Criptografía Visual con Algoritmos Genéticos

1st Christian Camilo Pabon Useche
Ciencias de la Computación
Universidad Nacional
Bogotá Colombia
ccpabonu@unal.edu.co

2nd David Enrique Molina Rodriguez
Ciencias de la Computación
Universidad Nacional de Colombia
Bogotá Colombia
demolinar@unal.edu.co

***Abstract*—En la actualidad la seguridad de las imágenes digitales han generado interés por el uso constante y creciente de imágenes en el día a día, cada vez aumenta la cantidad de imágenes sensibles en bases de datos de todo tipo de empresas, bases de datos que pueden ser vulneradas perdiendo la información y privacidad de los usuarios. En este trabajo presentaremos un criptosistema que con ayuda de Algoritmos genéticos y la ecuación logística del caos son capaces de cifrar imágenes, también se analiza y compara con algunos otros criptosistemas visuales, donde la imagen mejor cifrada será la que tenga un coeficiente de correlación más bajo y una entropía más alta, de igual manera compararemos el tiempo que puede tomar cifrar una imagen con cada método.**

***Index Terms*—Criptografía Visual, Algoritmos Genéticos, Optimización, Caos, Python.**

I. Introducción

La primera mitad de 2021 se vio un aumento del 102% en delitos cibernéticos relacionados con ransomware en comparación con principios de 2020 [1], adicionalmente desde COVID-19, el FBI ha informado un aumento del 300% en los ataques cibernéticos y en la actualidad se espera que aumenten más los ataques a empresas y personas [2]. Gran parte de los ataques además de encriptar la información, la roban para venderla en mercado negro o sacar provecho de ella, luego entonces cuando se violan dichas capas de seguridad en las empresas, lo más común es que no exista una capa de seguridad sobre los datos ya sean estructurados o no estructurados, como las imágenes [3], y en personas es más fácil aún, las personas usualmente no disponen de más capas de seguridad más allá de las que ofrecen sus dispositivos y adicionalmente son más vulnerables a ataques de ingeniería social, por lo que adquirir información sensible en imágenes como fotos personales, de las tarjetas de crédito, documentos o incluso de la firma puede ser sencillo, es por ello que ha aumentado el interés por la criptografía visual, una capa más de seguridad que ofrece cifrar ese tipo de imágenes, también firmarlas para asegurar la autenticidad de la misma [4], o incluso generar imágenes con una entropía alta que al unirlas generen una imagen única firmada o viceversa [5].

En este documento nos centraremos en el cifrado de imágenes, que es el proceso donde dada una imagen reconocible y un algoritmo de cifrado es posible transformar dicha imagen para garantizar que no sea reconocible con respecto a la imagen clara, adicionalmente no importa que se conozca cómo funciona el algoritmo, un buen criptosistema aún debe ser seguro incluso si se conoce el funcionamiento completo del algoritmo. En nuestro criptosistema utilizamos una función caótica, más precisamente la ecuación logística para generar números pseudoaleatorios en secuencia de forma rápida y económica computacionalmente [6], luego con nuestros valores generados de forma pseudoaleatoria nuestro algoritmo genético inicia su objetivo por cifrar la imagen con una población inicial aleatoria representada por conjuntos de bits de la imagen en secuencia que juegan el papel de cromosomas, y donde nuestra función de aptitud es el coeficiente de correlación para calificar y decidir quienes serán los que produzcan nuevas generaciones combinándolos, adicionalmente elegimos aleatoriamente algunos cromosomas para que también produzcan nuevas generaciones con los que decidió la función de aptitud y de esta manera crear una nueva población a la cual mutar y realizar los mismos pasos para cada generación, esto hasta que nuestro criterio de detención se cumpla, todo esto lo explicaremos con detenimiento más adelante.

II. Coeficiente de correlación y entropía de una imagen

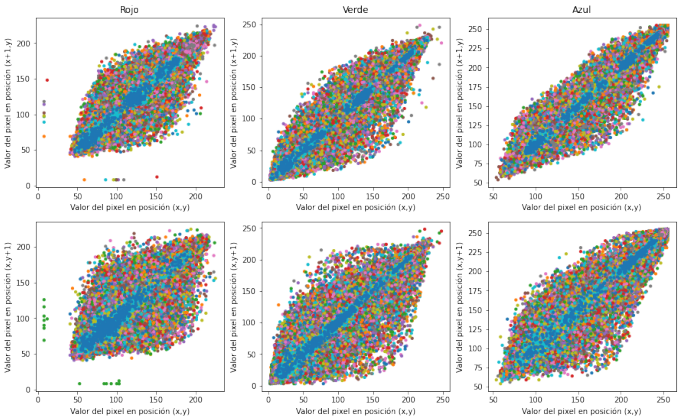

Fig. 1. Distribución de correlación

Un criptosistema efectivo debe ser robusto frente a cualquier ataque estadístico, por lo cual calculando la entropía y la correlación entre los píxeles vecinos en la fuente y en la imagen encriptada están los datos estadísticos para probar la

fortaleza del criptosistema propuesto contra cualquier ataque estadístico.

A. Coeficiente de correlación

Un buen algoritmo de cifrado, el coeficiente de correlación entre pares de píxeles adyacentes cifrados debe ser lo más pequeño posible [7], para el calculo se usa Eq.1:

$$r_{xy} = \frac{|\operatorname{cov}(x,y)|}{\sqrt{D(x)} \times \sqrt{D(y)}} \tag{1}$$

Donde la variable x es el valor en la componente de un pixel, y la componente y seria el pixel adyacente al de x, que depende del sentido ya sea vertical, horizontal o diagonal. Para calcular los coeficientes de correlación se emplean las siguientes ecuaciones:

$$\operatorname{Cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right) \tag{2}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i, -E(x)\right)^2. \tag{4}$$

B. Entropía de la imagen

La entropía en los píxeles la utilizaremos para evaluar la aleatoriedad de la imagen, la cual calcularemos con la Eq. 5:

$$H(S) = \sum_{i=0}^{2^N-1} P\left(s_i\right)\log\left(\frac{1}{P\left(s_i\right)}\right) \tag{5}$$

Donde N es el numero a s_i, $P\left(s_i\right)$ es la probabilidad que tiene el i-esima componente del píxel i que estemos calculando. Para una imagen de 8 bits, la entropía ideal de la imagen aleatoria es 8, se debe calcular la entropía de los componentes RGB de la imagen cifrada respectivamente para demostrar que el algoritmo propuesto puede resistir ataques de entropía.

III. Función Caótica

Las funciones caóticas son similares a las señales de ruido, sin embargo son completamente acertadas; es decir, si tenemos las condiciones iniciales es posible reproducir la señal exacta, o reproducir su comportamiento. [7].

A. Mapa logístico

Para nuestro proyecto utilizamos el mapa logístico, representado por la ecuación

$$X_{n+1} = rX_n\left(1 - X_n\right), \tag{6}$$

donde $X_n \in [0,1]$, $X_0 \geq 0$ y $r \in [0,4]$, sin embargo dependiendo donde tomemos r tenemos tres posibles situaciones:

1) Si $r \in [0,3]$, entonces la característica de la señal en las primeras 10 repeticiones muestra caos, y después de 10 repeticiones la señal es fija.
2) Si $r \in [3,3.57]$, entonces la característica de la señal en las primeras 20 repeticiones muestra caos, y después de 20 repeticiones la señal es fija.
3) Si $r \in [3,57,4]$, entonces la característica de la señal es completamente caótica. [6]

Es por lo anterior que aunque nuestro criptosistema como parte de la clave requiere r es aconsejable tomar $r \geq 3.7$, adicionalmente generaremos una cantidad determinada de puntos de acuerdo a la cantidad de épocas que se ejecuta nuestro algoritmo genético, de esta manera optimizamos el uso de memoria guardando en una pila aquellos valores pseudoaleatorios que utilizamos.

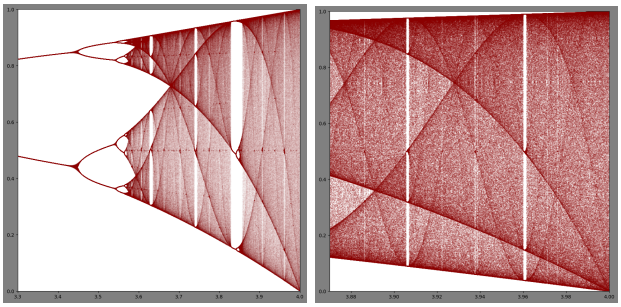

Fig. 2. Mapa logístico - Mapa logístico con dominio entre 3.85 y 4

Construir una lista de estos números puede resultar muy económico computacionalmente respecto al tiempo, sin embargo debemos tener cuidado con la memoria que usamos, ya que la generación de estos números puede tener una precisión alta con lo que debemos disminuir la precisión de estos números para optimizar el uso de memoria.

Generación de valores caóticos	
Cantidad	Tiempo en segundos
100	0.000195026
1.000	0.002964258
10.000	0.0052487850
100.000	0.0676186084
1′000.000	0.2904565334
10′000.000	3.7705308769

IV. Algoritmos genéticos

Los algoritmos genéticos son búsquedas heurísticas inspiradas en la teoría de la evolución, donde contamos con una población, cromosomas y genes, adicionalmente contamos con una función de aptitud que nos ayuda a optimizar de manera adecuada, y también con ciertas operaciones como mutación y crossover. En nuestro caso la población inicial es toda la imagen conformada por cromosomas que son escogido con un

tamaño adecuado o cantidad de genes de acuerdo a nuestras pruebas, donde no son de mayor tamaño que la mitad del ancho de la imagen multiplicada por tres (ya que por cada píxel contamos con tres canales diferentes de colores Red, Green y Blue), pero tampoco es menor a 4, ya que cada época podría tardar mucho tiempo, más adelante profundizaremos al respecto, adicionalmente cada gen cuenta con un valor entre 0 a 255 que es la amplitud de cada canal. La descendencia es posible gracias a nuestra operación crossover, la cual toma dos cromosomas y define un punto entre los genes para intercambiar las partes izquierdas y las partes derechas como se ve en la imagen 3. En nuestra nueva descendencia algunos genes pueden estar sujetos a mutar con cierta probabilidad aleatoria, esto para mantener la diversidad de colores en nuestros canales

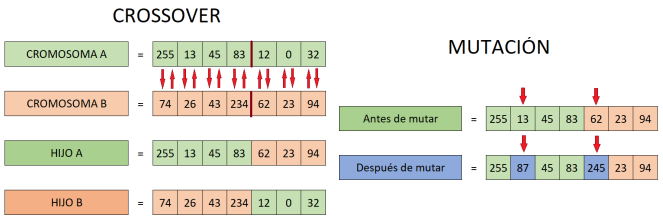

Fig. 3. Operación Crossover y Mutación

Finalmente es la función de aptitud la cual determina que tan buena forma tiene nuestro cromosoma, en nuestro caso, gracias una selección previa de los más aptos con respecto a al coeficiente de correlación de los colores.

V. Materiales y métodos

:

Una vez ya tenemos todos las particularidades entendidas vamos a unificarlas [8], para ello nos ayudaremos del diagrama 4 donde iniciamos con los argumentos que requiere nuestro programa, iniciamos con los argumentos que no son obligatorios pero que más sin embargo están a disposición para estudiar el criptosistema, estos argumentos son, la cantidad de épocas y la longitud de los cromosomas, por defecto son 70 y 64 respectivamente, sin embargo no hay un límite establecido por el programa, sin embargo hay limitaciones claras que son impuestas por el tamaño de la imagen, es decir, la longitud del cromosoma no puede ser más de la mitad del tamaño total de la imagen (Alto $\times$ Ancho $\times$ 3) ya que no había oportunidad de más de un cromosoma. también es configurable la cantidad de épocas, donde la mínima cantidad es una época. Los argumentos obligatorios son la imagen que puede ser de cualquier tamaño y de cualquier conjunto de colores y la siguiente variable r que representa la tasa de

$$X_{n+1} = rX_n\left(1 - X_n\right),$$

la ecuación de mapa logístico, además también representa nuestra llave, ya que es bastante sensible a cualquier cambio en sus decimales lo que nos puede garantizar un buen punto de partida para empezar a cifrar.

Una vez ingresados nuestros parámetros de manera exitosa procedemos al cálculo del arreglo con el valores pseudoaleatorios pero trazables gracias a nuestra llave r, este arreglo es de una longitud necesaria para cubrir la cantidad de iteraciones dadas por nuestros parámetros. Ahora procedemos a la creación de la población inicial, que por defecto es de un valor cercano a 64 por su rápida convergencia a una coeficiente de correlación bajo, decimos cercano porque es posible que por el tamaño de la imagen 64 no sea divisor del tamaño total, sin embargo muy seguramente se encuentren divisores cercanos a este número, aunque también esta la posibilidad de escoger la longitud con el argumento no obligatorio. Luego ingresamos al bucle de las épocas, el cuál es el encargado de producir las épocas establecidas por el usuario o las establecidas por el mismo proceso, donde ejecuta internamente el bucle de la población que representa recorrer la imagen clasificando buenos candidatos, realizando crossover y ejecutando mutaciones sobre descendientes.

vspace0.5cm

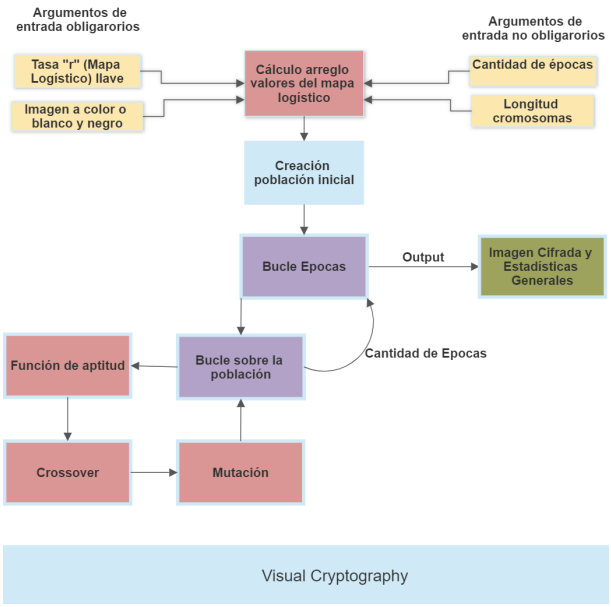

Fig. 4. Diagrama de Bloques

Cuando termina de recorrer la imagen la cantidad de veces representada por la época nuestro arreglo es transformado en la imagen cifrada, adicionalmente se entrega al usuario el coeficiente de correlación y el valor de entropía, con lo que culminaría su ejecución.

VI. Resultados

A continuación se muestra en la Fig.5, la imagen que se utilizó para ejemplificar el funcionamiento del criptosistema visual.

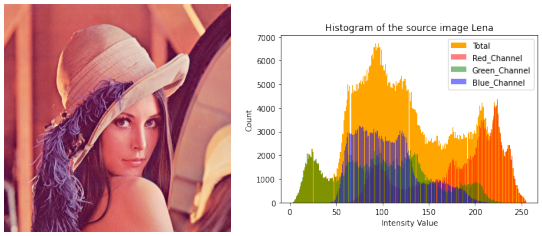

Fig. 5. Lenna - Histograma canales RGB.

Después de usar el algoritmo genético con cromosomas de tamaño 64 y una tasa de cambio de mapa logístico de 15, (En el notebook hay mas ejemplos con diferentes valores) da como resultado la imagen encriptada a continuación:

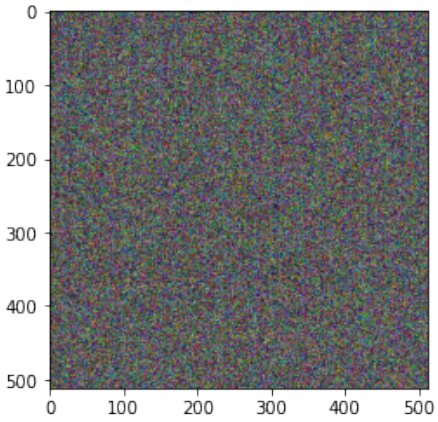

Fig. 6. Imagen Encriptada

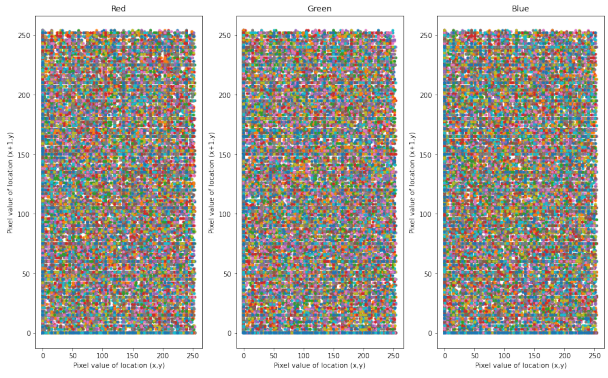

Fig. 7. Correlación vertical de las componentes de los pixeles

En la Fig.7 se puede ver que la correlación entre las componentes de los pixeles adyacentes verticalmente es muy dispersa, si se compara con la Fig.1 que es la correspondiente a la imagen original, se puede ver comportamientos totalmente diferentes. A continuacion vemos estos valores :

- Correlacion componente Rojo: -0.001268277
- Correlacion componente Verde: 0.002579044
- Correlacion componente Azul: 0.000250341

Imagen	Longitud Cromosoma	r	Correlación Rojo	Correlación Verde	Correlación Blue	Tiempo (segundos)
A	64	15	0,01103	0,01175	0,01518	12,33869
B	64	70	-0,00126	0,00257	0,00025	68,06606
C	64	140	0,00042	0,00092	0,00065	124,24073
D	768	15	0,5095	0,4956	0,4956	2,80005
E	768	70	0,08587	0,09121	0,08959	13,73939
F	768	140	0,01855	0,01705	0,01608	28,97851

Fig. 8. Imagen Encriptada

A.

VII. Discusión y conclusiones

El actual algoritmo es susceptible a muchas mejoras, sobretodo en la cantidad de tiempo que toma realizar estos cifrados, si bien no es el que más tarda con respecto a otros criptosistemas visuales en los que ya hemos trabajado anteriormente [8], [9] pero sí tarda más que el promedio, sin embargo como ya vimos anteriormente no es mucho respecto a otros trabajos los cuáles usan imágenes de un sólo canal, a escala de grises, ya que manejar una imagen a color implica tres veces el trabajo de una en escala de grises. Vale la pena seguir estudiando criptosistemas visuales y también utilizar herramientas de inteligencia artificial para hacer criptoanálisis a varios cifrados existentes, como captcha, firmas digitales, unión y separación de imágenes, ya que puede ser una buena alternativa para romper criptosistemas que posiblemente pueden estar siendo vulnerados actualmente.

REFERENCES

[1] Norton, Emerging threats (2021).
URL https://us.norton.com/blog/emerging-threats/cybersecurity-statistics#

[2] M. Miller, Fbi sees spike in cyber crime reports during coronavirus pandemic (2021).
URL https://thehill.com/policy/cybersecurity/493198-fbi-sees-spike-in-cyber-crime-reports-during-coronavirus-pandemic/

[3] IBM, Structured vs. unstructured data: What's the difference? (2022).
URL https://www.ibm.com/cloud/blog/structured-vs-unstructured-data

[4] M. H. Abood, An efficient image cryptography using hash-lsb steganography with rc4 and pixel shuffling encryption algorithms, researchgate 1 (1) (2017).

[5] M. A. O. A. Agustin Moreno, Brauer configuration algebras for multimedia based cryptography and security applications, Springer 1 (1) (2021).

[6] D. Brockmann, The logistic map, the mother of deterministic chaos (2022).
URL https://www.complexity-explorables.org/flongs/logistic/

[7] A. HananAbdullaha, R. Enayatifa, M. Lee, A hybrid genetic algorithm and chaotic function model for image encryption (Feb 2012).
URL https://www.sciencedirect.com/science/article/abs/pii/S1434841112000313?via3Dihub

[8] D. E. Molina, Herramientas criptográficas avanzadas - unalcryptomato (2022).
URL http://cryptomato-unal.s3-website-us-east-1.amazonaws.com/pages/crpvisual/visual

[9] D. E. Molina, Herramientas critograficas básicas - unalcryptotools (2021).
URL http://unalcryptotools.000webhostapp.com/pages/clasicos/hill