Revista de Ingeniería y Tecnologías para el Desarrollo Sustentable 3 (2017) 39-46

ISSN: 2448 - 7198

IMPLEMENTACIÓN EN HARDWARE DE UN (2 DE 2)-ESQUEMA DE CRIPTOGRAFÍA VISUAL PARA AUTENTICACIÓN DIGITAL DE USUARIOS

Cuadros-Romero, F. J.[a], Salazar-Pérez, P. J.[a], y Soto-Ortiz, S. I.[a]

[a]. Instituto Tecnológico Superior del Occidente del Estado de Hidalgo de la división de Ingeniería en Tecnologías de la Información y Comunicaciones (ITIC), Mixquiahuala de Juárez, Hidalgo, 42700. fcuadros@itsoeh.edu.mx

Recibido 3 de Noviembre de 2017; aceptado 29 de Diciembre de 2017

Palabras clave:
Criptografía visual, FPGA, Autenticación

RESUMEN. *En este trabajo, se presenta la implementación de un (2 de 2)-esquema de criptografía visual en un arreglo de compuertas programable en campo o FPGA (por sus siglas en inglés, Field Programmable Gate Array). La implementación permite cifrar una imagen en blanco y negro (p.ej. textos impresos, notas escritas a mano, fotos, etc.) en dos imágenes que contienen patrones de puntos aleatorios, en esencia, las dos imágenes resultantes que contienen la información del usuario lucen mucho como el ruido que era anteriormente observado en los televisores analógicos. El descifrado de la imagen original o ¨mensaje¨, es llevado a cabo utilizando el sistema visual humano: el usuario debe tomar las dos imágenes resultantes o ¨sombras¨ e imprimirlas en transparencias (sobre láminas de acetato), posteriormente, superponerlas y alinearlas cuidadosamente para poder ver el mensaje original. El proceso de descifrado anteriormente descrito, puede ser remplazado de manera digital, efectuando la operación lógica OR, entre las dos versiones digitales de las sombras, con lo cual, el mensaje original sería presentado al usuario a través de un monitor. Las ventajas de la implementación propuesta contra las existentes (fundamentalmente en computadoras), son básicamente dos: incremento en la seguridad de cifrado (debido al uso de hardware de propósito específico (FPGA) en lugar de uno de propósito general como la computadora) y el incremento en la velocidad del proceso de cifrado, el cual es hasta un 350% mayor. La implementación propuesta está orientada principalmente pero no limitada a aplicarse en procesos de autenticación digital de usuarios: movimientos bancarios, compras en línea, autenticación en sitios web, entre otros.*

Key words:
Visual cryptography
FPGA, Authentication

ABSTRACT. *In this work, we present an FPGA implementation of a 2 out of 2 visual cryptography scheme. This hardware implementation allows encrypting a blank and white image into a set of random noisy images called shares. The resulting shares look much like the interference watched in the old analog television. The decryption process to retrieve the original image or "message" is carried out by using the human visual system; this is, the user takes the two previously printed transparencies and align them carefully to be able to see the original image. The above decryption process in a digital manner can be replaced by performing the OR operation between the two digital random looking dot images, in such a way that the original message will be displayed in a screen monitor. The advantages of the proposed hardware implementation over the existing computer based ones are basically two: increased security in the encryption process and decreased time processing. The first advantage comes from the fact that we use a specific purpose computing hardware instead of a general one. The second advantage lies on the ability that the FPGAs have to process data in a parallel fashion; the speedup achieved by the proposed implementation is up to 350% faster than the computer based implementations. The proposed FPGA implementation pursuits digital user authentication applications such as bank transfers, online shopping, and authentication on websites, among others*.

INTRODUCCIÓN

Un esquema de criptografía visual o VCS (por sus siglas en inglés, visual cryptography scheme) es un proceso criptográfico que permite cifrar una imagen en un conjunto de N imagines; donde N indica el número de usuarios entre los cuales se desea dividir la información contenida dentro de la imagen secreto[1]. Un VCS es una extensión de un esquema de secreto compartido o SSS[2] (por sus siglas en inglés, secret sharing scheme). La diferencia entre ambos esquemas radica en el tipo de información a cifrar; en el caso de un VCS, la información a cifrar es una imagen, mientras que en un SSS, la información es una cadena de caracteres. Los VCS poseen dos características importantes: cifrado completamente seguro (esto desde un punto de vista matemático) y un proceso de descifrado muy sencillo (no requiere de un algoritmo complejo ni del uso de algún tipo de dispositivo electrónico como podría ser una computadora personal; solo el sistema visual humano es requerido para el proceso de descifrado). Los VCS por si mismos son una gran opción para aplicaciones de secreto compartido; donde la

autenticación depende de más de una persona/parte/dispositivo. La mejor forma de entender la aplicación de los VCS es considerando un ejemplo concreto. Supongamos que deseamos utilizar VCS para autenticar el pasaporte de una persona. La imagen original a cifrar en este caso específico sería una imagen que contenga el texto: “Pasaporte Valido”. Posteriormente, esa imagen se cifraría en dos imágenes (sombras) de tal manera que una se imprimiría en el pasaporte del usuario y la segunda se guardaría en una base de datos. La autenticación del pasaporte valido seria de la siguiente manera: se tomaría el pasaporte con su respectiva imagen, y la imagen almacenada en la base de datos correspondiente al nombre del usuario que aparece en el pasaporte, acto seguido, se llevaría a cabo la alineación de ambas imágenes, de tal forma, que en el caso de un pasaporte valido, la alineación de las dos imágenes mostraría el texto: “Pasaporte Valido”, en caso contrario, la imagen resultante no mostraría texto alguno. La implementación de este sistema de autenticación de pasaportes puede ser relativamente fácil de implementar en una computadora. El problema radica en que una computadora de forma general es muy vulnerable debido a su hardware y su software. La vulnerabilidad en el hardware se debe a que una computadora posee una gran variedad de puertos que ofrecen una puerta trasera para el acceso a los datos en ella almacenados. La vulnerabilidad del software, proviene de la gran variedad de errores en los sistemas operativos que estas poseen. Es así, como este trabajo tiene como objetivo implementar un (2 de 2)-esquema de criptografía visual en un arreglo de compuertas programable en campo o FPGA (por sus siglas en inglés, Field Programmable Gate Array) para incrementar la seguridad de cifrado y reducir la velocidad de procesamiento. Esto, a través del aprovechamiento del procesamiento en paralelo y la arquitectura cerrada y de propósito específico que puede ser implementada en un FPGA. De esta manera, tanto la generación de las sombras (cifrado), como la autenticación (superposición), se llevarían a cabo en un circuito integrado.

METODOLOGÍA

El (2 de 2)-esquema de criptografía visual. El modelo de un (2 de 2)-esquema de criptografía visual puede ser descrito de la siguiente manera. Asumimos que la imagen de entrada a este esquema es una imagen en blanco y negro (ver por ejemplo Figura 1-a); es decir, es una colección de pixeles blancos y negros sobre los cuales se puede operar de manera independiente. Cada pixel de la imagen de entrada (o imagen secreto) aparecerá en N versiones modificadas (sombras). En este trabajo N es igual a 2 y por lo tanto generaremos dos sombras S_1 y S_2. Si N-1 sombras son sobrepuestas es imposible recuperar la imagen original, para efectos de nuestro trabajo, esto significa que una sola sombra ya sea S_1 o S_2, no permite reconstruir la imagen secreto.

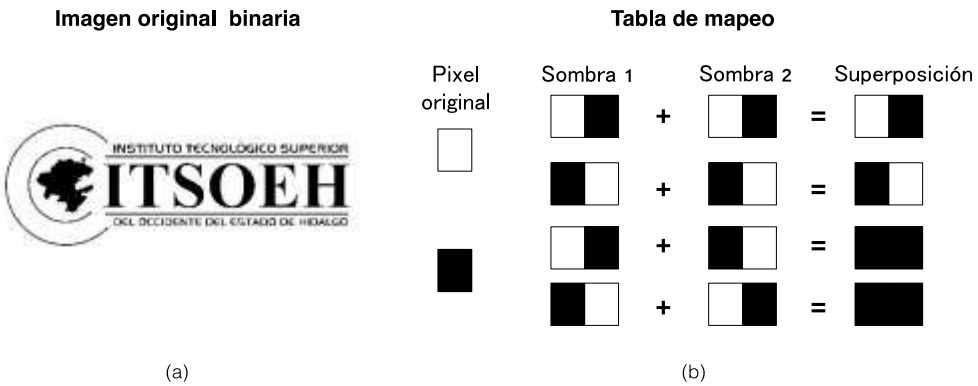


Figura 1. Imagen original binaria (a) y tabla de mapeo (b) de los pixeles originales a los pixeles en las sombras S_1 y S_2.

ISSN: 2448 - 7198

Cuando cada uno de los pixeles de la imagen binaria original es cifrado en las dos sombras S_1 y S_2, es necesario considerar la expansión de cada uno de los pixeles; en otras palabras, cada pixel original será representado con 2 pixeles en cada una de las sombras S_1 y S_2, de tal manera que las sombras tendrán el doble de columnas que la imagen original. En la Figura 1-b, se muestra la tabla de mapeo para cada uno de los pixeles de la imagen secreto según su color, así como el correspondiente par de pixeles que se asignaran a cada una de las sombras.

En la tabla de mapeo que aparece en la Figura 1-b, se puede apreciar que un pixel blanco perteneciente a la imagen secreto en la Figura 1-a, puede ser cifrado de dos maneras diferentes en la sobra 1: la primer opción, sería un pixel blanco y un pixel negro, mientras que la segunda opción, sería un pixel negro y un pixel blanco. De la misma forma, el cifrado de un pixel blanco en la sombra 2 tiene dos opciones. Esta peculiar característica de cifrado de los pixeles de la imagen secreto permite que las sombras tengan un aspecto ruidoso al seleccionar aleatoriamente entre las dos posibles opciones de pares de pixeles para cada sombra. Para ejemplificar la aleatoriedad de las sombras, en la Figura 2, se muestran las dos sombras resultantes (Figura 2-c y Figura 2-d) para la imagen secreto (Figura 2-a), así como la imagen resultante (Figura 2-b) de la operación OR llevada a cabo entre las dos sombras que aparecen en la Figura 2-c y Figura 2-d respectivamente.

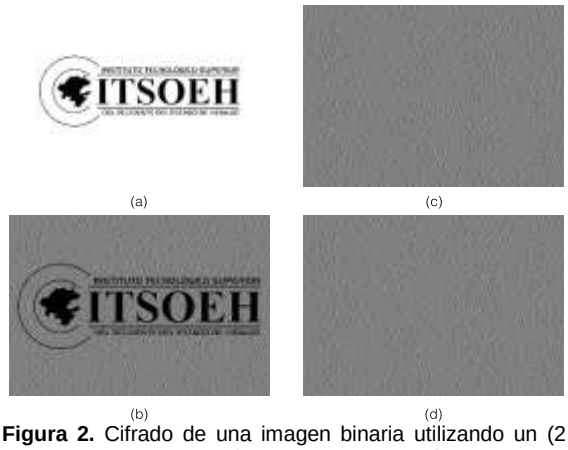


Figura 2. Cifrado de una imagen binaria utilizando un (2 de 2)-esquema de criptografía visual. Descripción del proceso: (a) imagen secreto, (b) resultado de la superposición de las sombras (c) y (d). (c) y (d): sombras resultantes de la aplicación de un (2 de 2)-esquema de criptografía visual.

Existen varias aplicaciones para un (2 de 2)-esquema de criptografía visual; tales como la autenticación de contenidos digitales[3], protección para los derechos de autor[4], comercio movil[5], transferencias bancarias en internet[6], entre otras. Aunque todas las aplicaciones antes mencionadas son viables, existe una sola barrera que a algunas de ellas les ha impedido realmente llegar a ser adoptadas de manera significativa; tal barrera es, la seguridad de la implementación del VCS. Es por esta razón, que el desarrollo de una implementación en hardware de un VCS es tan necesario. Esta, permitiría que las aplicaciones arriba listadas fueran completamente viables y escalables dentro de un mercado más demandante con plena certeza de que funcionan sobre una plataforma segura. En este contexto, seguro se puede entender como la capacidad del circuito integrado a no poder ser hackeado con virus de computadora convencionales, los cuales fueron diseñados para explotar vulnerabilidades de los sistemas operativos. Continuando sobre la misma línea, un circuito integrado de propósito específico, no contiene puertos de comunicación innecesarios que puedan ser manipulados para interferir en el proceso de cifrado o autenticación.

En resumen, debido a la gran variedad de aplicaciones potenciales de un (2 de 2)-esquema de criptografía visual, nace la necesidad de generar un circuito integrado que cumpla dos objetivos específicos: uno, incrementar la seguridad de cifrado de las imágenes secreto, dos, reducir el tiempo del proceso de cifrado. A continuación, se describe la arquitectura del circuito integrado que logra los dos objetivos específicos antes mencionados.

Arquitectura y modelado de la implementación propuesta. Dado que en un (2 de 2)-esquema de criptografía visual es necesario generar dos sombras ruidosas, dos matrices base M_0 y M_1 deben ser construidas. En la Figura 3, se muestran las matrices base M_0 y M_1 así como sus respectivas permutaciones. Nota que la permutación de las matrices base M_0 y M_1 se ejecuta sobre las columnas. Durante la construcción de las sombras S_1 y S_2 la permutación de las columnas es dominada por una señal aleatoria que toma valores entre 0 y 1; donde 0, es la matriz base original y uno, es la versión permutada de la matriz base.

Sin permutar (0) **Con permutación (1)**

$$M_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \qquad \hat{M}_0 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \hat{M}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Figura 3. Matrices base y sus respectivas permutaciones para la implementación de un (2 de 2)-esquema de criptografía visual.

En la Figura 4-a se muestra el diagrama a bloques de la implementación propuesta. La entrada digital etiquetada como In2, representa la señal aleatoria que define la selección entre las matrices base y sus permutaciones. Si el valor de In2 es cero, se utilizan las matrices de la columna izquierda que aparecen en la Figura 3, en caso contrario, se utilizan las matrices en la columna derecha mostradas en la Figura 3.
El diagrama a bloques de la implementación propuesta para el (2 de 2)-esquema de criptografía visual, esta subdividido a su vez en tres secciones: sección de combinaciones, sección de permutaciones y sección de operaciones. A continuación cada una de las tres secciones es descrita a detalle.

Sección de combinaciones. La sección de combinaciones (ver Figura 4-b) tiene dos señales digitales de entrada: la primera, correspondiente a la imagen secreto (etiquetada como Image), y la segunda, perteneciente a la señal aleatoria para la selección de las matrices de cifrado M_0 y M_1 (etiquetada como In2). La sección de combinaciones es básicamente un multiplexor de 2 entradas a cuatro salidas. Este envía los valores de Image e In2 al bloque Permutation 1 a través de las salidas out1 y out2 si In2 es igual a uno; en otro caso, los valores de Image e In2 son enviados al bloque Permutation 2 a través de las salidas out3 y out4.

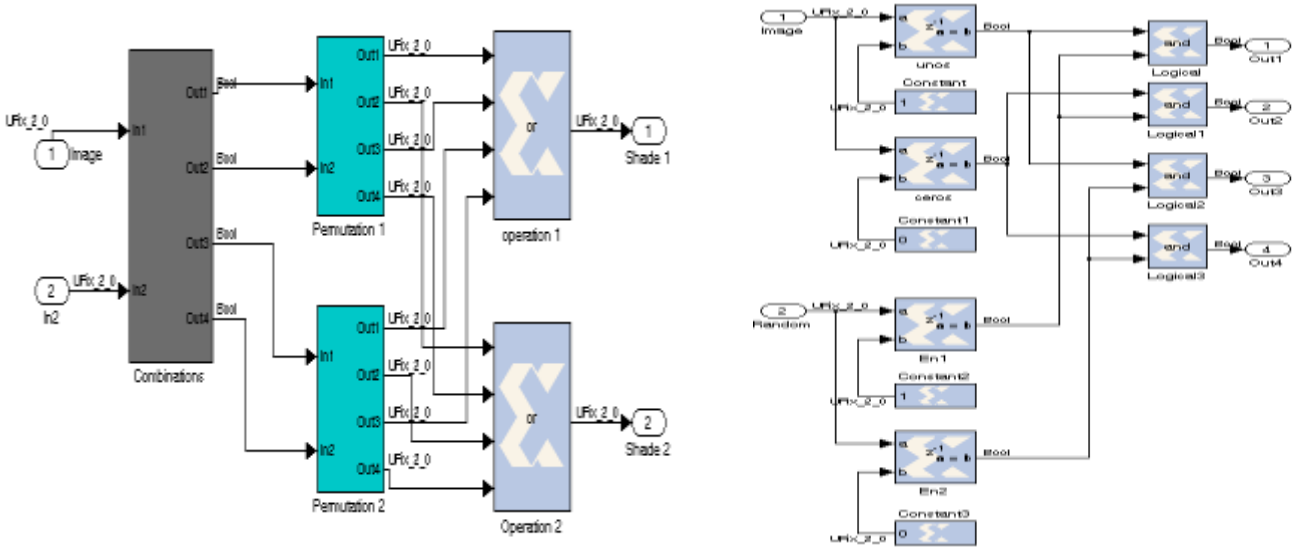
Sección de permutaciones. Como se puede apreciar en la Figura 5, la sección de permutaciones está compuesta por los bloques Permutation 1 y Permutation 2. Permutation 1 es usado cuando el valor de In2 es igual a uno, mientras que el valor del pixel de la imagen secreto determina si se utilizara la matriz base M_0 o M_1. Cuando el valor de In2 es igual a cero el bloque Permutation 2 es utilizado y de igual manera el valor del pixel en la imagen secreto determina si se utilizara la permutación de la matriz base M_0 o la de M_1. Nota que los bloques Permutation 1 y Permutation 2 tienen cuatro salidas, cada una con 2 bits de resolución. Durante la evaluación de los pixeles de la imagen secreto, solo un bloque de permutaciones funciona y a la vez solo dos de sus salidas transmiten información a la sección de operaciones. Esto es porque una salida genera 2 pixeles para la sombra 1 mientras que la otra salida lo hace para la sombra 2.

Sección de operaciones. Esta sección está compuesta por dos bloques que ejecutan una operación OR utilizando las cuatro entradas correspondientes a cada uno de ellos (ver Figura 4-a). Las dos salidas de esta sección corresponden a las dos sombras.

El funcionamiento de la implementación propuesta para un (2 de 2)-esquema de criptografía visual, se puede describir como sigue: el FPGA recibe dos entradas de tipo digital, una correspondiente a la imagen secreta binaria y la otra a una señal binaria aleatoria, después, tres secciones dentro del FPGA se encargan de generar dos vectores binarios correspondientes a las sombras; por lo tanto, el FPGA tiene dos salidas binarias, una para cada vector.

Diagrama a bloques de la implementación propuesta

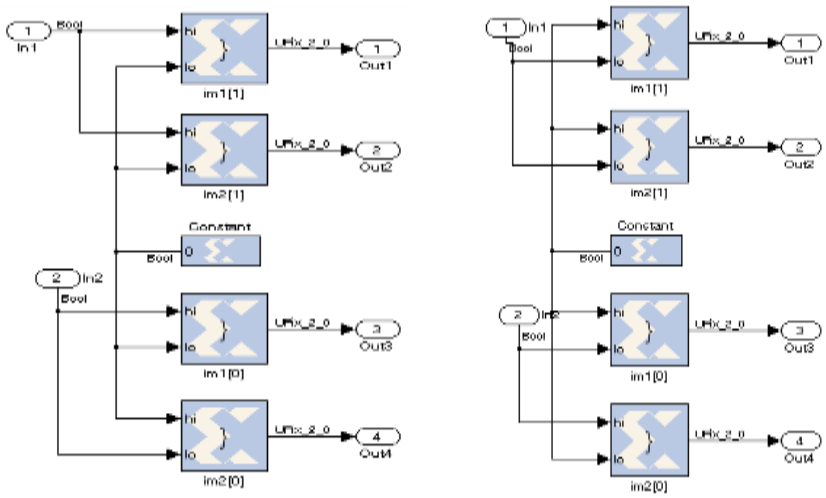
Sección de combinaciones



(a) (b)

Figura 4. Arquitectura del circuito integrado para implementar un (2 de 2)-esquema de criptografía visual. (a) Diagrama a bloques de la implementación propuesta. (b) Sección de combinaciones: multiplexor de 2 entradas a 4 salidas.

Sección de permutaciones



Permutation 1 Permutation 2

Figura 5. Bloques en la sección de permutaciones.

Revista de Ingeniería y Tecnologías para el Desarrollo Sustentable 3 (2017) 39-46

ISSN: 2448 - 7198

RESULTADOS Y DISCUSIÓN

Antes de entrar de lleno con la descripción de los resultados y su importancia, es necesario describir la maquetación utilizada para evaluar el desempeño de la implementación propuesta.

Se utilizaron 100 imágenes binarias con un numero de filas entre 256 y 720 pixeles y un numero de columnas entre 256 y 830 pixeles. De esas 100 imágenes, se decidió incluir 2 de ellas dentro de este documento: ver Figura 2-a y Figura 6-a. Las cien imágenes y la señal aleatoria In2 fueron enviadas al FPGA desde un programa en MATLAB a través de un puerto USB. Una vez que el FPGA cifraba cada una de las cien imágenes en un par de sombras, las sombras, se enviaban de regreso a MATLAB a través de otro puerto USB.

Durante la simulación, se registró el tiempo de cifrado de cada una de las imágenes y se comparó contra el tiempo obtenido por un programa desarrollado en MATLAB, el cual efectúa la misma operación. La Tabla 1, muestra los tiempos registrados tanto por nuestra implementación como por el programa en MATLAB para el cifrado de las dos imágenes que en este documento aparecen. La tabla 1 muestra que la implementación propuesta tiene una ganancia de un 350% sobre el programa desarrollado en MATLAB; en otras palabras, la implementación propuesta es 3.5 veces más rápida que el programa codificado en MATLAB (El programa en MATLAB “corre” sobre un procesador Intel Core i5 con una frecuencia de operación de 3.2~3.4 GHz y utilizando 4 GB de memoria RAM tipo DDR3 a una frecuencia de operación de 1333 MHz). Es importante notar que dada una imagen binaria, los tiempos registrados por la implementación propuesta son constantes, esto se debe a que la implementación no procesa otra cosa que no sea la imagen binaria, como es el caso de la computadora, donde varios procesos “corren” en segundo plano, por lo que los tiempos de procesamiento y de acceso a memoria varían dando como resultado tiempos variantes entre ensayos (o pruebas).

Aunque la aplicación de un (2 de 2)-esquema de criptografía visual es principalmente para cifrar imágenes que contienen algún tipo de texto o número, en la Figura 6, se demuestra que incluso la imagen de un rostro cifrado en un par de sombras se puede distinguir al llevar acabo la superposición de las sombras (operación OR entre las dos sombras).

Tabla 1. Tiempos de cifrado registrados por la implementación propuesta y un programa codificado en MATLAB. Los tiempos registrados se expresan en milisegundos (ms).

Ensayo/prueba	Logotipo del ITSOEH (332×480)		Lena (256×256)	
	Implementacion en FPGA (ms)	Programa en MATLAB (ms)	Implementacion en FPGA (ms)	Programa en MATLAB (ms)
1	3.187	11.155	1.310	4.898
2	3.187	11.713	1.310	4.445
3	3.187	10.987	1.310	4.516
4	3.187	11.200	1.310	4.624
5	3.187	10.793	1.310	4.516
Promedio (ms)	**3.187**	**11.160**	**1.310**	**4.590**
Ganancia $(\%)$	**350%**		**350%**	

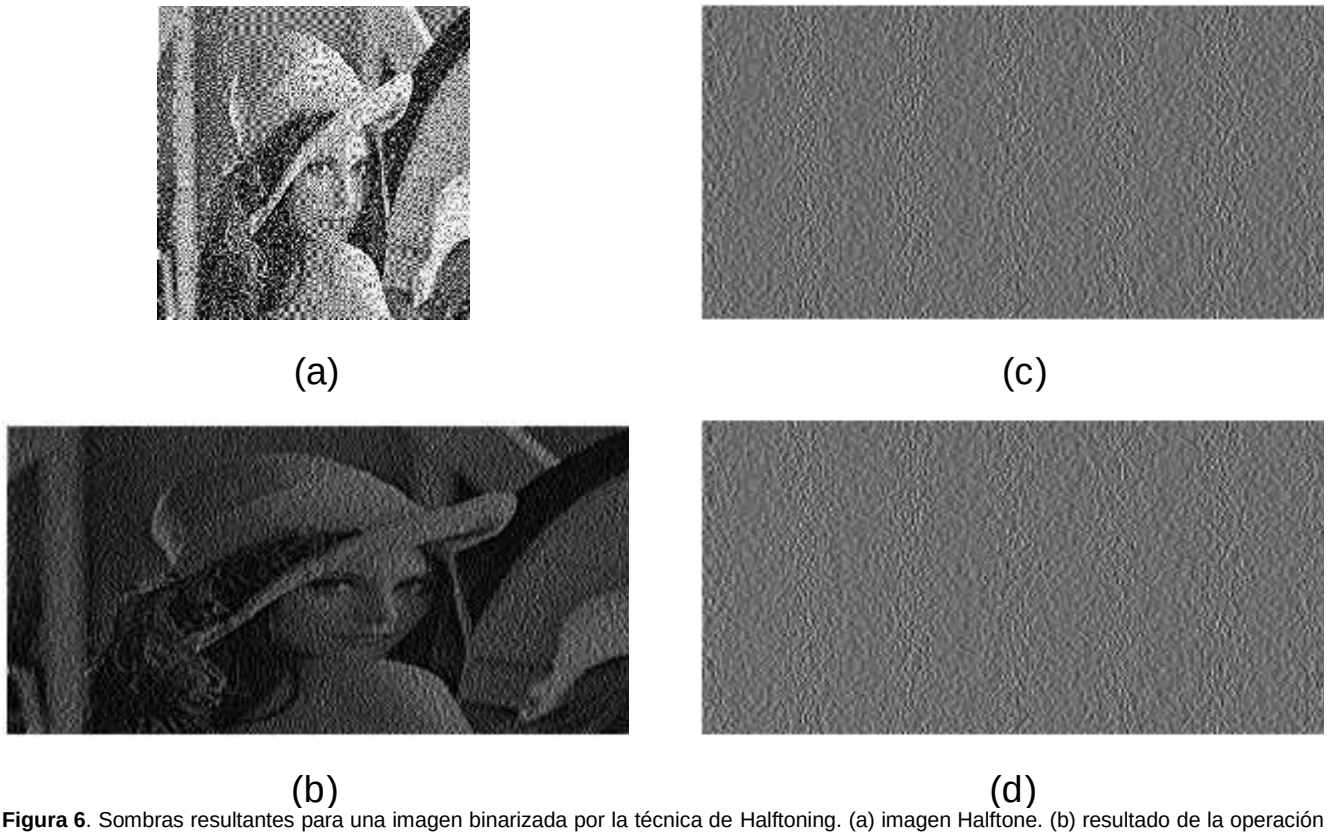


Figura 6. Sombras resultantes para una imagen binarizada por la técnica de Halftoning. (a) imagen Halftone. (b) resultado de la operación OR entre las sombras desplegadas en (c) y (d). (c) y (d) sombras resultantes.

CONCLUSIONES

Dentro de este documento se describe la implementación en un FPGA de un (2 de 2)-esquema de criptografía visual para aplicaciones de autenticaciones digitales tales como: autenticación de contenidos digitales[3], protección para los derechos de autor[4], comercio movil[5], transferencias bancarias en internet[6], entre otras. La implementación propuesta tiene dos ventajas sobre los programas de computadora que implementan el (2 de 2)-esquema de criptografía visual: incremento en la seguridad (debido al uso de hardware de propósito específico y no a uno de propósito general como lo es una computadora) y mayor velocidad de procesamiento durante el cifrado de la imagen secreto. Datos numéricos presentados dentro de este documento, muestran que la implementación propuesta puede ser 3.5 veces más rápida que su versión equivalente representada por un programa de computadora codificado en MATLAB.

AGRADECIMIENTOS

Los autores deseamos manifestar nuestro agradecimiento al Dr. Enrique Escamilla Hernández por haber proporcionado el FPGA y así poder llevar a cabo nuestras pruebas de manera física y no simulada. También deseamos agradecer al comité editorial del CONAINTE por asignar un espacio para la publicación de nuestra investigación.

REFERENCIAS

1. Naor M. and Shamir A. (1995). *Visual cryptography*. Advanced in Cryptology, EUROCRYPT'94, LNCS 950, pp. 1-12.
2. Shyu, S. J., et al. (2007). Sharing multiple secrets in visual cryptography. *Pattern Recognition*, vol. 40, no. 12, pp. 3633-3651.
3. Rao, Y. S., Sukonkina, Y., Bhagwati, C., & Singh, U. K. (2008). *Fingerprint based authentication application using visual cryptography methods (improved id card)*. In *TENCON 2008-2008 IEEE Region 10 Conference*, pp. 1-5.
4. Naor, M. and Pinkas, B. (1997). *Visual authentication and identification*. In Advances in Cryptology-CRYPTO'97: 17th Annual International Cryptology Conference, Santa Barbara, California, USA, pp. 322.

ISSN: 2448 - 7198

5. Chen, C. T., & Lu, T. C. (2004). *A mobile ticket validation by VSS tech with time-stamp*. 2004 IEEE International Conference on e-Commerce and e-Service, pp. 267-270.
6. Hegde, C., Manu, S., Shenoy, P. D., Venugopal, K. R., & Patnaik, L. M. (2008). *Secure authentication using image processing and visual cryptography for banking applications*. In Advanced Computing and Communications. ADCOM 2008. 16th International Conference on, pp. 65-72.