UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA

MEJORAMIENTO DE MAPAS CAÓTICOS DE 1D PARA APLICACIONES DE INTERNET DE LAS COSAS

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

DOCTOR EN CIENCIAS

presenta:

DIEGO ARMANDO TRUJILLO TOLEDO

Directores de tesis

Dr. Everardo Inzunza González Dr. Oscar Roberto López Bonilla

Ensenada, Baja California, México. Septiembre de 2022.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA

MEJORAMIENTO DE MAPAS CAÓTICOS DE 1D PARA APLICACIONES DE INTERNET DE LAS COSAS

TESIS

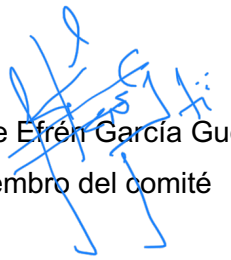
que para obtener el grado de DOCTOR en CIENCIAS presenta:

Diego Armando Trujillo Toledo

Aprobada por:

Dr. Everardo Inzunza González
Director de tesis

Dr. Oscar Roberto López Bonilla
Co-director de tesis



Dr. Enrique Efrén García Guerrero
Miembro del comité

Dr. José Ricardo Cárdenas Valdez
Miembro del comité

Dr. Esteban Tlelo Cuautle
Miembro del comité

Ensenada, Baja California, México. Septiembre de 2022.

RESUMEN de la tesis de Diego Armando Trujillo Toledo, presentada como requisito parcial para obtener el grado de DOCTOR EN CIENCIAS del programa de Maestría y Doctorado en Ciencias e Ingeniería de la UABC. Ensenada, B. C. México, septiembre de 2022.

Mejoramiento de mapas caóticos de 1D para aplicaciones de internet de las cosas

Resumen aprobado por:

Dr. Everardo Inzunza González
Director de tesis

Dr. Oscar Roberto López Bonilla
Co-director de tesis

En este trabajo de tesis se utilizan cuatro mapas caóticos como caso de estudio para diseñar un criptosistema embebido basado en un generador de números pseudoaleatorios (por sus siglas en inglés PRNG). La aleatoriedad de las secuencias se mejora aplicando la función $mod\ 1023$ y se verifica analizando los diagramas de bifurcación, el máximo exponente de Lyapunov y realizando las pruebas estadísticas NIST SP 800-22 y TestU01. El PRNG se aplica en un algoritmo sencillo para el cifrado en tiempo real de imágenes RGB en un esquema de máquina a máquina (M2M), utilizando el protocolo de transporte de telemetría de cola de mensajes (por sus siglas en inglés MQTT) a través de una red Wi-Fi y la Internet. El criptoanálisis confirma que el esquema propuesto de encriptamiento de imágenes es robusto para resistir la mayoría de los ataques existentes, como los histogramas estadísticos, la entropía, el espacio clave, la correlación de píxeles adyacentes y los ataques diferenciales. La implementación del criptosistema propuesto se realiza utilizando secuencias mejoradas del mapeo Logistic 1D, y alcanza un rendimiento de hasta 47.44 Mbit/s utilizando un ordenador personal con un reloj de 2.9 GHz, y 10.53 Mbit/s utilizando una Raspberry Pi 4. Como resultado, el criptosistema embebido propuesto es adecuado para aumentar la seguridad en la transmisión de imágenes RGB en tiempo real a través de redes Wi-Fi e Internet.

Palabras clave: Encriptamiento de imágenes; Mapa caótico; IoT; M2M; MQTT; PRNG.

ABSTRACT of the thesis of Diego Armando Trujillo Toledo, presented as requirement to obtain the degree of DOCTOR IN SCIENCES, of the Master and Doctorate program in Science and Engineering of the Universidad Autónoma de Baja California, Ensenada, B. C. México, September 2022.

Enhancement of 1D chaotic maps for internet of things applications

Abstract approved by:

Dr. Everardo Inzunza González
Thesis Director

Dr. Oscar Roberto López Bonilla
Thesis Co-director

In this thesis work, four chaotic maps are used as a case study to design an embedded cryptosystem based on a pseudorandom number generator (PRNG). The randomness of the sequences is improved by applying the $mod\ 1023$ function and verified by analysing the bifurcation diagrams, the maximum Lyapunov exponent and by performing the statistical tests NIST SP 800-22 and TestU01. The pseudorandom number generator is applied in a simple algorithm for real-time RGB image encryption in a machine-to-machine (M2M) scheme, using the message queue telemetry transport protocol (MQTT) over the Wi-Fi network and the Internet. The cryptanalysis confirms that the proposed image encryption scheme is robust to resist most existing attacks, such as statistical histograms, entropy, key space, adjacent pixel correlation and differential attacks. The implementation of the proposed cryptosystem is realized using enhanced Logistic 1D map sequences, and achieves a throughput of up to 47.44 Mbit/s using a personal computer with a 2.9 GHz clock, and 10.53 Mbit/s using a Raspberry Pi 4. As a result, the proposed embedded cryptosystem is suitable for increasing the security of real-time RGB image transmission over Wi-Fi networks and the Internet.

Keywords: Chaotic map; Image encryption; IoT; M2M; MQTT; PRNG.

A mi familia

Agradecimientos

*Quiero agradecer a **Dios**, por permitirme contar con salud y tenerme bajo su protección y cuidado a lo largo de todos los días de mi vida.*

*También quiero agradecer a la memoria de mi papá **José Luis Trujillo Meza**, por darme la vida, y que siempre lucho hasta sus últimos días por enseñarme el buen camino y nunca rendirme, gracias papá hasta el cielo.*

*A mi mamá, **Teresa de Jesús Toledo Torres**, por darme la vida, por su valioso apoyo en momentos difíciles y aunque a la distancia estos últimos años sus consejos y palabras siempre son atinados, te quiero mucho mamá.*

*A mis hermanos, **José Luis y Teresita**, por todos sus consejos y experiencias de vida que hemos compartido juntos. Los quiero mucho.*

*A mi compañera de vida **Verónica Medina Corral**, por toda la paciencia y motivación que lograron darme fuerzas para terminar este trabajo de tesis doctoral.*

*A mi hijo, **Diego Armando Kalel Trujillo Medina**, por toda su paciencia y comprensión que a pesar de ser un niño entendió que su padre también estaba estudiando.*

*A toda la familia **Trujillo** y **Toledo**, en especial a la memoria de mis Tíos y Tías que fallecieron causas del virus COVID-19, quienes en su momento me estuvieron guiando por el buen camino.*

*Al **Dr. Everardo Inzunza González**, por haberme dirigido en este camino tan complejo e interesante. Por todos los consejos y sobre todo la paciencia que permitieron la culminación de este trabajo de investigación, por su amistad y apoyo incondicional, mil gracias.*

*Al **Dr. Oscar López Bonilla**, también por guiarme en este camino del doctorado. Por sus consejos y comentarios acertados durante todo el proceso, por su amistad y apoyo.*

Agradecimientos

*Al **Dr. Enrique Efrén García Guerrero**, por sus valiosos consejos y comentarios durante todo el proceso, por su gran amistad y apoyo incondicional.*

*Al **Dr. José Ricardo Cárdenas Valdez**, primero como compañero de maestría y ahora como parte importante en la colaboración de este trabajo de tesis.*

*Al **Dr. Esteban Tlelo Cuautle**, un especial agradecimiento por su colaboración en el proceso de este trabajo de tesis de doctorado.*

*A nuestra alma mater: **Universidad Autónoma de Baja California**, por el apoyo que me dio para poder realizarme profesionalmente.*

*A todos los **profesores e investigadores** del MYDCI, que compartieron conmigo una parte de sus conocimientos y experiencias a lo largo de todos los cursos.*

*A mis compañeros **Paola, Cecilia, Oscar Adrián y Francisco**.*

*A todos mis **amigos**, **colegas**, **administrativos**, **secretarias**, **personal de apoyo**, **coordinadores** y **demás**, que con su apoyo, consejo y confianza fue posible llegar a esta meta.*

*Y por último, pero no menos importante si no lo contrario, a **José Jaime Esqueda Elizondo**, gracias por sus consejos y apoyo incondicional, desde que coincidimos en un evento en Chihuahua como estudiante de maestría, meses después en la FCQI como compañeros de trabajo y ahora como principal motivador en realizar este proyecto de tesis y compañeros de este viaje del doctorado, mil gracias.*

Ensenada, B. C. México
6 de septiembre de 2022

DIEGO ARMANDO TRUJILLO TOLEDO

Tabla de Contenido

Lista de Figuras

Página

Lista de Figuras

Lista de Figuras

Lista de Figuras

Lista de Tablas

Página

Capítulo I. Introducción

Hoy en día, el Internet de las cosas (por sus siglas en inglés IoT) se utiliza ampliamente en aplicaciones como la industria, la agricultura, la ciber-salud, las ciudades inteligentes y la domótica. Según el informe de Ericsson, en 2021 habrá unos 28 billones de dispositivos inteligentes conectados en todo el mundo. La comunicación máquina a máquina (M2M) se emplea en más de 15 billones de dispositivos [1]. Un informe de Cisco pronosticó que para 2030, aproximadamente 500 billones de dispositivos estarán asociados a Internet [2]. De este modo, se puede apreciar que el IoT ha atraído la atención de investigadores y desarrolladores, ya que ha introducido cambios revolucionarios en la vida humana. El IoT permite el intercambio de datos multimedia en una amplia variedad de aplicaciones, como los edificios inteligentes, la salud inteligente, el transporte inteligente y la Industria 4.0, por mencionar algunas [3]. Debido a que miles de millones de dispositivos interconectados se comunican entre sí e intercambian datos potencialmente sensibles, la cuestión más crítica en el IoT es la seguridad de los datos [4] y la privacidad [5]. En el campo del procesamiento de imágenes en tiempo real, la superresolución de imágenes y el cifrado son importantes puntos de investigación [6], [7]. En [8] se utiliza el modelo de reflexión de la red neuronal wavelet para reconstruir la imagen característica de un solo cuadro y mejorar la resolución de la adquisición de imágenes de una aplicación del IoT.

En la actualidad, la seguridad y la confidencialidad de las redes mundiales de telecomunicaciones se basan en criptosistemas clásicos como el Estándar de Cifrado de Datos (DES) [9], el Estándar de Cifrado Avanzado (AES) [10], el Estándar de Cifrado de Datos Triple (3DES) [11], el Algoritmo Internacional de Cifrado de Datos (IDEA) [9] y el Cifrado Rivest 4 (RC4) [4]. Sin embargo, aunque estos algoritmos son excelentes para el cifrado de texto [12], no son adecuados para cifrar

imágenes digitales de forma segura y eficiente [13], debido a sus características distintivas, que incluyen la capacidad de datos a gran escala, la fuerte correlación de píxeles adyacentes y la alta redundancia entre los píxeles en bruto, [14]. Estos problemas hacen que los algoritmos clásicos no sean adecuados para el cifrado de imágenes en tiempo real, debido a que estos cifrados necesitan una cantidad significativa de tiempo y energía de cálculo [15]. Por lo tanto, la criptografía basada en el caos se considera uno de los métodos más seguros para proteger los datos confidenciales [16], [17], gracias a los beneficios de sus propiedades únicas, que incluyen una alta sensibilidad y dependencia de las condiciones iniciales, un comportamiento impredecible, la complejidad de la topología, la ergodicidad, la aleatoriedad y una excelente adaptabilidad a la telecomunicación segura [18], [19]. Además, se descubrió que la criptografía basada en el caos era adecuada para el cifrado de imágenes y vídeos, donde las técnicas de criptografía tradicionales (DES, IDEA y AES) fallaban [5]. En efecto, la comunicación a través de un canal no protegido requiere la transmisión de una clave criptográfica entre el emisor y el receptor a través de un canal seguro; sin embargo, un espía potencial (hacker/intruso) intentará obtener esta clave secreta.

En los últimos años se han publicado varios ataques y criptoanálisis a criptosistemas [20]. A medida que los hackers e intrusos se vuelven más sofisticados, la tecnología de cifrado debe evolucionar también. Como resultado, se requieren nuevos criptosistemas con mayor eficiencia y complejidad [16], de los cuales, los mapas caóticos son buenos candidatos debido a la estrecha relación entre las propiedades del caos como la ergodicidad, la sensibilidad a las condiciones iniciales y los parámetros de control, la imprevisibilidad, el comportamiento aleatorio, y las propiedades de un buen cifrado como la sensibilidad al texto plano, la clave y la aleatoriedad en los procesos de confusión y difusión [18], [21], [22], [23]. Por lo tanto, los criptosistemas basados en el caos pueden considerarse métodos seguros para proteger datos confidenciales [24], [25].

El circuito Chua se inventó en el otoño de 1983 en respuesta a dos

búsquedas insatisfechas entre muchos investigadores sobre el caos en relación con dos aspectos que querían de las ecuaciones de Lorenz [26]. Además, en 1993, el profesor L.O. Chua propuso un circuito universal para estudiar y generar el caos [27]. En la actualidad, existen muchos circuitos con comportamientos caóticos, entre los cuales el más sencillo es el circuito de Chua [28]. Estos modelos se implementaron primero en circuitos analógicos para generar dinámicas caóticas y estudiar su comportamiento. En la actualidad, la implementación de hardware de circuitos caóticos o generadores de números pseudoaleatorios (PRNG) adopta principalmente métodos analógicos y digitales. Existen dos métodos para desarrollar PRNG en circuitos analógicos: el diseño de circuitos de componentes discretos [29], [30], [31], [32], [33] y el diseño de circuitos integrados en chip [34], [35], [36], [37]. Dado que los parámetros de los componentes discretos se ven afectados por factores ambientales como la temperatura y el desgaste por el tiempo, el sistema caótico es extremadamente sensible a las condiciones iniciales (por ejemplo, la energía inicial acumulada en inductores y condensadores). Por lo tanto, el efecto de utilizar un circuito de componentes discretos para desarrollar un PRNG analógico es muy limitado. El método de diseño de los circuitos analógicos integrados tiene las características de tamaño de chip pequeño y de fácil integración. Sin embargo, tiene las desventajas de un largo ciclo de diseño, una difícil sintonización y un alto coste. Sin embargo, los circuitos digitales no tienen problemas de temperatura y desgaste del tiempo del dispositivo; por lo tanto, los parámetros del dispositivo no afectarán a sus resultados. En consecuencia, el efecto de implementación es mejor que el de los circuitos analógicos, por lo que el uso de dispositivos digitales para desarrollar PRNG se ha convertido en la corriente principal en esta etapa [28]. Además, dado que las redes de telecomunicaciones actuales, como Internet, se basan en sistemas digitales, los PRNG digitales son los más adecuados para integrarlos en los nuevos protocolos de comunicación [38].

Recientemente, se han propuesto varios PRNG para aplicaciones criptográficas [39], [40], [41], [42], [43], [44], [45], y se han implementado en sistemas embebidos utilizando señales caóticas, que han reportado buenos niveles

de seguridad contra varios ataques [46], [47], [48], [49]. Sin embargo, su compatibilidad y rendimiento con los nuevos protocolos para el IoT, como MQTT, aún no han sido probados y verificados. MQTT es uno de los protocolos de mensajería más populares en el mundo del IoT, diseñado para dispositivos embebidos y comunicaciones M2M, basado en el modelo publish-subscribe y que ofrece una autenticación básica mediante nombre de usuario y contraseña. Sin embargo, este método de autenticación podría tener un problema en términos de seguridad y escalabilidad [50]. MQTT es adecuado para la computación en la nube, para lo cual utiliza un broker centralizado para recoger y transmitir datos [51], y recomienda encarecidamente el uso de los protocolos de capa de sockets seguros (SSL) y de seguridad de la capa de transporte (TLS) en los servidores (broker MQTT) [52]. Actualmente, algunos autores [53] muestran el cifrado mediante el uso de la sincronización del caos en los sistemas embebidos y su aplicación a la seguridad de MQTT [53], [54], [55]. Sin embargo, de acuerdo con las recomendaciones de seguridad del estándar MQTT, es probable que las implementaciones tengan que seguir el paso de un entorno de seguridad en desarrollo [52].

Para desarrollar una aplicación de IoT, se pueden aprovechar los sistemas embebidos como Raspberry Pi, que es muy útil para resolver problemas del mundo real en varios campos de aplicación [56], [57], [58]. De acuerdo con nuestra investigación, sólo tres artículos discuten el uso de dispositivos embebidos para el cifrado de imágenes y su aplicación al IoT a través del protocolo MQTT [53], [54], [55]. Sin embargo, aún quedan problemas abiertos por resolver, como el desarrollo de nuevos métodos y algoritmos para mejorar la seguridad, el rendimiento y la eficiencia. De este modo, este trabajo propone integrar un método de criptografía basado en el caos como seguridad adicional para proteger los datos confidenciales en imágenes RGB de extremo a extremo. Se presenta un algoritmo simple y eficiente para el cifrado de imágenes RGB mediante el uso de secuencias caóticas mejoradas implementadas en una Raspberry Pi 4 (RPi4). La RPi4 es útil para la investigación y el desarrollo de un criptosistema embebido de bajo coste, portátil y

seguro. Se puede codificar utilizando software de código abierto [59]; además, tiene un buen rendimiento y ofrece capacidades de multiprocesamiento y tareas de fácil programación, lo que permite un rápido desarrollo tecnológico [47].

La novedad de este trabajo de tesis doctoral en el área de criptografía es la introducción de un algoritmo simple, seguro y eficiente para mejorar la aleatoriedad de los mapas caóticos 1D para el cifrado de imágenes en tiempo real. Además, en las aplicaciones de IoT, el algoritmo propuesto es factible de implementar en dispositivos de telecomunicaciones modernos que emplean multiprocesadores, por ejemplo, smartphones, tabletas y cualquier dispositivo de IoT con capacidad de procesamiento de imágenes. Por ejemplo, en los mensajeros privados de las aplicaciones de redes sociales para enviar imágenes privadas, también se puede aplicar en los sistemas de vigilancia para enviar imágenes de alerta, finalmente en los procesos industriales que se monitorizan remotamente a través de internet. Además, el criptosistema embebido propuesto es portátil, fiable, de bajo coste y puede utilizarse en esquemas M2M que funcionen en tiempo real. Por otro lado, la complejidad de las secuencias caóticas mejoradas se verifica mediante el análisis de los diagramas de bifurcación, el cálculo del exponente máximo de Lyapunov y la realización de pruebas estadísticas NIST y TestU01. Asimismo, el análisis de seguridad demuestra que el criptosistema embebido propuesto es robusto y seguro frente a una variedad de ataques bien conocidos, incluyendo histogramas estadísticos, espacio-clave, sensibilidad, correlación, entropía y ataques diferenciales, demostrando cómo el cifrado caótico contribuye a la seguridad de la transferencia de datos a través de redes Wi-Fi e Internet.

Los siguientes Capítulos están organizados de la siguiente manera: El Capítulo 2 presenta los mapas caóticos 1D para el cifrado de imágenes en tiempo real, se describe el algoritmo para mejorar la aleatoriedad de las secuencias caóticas e ilustra sus diagramas de bifurcación y exponentes de Lyapunov. El Capítulo 3 describe el esquema propuesto para las aplicaciones de IoT a través de la red WiFi e Internet. El Capítulo 4 contiene los resultados del criptoanálisis, la evaluación del rendimiento, la comparación con el estado del arte y el desarrollo de la implementación. El Capítulo 5 resume las conclusiones y trabajos a futuro.

Capítulo

II. Mapas caóticos para el encriptado de imágenes en tiempo real

Los PRNG de alta velocidad se pueden lograr con mapas caóticos unidimensionales (1D) [23], [24] ya que tienen: naturaleza discreta, buena complejidad en su comportamiento dinámico, baja sobrecarga computacional [25], bajas operaciones aritméticas, mejor rendimiento para la operación en tiempo real [60], entre otras ventajas en comparación con los sistemas caóticos de mayores dimensiones como 2D [61], 3D [18], 4D [62] y 5D [63], [64]. De esta manera, en este Capítulo se presentan los cuatro mapas caóticos para desarrollar los PRNG mejorados como elemento importante en el criptosistema embebido propuesto, los cuales son: el mapa de desplazamiento Bernoulli [49], el mapa Tent [49], Zigzag [49] y Mapa logístico 1D [22]. Todos estos mapas caóticos comparten las características de un único parámetro de control, un único exponente de Lyapunov (LE), lo que los hace más fáciles de analizar y se ha informado de que son adecuados para aplicaciones de cifrado en tiempo real [49]. Estos mapas caóticos tienen un buen rendimiento y excelentes propiedades como la ergodicidad, el comportamiento aleatorio, la imprevisibilidad y otras propiedades de un buen PRNG [12], [22], [23], [49].

II.1 Mapa de Desplazamiento Bernoulli

El mapa de desplazamiento de Bernoulli [49] se define por la ecuación (1), que tiene dos funciones lineales como:

$$x_{n+1} = \begin{cases} b_{x_n} - a, & x_n \geq 0 \\ b_{x_n} + a, & x_n < 0 \end{cases} \quad (1)$$

La ecuación (1) también puede ser escrita como $x_{n+1} = bx_n - \text{asin}\,(x_n)$, donde b es el parámetro que controla las propiedades estocásticas del sistema caótico, y a es un factor escalar que simplemente incrementa o decrementa el producto de b_{x_n}, y los valores de salida están limitados en el rango $[-a, a]$. El estado unidimensional del mapa caótico es x_{n+1} con valores $\in [-1,1]$; la dinámica depende de dos parámetros, $a = 1$ y $b = 1.999$, con una condición inicial de $x_0 = 0.5$ para generar una condición caotica. Se realiza el diagrama de bifurcación para el valor de $a = 1$, el resultado se muestra en la Figura 1.

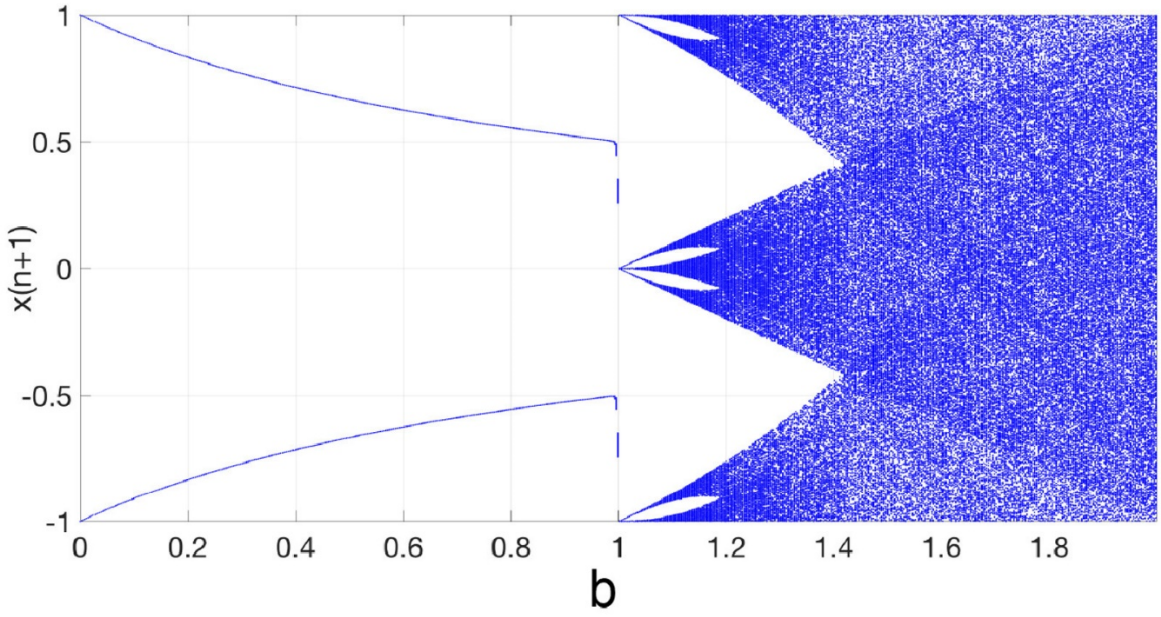


Figura 1. Diagrama de bifurcación Bernoulli $a = 1, b \in [0,2]$.

En la Figura 1 se muestra, el diagrama de bifurcación para el valor de $a = 1$, donde se pueden observar 4 regiones principales, la primera comprendida para los intervalos de los valores de b entre 0 y 1, el mapa oscila entre dos puntos estables bien definidos. La segunda sección se observar cuando $b = 1$ el sistema es

inestable, para los valores de $b$entre 1 y 1.4 hay una dispersión no uniforme en los valores de la salida del sistema y finalmente la cuarta región para los intervalos de valores de 1.4 a 2.0 se produce la mayor dispersión en la salida donde todos los valores para este rango cubren desde -1 hasta 1.

II.2 Mapa de Tent

El mapa de Tent [49] está descrito por la ecuación (2):

$$x_{n+1} = \begin{cases} u_{x_n}, & x_n \in \left[0, \frac{1}{u}\right] \\ \frac{u}{u-1}(1 - x_n), & x_n \in \left[\frac{1}{u}, 1\right] \end{cases} \quad (2)$$

Donde x_{n+1} es el estado unidimensional del mapa Tent con los valores $\in (0, 1]$, el parámetro $u = 1.999$, y la condición inicial $x_0 = 0.01$. En la Figura 2 se muestra el diagrama de bifurcación, en el cual se puede observar los valores que toma la variable u para que la salida del sistema sea caótica.

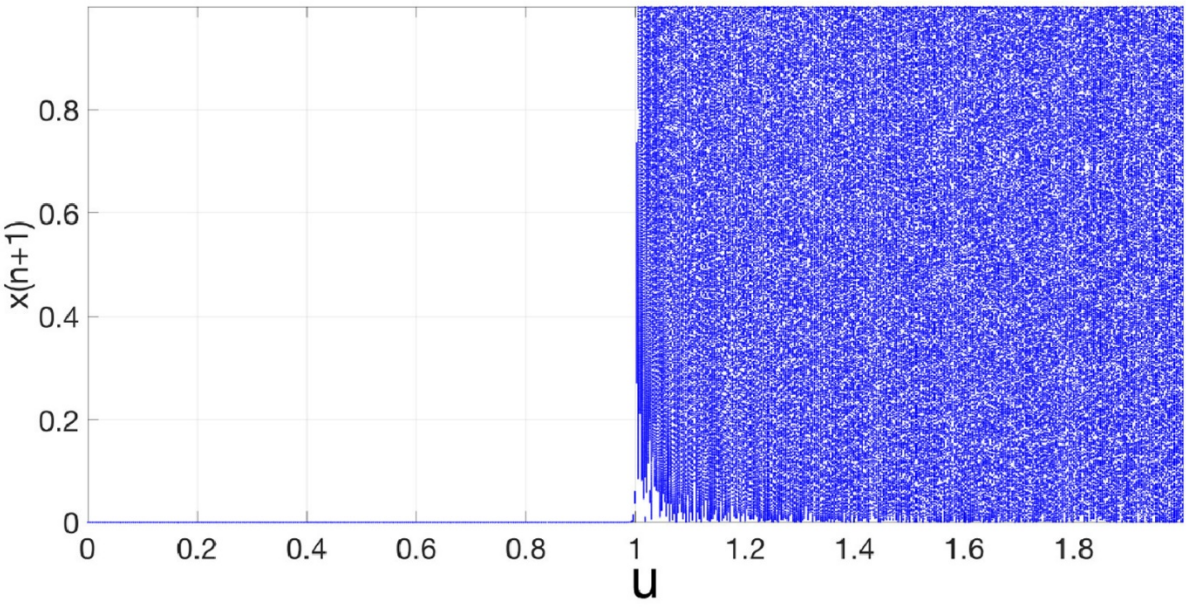


Figura 2. Diagrama de bifurcación Tent $u \in [0,2]$.

II.3 Mapa Zigzag

El mapa Zigzag [49] se define por la ecuación (3):

$$x_{n+1} = \begin{cases} -m\left(x_n + \frac{2}{|m|}\right) & , \quad x_n \in (-1, -\frac{1}{|m|}] \\ mx_n & , \quad x_n \in \left(-\frac{1}{|m|}, \frac{1}{|m|}\right] \\ -m\left(x_n - \frac{2}{|m|}\right) & , \quad x_n \in \left(\frac{1}{|m|}, 1\right] \end{cases} \tag{3}$$

donde x_{n+1} es el estado unidimensional del mapa Zigzag con valores $\in [-1, 1]$, el parámetro $m = 2.999$, y la condición inicial $x_0 = 0.01$.

El diagrama de bifurcación para este mapa zigzag se muestra en la Figura 3. Se puede observar que para valores de $m < 1$ el comportamiento en la salida del

sistema no es caótico, mientras que para los intervalos (1,2) y (2,3] el comportamiento es totalmente caótico, aunque se puede observar que, para el último intervalo, la salida abarca completamente los rangos de [-1,1].

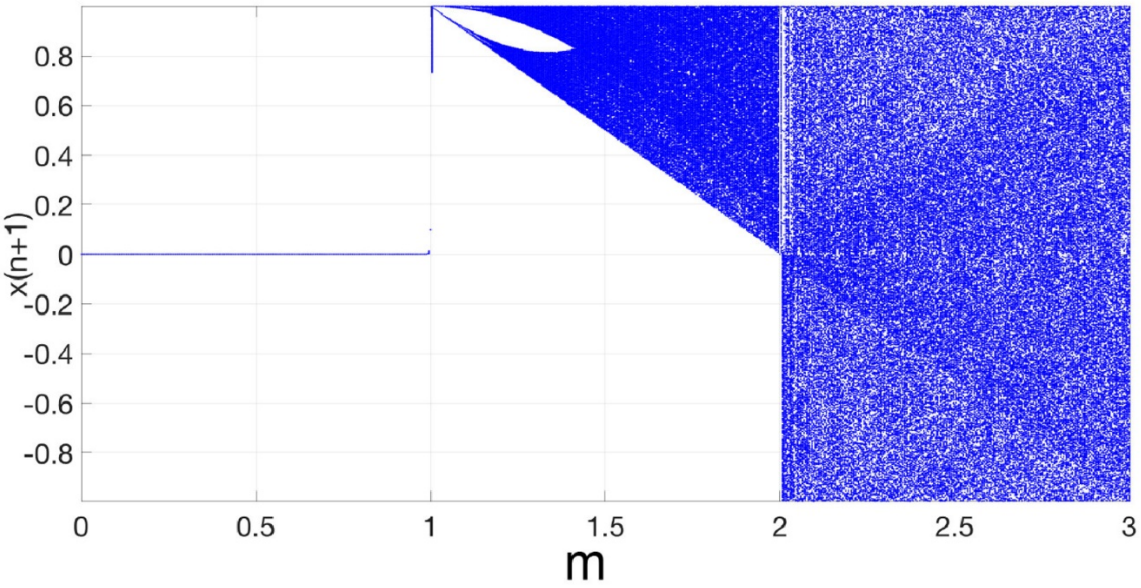


Figura 3. Diagrama de bifurcación del mapa Zigzag cuando el parámetro $m \in [0,3]$.

II.4 Mapa Logístico 1D

El mapa Logístico 1D [22] se define por la ecuación (4):

$$x_{n+1} = r x_n (1 - x_n) \tag{4}$$

donde r es el parámetro de control con el rango de $r \in [0,1]$, $x_0 = 0.1$ es el valor inicial del mapa Logístico 1D, x_n y x_{n+1}son la entrada y la salida respectivamente de la $(n+1)^{a}$ iteración del mapa Logístico 1D con valores $r \in (0,1]$. El diagrama de bifurcación se muestra en la Figura 4, se puede observar el comportamiento del sistema para los valores de r; cuando el sistema es uniforme con un solo valor en su salida, para posteriormente su complejidad va incrementando hasta que a partir

del valor de $r \cong 3.6$ la salida es caótica.

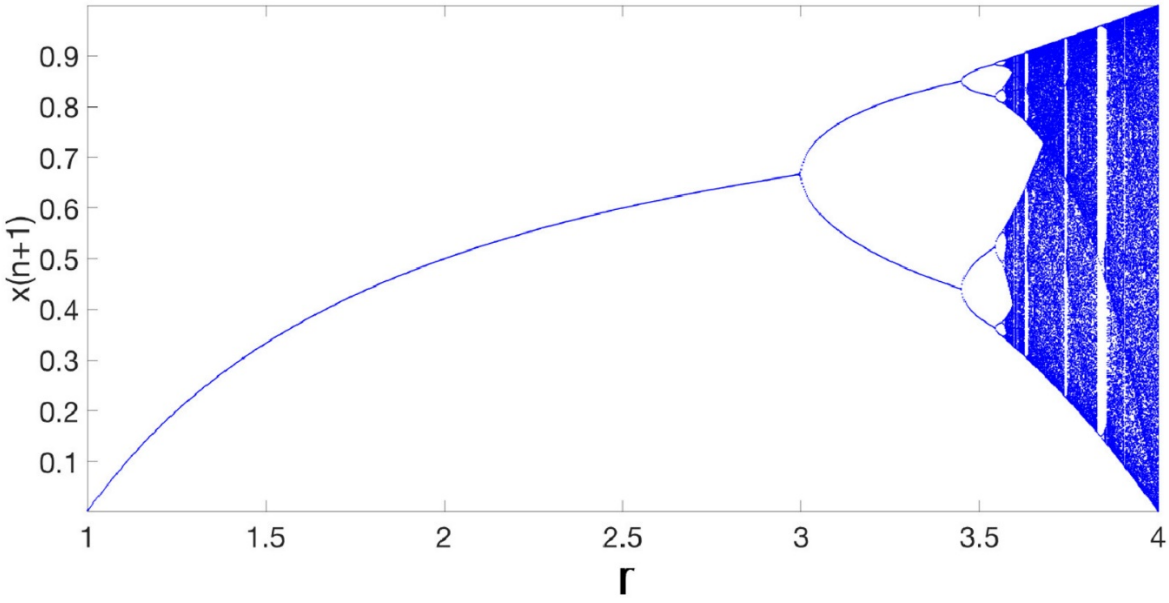


Figura 4. Diagrama de bifurcación del mapa Logístico 1D cuando el parámetro $r \in [1,4]$.

II.5 Mejorando la aleatoriedad

La Figura 5 ilustra un diagrama a bloques del algoritmo propuesto para incrementar la aleatoriedad de las secuencias caóticas representadas por las ec. (1) a la ec. (4), para el encriptado de imágenes en tiempo real. Estas series caóticas se generan utilizando aritmética de punto flotante de 64 bits y se multiplican por 1×10^8. Luego se redondean a enteros de 64 bits, y posteriormente se convierten en secuencias binarias mediante la función $mod(d_n, 1023)$. Por último, se mapean a un formato de 8 bits para que sean compatibles con los píxeles de las imágenes digitales. La operación X-OR con las secuencias caóticas mejoradas se utiliza para encriptar las imágenes digitales. Se comprueba que esta subrutina aumenta la aleatoriedad de las señales caóticas, como se puede evidenciar en los diagramas de bifurcación que se muestran de la Fig.6 a la Fig.9, en los exponentes de Lyapunov que se

presentan en la Tabla I en la sección II.6 y en los niveles de seguridad de la información encriptada que se presentará en el Capítulo 4.

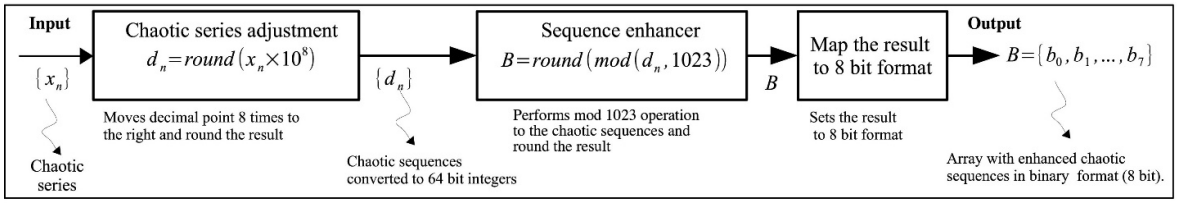


Figura 5. Algoritmo para mejorar las secuencias caóticas para el encriptado de imágenes digitales, modificado de [61].

Para verificar los efectos del algoritmo propuesto se presenta en las Figuras 6 a la 13, diagramas de bifurcación utilizando las funciones $mod255$ y $mod1203$ de los mapas coaticos Bernoulli, Tent, Zigzag y Losgitico 1D. La Fig. 6 muestra el diagrama de bifurcación del mapa de desplazamiento Bernoulli utilizando la función $mod(d_n, 255)$.

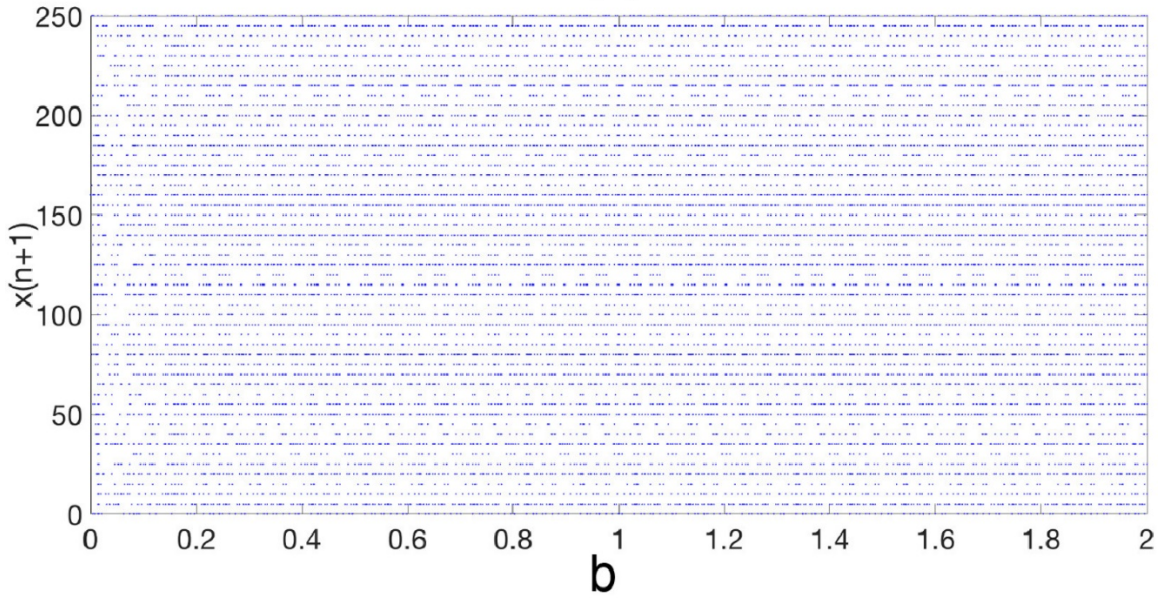


Figura 6. Mapa Bernoulli mejorado con la función $mod255$.

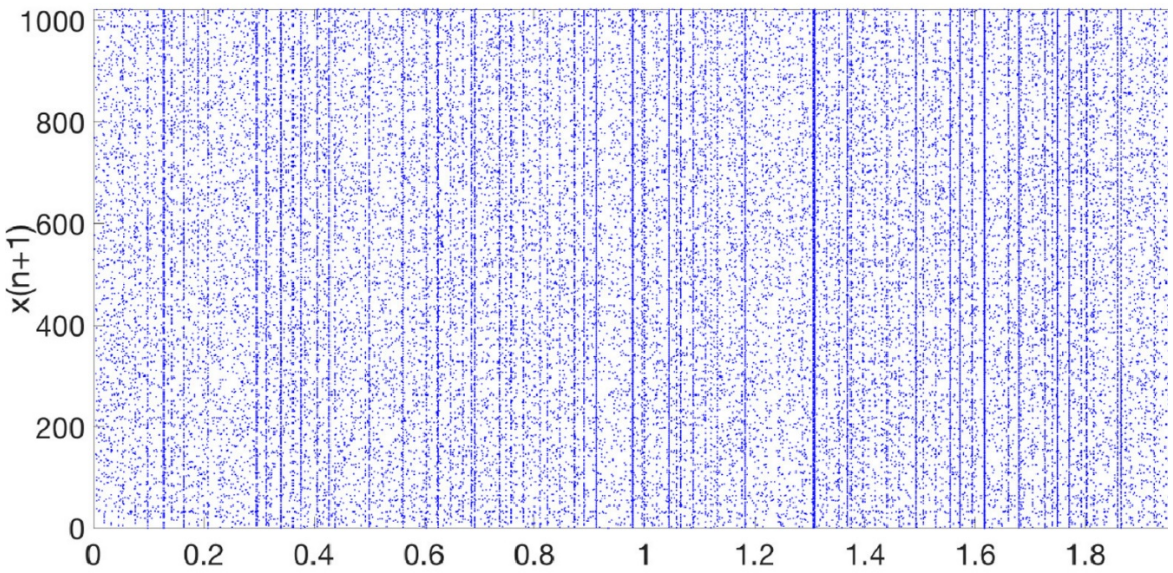


Figura 7. Mapa Bernoulli mejorado con la función $mod1023$

En la Figura 6 se muestra el diagrama de bifurcación del mapa de desplazamiento Bernoulli mejorado utilizando la función $mod255$, es evidente que la bifurcación comienza a partir de $b \in [0,2]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 255.

La Figura 7 muestra el diagrama de bifurcación del mapa de desplazamiento de Bernoulli mejorado utilizando la función $mod\ 1023$. Además, la bifurcación comienza en $b \in [0,2]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 1023.

La Figura 8 muestra el diagrama de bifurcación del mapa mejorado Tent utilizando la función $mod\ 255$, se observa que la bifurcación comienza a partir de $u \in [0,2]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 255.

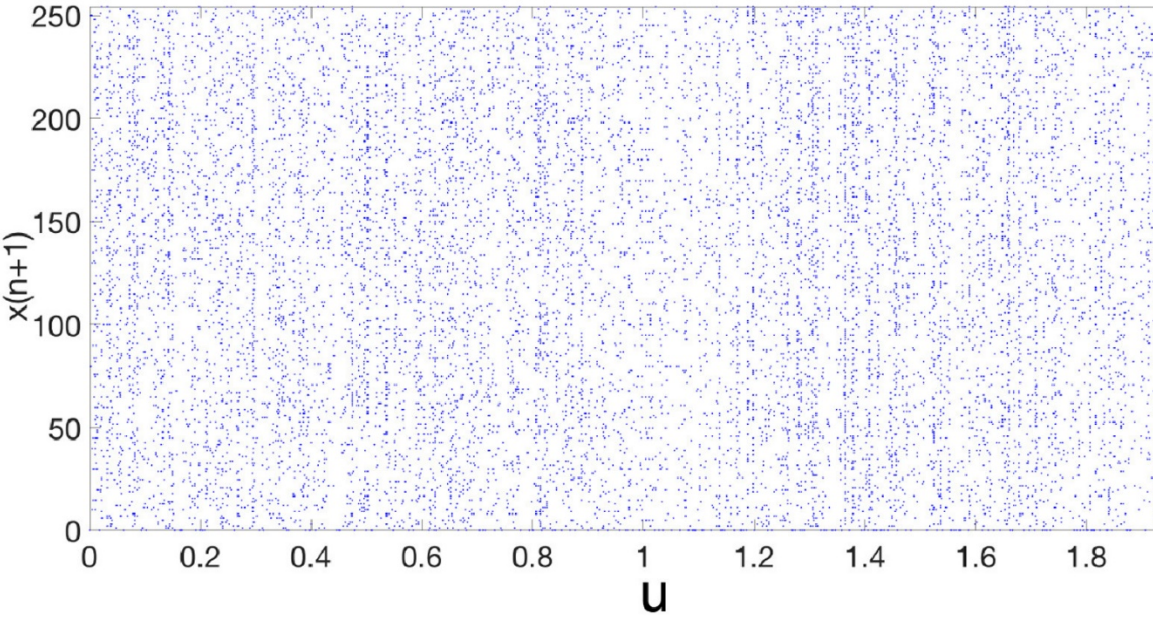


Figura 8. Mapa Tent mejorado con la función $mod255$.

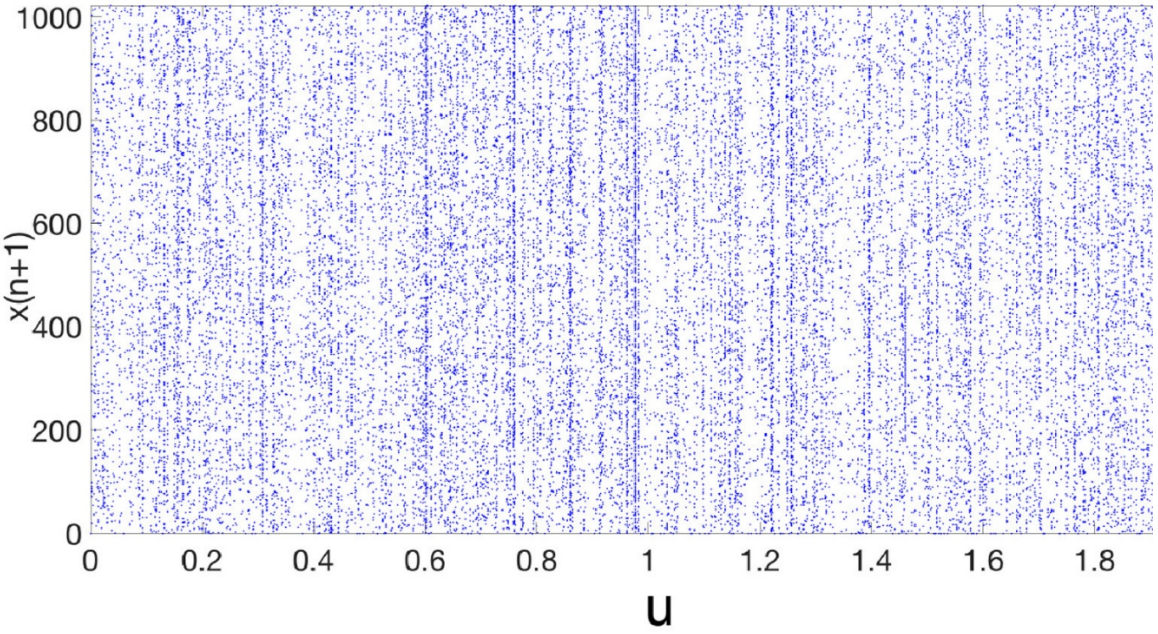


Figura 9. Mapa Tent mejorado con la función $mod1023$.

De la misma manera se realiza el diagrama de bifurcación del mapa mejorado Tent utilizando la función $mod1023$ y se presenta en la Figura 9. Para este caso, se muestra que la bifurcación la comienza a partir de $u \in [0, 2]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 1023.

En la Figura 10 se muestra el diagrama de bifurcación del mapa Zigzag mejorado utilizando la función $mod255$, se puede observar que ahora la bifurcación comienza a partir de $m \in [0,3]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 255.

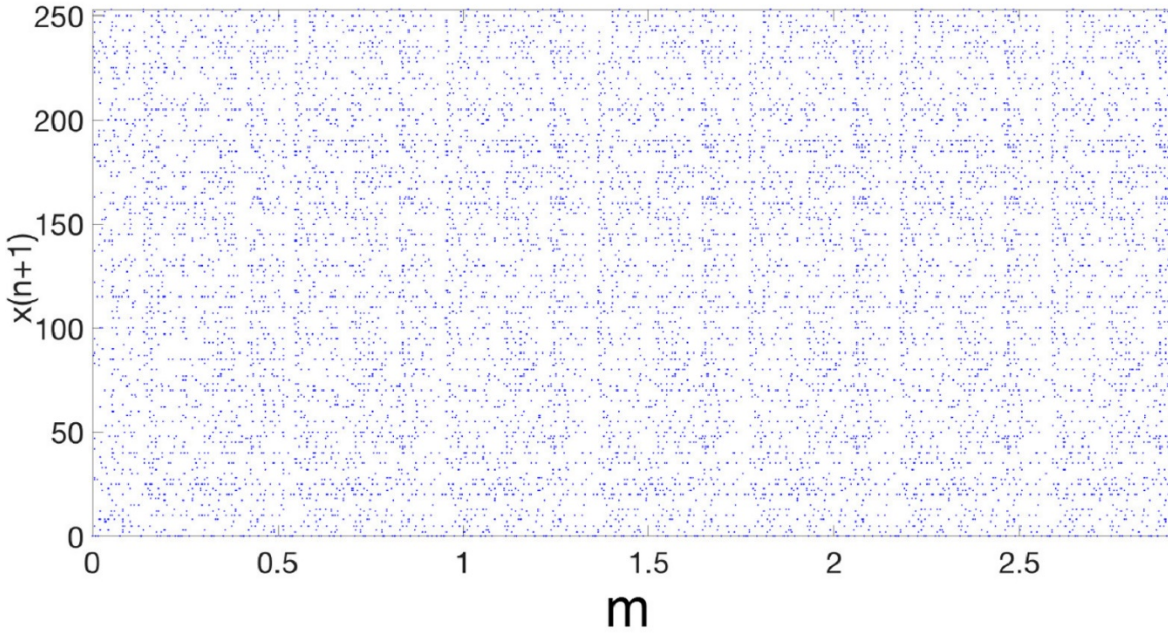


Figura 10. Mapa Zigzag mejorado con la función $mod255$.

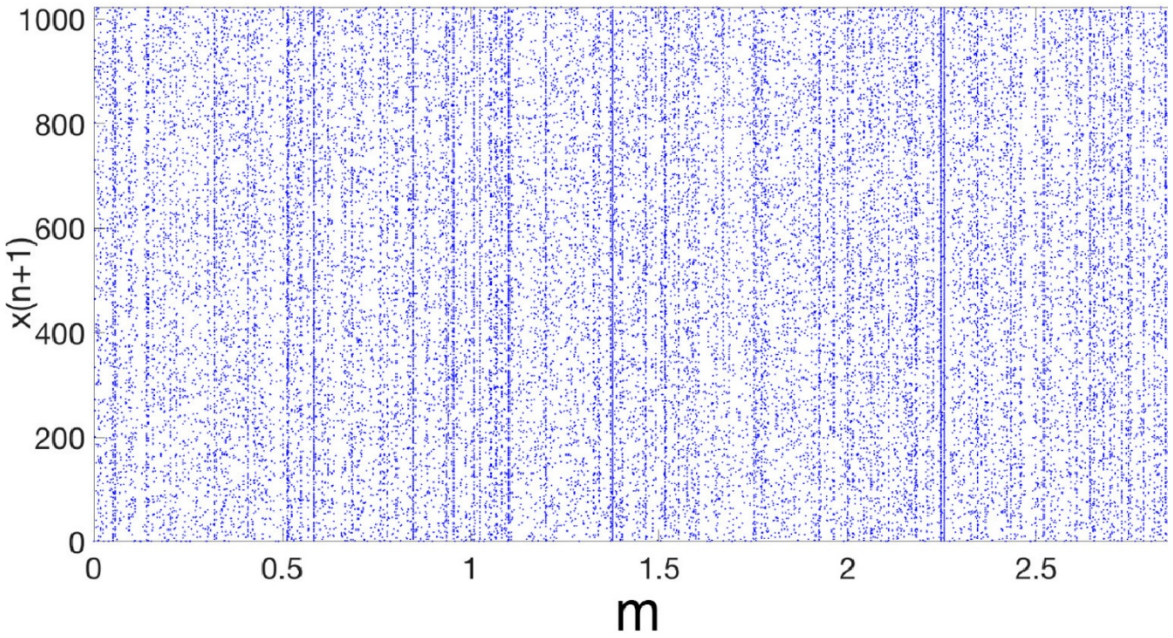


Figura 11. Mapa Zigzag mejorado con la función $mod1023$.

La Figura 11 muestra el diagrama de bifurcación del mapa Zigzag mejorado utilizando la función $mod\ 1023$. Además, la bifurcación comienza en $m \in [0, 3]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 1023.

En cuanto a los diagramas de bifurcación del mapa Logístico 1D, presenta el mismo comportamiento dinámico, como se muestra en la Figura 12 donde se utiliza la función $mod255$, la bifurcación comienza a partir de $r \in [1, 4]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 255.

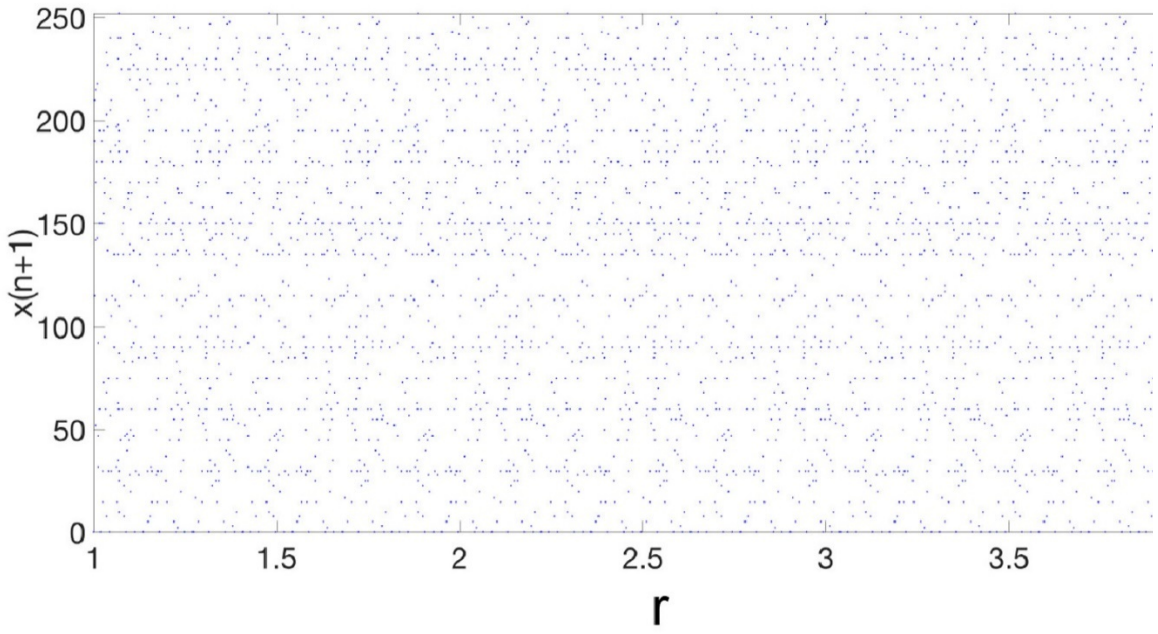


Figura 12. Mapa Logistic 1D mejorado con la función $mod255$.

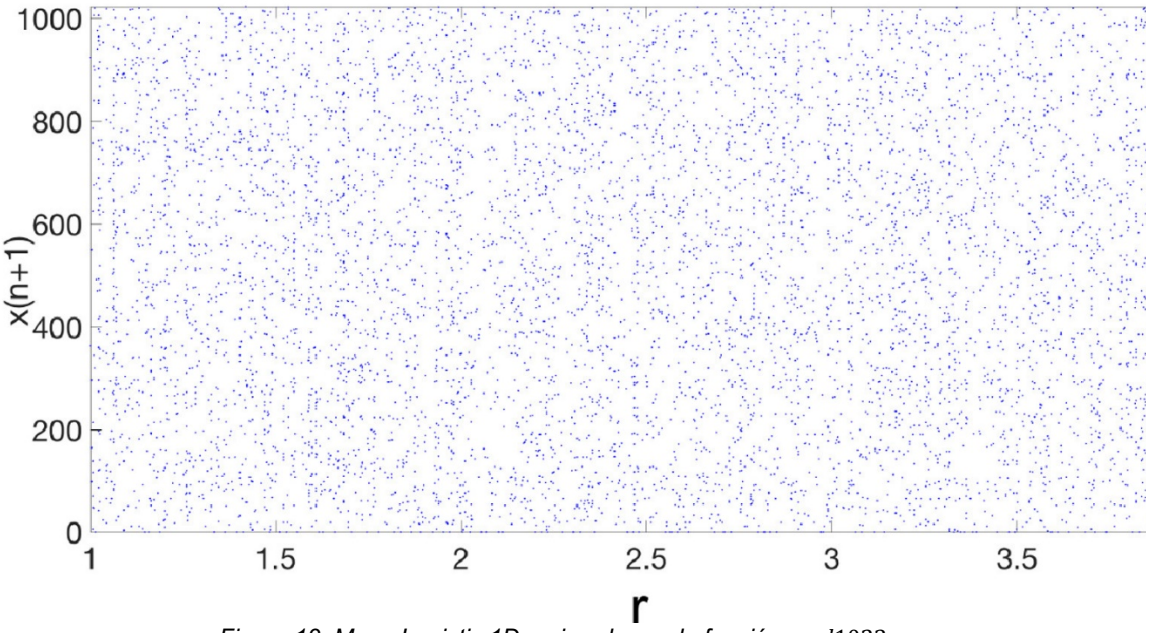


Figura 13. Mapa Logistic 1D mejorado con la función $mod 1023$.

Y finalmente, el diagrama de bifurcación del mapa Logístico 1D mejorado utilizando la función $mod\ 1023$ se muestra en la Figura 13, la bifurcación comienza en $r \in [1, 4]$ y el rango de valores de x_{n+1} oscila aleatoriamente entre 0 y 1023.

Por lo tanto, el espectro de las secuencias caóticas mejoradas se amplía, lo que ayuda a mejorar los niveles de seguridad en el proceso de encriptación de imágenes.

II.6 Exponentes de Lyapunov

Los exponentes de Lyapunov (LE) puede definirse como la divergencia o convergencia exponencial media de las trayectorias muy cercanas en el espacio de fases. Cualquier sistema con al menos un LE positivo se define como un sistema caótico [49]. Los LE de un mapa caótico pueden calcularse utilizando la ecuación (5):

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \log_2 \frac{p_i(t)}{p_i(0)} \quad (5)$$

Donde el i-ésimo exponente de Lyapunov unidimensional se define entonces en términos de la longitud del eje principal elipsoidal $p_i(t)$, además λ_i están ordenados de mayor a menor.

En este trabajo, los LE se calcularon utilizando el software TISEAN [65]. Se espera que un sistema caótico se vuelva más impredecible cuando aumenta su valor de LE positivo [7], [49]. Las secuencias mejoradas (sección II.5) de los mapas caóticos 1D se analizan para evaluar su LE asociado, determinando la aleatoridad del sistema respectivo. Esta tarea se lleva a cabo utilizando TISEAN 3.0.1, que puede operar en el ambiente de MATLAB, donde la subrutina Lyap_spec se utiliza para el análisis de los LE de los mapas caóticos. Se estima todo el espectro de LE para una serie temporal posiblemente multivariada [65]. Para todos los análisis de mapas caóticos, Lyap_spec se configura con los siguientes parámetros: $[-c1]$ que indica que sólo se lea una columna, $[-m1, 1]$ que es el número de componentes y la dimensión de incrustación, y $[-r, 0.2]$ que es el tamaño mínimo del vecindario. Los demás parámetros se ajustan a los valores predeterminados de TISEAN que se indican en [65]. El archivo de salida de Lyap_spec arroja: los LE, el o los errores medios de previsión del modelo lineal local, el tamaño medio de la vecindad utilizado para ajustar el modelo y la dimensión de Kaplan-Yorke estimada. La Tabla I muestra una comparación de los LE de los mapas caóticos 1D frente a los mapas mejorados utilizando las funciones $mod\ 255$ y $mod\ 1023$.

Tabla I. Exponentes de Lyapunov de los mapas caóticos 1D originales y mejorados.

Mapa Caótico	**Exponentes de Lyapunov**		
	Mapa Caótico Original	**Mejorado** $mod\ \mathbf{255}$	**Mejorado** $mod\ \mathbf{1023}$
Desplazamiento Bernoulli	0.7518	3.8273	4.9151
Tent	0.6793	3.8583	5.2145
Zigzag	1.0695	3.7539	4.9020
Logístico 1D	0.6763	3.8590	5.2247
Trabajos relacionados	LE1	LE2	LE3
STCS y L-LCS [24]	0.6763	1.9900	7.2500
Mapa Logístico 1D y 1DLSE [7]	0.6763	14.250	9.5250
PLM [66]	0.6763	4.4200	4.5500
3D-PLM [18]	2.8750	2.8000	2.8400
Nueva señal caótica [25]	0.3451	0.9249	0.7778
Nuevo sistema hipercaótico 5-D [63]	0.0630	0.0050	0.0000
PELM [23]	0.6900	10.500	-
Multi-modal [14]	0.6875	-	-
Logistico 1D [22]	0.6763	-	-

Se puede observar que los LE de los mapas originales son menores que los obtenidos con la función $mod 255$. A su vez, se observa que con la función $mod\ 1023$, todos los mapas caóticos 1D mejorados tienen aún un valor mayor en el LE, en el que el mapa Logístico 1D mejorado tiene el máximo valor.

II.7 Pruebas NIST

El conjunto de pruebas del NIST es un paquete estadístico que consta de 16 pruebas desarrolladas para comprobar la aleatoriedad de secuencias binarias (de longitud arbitraria) producidas por generadores de números aleatorios o pseudoaleatorios basados en hardware o software. Estas pruebas se centran en

una variedad de tipos diferentes de no aleatoriedad que podrían existir en una secuencia. Algunas pruebas son divididas en una variedad de subpruebas. Para cada prueba se calcula un valor-p, el cual indicará si la secuencia es o no aleatoria. Para un valor-$p \geq 0.01$ la secuencia es aleatoria. Este software ha sido desarrollado por el NIST US y es de dominio público.

Todas estas pruebas estadísticas se ejecutan utilizando 100 secuencias con el streaming de longitud = 1,000,000 bits para cada uno de los mapas caóticos 1D investigados en este trabajo de tesis. La tabla II muestra los resultados de las pruebas NIST para las secuencias binarias mejoradas. Se puede ver que para cada prueba, el valor-p es ≥ 0.01 y la proporción es superior a 0.96. Así pues, estas secuencias han superado con éxito todas las pruebas NIST. Por lo tanto, según los criterios de calidad definidos por el NIST SP 800-22 [67], las secuencias caóticas mejoradas son aleatorias y adecuadas para aplicaciones criptográficas.

Tabla II. Resultados de la prueba NIST para las secuencias mejoradas generadas con los mapas caóticos 1D.

Prueba estadística NIST SP 800-22	Secuencias caóticas 1D mejoradas mediante la función mod 1023							
	Bernoulli		Tent		Zigzag		Logistic 1D	
	p − value	Proportion	*p − value*	Proportion	*p − value*	Proportion	*p − value*	Proportion
Frequency	0.987896	99/100	0.699313	100/100	0.595549	99/100	0.289667	98/100
Block Frequency	0.739918	100/100	0.051942	99/100	0.699313	99/100	0.987896	99/100
Cumulative sums-Forward	0.153763	98/100	0.455937	100/100	0.514124	99/100	0.595549	98/100
Cumulative sums-Reverse	0.719747	99/100	0.437274	100/100	0.224821	99/100	0.401199	99/100
Runs	0.494392	97/100	0.55442	100/100	0.162606	96/100	0.026948	98/00
Longest runs of ones	0.494392	100/100	0.798139	100/100	0.419021	99/100	0.12962	100/100
Rank	0.171867	99/100	0.319084	100/100	0.437274	99/100	0.955835	100/100
Spectral DFT (FFT)	0.191687	98/100	0.224821	99/100	0.191687	100/100	0.383827	100/100
Non-overlapping Templates	0.997823	100/100	0.759756	99/100	0.574903	99/100	0.911413	99/100
Overlapping Templates	0.739918	99/100	0.12962	100/100	0.437274	99/100	0.080519	99/100
Universal	0.23681	99/100	0.616305	100/100	0.319084	98/100	0.437274	98/100
Approximate Entropy	0.102526	99/100	0.474986	97/100	0.162606	99/100	0.137282	97/100
Random Excursions	0.181557	59/59	0.911413	61/62	0.846579	70/70	0.202268	58/59
Random Excursions Variant	0.595549	59/59	0.804337	61/62	0.929192	70/70	0.275709	58/59
Linear Complexity	0.085587	99/100	0.080519	99/100	0.437274	99/100	0.23681	99/100
Serial (2m∇*W*)	0.051942	99/100	0.834308	100/100	0.996335	99/100	0.062821	97/100

II.8 TestU01

Debido a los importantes avances en el campo de la tecnología informática, los nuevos PRNGs deben ser sometidos a pruebas más rigurosas. Por esta razón, se aplicaron las pruebas TestU01 para probar la calidad de la aleatoriedad del PRNG propuesto. TestU01 es una biblioteca de software, implementada en el lenguaje ANSI C, que ofrece una colección de utilidades para la comprobación estadística empírica de generadores de números aleatorios uniformes.

La biblioteca implementa varios tipos de generadores de números aleatorios en forma genérica, así como muchos generadores específicos propuestos en la literatura o que se encuentran en software ampliamente utilizado. Proporciona implementaciones generales de las pruebas estadísticas clásicas para los generadores de números aleatorios, así como varias otras propuestas en la literatura, y algunas originales. Estas pruebas pueden aplicarse a los generadores predefinidos en la biblioteca y a generadores propuestos por el usuario. También están disponibles conjuntos de pruebas específicas para secuencias de números aleatorios uniformes en [0,1] o secuencias de bits. Aunque el TestU01 tiene seis pruebas predefinidas, sólo tres son para secuencias de bits: Rabbit, Alphabit y Block Alphabit [68]. Al utilizar las pruebas, Rabbit, Alphabit y Block Alphabit, se debe especificar el número de bits disponibles para cada prueba. En este trabajo, se utilizaron 1,000,000 de bits para 100 secuencias. Los demás parámetros de cada prueba se eligen automáticamente en función del número de bits disponibles. Rabbit y Alphabit aplican 40 y 17 pruebas estadísticas diferentes, respectivamente. Block Alphabit aplica un conjunto de pruebas de Alphabit repetidamente a un generador o a un archivo binario tras reordenar los bits por bloques de diferentes tamaños (con tamaños de 2, 4, 8, 16, 32 bits).

Tabla III. Comparación de los porcentajes de eficacia (%) según el TestU01.

	Secuencias caóticas 1D mejoradas mediante la función $mod\ 1023$			
TestU01	**Bernoulli**	**Tent**	**Zigzag**	**Logistic 1D**
Rabbit	87.50	92.50	85.00	90.00
Alphabit	82.35	94.11	76.47	88.23
Block Alphabit	84.26	93.50	82.35	89.17

La Tabla III muestra los resultados obtenidos al aplicar las pruebas TestU01 a las series binarias de los mapas caóticos 1D utilizados en el PRNG propuesto. Se puede observar que los porcentajes de pruebas superadas están entre el 76.47% y el 94.11%. Según [68], un PRNG que pasa todas las pruebas del NIST puede fallar en algún TestU01, pero muy pocas veces pasa lo contrario. Por lo tanto, las pruebas TestU01 se asume que son un conjunto de pruebas estadísticas más estrictas y completas [69] que las pruebas NIST [67].

Capítulo III. Esquema propuesto para aplicaciones IoT

MQTT es una especificación de protocolo de transporte para la transmisión de mensajes, que permite a los desarrolladores elegir entre tecnologías de red, privacidad, autenticación y autorización. Dado que las tecnologías de seguridad especificas son elegidas de acuerdo al contexto, es responsabilidad del desarrollador incluir las características apropiadas como parte de su diseño [52]. La ubicación del broker es esencial porque el protocolo MQTT intercambia información a través de tópicos, y todos los dispositivos con el mismo tópico deben estar conectados al broker MQTT [51]. Aunque el estándar MQTT recomienda encarecidamente el uso de los protocolos Secure Sockets Layer (SSL) y Transport Layer Security (TLS) en los servidores que se utilizan como broker con MQTT. En este trabajo, se propone añadir técnicas de criptografía basadas en el caos como medida de seguridad adicional para proteger la información confidencial (imágenes RGB) de extremo a extremo. El dispositivo IoT es el criptosistema integrado propuesto y debe conocer la ubicación del broker con un tópico. Un dispositivo IoT perteneciente a una determinada red de un extremo se suscribe al tópico de otra red en otro extremo [51].

El sistema criptográfico propuesto se implementa utilizando una Raspberry Pi 4 (RPi4), que incluye un CPU Broadcom BCM2711 Quad-Core ARM Cortex A72 a 1.5GHz, 4 Gb de RAM para almacenar datos temporales, 32 Gb de memoria Flash para almacenar el sistema operativo (OS) y el firmware. Se recomienda utilizar un sistema operativo en tiempo real basado en GNU Linux y el firmware del protocolo

de comunicación, en este caso MQTT, para la transmisión de información a través de una red de comunicación, como WiFi e Internet [61].

La Figura 14 muestra el diagrama de bloques del esquema propuesto para una aplicación IoT utilizando el protocolo MQTT para transmitir imágenes digitales encriptadas a través de la red WiFi y por internet, lo anterior en un esquema de máquina a máquina. Se puede ver que, para la implementación de este esquema, se utiliza un RPi4 como dispositivo central, y se programa utilizando el lenguaje Python para ejecutar un mapa caótico mejorado como PRNG. A continuación, el RPi4 encripta la imagen RGB mediante la operación X-OR entre la señal caótica y la imagen digital adquirida mediante el uso de una cámara web o un archivo, como se muestra en las Figuras 15 y 16. La salida del criptosistema embebido se envía a través del módulo WiFi a un punto de acceso/router, y luego se transmite por Internet a un broker MQTT externo, por ejemplo, mqtt.dioty.co o broker.mqtt-dashboard.com. En el receptor autorizado se realiza el proceso inverso de encriptación. La Fig. 14 también muestra a un intruso/hacker que podría suscribirse al mismo tema e intentar obtener la información confidencial que se está enviando. Para ello, el intruso/hacker debe conocer el algoritmo y la clave de cifrado.

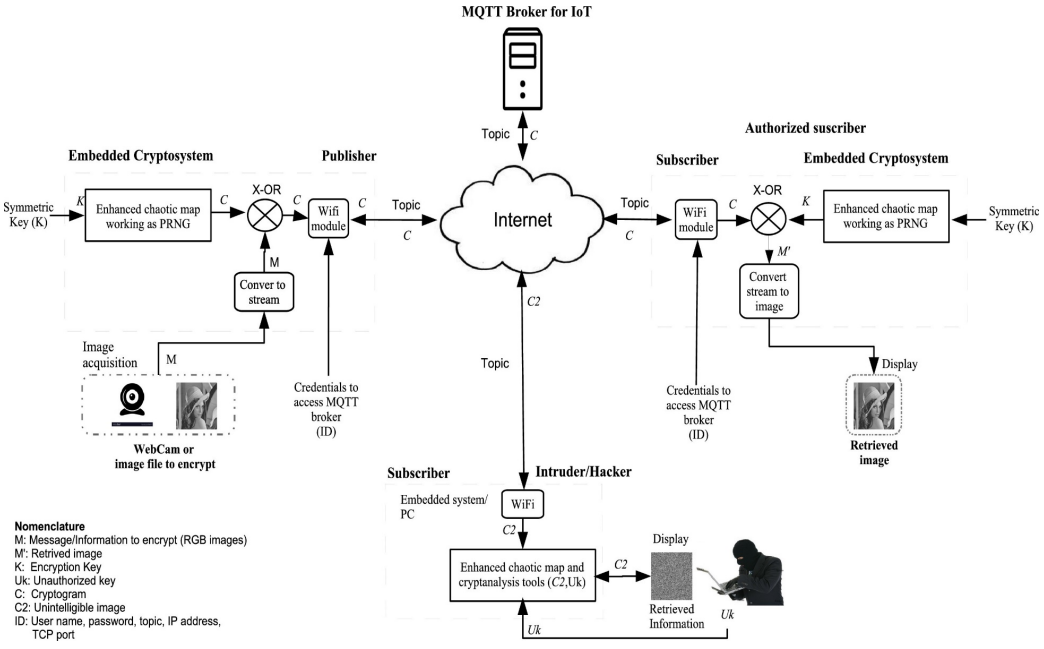


Figura 14. Esquema propuesto para la transmisión segura de imágenes RGB en tiempo real [46].

La Figura 15 muestra el encriptado con clave simétrica propuesto utilizando secuencias caóticas mejoradas en un editor (transmisor) para aplicaciones de IoT. En la parte izquierda, se pueden ver las entradas de datos al algoritmo; en este caso, la imagen original o imagen a encriptar, que debe ser convertida a un flujo de datos, y la clave de encriptado, que son las condiciones iniciales y los parámetros del mapa caótico 1D seleccionado. Para entrar en el criptosistema embebido, se requiere conocer las credenciales del broker MQTT (MQTT's ID), en este caso particular son: Nombre de usuario, contraseña, tema, dirección IP y puerto TCP. Además, se puede observar la operación X-OR, que encripta los píxeles con 0's y 1's dando la secuencia generada por el mapa caótico mejorado 1D. A continuación, el criptograma resultante se envía a un broker externo a través de Internet utilizando el protocolo MQTT y una red WiFi.

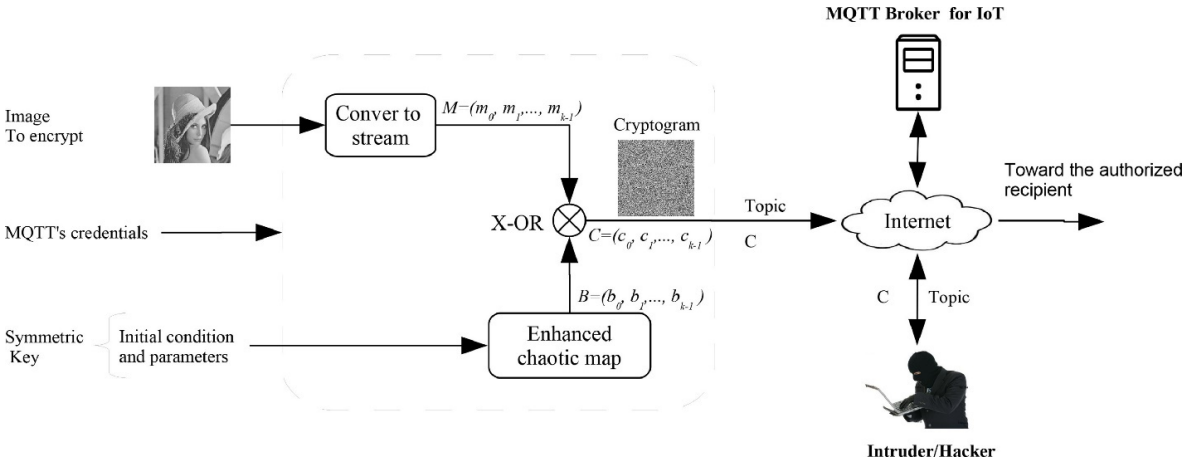


Figura 15. Propuesta de encriptado con clave simétrica para aplicaciones IoT utilizando mapas caóticos mejorados [46].

La Figura 16 describe el desencriptado de secuencias con clave simétrica propuesto (receptor autorizado) utilizando un mapa caótico mejorado. El suscriptor (receptor) debe realizar el proceso inverso del encriptado; es decir, recibe el criptograma, y debe introducir la clave simétrica y las credenciales de autenticación del servidor MQTT. Posteriormente, realiza el proceso de desencriptado a través de la operación X-OR, finalmente el flujo de datos debe ser convertido a una matriz 2D para recuperar la información.

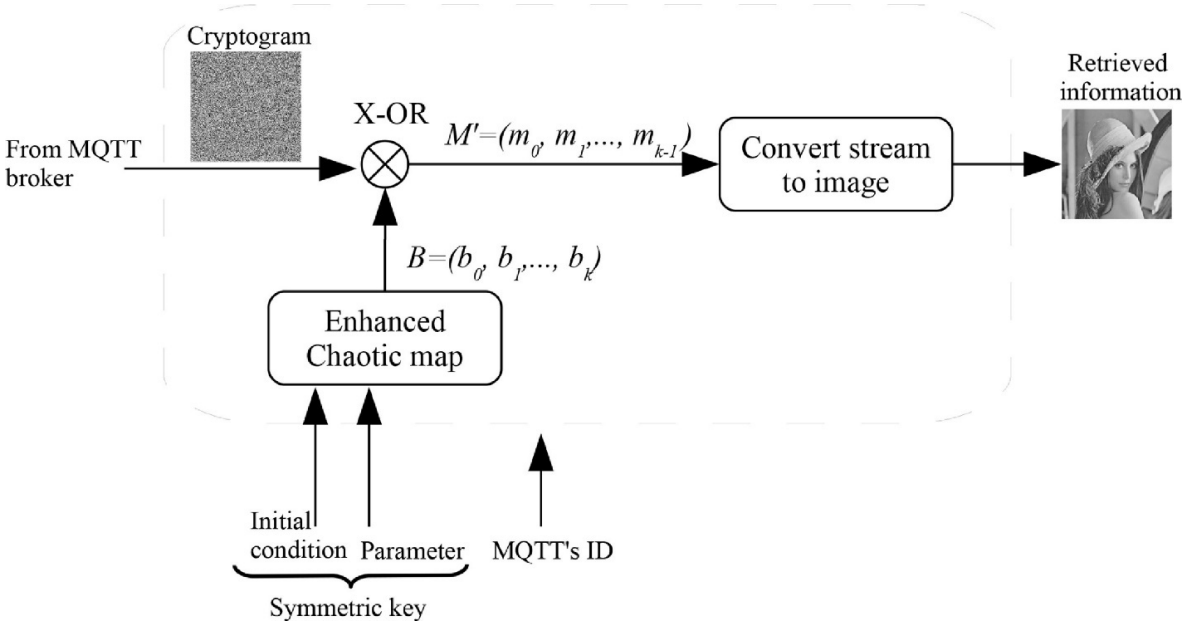


Figura 16. Propuesta de desencriptado con clave simétrica en un receptor autorizado, para aplicaciones IoT utilizando mapas caóticos mejorados [46].

Capítulo
IV. Resultados Experimentales

Se seleccionaron la conocida base de datos USC-SIPI6 y el Open Images Dataset V6+ [70] para verificar la funcionalidad y la seguridad del criptosistema propuesto. La base de datos de imágenes USC-SIPI es una colección de imágenes digitalizada, se utiliza principalmente para apoyar la investigación en el procesamiento de imágenes, el análisis de imágenes y la visión artificial. La primera edición de la base de datos de imágenes USC-SIPI se distribuyó en 1977 y desde entonces se han añadido muchas imágenes nuevas. El conjunto de datos se divide en diferentes grupos dependiendo según del carácter básico de las imágenes. Las imágenes de cada grupo tienen varios tamaños, como 256×256 píxeles, 512×512 píxeles o 1024×1024 píxeles. Todas las imágenes son de 8 bits/píxel para las imágenes en blanco y negro y de 24 bits/píxel para las imágenes en color. El conjunto de datos USC-SIPI incluye cuatro grupos de imágenes: Texturas (64 imágenes), Aéreas (38 imágenes), Misceláneas (39 imágenes) y Secuencias (69 imágenes). Respecto al Open Images Dataset V6+, se trata de un conjunto de datos de 9 millones de imágenes variadas con una rica anotación, las imágenes son muy diversas y a menudo contienen escenas complejas con varios objetos (8.4 por imagen en promedio). Contiene anotaciones de etiquetas a nivel de imagen, cuadros delimitadores de objetos, segmentaciones de objetos, relaciones visuales, narraciones localizadas, entre otras. Estas imágenes han sido empleadas para mostrar su potencial aplicación en comunicaciones privadas a través de diferentes plataformas sociales, como WhatsApp, Signal, Telegram, entre otras aplicaciones de mensajería que funcionan en tiempo real a través de Internet. En este trabajo, para los experimentos se utilizan imágenes estándar para verificar la funcionalidad y la seguridad. Se han realizado varios experimentos para validar la robustez, eficiencia y seguridad del esquema propuesto.

La Figura 17 muestra las imágenes seleccionadas de la base de datos USC-SIP, todas de tamaño 512×512 píxeles, y corresponden a: (a) Lena en escala de grises, (b) Hombre cámara en escala de grises, (c) Mandril en escala de grises, (d) Lena en formato RGB, (e) Mandril en formato RGB, y (f) Pimientos en formato RGB.

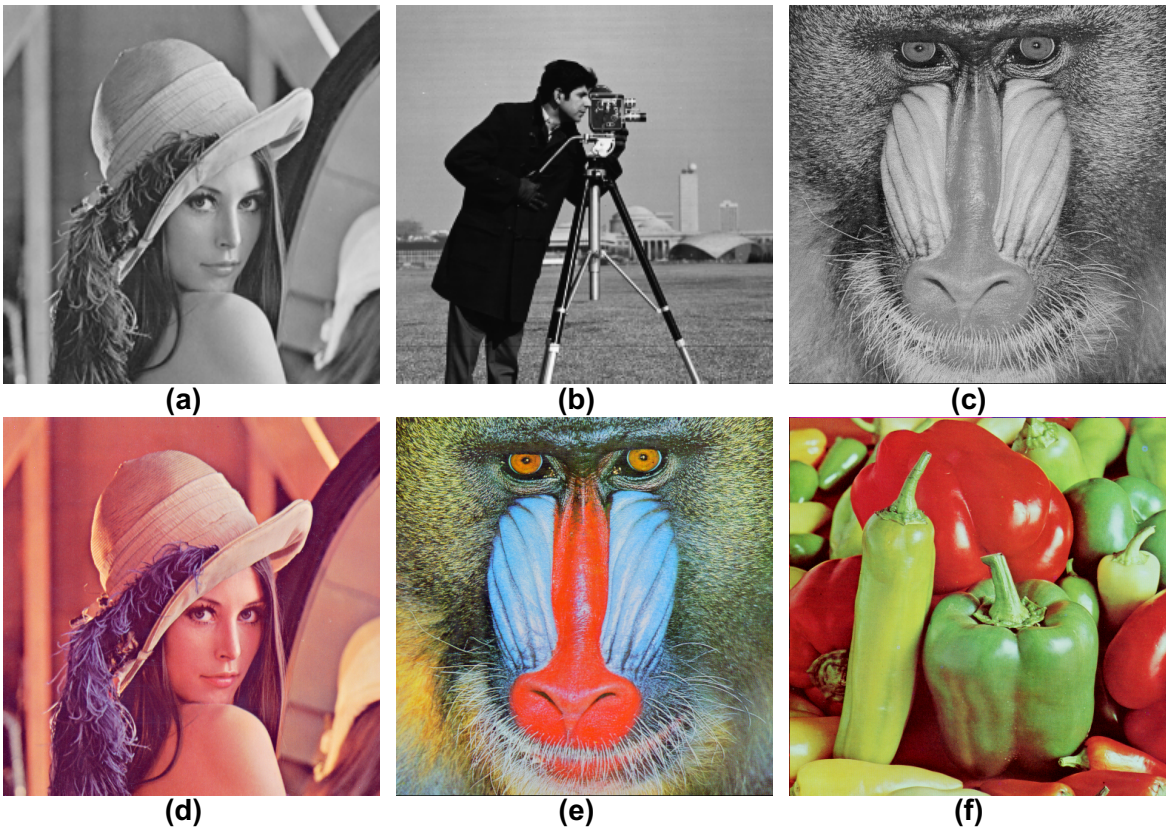


Figura 17. Imágenes de prueba del conjunto de datos USC-SIPI: (a) Lena, (b) Camera man, (c) Mandrill, (d) Lena en formato RGB, (e) Mandrill en formato RGB, y (f) Peppers en formato RGB.

Las versiones encriptadas, del modelo caótico logístico 1D mejorado $mod1023$ de las imágenes de la Figura 17, se muestran en la Figura 18. De los resultados de la encriptación se puede observar que la información de grises y de color de las imágenes de entrada queda efectivamente oculta.

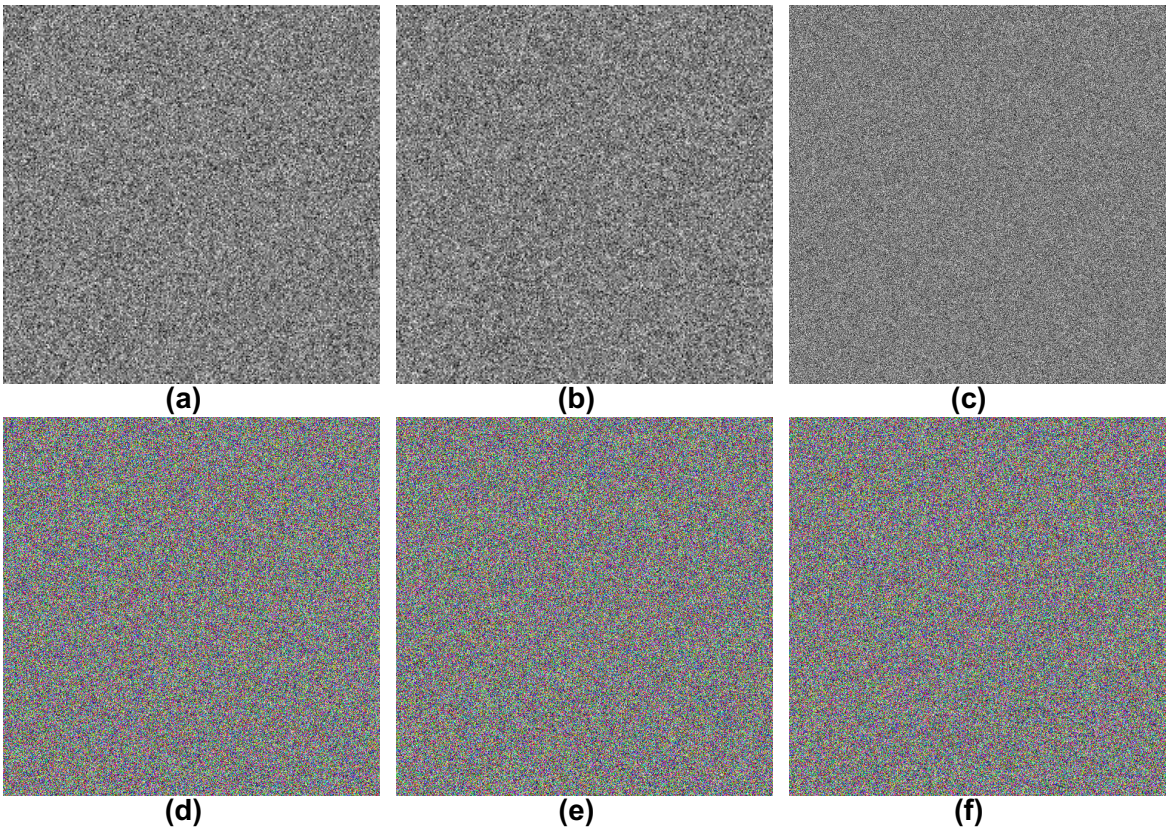


Figura 18. Resultados de encriptado de imágenes de prueba: (a) Criptograma de Lena, (b) Criptograma Camera man, (c) Criptograma del mandril, (d) Criptograma de Lena RGB, (e) Criptograma del mandril RGB, y (f) Criptograma de Peppers RGB.

La Figura 19 (a) (b) (c) muestran las imágenes seleccionadas del Open Images Dataset V6+, las imágenes no tienen un nombre en especifo y se pueden identificar con su número de ID 320eedf38ac2d655, 56b041da97f49fdd y c0e74c2aae1c7e9a respectivamente. Para una mayor comprensión se les denominaron como: Chica 1 para la Figura 19 (a), de tamaño 1200×1600 píxeles, Chica 2 de tamaño 1500×1000 píxeles para la Figura 19 (b), y Chica 3 para la Figura 19(c) de tamaño 4256×2832 píxeles, almacenadas en formato PNG y son de superresolución.

(a)

(b)

(c)

Figura 19. Imagen de prueba del Open Images Dataset V6+: (a) ID 320eedf38ac2d655, Chica 1, (b) ID 56b041da97f49fdd, Chica 2 y (c) ID c0e74c2aae1c7e9a, Chica 3.

Las versiones cifradas empleando el mapa caótico Logistic 1D mejorado $mod1023$ de estas imágenes se muestran en la Figura 20. Una vez más, los resultados del encriptado demuestran que la información de color y tamaño de las imágenes de entrada se ocultan de forma eficaz.

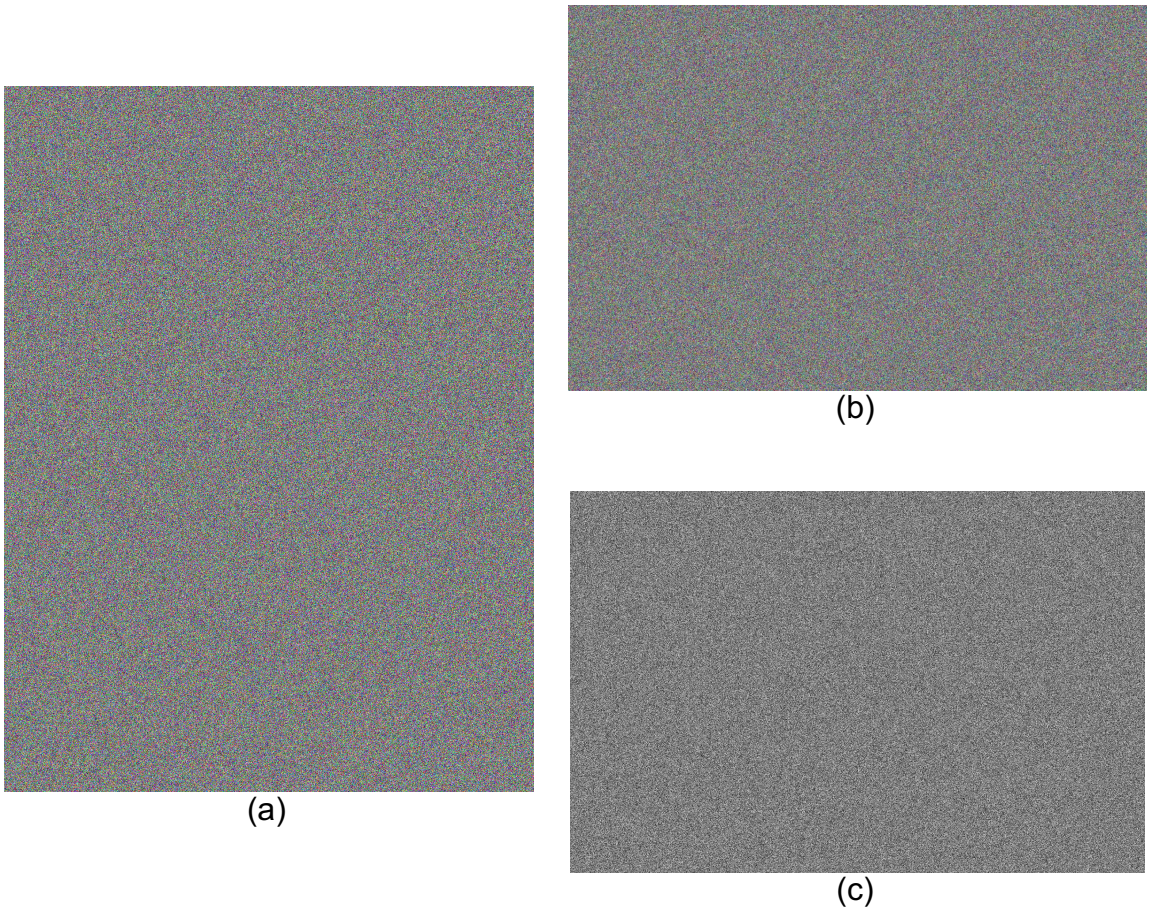


Figura 20. Criptograma de imagen: (a) ID 320eedf38ac2d655, Chica 1, (b) ID 56b041da97f49fdd, Chica 2 y (c) ID c0e74c2aae1c7e9a, Chica 3.

Para evaluar la robustez, la seguridad y el rendimiento del esquema propuesto, se realizaron los siguientes análisis: Histogramas estadísticos, pruebas estadísticas del NIST [67], TestU01 [69], espacio clave, correlación de píxeles adyacentes, correlación entre la imagen original y la cifrada, entropía de Shannon, NPCR y pruebas UACI. Además, la evaluación de la aleatoriedad se realiza para verificar la seguridad adecuada del esquema propuesto para las aplicaciones de IoT. Otro parámetro a evaluar es el rendimiento, el cual se realiza midiendo el rendimiento de los procesos principales.

IV.1 Histogramas

El análisis del histograma se utiliza para medir el rendimiento del esquema a fin de evitar que los atacantes adquieran los píxeles característicos de las imágenes [25]. En este trabajo de tesis se obtuvieron los histogramas para cada imagen seleccionada: seis de la base de datos USC-SIP y tres de super-resolución de la base de datos Open Images Dataset V6+, y con cada uno de los mapas caóticos propuestos mejorados $mod1023$ (Bernoulli, Tent, Zigzag y Logistic 1D). Para fines de ilustración, solo se presentan los resultados de los histogramas con el mapa Logístico 1D mejorado $mod1023$.

La Figura 21 muestra los histogramas de la imagen de Lena de la Fig. 17(a): 21(a) Lena simple y 21(b) Lena encriptada. Se puede observar que el histograma correspondiente a Lena encriptada representada en la Fig. 21(b) tiene una distribución uniforme, lo que significa que el esquema de encriptado tiene un buen rendimiento para resistir los ataques estadísticos [25], además cuando se utilizan los otros tres mapas caóticos, la imagen encriptada presenta el mismo comportamiento de distribución uniforme.

La Figura 22 muestra los histogramas de la imagen de Camera Man de la Fig. 17(b): 22(a) Camera Man simple y 22(b) Camera Man encriptada. Se observa que el histograma correspondiente a Camera Man encriptada representada en la Fig. 22(b) tiene una distribución uniforme, además en los otros tres mapas caóticos la imagen encriptada muestran el mismo comportamiento de distribución uniforme, lo que significa que el esquema de encriptado puede resistir los ataques estadísticos [25].

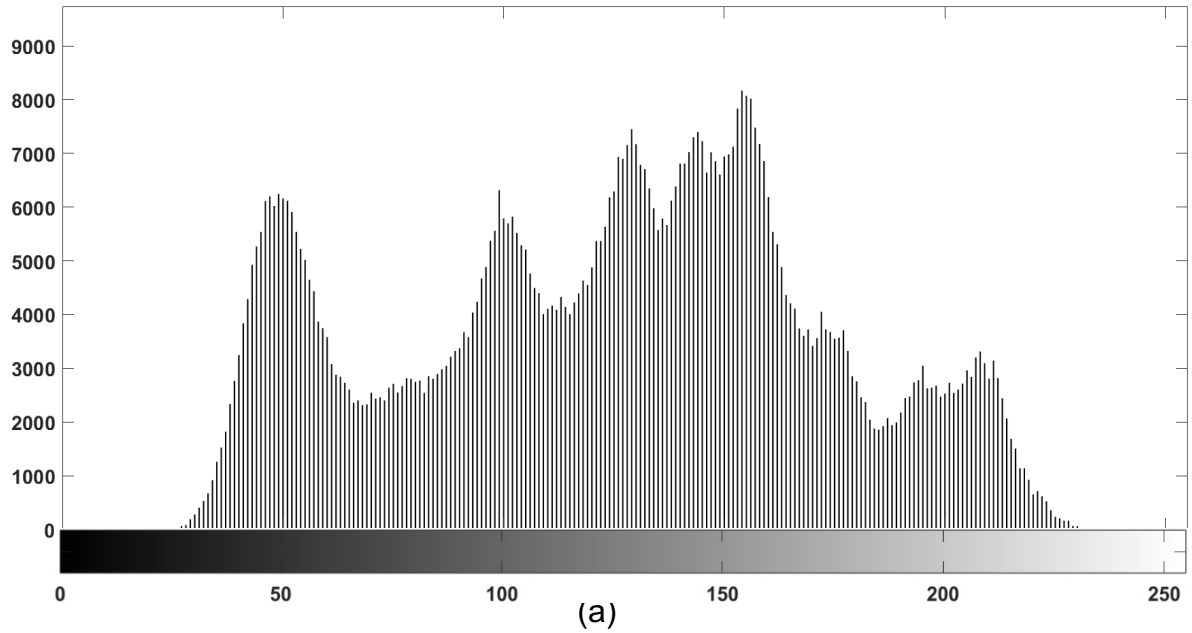
De la misa forma, las Figuras 23, 24, 25 y 26 presentan los histogramas de las imágenes Mandril, Lena RGB, Mandril RGB y Peppers RGB de las Figuras 17(c), 17(d), 17(e) y 17(f) respectivamente. También se observa que los histogramas correspondientes a los canales RGB encriptados representados en las Figs. 24(b), 24(d), 24(f), 25(b), 25(d), 25(f) y 26(b), 26(d), 26(f) tienen una distribución uniforme,

lo que significa que el esquema de encriptación tiene un buen rendimiento para resistir ataques estadísticos [25].

En el caso de las imágenes de super-resolución Girl 1, Girl 2 y Girl 3 de las Figuras 19(a), 19(b) y 19(c) respectivamente se muestran los histogramas en las Figuras 27, 28 y 29. Una vez más se observa que los histogramas correspondientes a los canales RGB encriptados representados en las Figuras 27(b), 27(d), 27(f) y 28(b), 28(d), 28(f) muestran una distribución uniforme, con mayor número de pixeles por su propia naturaleza de super-resolución, por lo que el esquema de encriptación cumple con el objetivo. Y en el caso de la Figura 29(b) también presenta una distribución uniforme.

Utilizando los otros tres mapas caóticos mejorados $mod1023$ (Bernoulli, Tent, Zigzag), las imágenes encriptadas también presentan una distribución uniforme.

Se puede concluir que, en las nueve imágenes utilizadas, los histogramas con las imágenes encriptadas, ya sea en escala de gris o en los correspondientes canales RGB, la distribución de es uniforme, por lo que los esquemas de encriptación cumplen con los ataques estadísticos.



(a)

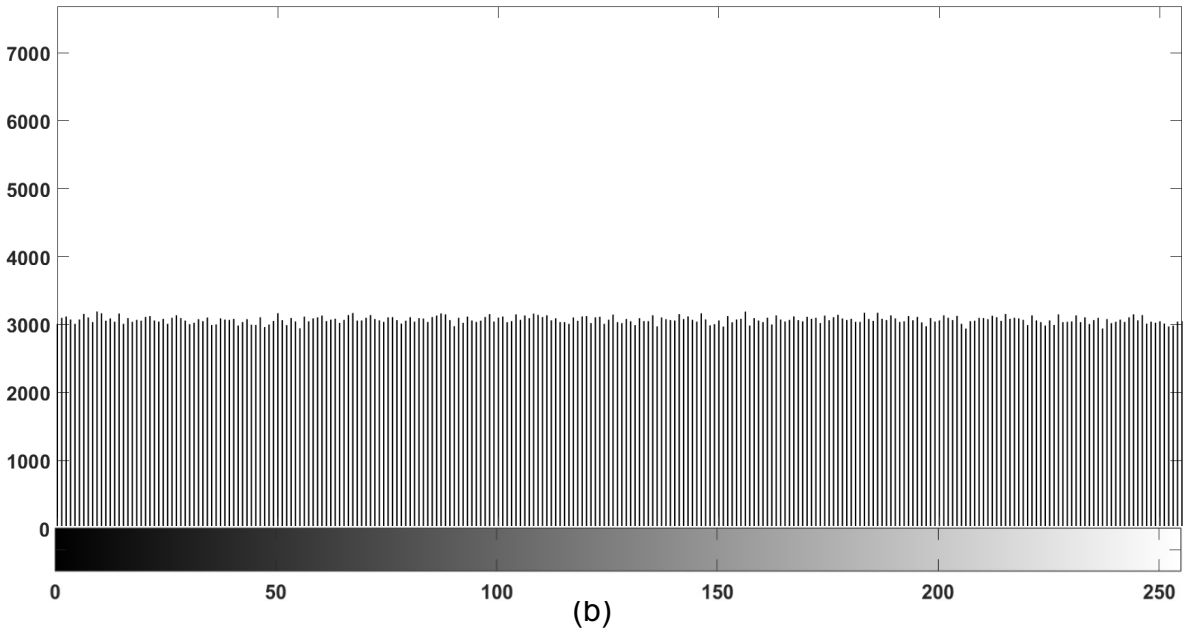


Figura 21. Histogramas de la imagen de Lena de la Figura 17(a): (a) Lena normal, (b) Lena encriptada.

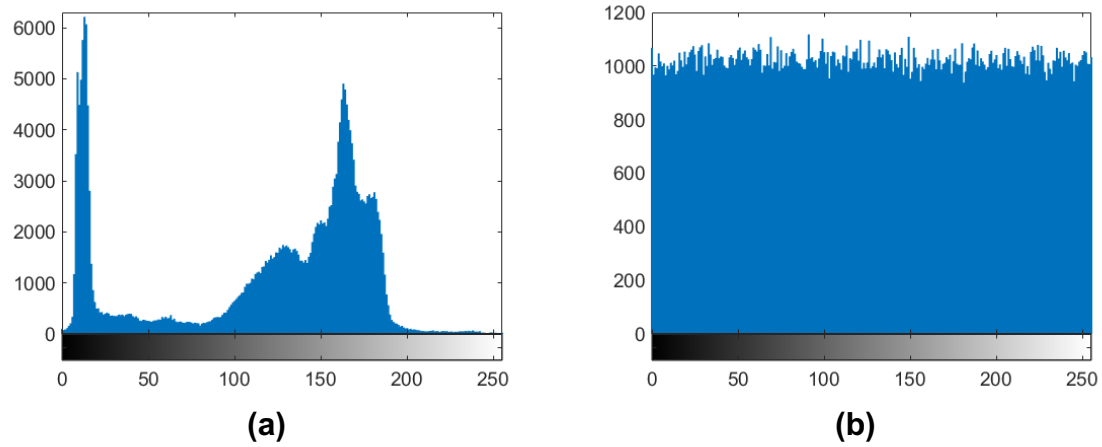
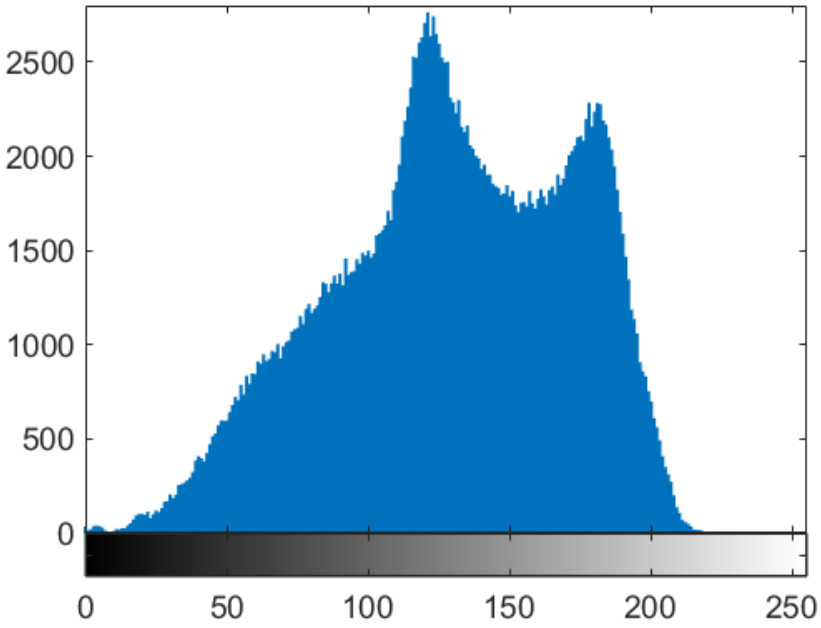
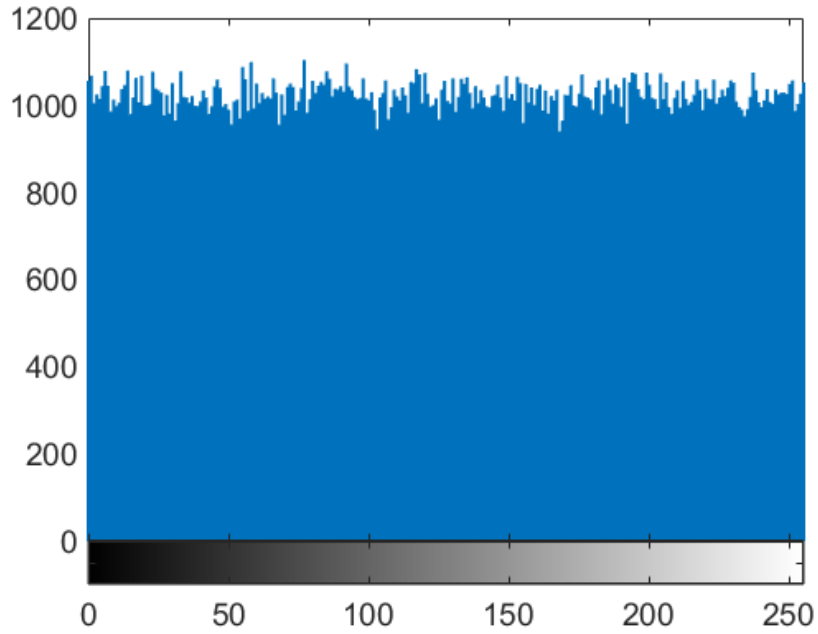


Figura 22. Histogramas de la imagen de Camera Man de la Figura 17(b): (a) Camera Man normal, (b) Camera Man encriptada.



(a)



(b)

Figura 23. Histogramas de la imagen de Mandril de la Figura 17(c): (a) Mandril normal, (b) Mandril encriptada.

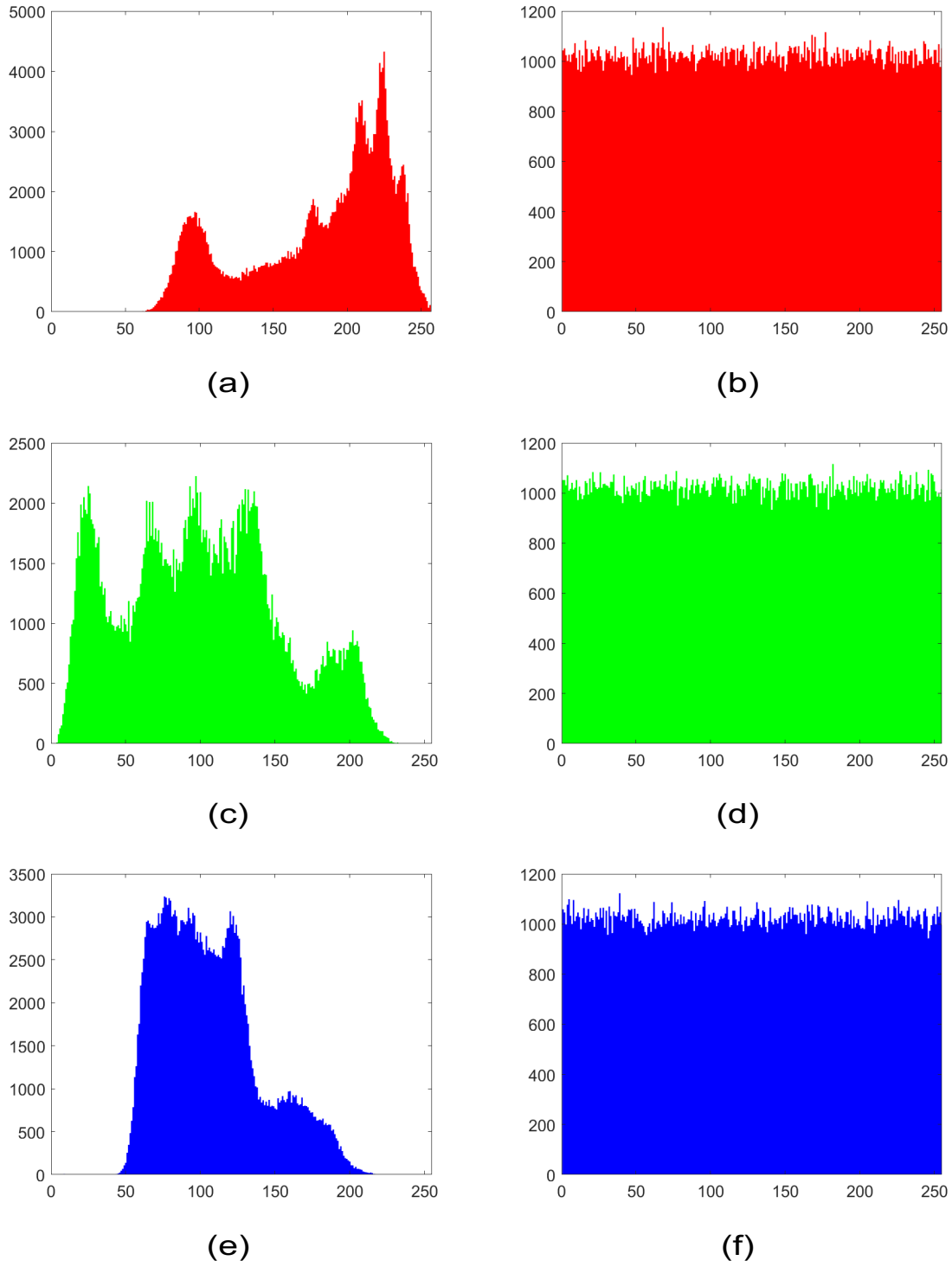


Figura 24. Histogramas de la imagen Lena RGB de la Fig. 17(d): (a) Lena normal en el canal R, (b) Lena encriptada en el canal R, (c) Lena normal en el canal G, (d) Lena encriptada en el canal G, (e) Lena normal en el canal B, (f) Lena encriptada en el canal B.

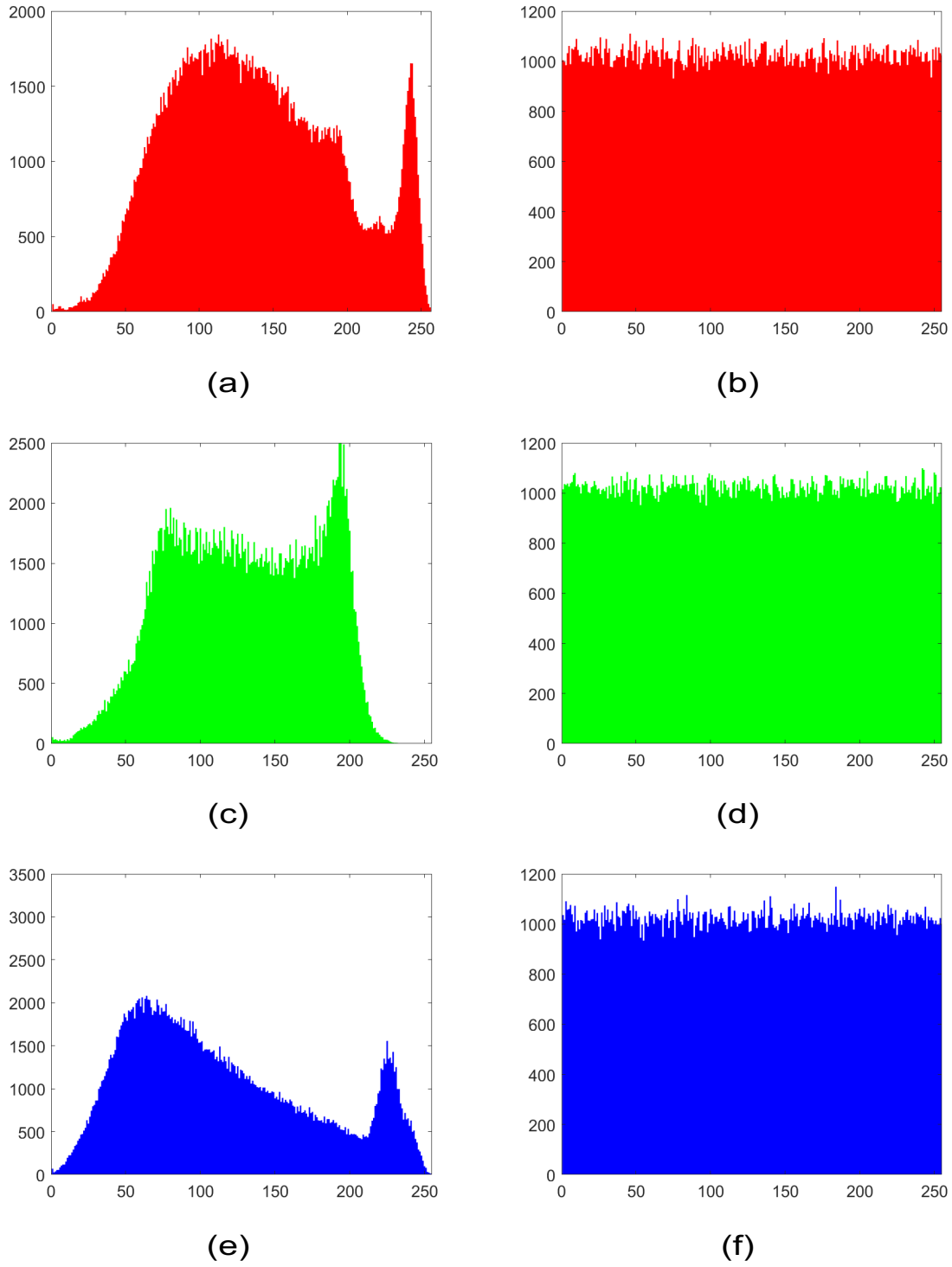


Figura 25. Histogramas de la imagen Mandril RGB de la Fig. 17(e): (a) Mandril normal en el canal R, (b) Mandril encriptada en el canal R, (c) Mandril normal en el canal G, (d) Mandril encriptada en el canal G, (e) Mandril normal en el canal B, (f) Mandril encriptada en el canal B.

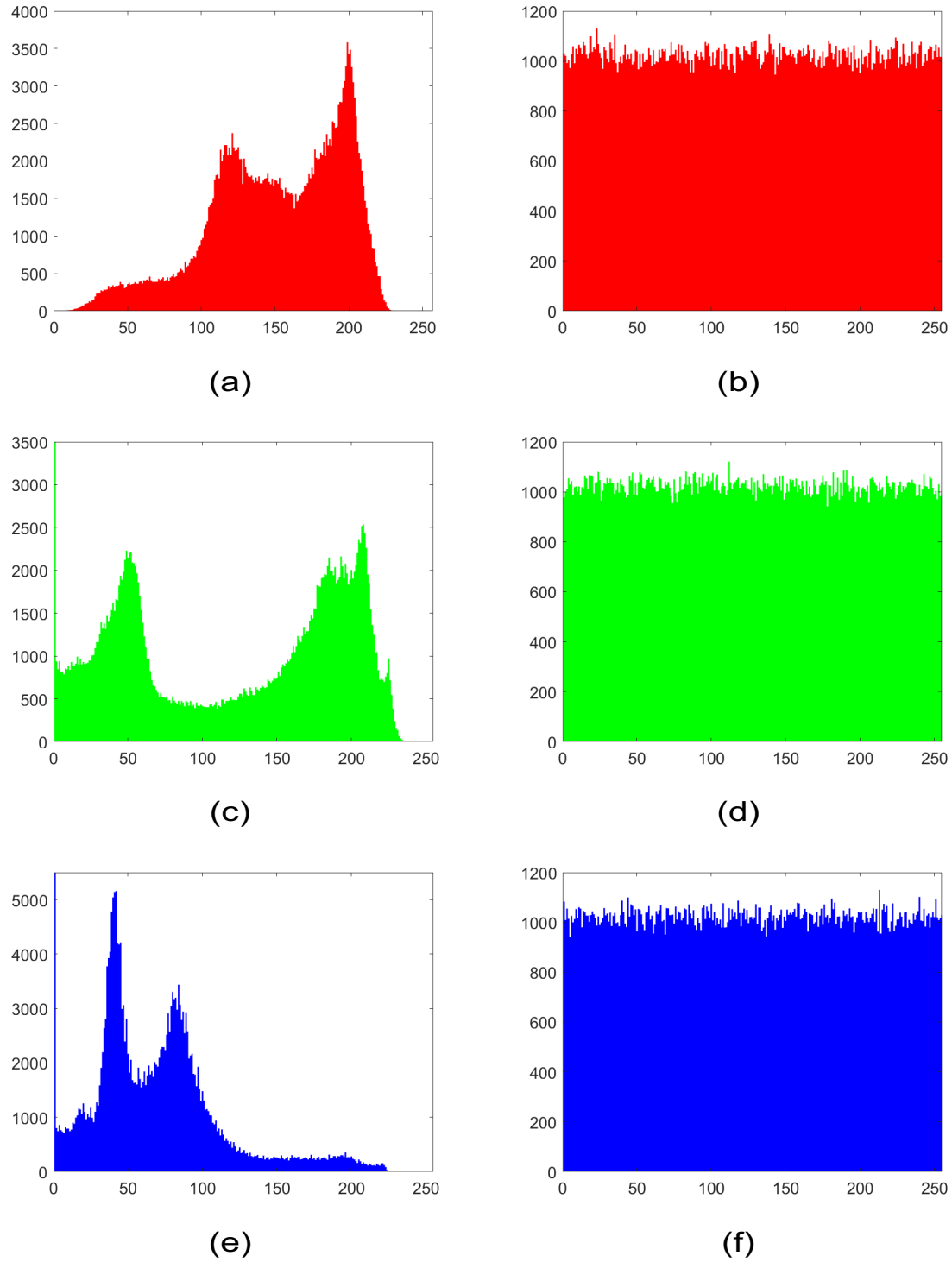


Figura 26. Histogramas de la imagen Peppers RGB de la Fig. 17(f): (a) Peppers normal en el canal R, (b) Peppers encriptada en el canal R, (c) Peppers normal en el canal G, (d) Peppers encriptada en el canal G, (e) Peppers normal en el canal B, (f) Peppers encripta en el canal B.

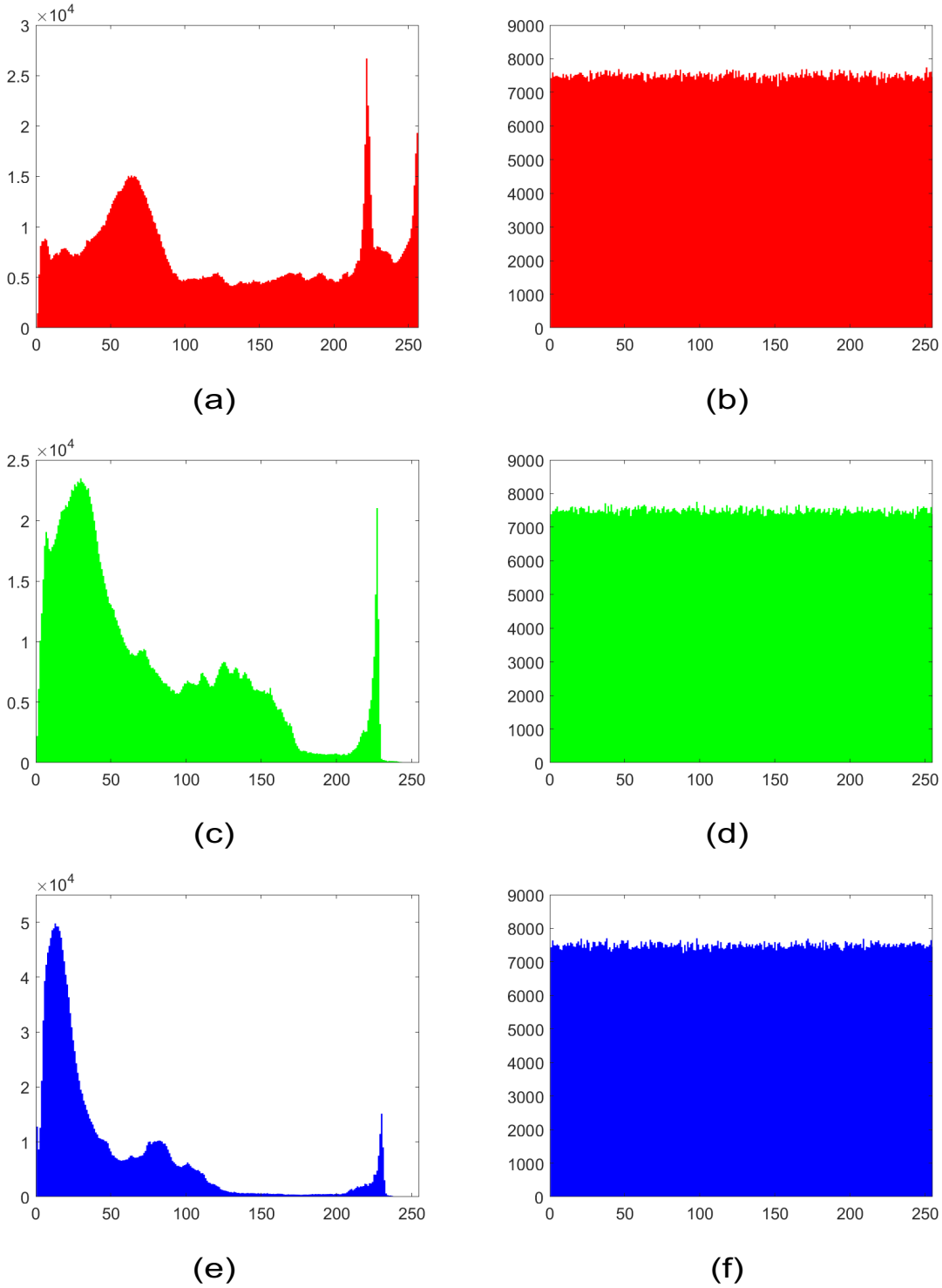


Figura 27. Histogramas de la imagen Girl 1 de la Fig. 19(a): (a) Girl 1 normal en el canal R, (b) Girl 1 encriptada en el canal R, (c) Girl 1 normal en el canal G, (d) Girl 1 encriptada en el canal G, (e) Girl 1 normal en el canal B, (f) Girl 1 encripta en el canal B.

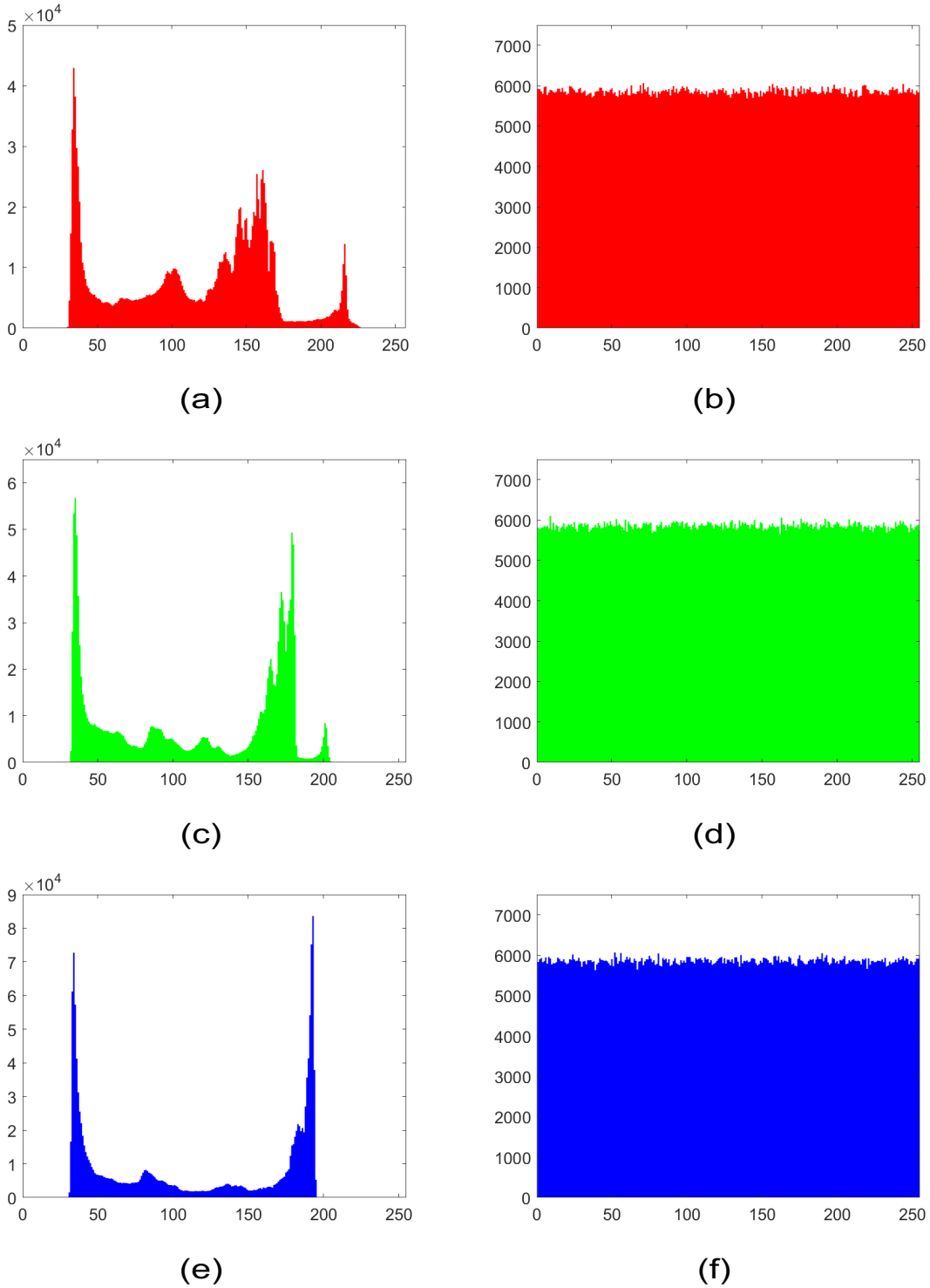
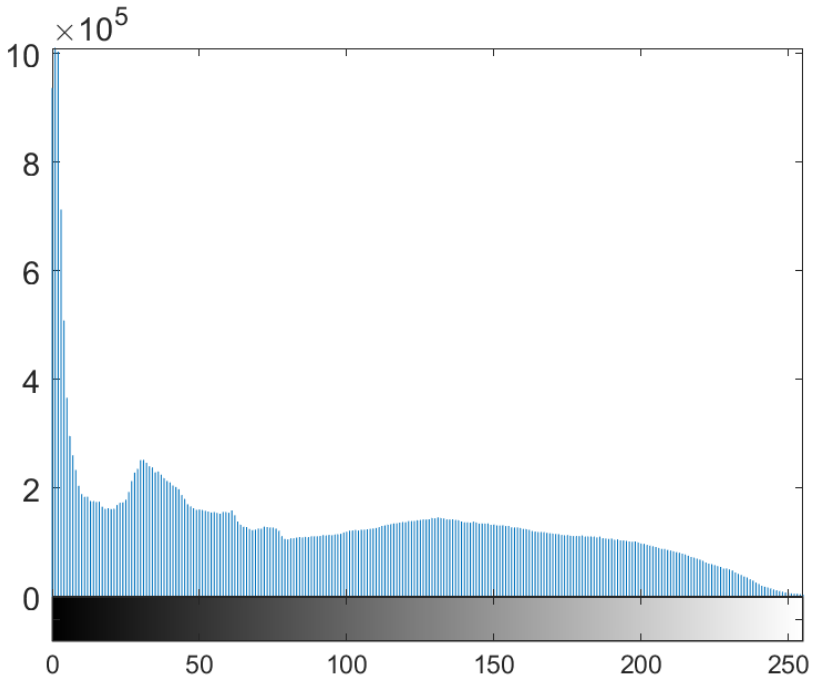
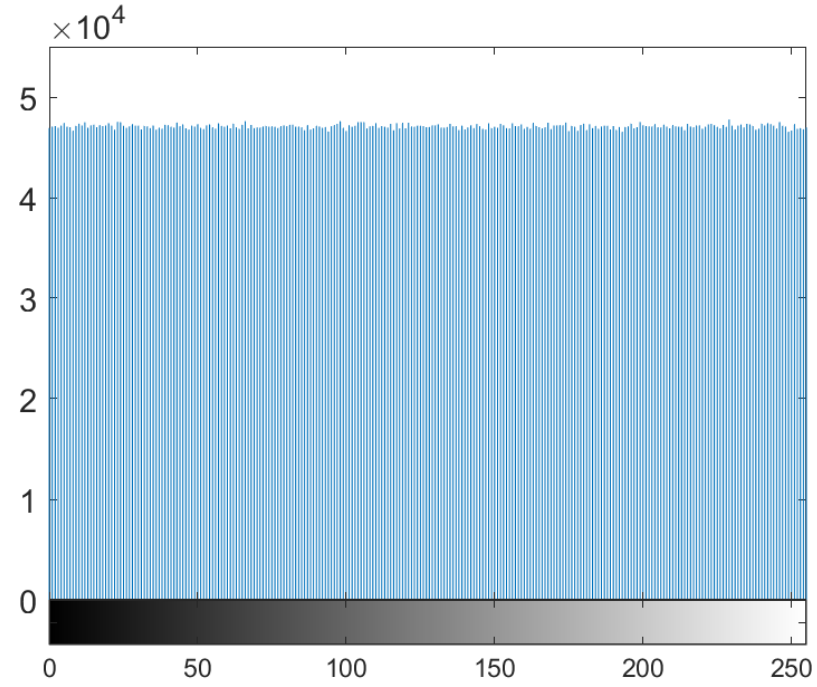


Figura 28. Histogramas de la imagen Girl 2 de la Fig. 19(b): (a) Girl 2 normal en el canal R, (b) Girl 2 encriptada en el canal R, (c) Girl 2 normal en el canal G, (d) Girl 2 encriptada en el canal G, (e) Girl 2 normal en el canal B, (f) Girl 2 encripta en el canal B.



(a)



(b)

Figura 29. Histogramas de la imagen de Girl 3 de la Fig. 19(c): (a) Girl 3 normal, (b) Girl 3 encriptada.

IV.2 Ataque de fuerza bruta

El tamaño del espacio de claves es directamente proporcional al número de condiciones iniciales y parámetros de control de los mapas caóticos utilizados en el esquema propuesto. La seguridad del criptosistema se basa en la seguridad de la clave secreta [71]. La Tabla IV muestra el espacio de claves del criptosistema propuesto utilizando los mapas caóticos mejorados $mod1023$ Bernoulli, Tent, Zigzag y Logistic 1D.

Tabla IV. Espacio de claves de diferentes PRNG.

Mapa Caótico	**Espacio de Clave**
Bernoulli	2^{192}
Tent	2^{128}
Zigzag	2^{128}
Logistic 1D	2^{128}
Trabajos relacionados	
Logistic 1D [12]	2^{128}
PELM [23]	2^{128}
PLM [66]	2^{136}
Multi modal [14]	2^{159}
3D-PLM [18]	2^{189}
Logistic 1D map [22]	2^{199}
Logistic 1D map and 1DLSE [7]	2^{256}
Fuzzy Multi Modular [21]	2^{325}
STCS and L-LCS [24]	2^{420}
Probabilistic image encryption [72]	2^{256}

Se puede observar que todos los mapas caóticos mejorados 1D propuestos cumplen el criterio de espacio de claves mínimo según [13]. El mapa Bernoulli tiene el mayor espacio de claves. De acuerdo con la Tabla IV, el espacio de claves de los mapas caóticos 1D mejorados es competitivo con los reportados en el estado del arte.

IV.3 Resultados de encriptamiento de imágenes

IV.3.1 Análisis de Correlación de pixeles adyacentes

El análisis de correlación de pixeles adyacentes se utiliza para cuantificar el grado de asociación entre los valores de los píxeles de la imagen. Normalmente, los píxeles adyacentes en una imagen original se caracterizan por poseer fuertes relaciones [71]. Los métodos criptográficos eficaces deben disminuir esta relación en la imagen encriptada para evitar cualquier ataque de análisis de relación de píxeles. En general, una menor correlación entre los píxeles vecinos en la imagen encriptada significa un algoritmo criptográfico más fuerte [71]. Las ecuaciones detalladas para calcular el coeficiente de correlación r_{xy} se muestra en la ecuación 6 [61].

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{6}$$

Donde

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))\,(y_i - E(y)) \tag{7}$$

donde $cov(x,y)$ es la covarianza, $D(x)$ es la varianza, x e y denotan los valores de escala de nivel de gris en la imagen digital analizada. Para este caso numérico, se utilizaron las siguientes formas discretas:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{8}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x)) \tag{9}$$

donde $E(x)$ es la media de los niveles de gris de los píxeles.

La Fig. 30 muestra la distribución de correlación de 5000 pares de píxeles adyacentes en dirección horizontal (a), vertical (c) y diagonal (e) de la Figura 17(a) de Lena de 512×512 píxeles en escala de grises y su correspondiente criptograma obtenido con el estado x_{n+1} de la función Logística 1D mejorada con $mod1023$ mostrado en la Figura 18(a). Los píxeles adyacentes de una imagen original están muy correlacionados; como se puede ver en la Fig. 30(a), 30(c) y 30(e). En cuanto a la correlación de los píxeles de la imagen encriptada, en la Fig. 30(b), 30(d), y 30(f), la correlación entre los píxeles adyacentes es completamente diferente, lo que indica que la dispersión se produce (al azar) en todo el plano (x, y).

En la Figura 31, se muestran los diagramas de correlación de píxeles adyacentes para la Figura 17(b) de Camera Man y también su respectivo criptograma usando la función logistic 1D mejorada. En la imagen original, 31(a), 31(c) y 31(e) se pueden observar que los pixeles adyacentes presentan una gran correlación. En cambio, la correlación de los pixeles adyacentes de la imagen encriptada es completamente dispersa como se observa en las Figuras 31(b), 31(d) y 31(f).

De la misma forma para la imagen de la Figura 17(c) Mandril 512x512 se muestran en la Figura 32 los diagramas de correlación de píxeles adyacentes horizontal (a), vertical (c) y diagonal (e). En cuanto a su criptograma mostrado en la Figura 18(c) también se presentan los diagramas de correlación de píxeles adyacentes horizontal 32 (b), vertical 32 (d) y diagonal 32 (f).

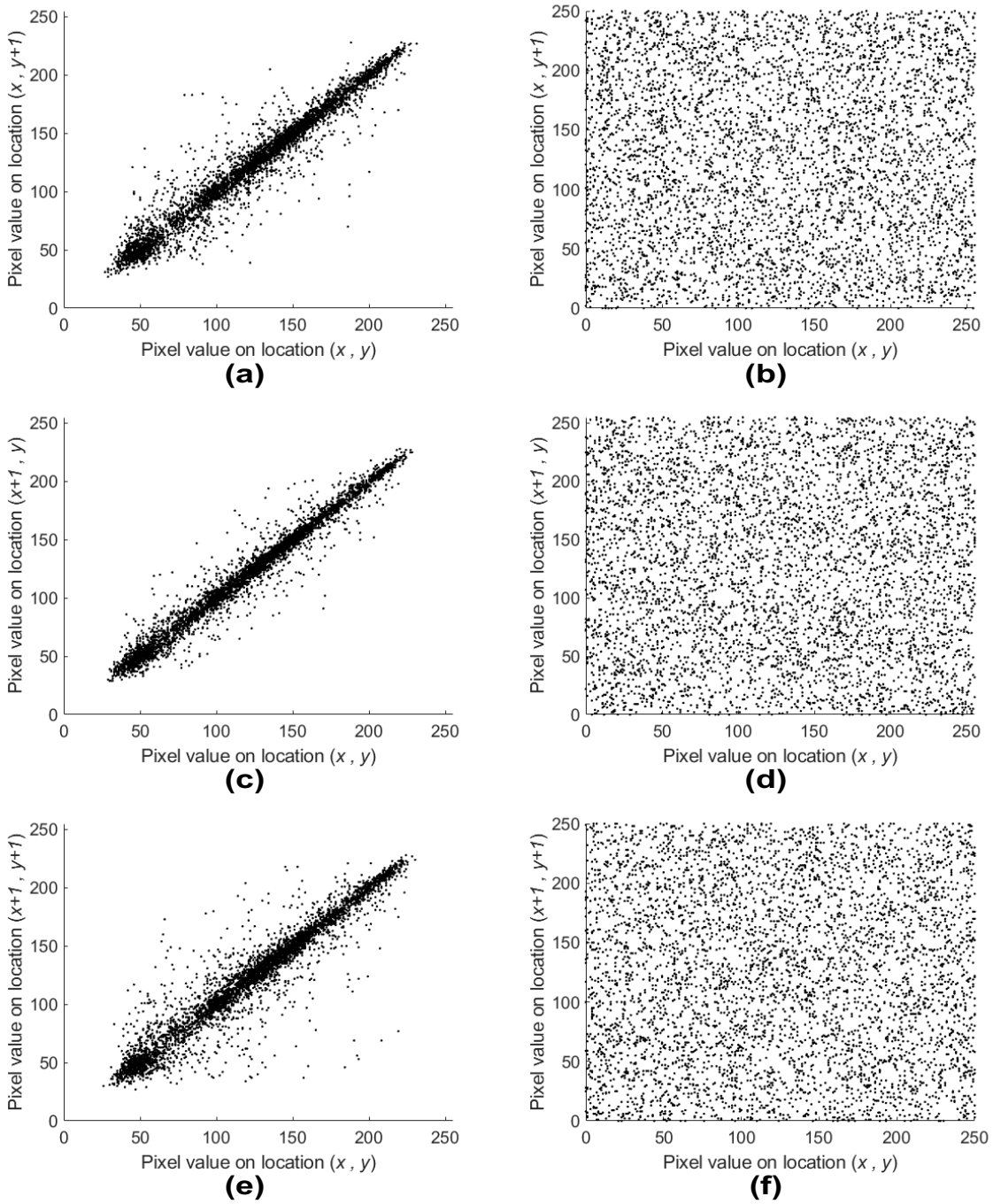


Figura 30. Correlación de píxeles adyacentes en Lena de 512 x 512 píxeles y en la imagen encriptada usando Logístic 1D mejorado: (a) Píxeles adyacentes horizontalmente en original, (b) Píxeles adyacentes horizontalmente en la imagen encriptada, (c) Píxeles adyacentes verticalmente en original, (d) Píxeles adyacentes verticalmente en encriptada, (e) Píxeles adyacentes diagonalmente en original, y (f) Píxeles adyacentes diagonalmente en encriptada.

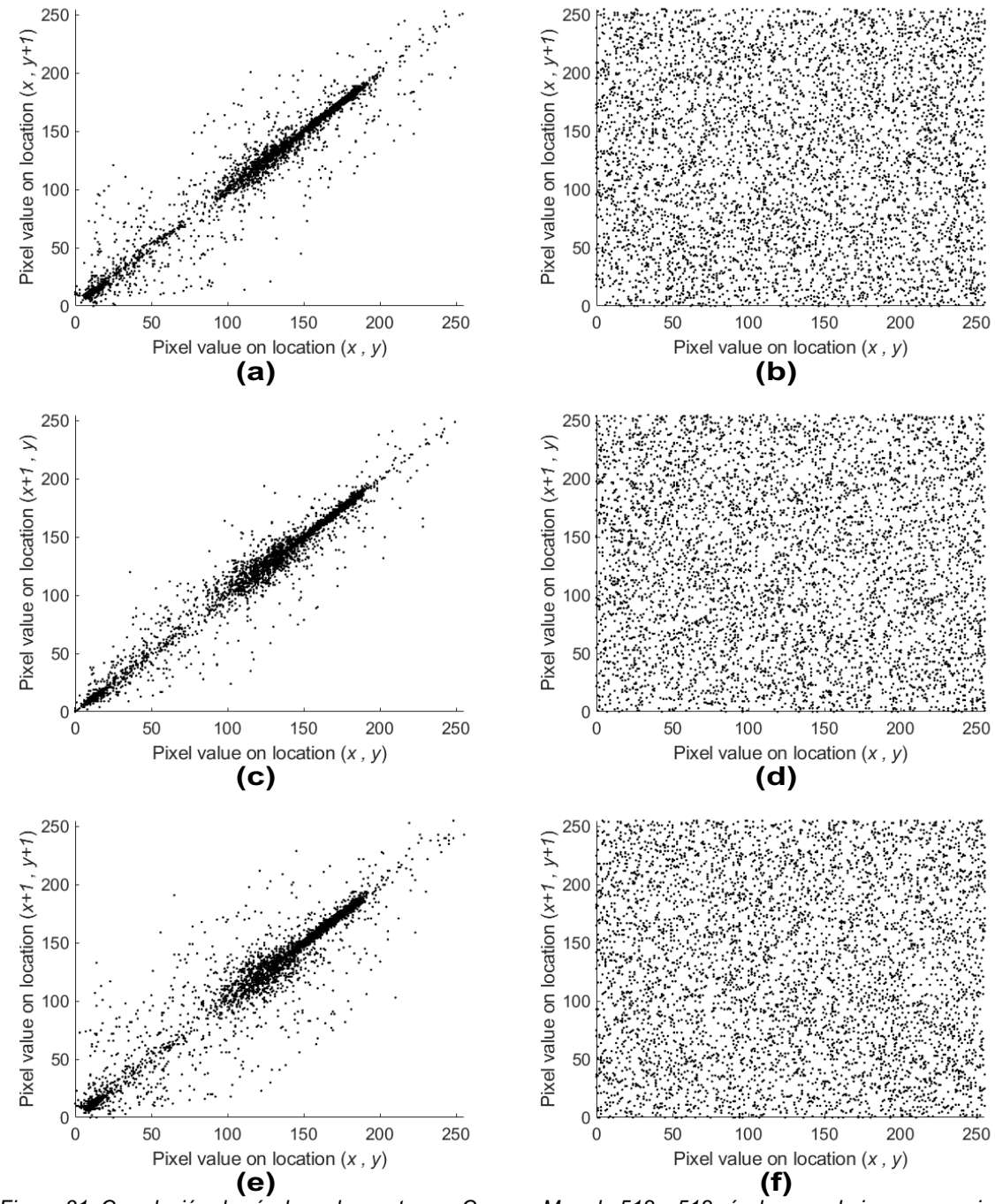


Figura 31. Correlación de píxeles adyacentes en Camera Man de 512 x 512 píxeles y en la imagen encriptada usando Logístic 1D mejorado: (a) Píxeles adyacentes horizontalmente en original, (b) Píxeles adyacentes horizontalmente en la imagen encriptada, (c) Píxeles adyacentes verticalmente en original, (d) Píxeles adyacentes verticalmente en encriptada, (e) Píxeles adyacentes diagonalmente en original, y (f) Píxeles adyacentes diagonalmente en encriptada.

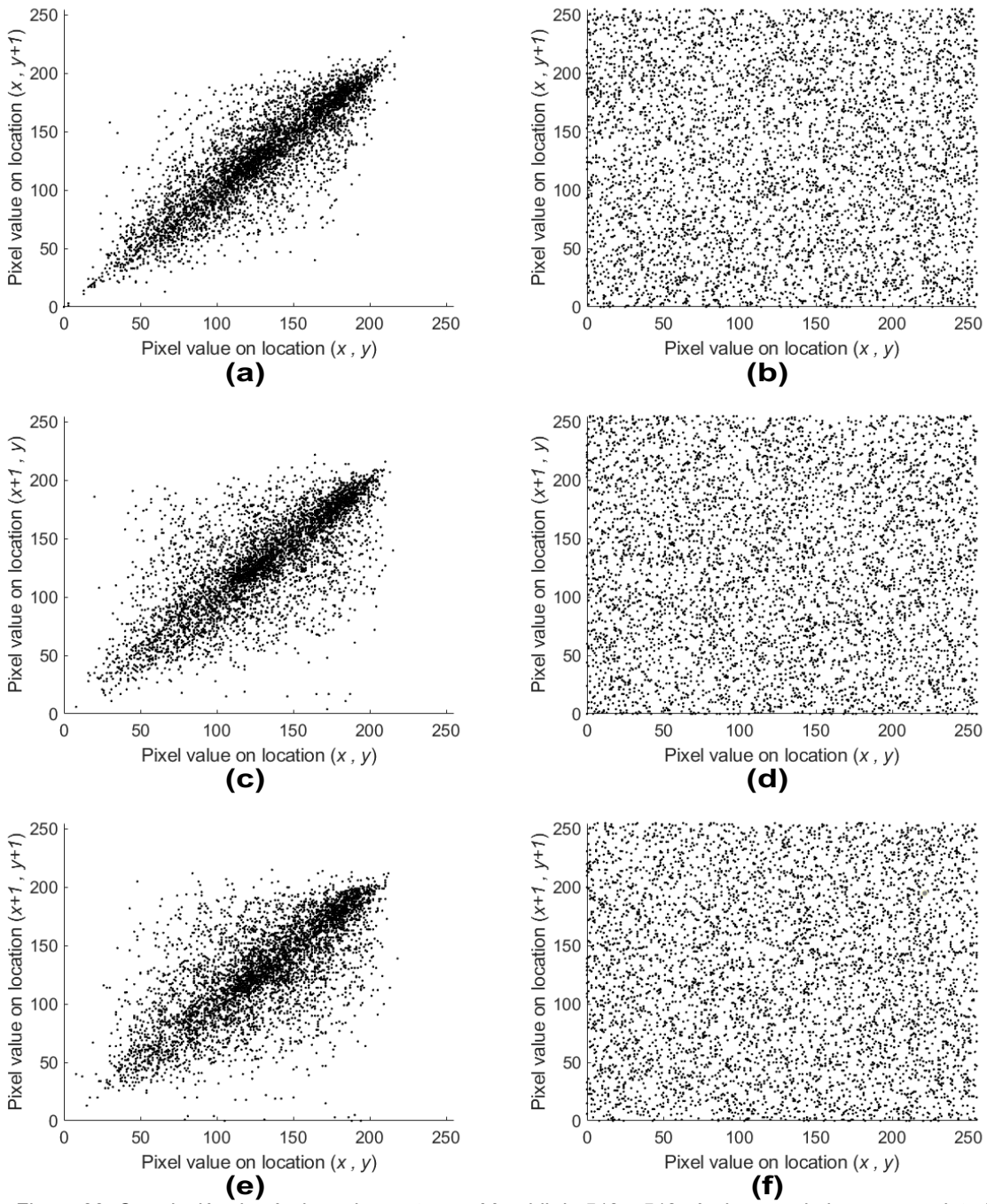


Figura 32. Correlación de píxeles adyacentes en Mandril de 512 x 512 píxeles y en la imagen encriptada usando Logístic 1D mejorado: (a) Píxeles adyacentes horizontalmente en original, (b) Píxeles adyacentes horizontalmente en la imagen encriptada, (c) Píxeles adyacentes verticalmente en original, (d) Píxeles adyacentes verticalmente en encriptada, (e) Píxeles adyacentes diagonalmente en original, y (f) Píxeles adyacentes diagonalmente en encriptada.

Para realizar una comparación con las imágenes existentes en el estado del arte de Lena 256×256 y Lena 512×512 se muestra la Tabla V de los coeficientes de correlación r_{xy} de los píxeles adyacentes (horizontales, verticales y diagonales) de la imagen original e imagen cifrada utilizando varios mapas caóticos. Debido a su similitud, los coeficientes de correlación r_{xy} en la imagen original de 256×256 y 512×512 píxeles son cercanos a uno. Sin embargo, el coeficiente de correlación r_{xy} en las imágenes cifradas, en cambio, está muy cerca de cero, lo que indica que las imágenes originales y encriptadas no tienen ninguna similitud, confirmando que son absolutamente diferentes. Se ha confirmado que el esquema propuesto funciona para cualquier tamaño de imagen y que los resultados son comparables al estado del arte actual.

Tabla V. Coeficientes de correlación de pixeles adyacentes de imágenes originales y encriptadas de Lena 25×256 *y Lena* 512×512.

Mapa caótico	**Correlation coefficients (r_{xy}).**			
	Lena 256 × 256	**Cryptogram 256 × 256**	**Lena 512 × 512**	**Cryptogram 512 × 512**
Bernoulli	0.9389	-0.0014	0.9718	-0.0010
	0.9684	-0.0012	0.9849	0.0019
	0.9135	0.0083	0.9592	0.0023
Tent	0.9389	0.0021	0.9718	-0.00005
	0.9684	0.0030	0.9849	0.0002
	0.9135	0.0039	0.9542	0.0008
Zigzag	0.9389	-0.0036	0.9718	-.0021
	0.9684	0.0023	0.9849	0.0013
	0.9135	-0.0009	0.9592	0.0023
Logistic 1D	0.9389	-0.0060	0.9718	-0.0004
	0.9684	0.0034	0.9849	-0.0000
	0.9135	0.0001	0.9592	-0.0022
Trabajo Relacionado				
Henon [61]	0.9242	0.0201	0.9855	-0.0021
	0.9529	0.0160	0.9866	0.0203
	0.8936	-0.0165	0.9789	0.0081
Tinkerbell [61]	0.9242	0.0033	0.9855	-0.0003

Mapa caótico	Correlation coefficients (r_{xy}).			
	Lena 256 × 256	**Cryptogram 256 × 256**	**Lena 512 × 512**	**Cryptogram 512 × 512**
	0.9529	-0.0083	0.9866	-0.0006
	0.8936	-0.0141	0.9789	-0.0071
Chen [61]	0.9242	-0.0246	0.9855	0.0081
	0.9529	-0.0146	0.9866	-0.0070
	0.8936	0.0105	0.9789	-0.0070
Rössler [61]	0.9242	0.0005	0.9855	0.0039
	0.9529	-0.0122	0.9866	-0.0382
	0.8936	-0.0287	0.9789	0.0108
Logistic 2D [61]	0.9242	0.0079	0.9855	0.0079
	0.9529	0.0011	0.9866	-0.001
	0.8936	0.0369	0.9789	0.0369
Fuzzy Multi Modular [21]	0.9747	-0.0050	-	-
	0.9513	-0.0012	-	-
	0.9276	0.0019	-	-
Multi modal [14]	0.9852	0.0226	-	-
	0.9658	0.0022	-	-
	0.9506	-0.0081	-	-
Chen [62]	0.9488	0.0005	-	-
	0.9477	0.0005	-	-
	0.9047	0.0001	-	-
5D Hyperchaotic [64]	0.9705	-0.0052	-	-
	0.9414	0.0031	-	-
	0.9136	-0.0003	-	-
Lorenz and S-Box [54]	0.4454 (size 50 × 64)	-0.0636 (size 50 × 64)	-	-
	0.9445 (size 50 × 64)	0.0465 (size 50 × 64)	-	-
	0.4454 (size 50 × 64)	0.0669 (size 50 × 64)	-	-
New chaotic signal [25]	-	-	0.9654	-0.0001
	-	-	0.9623	0.0003
	-	-	0.9469	0.0001
STCS and L-LCS [24]	-	-	0.9845	-0.0005
	-	-	0.9920	0.0029

Mapa caótico	Correlation coefficients (r_{xy}).			
	Lena 256 × 256	**Cryptogram 256 × 256**	**Lena 512 × 512**	**Cryptogram 512 × 512**
	-	-	0.9751	0.0030
IECA [73]	-	-	0.9719	0.0080
	-	-	0.9850	0.0896
	-	-	0.9593	0.0105

Los coeficientes de correlación r_{xy} obtenidos entre la imagen original y la imagen encriptada se muestran en la Tabla VI, se puede observar, el coeficiente de correlación r_{xy} es extremadamente cercano a cero en todos los casos, lo que indica que las imágenes original y encriptada no tienen ningún parecido y confirma su completa desigualdad. Además, el enfoque propuesto ha sido validado para funcionar con cualquier tamaño de imagen.

Tabla VI. Coeficiente de correlación (r_{xy}) entre la imagen original de Lena y la encriptada.

Mapa Caótico	**Lena 256 × 256**	**Lena 512 × 512**
Bernoulli	0.0003	-0.0009
Tent	0.0012	-0.0005
Zigzag	-0.0004	0.0011
Logistic 1D	-0.0043	0.0011
Trabajo relacionado		
Hénon [61]	-0.0054	-0.0011
Tinkerbell [61]	0.0027	0.0017
Chen [61]	0.0020	0.0009
Rössler [61]	-0.0068	-0.0042
Logistic 2D [61]	-0.0001	0.0001
Substitution-Box [74]	-	0.0008
Chaotic artificial neurons [55]	-	0.0009

Se puede observar que el coeficiente de correlación r_{xy} es muy cercano a cero en todas las situaciones, lo que indica que las imágenes originales y encriptadas no tienen ninguna similitud y confirma que son absolutamente diferentes.

IV.3.2 Análisis de Entropía

La entropía de la información es un criterio que cuantifica la aleatoriedad de los datos y se utiliza para determinar la seguridad del algoritmo de encriptado [47]. La entropía $H(s)$ puede calcularse mediante la ecuación (10):

$$H(s) = \sum_{i=0}^{2^N-1} P(s_i) \cdot Log_2\left(\frac{1}{P(s_i)}\right) bit, \quad (10)$$

Donde $P(s_i)$ es la probablidad del símbolo s_i.

La tabla VII muestra la entropía (bit/símbolo) de las imágenes originales (entrada) y los criptogramas (salida) en escala de grises de las bases de datos USC-SIPI y Open Images Dataset V6+. Estos resultados se obtuvieron con el mapa Logístico 1D mejorado utilizando la función $mod1023$. Se puede observar que la entropía se aproxima a 8 en ambos tamaños de imagen, que es la entropía ideal. También se comprueba que la entropía de salida (imagen encriptada) es independiente de la entropía de entrada y del tamaño de las imágenes.

Tabla VII. Entropía (bit/símbolo) de las imágenes originales y encriptadas en escala de grises de las bases de datos USC-SIPI y Open Images Dataset V6+.

Imagen	Tamaño (pixels)	Original	Encriptada
Lena	512 × 512	7.4455	7.9993
Camera man	512 × 512	7.0480	7.9993
Mandrill	512 × 512	7.3583	7.9993
Girl 3	4256 × 2832	7.5964	7.9999

La tabla VIII muestra la entropía (bit/símbolo) de las imágenes originales (entrada) y de las imágenes encriptadas (salida) en formato RGB de las bases de datos USC-SIPI y Open Images Dataset V6+. Estos resultados se obtuvieron con el mapa Logistic 1D mejorado utilizando la función $mod1023$. Se observa que la entropía se aproxima a 8 en todos los casos, que es la entropía ideal. También se comprueba que la entropía de salida (imagen encriptada) es independiente de la entropía de entrada y del tamaño de las imágenes.

Tabla VIII. Entropía (bit/símbolo) de las imágenes originales y encriptadas en formato RGB de las bases de datos USC-SIPI y Open Images Dataset V6+.

Imagen	Tamaño (pixels)	Imagen Original				Imagen Encriptada			
		R	G	B	Promedio	R	G	B	Promedio
Lena	512 × 512	7.2531	7.5952	6.9686	7.2723	7.9993	7.9992	7.9993	7.9992
Mandrill	512 × 512	7.7067	7.4744	7.7522	7.6444	7.9993	7.9993	7.9992	7.9992
Peppers	512 × 512	7.3388	7.4962	7.0583	7.2977	7.9992	7.9991	7.9992	7.9992
Girl 1	1200 × 1600	7.8600	7.4100	6.7600	7.3443	7.9999	7.9999	7.9999	7.9999
Girl 2	1500 × 1000	7.1134	6.6911	6.4578	6.7541	7.9999	7.9999	7.9999	7.9999

La entropía obtenida con los cuatro mapas caóticos mejorados utilizando la función $mod1023$ se muestra en la Tabla IX, junto con una comparación del estado del arte utilizando la imagen de Lena con 256×256 y 512×512 píxeles en el formato de 8 bits. Como se esperaba, la entropía obtenida utilizando el mapa logístico 1D es la más cercana a 8 en ambos tamaños de imagen, que es la entropía ideal. Además, se muestra en la Tabla IX, que la entropía no se ve afectada por el tamaño de la imagen, y los resultados son similares a los reportados en el estado del arte.

Tabla IX. Comparación de la entropía del criptograma de Lena (bit/símbolo) con otros trabajos.

Mapa caótico	**Lena 256 × 256**	**Lena 512 × 512**
Bernoulli	7.9989	7.9997
Tent map	7.9991	7.9997
Zigzag map	7.9989	7.9997
Logistic 1D	7.9990	7.9998
Related work		
STCS and L-LCS [24]	-	7.9992
Multimodal [14]	7.9989	-
Hénon [61]	7.9973	7.9992
5D Hyperchaotic [64]	7.9973	-
Fuzzy Multi Modular [21]	7.9972	-
Chen [62]	7.9972	-
Tinkerbell [61]	7.9969	7.9993
Logistic 1 [12]	7.9948	7.9972
3D-PLM [18]	-	7.9982
Logistic 1D [22]	-	7.9991
New chaotic signal [25]	-	7.9994
Substitution-Box [74]	-	7.9957
IECA [73]	-	7.9646
Probabilistic image encryption [72]	-	7.9933
Lorenz and S-Box [56]	7.9401 (50 × 64)	

IV.3.3 Ataque Diferencial

Se utilizan dos métricas para llevar a cabo el análisis contra los ataques diferenciales y comprender las diferencias entre las imágenes encriptadas. La primera es la tasa de cambio de números de píxeles (NPCR), y la otra es el promedio uniforme de cambio de intensidad (UACI) [47].

En la Tabla X, se puede observar que los cuatro mapas caóticos 1D mejorados con $mod1023$, propuestos en este trabajo, superaron la prueba NPCR según los valores críticos establecidos en [75]. Por lo tanto, el 100% de los mapas caóticos 1D mejorados pasaron todas las pruebas NPCR y son competitivos con los reportados en el estado del arte.

Tabla X. NPCR del criptograma de Lena 512x512 píxeles y comparación con el estado del arte.

Criptograma Tamaño		**Valores Críticos**		
512 × 512		N0.05=99.5893%	N0.01=99.5810%	N0.001=99.5717%
Mapa Caótico	**Promedio NPCR (%)**	**Resultados**		
Bernoulli	99.6087	Passed	Passed	Passed
Tent	99.5942	Passed	Passed	Passed
Zigzag	99.6045	Passed	Passed	Passed
Logistic 1D	99.6084	Passed	Passed	Passed
Related work				
Hénon [61]	99.8093	Passed	Passed	Passed
Chen [62]	99.8031	Passed	Passed	Passed
Tinkerbell [61]	99.7990	Passed	Passed	Passed
Tent map [42]	99.6300	Passed	Passed	Passed
New chaotic signal [25]	99.6200	Passed	Passed	Passed
Fuzzy Multi Modular [21]	99.6190	Passed	Passed	Passed
5D Hyperchaotic [64]	99.6122	Passed	Passed	Passed
3D-PLM [18]	99.6100	Passed	Passed	Passed
Logistic 1D [22]	99.6100	Passed	Passed	Passed
Logistic 1D [12]	99.6100	Passed	Passed	Passed
STCS and L-LCS [24]	99.6084	Passed	Passed	Passed
PELM [23]	99.5774	Failed	Failed	Passed
Multimodal [14]	99.0000	Failed	Failed	Failed
Probabilistic image encryption [72]	99.6314	Passed	Passed	Passed
IECA [73]	99.7834	Passed	Passed	Passed

En la Tabla XI, se puede observar que los cuatro mapas caóticos propuestos en este trabajo pasaron la prueba UACI de acuerdo con los valores críticos establecidos por [75]. Por lo tanto, el 100% de los mapas caóticos 1D mejorados usando $mod\ 1023$ pasaron todas las pruebas UACI y son competitivos con los reportados en el estado del arte.

Tabla XI. Análisis UACI del criptograma de la imagen de Lena $512x512$ *y comparación con el estado del arte.*

Tamaño de Criptograma		**Valores Criticos**		
512×512		U0.05−=33.3730%, U0.05+=33.5541%,	U0.01−=33.3445%, U0.01+=33.5826%,	U0.001−=33.3115%, U0.001+=33.6156%,
Mapa Caótico	**Promedio UACI (%)**	**Resultado**		
Bernoulli	33.3962	Passed	Passed	Passed
Tent	33.3864	Passed	Passed	Passed
Zigzag	33.4805	Passed	Passed	Passed
Logistic 1D	33.4850	Passed	Passed	Passed
Trabajo Relacioando				
Chen [62]	33.6236	Passed	Passed	Passed
Fuzzy Multi Modular [21]	33.4861	Passed	Passed	Passed
Hénon [61]	33.4805	Passed	Passed	Passed
STCS and L-LCS [24]	33.4783	Passed	Passed	Passed
New chaotic signal [25]	33.4700	Passed	Passed	Passed
5D Hyperchaotic [64]	33.4573	Passed	Passed	Passed
Tinkerbell [61]	33.4524	Passed	Passed	Passed
3D-PLM [18]	33.4500	Passed	Passed	Passed
Logistic 1D [22]	33.4500	Passed	Passed	Passed
Logistic 1D [12]	33.3600	Passed	Passed	Passed
Tent map [42]	33.2800	Passed	Passed	Passed
PELM [23]	33.3014	Failed	Failed	Passed
Multimodal [14]	34.8353	Failed	Failed	Failed
Probabilistic image encryption [72]	33.5513	Passed	Passed	Passed
IECA [73]	33.8412	Failed	Failed	Failed

IV.4 Análisis de desempeño

El rendimiento de los PRNG depende de muchos factores como lo son la precisión de la CPU, la cantidad de operaciones aritméticas y lógicas, las dimensiones de los mapas caóticos, el hardware y el software. Por ello, a continuación, se presenta un análisis del rendimiento de los PRNGs propuestos.

IV.4.1 Análisis de Rendimiento

La Tabla XII muestra el rendimiento de los PRNGs, el reloj de la CPU y el Software utilizado. Se puede observar que el Logistic 1D mejorado que se ejecuta en un ordenador personal con una CPU de 2.9 GHz con MAC OS X 10.13.6 y que realiza la función $mod\ 1023$ tiene un rendimiento de 47.4403 Mbit/s, y es el que tiene el mejor rendimiento que los reportados en el estado del arte. A continuación, le siguen el mapa Tent mejorado con 43.3298 Mbit/s, el mapa de Bernoulli mejorado con 37.7756 Mbit/s y el mapa Zigzag mejorado con 27.5741 Mbit/s, que también superan los trabajos relacionados.

En cambio, cuando los PRNGs propuestos se implementan en un sistema embebido como Raspberry Pi 4 con un CPU de 1.5 GHz, se puede observar que el mapa Logistic 1D mejorado tiene el mejor rendimiento con 10.5349 Mbit/s, seguido por el mapa Tent mejorado con 10.0269 Mbit/s, el siguiente mapa es el Bernoulli mejorado con 8.66 Mbit/s y finalmente el mapa Zigzag mejorado con 6.5314 Mbit/s. Además, se puede observar que los resultados de rendimiento del RPi4 utilizando la función $mod\ 1023$ son ligeramente superiores a los de $mod\ 255$, excepto cuando se utiliza el mapa Bernoulli. Por lo tanto, de acuerdo con los resultados presentados en la Tabla XII, la eficiencia del modelo (PRNG), en este caso particular, lo estamos llamando como "throughtput", que se expresa en Mbit/segundos, depende de: las

operaciones aritméticas y lógicas realizadas por los mapas caóticos mejorados, los recursos de software como el software de programación y el sistema operativo. También depende de los recursos de hardware, como la frecuencia de reloj de la CPU, la memoria RAM y la arquitectura del microprocesador. Por ello, se considera que la eficiencia (throughtput) del PRNG no se ve afectada por la entropía de la información de entrada, ya que son independientes.

Tabla XII. Análisis del rendimiento de PRNG.

Mapa Caótico	**Hardware y Software**		**Comparison of throughtput**	
PRNG	**CPU Reloj (GHz)**	**Software**	**Throughput (Mbit/s)** mod **255**	**Throughput (Mbit/s)** mod **1023**
Bernoulli	1.500	Python 2.7.16	8.8362	8.6600
Tent	1.500	Python 2.7.16	9.8856	10.0269
Zigzag	1.500	Python 2.7.16	6.4351	6.5314
Logistic 1D	1.500	Python 2.7.16	10.4120	10.5349
Bernoulli	2.900	Python 3.8.8	38.7991	37.7756
Tent	2.900	Python 3.8.8	43.8972	43.3298
Zigzag	2.900	Python 3.8.8	27.6177	27.5741
Logistic 1D	2.900	Python 3.8.8	47.7967	47.4403
Trabajos relacionados				
Logistic 1D [12]	2.000	Matlab v7.6	24.0000	-
PLM [66]	3.300	Visual C+ 6.0	20.5383	-
5D hyperchaotic on FPGA [40]	0.138	Vivado 2018.3	15.3700	-
Hénon [47]	2.700	Python 3.5.2	12.8008	-
Chen [47]	2.700	Python 3.5.2	12.8000	-
Tinkerbell [47]	2.700	Python 3.5.2	9.1491	-
Logistic 2D [47]	2.700	Python 3.5.2	9.1491	-
Rössler [47]	2.700	Python 3.5.2	6.4000	-
PELM [23]	2.000	Matlab v7.6	1.7000	-
Discrete [43]	1.900	-	1.2600	-
Tinkerbell [44]	2.800	-	0.4901	-
Complex [45]	2.100	GCC	0.4844	-

Los resultados de rendimiento obtenidos con los PRNGs propuestos demuestran que pueden utilizarse para el encriptado de imágenes en tiempo real implementado en sistemas embebidos, y para dispositivos de telecomunicaciones modernos, como smartphones, tabletas, dispositivos usables, que tienen una arquitectura de hardware similar. Además, estos dispositivos están preparados para aplicaciones IoT y podrían utilizarse en la videovigilancia remota a través de Internet. También tiene aplicación potencial en comunicaciones privadas a través de aplicaciones de mensajería de redes sociales, como WhatsApp, Signal, Telegram, entre otros mensajeros que funcionan en tiempo real a través de internet.

IV.5 Implementación en Raspberry Pi 4

El sistema de comunicación IoT que utiliza el protocolo MQTT se implementó en una Raspberry Pi 4, como sistema embebido, y en una computadora personal. Esta red de comunicación se muestra en la Figura 33, donde se pueden observar tres elementos básicos, los cuales son: un dispositivo subscriptor (receptor), un dispositivo que publica (transmisor) y un intermediario. En este sistema la información es vulnerable, al contar con una limitada seguridad. El sistema propuesto, incluye la elección de algún mapa caótico mejorado, lo que agrega seguridad a la imagen transmitida a través de internet en dispositivos IoT, y el intruso sin conocer la clave correcta no podrá reconocer la imagen, como se observa en la Figura 34.

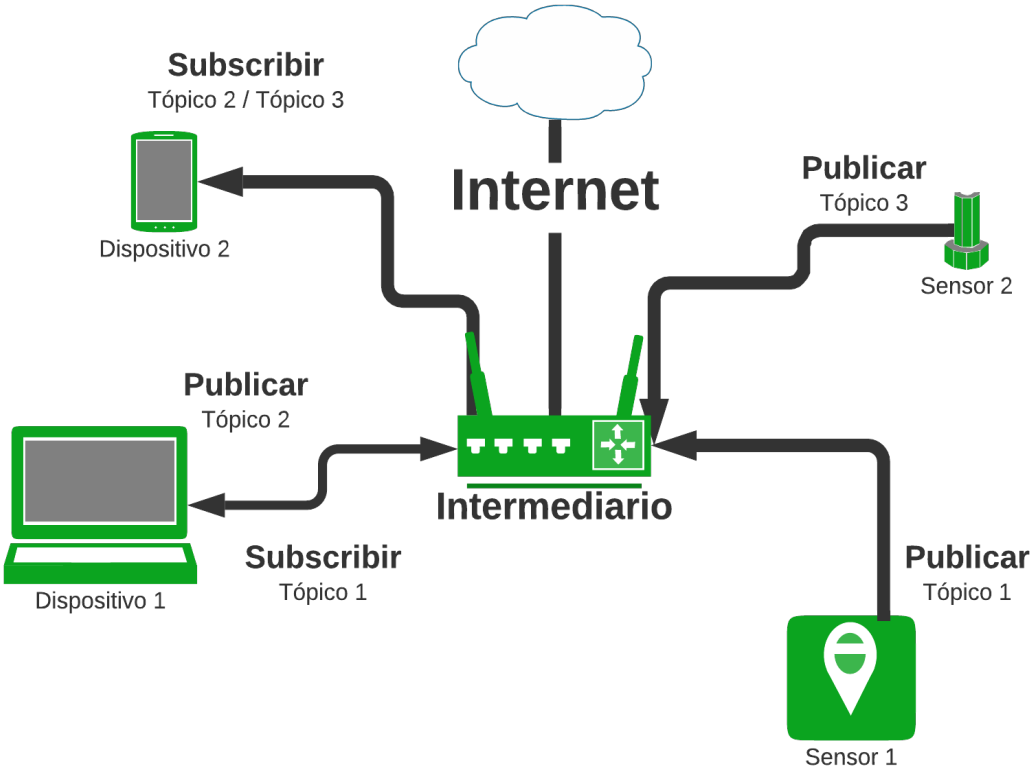


Figura 33. Red de comunicación IoT que utiliza MQTT.

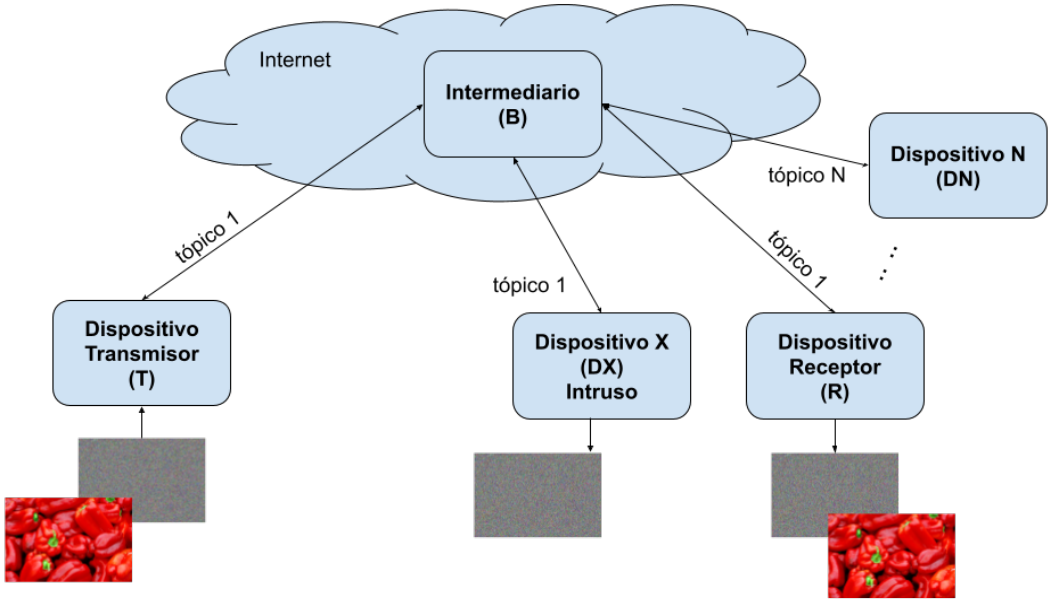


Figura 34. Red de comunicación IoT que utiliza MQTT con sistema criptográfico caótico.

En otras palabras, se puede decir que el sistema se divide en dos programas principales: el transmisor y receptor. Los cuales a su vez se dividen en cuatro procesos principales: 1.- selección o capturar imagen de cámara, 2.- selección y ajuste de mapa caóticos mejorado y parámetros, 3.- generación del criptograma y 4.- generar un tópico y enviar imagen. Esto se puede observar en la Figura 35.

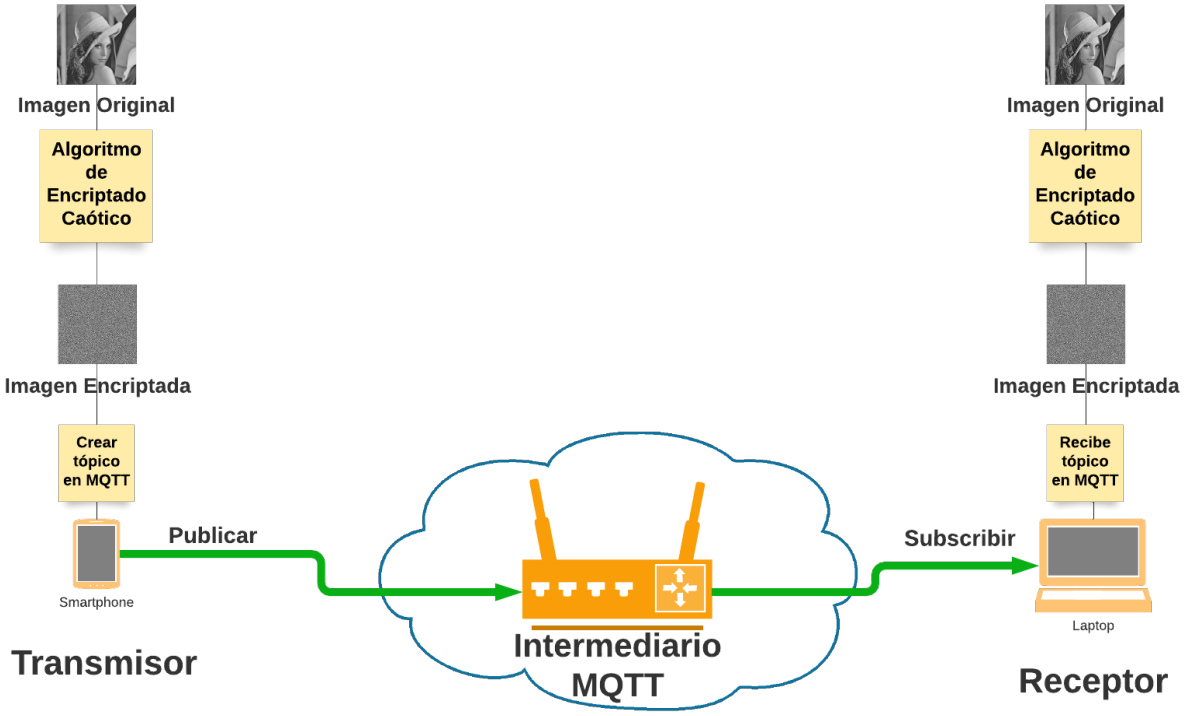


Figura 35. Diagrama a bloques del sistema criptográfico caótico en dispositivos IoT.

IV.5.1 Desarrollo de Transmisor

Para la etapa del transmisor se desarrolló un programa en el que las funciones principales son: seleccionar una imagen, seleccionar un algoritmo criptográfico, encriptar la imagen y publicar la imagen a través de internet usando MQTT. En la Figura 36, se muestra un diagrama de flujo del sistema del transmisor, donde podemos enviar la información encriptada o en su forma original sin encriptar.

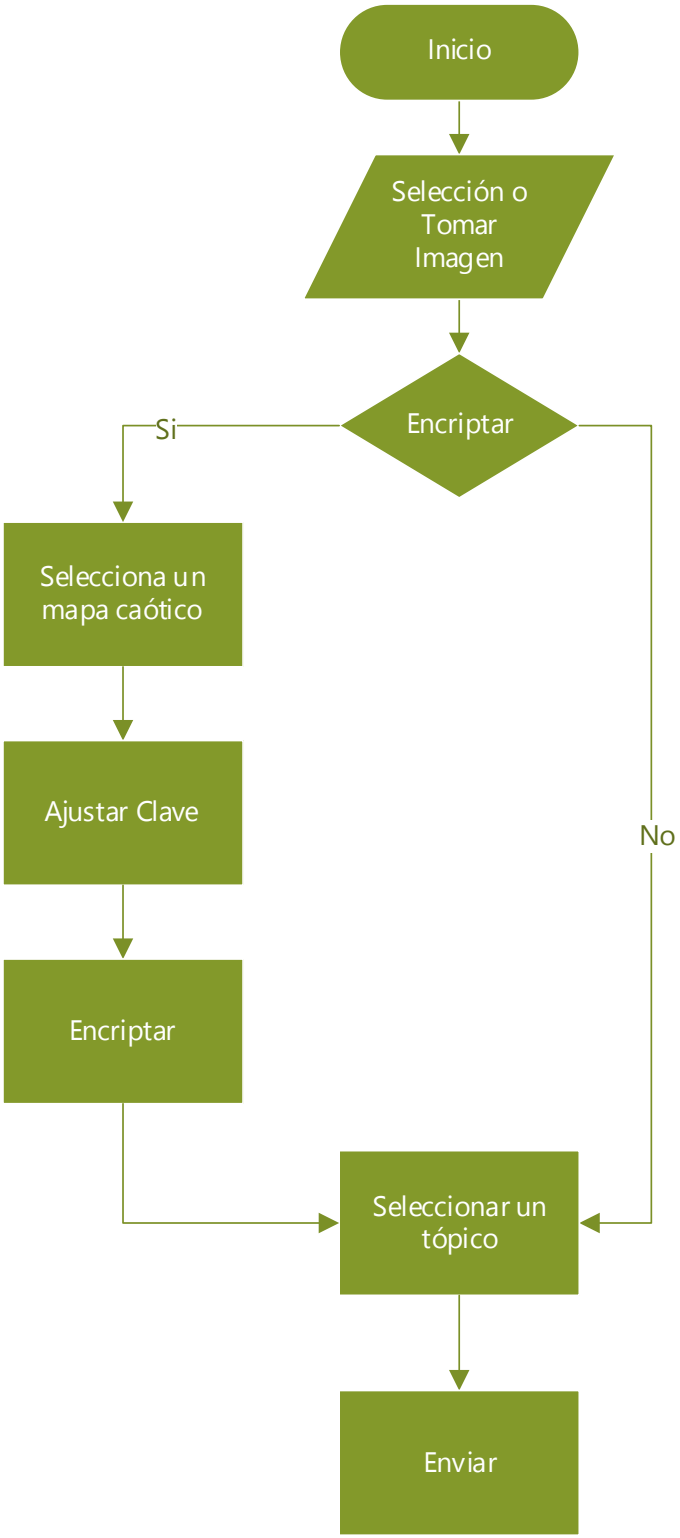


Figura 36. Diagrama de flujo de programa para publicar una imagen.

En la Figura 37, se muestra la ventana del programa principal instalado en el dispositivo que realiza la encriptación y publicación de la imagen usando MQTT a través de la internet.

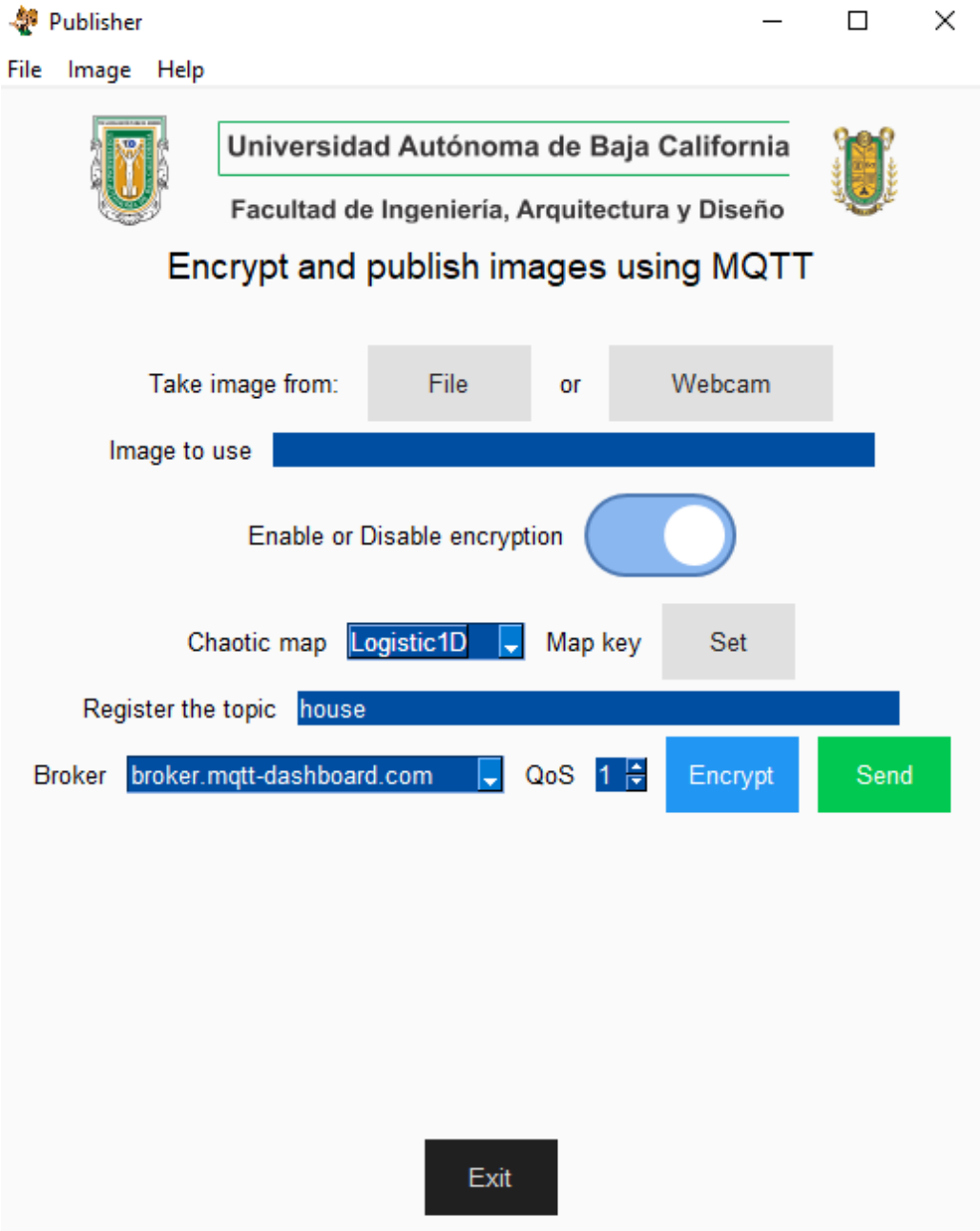


Figura 37. Ventana del programa principal que encripta y publica imágenes usando MQTT a través de internet.

Como primera opción, el usuario selecciona una imagen en su dispositivo, los archivos permitidos se limitaron a PNG o JPG o en su caso también esta la otra opción de tomar una imagen directamente de una cámara web que este conectada al dispositivo.

Posteriormente se puede seleccionar habilitar o deshabilitar el encriptado de imagen mediante el interruptor, como se muestra en la Figura 38, un mensaje le aparece al usuario cuando se desliza para confirmar si está o no activo el encriptado.

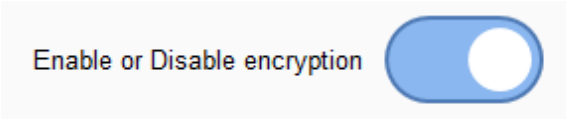


Figura 38. Botón para activar o desactivar el encriptado de imagen.

En caso de que el usuario seleccione activar el encriptado, este tendrá la opción de elegir uno de los cuatro mapas caóticos mejorados, Bernoulli, Logistic, Tent o Zigzag, como se puede observar en la Figura 39.

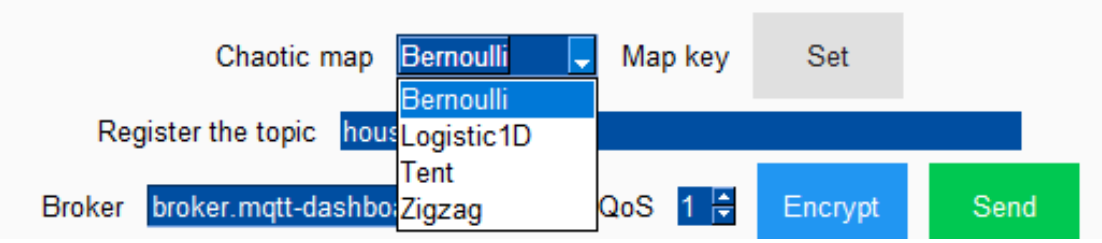


Figura 39. Selección de mapa caótico.

Posteriormente se deberá ingresar la clave para el mapa caótico seleccionado, como se observa en la Figura 40, estas claves no se almacenan y solo se utilizan para encriptar la imagen seleccionada, en caso de que el usuario desactive la casilla de encriptado o seleccione un nuevo mapa, una nueva clave será solicitada.

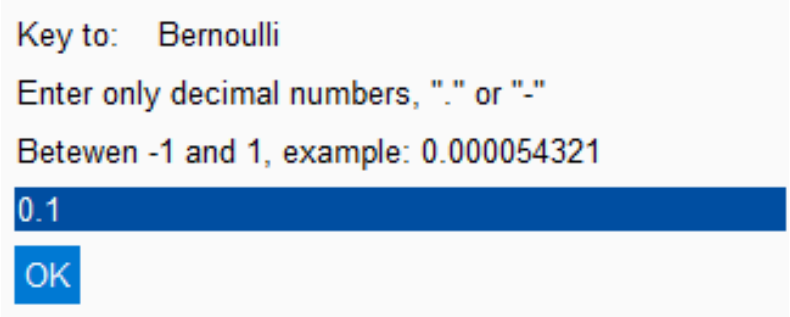


Figura 40. Registro de clave para el mapa seleccionado.

Una vez que se tiene el mapa caótico y la clave registrada, se ingresa el tópico o tema en el que se publicará la imagen. Este debe seguir las reglas generales para un tema valido, donde prácticamente es aceptado cualquier carácter, excepto $\$SYS$ por lo que se agrega como medida de seguridad no escribir el símbolo $ para no caer en algún error de comunicación.

Por último, se selecciona el intermediario (broker) a utilizar y los parámetros de calidad de servicio (QoS), como se observa en la Figura 41, con el fin de que el mensaje sea transmitido a través de internet se selecciona un intermediario en internet como el broker.mqtt-dashboard.com o test.mosquitto.org.

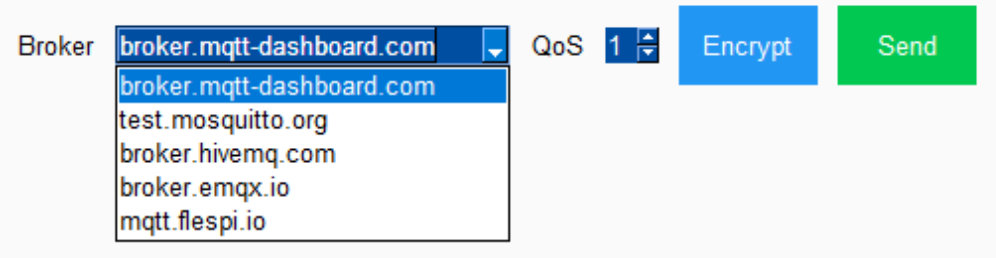


Figura 41. Selección de broker y QoS.

Para encriptar la imagen solo es necesario utilizar el botón de encriptar el cual utiliza la imagen seleccionada, el mapa caótico seleccionado y la clave dada, en caso de que falte alguna información, se informará al usuario.

El encriptado caótico sigue los siguientes pasos:

- *Cargar imagen…*
- *Extraer medidas de la imagen…*
- *Aplanar imagen usando la función imagen.ravel()…*
- *Generar números caóticos según el mapa seleccionado, uno por cada pixel de la imagen…*
- *Ajustar valores…*
- *Realizar combinación XOR de la imagen aplanada con los números caóticos ajustados…*
- *Redimensionar la matriz…*
- *Guarda imagen encriptada…*

Una vez la imagen encriptada y guardada se informa al usuario que ya se realizó el proceso para que pueda enviarse.

El botón de enviar, realiza el envío de la imagen encriptada o sin encriptar según el interruptor. Para cualquier caso, primero se realiza la conexión con el intermediario seleccionado, posteriormente se subscribe al tema previamente escrito y por último se envía la imagen. Por defecto, la comunicación tiene un QoS de 1, lo que significa que el intermediario se asegura de que todos los que se subscribieron al tema les llegue el mensaje, sin importar que llegue repetido el mensaje. Un QoS de 0, significa que el intermediario no se asegura que los subscriptores reciban el mensaje, solo se envía una sola vez sin tener certeza de la correcta recepción. Un QoS de 2, significa que el intermediario se asegura que el mensaje llegue solo una sola vez y no llega repetido.

IV.5.2 Desarrollo del Receptor

Para el dispositivo receptor, el cual se debe subscribir a un tema y recibir la imagen,

se desarrolló un programa con las siguientes características: Selecciona ruta de almacenamiento, habilitar el desencriptado, seleccionar el mapa caótico, ajustar clave, inscribirse a un tema, seleccionar y conectarse a un intermediario, desencriptar la imagen y guardarla. Este diagrama se puede observar en la Figura 42.

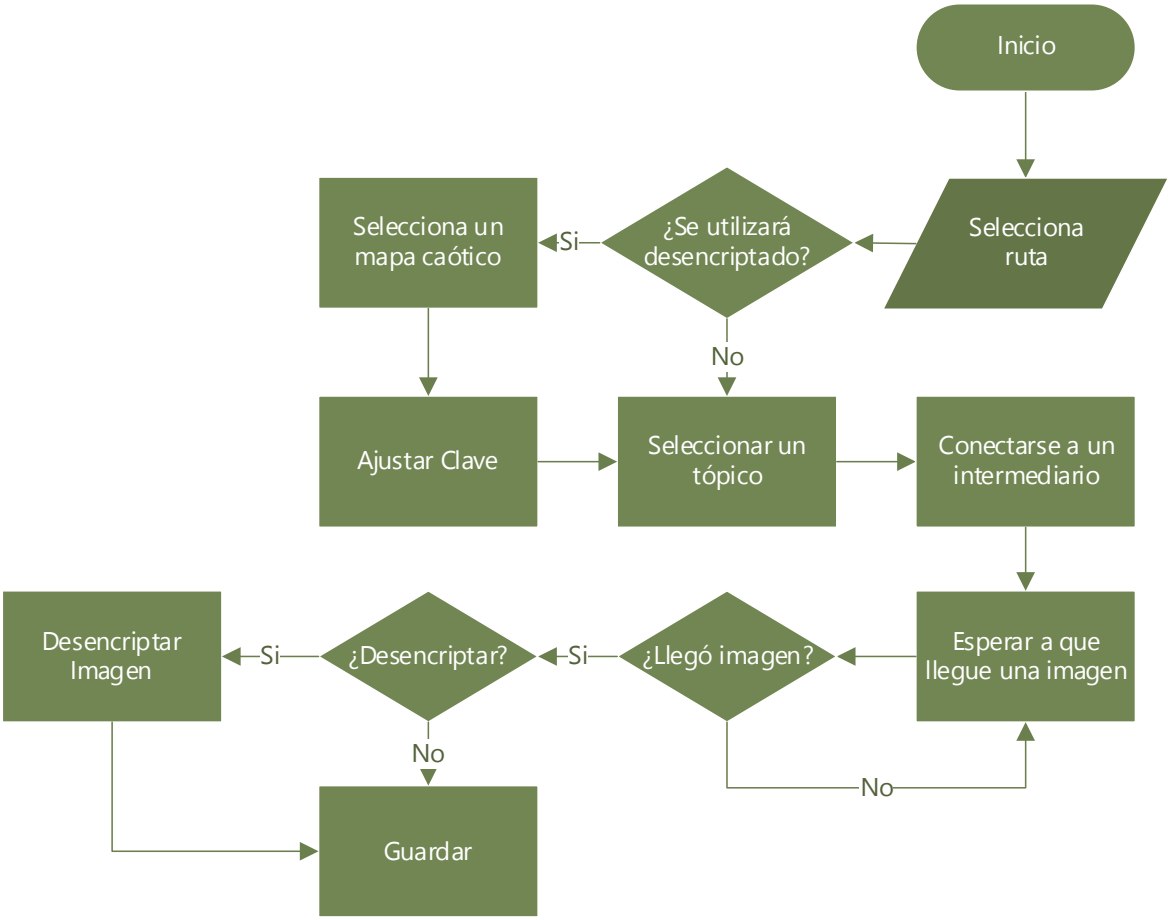


Figura 42. Diagrama de flujo de programa para subscribirse a un tópico y recibir imagen.

En la Figura 43 se muestra la ventana principal del programa que se ejecuta en el dispositivo que se subscribe y desencripta la imagen usando el protocolo MQTT sobre internet.

Se puede observar una similitud con el programa que publica y encripta la información, sin embargo, la principal diferencia es la opción de la QoS, ya que el que publica tiene un nivel de QoS mientras que este se subscribe no cuenta con esa característica, esto debido a que su función principal es de recibir información y no tiene la obligación de enviar información.

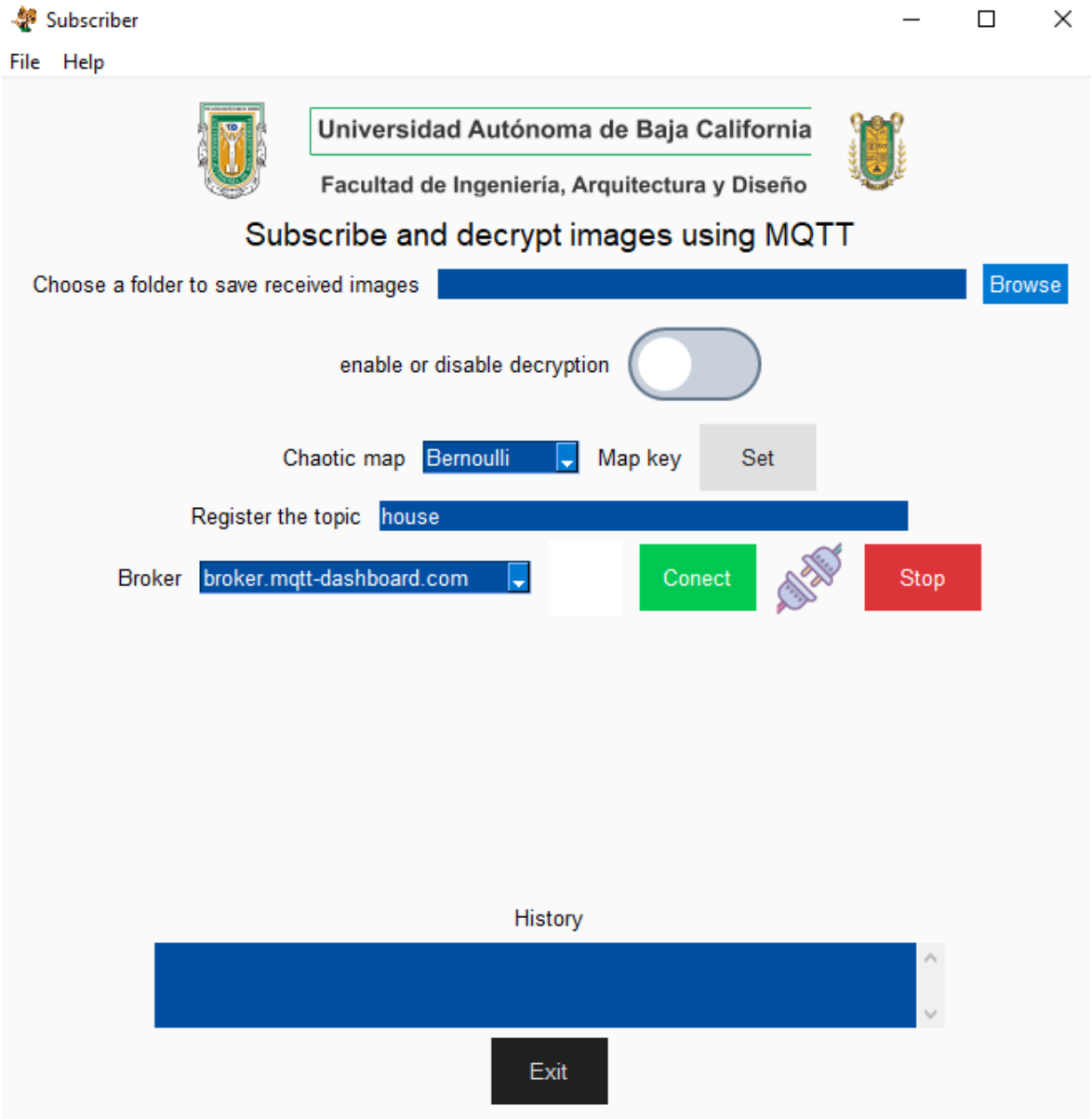


Figura 43. Ventana del programa principal que se subscribe a un tópico y desencripta imágenes usando MQTT a través de internet.

Como primera opción el usuario debe escoger una ruta donde guardara la imagen que desencripte. Como segundo paso, se observa que se tiene el botón para habilitar o deshabilitar el desencriptado. Posteriormente se cuenta con una selección del mapa caótico mejorado y su parámetro de clave. De la misma forma se cuenta con el registro del tópico o tema al cual se va a subscribir el dispositivo. Y por último se cuenta con la lista de intermediarios. Una vez que se cuenta con toda la información necesaria se puede proceder a conectarse con el botón correspondiente, lo que entablará una comunicación con el broker, avisando que se suscribió a un tópico, en espera de algún mensaje con la información que sea con el mismo tópico al que se suscribió. Una vez que el mensaje llega, el proceso del programa es:

- *Llega imagen…*
- *Revisar si se quiere desencriptar…*
- *En caso que si se quiere desencriptar…*
- *Extraer medidas de la imagen…*
- *Aplanar imagen…*
- *Generar secuencia caótica según el mapa y clave seleccionada…*
- *Ajustar valores…*
- *Realizar combinación XOR…*
- *Redimensionar…*
- *Guardar Imagen…*
- *En caso que no se quiere desencriptar…*
- *Guardar Imagen…*

Una vez la imagen desencriptada y guardada se informa al usuario por medio del historial.

Capítulo V. Conclusiones

Se utilizaron cuatro mapas caóticos 1D como caso de estudio para proponer un algoritmo simple, eficiente y seguro para el encriptado en tiempo real de imágenes RGB en un esquema de comunicación inalámbrica para aplicaciones de IoT. El criptosistema embebido propuesto se implementó en una Raspberry Pi 4 y una computadora personal con fines comparativos. Los resultados experimentales se verificaron utilizando enlaces M2M a través del protocolo MQTT en Internet. Se verificó que los exponentes de Lyapunov aumentaron cuando se aplicó la función $mod1023$ a las secuencias caóticas mejoradas, siendo el mapa logístico 1D con LE máximos de 5.2247, le sigue el mapa Tent con LE máxima de 5.2145 y el mapa Bernoulli con una LE máxima de 4.9151. El criptoanálisis confirmó que el algoritmo de cifrado propuesto que utiliza la función $mod1023$ ofrece robustez y alta seguridad contra varios ataques. Así, el esquema propuesto es fuerte frente a pruebas estadísticas como el análisis de histogramas, las pruebas NIST SP 800-22, el TestU01, el espacio-clave, el análisis de correlación, la entropía y los ataques diferenciales, como NPCR y UACI. Cabe destacar que los cuatro mapas caóticos utilizados en este trabajo satisfacen el requisito de espacio-clave mínimo de más de 2^{100}. De acuerdo a los resultados del análisis de seguridad, se observó que no es necesaria una dimensión mayor en el mapa caótico para garantizar la seguridad en el algoritmo de cifrado. Además, según los resultados de rendimiento, cuando el esquema propuesto se implementa en una computadora personal con un reloj de 2.9 GHz y utilizando las secuencias mejoradas con el mapa Logístico 1D alcanza un rendimiento de hasta 47.44 Mbit/s, superando a los trabajos relacionados previos. En cambio, si se implementa en la Raspberry Pi 4, alcanza los 10.53 Mbit/s. Sin embargo, estos resultados son competitivos con lo reportado en el estado del arte, que incluso utilizaba un ordenador personal. Finalmente, se verificó que estos

mapas caóticos 1D mejorados podrían ser utilizados en aplicaciones en tiempo real como esquemas M2M para IoT.

Finalmente, el esquema propuesto se codificó utilizando Python y se beneficia de ser de licencia libre, de código abierto y multiplataforma. Como resultado, se puede implementar en una variedad de sistemas operativos y plataformas de hardware, incluyendo GNU Linux, OS X, Windows 10, Android e iOS.

V.1 Trabajos a futuro

Los sistemas de comunicación día a día se están actualizando y cada vez más son los dispositivos que se conectan a internet, por lo que siempre es necesario continuar proponiendo nuevas, mejores y seguras formas de comunicación empleando las fortalezas de los algoritmos ya implementados. Aún quedan algunos problemas abiertos de investigación con gran oportunidad de aplicación, que a continuación se enlistan:

- Determinar una configuración en software óptima para la ejecución del sistema criptográfico, esto debido a que se desarrolló en un sistema de código abierto y multiplataforma.

- Desarrollar un nuevo algoritmo de encriptado empleando técnicas de cómputo paralelo con otros dispositivos embebidos de alto rendimiento y múltiples núcleos.

- Por la viabilidad del sistema criptográfico con relación a las imágenes de alta resolución, se propone implementar el algoritmo utilizando otro tipo de información audio/video en tiempo real.

- Implementar el algoritmo propuesto en aplicaciones e-salud.

- Combinar algoritmos caóticos con redes neuronales y otros algoritmos de aprendizaje máquina.

- Realizar evaluaciones de consumo energético con diferentes algoritmos caóticos.

- Desarrollar un sistema que involucre la generación dinámica de claves.

- Implementar el algoritmo propuesto en otro tipo de sistemas embebidos, tal como FPGA (Field Programmable Gate Array).

Bibliografía

[1] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, “Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT Scenarios,” *IEEE Access*, vol. 8, pp. 23022–23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[2] A. H. Mohd Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, and Y.-J. Park, “A Survey on Trend and Classification of Internet of Things Reviews,” *IEEE Access*, vol. 8, pp. 111763–111782, 2020, doi: 10.1109/ACCESS.2020.3002932.

[3] Y. Harbi, Z. Aliouat, S. Harous, A. Bentaleb, and A. Refoufi, “A Review of Security in Internet of Things,” *Wirel. Pers. Commun.*, vol. 108, no. 1, pp. 325–344, 2019, doi: 10.1007/s11277-019-06405-y.

[4] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, “Security of Internet of Things using RC4 and ECC Algorithms (Case Study: Smart Irrigation Systems),” *Wirel. Pers. Commun.*, vol. 116, no. 3, pp. 1713–1742, 2021, doi: 10.1007/s11277-020-07758-5.

[5] Y. P. K. Nkandeu and A. Tiedeu, “An image encryption algorithm based on substitution technique and chaos mixing,” *Multimed. Tools Appl.*, vol. 78, no. 8, pp. 10013–10034, 2019, doi: 10.1007/s11042-018-6612-2.

[6] X. Liu, S. Chen, L. Song, M. Woźniak, and S. Liu, “Self-attention negative feedback network for real-time image super-resolution,” *J. King Saud Univ. - Comput. Inf. Sci.*, 2021, doi: 10.1016/j.jksuci.2021.07.014.

[7] X. Wang, N. Guan, and J. Yang, “Image encryption algorithm with random scrambling based on one-dimensional logistic self-embedding chaotic map,” *Chaos, Solitons and Fractals*, vol. 150, 2021, doi:

10.1016/j.chaos.2021.111117.

[8] L.-L. Guo and M. Woźniak, “An Image Super-Resolution Reconstruction Method with Single Frame Character Based on Wavelet Neural Network in Internet of Things,” *Mob. Networks Appl.*, vol. 26, no. 1, pp. 390–403, 2021, doi: 10.1007/s11036-020-01681-6.

[9] T. Kavitha, O. Rajitha, K. Thejaswi, and N. B. Muppalaneni, *Classification of Encryption Algorithms Based on Ciphertext Using Pattern Recognition Techniques*, vol. 31. 2020.

[10] K. N. Qureshi, S. Qayyum, M. N. Ul Islam, and G. Jeon, “A secure data parallel processing based embedded system for internet of things computer vision using field programmable gate array devices,” *Int. J. Circuit Theory Appl.*, vol. 49, no. 5, pp. 1450–1469, 2021, doi: 10.1002/cta.2964.

[11] Z. Chang and M. Woźniak, “Encryption technology of voice transmission in mobile network based on 3DES-ECC algorithm,” *Mob. Networks Appl.*, vol. 25, no. 6, pp. 2398–2408, 2020, doi: 10.1007/s11036-020-01617-0.

[12] M. A. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, R. M. López-Gutiérrez, and O. R. Acosta Del Campo, “A RGB image encryption algorithm based on total plain image characteristics and chaos,” *Signal Processing*, vol. 109, pp. 119–131, 2015, doi: 10.1016/j.sigpro.2014.10.033.

[13] M. A. Murillo-Escobar, M. O. Meranza-Castillón, R. M. López-Gutiérrez, and C. Cruz-Hernández, “Suggested integral analysis for chaos-based image cryptosystems,” *Entropy*, vol. 21, no. 8, 2019, doi: 10.3390/E21080815.

[14] M. García-Martínez and E. Campos-Cantón, “Pseudo-random bit generator based on multi-modal maps,” *Nonlinear Dyn.*, vol. 82, no. 4, pp. 2119–2131, 2015, doi: 10.1007/s11071-015-2303-y.

[15] H. S. Kwok and W. K. S. Tang, “A fast image encryption system based on chaotic maps with finite precision representation,” *Chaos, Solitons and*

Fractals, vol. 32, no. 4, pp. 1518–1529, 2007, doi: 10.1016/j.chaos.2005.11.090.

[16] L. G. Nardo, E. G. Nepomuceno, J. Arias-Garcia, and D. N. Butusov, “Image encryption using finite-precision error,” *Chaos, Solitons and Fractals*, vol. 123, pp. 69–78, 2019, doi: 10.1016/j.chaos.2019.03.026.

[17] Q. Lai, B. Norouzi, and F. Liu, “Dynamic analysis, circuit realization, control design and image encryption application of an extended Lü system with coexisting attractors,” *Chaos, Solitons and Fractals*, vol. 114, pp. 230–245, 2018, doi: 10.1016/j.chaos.2018.07.011.

[18] M. L. Sahari and I. Boukemara, “A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption,” *Nonlinear Dyn.*, vol. 94, no. 1, pp. 723–744, 2018, doi: 10.1007/s11071-018-4390-z.

[19] Ü. Çavuşoğlu, S. Kaçar, I. Pehlivan, and A. Zengin, “Secure image encryption algorithm design using a novel chaos based S-Box,” *Chaos, Solitons and Fractals*, vol. 95, pp. 92–101, 2017, doi: 10.1016/j.chaos.2016.12.018.

[20] I. El Hanouti, H. El Fadili, and K. Zenkouar, “Cryptanalysis of an embedded systems’ image encryption,” *Multimed. Tools Appl.*, vol. 80, no. 9, pp. 13801–13820, 2021, doi: 10.1007/s11042-020-10289-7.

[21] M. Gad, E. Hagras, H. Soliman, and N. Hikal, “A new parallel fuzzy multi modular chaotic logistic map for image encryption,” *Int. Arab J. Inf. Technol.*, vol. 18, no. 2, pp. 227–236, 2021, doi: 10.34028/IAJIT/18/2/12.

[22] W. K. Lee, R. C. W. Phan, W. S. Yap, and B. M. Goi, “SPRING: a novel parallel chaos-based image encryption scheme,” *Nonlinear Dyn.*, vol. 92, no. 2, pp. 575–593, 2018, doi: 10.1007/s11071-018-4076-6.

[23] M. A. Murillo-Escobar, C. Cruz-Hernández, L. Cardoza-Avendaño, and R. Méndez-Ramírez, “A novel pseudorandom number generator based on

pseudorandomly enhanced logistic map," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 407–425, 2017, doi: 10.1007/s11071-016-3051-3.

[24] X. Wang and X. Chen, "An image encryption algorithm based on dynamic row scrambling and Zigzag transformation," *Chaos, Solitons and Fractals*, vol. 147, 2021, doi: 10.1016/j.chaos.2021.110962.

[25] S. Zhou, X. Wang, M. Wang, and Y. Zhang, "Simple colour image cryptosystem with very high level of security," *Chaos, Solitons and Fractals*, vol. 141, 2020, doi: 10.1016/j.chaos.2020.110225.

[26] "No Title."

[27] L. O. Chua, C. W. Wu, A. Huang, and G.-Q. Zhong, "A Universal Circuit for Studying and Generating Chaos—Part I: Routes to Chaos," *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 40, no. 10, pp. 732–744, 1993, doi: 10.1109/81.246149.

[28] F. Yu, H. Shen, Z. Zhang, Y. Huang, S. Cai, and S. Du, "A new multi-scroll Chua's circuit with composite hyperbolic tangent-cubic nonlinearity: Complex dynamics, Hardware implementation and Image encryption application," *Integration*, vol. 81, pp. 71–83, 2021, doi: 10.1016/j.vlsi.2021.05.011.

[29] J. Kengne and R. L. T. Mogue, "Dynamic analysis of a novel jerk system with composite tanh-cubic nonlinearity: chaos, multi-scroll, and multiple coexisting attractors," *Int. J. Dyn. Control*, vol. 7, no. 1, pp. 112–133, 2019, doi: 10.1007/s40435-018-0444-9.

[30] J. Kengne, S. M. Njikam, and V. R. F. Signing, "A plethora of coexisting strange attractors in a simple jerk system with hyperbolic tangent nonlinearity," *Chaos, Solitons and Fractals*, vol. 106, pp. 201–213, 2018, doi: 10.1016/j.chaos.2017.11.027.

[31] J. M. Muñoz-Pacheco, E. Tlelo-Cuautle, I. Toxqui-Toxqui, C. Sánchez-López, and R. Trejo-Guerra, "Frequency limitations in generating multi-scroll chaotic

attractors using CFOAs," *Int. J. Electron.*, vol. 101, no. 11, pp. 1559–1569, 2014, doi: 10.1080/00207217.2014.880999.

[32] J. Ma, X. Wu, R. Chu, and L. Zhang, "Selection of multi-scroll attractors in Jerk circuits and their verification using Pspice," *Nonlinear Dyn.*, vol. 76, no. 4, pp. 1951–1962, 2014, doi: 10.1007/s11071-014-1260-1.

[33] C. H. Wang, H. Xu, and F. Yu, "A novel approach for constructing high-order Chua's circuit with multi-directional multi-scroll chaotic attractors," *Int. J. Bifurc. Chaos*, vol. 23, no. 2, 2013, doi: 10.1142/S0218127413500223.

[34] J. Jin and L. Cui, "Fully integrated memristor and its application on the scroll-controllable hyperchaotic system," *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/4106398.

[35] X. Zhang and C. Wang, "A Novel Multi-Attractor Period Multi-Scroll Chaotic Integrated Circuit Based on CMOS Wide Adjustable CCCII," *IEEE Access*, vol. 7, pp. 16336–16350, 2019, doi: 10.1109/ACCESS.2019.2894853.

[36] J. Jin and L. Zhao, "Low voltage low power fully integrated chaos generator," *J. Circuits, Syst. Comput.*, vol. 27, no. 10, 2018, doi: 10.1142/S0218126618501554.

[37] J. Jin, "Programmable multi-direction fully integrated chaotic oscillator," *Microelectronics J.*, vol. 75, pp. 27–34, 2018, doi: 10.1016/j.mejo.2018.02.007.

[38] P. Kietzmann, T. C. Schmidt, and M. Wählisch, "A Guideline on Pseudorandom Number Generation (PRNG) in the IoT," *ACM Comput. Surv.*, vol. 54, no. 6, 2021, doi: 10.1145/3453159.

[39] S. Cang, Z. Kang, and Z. Wang, "Pseudo-random number generator based on a generalized conservative Sprott-A system," *Nonlinear Dyn.*, 2021, doi: 10.1007/s11071-021-06310-9.

[40] F. Yu *et al.*, "Pseudorandom number generator based on a 5D hyperchaotic

four-wing memristive system and its FPGA implementation: PRNG based on a 5D hyperchaotic four-wing memristive system and its FPGA implementation," *Eur. Phys. J. Spec. Top.*, vol. 230, no. 7–8, pp. 1763–1772, 2021, doi: 10.1140/epjs/s11734-021-00132-x.

[41] A. V. Tutueva, E. G. Nepomuceno, A. I. Karimov, V. S. Andreev, and D. N. Butusov, "Adaptive chaotic maps and their application to pseudo-random numbers generation," *Chaos, Solitons and Fractals*, vol. 133, 2020, doi: 10.1016/j.chaos.2020.109615.

[42] L. Palacios-Luengas, J. L. Pichardo-Méndez, J. A. Díaz-Méndez, F. Rodríguez-Santos, and R. Vázquez-Medina, "PRNG Based on Skew Tent Map," *Arab. J. Sci. Eng.*, vol. 44, no. 4, pp. 3817–3830, 2019, doi: 10.1007/s13369-018-3688-y.

[43] D. Lambić and M. Nikolić, "Pseudo-random number generator based on discrete-space chaotic map," *Nonlinear Dyn.*, vol. 90, no. 1, pp. 223–232, 2017, doi: 10.1007/s11071-017-3656-1.

[44] B. Stoyanov and K. Kordov, "Novel secure pseudo-random number generation scheme based on two tinkerbell maps," *Adv. Stud. Theor. Phys.*, vol. 9, no. 9, pp. 411–421, 2015, doi: 10.12988/astp.2015.5342.

[45] Y. Liu and X.-J. Tong, "A new pseudorandom number generator based on a complex number chaotic equation," *Chinese Phys. B*, vol. 21, no. 9, 2012, doi: 10.1088/1674-1056/21/9/090506.

[46] D. A. Trujillo-Toledo *et al.*, "Real-time RGB image encryption for IoT applications using enhanced sequences from chaotic maps," *Chaos, Solitons and Fractals*, vol. 153, 2021, doi: 10.1016/j.chaos.2021.111506.

[47] A. Flores-Vergara *et al.*, "Implementing a chaotic cryptosystem by performing parallel computing on embedded systems with multiprocessors," *Entropy*, vol. 21, no. 3, 2019, doi: 10.3390/e21030268.

[48] E. Rodríguez-Orozco *et al.*, “FPGA-based chaotic cryptosystem by using voice recognition as access key,” *Electron.*, vol. 7, no. 12, 2018, doi: 10.3390/electronics7120414.

[49] L. G. de la Fraga, E. Torres-Pérez, E. Tlelo-Cuautle, and C. Mancillas-López, “Hardware implementation of pseudo-random number generators based on chaotic maps,” *Nonlinear Dyn.*, vol. 90, no. 3, pp. 1661–1670, 2017, doi: 10.1007/s11071-017-3755-z.

[50] I. Sahmi, A. Abdellaoui, T. Mazri, and N. Hmina, “MQTT-PRESENT: Approach to secure internet of things applications using MQTT protocol,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 5, pp. 4577–4586, 2021, doi: 10.11591/ijece.v11i5.pp4577-4586.

[51] J.-H. Park, H.-S. Kim, and W.-T. Kim, “DM-MQTT: An efficient MQTT based on SDN multicast for massive IoT communications,” *Sensors (Switzerland)*, vol. 18, no. 9, 2018, doi: 10.3390/s18093071.

[52] S. Jenisch and A. Uhl, “A detailed evaluation of format-compliant encryption methods for JPEG XR-compressed images,” *Eurasip J. Inf. Secur.*, 2014, doi: 10.1186/1687-417X-2014-6.

[53] T.-L. Liao, H.-R. Lin, P.-Y. Wan, and J.-J. Yan, “Improved attribute-based encryption using chaos synchronization and its application to MQTT security,” *Appl. Sci.*, vol. 9, no. 20, 2019, doi: 10.3390/app9204454.

[54] B. M. Alshammari, R. Guesmi, T. Guesmi, H. Alsaif, and A. Alzamil, “Implementing a symmetric lightweight cryptosystem in highly constrained iot devices by using a chaotic s-box,” *Symmetry (Basel).*, vol. 13, no. 1, pp. 1–20, 2021, doi: 10.3390/sym13010129.

[55] A. M. González-Zapata, E. Tlelo-Cuautle, I. Cruz-Vega, and W. D. León-Salas, “Synchronization of chaotic artificial neurons and its application to secure image transmission under MQTT for IoT protocol,” *Nonlinear Dyn.*, 2021, doi:

10.1007/s11071-021-06532-x.

[56] C. Nykvist, M. Larsson, A. H. Sodhro, and A. Gurtov, “A lightweight portable intrusion detection communication system for auditing applications,” *Int. J. Commun. Syst.*, vol. 33, no. 7, 2020, doi: 10.1002/dac.4327.

[57] K. O. Alessio *et al.*, “Open source, low-cost device for thermometric titration with non-contact temperature measurement,” *Talanta*, vol. 216, 2020, doi: 10.1016/j.talanta.2020.120975.

[58] O. A. Aguirre-Castro *et al.*, “Design and construction of an rov for underwater exploration,” *Sensors (Switzerland)*, vol. 19, no. 24, 2019, doi: 10.3390/s19245387.

[59] F. Zamora-Arellano *et al.*, “Development of a portable, reliable and low-cost electrical impedance tomography system using an embedded system,” *Electron.*, vol. 10, no. 1, pp. 1–24, 2021, doi: 10.3390/electronics10010015.

[60] B. Mondal, P. K. Behera, and S. Gangopadhyay, “A secure image encryption scheme based on a novel 2D sine–cosine cross-chaotic (SC3) map,” *J. Real-Time Image Process.*, vol. 18, no. 1, 2021, doi: 10.1007/s11554-019-00940-4.

[61] E. E. García-Guerrero, E. Inzunza-González, O. R. López-Bonilla, J. R. Cárdenas-Valdez, and E. Tlelo-Cuautle, “Randomness improvement of chaotic maps for image encryption in a wireless communication scheme using PIC-microcontroller via Zigbee channels,” *Chaos, Solitons and Fractals*, vol. 133, 2020, doi: 10.1016/j.chaos.2020.109646.

[62] B. Norouzi and S. Mirzakuchaki, “A fast color image encryption algorithm based on hyper-chaotic systems,” *Nonlinear Dyn.*, vol. 78, no. 2, pp. 995–1015, 2014, doi: 10.1007/s11071-014-1492-0.

[63] A. Alhudhaif, M. Ahmad, A. Alkhayyat, N. Tsafack, A. K. Farhan, and R. Ahmed, “Block Cipher Nonlinear Confusion Components Based on New 5-D Hyperchaotic System,” *IEEE Access*, vol. 9, pp. 87686–87696, 2021, doi:

10.1109/ACCESS.2021.3090163.

[64] H.-M. Yuan, Y. Liu, T. Lin, T. Hu, and L.-H. Gong, “A new parallel image cryptosystem based on 5D hyper-chaotic system,” *Signal Process. Image Commun.*, vol. 52, pp. 87–96, 2017, doi: 10.1016/j.image.2017.01.002.

[65] R. Hegger, H. Kantz, and T. Schreiber, “Practical implementation of nonlinear time series methods: The TISEAN package,” *Chaos*, vol. 9, no. 2, pp. 413–435, 1999, doi: 10.1063/1.166424.

[66] Y. Wang, Z. Liu, J. Ma, and H. He, “A pseudorandom number generator based on piecewise logistic map,” *Nonlinear Dyn.*, vol. 83, no. 4, pp. 2373–2391, 2016, doi: 10.1007/s11071-015-2488-0.

[67] S. Sadoudi, C. Tanougast, M. S. Azzaz, and A. Dandache, “Design and FPGA implementation of a wireless hyperchaotic communication system for secure real-time image transmission,” *Eurasip J. Image Video Process.*, vol. 2013, pp. 1–18, 2013, doi: 10.1186/1687-5281-2013-43.

[68] J. Li, J. Zheng, and P. Whitlock, “Efficient deterministic and non-deterministic pseudorandom number generation,” *Math. Comput. Simul.*, vol. 143, pp. 114–124, 2018, doi: 10.1016/j.matcom.2016.07.011.

[69] P. L’ecuyer and R. Simard, “TestU01: A C library for empirical testing of random number generators,” *ACM Trans. Math. Softw.*, vol. 33, no. 4, 2007, doi: 10.1145/1268776.1268777.

[70] A. Kuznetsova *et al.*, “The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale,” *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1956–1981, 2020, doi: 10.1007/s11263-020-01316-z.

[71] S. F. Yousif, A. J. Abboud, and H. Y. Radhi, “Robust Image Encryption with Scanning Technology, the El-Gamal Algorithm and Chaos Theory,” *IEEE Access*, vol. 8, pp. 155184–155209, 2020, doi:

10.1109/ACCESS.2020.3019216.

[72] J. Khan *et al.*, “SMSH: Secure surveillance mechanism on smart healthcare IoT system with probabilistic image encryption,” *IEEE Access*, vol. 8, pp. 15747–15767, 2020, doi: 10.1109/ACCESS.2020.2966656.

[73] S. Roy, U. Rawat, H. A. Sareen, and S. K. Nayak, “IECA: an efficient IoT friendly image encryption technique using programmable cellular automata,” *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 11, pp. 5083–5102, 2020, doi: 10.1007/s12652-020-01813-6.

[74] A. H. Zahid, E. Al-Solami, and M. Ahmad, “A Novel Modular Approach Based Substitution-Box Design for Image Encryption,” *IEEE Access*, vol. 8, pp. 150326–150340, 2020, doi: 10.1109/ACCESS.2020.3016401.

[75] “(No Title).” https://downloads.hindawi.com/journals/mpe/2016/7683687.pdf (accessed May 18, 2021).

Apéndice A

Publicaciones derivadas del trabajo de tesis doctoral

Trujillo-Toledo, D.A.; López-Bonilla, O.R.; García-Guerrero, E.E.; Tlelo-Cuautle, E.; López-Mancilla, D.; Guillén-Fernández, O.; Inzunza-González, E. ***Real-time RGB image encryption for IoT applications using enhanced sequences from chaotic maps***. Chaos Solitons Fractals **2021**, Vol. 153 Part 2, 111506.
DOI: http://doi.org/10.1016/j.chaos.2021.111506

Acceso Universal del Conocimiento

Trujillo-Toledo, D.A., Inzunza-González, E., López-Bonilla, O.R.; García-Guerrero, E.E. Expociencia y Tecnología 2021, ***Internet de las Cosas (Internet of things, IoT)***. Enlace: https://youtu.be/5oml0mC4KjQ

Trujillo-Toledo, D.A., Inzunza-González, E., López-Bonilla, O.R.; García-Guerrero, E.E. Cardenas-Valdez J.R. Expociencia y Tecnología 2020, **Internet de las Cosas (Internet of things, IoT)**. Enlace: https://youtu.be/r11wg4EYpBE