Development of an improved image scrambling encryption algorithm

J.E. Treviño-Ortiz * **C. Posadas-Castillo** *
J.R. Rodríguez-Cruz * **E. Zambrano-Serrano** *
M.A. Platas-Garza *

* *Universidad Autónoma de Nuevo León,*
Facultad de Ingeniería Mecánica y Eléctrica,
Av. Universidad s/n. Cd. Universitaria, San Nicolás de los Garza,
Nuevo León, México, C.P. 66450

Abstract: The fractional-order systems with chaotic behavior have been recently proposed for the image encryption process because of their properties, such as ergodicity, pseudorandomness, sensitivity to initial conditions and control parameters. Furthermore, these systems have introduced an extra security parameter since they are allowed to increase the key space of an encryption scheme. This paper presents an image encryption algorithm based on a fractional-order chaotic system using bit-level permutation and a diffusion operation. The Grünwald-Letnikov definition is used to obtain the numerical solution of the fractional-order system, the Lyapunov exponents, phase plane and bifurcation diagram are also calculated. Simulations results and performance analysis show that the proposed algorithm has a high plaintext sensitivity advantage.

Keywords: Encryption, Dynamical Systems, Chaos, Fractional-order Systems, Cybersecurity.

1. INTRODUCCIÓN

El caos se caracteriza por una sensibilidad extrema a las condiciones iniciales, comportamiento aperiódico a largo plazo e irregularidad en la respuesta debido a la no linealidad en un sistema determinista y la presencia de al menos un exponente de Lyapunov no negativo como se menciona en Layek (2015). El caos ha sido ampliamente investigado por sus interesantes aplicaciones en generadores de números aleatorios, sincronización, procesamiento de imaǵenes, comunicación segura y redes neuronales artificiales como se muestra en Meranza-Castillón et al. (2019); Zambrano-Serrano et al. (2020); Yu et al. (2019). De acuerdo con Alvarez and Li (2006) existe una estrecha relación entre las propiedades del caos determinista y los requisitos para desarrollar un esquema de encriptado. Alvarez and Li (2006) menciona que cualquier esquema de encriptado debe implementar los procesos de confusión y difusión. En sistemas caóticos, esos procesos coinciden con la ergodicidad, una propiedad de mezcla y alta sensibilidad a las pequeñas variaciones de las condiciones iniciales o parámetros de control. Es por ello que, el comportamiento caótico puede ser considerado en los esquemas de encriptado. Por otra parte, el cálculo fraccionario es una generalización natural del cálculo entero al considerar derivadas e integrales de orden no entero. Específicamente, las derivadas fraccionarias son consideradas operadores no locales debido a que son definidos por una integral, la cual proporciona el efecto de memoria en aplicaciones temporales. La capacidad de describir las características hereditarias de un sistema, así como su memoria, son las ventajas más importantes del cálculo fraccionario sobre el de orden entero. Si el operador diferencial fraccionario se introduce en el sistema caótico, el sistema no solo puede producir nuevos comportamientos dinámicos en atractores, si no también comportamientos dinámicos más precisos y complejos como se observa en Zambrano-Serrano et al. (2018); García-Sepúlveda et al. (2020). Por lo tanto la motivación de este artículo es analizar la viabilidad de implementar sistemas de orden fraccionario en un algoritmo de encriptado.

En este artículo, principalmente abordamos el encriptado de imágenes empleando un oscilador caótico de orden fraccionario. Un algoritmo de encriptado fue presentado por Ye (2010), donde se plantea una permutación de los bits en los píxeles de una imagen de tamaño $M \times N$ a partir de dos vectores resultantes $\mathbf{T}_M$ y $\mathbf{T}_N$ de un sistema caótico. El algoritmo posee características interesantes, tales como, vectores de permutación de filas y columnas los cuales modifican la posición y el valor de los píxeles

⋆ J.E. Treviño-Ortiz, elizabeth.trevinortz@uanl.edu.mx, C. Posadas-Castillo, cornelio.posadascs@uanl.edu.mx, J.R. Rodríguez-Cruz, jose.rodriguezcu@uanl.edu.mx, E. Zambrano-Serrano, erneszambrano@gmail.com, M.A. Platas-Garza, miguel.platasgrz@uanl.edu.mx.

de forma simultánea; además, una nueva posición de los bits es definida mediante el uso de una señal caótica, por lo que cada pequeño cambio en la llave está generando posiciones permutadas diferentes. Un criptosistema basado en el comportamiento de un sistema caótico de orden fraccionario fue propuesto por Montero-Canela et al. (2020), demostrando que el tamaño de la llave puede ser seleccionado de forma arbitraria en $64N$-bits. En contraste Li and Lo (2011) mostró que el algoritmo propuesto por Ye (2010) presenta algunas debilidades de las cuales destacan, aleatoriedad débil de los vectores $\mathbf{T}_M$ y $\mathbf{T}_N$, una nula sensibilidad con respecto a cambios en el texto plano y baja eficiencia en el método de generación de los vectores de permutación, entre otros.

Tomando en cuenta los puntos mencionados previamente, se propone un algoritmo que mejore la seguridad del proceso de encriptado mostrado en Ye (2010), agregando una operación que aumente la sensibilidad en el texto plano, el uso de un sistema caótico de mayor orden y eficiencia en la generación de los vectores de permutación. Las operaciones digitales reversibles de permutación se pueden aplicar a los bits a nivel píxel o a los píxeles a nivel imagen, ayudando a garantizar la no distorsión del mensaje original. Asimismo, se añade una operación extra de difusión para evitar que la imagen original y la cifrada tengan el mismo número de unos y ceros, de esta forma se mejora la sensibilidad a cambios en el texto plano. El uso de operaciones digitales que agreguen información al mensaje original, por ejemplo datos de la señal caótica, incrementará la seguridad del proceso de encriptado. Con la finalidad de generar dinámicas más complejas, se toma la alternativa de emplear osciladores caóticos de orden fraccionario como se menciona en Li and Chen (2004). Además, es posible seleccionar condiciones iniciales y orden fraccionario dentro de un rango permitido de valores, garantizando un comportamiento caótico en el sistema dinámico y como consecuencia una sensibilidad extrema. Cabe mencionar, que el presente trabajo se enfoca en la metodología del proceso de encriptado, por lo que la extensión de la llave se considera como un punto de oportunidad para trabajo futuro.

El artículo se organiza como se detalla a continuación. La Sección 2 muestra los preliminares matemáticos. La Sección 3 detalla el algoritmo de encriptado propuesto. Los resultados de las pruebas de simulación y evaluación de índices de desempeño del algoritmo se muestran en la Sección 4. Finalmente cerramos el artículo con conclusiones mostradas en la Sección 5.

2. CÁLCULO FRACCIONARIO.

La teoría de cálculo fraccionario generaliza las operaciones de integración y diferenciación a órdenes no enteros Petráš (2011). El operador integro-diferencial fraccionario continuo que define tales operaciones está determinado por ${}_aD_t^q$ donde $a, t \in R$ son las cotas de la operación y $q \in R$ representa el orden. Con la finalidad de resolver el problema de integración fraccionaria de manera numérica, la definición de Grünwald-Letnikov (GL) para el operador ${}_aD_t^q$ puede ser usada para la aplicación del operador fraccionario a una secuencia $f(k)$ muestreada con un paso h. Lo anterior deriva en la siguiente relación

$$ {}_{(k-L_m/h)}D_{t_k}^q f(t) \approx h^{-q} \sum_{j=0}^{L_m-1} c_j^{(q)} f(k-j), \tag{1}$$

con $L_m \in R$ representando la cantidad de muestras considerada en la longitud de memoria, y $c_j^{(q)} \in R$ los coeficientes binomiales

$$ c_0^{(q)} = 1, \quad c_j^{(q)} = (1 - \frac{1+q}{j}) c_{j-1}^{(q)}. \tag{2}$$

Entonces, una aproximación a la solución numérica de la ecuación diferencial fraccionaria dada por

$$ {}_aD_t^q y(t) = f(y(t), t), \tag{3}$$

se puede expresar por

$$ y(t_k) = f(y(t_k), t_k) h^q - \sum_{j=1}^{L_m-1} c_j^{(q)} y(t_k - j). \tag{4}$$

Asumiendo que el tamaño del paso h es lo suficientemente pequeño, vemos que (1) puede usarse para evaluar las diferenciaciones de la función dada, además se ha demostrado que la precisión del método es $O(h)$.

2.1 Oscilador caótico fraccionario de Lorenz.

El modelo matemático para un oscilador de Lorenz de orden fraccionario está definido por

$$ \begin{cases} {}_0D_t^{q_1} x(t) = \sigma(y(t) - x(t)), \\ {}_0D_t^{q_2} y(t) = x(t)(\rho - z(t)) - y(t), \\ {}_0D_t^{q_3} z(t) = x(t)y(t) - \beta z(t). \end{cases} \tag{5}$$

Donde $x(t), y(t)$ y $z(t)$ son vectores de estado, (σ, ρ, β) son parámetros reales positivos, $0 < q_i < 1$ con $i = 1, 2, 3$ es el orden fraccionario. Los puntos de equilibrio del sistema (5) son calculados al resolver $f_i(x(t), y(t), z(t)) = 0$, donde se obtiene que el sistema tiene tres puntos de equilibrio de la siguiente forma $E_1 = (0, 0, 0)^T$, $E_2 = \left(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1\right)$, $E_3 = -\left(\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho - 1\right)$. El conjunto de ecuaciones (5) exhibe un comportamiento caótico para los parámetros $(\sigma, \rho, \beta) = (10, 28, 8/3)$ y un orden conmensurado mínimo $q_1 = q_2 = q_3 > 0.9941$. La Fig. 1 muestra el atractor extraño proyectado en el plano de fase considerando sus tres estados y la Fig. 2 los diagramas de bifurcación de los parámetros σ, ρ, β, y los exponentes de Lyapunov de dicho oscilador considerando un orden conmensurado $q = 0.995$. Los exponentes de Lyapunov para el sistema (5) son $\lambda_1 = 0.8260$, $\lambda_2 = 0$ y $\lambda_3 = -1.9202$ respectivamente.

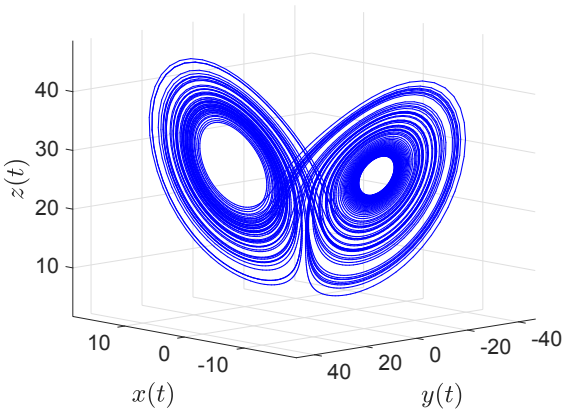


Figura 1. Atractor del oscilador de Lorenz fraccionario para parámetros $(\sigma, \rho, \beta) = (10, 28, 8/3)$, y condiciones iniciales $(x(0), y(0), z(0))$ =(0.65,0.4,0.7) .

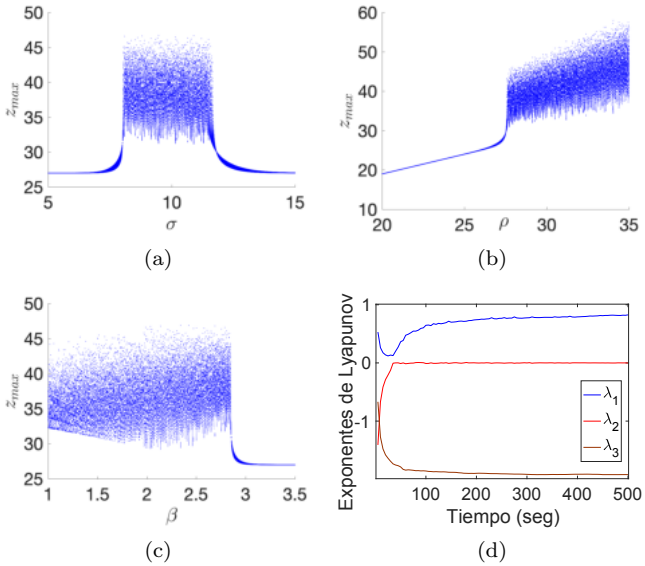


Figura 2. Dinámica del oscilador de Lorenz de orden fraccionario. (a) Diagrama de bifurcación para σ; (b) Diagrama de bifurcación para ρ; (c) Diagrama de bifurcación para β; (d) Exponentes de Lyapunov con $(\sigma, \rho, \beta) = (10, 28, 8/3)$, orden fraccionario $q = 0.995$, y condiciones iniciales $(0.1,0.1,0.1)^T$.

3. ALGORITMO IMAGE-SCRAMBLING PROPUESTO

Se considera el encriptado de una imagen en escala de grises de $M \times N$ píxeles. Para este caso, el algoritmo propuesto cuenta con las etapas descritas a continuación.

1. Se generan los vectores de permutación $\mathbf{T}_M \in \mathcal{Z}^{+M}$ y $\mathbf{T}_N \in \mathcal{Z}^{+N}$ cuantificando en 8 bits 512 muestras resultantes de la integración numérica de un oscilador caótico de orden fraccionario.
2. Se forma la matriz de datos $\mathbf{A} \in \mathcal{Z}^{+M\times N}$ del texto plano, los elementos provienen de una imagen en escala de grises por lo que tienen un valor entero entre 0 y 255.
3. Se aplican los vectores de permutación a la matriz $\mathbf{A}$, se modifica la posición de las filas de acuerdo a $\mathbf{T}_M$ y después la posición de las columnas de acuerdo a $\mathbf{T}_N$, obteniendo una matriz $\mathbf{B}$.
4. Para mejorar la sensibilidad a cambios en el texto plano, se calculan nuevos valores de píxel, formando la matriz $\mathbf{C}$ a partir de $\mathbf{B}$. La matriz $\mathbf{C}$ se forma usando

$$\begin{cases} c(r,c) = \phi(r,c) \oplus [b(r,c) + \phi(k)] \text{mod} 256 \oplus c(r-1,c), \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{para: } r \neq 1, \\ c(r,c) = \phi(r,c) \oplus [b(r,c) + \phi(k)] \text{mod} 256 \oplus c(M,c-1), \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{para: } r = 1, c \neq 1, \\ c(r,c) = \phi(r,c) \oplus [b(r,c) + \phi(k)] \text{mod} 256 \oplus b(1,1), \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{para: } r, c = 1, \end{cases} \tag{6}$$

donde $c(r,c)$ es el valor de píxel encriptado correspondiente al r-ésimo elemento de la c-ésima columna de $\mathbf{C}$, $\phi(r,c)$ corresponde al elemento (r,c) de una matriz $\mathbf{\Phi}$ en la que se cuantiza a 8 bits y almacena el estado $x(t)$ del sistema caótico, $b(r,c)$ es el r-ésimo píxel de la c-ésima columna de la matriz $\mathbf{B}$ obtenida en el paso 3, y $b(1,1)$ se usa como semilla. Esta operación forma parte de un algoritmo de encriptado descrito en Koduru and Chandrasekaran (2008).

5. La matriz $\mathbf{C}$ se divide en 8 matrices $\mathbf{C}_i \in \mathcal{Z}^{+M\times\frac{N}{8}}$ para $i = 1, 2, \ldots, 8$.
6. Los píxeles de cada matriz $\mathbf{C}_i$ se transforman a numeración binaria y cada bit ocupa una nueva posición de columna, formando así 8 matrices con elementos booleanos $\mathbf{D}_i \in \mathcal{F}_2^{M\times N}$. Este proceso se ejemplifica en la Fig. 3.
7. Se aplican los vectores de permutación $\mathbf{T}_M$ y $\mathbf{T}_N$ obtenidos de la señal caótica a cada matriz $\mathbf{D}_i$, se obtienen 8 matrices $\mathbf{E}_i \in \mathcal{F}_2^{M\times N}$.
8. Cada $\mathbf{E}_i$ se convierte a información numérica mediante la operación inversa a la descrita en el paso 6 para obtener de nuevo 8 matrices $\mathbf{F}_i \in \mathcal{Z}^{+M\times\frac{N}{8}}$
9. Se forma la matriz $\mathbf{G} \in \mathcal{Z}^{+M\times N}$, uniendo los bloques de matrices $\mathbf{F}_i$.
10. $\mathbf{G}$ es la imagen encriptada a la salida del esquema de encriptado.

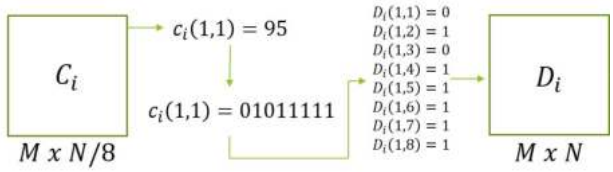


Figura 3. Expansión de elementos decimales de matriz $\mathbf{C_i}$ a matriz binaria $\mathbf{D_i}$.

3.1 Generación de vectores de permutación

En Wang et al. (2018) se muestra un pseudocódigo para la generación de una S-Box o caja de sustitución. Siguiendo esta idea, se generan los vectores $\mathbf{T}_M$ y $\mathbf{T}_N$. A diferencia de Ye (2010), para este algoritmo se emplea

un sistema caótico de mayor orden, y se define una nueva manera de generar los vectores de permutación.

Primero, se emplea el método Grünwald-Letnikov al sistema, obteniendo como resultado secuencias de estado $x(k)$ y $y(k)$. Posteriormente, $x(k)$ y $y(k)$ se procesan de acuerdo a

$$n_1(k) = (\text{fix}(\text{rem}(x(k), 1) * 1000))\text{mod}512, \tag{7}$$

y

$$n_2(k) = (\text{fix}(\text{rem}(y(k), 1) * 1000))\text{mod}512, \tag{8}$$

donde mod, fix y rem representan las funciones módulo, redondeo y residuo respectivamente. Finalmente, los pasos posteriores para la generación de los vectores de permutación obtienen $\mathbf{T}_M$ y $\mathbf{T}_N$ a partir de $n_1(k)$ y $n_2(k)$, de acuerdo al pseudocódigo desarrollado en Wang et al. (2018), mismo que es mostrado en la tabla 3.1.

Psuedocódigo de algoritmo para generación de vectores de permutación.

1. Inicio;
2. i=1; j=1; k=1;
3. mientras (i < 513) hacer;
4. si (Se encuentra $n_1(k)$ en vector T_M = sí) entonces;
5. Ir a paso 10;
6. sino (Se encuentra $n_1(k)$ en vector T_M = no);
7. $T_M[i] = n_1(k)$;
8. i=i+1;
9. terminar si;
10. k=k+1;
11. terminar mientras;
12. k=1; 13. mientras (j < 513) hacer;
14. si (Se encuentra $n_2(k)$ en vector T_N = sí) entonces;
15. Ir a paso 20;
16. sino (Se encuentra $n_2(k)$ en vector T_N = no);
17. $T_N[j] = n_2(k)$;
18. j=j+1;
19. terminar si;
20. k=k+1;
22. terminar mientras;
23. Fin;

3.2 Obtención de señal caótica

La aportación principal de este trabajo radica en la metodología usada para encriptar y no en la selección de la llave. Sin embargo, la llave considerada para este algoritmo contiene la información requerida de las condiciones iniciales y orden fraccionario para el sistema caótico, estas son $[x(0), y(0), z(0), q_1, q_2, q_3]$ y los valores se manejan cuantizando en 16 bits rangos acotados. Para este estudio los juegos de condiciones iniciales y órdenes del sistema se evaluaron de manera individual con la finalidad de garantizar que las señales obtenidas muestren un comportamiento caótico. En todos los casos considerados, al implementar el método de integración se usa una longitud de memoria de $L_m = 100$ muestras y un paso de integración de $h = 0.005$ segundos.

3.3 Proceso de desencriptado de la imagen

Todas las operaciones aplicadas en el proceso de encriptado de la imagen son operaciones reversibles. Los pasos para realizar el proceso de desencriptado son los siguientes:

1. Se generan los vectores de permutación $\mathbf{T}_M \in \mathcal{Z}^{+M}$ y $\mathbf{T}_N \in \mathcal{Z}^{+N}$ tal y como en el proceso de encriptado.
2. Se forma la matriz $\mathbf{A} \in \mathcal{Z}^{+M\times N}$ de los datos provenientes de la imagen encriptada.
3. La matriz $\mathbf{A}$ se divide en 8 matrices $\mathbf{A}_i \in \mathcal{Z}^{+M\times \frac{N}{8}}$ para $i = 1, 2, \ldots, 8$.
4. Los píxeles de cada matriz $\mathbf{A}_i$ se transforman a numeración binaria y cada bit ocupa una nueva posición de columna, formando así 8 matrices con elementos booleanos $\mathbf{B}_i \in \mathcal{F}_2^{M\times N}$.
5. Se aplican los vectores de permutación $\mathbf{T}_M$ y $\mathbf{T}_N$ de forma inversa, obtenidos de la señal caótica a cada matriz $\mathbf{B}_i$, se obtienen 8 matrices $\mathbf{C}_i \in \mathcal{F}_2^{M\times N}$.
6. Cada $\mathbf{C}_i$ se convierte a información numérica mediante la operación inversa a la descrita en el paso 4 para obtener de nuevo 8 matrices $\mathbf{D}_i \in \mathcal{Z}^{+M\times \frac{N}{8}}$.
8. Se calculan los valores de píxel de acuerdo a la ecuación

$$\begin{cases} f(r,c) = [\phi(r,c) \oplus d(r-1,c) \oplus d(r,c) + 255 - \phi(k)]\text{mod}256 \\ \qquad\qquad \text{para: } r \neq 1, \\ f(r,c) = [\phi(r,c) \oplus d(M,c-1) \oplus d(r,c) + 255 - \phi(k)]\text{mod}256 \\ \qquad\qquad \text{para: } r = 1, c \neq 1, \\ f(r,c) = [\phi(r,c) \oplus d(1,1) \oplus d(r,c) + 255 - \phi(k)]\text{mod}256 \\ \qquad\qquad \text{para: } r, c = 1, \end{cases} \tag{9}$$

se forma la matriz $\mathbf{F}$, esta operación es inversa a la mostrada en el paso 4 del encriptado.
9. Se aplican los vectores de permutación de manera inversa a la matriz $\mathbf{F}$ obtenida, formando la matriz $\mathbf{G}$.
10. Se obtiene la imagen recuperada.

4. ANÁLISIS DE DESEMPEÑO DEL ALGORITMO DE ENCRIPTADO

Las métricas que se consideran para la evaluación de desempeño del algotimo son: entropía (H), coeficiente de correlación de Pearson (PCC), intensidad de cambio promedio unificado (UACI) y razón de cambio de píxeles (NPCR). Se utiliza la imagen de Lena en tamaño 512×512 píxeles en escala de grises con 8 bits de profundidad.

4.1 Entropía y coeficiente de correlación

La entropía ayuda a conocer el nivel de desorden, determina la aleatoriedad y complejidad de la información Murillo-Escobar et al. (2019). Para una imagen en escala de grises, el valor de sus elementos es representado por 8 bits por lo tanto la entropía máxima es 8. En la Tabla 1 se muestran los valores mínimo, máximo y promedio de la entropía de una imagen cifrada con el algoritmo

image-scrambling y el propuesto, se consideraron 15 llaves y diferentes cantidades de rondas k.

	k	mín(H)	máx(H)	prom(H)
Image-scrambling	1	7.9923	7.9937	7.9932
	2	7.9925	7.994	7.9934
	4	7.9919	7.9939	7.9932
Algoritmo propuesto	1	7.9992	7.9994	7.9993
	2	7.9992	7.9994	7.9993
	4	7.9992	7.9994	7.9993

Tabla 1. Valores mínimo, máximo y promedio de entropía (H), obtenidos de 15 experimentos con llaves diferentes.

El PCC sirve para determinar el grado de relación lineal entre la imagen original y la cifrada. El PCC obtenido puede tener un valor entre [-1,1], donde 1 muestra una relación directa perfecta, -1 una relación inversa perfecta y 0 una correlación nula entre ambas imágenes. En la Tabla 2 se muestran los valores mínimo, máximo y promedio de PCC para diferentes cantidades de rondas k, con el algoritmo image-scrambling y el propuesto.

	k	mín(H)	máx(H)	prom(H)
Image-scrambling	1	0.0132	0.114	0.0631
	2	0.0122	0.0944	0.0842
	4	0.0059	0.1233	0.0503
Algoritmo propuesto	1	0.0006	0.0298	0.0016
	2	0.0008	0.0224	0.0034
	4	-0.0019	-0.0222	-0.0043

Tabla 2. Valores mínimo, máximo y promedio de PCC, obtenidos de 15 experimentos con llaves diferentes.

4.2 Sensibilidad de texto plano y de la llave

Evaluar la sensibilidad del texto plano es importante para determinar la robustez del encriptado contra ataques diferenciales. Los índices de desempeño que se usan en esta sección son UACI y NPCR, ambos muestran en porcentaje el efecto causado por un pequeño cambio en el texto plano. Para imágenes de tamaño 512 x 512 el valor crítico mostrado en Murillo-Escobar et al. (2019) de NPCR es 99.5717 % y para UACI se tiene el intervalo de aceptación 33.3115 %–33.6156 %. Los valores mínimo, máximo y promedio de NPCR y UACI obtenidos de los experimentos se muestran en la Tabla 3.

La sensibilidad en la llave es estudiada bajo los mismos índices, encriptando la misma imagen con dos llaves que tienen un bit diferente entre ellas. Los valores mínimo, máximo y promedio de NPCR y UACI obtenidos de los experimentos se muestran en la Tabla 4.

4.3 Tiempo de encriptado

Una comparación de los tiempos de encriptado de ambos algoritmos fue realizada. Los valores mínimo, máximo y promedio obtenidos de los experimentos se muestran en la Tabla 5. Ambos esquemas fueron implementados en un sistema operativo Windows 10, con procesador AMD FX-7500 R7 a 2.1 GHz, 8 GB de RAM y 885 GB en disco duro.

		k	mín	máx	prom
(a)	NPCR	1	0.0007	0.0023	0.0015
		2	0.0003	0.0023	0.0014
		4	0.0003	0.0027	0.0017
(a)	UACI	1	0.00001	0.00040	0.00017
		2	0.000001	0.00048	0.00012
		4	0.000001	0.00036	0.00018
(b)	NPCR	1	10.5469	74.7463	41.4842
		2	96.8208	99.7597	98.9990
		4	99.5752	99.6307	99.6077
(b)	UACI	1	2.0286	28.6621	12.7476
		2	26.4423	34.1244	31.4647
		4	33.3355	33.6047	33.4358

Tabla 3. Valores mínimo, máximo y promedio de NPCR y UACI , obtenidos de 15 experimentos con la misma imagen. En cada ronda el valor de un píxel aleatorio fue modificado, (a) algoritmo image-scrambling, (b) algoritmo propuesto.

		k	mín	máx	prom
(a)	NPCR	1	99.5758	99.6159	99.5976
		2	99.5995	99.6456	99.6159
		4	99.5850	99.6262	99.6027
(a)	UACI	1	33.3801	33.6610	33.5251
		2	33.2689	33.7730	33.5358
		4	33.3737	33.5897	33.4671
(b)	NPCR	1	99.5846	99.6387	99.6109
		2	99.5818	99.6311	99.6096
		4	99.5918	99.6307	99.6074
(b)	UACI	1	33.3625	33.5707	33.4598
		2	33.3636	33.5562	33.4626
		4	33.3721	33.5312	33.4642

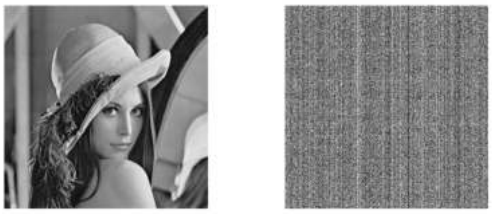
Tabla 4. Valores mínimo, máximo y promedio de NPCR y UACI , obtenidos de 15 experimentos con diferentes llaves. En cada ronda el valor de un bit aleatorio de la llave fue modificado, (a) algoritmo image-scrambling, (b) algoritmo propuesto.

	k	mín	máx	prom
Image-scrambling	1	45.0431	47.1990	46.1399
	2	88.9507	95.2701	90.2919
	4	172.7928	184.9595	175.4669
Algoritmo propuesto	1	42.6249	45.4413	44.2244
	2	85.7085	91.9786	88.2037
	4	170.9736	175.0174	172.6514

Tabla 5. Valores mínimo, máximo y promedio de tiempos de encriptado en segundos, obtenidos de 30 experimentos con llaves diferentes.

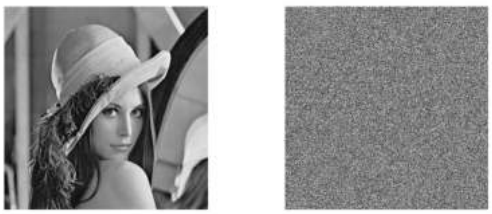
4.4 Imagen encriptada

Se considera la imagen Lena en el presente trabajo. En la Fig. 4 se muestra la imagen original y la encriptada con 4 rondas del algoritmo image-scrambling; en la Fig. 5 se muestra la imagen original y la encriptada con 4 rondas del algoritmo propuesto. Es posible observar en las imagenes resultantes del algoritmo image-scrambling cierto



(a) (b)

Figura 4. Lena (a) original y (b) encriptada con 4 rondas del algoritmo image-scrambling.



(a) (b)

Figura 5. Lena (a) original y (b) encriptada con 4 rondas del algoritmo propuesto.

patrón de color en algunas filas y columnas de la imagen, mientras que en las imágenes cifradas por el algoritmo propuesto se observa una aleatoriedad uniforme.

5. CONCLUSIÓN

El algoritmo de encriptado propuesto muestra mejoría en todos los índices de desempeño evaluados. Se logró aumentar notablemente la sensibilidad de texto plano con la operación añadida donde se emplean elementos de la señal caótica para modificar el valor de los píxeles y así afectar la información a nivel bit de la imagen. El tiempo de proceso de encriptado se mantuvo mientras se mejoró el nivel de seguridad del algoritmo. Se considera como trabajo a futuro aumentar el tamaño de la llave cuantizando las condiciones iniciales y órdenes del sistema a una mayor cantidad de bits; además, evaluar y determinar rangos de valores para las condiciones iniciales donde se garantiza comportamiento caótico. También se propone realizar una evaluación de robustez del algoritmo ante la existencia de ruido en el proceso de transmisión de la imagen.

6. AGRADECIMIENTOS

J.E. Treviño-Ortiz agradece a CONACyT por el apoyo recibido a través de la beca de maestría. Todos los autores agradecen a CONACyT por el apoyo recibido a través del proyecto ref. 166654 y A1-S-31628 y a la FIME-UANL.

REFERENCIAS

Alvarez, G. and Li, S. (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *International journal of bifurcation and chaos*, 16(08), 2129–2151.

García-Sepúlveda, O., Posadas-Castillo, C., Cortés-Preciado, A., Platas-Garza, M., Garza-González, E., and Sanchez, A.G. (2020). Synchronization of fractional-order lü chaotic oscillators for voice encryption. *Revista Mexicana de Física*, 66(3 May-Jun), 364–371.

Koduru, S.C. and Chandrasekaran, V. (2008). Integrated confusion-diffusion mechanisms for chaos based image encryption. In *2008 IEEE 8th International Conference on Computer and Information Technology Workshops*, 260–263. IEEE.

Layek, G. (2015). *An introduction to dynamical systems and chaos*. Springer.

Li, C. and Lo, K.T. (2011). Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal processing*, 91(4), 949–954.

Li, C. and Chen, G. (2004). Chaos and hyperchaos in the fractional-order rössler equations. *Physica A: Statistical Mechanics and its Applications*, 341, 55–61.

Meranza-Castillón, M., Murillo-Escobar, M., López-Gutiérrez, R., and Cruz-Hernández, C. (2019). Pseudorandom number generator based on enhanced hénon map and its implementation. *AEU-International Journal of Electronics and Communications*, 107, 239–251.

Montero-Canela, R., Zambrano-Serrano, E., Tamariz-Flores, E.I., Muñoz-Pacheco, J.M., and Torrealba-Meléndez, R. (2020). Fractional chaos based-cryptosystem for generating encryption keys in ad hoc networks. *Ad Hoc Networks*, 97, 102005.

Murillo-Escobar, M.A., Meranza-Castillón, M.O., López-Gutiérrez, R.M., and Cruz-Hernández, C. (2019). Suggested integral analysis for chaos-based image cryptosystems. *Entropy*, 21(8), 815.

Petráš, I. (2011). *Fractional-order nonlinear systems: modeling, analysis and simulation*. Springer Science & Business Media.

Wang, X., Akgul, A., Cavusoglu, U., Pham, V.T., Vo Hoang, D., and Nguyen, X.Q. (2018). A chaotic system with infinite equilibria and its s-box constructing application. *Applied Sciences*, 8(11), 2132.

Ye, G. (2010). Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*, 31(5), 347–354.

Yu, W., Wang, J., Wang, J., Zhu, H., Li, M., Li, Y., and Jiang, D. (2019). Design of a new seven-dimensional hyperchaotic circuit and its application in secure communication. *IEEE Access*, 7, 125586–125608.

Zambrano-Serrano, E., Munoz-Pacheco, J., Gomez-Pavon, L., Luis-Ramos, A., and Chen, G. (2018). Synchronization in a fractional-order model of pancreatic β-cells. *The European Physical Journal Special Topics*, 227(7-9), 907–919.

Zambrano-Serrano, E., Posadas-Castillo, C., and Rivera-Durón, R.R. (2020). Diferentes familias de atractores extraños en un sistema caótico memristivo de 4d. *Ingenierías*, 23(86), 13.

