Cifrador de Bloque de Dos Algoritmos Cifradores Paralelos Conformados por Secuencias Entrelazadas de un Único Polinomio Primitivo, en Modo de Encadenamiento de Bloques de Cifrado en Propagación

Andrés Francisco Farías[1], Germán Antonio Montejano[2],

Ana Gabriela Garis[3], Andrés Alejandro Farías[4]

National University of La Rioja, La Rioja, Argentina[1, 4]
National University of San Luis, San Luis, Argentina[2, 3]
afarias665@yahoo.com.ar[1], gmonte@unsl.edu.ar[2],
agaris@gmail.com[3], andresaf86@hotmail.com[4]

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 con polinomio primitivo único, originada por un Linear Feedback Shift Registers (LFSR) de 71 bits. El segundo algoritmo está conformado por una secuencia 2-entrelazada con polinomio primitivo único, produida por un Linear Feedback Shift Registers (LFSR) de 67 bits. 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 un único polinomio primitivo que opera 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 un único polinomio primitivo que trabaja sobre un 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 95 rondas, con Modo de Encadenamiento de Bloques de Cifrado de Propagación (PCBC)
- Red de Feistel para descifrado
    - De 95 rondas, con Modo de Encadenamiento de Bloques de Cifrado de Propagación (PCBC)
- 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 95 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 95 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.

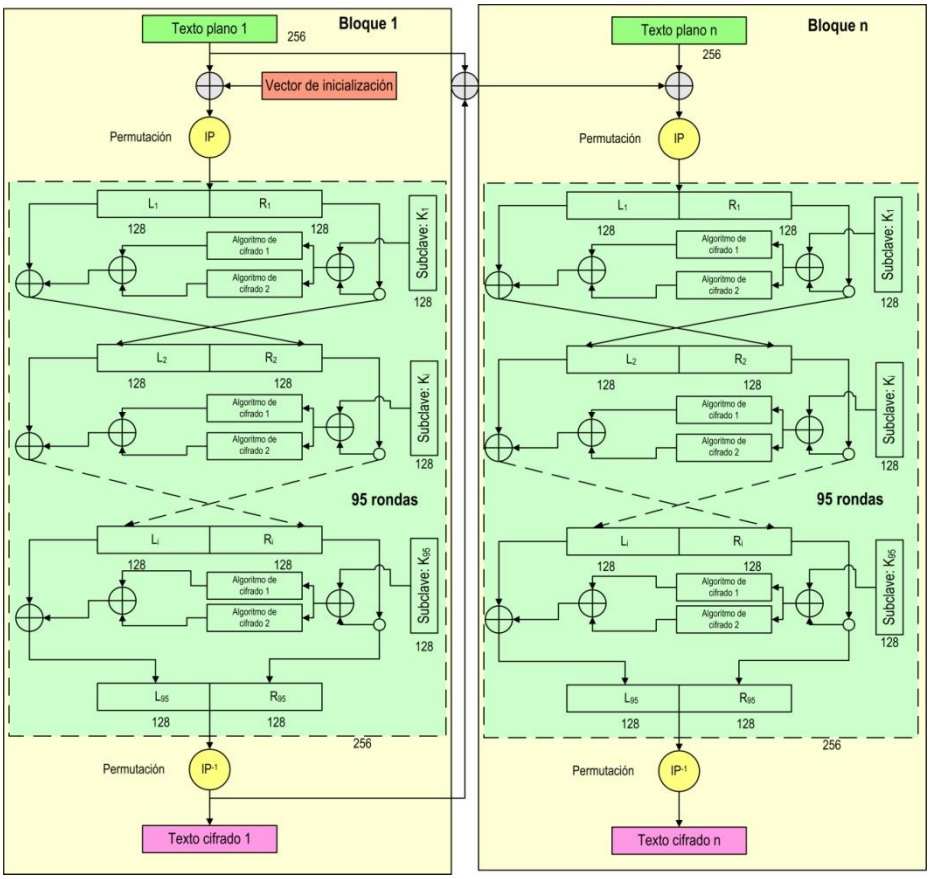

Fig. 1. Red de Feistel para cifrado

2.2 Red de Feistel para descifrado

La Red de Feistel para descrifrado 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 95 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 95 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.

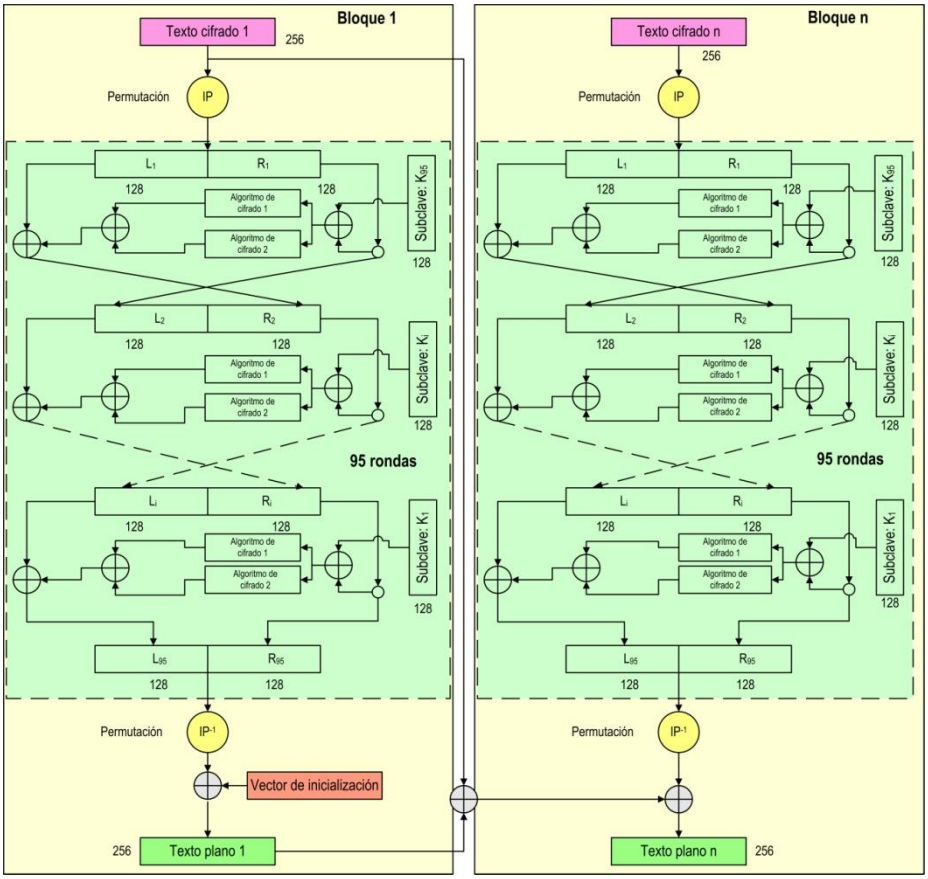

Fig. 2. Red de Feistel para descifrado

2.3 Clave y subclaves

Como se dijo previamente, la clave está conformada con 16 caracteres (128 bits), de la que se obtienen 95 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 95 pares de bloques de 64 bits que corresponderán a las 95 subclaves.

En los pares de las rondas: múltiplos de 5, los bits se desplazan dos posicionres 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 95 subclaves finales.

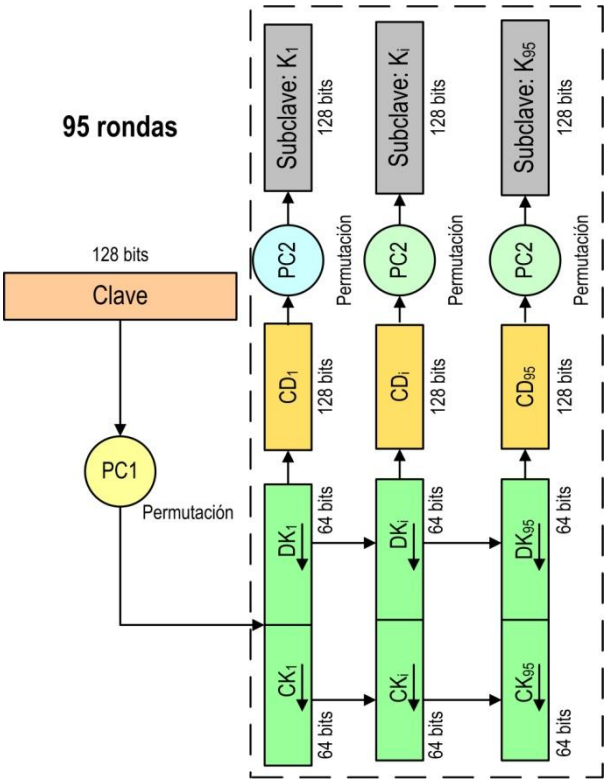

Fig. 3. Tratamiento de las subclaves

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. 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 polinomio primitivo único para los algoritmos de cifrado 1 y 2 [4] :

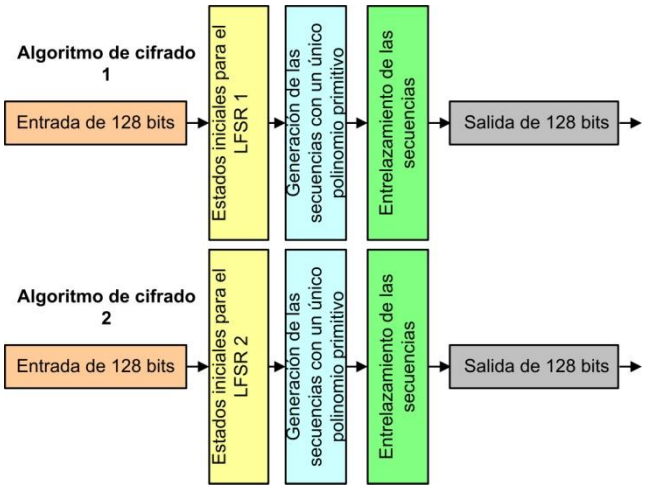

Fig. 4. Algoritmos de cifrado 1 y 2

Secuencia 3-entrelazada con polinomio primitivo único. Los LFSR tienen una longitud de 71 bits y en Tabla 1, se indica el polinomio.primitivo único [5], [6], [7] y [8].

Tabla 1. LFSR, longitudes y polinomio primitivo del generador

LFSR	Longitud	Polinomio primitivo
1	71	$P(x)_2 = x^{71} + x^{49} + x^{45} + x^{34} + x^{30} + x^{21} + 1$

Secuencia 2-entrelazada con polinomio primitivo único. Los LFSR tienen una longitud de 67 bits y en Tabla 2, se indica el polinomio primtivo único [5], [6], [7] y [8].

Tabla 2. LFSR, longitudes y polinomio primitivo del generador

LFSR	Longitud	Polinomio primitivo
2	67	$P(x)_7 = x^{67} + x^{64} + x^{44} + x^{28} + x^{26} + x^{25} + 1$

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:

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

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

Tabla 3. Matriz IP, PC1 y PC2

Matriz	módulo	multiplicador	semilla
IP	1048576	3137	7369
PC1	1048576	3163	7393
PC2	1048576	3167	7411

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	Prueba de frecuencia dentro de un bloque
3	Prueba de entropía aproximada
4	Prueba de sumas acumuladas
5	Prueba de rachas
6	Prueba serial
7	Prueba estadística universal de Maurer
8	Prueba de coincidencia de plantillas sin superposición
9	Prueba de complejidad lineal
10	Prueba de transformada de Fourier discreta (espectral)

3.2 Pruebas sobre el cifrador

Se analizaron cien secuencias binarias de 100.000 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 y Prueba de Uniformidad de los P-valor

3.4 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: $LS = 1{,}02 \quad LI = 0{,}96$

Se consideran todas pruebas, los resultados fueron satisfactorios como muestran la Tabla 5 y la Figura 5.

Tabla 5. Pruebas estadísticas para números aleatorios y pseudoaleatorios

Pruebas estadísticas para números aleatorios y pseudoaleatorios	Total	Pasan	Propor.	Superior	Inferior
Frecuencia (Monobit)	100	98	0,98	1,02	0,96
Prueba de frecuencia dentro de un bloque	100	100	1,00	1,02	0,96
Prueba de entropía aproximada	100	99	0,99	1,02	0,96
Prueba de sumas acumuladas	100	98	0,98	1,02	0,96
Prueba de rachas	100	99	0,99	1,02	0,96
Prueba serial	100	100	1,00	1,02	0,96
Prueba estadística universal de Maurer	100	99	0,99	1,02	0,96
Prueba de coincidencia de plantillas s/super.	100	97	0,97	1,02	0,96
Prueba de complejidad lineal	100	98	0,98	1,02	0,96
Prueba de transformada de Fourier discreta	100	99	0,99	1,02	0,96

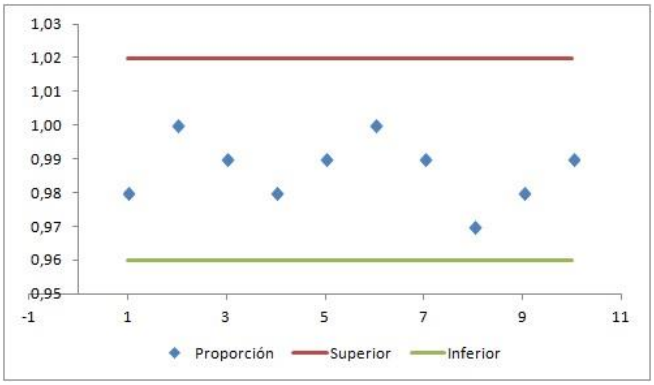

Fig. 5. Proporción de muestras que superan las pruebas

3.5 Distribución Uniforme de los P-valor

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. Se consideran todas pruebas, los resultados fueron satisfactorios como muestran la Tabla 6 y la Figura 6

Tabla 6. Distribución Uniforme de los P-valor

Pruebas	p-valor	p-valor límite	Pasa
Frecuencia (Monobit)	0,8677	0,0001	Sí
Prueba de frecuencia dentro de un bloque	0,4559	0,0001	Sí
Prueba de entropía aproximada	0,0002	0,0001	Sí
Prueba de sumas acumuladas	0,5544	0,0001	Sí
Prueba de rachas	0,8514	0,0001	Sí
Prueba serial	0,5749	0,0001	Sí
Prueba estadística universal de Maurer	0,8514	0,0001	Sí
Prueba de coincidencia de plantillas s/super	0,2493	0,0001	Sí
Prueba de complejidad lineal	0,2368	0,0001	Sí
Prueba de transformada de Fourier discreta	0,7792	0,0001	Sí

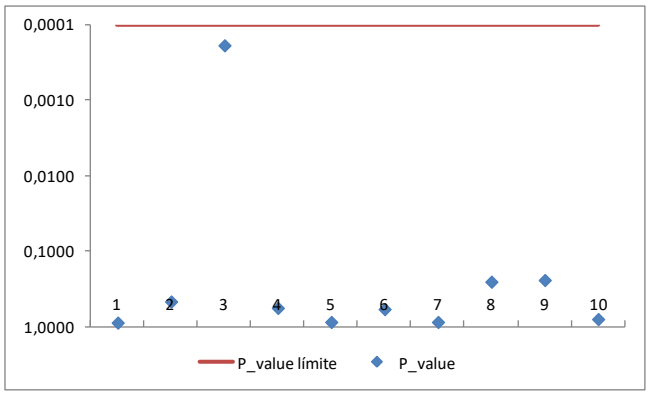

Fig. 6. Distribución Uniforme de los P-valor, eje vertical en escala logarítmica invertida

4 Comparación de frecuencias de caracteres

Superposición de gráficos de frecuencias de caracteres de texto plano y texto cifrado, para observar las diferencias entre ambos, en Figura 7:

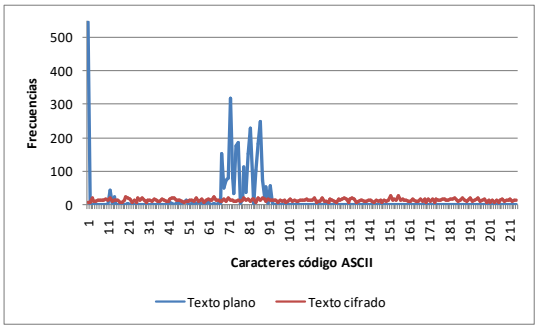

Fig. 7. Frecuencias de caracteres de texto plano y texto cifrado

5 Conclusiones y Trabajos Futuros

Un cifrador de bloque de 256 bits, que cuenta con una clave de 128 bits y vector de inicialización de 256 bits, para los procesos de encadenamiento. Que presenta un diseño novedoso con la incorporación de algoritmos de cifrado paralelos que contienen secuencias t-entrelazadas de un único polinomio primitivo.

Además se presenta un sistema de generación de 95 subclaves a partir de la clave inicial, que se utilizan en las 95 rondas de cifrado y descifrado de la red.

El resultado obtenido del texto cifrado tiene una secuencia de caracteres aleatorios, lo que hace difícil un criptoanálisis basado en la estadística de aparición de caracteres, dicha aleatoriedad es verificada mediante pruebas respectivas.

Para futuras versiones se pueden incorporar entre otras cosas: claves más largas, mayor cantidad de rondas, nuevos procesos de generación de subclaves, otros algoritmos cifrado y la utilización de distintos métodos de concatenación de bloques.

Referencias

1. Karakoç, F., Demirci, H., & Harmancı, A. E. (2015). AKF: *A key alternating Feistel scheme for lightweight cipher design*s. Information Processing Letters, 115(2), 359-367.
2. Bogdanov, A. (2010). *Analysis and design of block cipher constructions*. Europäischer Univ.-Verlag.
3. García Méndez, P. (2011*). Descripción Polinomial de los Sistemas de Cifrado DES y AES*. Universidad Autónoma Mexicana, México.
4. Cardell, S., Fúster Sabater, A., & Requena, V. (2022). *PN-secuencias entrelazadas de polinomios diferentes*.
5. Masoodi, F., Alam, S., & Bokhari, M. U. (2012). *An analysis of linear feedback shift registers in stream ciphers.* International Journal of Computer Applications, 46(17), 46-49.
6. Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of applied cryptography*. CRC press.
7. Paar, C., & Pelzl, J. (2010). *Understanding cryptography (Vol. 1)*. Springer-Verlag Berlin Heidelberg.
8. Stahnke, W. (1973). *Primitive binary polynomials*. Mathematics of computation, 27(124), 977-980.
9. Fishman, G. S. (1990). *Multiplicative congruential random number generators with modulus* 2^{β} *: an exhaustive analysis for* $\beta = 32$ *and a partial analysis for* $\beta = 48$. Mathematics of Computation, 54(189), 331-344.
10. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., ... & Vo, S. (2001). *A statistical test suite for random and pseudorandom number generators for cryptographic applications* (Vol. 22, p. 1). Gaithersburg, MD, USA: US Department of Commerce, Technology Administration, National Institute of Standards and Technology.