



UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

Proyecto de Tesis 2

*Encriptación de Sonidos basado en los Sistemas
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 sonidos, aprovechando las propiedades de los Sistemas Caóticos. Esta tesis explora los alcances producidos al encriptar sonidos basados en las propiedades de los Sistemas 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

Resumen	III
1. Introducción	2
1.1. Motivación	2
1.2. Objetivos	3
1.3. Estructura de la Tesis	4
2. Estado del Arte	5
2.1. Trasfondo Teórico	5
2.1.1. Introducción a la Seguridad Informática	5
Confidencialidad	6
Integridad	6
Disponibilidad	6
2.1.2. Encriptación	7
2.1.3. Digital Audio	7
2.1.4. Espectrograma	7
2.1.5. Cryptography	8
2.1.6. Audio Encryption	8
2.1.7. Métricas	8
Análisis de Histogramas	8
Coeficiente de Correlación de Pearson	9
2.1.8. Teoría del Caos	9
2.1.9. Sistema Caótico de Lorenz	9
2.1.10. Sistema Caótico de Rössler	11
2.1.11. Runge Kutta 4	11
2.2. Revisión de literatura	13

2.2.1. A New Audio Encryption Algorithm Based on Chaotic Block Cipher	13
2.2.2. An Audio Encryption Algorithm Based on DNA Coding and Chaotic System	13
2.2.3. Conclusiones	14
3. Recursos y Herramientas	15
3.1. Hardware	15
3.1.1. HP 15-dy1xxx	15
3.2. Software	16
3.2.1. Python	16
3.2.2. Numpy	16
3.2.3. ffmpeg	17
3.2.4. SoundFile	17
3.2.5. CV2	17
3.2.6. Algoritmo de Conversión de Audio a Imagen	17
4. Estructuración y Método	18
4.1. Encriptación del Audio	18
4.1.1. Conversión de Sonido a Imagen	18
4.1.2. Solución del Sistema Caótico de Lorenz	18
4.1.3. Solución del Sistema Caótico de Rössler	19
4.1.4. Proceso de permutación con Lorenz	19
4.1.5. Proceso de permutación con Rössler	20
4.1.6. Conversión de Imagen a Sonido	20
4.2. Descifrado del Audio	21
4.2.1. Conversión de Sonido a Imagen	22
4.2.2. Solución del Sistema Caótico de Lorenz	22
4.2.3. Solución del Sistema Caótico de Rössler	22
4.2.4. Proceso de descifrado con Lorenz	22
4.2.5. Proceso de descifrado con Rössler	23
4.2.6. Conversión de Imagen a Sonido	24

5. Resultados	25
5.1. Performance	25
5.2. Métricas	26
6. Conclusiones y Trabajo Futuro	30
6.1. Conclusiones	30
6.2. Trabajo a Futuro	31

Índice de Figuras

Índice de Acrónimos

SI	Seguridad Informática
SCL	Sistema Caótico de Lorenz
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 un archivo de sonido, de tal forma que no se pueda reproducir el sonido original a simple vista, y tras aplicar el algoritmo de descifrado con las claves correctas, podemos obtener el sonido original. Existen muchas formas de encriptación de sonidos, en esta tesis se propone aplicar una transformación de sonido a imagen, y posteriormente aplicar las propiedades de los Sistemas Caóticos de Lorenz y Rössler para permutar las filas y columnas de la imagen.

De este modo, el objetivo de encriptar el sonido surge de la necesidad de ocultar información o datos sensibles, desde audios personales, hasta audios producto de alguna actividad de espionaje o militar.

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 debería 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 sonidos), 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 más 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 sonidos, utilizando las propiedades de los Sistemas Caóticos de Rössler.

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

- Investigar las propiedades de un sistema caótico para encriptación.
- Implementar un algoritmo de encriptación de sonidos que cumpla con los estándares que tienen otros algoritmos de encriptación.
- 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 audios.

- 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 los audios 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 [37]:

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.

Para intentar asegurar la confidencialidad, los dueños y protectores de la información sensible, implementan políticas, basadas en un número de procesos sobre dispositivos y las personas que se encargan de manejar la data [16].

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.

La integridad de la información se puede definir como la confiabilidad de la información, en los procesos y sistemas de la información [2].

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.

La disponibilidad está asociado con la confiabilidad y el tiempo de actividad del sistema, que se puede ver afectado por problemas maliciosos como ciberataques o problemas no maliciosos como fallas de hardware y error humano [39].

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 [23].

2.1.3. Digital Audio

El audio digital es una tecnología que es usada para grabar, almacenar, manipular y reproducir sonidos usando señales de audio que han sido codificados en forma digital. También se refiere a la secuencia de muestras discretas que se toman de una forma de onda de audio analógica. En lugar de una onda sinusoidal continua, el audio digital se compone de puntos discretos que representan la amplitud de la forma de onda aproximadamente. La mayoría de los dispositivos multimedia modernos solo pueden procesar audio digital [40].

Los formatos de audio digital son los contenedores que almacenan los audios digitales, se diferencian principalmente en: Si están comprimidos o no, la calidad de impresión si están comprimidos y el tipo de etiquetado que puede soportar. Los formatos más conocidos son: MP3, WAV, WMA, OGG, etc [11].

2.1.4. Espectrograma

Un espectrograma es una forma visual de representar la intensidad de la señal, o "sonoridad", de una señal a lo largo del tiempo en varias frecuencias presentes en una forma de onda particular [28]. El espectrograma es una herramienta básica en el análisispectral de audio y otros campos. Es una representación importante de los datos de audio porque el oido humano se basa en un tipo de espectrograma en tiempo real codificado por la cóclea del oido interno [8].

2.1.5. Cryptography

Criptografía es el estudio de las técnicas de comunicaciones seguras que permiten solo al emisor y receptor, ver el mensaje. Este mensaje no necesariamente es un texto, puede ser también una imagen, audio o video [20]. En el campo de la informática, la criptografía se refiere a las técnicas de información y comunicación seguras derivadas de conceptos matemáticos y un conjunto de cálculos basados en reglas llamados algoritmos, para transformar mensajes de manera difícil de descifrar [36].

2.1.6. Audio Encryption

La encriptación es una técnica usada para transmitir información segura. A lo largo de los años, muchas técnicas de encriptación han sido implementadas, pero la mayoría de estas técnicas solo encriptan texto, y muy pocas fueron propuestas para datos multimedia como audio. Las técnicas que encriptan datos de texto pueden ser aplicados a datos de audio, pero no logran resultados satisfactorios. La encriptación de audio suele ser difícil y compleja [25].

2.1.7. Métricas

Análisis de Histogramas

Los histogramas son una herramienta muy poderosa para analizar datos porque muestran la distribución de una variable continua en un diagrama y su apariencia es similar a los gráficos de barras. Los valores de audio son graficados antes y después de encriptar, para observar su efecto. Al ver las imágenes, se debe notar que el audio encriptado no muestra información útil sobre el audio original. Estas características ayudan a proteger de ataques contra el archivo encriptado [32].

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 audios encriptados con una baja correlación entre los píxeles adyacentes de sus imágenes [1]. Matemáticamente, el coeficiente de correlación de Pearson entre 2 píxeles adyacentes se define como [27]:

$$r = \frac{\sum_i(x_i - x_m)(y_i - y_m)}{\sqrt{\sum_i(x_i - x_m)^2} \sqrt{\sum_i(y_i - y_m)^2}} \quad (2.1)$$

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.

2.1.8. Teoría del Caos

La Teoría del Caos está muy relacionada con los eventos impredecibles o comportamiento aleatorio [18]. Podemos pensar en aleatoriedad como el 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 [12].

2.1.9. Sistema 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 [19], biología y modelización de poblaciones integradas, astrodinámica moderna, dimensiones fractales en

dinámica [22]. El sistema de ecuaciones diferenciales del Sistema de Lorenz tiene la forma:

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = x(c - x) - y \\ \frac{dz}{dt} = -bz + xy \end{cases} \quad (2.2)$$

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 [17].

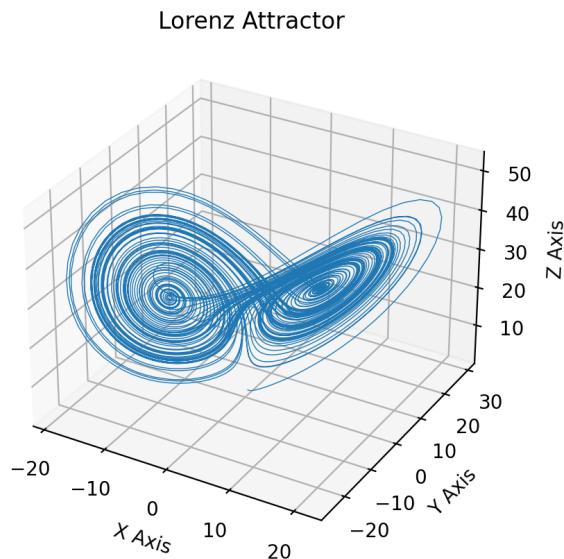


FIGURA 2.1: Sistema Caótico de Lorenz con $a = 10$, $b = 8/3$ y $c = 28$.

Utilizamos este Sistema Caótico de Lorenz por sus propiedades caóticas como imprevisibilidad y sensibilidad a las condiciones iniciales [33].

2.1.10. Sistema 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 [9].

El sistema de ecuaciones diferenciales del Sistema de Rossler tiene la forma:

$$\begin{cases} \frac{dx}{dt} = -(y + z) \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + xz - cz \end{cases} \quad (2.3)$$

Donde $a = 0.2$, $b = 0.2$ y $c = 8.0$ para que el sistema de ecuaciones presente un comportamiento caótico [26], aunque puede tener distintas formas variando estos parámetros, como se ve en [41].

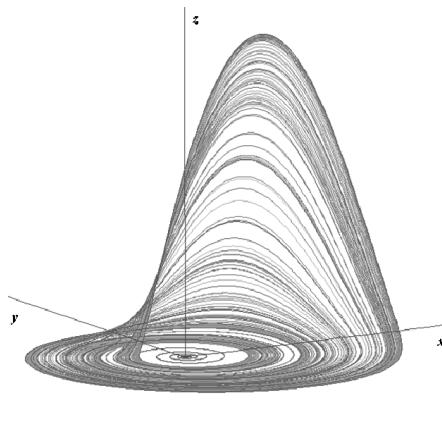


FIGURA 2.2: Sistema Caótico de Rössler con $a = 0.2$, $b = 0.2$ y $c = 8.0$ [24].

2.1.11. 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 ← (b - a)/N
t0 ← a
y0 ← α
Para i desde 0 hasta N-1 hacer
    ti ← a + i * h
    k1 ← hf(ti, yi)
    k2 ← hf(ti +  $\frac{1}{2}h$ , yi +  $\frac{1}{2}k_1$ )
    k3 ← hf(ti +  $\frac{1}{2}h$ , yi +  $\frac{1}{2}k_2$ )
    k4 ← hf(ti + h, yi + k3)
    yi+1 ← yi +  $\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ 
Fin Para
Mostrar (t0, y0), (t1, y1), ... (tN, yN)
FIN

```

FIGURA 2.3: Algoritmo de Runge Kutta 4 [29].

El método se describe con las siguientes ecuaciones [14]:

$$\begin{cases} y_0 = y_a \\ y_{k+1} = y_k + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), k = 0, 1, \dots \end{cases} \quad (2.4)$$

donde los K_i se calculan como sigue:

$$\begin{cases} K_1 = f(x_k, y_k) \\ k_2 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_1) \\ k_3 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}K_2) \\ k_4 = f(x_k + h, y_k + hK_3) \end{cases} \quad (2.5)$$

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 audios digitales.

2.2.1. A New Audio Encryption Algorithm Based on Chaotic Block Cipher

Investigación realizada por la PhD Ekhlas Abbas Albahrani en 2017 [1]. Este trabajo propone un algoritmo de encriptación basado en combinación entre bloques de cifrado y mapas caóticos. Este algoritmo encripta y desencripta bloques de 625 bytes, que son las partes en las que divide el archivo de audio original. Consta de 3 fases: Fase de permutación (en la que se aplica el mapa caótico), fase xor y fase de sustitución. La llave es generada por un algoritmo basado en el polinomio de Chebyshev.

Finalmente, en el análisis de resultados, se muestra con las métricas, que el algoritmo no es vulnerable a ataques de fuerza bruta, ataques estadísticos y logra un alto nivel de seguridad.

2.2.2. An Audio Encryption Algorithm Based on DNA Coding and Chaotic System

Investigación realizada por Xingyuan Wang y Yining Su en el 2020 [42]. Este trabajo propone un método de encriptación de audios basado en sistemas caóticos y códigos de ADN para realizar procesos de confusión y difusión a los audios. El valor inicial del sistema caótico es controlado por el valor hash del audio, haciendo que la trayectoria caótica sea inpredecible.

Finalmente, comparando con otros experimentos, se muestra que el algoritmo funciona bien y es seguro ante ataques comunes.

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 sonidos, puede ser muy útil para muchos ámbitos como el espionaje 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 audios 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, dedicadas 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 algoritmo; 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 3.1, son modestas y no tiene ningún componente resaltante.

TABLA 3.1: Especificaciones de HP 15-dy1xx.

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 [13] 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 [6], 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 álgebra lineal para procesar matrices [35].

3.2.3. **ffmpy**

ffmpy es un wrapper de Python para FFmpeg [10]. Este compila las líneas de comando de FFmpeg desde los argumentos proveidos y sus respectivas opciones, y lo ejecuta como un subprocesso de Python.

Esta librería es capaz de cambiar el formato de archivos de audio y video [43].

3.2.4. **SoundFile**

SoundFile puede leer y escribir sobre archivos de audio. La lectura y escritura de archivos es soportado a través de libsndfile [5], el cual es una biblioteca que trabaja con distintos formatos de archivos de audio. SoundFile representa los datos de audio como arrays de numpy [3].

3.2.5. **CV2**

CV2 es paquete de envoltura para los enlaces de Python de OpenCV [7]. OpenCV provee bibliotecas, herramientas, y hardware de Computer Vision. Desarrollada originalmente por Intel [38].

3.2.6. **Algoritmo de Conversión de Audio a Imagen**

Algoritmo propuesto por Harsh Patel [31] que convierte archivos de audio a imágenes utilizan las librerías de Python ffmpy, Soundfile, cv2 y numpy para ocultar información en imágenes [30]. Utilizamos parte de este código para convertir el archivo de audio a imagen y viceversa.

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 desencriptado, este último es el proceso inverso del otro.

4.1. Encriptación del Audio

4.1.1. Conversión de Sonido a Imagen

Utilizamos un algoritmo propuesto por Harsh Patel [31], se puede ingresar archivos de audio de formato MP3 o WAV, y lo convierte a un formato de imagen PNG.

4.1.2. Solución del Sistema Caótico de Lorenz

Aplicamos el método de Runge Kutta de cuarto orden (2.4) 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 .

4.1.3. Solución del Sistema Caótico de Rössler

Aplicamos el método de Runge Kutta de cuarto orden (2.4) para hallar la solución del Sistema de Rössler con sus respectivas constantes que lo vuelven un sistema caótico ($a = 0.2$, $b = 0.2$ y $c = 8.0$). Estas soluciones se guardan en 3 tuplas distintas para cada solución de x, y, z .

4.1.4. Proceso de permutación con Lorenz

En este paso, se realiza un algoritmo de permutación basado en la permutación de linea de onda propuesta en [33]. Seguiremos los siguientes pasos:

1. Obtener las soluciones del Sistema 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.

4.1.5. Proceso de permutación con Rössler

En este paso, se realiza un algoritmo de permutación basado en la permutación de linea de onda propuesta en [33]. Seguiremos los siguientes pasos:

1. Obtener las soluciones del Sistema 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.

4.1.6. Conversión de Imagen a Sonido

Utilizamos un algoritmo propuesto por Harsh Patel [31], se puede ingresar la imagen encriptada y nos retorna un archivo de audio en formato MP3. Este paso se traduce en una capa más de seguridad, ya que al reproducir el audio encriptado, no se puede reconocer el audio original.

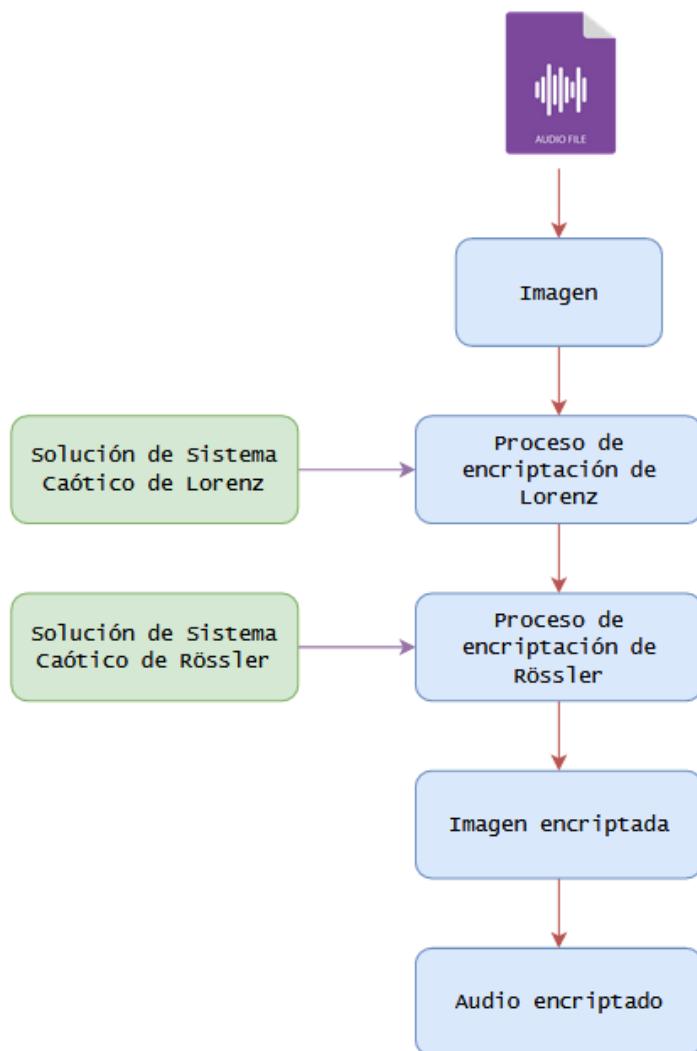


FIGURA 4.1: Proceso de encriptación de imagen. Fuente:
Elaboración propia.

4.2. Descifrado del Audio

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

4.2.1. Conversión de Sonido a Imagen

Utilizamos un algoritmo propuesto por Harsh Patel [31], se puede ingresar archivos de audio de formato MP3 o WAV, y lo convierte a un formato de imagen PNG.

4.2.2. Solución del Sistema Caótico de Lorenz

En este paso se aplica el método de Runge Kutta de cuarto orden (2.4) 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 .

4.2.3. Solución del Sistema Caótico de Rössler

Aplicamos el método de Runge Kutta de cuarto orden (2.4) para hallar la solución del Sistema de Rössler con sus respectivas constantes que lo vuelven un sistema caótico ($a = 0.2$, $b = 0.2$ y $c = 8.0$). Estas soluciones se guardan en 3 tuplas distintas para cada solución de x, y, z .

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 [33]. 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.

4.2.5. Proceso de descifrado con Rössler

En este paso, se realiza un algoritmo de permutación basado en la permutación de linea de onda propuesta en [33]. Seguiremos los siguientes pasos:

1. Obtener las soluciones del Sistema 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.

4.2.6. Conversión de Imagen a Sonido

Utilizamos el algoritmo propuesto con la ayuda de la biblioteca SoundFile, para convertir imágenes a audios. Nos debe retornar el audio original siempre que no se haya perdido datos al realizar todo el proceso de encriptación.

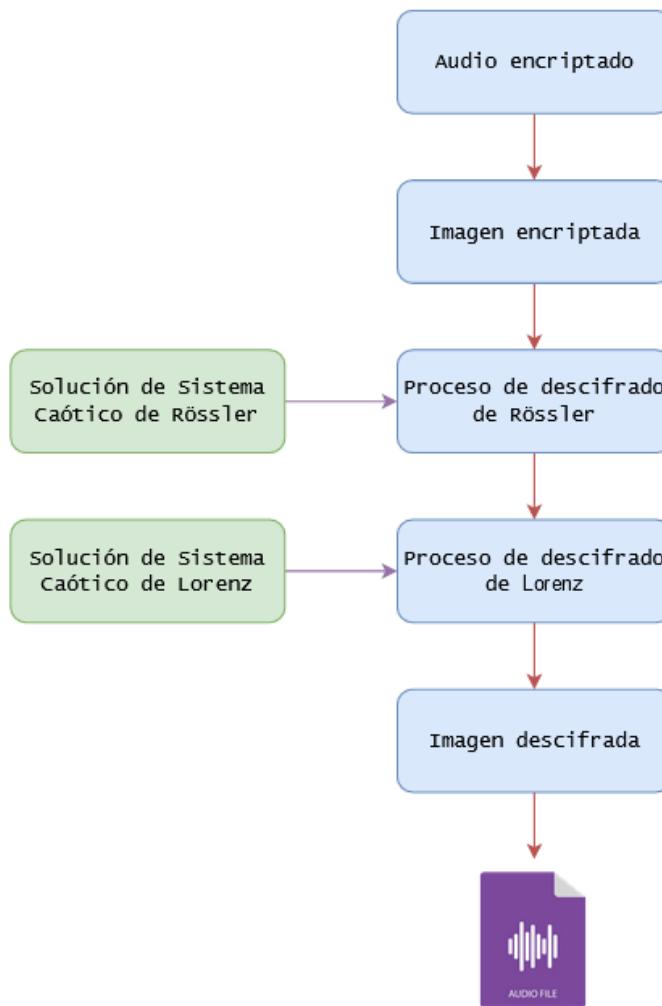


FIGURA 4.2: Proceso de descifrado de imagen. Fuente:
Elaboración propia.

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 sistemas 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 el audio original. Ni si quiera debería percibirse un audio parecido al audio original.

Haciendo un análisis del algoritmo propuesto, se evidencia una complejidad de $O(n^2)$, donde n es el tiempo del audio.

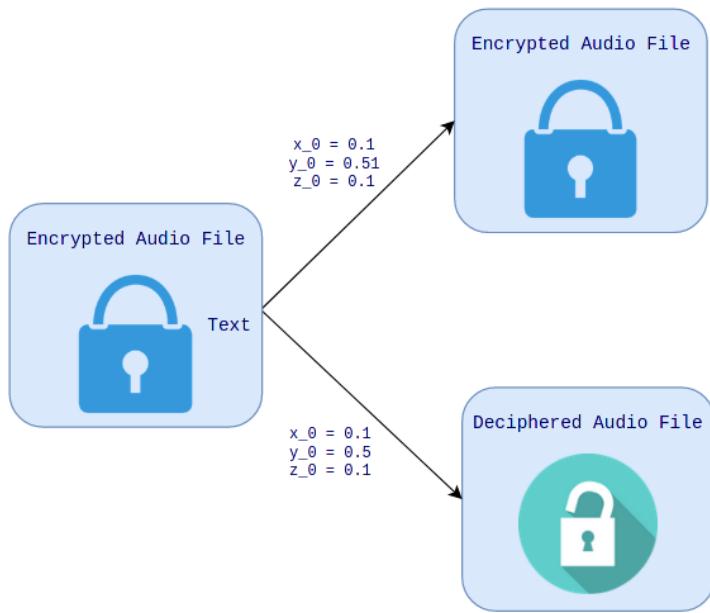


FIGURA 5.1: Descifrado con claves correctas e incorrectas . Fuente:
Elaboración propia.

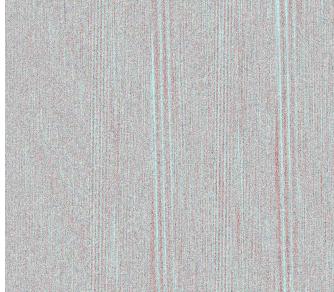
5.2. Métricas

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 en la imagen encriptada que resultó a partir del proceso de encriptación del audio de entrada. Utilizamos un audio de 30 segundos de la canción de Blinding Lights - The Weeknd.

Analizamos la correlación entre los pixeles adyacentes en forma horizontal, vertical y diagonal, en la imagen encriptada, previa a la conversión a audio. Se obtiene que la correlación entre pixeles tiende a desaparecer al aplicar el algoritmo de encriptación.

El cuadro 5.2 nos muestra que la imagen del audio, transformada a escala de grises, tiene un valor de coeficiente de correlación cercano a 1, lo que indica que los valores entre pixeles adyacentes son cercanamente idénticos; mientras que la imagen encriptada y transformada a escala de grises, tiene un valor cercano a cero, lo que indica que los pixeles adyacentes no guardan relación.

TABLA 5.1: Proceso de encriptación de sonido.

Audio Original	Conversión a Imagen
	
Encriptación	Conversión a Audio
	

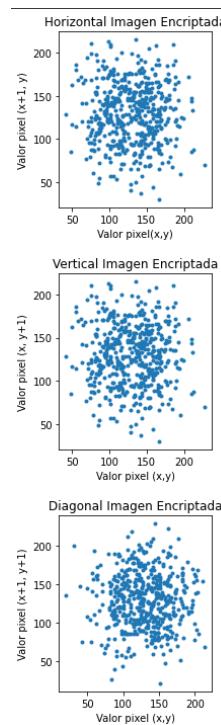


FIGURA 5.2: Análisis de correlación entre píxeles adyacentes.

TABLA 5.2: Valores de correlación de pixeles adyacentes.

Imagen	Horizontal	Vertical	Diagonal
Imagen original (audio)	0.98764	0.97489	0.98612
Imagen encriptada	0.00331	-0.01581	0.02512

El cuadro 5.3 nos muestra el valor de entropía de la imagen encriptada y transformada a escala de grises, este valor es cercano 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 5.3: Valores de entropía.

Imagen encriptada	Entropia
Blinding Lights (image)	7.6413

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 [34, 21, 15] se muestran en el cuadro 5.4; 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 5.4: Comparación de correlación de pixeles adyacentes de distintos algoritmos.

Imagen encriptada	Horizontal	Vertical	Diagonal
Algoritmo propuesto	-0.00159	0.01847	0.01904
Algoritmo de Khalid M. Hosny [34]	-0.0182	-0.0108	0.0165
Algoritmo de Sumit, Rajib y Bhaskar Panna. [21]	0.0944	0.0057	0.0067
Algoritmo de Zhongyun, Shuang y Yicong. [15]	0.0098	-0.0078	0.0181

El cuadro 5.5 muestra los tiempos de ejecución del algoritmo propuesto al encriptar la imagen del audio 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 podría tomar para encriptar muchos frames de un video.

TABLA 5.5: Tiempo de ejecución del algoritmo propuesto para audios de 30 segundos.

Cantidad de imágenes	Tiempo de ejecución
1	12.512 s
10	130.359 s
100	1305.786 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 Sistemas 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 Sistemas Caóticos optimiza los procesos de encriptación de sonidos 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 al sonido 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 un video. Por supuesto, que al aplicar este algoritmo de encriptación a videos, se debe sincronizar con otro algoritmo para encriptar los frames del video; por tal motivo, antes de desarrollar este algoritmo de encriptación de videos, se ha desarrollado un algoritmo de encriptación de imágenes basado igualmente en Sistemas Caóticos.

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 si surjen problemas al momento de desarrollar su propuesta, es normal, porque parte del desarrollo de la investigación, es encontrar errores y solucionarlos.

Bibliografía

- [1] Ekhlas Albahrani. «A new audio encryption algorithm based on chaotic block cipher». En: mar. de 2017, págs. 22-27. DOI: 10.1109/NTICT.2017.7976129.
- [2] ASQ. *Information Integrity*. URL: <http://asq.org/ii/about/understanding.html>.
- [3] Bastian Bechtold y Matthias Geier. *SoundFile*. URL: <https://pysoundfile.readthedocs.io/en/latest/>.
- [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] Erik de Castro Lopo. *Libsndfile*. URL: <http://www.mega-nerd.com/libsndfile/>.
- [6] Numpy Community. *Numpy*. URL: <https://numpy.org/>.
- [7] Python community. *Opencv-python*. URL: <https://pypi.org/project/opencv-python/>.
- [8] Center for Computer Research in Music y Acoustics. *Spectrograms*. URL: <https://ccrma.stanford.edu/~jos/st/Spectrograms.html>.
- [9] Comsol. *Rössler Attractor*. URL: <https://www.comsol.com/model/rossler-attractor-10656>.
- [10] Baptiste Coudurier. *FFmpeg*. URL: <http://ffmpeg.org/>.
- [11] ACDeS Digital. *Formatos de audio digital*. URL: <https://acdesdigital.org/formatos-de-audio-digital/>.
- [12] Editors of Encyclopedia Britannica. *Chaos theory*. URL: <https://www.britannica.com/science/chaos-theory>.

- [13] Python Software Foundation. *Python*. URL: <https://www.python.org/>.
- [14] Universidad de Granada. *Método de Runge-Kutta*. URL: <https://www.ugr.es/~lorente/APUNTESMNQ/cap23.pdf>.
- [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] hypr. *Confidentiality*. URL: <https://www.hypr.com/confidentiality/>.
- [17] 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.
- [18] 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>.
- [19] 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>.
- [20] Kaspersky. *Cryptography Definition*. URL: <https://www.kaspersky.com/resource-center/definitions/what-is-cryptography>.
- [21] 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.

- [22] George Lindfield y John Penny. *Solution of Differential Equations*. URL: <https://www.sciencedirect.com/topics/mathematics/lorenz-system>.
- [23] Nate Lord. *What Is Data Encryption? Definition, Best Practices More*. URL: <https://digitalguardian.com/blog/what-data-encryption>.
- [24] René Lozi. «Can we trust in numerical computations of chaotic solutions of dynamical systems?» En: 84 (feb. de 2012). DOI: 10.1142/9789814434867_0004.
- [25] Raman Chawla Mansy. «A Review in Audio Cryptography». En: *International Journal of Modern Communication Technologies and Research* 03 (jul. de 2015).
- [26] Wolfram Mathworld. *Rössler Attractor*. URL: <https://mathworld.wolfram.com/RoesslerAttractor.html>.
- [27] 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.
- [28] Pacific Northwest Seismic Network. *What is a Spectrogram?* URL: <https://pnsn.org/spectrograms/what-is-a-spectrogram>.
- [29] Facultad Regional San Nicolás. *Métodos de Runge Kutta*. URL: http://www.frsn.utn.edu.ar/gie/an/mnedo/34_rk.html.
- [30] Harsh Patel. *Idea to Hiding Audio, by it convert into Image.(Data hiding)*. URL: <https://medium.com/@harsh20111997/idea-to-hiding-audio-by-convert-into-image-e06265198dc6>.
- [31] Harsh Patel. *Steganography*. URL: <https://github.com/harsh2011/Steganography>.

- [32] Ivan Rodriguez. *Uso de histogramas en el análisis de datos en una auditoría*. URL: <https://www.auditool.org/blog/auditoria-externa/6446-uso-de-histogramas-en-el-analisis-de-datos>.
- [33] 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.
- [34] 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>.
- [35] W3 Schools. *NumPy Introduction*. URL: https://www.w3schools.com/python/numpy_numpy_intro.asp.
- [36] Search Security. *Cryptography*. URL: <https://searchsecurity.techtarget.com/definition/cryptography>.
- [37] William Stallings y Lawrie Brown. *Computer Security: Principles and Practice*. Pearson Education, 2018. ISBN: 9780134794396.
- [38] OpenCV team. *OpenCV*. URL: <https://opencv.org/>.
- [39] Smart Eye Technology. *Confidentiality, Integrity, Availability: Basics of Information Security*. URL: <https://getsmarteye.com/confidentiality-integrity-availability-basics-of-information-security/>.
- [40] Techopedia. *Digital Audio*. URL: <https://www.techopedia.com/definition/226/digital-audio>.
- [41] 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.

-
- [42] Xingyuan Wang y Yining Su. «An Audio Encryption Algorithm Based on DNA Coding and Chaotic System». En: *IEEE Access* 8 (2020), págs. 9260-9270. DOI: 10.1109/ACCESS.2019.2963329.
 - [43] Andriy Yurchuk. *ffmpy*. URL: <http://ffmpy.rtfd.io/>.