

# **Cifrador de Bloque de Dos Algoritmos Cifradores Paralelos Conformados por Secuencias Entrelazadas de Polinomios Diferentes, en Modo de Encadenamiento de Bloques de Cifrado en Propagación**

Andrés Francisco Farías<sup>1</sup>, Germán Antonio Montejano<sup>2</sup>,

Ana Gabriela Garis<sup>3</sup>, Andrés Alejandro Farías<sup>4</sup>

National University of La Rioja, La Rioja, Argentina<sup>1,4</sup>

National University of San Luis, San Luis, Argentina<sup>2,3</sup>

afarias665@yahoo.com.ar<sup>1</sup>, gmonte@unsl.edu.ar<sup>2</sup>,

agaris@gmail.com<sup>3</sup>, andres\_af86@hotmail.com<sup>4</sup>

## **Abstract.**

Diseño de un Cifrador de bloque de 256 bits, con clave de 128 bits, y vector de inicialización de 256 bits, a partir de la estructura de una red de Feistel, con dos algoritmos cifradores paralelos, con modo de encadenamiento de bloques de cifrado de propagación (PCBC, Propagating Cipher Block Chaining).

El primer algoritmo está conformado por una secuencia 3-entrelazada obtenida a partir de secuencias producidas por polinomios primitivos diferentes.

El segundo algoritmo está conformado por una secuencia 2-entrelazada lograda de secuencias entregadas por polinomios primitivos diferentes. Finalmente el texto cifrado obtenido fue sometido a conjunto de pruebas estadísticas de aleatoriedad.

**Keywords:** LFSR, cipher, key, boolean function, non-linearity

## **1 Introducción**

El presente documento expone el desarrollo de un cifrador de bloque, basado en una red de Feistel que permite el cifrado y descifrado utilizando la misma estructura, donde para el caso del descifrado se utilizan las subclaves cambiando el orden de las mismas [1], [2] y [3]. La clave adoptada es de 16 caracteres, es decir 128 bits,

El tamaño de los bloques es de 256 bits, con clave de 128 bits, y vector de inicialización de 256 bits. El cifrador es una red de Feistel, con dos algoritmos cifradores paralelos, con modo de encadenamiento de bloques de cifrado de propagación (PCBC, Propagating Cipher Block Chaining),

El primer algoritmo de cifrado está conformado por una secuencia 3-entrelazada obtenida a partir de secuencias pseudoaleatorias producidas por polinomios primitivos diferentes que operan sobre un Linear Feedback Shift Registers (LFSR) de 71 bits .

El segundo algoritmo está compuesto por una secuencia 2-entrelazada lograda a partir de cadenas pseudoaleatorias obtenidas por polinomios primitivos distintos que trabajan sobre Linear Feedback Shift Registers (LFSR) de 67 bits.

El texto cifrado completo obtenido al final del proceso de encriptación, fue sometido a conjunto de pruebas estadísticas, para verificar su aleatoriedad.

## 2 Esquema del cifrador

El cifrado de bloque se denomina así por realizar el proceso de encriptación trabajando sobre cadenas de texto de igual longitud. En este caso se utilizaron bloques de 256 bits, luego esos bloques son ensamblados siguiendo el modo de encadenamiento de bloques de cifrado de propagación (Propagating Cipher Block Chaining, PCBC)). Básicamente la estructura del cifrador está conformada por una red de Feistel que para desarrollarla requiere trabajar los siguientes aspectos:

- Red de Feistel para cifrado
  - De 64 rondas
  - Con modo de encadenamiento de bloques de cifrado de propagación
- Red de Feistel para descifrado
  - De 64 rondas
  - Con modo de encadenamiento de bloques de cifrado de propagación
- Clave y subclaves
- Vector de inicialización
- Algoritmos de cifrado
- Secuencias entrelazadas:
  - Secuencia 3-entrelazada
  - Secuencia 2-entrelazada
- Matrices de permutación:
  - IP de 256 bits
  - PC1 de 128 bits
  - PC2 de 128 bits

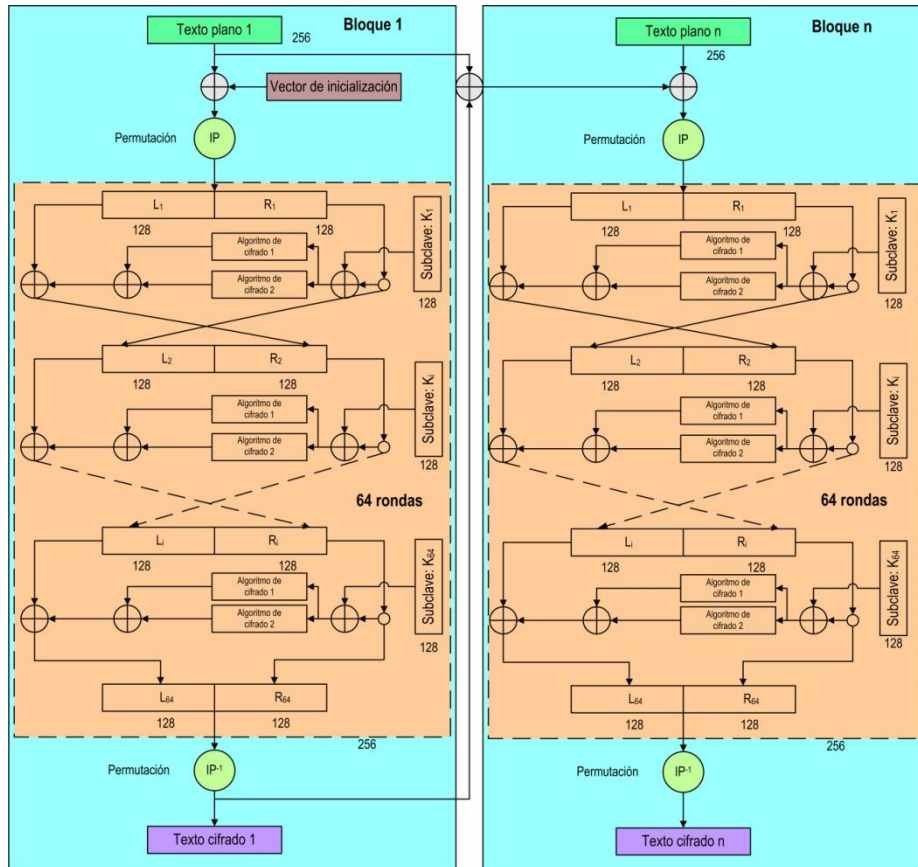
### 2.1 Red de Feistel para cifrado

El proceso de cifrado consiste en dividir el texto plano en bloques de 256 bits, el primer bloque es sometido a una operación XOR con el vector de inicialización, luego al resultado se le realiza una permutación IP.

La salida de la permutación entra en la red de Feistel, que se detalla en la figura 1 y se producen 64 rondas, con sus respectivas subclaves, después se realiza una permutación  $IP^{-1}$ , para obtener el primer bloque de texto cifrado.

Para los siguientes bloques de texto plano, se realiza una operación XOR con los bloques de texto plano y cifrado del primer bloque y al resultado se le ejecuta una nueva operación XOR con el texto plano del bloque y la salida sufre una permutación IP antes de entrar a la red de Feistel y producir 64 rondas, con las subclaves correspondientes,,

Después de esta operación se calcula la permutación  $IP^{-1}$  y se consigue un nuevo bloque de texto cifrado y así sucesivamente hasta completar el cifrado de todos los bloques de texto plano.



**Fig. 1.** Red de Feistel para cifrado

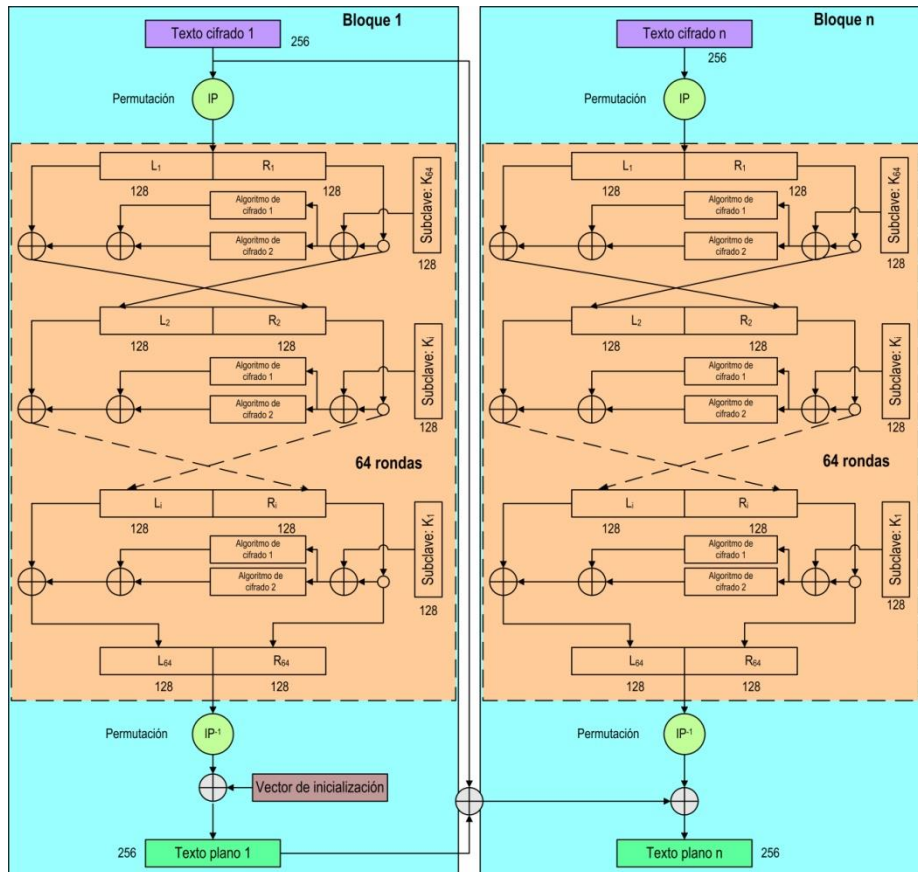
## 2.2 Red de Feistel para descifrado

La Red de Feistel para descifrado es similar a la anterior, pero en este caso se toma el texto cifrado y se lo divide en bloques de 256 bits, figura 2.

Para el primer bloque de texto cifrado se realiza una permutación  $IP$  antes de entrar a la red de Feistel y realizar 64 rondas, con las claves introducidas en modo inverso, al resultado se le realiza una permutación  $IP^{-1}$  y luego se produce una operación XOR con el vector de inicialización para obtener el primer bloque de texto plano.

Para el resto de los bloques de texto cifrado, el proceso comienza con la permutación  $IP$ , después se ingresa a la red de Feistel y se llevan a cabo 64 rondas, con las subclaves ingresadas en modo inverso.

Finalmente después de este proceso se hace una permutación  $IP^{-1}$  y a la salida se le aplica una operación XOR con la resultante de la operación XOR entre el texto cifrado y texto plano del bloque anterior, para lograr un nuevo bloque de texto plano.



**Fig. 2.** Red de Feistel para descifrado

### 2.3 Clave y subclaves

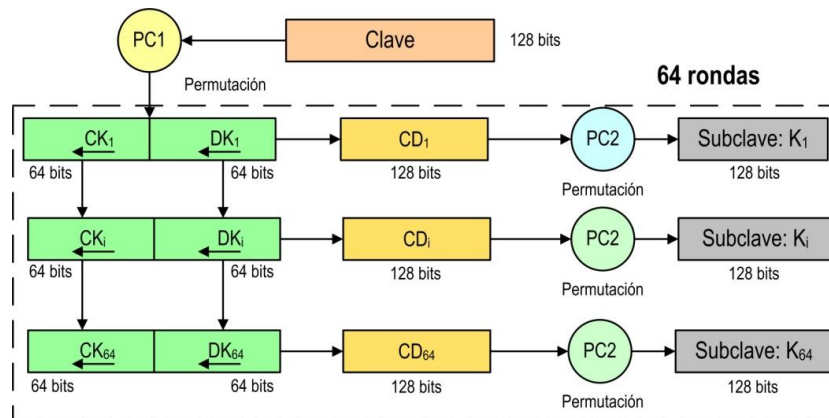
Como se dijo previamente, la clave está conformada con 16 caracteres (128 bits), de la que se obtienen 64 subclaves de 128 bits, siguiendo los pasos que se muestran en la figura 3.

La clave es sometida a una permutación según la matriz de permutación PC1, luego se divide el bloque de 128 bits resultante en dos bloques de 64 bits, los que sufren desplazamiento de las posiciones de los bits de manera de tener 64 pares de bloques de 64 bits que corresponderán a las 64 subclaves.

En los pares de las rondas: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55 y 60 los bits se

desplazan dos posiciones a la izquierda, en el resto de los pares el desplazamiento es de una posición a la izquierda.

Esos pares son ensamblados y luego sometidos a la permutación PC2, para obtener las 64 subclaves finales.



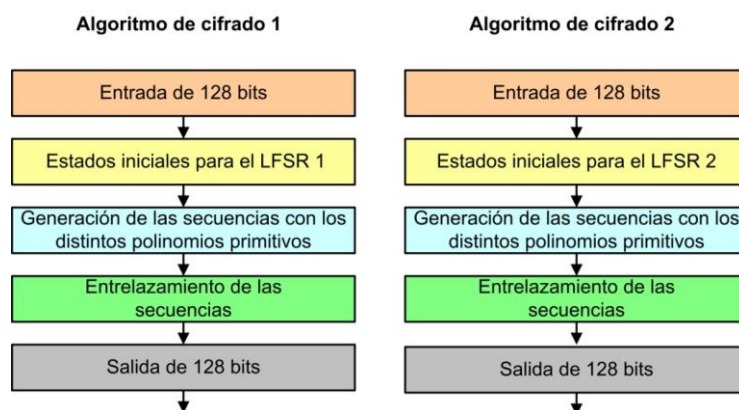
**Fig. 3.** Tratamiento de las subclaves

## 2.4 Vector de inicialización

Es para iniciar las tareas de encadenamiento de bloques, tanto de cifrado como de descifrado. Es única para todo el proceso, debe ser secreta como la clave y su longitud es igual a la de los bloques: 256 bits.

## 2.5 Algoritmos de cifrado

Los algoritmos de cifrado tienen la configuración que se indica en la figura 4.



**Fig. 4.** Algoritmos de cifrado 1 y 2

Tienen una entrada de 128 bits, que conforman los estados iniciales para los LFSR, que una vez cargados, realizan 128 ciclos con los distintos polinomios primitivos de conexión que producen la secuencias respectivas, las que luego se entrecruzan, entregando 128 bits de salida.

## 2.6 Secuencias t-entrelazadas

Tenemos las siguientes t-secuencias entrelazadas con polinomios primitivos diferentes [4] :

**Secuencia 3-entrelazada con polinomios primitivos diferentes.** Los LFSR tienen una longitud de 71 bits y en tabla 1, se indican los polinomios primitivos [5], [6], [7] y [8].

**Tabla 1.** LFSR, longitudes y polinomios primitivos del generador

LFSR	Longitud	Polinomios primitivos
1	71	$P(x)_1 = x^{71} + x^{59} + x^{53} + x^{48} + 1$
2	71	$P(x)_2 = x^{71} + x^6 + 1$
3	71	$P(x)_3 = x^{71} + x^{49} + x^{45} + x^{34} + x^{30} + x^{21} + 1$

**Secuencia 2-entrelazada con polinomios primitivos diferentes.** Los LFSR tienen una longitud de 67 bits y en tabla 2, se indican los polinomios primitivos [5], [6], [7] y [8].

**Tabla 2.** LFSR, longitudes y polinomios primitivos del generador

LFSR	Longitud	Polinomios primitivos
1	67	$P(x)_1 = x^{67} + x^{61} + x^{33} + x^3 + 1$
2	67	$P(x)_2 = x^{67} + x^{64} + x^{44} + x^{28} + x^{26} + x^{25} + 1$

## 2.7 Matrices de Permutación

Se recurre a una matriz con una distribución aleatoria de las posiciones, para obtenerlas se utiliza un generador de números aleatorios, en esta ocasión se adopta un generador congruencial multiplicativo [9]. En tabla 3 se observan los valores:

**Generador congruencial multiplicativo.** El generador tiene la siguiente expresión:

$$x_{i+1} = (a_x \cdot x_i) \bmod m_x$$

Donde:

$$a_x = \text{multiplicador}$$

$$m_x = \text{módulo}$$

$$x_0 = \text{semilla}$$

**Tabla 3.** Matriz IP, PC1 y PC2

Matriz	módulo	multiplicador	semilla
IP	1048576	1279	1153
PC1	1048576	1597	1531
PC2	1048576	1933	1759

### 3 Elección de las pruebas estadísticas

#### 3.1 Pruebas de aleatoriedad

El conjunto de pruebas estadísticas para generadores de números aleatorios y pseudoaleatorios para aplicaciones criptográficas fueron seleccionadas de la Publicación especial 800-22 revisión 1a del Instituto Nacional de Estándares y Tecnología (NIST), del trabajo de Rukhin (et al.) [10]. La tabla 4 muestra las pruebas estadísticas para números aleatorios y pseudoaleatorios adoptadas.

**Tabla 4.** Pruebas estadísticas para números aleatorios y pseudoaleatorios

Pruebas estadísticas para números aleatorios y pseudoaleatorios	
1	Frecuencia (monobit)
2	Frecuencia dentro de un bloque
3	Entropía aproximada
4	Sumas acumuladas
5	Rachas
6	Prueba en serie
7	Universal
8	Coincidencia de plantillas no superpuestas
9	Complejidad lineal
10	Transformada discreta de Fourier (espectral)

#### 3.2 Pruebas sobre el cifrador

Se analizaron cien secuencias binarias de 25.600 de bits, obtenidas del cifrador a partir de cien claves diferentes.

El nivel de significancia adoptado para las pruebas estadísticas es:  $\alpha = 0,01$

La hipótesis nula es:  $H_0 \rightarrow p\_value > 0,01$

#### 3.3 Interpretación de los resultados

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

- Proporción de muestras que pasan las pruebas
- Prueba de Uniformidad de los p-valor

**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:

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

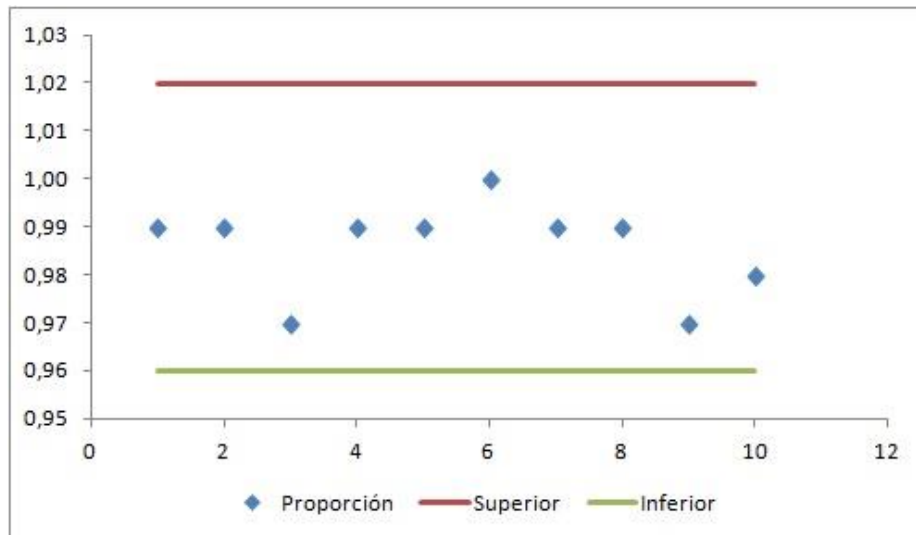
En nuestro caso el número de muestras  $k = 100$  y el nivel de significancia elegido es:  $\alpha = 0.01$ , los límites quedan:  $LS = 1,02$   $LI = 0,96$

Se consideran todas pruebas, los resultados se indican en la tabla 5:

**Tabla 5.** Pruebas estadísticas para números aleatorios y pseudoaleatorios

	Pruebas estadísticas para números aleatorios y pseudoaleatorios	Total	Pasan	Propor.	Superior	Inferior
1	Frecuencia (monobit)	100	99	0,99	1,02	0,96
2	Frecuencia dentro de un bloque	100	99	0,99	1,02	0,96
3	Entropía aproximada	100	97	0,97	1,02	0,96
4	Sumas acumuladas	100	99	0,99	1,02	0,96
5	Rachas	100	99	0,99	1,02	0,96
6	Prueba en serie	100	100	1,00	1,02	0,96
7	Universal	100	99	0,99	1,02	0,96
8	Coincidencia de plantillas no superpuestas	100	99	0,99	1,02	0,96
9	Complejidad lineal	100	97	0,97	1,02	0,96
10	Transformada discreta de Fourier (espectral)	100	98	0,98	1,02	0,96

En la figura 5, se aparece el gráfico de puntos en función de los datos de la tabla 5



**Fig. 5.** Gráfico de puntos



**Distribución Uniforme de los P-valor.** Se realizan pruebas 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. En tabla 6, se indican los resultados satisfactorios obtenidos:

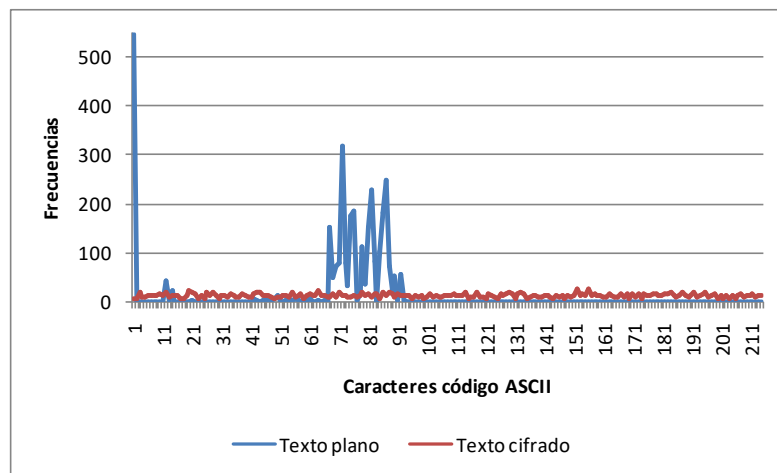
**Tabla 6.** Pruebas p-valor

Pruebas	p-valor	p-valor límite	Pasa
Frecuencia (monobit)	0,262	0,0001	Sí
Frecuencia dentro de un bloque	0,029	0,0001	Sí
Entropía aproximada	0,000	0,0001	Sí
Sumas acumuladas	0,475	0,0001	Sí
Rachas	0,475	0,0001	Sí
Prueba en serie	0,067	0,0001	Sí
Universal	0,475	0,0001	Sí
Coincidencia de plantillas no superpuestas	0,972	0,0001	Sí
Complejidad lineal	0,103	0,0001	Sí
Transformada discreta de Fourier (espectral)	0,071	0,0001	Sí

**Análisis final.** En definitiva, las secuencias que entrega el cifrador superan los dos métodos de interpretación de resultados, por lo tanto son pseudoaleatorias

#### 4 Comparación de frecuencias

Superposición de gráficos de frecuencias para observar las diferencias entre texto plano y texto cifrado, en figura 6:



**Fig. 6.** Frecuencias de caracteres del texto plano y cifrado

## 5 Conclusiones y Trabajos Futuros

Un cifrador de bloque de 256 bits, con algunas propiedades importantes tales como clave de 128 bits y vector de inicialización de 256 bits y la incorporación de algoritmos de cifrado que contienen secuencias entrelazadas.

Para futuras versiones se pueden incorporar entre otras cosas: claves más largas y mayor cantidad de algoritmos y otros métodos de concatenación de bloques.

El resultado obtenido del texto cifrado tiene una frecuencia de caracteres aleatorios, lo que hace difícil un criptoanálisis basado en la estadística de aparición de caracteres.

### Referencias

1. Karakoç, F., Demirci, H., Harmanc, A.: AKF: A Key Alternating Feistel Scheme for Lightweight Cipher Designs, *Information Processing Letters*. 115, 359--367 (2015)
2. Bogdanov, A.: Analysis and Design of Block Cipher Constructions. Fakultät für Elektrotechnik und Informationstechnik an der Ruhr-Universität Bochum (2009)
3. García Méndez, P.: Descripción Polinomial de los Sistemas de Cifrado DES y AES. Universidad Autónoma Mexicana, México (2011)
4. Cardell, S., Fúster Sabater, A., Requena, V., PN-secuencias entrelazadas de polinomios diferentes, RECSI 2022
5. F. Massodi, S. Alam S. and M. Bokhari, "A Analysis of Linear Feedback Shift Registers in Stream Ciphers", 2012, *International Journal of Computer Application*, 16 (17), pp. 0975-887.
6. A. Menezes, P. Van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", 1996, Massachusetts Institute of Technology.
7. C. Parr and L. Pelzl, "Understanding Cryptography", 2010, Springer.
8. W. Stahnke, "Primitive Binary Polynomials", 1973, *Mathematics of Computation*, 27 (124), pp. 977-980.
9. Fishman, G.: Multiplicative Congruential Random Number Generators with Modulus  $2^{\beta}$  : An Exhaustive Analysis for  $\beta = 32$  and a Partial Analysis for  $\beta = 48$ . *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).