ISSN 2007-9737

Partial Image Encryption Using Cellular Automata

Marco Tulio Ramírez Torres[1], Marcela Mejía Carlos[2], José S. Murguía Ibarra[3], Luis Javier Ontañon García[1]

[1] Universidad Autónoma de San Luis Potosí, CARAO,
México

[2] Universidad Autónoma de San Luis Potosí, IICO,
México

[3] Universidad Autónoma de San Luis Potosí,
Facultad de Ciencias,
México

{tulio.torres, marcela.mejia, ondeleto, luis.ontanon}@uaslp.mx

Abstract. In this work is proposed a partial image encryption method based on the synchronization of the cellular automaton rule 90. The security analysis proves that this cryptosystem is resistant to different tests and attacks such as the Chosen/Knownplain image attack and bit-replacement. This algorithm is a variant from another encryption algorithm named ESCA, in this case we modified the ESCA system to a partial encryption version reducing the latency time up to a 50%. This version has two changes a)it only encrypts the three most significant bits, and b) the bits of the secret key are rotated instead of calculating a key for each block of plaintext. These changes allow to reduce the latency time and according to the tests do not compromise the security. This proposal could be a feasible and secure option for real-time applications.

Keywords. Partial encryption, digital images, cellular automata.

1 Introduction

Nowadays, many of our information is in digital formats and it is sent through networks or is stored in different devices. And, with technologies as Internet of things our personal data could be more and more exposed, therefore we need to learn about internet and its risks [14] and develop new security techniques. For this reason, it exists a great interest in the protection and manipulation of the data.

Due to the great advances in technology, each time is required to have better and more efficient algorithms for confidential and secure data handling. This information may vary depending on the application area and in many cases is necessary processing it in real time.

For example, pictures, medical images, diagrams, surveillance video, video conference, etc. could contain confidential information, and if these data are transmitted and are not encrypted the confidentiality of information is exposed in the links allowing access to third parties without being detected. To overcome the eavesdropping problem several encryption systems have been proposed, such as AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), RSA (Rivest, Shamir y Adleman) among others.

These systems are generally used in text and binary data, but they are not suitable for the encryption of multimedia content as digital images and audio files, due to their massive volumes, high adjacent correlation and sometimes the multimedia data require real-time interactions (displaying, bit rate conversion, etc.) [3]. That is why the image encryption is a particular studying area. The image encryption algorithms should provide two kinds of security: cryptographic security and perceptual security.

This means that the data can not be recovered without the correct key and also the information must be transformed in an unintelligible form. For example Fig. 1 shows an image encrypted with a scheme that not provide perceptual security. It is easy to guess what is the original image, even when we do not know the secret key.

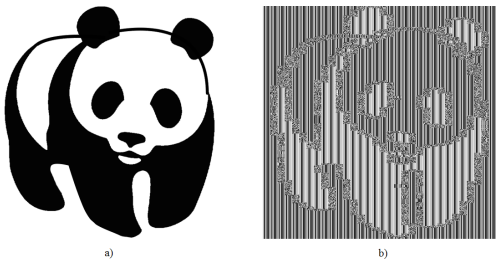

Fig. 1. (a) Original image. (b) Image encrypted in ECB mode

Hence, many encryption systems with different approaches have been developed for image encryption area [19, 17, 12]. In some cases, the algorithms provide perceptual security but are vulnerable to cryptanalysis attacks [16, 18, 10]. For this reason, it is crucial to validate the proposed algorithms.

On the other hand, it exists different techniques to try reduce computational cost due to cryptographic algorithms. There are proposes as compression-encryption algorithms [9], techniques to optimize calculations [6], partial encryption algorithms [1], among others. Partial encryption algorithms encrypt only a part of the information and the rest is not modified. This means that data blocks are partitioned, one part is encoded and the other keep unchanged and later are combined together.

The main aim of these algorithms is to increase the efficiency by reducing the data rate to encrypt. There are different classifications of these algorithms according to type of data, relation between encryption and compression, encryption operation and others. In this work, we focus on a partial image encryption for raw data. The encryption system considered in this implementation is based on the synchronization of the cellular automaton rule 90 [13].

This cryptosystem is named ESCA (for short) and it has been validated and implemented for image encryption in Ref. [7, 8]. The ESCA system can be considered secure for images encryption but latency time is too high to incorporate it in real-time applications. In this work we developed and validated a partial encryption version. The results show that this scheme could be an efficient solution to protect information with a low latency. The structure of this paper is organized as follows. In section 2 are defined one dimension cellular automata, later in section 3 are described ESCA system and its partial version, the results of the security analysis and comparison of latency are discussed in section 4. Finally, the conclusions are drawn in section 5.

2 Cellular Automata

The concept of cellular automata (CA) was introduced in the 1940's by the mathematicians John von Neumann and Stanislaw Ulam [15]. CA are used to model complex behavior of natural systems, where local interactions are involved. In fact, CA represent a class of dynamical systems that enable to describe the evolution of complex systems with simple rules, without using differential equations. Besides, they can be easily implemented in hardware. Cellular automata consist in an organized lattice of cells, where each cell has a finite number of states. The CA form a two-dimensional lattice whose cells evolve in discrete steps, according to a local update rule applied uniformly over all the cells. At the beginning, a state is assigned to the cells at time $t = 0$, where the new states of a cell will depend on its own previous state and states of its neighborhood, as is shown in Fig. 2.

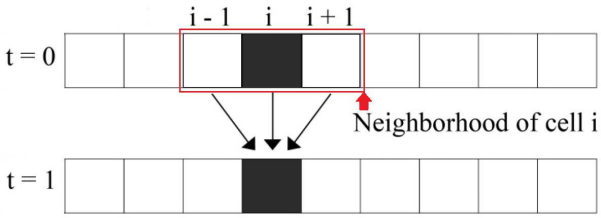

Fig. 2. Space-time in 1-D CA

A local rule is the algorithm to compute the next state of a cell, using its neighborhood. Generally, a neighborhood is enunciated by its radius, this is the range of cells on each side affecting the cell in time t. In Ref. [11], the local rule 90 is described by the formula $c_i(t+1) = c_{i-1}(t) + c_{i+1}(t)$ mod 2, this means that could be obtained by XOR-ing the extreme cells of the radius-1 neighborhood, and the cells have only two states 0 and 1. The name of the rule is obtained from the decimal equivalent of the binary resulting expressions of the 8 possible combinations in the neighborhood.

3 Partial Image Encryption

3.1 ESCA Encryption System

In this work is considered the encryption scheme used in [5], where the synchronization phenomenon of cellular automata has been applied to design two families of permutations Ψ and Φ for the encryption and decryption processes, and an asymptotically perfect pseudo-random number generator. This cryptosystem is flexible and reconfigurable for different bit-lengths. Fig. 3 illustrates a general block diagram of the encryption system ESCA.

The ESCA system comprises the sets M, C and K of binary words, M and C correspond to the plaintexts and ciphertexts respectively, the length of these words is $J = 2^j$, for $j = 1, 2, 3....$ Also it is possible to concatenate several blocks of these lengths.

The set K corresponds to the enciphering keys of length $N = 2^n - 1$ for $n = 1, 2, 3...$, for a complete encryption $n > j$ such that N must be larger than J. The two indexed families of permutations $\Psi = \{\psi_{\mathbf{k}} : \mathbf{k} \epsilon K\}$ and $\Phi = \{\phi_{\mathbf{k}} : \mathbf{k} \epsilon K\}$ are called encryption and decryption functions respectively. Basically, the cryptosystem transforms a plaintext sequence $\mathbf{m}$ into a ciphertext sequence $\mathbf{c}$, i.e. for every $\mathbf{k} \epsilon K$ one has $\mathbf{c} = \psi_{\mathbf{k}}(\mathbf{m})$, whereas to disclose from the sequence of cipher-blocks, one uses the decryption function $\mathbf{m} = \phi_{\mathbf{k}}(\psi_{\mathbf{k}}(\mathbf{m}))$. Since the complete encryption scheme is a symmetric algorithm, the encryption and decryption processes use the same enciphering key $\mathbf{k}$.

To calculate the enciphering keys $\mathbf{k}$ is necessary a pseudo-random number generator, in Ref. [4] the authors present an ergodic and mixing transformation of binary sequences in terms of a cellular automaton, which is the main element of a pseudo-random number generator (PRNG). The PRNG in its basic form, follows the algorithm shown in Fig. 4. At first, the key generator requires two seeds, $\mathbf{x} = \mathbf{x}_0^{k+1}$, of N bits, and $\mathbf{y} = \mathbf{x}_0^k$, of $(N+1)$ bits, which are the input of function $\mathbf{k} = h(\mathbf{x}, \mathbf{y})$.

The seeds are $\mathbf{x} = \{x_1, x_2, x_3, \ldots, x_N\}$ and $\mathbf{y} = \{y_1, y_2, y_3, \ldots, y_{N+1}\}$, and the first number generated of N bits is the sequence output of function h, $\mathbf{k} = \mathbf{x}_0^{k+2}$ of N bits. Now this sequence is feeding back to the input, which becomes the next value of $\mathbf{x}$, and the previous value of $\mathbf{x}$ becomes the initial bits of the new $\mathbf{y}$, where the missing bit is the least significant bit (LSB) of the previous $\mathbf{y}$, which becomes the most significant bit (MSB) of this sequence, and the same procedure is iterated repeatedly.

In Ref. [7], was added a pre-processing to make this scheme resistant to Chosen/Known-plaintext attacks. This function works like a dynamic substitution, the process is similar to PRNG, where the function $\widehat{\mathbf{m}} = h(\mathbf{m}, \mathbf{z})$ is used to transform the blocks $\mathbf{m}$ into an unintelligible form denominated $\widehat{\mathbf{m}}$.

Fig. 5 shows a block diagram of this process, the inputs are a block $\mathbf{m}$ of J bits and a seed $\mathbf{z}$ of $J+1$ bits. At the output, a block $\widehat{\mathbf{m}}$ of J bits is obtained, which will be encrypted later with the permutation Ψ, $\mathbf{c} = \psi_{\mathbf{k}}(\widehat{\mathbf{m}})$.

Also in Fig. 5 is shown the feedback, this is different from the key generation, the next $\mathbf{z}$ is obtained from the joint of $\widehat{\mathbf{m}}$ and the LSB of previous $\mathbf{z}$ as the MSB. This change allows to process images with a high adjacent correlation.

The encryption of an image with the ESCA system proceeds as follows:

1. Load the plain-image $\mathbf{I}$ of size $V \times U$.
2. By scanning the image $\mathbf{I}$ row by row, arrange its respective pixels as a sequence or a vector, and convert each pixel value to their corresponding binary value.

ISSN 2007-9737

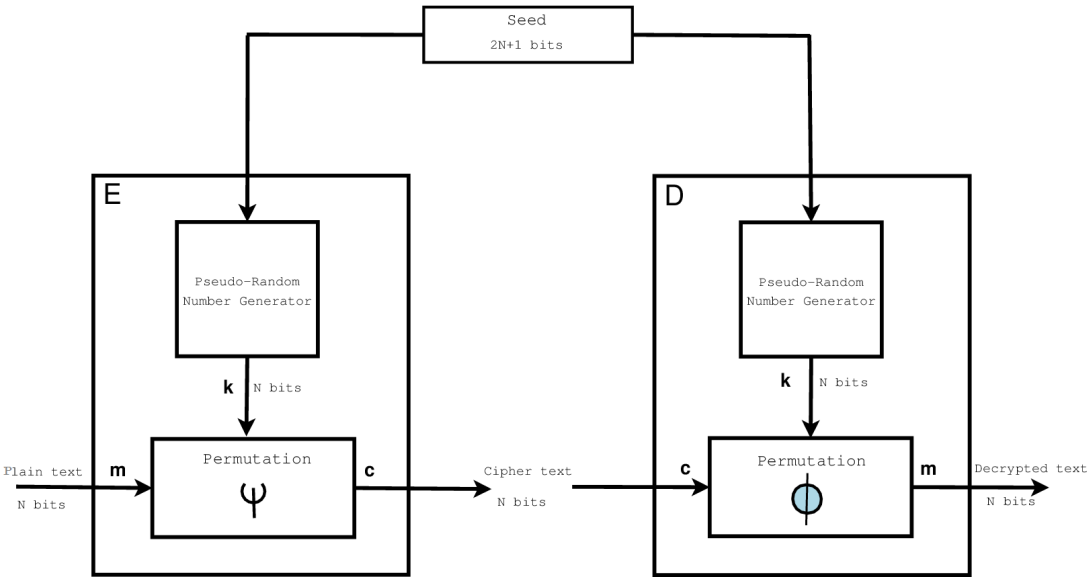

Fig. 3. The encryption scheme ESCA with its main components: the indexed families of permutations and the pseudorandom generator keys

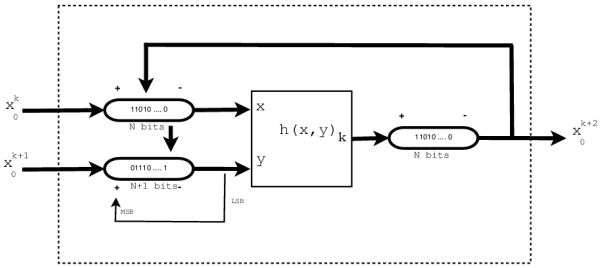

Fig. 4. Basic form of the pseudo-random number generator

3. Establish the length of the encryption key, it must be larger than plaintext bit-length.
4. Compute the modified plaintext sequence $\widehat{\mathbf{m}}$ using the pre-processing function.
5. Encrypt each modified block, $\mathbf{c} = \psi_{\mathbf{k}}(\widehat{\mathbf{m}})$ with a different $\mathbf{k}$ each one.
6. By reshaping the set of ciphered sequences of the previous step into an $V \times U$ image, obtain the ciphered image.

Fig. 6 illustrates preprocessed and encrypted images, using the ESCA system.

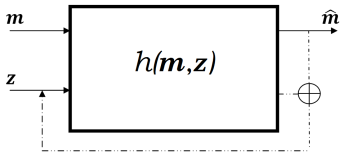

Fig. 5. Pre-processing to obtain $\widehat{\mathbf{m}}$

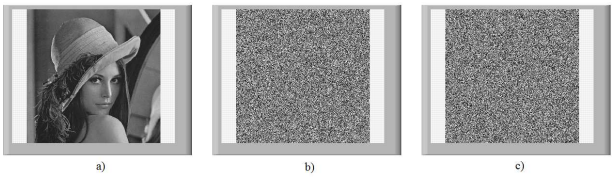

Fig. 6. a) Original image, b) preprocessing image and c) encrypted image.

3.2 Partial Encryption Version

For the ESCA system we developed a partial encryption version for images, due to high latency that it presents in the complete encryption version. This version focus on two aspects: first, the pre-processing function makes the ESCA system resistant to Chosen-plain Image Attack, but the

initial condition is too short against brute force attacks. And second, the process with the highest latency is the PRNG, hence, a way to reduce processing time could be encrypt only certain bit-plains, therefore, the secret keys use only a few bits and could be reused several times, giving a rotation to its bits.

The perceptual security is not endangered because the pre-processing function output is a mixed image with an uniform histogram (as we will see in section 3). Therefore, in this investigation we encrypt 8-bit grayscale images using secret keys of 31 bits. Each pixel coefficient conforms a block **m**, that is a boolean vector of 8 bits, $\mathbf{m} = \{m_1, m_2, m_3, \ldots, m_8\}$. From each block **m** a block $\widehat{\mathbf{m}}$ is calculated using the preprocessing function $\widehat{\mathbf{m}} = h(\mathbf{m}, \mathbf{z})$, and also is a vector of 8 bits, $\widehat{\mathbf{m}} = \{\widehat{m_1}, \widehat{m_2}, \widehat{m_3}, \ldots, \widehat{m_8}\}$. Finally, the secret key **k** of 31 bits, $\mathbf{k} = \{k_1, k_2, k_3, \ldots, k_{31}\}$, is used to encrypt the block $\widehat{\mathbf{m}}$ to obtain the corresponding block **c**, in the full encryption version this process is described by the next enciphering equations:

$$\begin{aligned}
c_1 &= \widehat{m_1} \oplus k_1 \oplus k_{17} \oplus k_{25} \oplus k_{29} \oplus k_{31};\\
c_2 &= \widehat{m_2} \oplus k_2 \oplus k_{18} \oplus k_{26} \oplus k_{30};\\
c_3 &= \widehat{m_1} \oplus \widehat{m_3} \oplus k_3 \oplus k_{19} \oplus k_{27} \oplus k_{31};\\
c_4 &= \widehat{m_4} \oplus k_4 \oplus k_{20} \oplus k_{28};\\
c_5 &= \widehat{m_3} \oplus \widehat{m_5} \oplus k_5 \oplus k_{21} \oplus k_{29};\\
c_6 &= \widehat{m_2} \oplus \widehat{m_6} \oplus k_6 \oplus k_{22} \oplus k_{30} :\\
c_7 &= \widehat{m_1} \oplus \widehat{m_3} \oplus \widehat{m_5} \oplus \widehat{m_7} \oplus k_7 \oplus k_{23} \oplus k_{31};\\
c_8 &= \widehat{m_8} \oplus k_8 \oplus k_{24};
\end{aligned} \tag{1}$$

where c_i, $\widehat{m_i}$ and k_i, with $i = 1, 2, ..., 31$, correspond to the i bit of the ciphertext, modified plaintext and secret key, respectively. Hence, this partial encryption is obtained encrypting the three most significant bits of the modified plaintext $\widehat{m_6}, \widehat{m_7}$ and $\widehat{m_8}$. The secret keys that we consider are at least 31 bits length, where the initial condition could be of 136 bits [8]. If the system encrypts the three most significant bits of the modified plaintext to obtain c_6, c_7 and c_8, only eight specific bits are required from the secret key: $k_6, k_7, k_8, k_{22}, k_{23}, k_{24}, k_{30}$ and k_{31}, as is shwon in eqs. 1.

Looking this aspect, the bits of secret key are rotated one position and the LSB become the MSB, and it is used again when another block of plaintext is encrypted, the enciphering equations use eight different bits of the same secret key. This process is repeated 14 times for each key. Fig. 7 shows how we rotate the bits of the secret key.

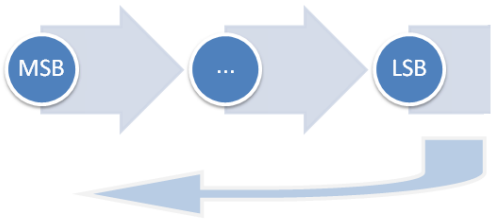

Fig. 7. Bit rotation of secret key

Finally, with this version the encryption system calculate only one secret key for 14 $\widehat{\mathbf{m}}$ blocks, and reduce the enciphering equations from 8 to 3. In the next section is shown the security analysis and a comparison of the latency time using the partial encryption version.

4 Results

4.1 Security Analysis

To measure the quality and the robustness of the partial encryption ESCA system, it has been evaluated with some common statistical tests and attacks. This is done by testing the distribution of pixels of the ciphered-images, studying the correlation between the plain and ciphered images, the chosen-plain image attack, and bit-replacement attack.

4.1.1 Histogram Analysis

An image histogram shows how pixels in an image are distributed by plotting the number of pixels at each color intensity level. If the histogram of an encrypted image (or ciphered-image) has an uniform distribution, then the cipher is able to hide the redundancy of original image. We calculate the histograms for three different 256 gray-level images, the Lena, peppers and mandrill images. All of them have dimensions of 512×512 pixels, and we consider these images because they are widely used as standard test images in the field of image processing. The histograms for the

grayscale Lena test image (plain-image) and its encrypted version are shown in Fig. 8, one can see that the histogram of the ciphered images is uniformly distributed and significantly different from the respective histograms of the plain-images.

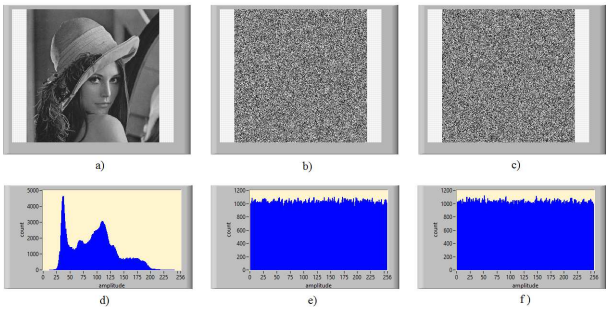

Fig. 8. Histogram analysis for the gray-scale Lena test image. Top row: (a) The plain-image, (b) the preprocessed-image and (c)the ciphered-image. Bottom row: (d) The histogram of (a), (e) the histogram of (b) and (f) the histogram of (c)

4.1.2 Correlation

To show that the ciphered-image is independent from the plain-image, we calculate the correlation coefficient between both images. If the coefficient is close to 0, it suggests that there is no linear correlation or a weak linear correlation. The correlation coefficients for the three grayscale test images are showing in the Table 1, as we can see there is no correlation between the original image and encrypted image.

Table 1. Correlation coefficient between plain-images and their corresponding recovered images

Images	Correlation coefficient
Lena	0.0037
Peppers	0.0031
Mandrill	0.0004

4.1.3 Chosen-Plain Image Attack

Although the ESCA system seems to be secure against the previous statistical attacks, we analyse it with the Chosen-plain Image Attack (CPIA). As is pointed out in Ref. [2], if the cryptosystem is secure against the chosen-plain image attack, it is also secure against other cryptanalytic attacks such as cipher image only attack or known-plain image attack. In this case, the attacker is able to choose the plain images and obtain the corresponding encrypted images, but does not know the decryption key. We perform this attack as follows: (a) We look for a mask image $\mathbf{I}_M$, which is obtained by applying the bitwise XOR operation to the chosen plain-image with its corresponding ciphered image pixel by pixel.

A "good" mask can be obtained if the pixels of the chosen plain-image $\mathbf{m}$ are all fixed to be the same gray value, e.g. $\mathbf{m} = (0, 0, ..., 0, 0)$. (b) Once the mask image $\mathbf{I}_M$ is obtained, we choose another image $\mathbf{I}_O$ and get the corresponding ciphered image $\mathbf{I}_C$. (c) Next, we try to recover the plain-image $\mathbf{I}_O$ by XOR-ing the mask image $\mathbf{I}_M$ and the cipher image $\mathbf{I}_C$ pixel by pixel. In case that the obtained image in step (c) conveys significant information of the image $\mathbf{I}_O$, thus the encryption system is insecure, otherwise the proposed encryption scheme resist the attack. In Fig. 9 is shown an example of this attack.

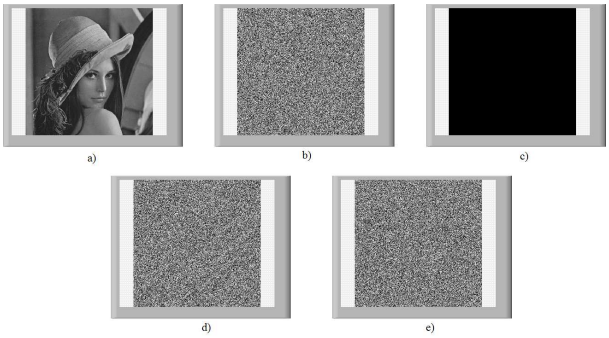

Fig. 9. Top row: (a) The plain-image. (b) The partial encrypted image of (a). (c) The second chosen image. Bottom row: (d) The mask image of (c). (e) The recovered image

The grayscale Lena is the plain-image $\mathbf{I}_O$ (Fig. 9(a)), and its respective ciphered version is shown in Fig. 9(b). The chosen-image used was an image with pixels of value zero, $\mathbf{m} = (0, 0, ..., 0, 0)$, see Fig. 9(c), and the mask image $\mathbf{I}_M$ is illustrated

in Fig. 9(d). The recovered plain-image is shown in Fig. 9(e), and a visual inspection reveals the effectiveness of our proposed encryption scheme. Similar results were obtained for the test images considered in this work. Also we calculated the correlation coefficient between the recovered image and the original image, to ensure that they are independent of each other. The results are in Table 2.

Table 2. Correlation coefficient between plain-images and their corresponding recovered images

Images	Correlation coefficient
Lena	0.0031
Peppers	0.0004
Mandrill	0.0016

4.1.4 Bit-Replacement

This attack is for partial encryption algorithms, it tries to recover the original image replacing the encrypted bits by a constant, usually zero. The luminance is compensated adding constants depending the number of bits and their position. Fig. 10 shows the results of this attack replacing the three encrypted bits, and the Table 3 shows their corresponding correlation coefficient for the three images under this attack.

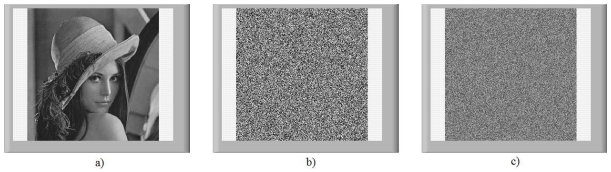

Fig. 10. (a) Original image. (b) Encrypted image. (c) Recovered image

4.2 Comparative Analysis

Once that the security of this scheme was guaranteed, we calculated the execution time of the complete and partial versions to make a comparative. The Table 4 shows the results of encrypt an image of 128×128 pixels.

Table 3. Correlation coefficient between plain-images and their corresponding recovered images

Images	Correlation coefficient
Lena	0.0011
Peppers	0.0021
Mandrill	0.0017

As the images have the same resolution the result is the same for the three test images. The computer used in this test has a quad-core processor and 8 GB of RAM.

Table 4. Correlation coefficient between plain-images and their corresponding recovered images

Algorithm	Execution time (ms)
Complete encryption	31
Partial encryption	15

As we can see, the latency is reduced more than half. And this time is adequate to real-time video, with 30 frames per second.

5 Conclusion and Future Work

In this work we can confirm that this partial version could be a secure option to encrypt images. The latency time was reduced, and in a low resolution could be an option to real-time applications.

In future work, we will try to combine this partial encryption system with a compression scheme to get better results and applications. The compression procedure that is considered is based on an energy criterion of the Haar wavelet coefficients, because it has a low degradation and a simple implementation. Currently, the patent of ESCA system is pending, this variation also could be patented as an improvement.

Acknowledgements

This work has been done with financial support through project PRODEP DSA/103.5/16/10419 and UASLP.

References

1. **Belazi, A., El-Latif, A. A. A., Diaconu, A.-V., Rhouma, R., & Belghith, S. (2017).** Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. *Optics and Lasers in Engineering*, Vol. 88, pp. 37–50.
2. **del Rey, A., Sánchez, G., & De La Villa Cuenca, A. (2012).** Encrypting digital images using cellular automata. *Hybrid Artificial Intelligent Systems*, pp. 78–88.
3. **Lian, S. (2008).** *Multimedia content encryption: techniques and applications*. CRC press.
4. **Mejía, M. & Urıas, J. (2001).** An asymptotically perfect pseudorandom generator. *Discrete and Continuous Dynamical Sys*, Vol. 7, pp. 115–126.
5. **Murguía, J., Flores-Erana, G., Carlos, M. M., & Rosu, H. (2012).** Matrix approach of an encryption system based on cellular automata and its numerical implementation. *International Journal of Modern Physics C*, Vol. 23, No. 11, pp. 1250078.
6. **Olguin Carbajal, M., Herrera-Lozada, J. C., Rivera-Zárate, I., Serrano-Talamantes, J. F., Cadena-Martínez, R., & Vásquez-Gómez, J. I. (2018).** Minimum addition chains generation using evolutionary strategies. *Computación y Sistemas*, Vol. 22, No. 4.
7. **Ramírez-Torres, M., Murguía, J., & Carlos, M. M. (2014).** Image encryption with an improved cryptosystem based on a matrix approach. *International Journal of Modern Physics C*, Vol. 25, No. 10, pp. 1450054.
8. **Ramírez-Torres, M., Murguía, J., & Mejía-Carlos, M. (2014).** Fpga implementation of a reconfigurable image encryption system. *ReConFigurable Computing and FPGAs (ReConFig), 2014 International Conference on*, IEEE, pp. 1–4.
9. **Ramírez-Torres, M., Murguía, J., Mejía-Carlos, M., & Aboytes-González, J. (2015).** A secure compression scheme for real-time applications using 2d-wt and cellular automata. *Research in Computing Science*, Vol. 104, pp. 103–114.
10. **Rhouma, R. & Belghith, S. (2008).** Cryptanalysis of a spatiotemporal chaotic image/video cryptosystem. *Physics Letters A*, Vol. 372, No. 36, pp. 5790–5794.
11. **Schiff, J. L. (2011).** *Cellular automata: a discrete view of the world*, volume 45. John Wiley & Sons.
12. **Suresh, V. & Madhavan, C. V. (2012).** Image encryption with space-filling curves. *Defence Science Journal*, Vol. 62, No. 1, pp. 46–50.
13. **Urías, J., Salazar, G., & Ugalde, E. (1998).** Synchronization of cellular automaton pairs. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 8, No. 4, pp. 814–818.
14. **Vallejo, M. G., Muñoz, G. E., & Hernando Rosales, J. (2018).** Kids and parents privacy exposure in the internet of things: How to protect personal information? *Computación y Sistemas*, Vol. 22, No. 4.
15. **Von Neumann, J. & Burks, A. W. (1996).** *Theory of self-reproducing automata*. University of Illinois Press Urbana.
16. **Wang, H., Xiao, D., Chen, X., & Huang, H. (2018).** Cryptanalysis and enhancements of image encryption using combination of the 1d chaotic map. *Signal Processing*, Vol. 144, pp. 444–452.
17. **Wang, X., Qin, X., & Liu, C. (2018).** Color image encryption algorithm based on customized globally coupled map lattices. *Multimedia Tools and Applications*, pp. 1–19.
18. **Wu, J., Liao, X., & Yang, B. (2018).** Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation. *Signal Processing*, Vol. 142, pp. 292–300.
19. **Yu, C., Li, J., Li, X., Ren, X., & Gupta, B. (2018).** Four-image encryption scheme based on quaternion fresnel transform, chaos and computer generated hologram. *Multimedia Tools and Applications*, Vol. 77, No. 4, pp. 4585–4608.

Article received on 05/04/2018; accepted on 04/01/2019.
Corresponding author is Marco Tulio Ramírez Torres.