Congreso Internacional en Ingeniería Electrónica. Mem. Electro 2015, Vol. 37, pp 285-291, Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2015
ISSN 1405-2172

IMPLEMENTACIÓN DE UN SISTEMA CRIPTOGRÁFICO CAÓTICO EN FPGA CON APLICACIÓN EN IMÁGENES DIGITALES

Rodríguez Orozco Eduardo, García Guerrero E. Efrén, Inzunza González Everardo y López Bonilla Oscar R.
Universidad Autónoma de Baja California, Facultad de Ingeniería, Arquitectura y Diseño.
Carretera Transpeninsular Ensenada-Tijuana 3917, Colonia Playitas. C.P. 22860. Ensenada, B.C. México.
Tel.+52(646)175-07-44, Fax. +52(646)174-43-33
eduardo.rodriguez.orozco@uabc.edu.mx, eegarcia@uabc.edu.mx, einzunza@uabc.edu.mx, olopez@uabc.edu.mx

RESUMEN
En este trabajo se presenta la implementación de un sistema criptográfico caótico en una FPGA Spartan 3E-1600 de Xilinx, con aplicación en imágenes digitales. El propósito del trabajo es presentar una alternativa viable, adaptativa, escalable, de estándares en seguridad y privacidad competitivos, e instrumentado con tecnología de vanguardia para aplicaciones de cifrado de información digital. El sistema criptográfico opera como un dispositivo periférico de una computadora personal (PC), lo que adiciona un nivel más de seguridad al proceso de encriptado. El algoritmo embebido en la FPGA tiene una lógica operacional simple, lo que permite ejecutar sin distinción el ciclo de encriptado-desencriptado de imágenes, sin cambio alguno en su estructura. De los resultados experimentales y del análisis de seguridad frente a diferentes tipos de ataques, se muestra que el sistema propuesto es robusto y seguro. Por lo que, el sistema se potencializa para integrarse en aplicaciones diversas tales como en sistemas de reconocimiento biométrico.
Palabras clave: Criptografía, cifrado caótico embebido, FPGA.

ABSTRACT
In this work the implementation of a chaotic cryptographic system is presented in a FPGA Spartan 3E-1600 Xilinx, with applications in digital images. The purpose of the work is to present a viable, adaptive, scalable alternative of standards in security and competitive privacy, and instrumented with the latest technology for encryption applications of digital information. The cryptographic system operates as a peripheral device of a personal computer (PC), which adds another level of security to the encryption process. The algorithm embedded in the FPGA has a simple operational logic, which allows us to perform without distinction the encryption-decryption cycle of images, without any change in its structure. From the experimental results and analysis of security against different types of attacks, it is shown that the proposed system is strong and secure. Therefore, the system is potentiated to be integrated in various applications such as biometric recognition systems.
Keywords: Cryptography, chaotic embedded encryption, FPGA.

1. INTRODUCCIÓN
Es inminente la presencia de la era digital en la que nos encontramos y más aún, del advenimiento de la era de la información cuántica y en particular de las computadoras cuánticas, que son capaces de realizar con mucha mayor rapidez factorizaciones sin precedentes (procedimiento en el que se basan los algoritmos de cifrado actuales), que supone obviamente el declive de los algoritmos de encriptado convencional. Varios sistemas actuales como los de comunicación o identificación biométrica, emplean programación para encriptar información con lo que logran adicionar la privacidad y seguridad requerida. Sin embargo, el desarrollo creciente de computadoras más eficientes, condicionan y limitan la viabilidad de esta técnica. Por lo que, a partir de esta perspectiva surge la necesidad imperiosa de desarrollar e implementar nuevas técnicas que nos permitan optimizar el ciclo de codificación-decodificación de la información, que sean lo suficientemente eficientes para incrementar y garantizar los niveles de seguridad y privacidad que ya son necesarios en nuestros días y que seguramente serán más demandantes en el futuro mediato. Por otra parte, después de muchos estudios teóricos e investigación experimental se reconoce que el caos es útil en muchas aplicaciones y, en los últimos tiempos se ha incrementado el interés por utilizarlo en diversas disciplinas. La razón principal por el interés en el caos, se debe a su complejidad, a su comportamiento dinámico muy parecido al ruido y a la sensibilidad a las condiciones iniciales. Aunado a esto, el avance tecnológico en el desarrollo de dispositivos electrónicos para el procesamiento de datos e información como lo son los microprocesadores y sistemas embebidos, nos abre camino a la posibilidad de desarrollar nuevos métodos y técnicas para resolver los problemas derivados de las necesidades de cifrado de información en forma segura. Actualmente, se ha desarrollado hardware programable como los FPGA (Field Programable Logic Array) que están tomando relevancia significativa en el paradigma de diseño en los sistemas digitales embebidos, esto debido al excelente equilibrio entre el poder de cómputo y flexibilidad de procesamiento que provee. El objetivo de este trabajo es hacer uso de la tecnología de vanguardia que ofrecen las FPGA's, e integrar un sistema de encriptado-desencriptado de imágenes digitales basado en caos con niveles de seguridad competitivos ante diferentes tipos de ataques, bajo el esquema de un algoritmo embebido operacionalmente simple.

2. METODOLOGÍA
En esta sección se presenta la metodología desarrollada para llevar a cabo los procesos de encriptado y desenciptado caótico de imágenes digitales, implementada en la FPGA Spartan 3E-1600 utilizada en este trabajo. Las etapas a considerar son: *i*) selección del mapeo caótico como base para la generación de caos, *ii*) generación del código VHDL correspondiente al sistema de ecuaciones en diferencias del sistema caótico, *iii*) implementación propia del sistema caótico en la FPGA, *iv*) desarrollo e implementación del algoritmo embebido que ejecuta el ciclo encriptado-desencriptado de imágenes digitales y *v*) análisis de los niveles seguridad obtenidos de las imágenes encriptadas

(criptogramas) generadas por el sistema criptográfico implementado.

2.1. MAPEO CAÓTICO

Para la generación de caos, se seleccionó a manera de ejemplo el mapeo hipercaótico de Rössler. O.E. Rössler derivó un conjunto de tres ecuaciones diferenciales que es más sencillo que el sistema caótico de Lorenz [1], al constituirse por un solo término no lineal. A partir de la aplicación de Poincaré [2], el sistema de ecuaciones que describe al sistema dinámico hipercaótico de Rössler en forma discreta [3], queda definido por (1).

$$\begin{aligned} X_1(n+1) &= \alpha X_1(n)\big(1-X_1(n)\big) - \beta\big(X_3(n)+\gamma\big)\big(1-2X_2(n)\big) \\ X_2(n+1) &= \delta X_2(n)\big(1-X_2(n)\big) + \zeta X_3(n) \\ X_3(n+1) &= \eta\Big(\big(X_3(n)+\gamma\big)\big(1-2X_2(n)\big)-1\Big)\big(1-\theta X_1(n)\big) \end{aligned} \quad (1)$$

Del sistema de ecuaciones (1), se observa que el mapeo depende de los parámetros $\alpha, \beta, \gamma, \delta, \zeta, \eta$ y θ. Del análisis del sistema a partir de sus diagramas de bifurcación [4], se eligen: $\alpha = 3.8, \beta = 0.05, \gamma = 0.35, \delta = 3.78, \zeta = 0.2, \eta = 0.1$ y $\theta = 0.1$, de tal manera que el mapeo exhibe una dinámica hipercaótica. Algunas de las características más atractivas que ofrece el sistema de Rössler son: *i*) es un sistema tridimensional con alta dependencia entre sus estados $X_1(n), X_2(n)$ y $X_3(n)$, *ii*) es hipercaótico, ya que posee dos exponentes de Lyapunov positivos y *iii*) sus 7 parámetros y las 3 condiciones iniciales lo potencializan con un amplio margen de espacios de claves, haciéndolo atractivo para aplicaciones criptográficas. De la amplia gama de mapeos caóticos ampliamente documentados en la literatura (Hénon, Chen, Arnold, Logístico 2D, Ikeda, Tinkerbell, etc), la implementación específica en este trabajo del mapeo Rössler, obedece en gran medida en darle continuidad a una serie de trabajos previos en los que hemos explorado el sistema a nivel de software [4,5]. Bajo la implementación del mapeo ahora en la tecnología que ofrecen las FPGA's, nos brinda la posibilidad de comparar y evaluar la viabilidad y potencialidad de las tecnologías emergentes, a fin de desarrollar aplicaciones reales basadas en caos tales como la que se presenta en este trabajo referente al encriptado-desencriptado de imágenes digitales.

2.2. GENERACIÓN DE CÓDIGO VHDL

La generación del código VHDL del sistema dinámico representado a partir de sus ecuaciones en diferencias (1), en función de la FPGA Spartan 3E-1600, se obtiene a partir de modelar el sistema con la plataforma SysGen (System Generator) de la familia Xilinx [6] en Simulink. En la Figura 1, se muestra una sección de la representación gráfica del estado $X_1(n)$, del mapeo dado por el sistema (1).

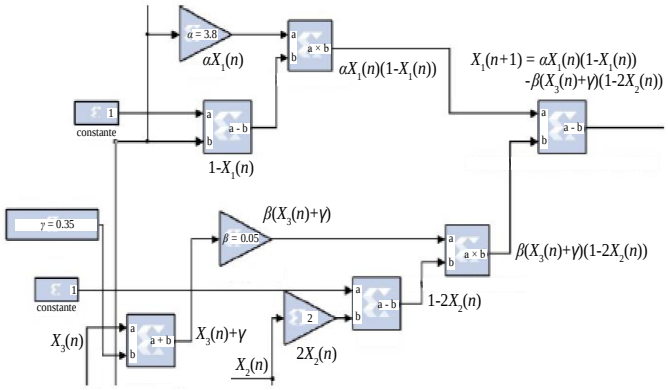

Figura 1. Estado $X_1(n)$ del mapeo Rössler, modelado con SysGen.

A partir del ToolBox de SysGen y la compilación respectiva, generamos el código VHDL del sistema, creando una entidad que denominamos Rössler.xixe.

2.3. IMPLEMENTACIÓN DEL SISTEMA CAÓTICO EN LA FPGA

La programación embebida en la FPGA se lleva a cabo empleando iMPACT, herramienta de ISE Design Suite: System Edition y corresponde a una plataforma de diseño de alto desempeño para la familia de FPGA's de Xilinx. La programación se lleva acabo a partir de la conectividad entre una PC y la FPGA vía el puerto USB, con protocolo de comunicación JTAG (Joint Test Action Group). Esta conexión corresponde al cable color negro que se muestra en la Figura 2. Por otra parte, la adquisición de resultados que se ejecutan en la FPGA como por ejemplo los derivados del ciclo de encriptado-desenciptado, se obtienen a través del protocolo de comunicación RS-232 y que corresponde en este caso al cable de color blanco que se muestra en la misma figura.

Figura 2. Conexiones entre la PC y la FPGA Spartan 3E-1600.

A fin de validar y evaluar la funcionalidad operativa del sistema experimental que se presenta en la Figura 2, se lleva acabo una comparación del atractor de Rössler generado por la FPGA versus el que se obtiene por cálculo numérico en una PC. En la Figura 3, se muestra la superposición de los atractores que siguen el sistema de ecuaciones (1).

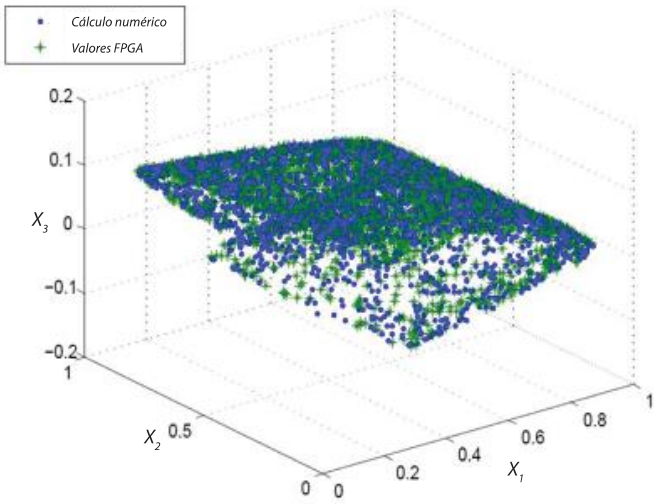

Figura 3. Comparación del atractor extraño del sistema caótico de Rössler obtenido en FPGA versus PC.

Se observa claramente que el comportamiento caótico que siguen los estados $X_1(n), X_2(n)$ y $X_3(n)$ generados en la FPGA y los obtenidos por cálculo numérico, dan forma al mismo atractor extraño de Rössler (una especie de hoja doblada). Esto nos permite validar y certificar que el proceso experimental implementado es congruente a lo esperado teóricamente. Se observa sin embargo, una diferencia entre los valores específicos para cada uno de los estados del sistema caótico generados por la FPGA con respecto a los obtenidos con la PC. Esta discrepancia numérica se deriva de los niveles propios de precisión que maneja la PC y la FPGA respectivamente. Para el caso de las PC's, la manipulación numérica se rige por el estándar IEEE 754 [7] para operaciones con punto flotante. Para el caso de la FPGA Spartan 3E-1600, las operaciones aritméticas son con punto fijo y para los cálculos específicos reportados en este trabajo se implementó una mantisa del tipo 22Q20, es decir, 1 bit para signo, 21 bits para la parte entera y 20 bits para la parte decimal.

Parte de la operatividad interna en la FPGA se presenta en la Figura 4. El diagrama muestra un fragmento de los elementos básicos que integran la secuencia interna de ejecución. El sistema consta de 3 entidades generales: *i*) UART (Universal Asynchronous Receiver Transmitter), *ii*) encriptado-desencriptado (Init-all) y *iii*) Rössler.xixe.

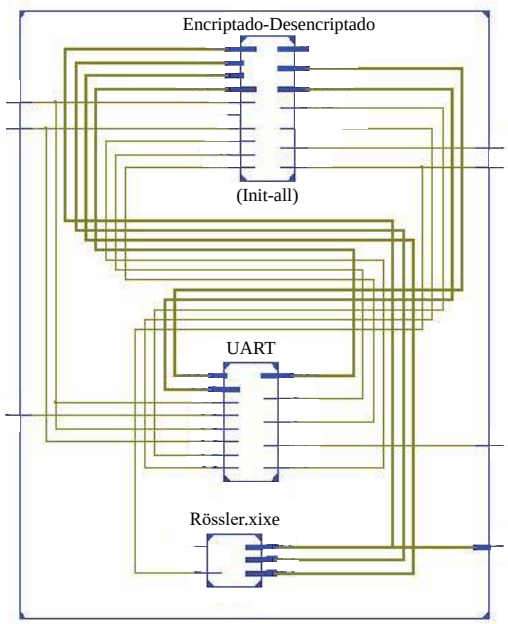

Figura 4. Operatividad general interna del sistema de ejecución de la FPGA Spartan 3E-1600.

Las entidades UART e Init_all son parte del protocolo de comunicación RS-232 integradas por la librería GNU [8] y UART para System on Chip (SoC) integrada con lenguaje VHDL. Una de las funciones de la entidad UART es hacer posible la comunicación directa entre la PC y la FPGA. En este caso, la comunicación se lleva acabo a través del protocolo de comunicación RS-232. En la Figura 2, se muestra esta conexión de comunicación vía el convertidor RS-232 a USB que corresponde al cable de color blanco. En la entidad Init_all en conjunto con la entidad Rössler.xixe, se ejecuta el proceso de encriptado o desencriptado según sea el caso, de la imagen respectiva.

2.4. ALGORITMO EMBEBIDO

En la Figura 5, se muestra en un diagrama a bloques el algoritmo embebido implementado en la FPGA para llevar a cabo el proceso de encriptado caótico para imágenes digitales. El proceso de cifrado se inicia con la entrada de la imagen digital a encriptar. La imagen se envía pixel por pixel de la PC a la FPGA a través del puerto RS-232. Por otra parte, es conocido el hecho de que en un sistema criptográfico se requiere una llave o una clave de acceso que es solamente conocida por los interesados en salvaguardar la información. Dado que el cifrado de imágenes que presentamos en este trabajo se basada en caos, la clave de acceso queda establecida por los valores asignados a los parámetros del sistema caótico que definen su dinámica, así como por las condiciones iniciales con las que se da inicio a la generación de las secuencias caóticas respectivas. Específicamente la clave de acceso dado el sistema de Rössler implementado y definido por el sistema de ecuaciones (1), se conforma por: $\alpha = 3.8, \beta = 0.05, \gamma = 0.35, \delta = 3.78, \zeta = 0.2, \eta = 0.1$ y $\theta =$

0.1, para los valores de los parámetros del sistema y $X_1(0) = 0.1, X_2(0) = 0.15$ y $X_3(0) = 0.01$, para los valores de las condiciones iniciales. Acorde a la implementación experimental que se presenta en este trabajo, la clave de "acceso" está implícita en el algoritmo embebido de ejecución específicamente en la entidad Rössler.xixe, como se muestra en la Figura 5.

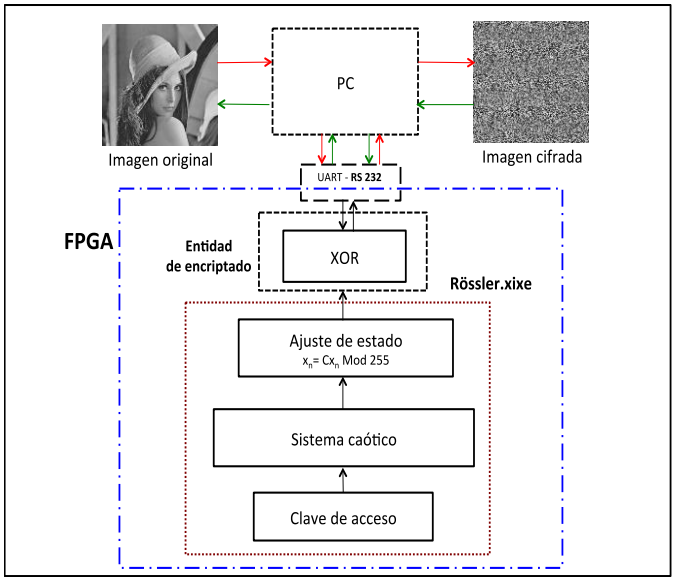

Figura 5. Diagrama a bloques del algoritmo embebido de encriptado-desencriptado de imágenes digitales.

La imagen original una matriz de números $(a_{i,j})$, con $i = 1, \ldots, m$ y $j = 1, \ldots, n$, se re-ordena en un vector columna (a_{l1}), con $l = m$ x n. Cada número de este vector es un pixel codificado por un byte, permitiendo así 256 posibilidades (0 a 255) de variación en escala de grises. Cuando un pixel ingresa a la FPGA por el puerto RS-232, en base a la sincronización entre las entidades UART, Init_all y Rössler.xixe (Figuras 2, 4 y 5), se genera un estado X_n caótico cuyo valor numérico decimal se adecúa a una escala binaria de magnitud 8 bits. El ajuste se lleva acabo a partir de una operación de conversión definida por: $x_n = C(X_n mod\ 255)$, donde $C = 10\text{x}10^6$. Bajo estas condiciones a partir de la operación lógica XOR, cada pixel (a_{l1}) de la imagen original se va enmascarando por el valor numérico del estado caótico (x_n) generado. Finalmente cada cadena de bits encriptados se reenvía a la PC a través del puerto RS-232 para integrar la imagen encriptada. Experimentalmente seleccionamos el estado X_1 definido en el sistema de ecuaciones (1), como secuencia pseudoaleatoria base para llevar a cabo el ciclo de encriptado-desencriptado, no existiendo limitante para utilizar cualesquiera de los otros dos estados. El algoritmo propuesto tiene una estructura lógica simple, debido a que la complejidad del encriptado de una imagen recae en esencia, en la complejidad propia del sistema caótico implementado. Una de las ventajas del algoritmo propuesto, es el hecho de mantenerse invariante en su estructura cuando se lleva acabo el proceso inverso de desencriptado. Es decir, el algoritmo que se muestra en la Figura 5, es el mismo tanto para el proceso de encriptado de una imagen, como para el proceso inverso de desencriptado del criptograma respectivo. La diferencia en un momento dado, radica en el "tipo" de imagen de entrada. Bajo este contexto, el sistema experimental que se presenta en este trabajo y que se muestra en la Figura 2, opera como un sistema periférico de una computadora unificado encriptador-desencriptador caótico para imágenes digitales. El hecho de que el proceso de encriptado esté embebido en la FPGA, adiciona al proceso en sí, un nivel más de seguridad con respecto a los algoritmos implementados para tal fin. La operatividad interna del algoritmo, hace uso de la propiedad bidireccional de la compuerta lógica XOR que se aplica en algoritmos de encriptado [9].

3. RESULTADOS

En la Figura 6, se muestran la imágenes digitales empleadas para validar experimentalmente el sistema criptográfico implementado. Las imágenes son: *i*) "Lena" de 256x256 pixeles, *ii*) "Cameraman" de 512x512 pixeles y *iii*) "Mandril" 512x512 pixeles.

Cada una de las imágenes se procesa individualmente siguiendo el algoritmo que se presenta en la Figura 5. La imagen original (Lena, Cameraman o Mandril) ingresa al algoritmo embebido en la FPGA pixel a pixel para enmascararse por una de las secuencias pseudoaleatorias generadas por el mapeo caótico de Rössler.

Figura 6. Imágenes empleadas en el sistema experimental: a) Lena, b) Cameraman y c) Mandril.

Por su parte, cada pixel encriptado emerge del sistema encriptador ingresando a la PC para integrar el criptograma respectivo. La operatividad numérica del algoritmo embebido, está acotada por las limitaciones físicas propias de la FPGA Spartan 3E-1600 implícitas en las operaciones de punto fijo 22Q20. En este sentido, los valores reales utilizados para las condiciones iniciales que forman parte de la clave de "acceso" del sistema criptográfico se presentan en la Tabla 1.

Tabla1. Valores reales de las condiciones iniciales del sistema criptográfico.

Condición inicial	Teórico	Experimental 22Q20
$X_1(0)$	0.1	0.100000000000364
$X_2(0)$	0.15	0.149999999999634
$X_3(0)$	0.01	0.010000000000218

Para el caso de la imagen de "Lena" y en relación a su tamaño (256x256 pixeles) el algoritmo ejecuta 65536 iteraciones, ya sea para encriptar o desencriptar la imagen respectiva. El criptograma que se obtiene bajo este proceso, así como la imagen recuperada para el caso del proceso inverso, se muestran en la parte superior de la Figura 7.

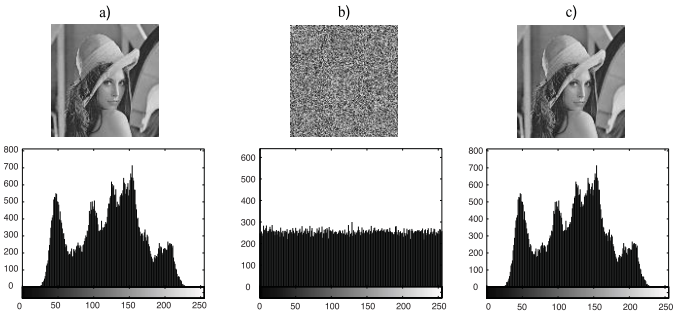

Figura 7. Parte superior: a) imagen de Lena original, b) criptograma y c) imagen recuperada. Parte inferior: a) histograma de la imagen original, b) histograma del criptograma y c) histograma de la imagen recuperada.

A partir del análisis de los histogramas correspondientes y que se muestran en la parte inferior de la figura 7, se puede observar que el criptograma enmascara totalmente la imagen original de "Lena" como se hace evidente a partir del comportamiento uniforme de su histograma. Así mismo, se percibe que no existe diferencia entre la imagen recuperada con respecto a la original, a partir de la igualdad entre sus histogramas respectivos. De manera análoga se procede con las Figuras 6b y 6c. Para el "Cameraman" y el "Mandril", el algoritmo ejecuta 262144 iteraciones para cada uno de los procesos respectivos. En la Figura 8, se muestra el criptograma y la imagen recuperada para "Cameraman".

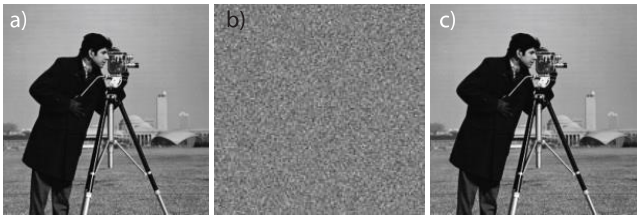

Figura 8. a) imagen de Cameraman original, b) imagen cifrada (criptograma) y c) imagen recuperada.

En la Figura 9, se muestra el criptograma y la imagen recuperada para el "Mandril".

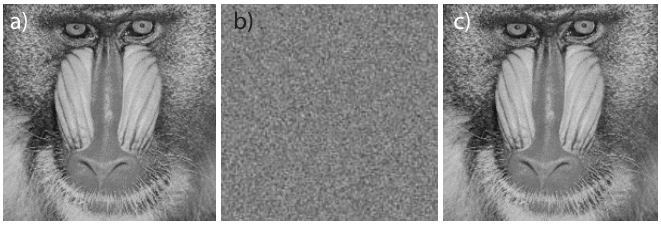

Figura 9. a) imagen de Mandril original, b) imagen cifrada (criptograma) y c) imagen recuperada.

De los resultados obtenidos experimentalmente y que se muestran en las Figuras 7, 8 y 9, se infiere la funcionalidad del sistema criptográfico propuesto.

3.1. ANÁLISIS DE SEGURIDAD

El sistema criptográfico implementado en este trabajo en base a la FPGA Spartan 3E-1600, cumple operativamente con el objetivo de encriptar y desencriptar caóticamente imágenes digitales. Para evaluar cuantitativamente la capacidad del sistema de resistir los diferentes ataques de usuarios no autorizados a fin de extraer información confidencial de las imágenes encriptadas, se aplican los métodos más comunes de análisis de seguridad a los criptogramas obtenidos [10-11].

3.1.1. ATAQUES DE FUERZA BRUTA [12]

3.1.1.1. ANÁLISIS DE ESPACIO DE CLAVES

El límite operativo de la FPGA Spartan 3E-1600 para valores numéricos, está acotada por su representación a partir de una mantisa del tipo 22Q20 en formato de punto fijo de 42 bits. Bajo estas circunstancias, experimentalmente el valor mínimo operacional corresponde a $953.674317002995\text{x}10^{-09}$. Este valor, en combinación con las 10 variables que forman la clave de "acceso" definidas por el sistema caótico de Rössler, nos permite obtener que el sistema criptográfico contiene un espacio de claves de 397 bits. Este dato, es mayor a los 128 bits típicamente necesarios para que el sistema sea considerado como viable para aplicaciones criptográficas.

3.1.1.2. ANÁLISIS DE SENSIBILIDAD

La sensibilidad del sistema criptográfico implementado se analiza en este caso, a partir de la imagen encriptada (criptograma). Cuando la imagen encriptada de "Lena" por ejemplo (Figura 7b), ingresa al algoritmo con la clave de "acceso" correspondiente, se obtiene a la salida la imagen original (Figura 7c).

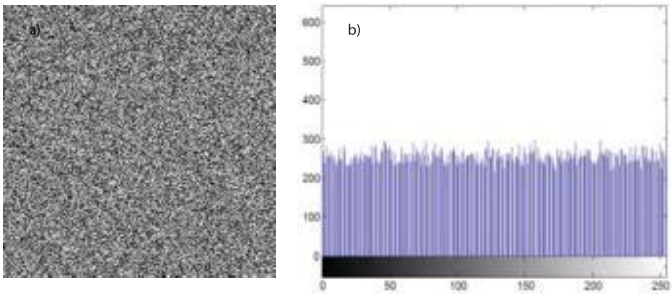

Figura 10. a) imagen recuperada con un cambio mínimo en $X_1(0) = 0.1$ y b) histograma de la imagen recuperada.

Sin embargo, al efectuar un mínimo cambio en el valor de alguna de las condiciones iniciales, en este caso $X_1(0) = 0.1$ (Tabla1), manteniendo sin variación el resto de las variables de "acceso", se obtiene la Figura 10a. Esto es, la imagen original no se recupera, por lo que el sistema es muy sensible a pequeñas variaciones en alguna de sus condiciones iniciales. Específicamente el valor mínimo efectuado en $X_1(0) = 0.1$, que admite el sistema implementado se muestra en la Tabla 2.

ISSN 1405-2172

Tabla2. Sensibilidad del sistema criptográfico.

Condición inicial teórica	$X_1(0)$ en la FPGA	
	Experimental	Cambio mínimo
$X_1(0) = 0.1$	0.100000000000364	0.099999046326047
	La Figura 7b genera la Figura 7c	La Figura 7b genera la Figura 10a

La diferencia numérica en las condiciones iniciales de la Tabla 2, corresponde al valor mínimo operacional de la FPGA $953.674317002995\text{x}10^{-09}$. Por otra parte, en la Figura 10b, se muestra la distribución de frecuencias correspondiente a la Figura 10a, notándose una distribución de frecuencias uniforme sobre toda la escala de grises. Esta condición es muy deseable para el sistema criptográfico ya que para valores muy próximos a la clave de "acceso", no se obtiene indicio alguno de información confidencial.

3.1.2. ANÁLISIS ESTADISTÍCOS

3.1.2.1. ANÁLISIS DE CORRELACIÓN DE PIXELES ADYACENTES [13]

En la Figura 11, se muestra una gráfica de dispersión asociada al criptograma de la Figura 7b. El diagrama se genera con la selección aleatoria de 2025 pares de pixeles (X_i, Y_i) horizontales y adyacentes correspondientes al criptograma. El diagrama presenta un resultado satisfactorio para el sistema de encriptado de imágenes, ya que valida visualmente la ausencia de información que pudiera comprometer la confidencialidad de la de la imagen original. Este resultado es análogo en gran medida al presentado en la Figura 10b. Numéricamente evaluamos los coeficientes de correlación (r_{xy}) para pixeles horizontales, verticales y diagonales, asociados a cada uno de los criptogramas mostrados en las Figuras 7b, 8b y 9b.

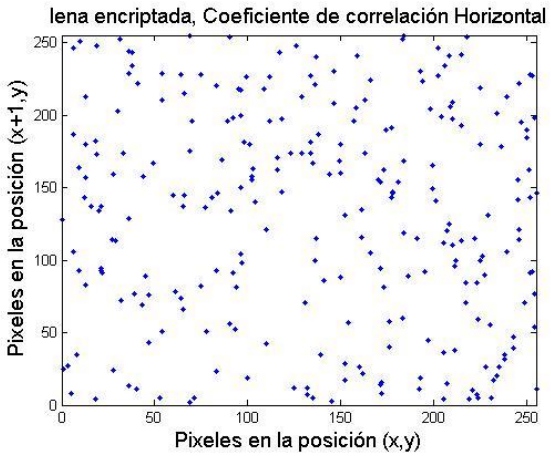

Figura 11. Diagrama de dispersión de pixeles horizontales.

En la Tabla 3, se presentan los valores correspondientes de los coeficientes de correlación. En todos los casos, se observa que los valores obtenidos son muy próximos a cero, lo que es equivalente a desplegar diagramas de dispersión del tipo dado por la Figura 11, para cada caso respectivo. Los valores próximos a cero, son altamente benéficos para el sistema criptográfico implementado.

Tabla 3. Coeficientes de correlación.

Pixeles	Lena	Cameraman	Mandril
Horizontal	**-9.85**39x10^{-02}	**-4.92**63x10^{-03}	**-4.01**11x10^{-02}
Vertical	**-7.47**48x10^{-02}	**-9.21**14x10^{-02}	**8.27**5x10^{-03}
Diagonal	**5.93**99x10^{-02}	**-3.77**23x10^{-02}	**-3.59**1x10^{-02}

3.1.2.2. ENTROPÍA DE LA INFORMACIÓN [14]

Es conocido y aceptado el hecho de que la Entropía de la información es un criterio que muestra la aleatoriedad de los datos y que para imágenes con pixeles completamente aleatorios en escala de grises de 8 bits, arroja un valor teórico de Entropía H(s)=8 bits. En la tabla 4, se muestran los valores correspondientes de la Entropía evaluada para los criptogramas que se muestran en las Figuras 7b, 8b y 9b, generados por el sistema criptográfico implementado.

Tabla 4. Entropía de las imágenes encriptadas.

Imagen Encriptada	Entropía
Lena (Figura 7b)	**7.99756**955424879
Cameraman (Figura 8b)	**7.99870**847533247
Mandril (Figura 9b)	**7.99918**512208053

En la tabla 4, se observa claramente que los valores obtenidos de la Entropía para cada criptograma, se aproxima en gran medida al valor ideal de 8. Esto certifica la funcionalidad operativa y el nivel de seguridad del sistema criptográfico experimental.

3.1.2.3. ATAQUES DIFERENCIALES [15]

Acorde a la metodología para llevar acabo el análisis contra ataques diferenciales, se generan dos criptogramas (C_1 y C_2) correspondientes a una misma imagen a partir de claves de acceso muy similares. Siguiendo a manera de ejemplo la imagen de "Lena" (Figura 7a) y en relación al análisis de sensibilidad plasmado en la Tabla 2, C_1 lo asociamos a la Figura 7b y C_2 a la Figura 10a. A partir de estos criptogramas calculamos el NPCR (Number of Pixels Change Rate) y el UACI (Unifed Average Changing Intensity). En la Tabla 5, se plasman los valores obtenidos para estas unidades, evaluados para las imágenes mostradas en la Figura 6.

Tabla5. Ataque diferencial a las imágenes cifradas.

Imagen	NPCR %	UACI %
Lena	**99.6261**596679688	**33.5093**120500231
Cameraman	**99.6269**226074219	**33.5481**561399082
Mandril	**99.6273**040771484	**33.5679**297353819

Los resultados mostrados en la Tabla 5, son altamente satisfactorios ya que son muy próximos a los valores teóricos esperados para estas medidas. Para el NPCR su valor teórico esperado es el 100 %, mientras que para el UACI corresponde al 34 %.

4. CONCLUSIONES

El sistema criptográfico caótico implementado en la FPGA Spartan 3E-1600 de Xilinx, cumple con los requerimientos de ejecución del ciclo de encriptado-desencriptado de imágenes digitales, obteniéndose estándares de seguridad muy competitivos frente a diferentes tipos de ataques. La potencialidad del sistema se integra a partir de las bondades de los elementos que lo componen: *i*) la complejidad del sistema caótico empleado, *ii*) la tecnología emergente de las FPGA's y *iii*) un algoritmo embebido de estructura operacional simple. Por lo que, el empleo de portadoras caóticas embebidas en este tipo de tecnologías, son un medio prometedor para proporcionar privacidad y seguridad en el cifrado de información confidencial. Bajo esta perspectiva, las implementaciones basadas en este tipo de tecnología de vanguardia, son completamente viables a integrarse a sistemas que tiendan a resolver los problemas derivados de las necesidades del manejo de información en forma segura y confidencial.

5. AGRADECIMIENTOS

Este trabajo fue apoyado por el proyecto de investigación aprobado en la 18va Convocatoria Interna de Proyectos de Investigación de la UABC, con el número 485 y vigente para los años 2015-2017. El investigador R.O.E. es apoyado por CONACyT en sus estudios de posgrado a nivel Doctoral.

6. REFERENCIAS

[1] E. N. Lorenz, Deterministic nonperiodic flow, Journal of the atmospheric sciences, Vol. 20, 1963, 130-141.
[2] S. H. Strogatz. Nonlinear Dynamics and Chaos whit applications to Physics, Biology, Chemistry, and Engineering. Westview Capitulo 8.
[3] O. Rossler, An equation for hyperchaos *Physics Letters A, Elsevier, 71*, 1979, 155-157.
[4] E. Inzunza González, Encriptado caótico en sistemas biométricos, Tesis de Doctorado, Facultad de Ingeniería, Arquitectura y Diseño (FIAD), Universidad Autónoma de Baja California, Ensenada, B.C., México, Enero 2013.
[5] E. Rodriguez, E. Garcia, and E. Inzunza, Image Encryption Based on Improved Rössler Hyperchaotic Map., Difu100cia Vol. 8 No. 2 - Sep-Dic. 2014, 8–15.
[6] Xilinx. Xilinx, www.xilinx.com, 23 de mayo 2015.
[7] IEEE, "ESTÁNDAR IEEE 754", https://www.ieee.org/index.html , 23 de mayo de 2015.
[8] Cosmiac. Cosmiac, http://www.cosmiac.org. 23 de mayo 2015.
[9] M.S. Azzaz, C. Tanougast, S. Sadoudi, A. Bouridane, and A. Dandache. FPGA implementation of new real-time image encryption based switching chaotic systems. In Signals and Systems Conference (ISSC 2009), IET Irish, pages 1–6, 2009.
[10] S Behnia, A Akhshani, H Mahmodi, and A Akhavan. A novel algorithmfor image encryption based on mixture of chaotic maps. Chaos, Solitons &Fractals, 35(2):408–419, 2008.
[11] C.E. Shannon. A mathematical theory of communication, 1948. Bell System Technical Journal, The.
[12] V. Patidar, NK Pareek, G Purohit, and KK Sud. A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption. Optics Communications, 284(19):4331–4339, 2011.
[13] C. Fu, Bin-bin Lin, Yu-sheng Miao, Xiao Liu, and Jun-jie Chen. A novel chaos-based bit-level permutation scheme for digital image encryption. Optics Communications, 284(23):5415–5423, 2011.
[14] Z. Mao, Y. y Deng. A new image encryption algorithm of input-output feedback based on multi-chaotic system. Applied Mechanics and Materials, 40-41:924–929, 2011.
[15] Y. Mao y C. K. Chui, G. A Chen, symmetric image encryption based on 3d chaotic cat maps. Chaos, Solitones and Fractals, 21(3):749–761, 2004.