limitada por un bit de valor diferente. El propósito de la prueba de ejecuciones es determinar si el número de ejecuciones (de ceros o unos) de varias longitudes en la secuencia s es el esperado para una secuencia aleatoria. El número esperado de espacios (o bloques) de longitud i en una secuencia aleatoria de longitud n viene dado por la fórmula $e_i = \frac{(n-i+3)}{2^{i+2}}$. El estadístico χ_4 se determina con la expresión $\chi_4 = \sum_{i=1}^{k} \frac{(B_i-e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i-e_i)^2}{e_i}$. El valor calculado se compara con la distribución χ^2con un grado de libertad $\nu = 2k - 2$.

10 Puesta en Funcionamiento y Realización de las Pruebas Básicas

El paso siguiente es poner en funcionamiento el dispositivo con 100 claves distintas, generando secuencias de 1.000.000 bits y verificar la condición de aleatoriedad de las mismas, para un nivel de significancia: $\alpha = 0{,}01$, analizados de un lado y de los dos lados.

Debido al gran volumen de procesamiento requerido, se desarrolló un programa escrito en lenguaje C++, que contenía 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.

Para agilizar el trabajo computacional, se optó por un conjunto de cinco pruebas estadísticas básicas para secuencias binarias, recomendadas por Menezes (et al.) [7].

Esta elección no pretendió descartar las otras baterías de pruebas, después para exámenes más minuciosos, se pueden someter los resultados a otros conjuntos de pruebas, tal como las indicadas en la Norma NIST Special Publication 800-22, en el trabajo de Rukhin (et al.) [8].

Se debe cumplir la hipótesis nula:

$$H_0 \rightarrow estadístico\ obtenido < valor\ máximo$$

Tabla.2. Valores máximos para $\boldsymbol{\alpha = 0,01}$

Prueba	X_i	α	ν	Tabla	Valor máximo.
Frecuencia	X_1	0,01	1	χ^2	6,6349
Series	X_2	0,01	2	χ^2	9,2103
Póquer	X_3	0,01	3	χ^2	11,3449
Autocorrelación	X_4	0,01	-	z	2,3263
Rachas	X_5	0,01	28	χ^2	48,2782

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, procedimiento similar a los gráficos de control de calidad, donde k es el número de muestras.

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

En nuestro caso $k = 100$ y el nivel de significancia elegido es: α = 0.05, los límites quedan en:

$$LS = (1-0{,}01) + 3 \cdot \sqrt{\frac{0{,}01(1-0{,}01)}{100}} = 1{,}02$$

$$LI = (1-0{,}01) - 3 \cdot \sqrt{\frac{0{,}01(1-0{,}01)}{100}} = 0{,}96$$

10.2 Pruebas sobre el Generador Combinacional

Se realizan pruebas estadísticas al generador, para verificar la aleatoriedad de sus secuencias de salida.

Tabla.3. Resultados finales del generador

Prueba	Pasan	Proporción	Límite superior	Límite inferior
Frecuencia	99	0,99	1,02	0,96
Series	98	0,98	1,02	0,96
Póquer	99	0,99	1,02	0,96
Autocorrelación	98	0,98	1,02	0,96
Rachas	98	0,98	1,02	0,96

Los resultados se muestran en un diagrama de puntos, para ver si los resultados están dentro de los límites.

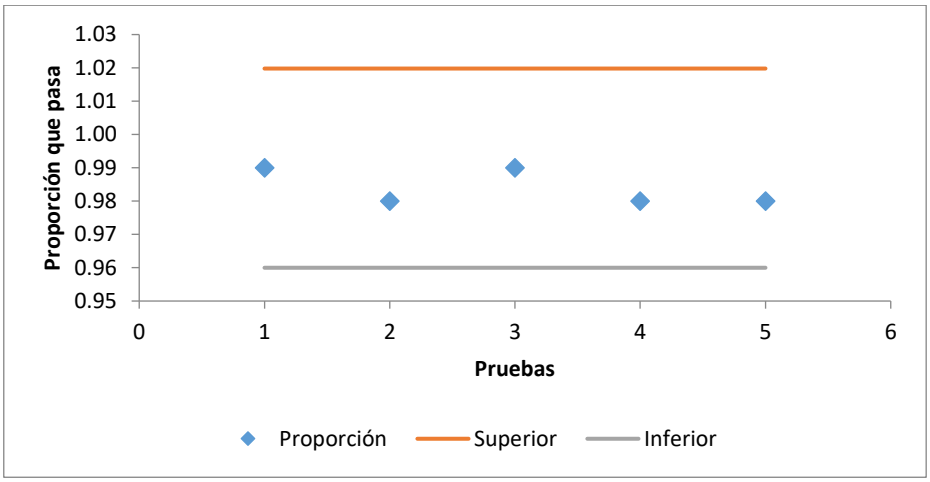

Fig. 9. Proporción de muestras que pasan las pruebas

El generador, con la función indicada, es aceptado porque las proporciones de las muestras que pasan las pruebas están dentro de los límites.

11 Comparación de frecuencias de carácteres

Gráfico de frecuencias de carácteres de:texto plano y texto cifrado.

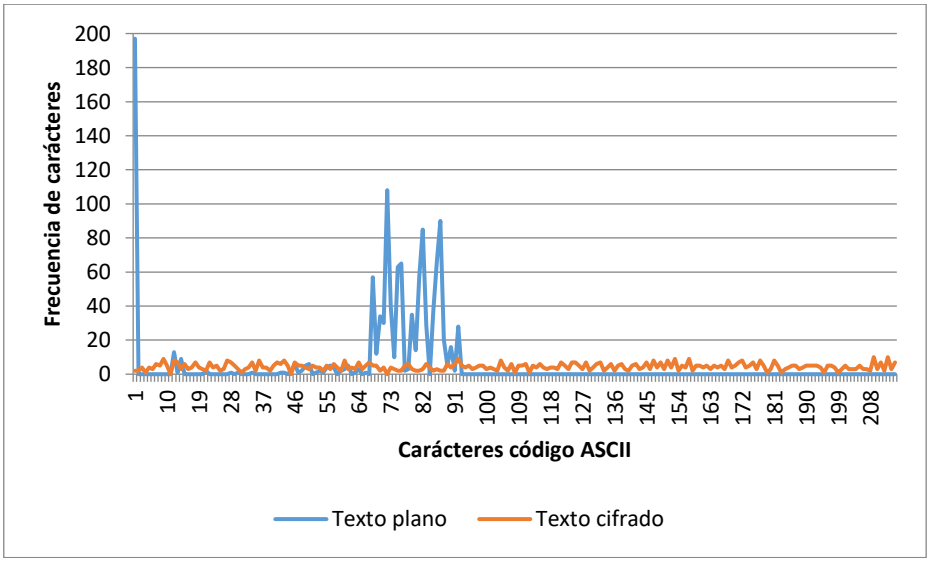

Fig. 10. Frecuencias de caracteres del texto plano y cifrado

Conclusiones

Tal como se mencionó oportunamente, la generación de bits aleatorios de alta calidad criptográfica resulta de alto interés. En consecuencia, se desarrolló un generador de 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 tres generadores, conformados por tres LFSR combinados por una función booleana. Las funciones de combinación final de estos tres generadores, son dos funciones de tres y cuatro variables.

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

La función booleana que es la responsable del proceso no lineal, asegura las mejores prestaciones criptográficas, partiendo de funciones balanceadas expresadas de diversas formas, los resultados fueron positivos, por lo que el modelo propuesto se considera válido para la generación de secuencias pseudoaleatorias, aptas para el cifrado de flujo.

Para trabajos futuros se pueden utilizar otros generadores de secuencias binarias pseudoaleatorias de mayor complejidad y con claves de mayor longitud.

Referencias

1. Paar, C., Pelzl, L., D., “Understanding Cryptography. Springer, 2010.”, Springer, 2010.
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. Stahnke, W., “Primitive Binary Polynomials”, Mathematics of Computation, 27. 124, pp. 977-980, 1973.
4. Mioc, M. and Stratulat, M., “Study of Software implementation for Linear Feedback Shift Register Based on 8th Degree Irreducible Polynomials”, International Journal of Computers, 8, 2014.
5. Braeken, A.: Cryptographic Properties of Boolean Functions and S-Boxes. Faculteit Ingenieurswetenschappen. Katholieke Universiteit Leuven (2003)
6. 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)
7. Menezes, A.,Van Oorschot, P. and Vanstone, S., “Handbook of Applied Cryptography”, Massachusetts Institute of Technology, 1996.
8. 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.