UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

Proyecto de Tesis 1

Encriptación de Imágenes basado en los Atractores Caóticos de Lorenz y Rössler

Autor: Cristopher Sebastián García Pacheco

Asesor: Yuri Nuñez Medrano

Lima-Perú, 2021

Resumen

El área de investigación de la seguridad informática ha sido beneficiada en los últimos años gracias a los avances computacionales. Por esta razón, se puede ejecutar algoritmos de encriptación complejos; pero también, eficientes algoritmos de ataque de fuerza bruta. Para fines de esta investigación, nos centramos específicamente en la encriptación de imágenes a color, aprovechando las propiedades de los Atractores Caóticos. Esta tesis explora los alcances producidos al encriptar imágenes basados en las propiedades de los Atractores Caóticos de Lorenz y Rössler. Esta implementación posee un desempeño favorable basado en las métricas propuestas y comparándolo con otros algoritmos de encriptación.

Índice general

Índice de Figuras

Índice de Acrónimos

SI	Seguridad Informática
SCL	Sistema Caótico de Lorenz
SCR	Sistema Caótico de Rössler
CPU	Central Processing Unit
GPU	Graphic Processing Unit
ETC	Etcétera

Agradezco

A mi familia, quienes se esforzaron por ayudarme a ser quien soy.

A mis amigos, por disfrutar conmigo los momentos de nuestra formación académica.

A mi asesor, por mostrarme el camino en el proceso de la realización de la tesis.

Capítulo 1

Introducción

Cuando se habla de encriptación, se busca trabajar en el fundamento de confidencialidad desde el enfoque de seguridad informática.
En esta tesis, se propone distorsionar la imagen de tal forma que no se pueda reconocer la imagen original a simple vista, y tras aplicar el algoritmo de descifrado con las claves correctas, podemos obtener la imagen original. Existen muchas formas de encriptación de imágenes, en esta tesis se propone aplicar las propiedades del Sistema Caótico de Lorenz (SCL) y el Sistema Caótico de Rössler para encriptar una imagen utilizando un algoritmo de permutación de filas y columnas.

De este modo, el objetivo de encriptar la imagen surge de la necesidad de ocultar infomación o datos sensibles como imágenes médicas de radiografías de pacientes con cáncer o imágenes satelitales de una base de datos militar, por lo que el algoritmo de encriptación debe ser fuerte en el sentido de que sea muy difícil romper la encriptación por ataque de fuerza bruta.

1.1. Motivación

La seguridad informática es una amplia área de investigación impulsada muchas veces por la curiosidad de aprender a asegurar tus datos, de sentirte seguro frente a los peligros que amenazan tus datos sensibles o de pensar cómo un delincuente cibernético puede acceder a datos a los que no deberia tener

acceso, y de este modo, ver cómo prevenir ese tipo de situaciones.

Desde esta perspectiva, este trabajo está fomentado por el deseo de asegurar datos sensibles (en este caso imágenes), además de entender los algoritmos de encriptación basados en sistemas caóticos y sus propiedades.

Dentro de la seguridad informática, los procesos de encriptación son un buen problema introductorio al área de investigación, con lo cual, se puede empezar con algoritmos simples creados por el investigador, hasta basarse en sistemas matemáticos más complejos. Además, la encriptación es una de las tantas capas de seguridad que se le puede aplicar a los datos, con lo cual, abre paso a otros campos de la seguridad informática. Hoy en día es mas fácil adentrarse en esta área gracias a la documentación y contenido académico capaz de abordar muchos campos.

1.2. Objetivos

El objetivo principal de esta tesis es el de encriptar imágenes, utilizando las propiedades del Sistema Caótico de Lorenz (SCL).

Específicamente, los objetivos de este trabajo con respecto al sistema son:

- Entender las propiedades de un sistema caótico para encriptación.
- Implementar un algoritmo de encriptación que cumpla con los estándares que otros algoritmos de encriptación tienen.
- Evaluar el nivel de seguridad que puede dar este algoritmo y compararlo con otros alternativos.

Y los objetivos con respecto a las competencias académicas desplegadas en el trabajo son:

- Desarrollar métodos originales de encriptación sobre imágenes a color.
- Obtener el conocimiento necesario para implementar algoritmos de encriptación en este y otros proyectos de seguridad informática, empleando las herramientas de programación disponibles.

1.3. Estructura de la Tesis

- **Introducción:**
  En este capítulo se introduce al lector en el tema a tratar, explicando las motivaciones y objetivos del proyecto con intención de sembrar interés por el tema.
- **Estado del Arte:**
  Esta sección expondrá el fundamento teórico sobre el que se soporta la presente tesis, además de trabajos e investigaciones ya realizados alusivos al problema planteado.
- **Recursos y Herramientas:**
  Aquí se detallan las especificaciones del hardware y software empleados en la investigación. Además, se describe las especificaciones de las imágenes a tratar.
- **Estructuración y Método:**
  En este capítulo se explicará el procedimiento central de la investigación, además de la estructura del sistema implementado y su funcionamiento aplicando el algoritmo.
- **Conclusiones y Trabajo a Futuro:**
  Finalmente, se discutirán los resultados obtenidos, teorizando posteriormente cómo se podría extender el trabajo realizado mejorando sus resultados.

Capítulo 2

Estado del Arte

En el presente capítulo se explicará a detalle el fundamento teórico de los tópicos involucrados en el desarrollo de la investigación, los trabajos previos por los que fueron influenciados, y la diferencia entre estos y la metodología presentada en esta tesis.

2.1. Trasfondo Teórico

En la presente sección se revisarán los conceptos asimilados necesarios para el desarrollo del tema. Se tomará como punto de partida la unidad nuclear de un sistema de información, y se continuará particularizando los tópicos hasta cubrir todo lo requerido para el entendimiento de la presente investigación.

2.1.1. Introducción a la Seguridad Informática

La seguridad informática es un área de investigación propia de la ciencia de la computación, que estudia las medidas y controles que aseguran sus tres pilares: Confidencialidad, Integridad y Disponibilidad de los sistemas de información, software, hardware e información que está siendo procesada, en reposo o en comunicación. A continuación, veremos los 3 pilares de la seguridad informática o el *CIA Triad* según William Stallings [36]:

Confidencialidad

La protección de confidencialidad consiste en preservar las restricciones autorizadas y cumplir ciertos accesos de nivel a la información. Asegura que la información confidencial no estará disponible o será divulgada a personas no autorizadas. Una pérdida de confidencialidad es la divulgación no autorizada de información.

Integridad

La protección de integridad consiste en prevenir la modificación o destrucción parcial o completa de la información por agentes no autorizados. Una pérdida de integridad es la modificación o destrucción no autorizada de la información.

Disponibilidad

La protección de disponibilidad consiste en asegurar el acceso confiable de la información a los agentes autorizados. Una pérdida de disponibilidad es la interrupción del acceso o el uso de información.

2.1.2. Encriptación

El propósito de la encriptación de datos es la de proteger la confidencialidad de datos digitales en un sistema informático; estos datos pueden estar en reposo, en transmisión o estar siendo procesados [26].

2.1.3. Digital Image

Es una imagen representada como un ordenamiento de números, más especificamente, una matriz de números. Su unidad más pequeña son los píxeles. En una imagen a escala de grises, estos números varían de 0 a 255 que va desde negro a blanco, se trata de una matriz bidimensional. Para imágenes a color, cada pixel tiene 3 valores de 0 a 255, cada valor para la escala RGB, se

trata de una matriz tridimensional [31].
A la manipulación de estas imágenes digitales por medio de una computadora se le denomina: **Ïmage Processing"**, esto sirve para muchos campos como la medicina, los videojuegos, la ingeniería, etc. Estas imágenes puedes ser procesadas por medio de algoritmos para encriptarlas, analizarlas, comprimirlas o restaurarlas [8].

2.1.4. Asegurar Imágenes Digitales

Se pueden asegurar las imágenes digitales por medio de distintas formas como esteganografía [22, 37, 42], image watermarking [14, 13, 12] o encriptación [38, 24, 25, 32]. La encriptación es método más sencillo y eficiente para asegurar imágenes y convertirlas en imágenes con una clave secreta [33].

Esteganografía

La esteganografía es la práctica de ocultar un mensaje secreto dentro de (o incluso encima de) algo que no es secreto como un archivo u otra imagen que si es pública [6]. Su uso puede combinarse con la encriptación para añadir una capa más de seguridad [35].

Image Watermarking

Image Watermarking es un texto o logotipo añadida en una imagen para indicar el dueños o los derechos de autor de dicha imagen. Esto hace que sea difícil para alguien usar la imagen sin permiso o reclamar la propiedad del original [3].

Encriptación de imágenes

Con el aumento de la comunicación digital y el uso de imágenes, la encriptación de imágenes es un método que sirve para cominicar imágenes confidenciales, este método puede combinarse con muchos otros métodos y sistemas matemáticos [1]. En [2] se hace una comparación entre la encriptación

de imágenes y esteganografía, y demuestra que se debe preferir la encriptación de imágenes por más que es un método bastante antiguo, lo que lo hace novedoso son los métodos con los que se puede combinar.

2.1.5. Métricas

Entropía

La aleatoriedad de una imagen es medida por la información de la entropía. La definición matemática de la entropía está dada por [21]:

$$H(x) = -\sum_{i=1}^{n} p_i log_2(p_i) \tag{2.1}$$

Donde H(x) es la probabilidad de la aparición de n, para imágenes a escala de grises, el valor máximo de la entropía es 8. Cuando el valor de la entropía es cercano a 8, la aleatoriedad de pixeles en la imagen es alta [33]. Siendo más precisos, H (X) es la cantidad total de información en una distribución de probabilidad completa, según la ecuación de Entropía de Shannon [19]. En este experimento, encriptamos imágenes a color, con lo cual, el cálculo sería distinto. Podemos aplicar la fórmula directamente o, transformar la imagen a escala de grises y aplicar el método anteriormente expuesto .

Coeficiente de Correlación de Pearson

Los píxeles adyacentes muestran una alta correlación cuando sus valores son cercanamente idénticos. La eficiencia del algoritmo de encriptación está basada en generar imágenes encriptadas con una baja correlación entre los pixeles adyacentes [33]. Matemáticamente, el coeficiente de correlación de Pearson entre 2 píxeles adyacentes se define como [29]:

$$r = \frac{\sum_i (x_i - x_m)(y_i - y_m)}{\sqrt{\sum_i (x_i - x_m)^2}\sqrt{\sum_i (y_i - y_m)^2}} \tag{2.2}$$

Donde x_i es la intensidad del píxel i^{th} en la imagen 1, y_i es la intensidad del píxel i^{th} en la imagen 2, x_m es la intensidad media de la imagen 1 e y_m es la intensidad media de la imagen 2.

2.1.6. Teoría del Caos

La Teoria del Caos está muy relacionada con los eventos impredecibles o comportamiento aleatorio [17]. Podemos pensar en aleatoriedad comoel movimiento de las moléculas de gas. El caos está muy bien ilustrado por el efecto mariposa de Lorenz, que sugiere que el mero aleteo de la ala de una mariposa puede cambiar el clima. Otro ejemplo, puede ser la trayectoria de una bola de pinball, que está regida por las leyes gravitacionales y elásticas, pero el resultado final es impredecible [9].

2.1.7. Atractor

Antes de citar la definición de un Atractor, definimos primero la cuenca de atracción. Es el conjunto de puntos en el espacio de un sistema de variables, de modo que las condiciones iniciales elegidas en este conjunto evolucionen dinámicamente a un atractor en particular [41].
Según [40]: "Es un conjunto de estados (puntos en el espacio de fase), invariante bajo la dinámica, hacia los que los estados vecinos en una *cuenca de atracción* dada, se enfocan asintóticamente en el curso de evolución dinámica. Se define como la unidad más pequeña que no puede ser descompuesta por si misma en dos o más atractores con distintas *cuenca de atracción*"

2.1.8. Atractor Caótico de Lorenz

Es un sistema de ecuaciones diferenciales estudiado por primera vez por Edward Lorenz en 1963. El sistema tiene una gran cantidad de aplicaciones incluidas la convección atmosférica [18], biología y modelización de poblaciones integradas, astrodinámica moderna, dimensiones fractales en

dinámica [23]. El sistema de ecuaciones diferenciales del Sistema de Lorenz tiene la forma:

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = x(c - x) - y \\ \frac{dz}{dt} = -bz + xy \end{cases} \tag{2.3}$$

Donde $a = 10$, $b = 8/3$ y $c = 28$ para que el sistema de ecuaciones presente un comportamiento caótico, aunque puede tener distintas formas variando estos parámetros, como se ve en [16].

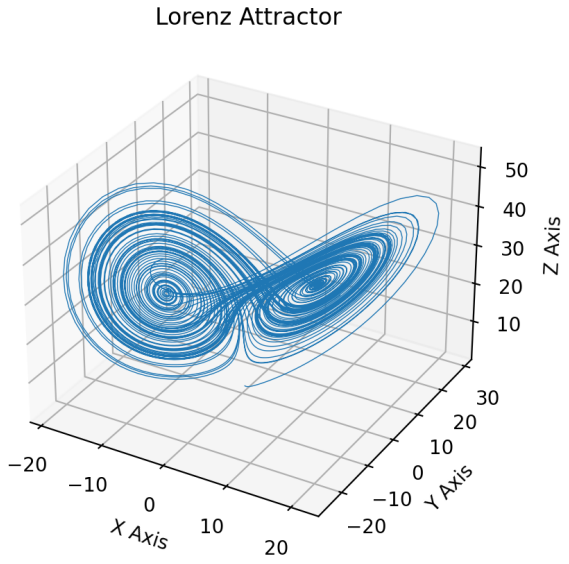


FIGURA 2.1: Sistema Caótico de Lorenz con $a = 10$, $b = 8/3$ y $c = 28$.

Utilizamos este Sistema Caótico de Lorenz por sus propiedades caóticas como imprevisibilidad y sensibilidad a las condiciones iniciales [32].

2.1.9. Atractor Caótico de Rössler

Un atractor de Rössler es un sistema de tres ecuaciones diferenciales ordinarias no lineales. El atractor de Rössler es de naturaleza similar al atractor de Lorenz [7].
El sistema de ecuaciones diferenciales del Sistema de Rossler tiene la forma:

$$\begin{cases} \frac{dx}{dt} = -(y+z) \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + xz - cz \end{cases} \tag{2.4}$$

Donde $a = 0.2$, $b = 0.2$ y $c = 8.0$ para que el sistema de ecuaciones presente un comportamiento caótico [28], aunque puede tener distintas formas variando estos parámetros, como se ve en [39].

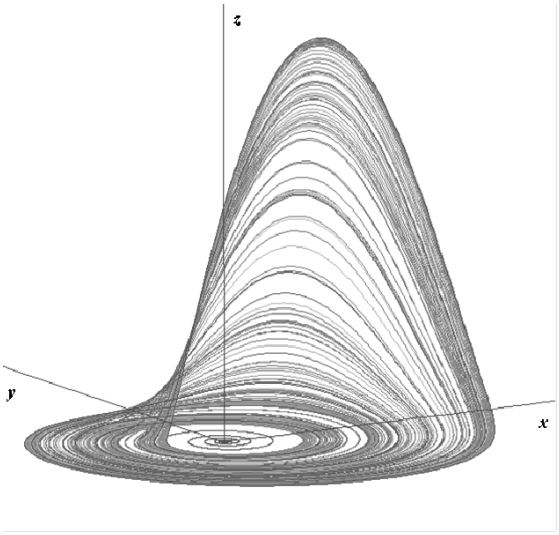


FIGURA 2.2: Sistema Caótico de Rössler con $a = 0.2$, $b = 0.2$ y $c = 8.0$ [27].

2.1.10. Runge Kutta 4

El método Runge-Kutta de orden 4 es la forma de los métodos de Runge-Kutta de uso mas común y así mismo más exactos para obtener soluciones aproximadas de ecuaciones diferenciales. La solución que ofrece este método, es una tabla de la función solución, con valores de

"y" correspondientes a valores específicos de "x" [4].

El algoritmo de Runge Kutta 4 se describe como:

RK4(a, b, N, α)
h $\leftarrow (b-a)/N$
$t_0 \leftarrow a$
$y_0 \leftarrow \alpha$
Para i desde 0 hasta N-1 hacer
$t_i \leftarrow a + i * h$
$k_1 \leftarrow h\mathrm{f}(\mathrm{t_i},\ \mathrm{y_i})$
$k_2 \leftarrow h\mathrm{f}\left(\mathrm{t_i} + \frac{1}{2}\mathrm{h},\ \mathrm{y_i} + \frac{1}{2}k_1\right)$
$k_3 \leftarrow h\mathrm{f}\left(\mathrm{t_i} + \frac{1}{2}\mathrm{h}, \mathrm{y_i} + \frac{1}{2}k_2\right)$
$k_4 \leftarrow h\mathrm{f}(\mathrm{t_i} + \mathrm{h},\ \ \mathrm{y_i} + k_3)$
$\mathrm{y_{i+1}} \leftarrow y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
Fin Para
Mostrar $(t_0, y_0),\ (t_1, y_1), \ldots\ (t_N, y_N)$
FIN

FIGURA 2.3: Algoritmo de Runge Kutta 4 [30].

El método se describe con las siguientes ecuaciones [11]:

$$\begin{cases} y_0 = y_a \\ y_{k+1} = y_k + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), k = 0, 1, \ldots \end{cases} \tag{2.5}$$

donde los K_i se calculan como sigue:

$$\begin{cases} K_1 = f(x_k, y_k) \\ k_2 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_1) \\ k_3 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_2) \\ k_4 = f(x_k + h, y_k + hK_3) \end{cases} \tag{2.6}$$

2.2. Revisión de literatura

Habiendo investigado el fundamento teórico relacionado al tema desarrollado, a continuación mostraremos los trabajos previos que influenciaron el presente trabajo, y especificando su aporte a este proyecto. Estos trabajos previos involucran encriptación de imágenes digitales.

2.2.1. Algoritmo de Encriptación de Imágenes Utilizando el Atractor Caótico de Lorenz

Investigación realizada por Iván Felipe Rodriguez, Edilma Isabel Amaya, César Augusto Suárez y José David Moreno en 2017 [32]. Este trabajo propone un algoritmo de encriptación basado en el Atractor Caótico de Lorenz para imágenes a escala de grises. El algoritmo consiste en conseguir sucesiones a partir de la solución del sistema de ecuaciones del Atractor Caótico de Lorenz; luego, a la imagen se le aplica un proceso de permutación de filas y columnas basado en las soluciones obtenidas. Posteriormente se le aplica un proceso de difusión con una matriz obtenida con las soluciones del sistema de ecuaciones de Atractor Caótico de Lorenz.
Finalmente se obtiene una imagen cifrada con una baja correlación entre sus píxeles adyacentes y con la propiedad de descifrarla solo con los valores iniciales con los que se resolvió el sistema de ecuaciones diferenciales del Atractor Caótico de Lorenz. Los autores sugieren que en trabajos a futuro, esta idea se puede extender con otros atractores caóticos y con imágenes a color.

2.2.2. A New Image Encryption Algorithm for Grey and Color Medical Images

Investigación realizada por Sara T. Kamal, Khalid M. Hosny, Taha M. Elgindy, Mohamed M. Darwish y Mostafa M. Fouda en 2021 [33]. Este trabajo propone un método de encriptación basados en los procesos de confusión y difusión para imágenes a escala de grises y a color. La seguridad del algoritmo

propuesto es probado con métricas de entropía, coeficiente de correlación y sensibilidad comparándolo con otros métodos de encriptación.

2.2.3. Conclusiones

Hemos revisado los avances e investigaciones previas a la actual, se puede concluir que el campo de investigación de *encriptación*, es capaz de aportar una capa de seguridad al dato que se desea cifrar, independiente del tipo de dato. Además, si se particulariza en encriptar imágenes, puede ser muy útil para muchos ámbitos como la medicina o la milicia.

Por ello, esta tesis se concentrará en aportar métodos de encriptación y reunir los aportes de las investigaciones previas para proponer un algoritmo de encriptación de imágenes a color en cualquier ámbito.

Capítulo 3

Recursos y Herramientas

En este capítulo se expondrán a detalle los recursos y herramientas tanto de hardware como de software, que fueron necesarias para llevar a cabo la tesis, abordando sus detalles y utilidades.

3.1. Hardware

Así como se mencionó en el primer capítulo, estos últimos años se ha desarrollado una gran cantidad de datos, y con ello, nos vemos en la profunda necesidad de protegerlos. Por ello, hay computadoras muy potentes, dedidadas a realizar un proceso de encriptación para que los datos puedan guardarse o procesarse de una forma más segura. Ahora, también existen computadoras que realizan el proceso opuesto, se dedican a aplicar ataques de fuerza bruta a información encriptada, puede hacerse por distintos motivos; ya sea por seguridad, para demostrar la confianza del algorotmo; o puede ser realizada por un tercero con fines maliciosos.

Procedemos a especificar las características del equipo con el que realizó este proceso de encriptación.

3.1.1. HP 15-dy1xxx

El equipo con que se contó fue esta laptop HP, usada principalmente para el proceso de encriptación de la imagen. Sus especificaciones, mostradas en la Tabla **??**, son modestas y no tiene ningún componente resaltante.

TABLA 3.1: Especificaciones de HP 15-dy1xxx.

Arquitectura de procesador	Intel Core i7-1065G7
Procesador gráfico	Intel Iris Plus Graphics
Frecuencia de procesamiento	1.5 GHz
Núcleos de procesamiento	4
Memoria principal	8 GB
Almacenamiento	512 GB SSD
Sistema Operativo	Linux Manjaro 21.0.7

3.2. Software

A continuación se especifican los detalles del lenguaje y librería utilizados para la implementación del algoritmo propuesto.

3.2.1. Python

Python [10] es un lenguaje interpretado multiuso, considerado como una de las mejores opciones (sino la mejor) para desarrollar programación científica. Se eligió frente a otras alternativas por sus capacidades de implementación y por la gran variedad de librerías que posee para procesamiento de imágenes.

3.2.2. Numpy

Entre los módulos disponibles de python, tenemos a Numpy [5], capaz de procesar datos e imágenes como matrices. Utilizamos numpy a diferencia de las ya conocidas listas en python, porque cuentan con los *ndarray* que son 50 veces más rápidas de procesar que las listas. Además, numpy cuenta con funciones de algebra lineal para procesar matrices [34].

Capítulo 4

Estructuración y Método

En el presente capítulo se detallará el proceso seguido durante la investigación, se divide en 2 partes, el proceso de encriptación y el proceso de descrifrado, este último es el proceso inverso del otro.

4.1. Encriptación de la imagen

4.1.1. Solución del Atractor Caótico de Lorenz

El primer paso es aplicar el método de Runge Kutta de cuarto orden (2.5) para hallar la solución del Sistema de Lorenz con sus respectivas constantes que lo vuelven un sistema caótico ($s = 10$, $r = 28$, $b = 8/3$). Estas soluciones se guardan en 3 tuplas distintas para cada solución de x, y, z.

4.1.2. Solución del Atractor Caótico de Rössler

El segundo paso de nuestro algoritmo es hallar una secuencia de números x, y y z que serán soluciones del Atractor Caótico de Rössler (2.4). Hallaremos estos puntos como lo muestran en [16]. Seguiremos los siguientes pasos:

1. Fijar los valores de las constantes: $a = 0.2$, $b = 0.2$ y $c = 8.0$. Son estas constantes las que le dan un comportamiento caótico.

2. Fijar los valores iniciales que darán inicio a nuestra secuencia de puntos.

3. Para cada punto, podemos hallar el siguiente, estableciendo un dt y un número de iteraciones. Estos valores se reemplazan en la ecuación

y hallamos el siguiente punto. De esta forma, obtenemos una lista de soluciones del Sistema Caótico de Rössler.

4.1.3. Proceso de permutación con Lorenz

En este tercer paso, se realiza un algoritmo de permutación basado en la permutación de linea de onda propuesta en [32]. Seguiremos los siguientes pasos:

1. Obtener las soluciones del Atractor Caótico de Lorenz.
2. Para la primera capa de la imagen, asociar a las filas de la matriz la lista x, y a las columnas, la lista y.
3. Desplazar cada fila de la matriz horizontalmente de acuerdo al valor de la lista x. Se desplaza la fila i, hacia la derecha, de acuerdo al valor en la posición i de la lista.
4. Desplazar cada columna de la matriz verticalmente de acuerdo al valor de la lista y. Se desplaza la columna j, hacia abajo, de acuerdo al valor en la posición j de la lista.
5. Se repite a partir del proceso para las otras 2 capas: Para la segunda capa se realiza el proceso con las soluciones de y, z y para la última capa, se realiza el proceso con las soluciones de z, x. De este modo se encriptan las 3 capas de la imagen.
6. Se repite el proceso de permutación 4 veces para completar un ciclo de permutación de línea de onda.

4.1.4. Proceso de permutación con Rössler

En este cuarto paso, se realiza un algoritmo de permutación basado en la permutación de linea de onda propuesta en [32]. Seguiremos los siguientes pasos:

1. Obtener las soluciones del Atractor Caótico de Rössler.

2. Para la primera capa de la imagen, asociar a las filas de la matriz la lista x, y a las columnas, la lista y.

3. Desplazar cada fila de la matriz horizontalmente de acuerdo al valor de la lista x. Se desplaza la fila i, hacia la derecha, de acuerdo al valor en la posición i de la lista.

4. Desplazar cada columna de la matriz verticalmente de acuerdo al valor de la lista y. Se desplaza la columna j, hacia abajo, de acuerdo al valor en la posición j de la lista.

5. Se repite a partir del proceso para las otras 2 capas: Para la segunda capa se realiza el proceso con las soluciones de y, z y para la última capa, se realiza el proceso con las soluciones de z, x. De este modo se encriptan las 3 capas de la imagen.

6. Se repite el proceso de permutación 4 veces para completar un ciclo de permutación de línea de onda.

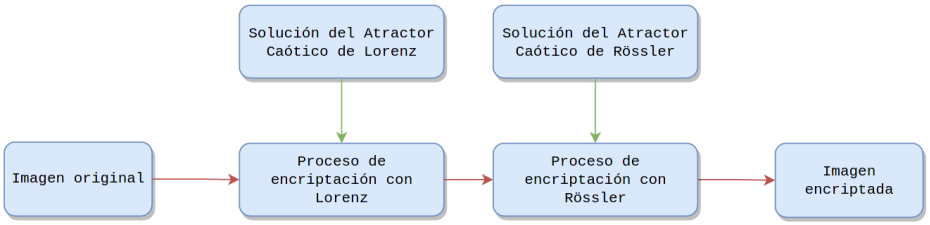


FIGURA 4.1: Proceso de encriptación de imagen. Fuente: Elaboración propia.

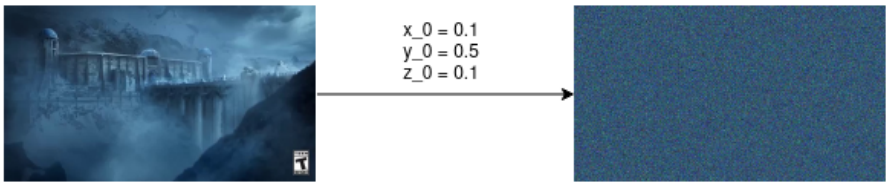


FIGURA 4.2: Aplicación del algoritmo de encriptación de imagen a color.

4.2. Descifrado de la imagen

El proceso de descifrado es el proceso inverso del proceso de encriptación, se trata de obtener la imagen original, teniendo conocimiento de las claves con la que se encriptó la imagen.

4.2.1. Solución del Atractor Caótico de Lorenz

El primer paso es aplicar el método de Runge Kutta de cuarto orden (2.5) para hallar la solución del Sistema de Lorenz con sus respectivas constantes que lo vuelven un sistema caótico ($s = 10$, $r = 28$, $b = 8/3$). Estas soluciones se guardan en 3 tuplas distintas para cada solución de x, y, z.

4.2.2. Solución del Atractor Caótico de Rössler

El segundo paso de nuestro algoritmo es hallar una secuencia de números x, y y z que serán soluciones del Atractor Caótico de Rössler (2.4). Hallaremos estos puntos como lo muestran en [16]. Seguiremos los siguientes pasos:

1. Fijar los valores de las constantes: $a = 0.2$, $b = 0.2$ y $c = 8.0$. Son estas constantes las que le dan un comportamiento caótico.

2. Fijar los valores iniciales que darán inicio a nuestra secuencia de puntos.

3. Para cada punto, podemos hallar el siguiente, estableciendo un dt y un número de iteraciones. Estos valores se reemplazan en la ecuación

y hallamos el siguiente punto. De esta forma, obtenemos una lista de soluciones del Sistema Caótico de Rössler.

4.2.3. Proceso de descifrado con Rössler

En este tercer paso, se realiza un algoritmo de permutación basado en la permutación de linea de onda propuesta en [32]. Seguiremos los siguientes pasos:

1. Obtener las soluciones del Atractor Caótico de Rössler.
2. Para la primera capa de la imagen, asociar a las filas de la matriz la lista x, y a las columnas, la lista y.
3. Desplazar cada columna de la matriz verticalmente, de acuerdo al valor de la lista y. Se desplaza la columna j, hacia arriba, de acuerdo al valor en la posición j de la lista.
4. Desplazar cada fila de la matriz horizontalmente, de acuerdo al valor de la lista x. Se desplaza la fila i, hacia la izquierda, de acuerdo al valor en la posición i de la lista.
5. Se repite a partir del proceso para las otras 2 capas: Para la segunda capa se realiza el proceso con las soluciones de y, z y para la última capa, se realiza el proceso con las soluciones de z, x. De este modo se encriptan las 3 capas de la imagen.
6. Se repite el proceso de permutación 4 veces para completar un ciclo de permutación de línea de onda.

4.2.4. Proceso de descifrado con Lorenz

En este cuarto paso, se realiza un algoritmo de permutación basado en la permutación de linea de onda propuesta en [32]. Seguiremos los siguientes pasos:

1. Obtener las soluciones del Atractor Caótico de Lorenz.

2. Para la primera capa de la imagen, asociar a las filas de la matriz la lista x, y a las columnas, la lista y.

3. Desplazar cada columna de la matriz verticalmente de acuerdo al valor de la lista y. Se desplaza la columna j de acuerdo al valor en la posición j de la lista.

4. Desplazar cada fila de la matriz horizontalmente de acuerdo al valor de la lista x. Se desplaza la fila i de acuerdo al valor en la posición i de la lista.

5. Se repite a partir del proceso para las otras 2 capas: Para la segunda capa se realiza el proceso con las soluciones de y, z y para la última capa, se realiza el proceso con las soluciones de z, x. De este modo se encriptan las 3 capas de la imagen.

6. Se repite el proceso de permutación 4 veces para completar un ciclo de permutación de línea de onda.

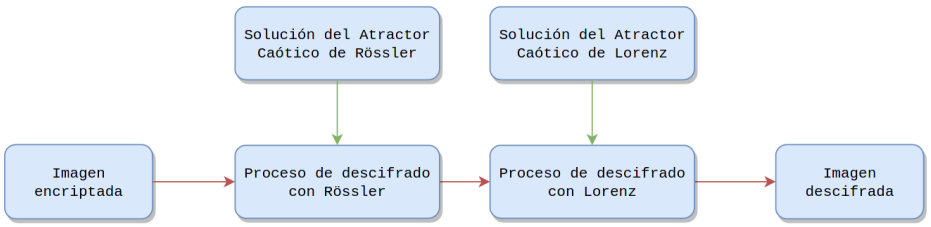


FIGURA 4.3: Proceso de descifrado de imagen. Fuente: Elaboración propia.

Capítulo 5

Resultados

En este capítulo se detallará el resultado final, considerando las métricas que describen la eficiencia del algoritmo y evidenciando su eficacia frente a otros algoritmos.

5.1. Performance

La principal característica de este proceso de encriptación, es que se requiere exactamente las claves iniciales con las que se encriptó para que se logre descifrar la imagen. Si se ingresan números cercanos a los originales, no se obtendrá una imagen ni siquiera similar a la imagen original. Esto se debe a la propiedad caótica de los atractores caóticos, que generan listas con valores bastante alejados unos de otros; entonces, con valores iniciales distintos, por más cercanos que sean, se obtienen iteraciones con valores completamente diferentes, y estos no pueden descifrar la imagen original. Ni si quiera permite percibir una imagen cercana a la imagen original.

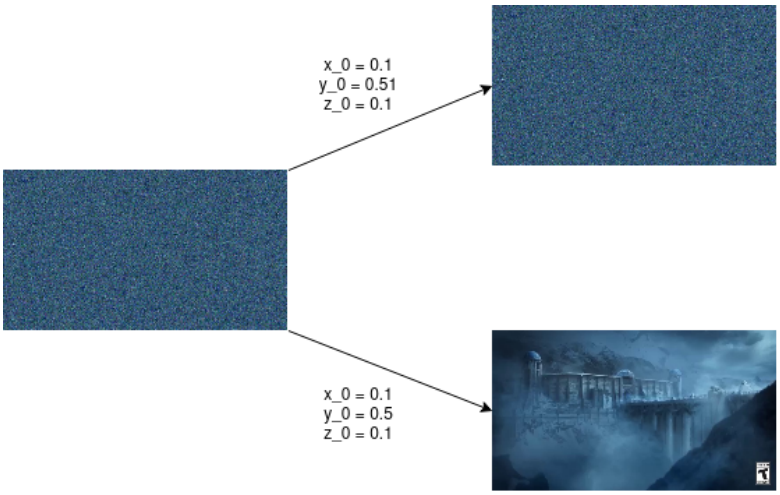


FIGURA 5.1: Descifrado con claves correctas e incorrectas . Fuente: Elaboración propia.

5.2. Métricas

Como se describió previamente en el Capítulo 2, las métricas empleadas evalúan qué tan óptimo es el algoritmo de encriptación, evidenciando la aleatoriedad de los píxeles resultantes y la distribución de píxeles para que no se pueda evidenciar la imagen original a simple vista. Utilizamos las imágenes de Lena y Baboon para realizar nuestras pruebas de métricas.

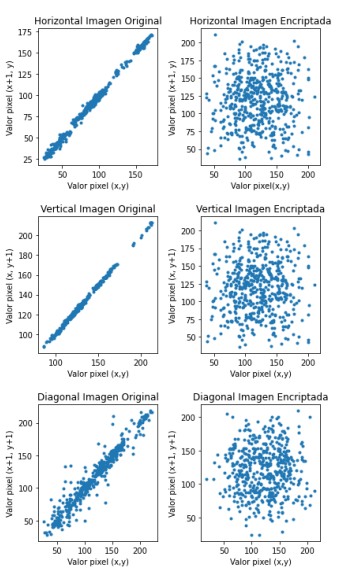
(A) Lena original. (B) Lena encriptado.

FIGURA 5.2: Imagen "Lena", tamaño 512x512.

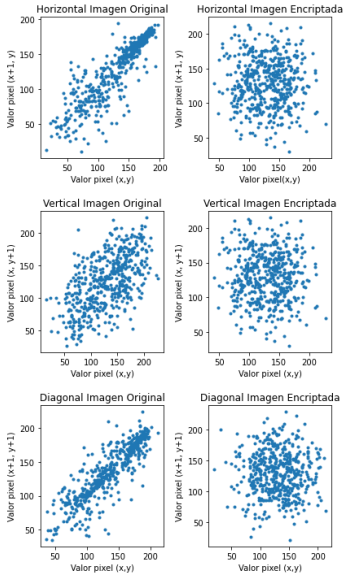
(A) Baboon original. (B) Baboon encriptado.

FIGURA 5.3: Imagen "Baboon", tamaño 512x512.

Analizamos la correlación entre los pixeles adyacentes en forma horizontal, vertical y diagonal, tanto para la imagen original y la cifrada. Se obtiene que la correlación entre pixeles tiende a desaparecer al aplicar el algoritmo de encriptación, como vemos en las figuras 5.4a y 5.4b.



(A) Correlación entre píxeles adyacentes de la imagen Lena.



(B) Correlación entre píxeles adyacentes de la imagen Baboon.

FIGURA 5.4: Análisis de correlación entre pixeles adyacentes.

El cuadro 5.1 nos muestra que las imágenes de prueba, transformadas a escala de grises, tienen valores de coeficiente de correlación cercanos a 1, lo que indica que los valores entre pixeles adyacentes son cercanamente idénticos; mientras que las imágenes encriptadas y transformadas a escala de grises, tienen valores cercanos a cero, lo que indica que los pixeles adyacentes no guardan relación entre si.

TABLA 5.1: Valores de correlación de pixeles adyacentes.

Imagen	Horizontal	Vertical	Diagonal
Lena original	0.99784	0.99859	0.96335
Lena encriptado	-0.00271	0.02619	0.02029
Baboon original	0.89109	0.57361	0.86805
Baboon encriptado	-0.10606	-0.03397	-0.01724

El cuadro 5.2 nos muestra los valores de entropía de las imágenes encriptadas y transformada a escala de grises, estos valores son cercanos a 8, lo que indica que hay una aleatoriedad alta en los pixeles de la imagen. Este valor puede ser más cercano a 8 si se realizan más ciclos de permutación.

TABLA 5.2: Valores de entropía.

Imagen encriptada	Entropia
Lena	7.4321
Baboon	7.3738

El algoritmo de encriptación fue ejecutado en una laptop, cuyas especificaciones son indicadas en el cuadro 3.1; los resultados del coeficiente de correlación de este y los obtenidos por [33, 20, 15] se muestran en el cuadro 5.3; podemos notar valores muy similares, esto se debe a que también presentan algoritmos de permutación, con otros métodos, que pueden tener más o menos ciclos de permutación, o aplicar otro tipo de algoritmo adicional.

TABLA 5.3: Comparación de correlación de pixeles adyacentes.

Imagen encriptada	Horizontal	Vertical	Diagonal
Lena	-0.00271	0.02619	0.02029
Baboon	-0.10606	-0.03397	-0.01724
[33]	-0.0182	-0.0108	0.0165
[20]	0.0944	0.0057	0.0067
[15]	0.0098	-0.0078	0.0181

El cuadro 5.4 muestra los tiempos de ejecución del algoritmo propuesto al encriptar la imagen 5.2a varias cantidades de veces, se hace este cálculo, por si se tiene la necesidad de ejecutar el algoritmo con varias imágenes, como trabajo a futuro, se podria tomar para encriptar muchos frames de un video.

TABLA 5.4: Tiempo de ejecución con varias imágenes.

Cantidad de imagenes	Tiempo de ejecución
1	1.967 s
10	20.587 s
100	210. 541 s
500	1059.152 s

Capítulo 6

Conclusiones y Trabajo Futuro

En este último capítulo, se muestran las conclusiones generales a partir de los resultados obtenidos, que son fruto del proceso de elaboración de esta tesis. Además, tiene en cuenta aspectos que se estudiarán para trabajos similares en el futuro y brinda orientación en caso se desee realizar una investigación tomando como base esta tesis.

6.1. Conclusiones

Esta tesis propone un algoritmo de encriptación basado en el comportamiento de los atractores caóticos de Lorenz y Rössler, el cual presenta un buen desempeño de seguridad y tiempo de ejecución. En un inicio, se presentó la dificultad de aplicar correctamente los valores de las soluciones de los Sistemas Caóticos, ya que se deben aprovechar sus propiedades para lograr una baja correlación entre los pixeles adyacentes; por ello, se tomó como referencia el algoritmo implementado en otras referencias.

De lo explicado hasta este punto, y de manera más descriptiva, se concluye lo siguiente:

- La propiedad caótica de los Atractores Caóticos optimiza los procesos de encriptación si se procede con un algoritmo que aproveche esta propiedad.
- El algoritmo de encriptación propuesto cumple con los estándares de seguridad según las métricas mostradas en el capítulo anterior.

- El algoritmo de encriptación propuesto está al nivel de los demás algoritmos de encriptación, comparando las métricas con otros algoritmos de referencia.

6.2. Trabajo a futuro

Una de las razones por las que se desarrolló la presente tesis, fue para aplicarla a los frames de videos, y adaptarlo a un algoritmo de encriptación de videos; por lo que el punto principal es obtener un algoritmo que se pueda optimizar en tiempo de ejecución porque se debe aplicar a muchos frames de un video. Por supuesto que al aplicar este algoritmo de encriptación a videos, se debe sincronizar con otro algoritmo para encriptar sonidos; por tal motivo, antes de desarrollar un algoritmo de encriptación de videos, se debe investigar previamente algún método de encriptación de sonido.

Si el lector desea profundizar en el presente tema, se aconseja tener en cuenta la aplicación de la aleatoriedad del Sistema Caótico. Académicamente, se debe fijar el objetivo general desde un inicio, y debe adaptarse a los problemas que surjan en el trayecto. Cabe destacar que los alcances de la investigación lo define el investigador, y su surjen problemas al momento de desarrollar su propuesta, es normal, porque parte del desarrollo de la investigación, es encontrar errores y solucionarlos.

Bibliografía

[1] Narasimhan Aarthie y R. Amirtharajan. «Image Encryption: An Information Security Perceptive». En: *Journal of Artificial Intelligence* 7 (sep. de 2014), págs. 123-135. DOI: 10.3923/jai.2014.123.135.

[2] R. Amirtharajan, P. Archana y John Bosco Balaguru Rayappan. «Why Image Encryption for Better Steganography». En: *Research Journal of Information Technology* 5 (mar. de 2013), págs. 341-351. DOI: 10.3923/rjit.2013.341.351.

[3] Canto. *Image watermarking: An effective way to protect digital images*. URL: https://www.canto.com/image-watermarking/.

[4] Dra. María del Carmen Gómez Fuentes. *Runge-Kutta orden 4*. URL: http://test.cua.uam.mx/MN/Methods/EcDiferenciales/Runge-Kutta/RungeKutta.php.

[5] Numpy Community. *Numpy*. URL: https://numpy.org/.

[6] CompTia. *The Ancient Practice of Steganography: What Is It, How Is It Used and Why Do Cybersecurity Pros Need to Understand It*. 2020. URL: https://www.comptia.org/blog/what-is-steganography.

[7] Comsol. *Rössler Attractor*. URL: https://www.comsol.com/model/r-ssler-attractor-10656.

[8] Eduardo A.B. da Silva y Gelson V. Mendonça. «Digital Image Processing». En: *The Electrical Engineering Handbook*. Ed. por WAI-KAI CHEN. Burlington: Academic Press, 2005, págs. 891-910. ISBN: 978-0-12-170960-0. DOI: https://doi.org/10.1016/B978-012170960-0/50064-5. URL: https://www.sciencedirect.com/science/article/pii/B9780121709600500645.

[9] Editors of Encyclopedia Britannica. *Chaos theory*. URL: https://www.britannica.com/science/chaos-theory.

[10] Python Software Foundation. *Python*. URL: https://www.python.org/.

[11] Universidad de Granada. *Método de Runge-Kutta*. URL: https://www.ugr.es/~lorente/APUNTESMNQ/cap23.pdf.

[12] Khalid Hosny y Mohamed Darwish. «Resilient Color Image Watermarking Using Quaternion Radial Substituted Chebychev Moments». En: *ACM Transactions on Multimedia Computing, Communications and Applications* 15 (abr. de 2019). DOI: 10.1145/3325193.

[13] Khalid Hosny y Mohamed Darwish. «Robust color image watermarking using invariant quaternion Legendre-Fourier moments». En: *Multimedia Tools and Applications* 77 (oct. de 2018). DOI: 10.1007/s11042-018-5670-9.

[14] Khalid M. Hosny y col. «Parallel Multi-Core CPU and GPU for Fast and Robust Medical Image Watermarking». En: *IEEE Access* 6 (2018), págs. 77212-77225. DOI: 10.1109/ACCESS.2018.2879919.

[15] Zhongyun Hua, Shuang Yi y Yicong Zhou. «Medical image encryption using high-speed scrambling and pixel adaptive diffusion». En: *Signal Processing* 144 (oct. de 2017). DOI: 10.1016/j.sigpro.2017.10.004.

[16] Darren Dale John Hunter, Michael Droettboom Eric Firing y the Matplotlib development team. *Lorenz Attractor*. 2012. URL: https://matplotlib.org/devdocs/gallery/mplot3d/lorenz_attractor.html.

[17] S.E. Jørgensen. «Chaos». En: *Encyclopedia of Ecology*. Ed. por Sven Erik Jørgensen y Brian D. Fath. Oxford: Academic Press, 2008, págs. 550-551. ISBN: 978-0-08-045405-4. DOI: https://doi.org/10.1016/B978-008045405-4.00148-8. URL: https://www.sciencedirect.com/science/article/pii/B9780080454054001488.

[18] Yu-Lan Wang Jun-Mei Li y Wei Zhang. «Numerical Simulation of the Lorenz-Type Chaotic System Using Barycentric Lagrange Interpolation Collocation Method». En: *Advances in Mathematical Physics* 2019 (abr. de 2019), 9 pages, 2019. DOI: https://doi.org/10.1155/2019/1030318.

[19] Aerin Kim. *The intuition behind Shannon's Entropy*. URL: https://towardsdatascience.com/the-intuition-behind-shannons-entropy-e74820fe9800.

[20] Sumit Kumar, Rajib Kumar Jha y Bhaskar Panna. «Medical Image Encryption Using Fractional Discrete Cosine Transform with Chaotic Function». En: *Medical Biological Engineering Computing* (ago. de 2019). DOI: 10.1007/s11517-019-02037-3.

[21] Sebastian Kwiatkowski. *Entropy is a measure of uncertainty*. URL: https://towardsdatascience.com/entropy-is-a-measure-of-uncertainty-e2c000301c2c.

[22] Xin Liao y col. «Medical JPEG image steganography based on preserving inter-block dependencies». En: *Computers Electrical Engineering* 67 (2018), págs. 320-329. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2017.08.020. URL: https://www.sciencedirect.com/science/article/pii/S0045790617302756.

[23] George Lindfield y John Penny. *Solution of Differential Equations*. URL: https://www.sciencedirect.com/topics/mathematics/lorenz-system.

[24] Jizhao Liu y col. «A new simple chaotic system and its application in medical image encryption». En: *Multimedia Tools and Applications* 77 (sep. de 2018). DOI: 10.1007/s11042-017-5534-8.

[25] Jizhao Liu y col. «A novel fourth order chaotic system and its algorithm for medical image encryption». En: *Multidimensional Systems and Signal Processing* 30 (oct. de 2019). DOI: 10.1007/s11045-018-0622-0.

[26] Nate Lord. *What Is Data Encryption? Definition, Best Practices More*. URL: https://digitalguardian.com/blog/what-data-encryption.

[27] René Lozi. «Can we trust in numerical computations of chaotic solutions of dynamical systems?» En: 84 (feb. de 2012). DOI: 10.1142/9789814434867_0004.

[28] Wolfram Mathworld. *Rössler Attractor*. URL: https://mathworld.wolfram.com/RoesslerAttractor.html.

[29] A. Miranda Neto y col. «Image processing using Pearson's correlation coefficient: Applications on autonomous robotics». En: *2013 13th International Conference on Autonomous Robot Systems*. 2013, págs. 1-6. DOI: 10.1109/Robotica.2013.6623521.

[30] Facultad Regional San Nicolás. *Métodos de Runge Kutta*. URL: http://www.frsn.utn.edu.ar/gie/an/mnedo/34_rk.html.

[31] He Liu - Quora. *What is digital image and digital image processing?* URL: https://www.quora.com/What-is-digital-image-and-digital-image-processing.

[32] Iván Rodríguez y col. «Algoritmo de Encriptación de Imágenes Utilizando el Atractor Caótico de Lorenz». En: *Ingeniería* 22 (sep. de 2017), pág. 396. DOI: 10.14483/23448393.11976.

[33] Khalid M. Hosny Sara T. Kamal, Mohamed M. Darwish Taha M. Elgindy y Mostafa M. Fouda. *A New Image Encryption Algorithm for Grey and Color Medical Images*. URL: https://ieeexplore.ieee.org/document/9366688.

[34] W3 Schools. *NumPy Introduction*. URL: https://www.w3schools.com/python/numpy/numpy_intro.asp.

[35] Margie Semilof y Casey Clark. *steganography*. 2018. URL: https://searchsecurity.techtarget.com/definition/steganography.

[36] William Stallings y Lawrie Brown. *Computer Security: Principles and Practice*. Pearson Education, 2018. ISBN: 9780134794396.

[37] Muhammad Arslan Usman y Muhammad Rehan Usman. «Using image steganography for providing enhanced medical data security». En: *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 2018, págs. 1-4. DOI: 10.1109/CCNC.2018.8319263.

[38] A.M. Vengadapurvaja y col. «An Efficient Homomorphic Medical Image Encryption Algorithm For Cloud Storage Security». En: *Procedia Computer Science* 115 (dic. de 2017), págs. 643-650. DOI: 10.1016/j.procs.2017.09.150.

[39] Huihai Wang, Shaobo He y Kehui Sun. «Complex Dynamics of the Fractional-Order Rössler System and Its Tracking Synchronization Control». En: 2018 (dic. de 2018), pág. 13. DOI: 10.1155/2018/4019749.

[40] Eric W. Weisstein. *Attractor*. URL: https://mathworld.wolfram.com/Attractor.html.

[41] Eric W. Weisstein. *Basin of Attraction*. URL: https://mathworld.wolfram.com/BasinofAttraction.html.

[42] Yue Zhang y col. «Zernike Moment-Based Spatial Image Steganography Resisting Scaling Attack and Statistic Detection». En: *IEEE Access* 7 (2019), págs. 24282-24289. DOI: 10.1109/ACCESS.2019.2900286.