Efficient Image Encryption and Authentication Scheme Based on Chaotic Sequences

M. Farajallah, Z. Fawaz, S. El Assad, Member IEEE
IETR / LUNAM University/LI University
Nantes, France
safwan.elassad@univ-nantes.fr
mousa.farajallah@etu.univ-nantes.fr

O. Deforges
IETR / INSA Rennes
Rennes, France
olivier.deforges@insa-rennes.fr

***Abstract*— Many image encryption algorithms based on chaos have been proposed since 1989. Most of them are slow and use a secret key of encryption/decryption independent of the plain image. We introduce a new fast and secure image encryption and authentication scheme based on the chaotic sequences. The main structure of the proposed algorithm consists of two layers (substitution and permutation) for encryption and decryption image values, and two components: a hash function and a chaotic generator. The hash function generate the secret key of the chaotic generator that provides the dynamic keys for the substitution-permutation layers, while the secret hash key is used to authenticate the decrypted image. The proposed algorithm is at least ten times faster than the AES (Advanced Encryption Standard) algorithm and faster than many chaos-based encryption algorithms of the literature. Furthermore, it is very secure against chosen/known plaintext and statistical attacks because the key of the chaotic generator is dependent on the plain-image. Simulation results show that the efficient performance is reached in only one round.**

Keywords- Chaos-based cryptosystem; Image encryption/authentication algorithm; Chaoti cgenerator; security analysis.

I. INTRODUCTION

Chaos based encryption algorithms are extremely sensitive to the initial conditions and system parameters [1][2][3]. This helps in producing the required confusion and the diffusion effects. Also, normally they are faster than the standard encryption algorithms because of the low complexity of their structures [4]. For that and during the last decade, a number of chaos-based encryption algorithms have been proposed [5][6]. Guanrong et al, introduced a symmetric image encryption scheme based on 3D chaotic maps, he generalized the 2D chaotic maps into 3D chaotic maps to shuffle the pixel positions and to confuse the relationship between the plain and the cipher images. The speed of this algorithm is not high for real time applications, and also the propagation error is large, since it works in cubes and each cube affects all other [6]. In his paper, Fridrich proposed a chaos-based encryption algorithm consisting of a permutation layer based on the Baker Map, following by a nonlinear feedback process based on a nonlinear feedback register. It is clear that Fridrich algorithm is time consuming and slow encryption scheme [7]. Song et al, presented a new image encryption scheme based on a new spatiotemporal chaos. From the security result was presented, it is clear that this algorithm has good security level, but it has also slow time of encryption/decryption related to the sort operation, which is well known that is a time consuming operation [8]. In [9], a fast and robust encryption algorithm for images has been proposed. The algorithm makes r rounds of an SP-network (Substitution-Permutation network) on each pixel using two PWLCM (Piecewise Linear Chaotic Map) maps. This algorithm is faster than previous algorithms but one of the weaknesses is the high error propagation caused by the used technique of perturbation. The paper is organized as follows, the next section describes the main structure of the proposed algorithm, and in its subsections the details of the cryptosystem components are described. Section 3 presents the security and time analysis results, while the conclusion part is at the last section.

II. GENERAL STRUCTURE OF THE PROPOSED CRYPTOSUSTEM

The general structure of the proposed cryptosystem is presented in Figure 1 for the encryption and in Figure 2 for the decryption. The encryption process consists of two layers. The first layer is the substitution one achieved by the skew tent map, followed by a permutation layer realized by the 2D cat map. The both layers need a dynamic key during the encryption and decryption processes, these keys are provided from a simplified version of the chaotic generator from El Assad et al [10]. The key of the chaotic generator is derived from a hash function (we use SHA-1 (Secure Hash Algorithm) for testing and we are working on SHA-256) [11], whose inputs are the secret hash key and the plain image. So, the confusion-diffusion properties of the cryptosystem are reached and the immunity against known-chosen plaintext is obtained.

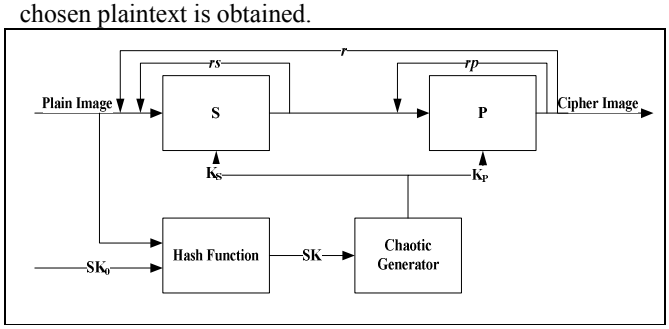


Figure 1. Encryption Parts of the Proposed Cryptosystem

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-298-1

The decryption process (see Figure 2), is based on an inverse permutation layer achieved by the same 2D cat map, following by the inverse substitution achieved by the inverse skew tent map. During this decryption process the rounds of each layer are starting in reverse order and also the dynamic keys are using in reverse order. Notice that, from the estimated plain-image and the hash secret key we can determine whether the estimated plain-image (the decrypted image) is the same one sent.

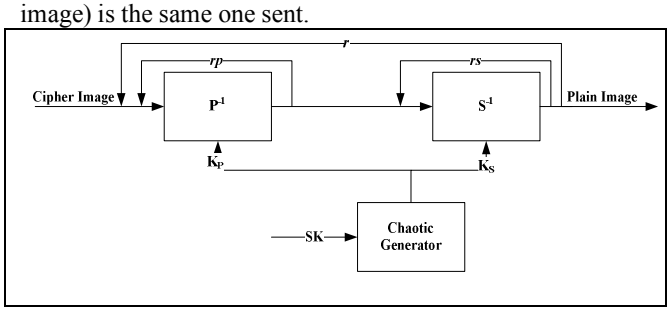


Figure 2. Decryption Parts of the Proposed Cryptosystem

The encryption steps are described in details in Figure 3. The plain image is divided into *NB* blocks, each one contains *BS* bytes. The first step is to generate the dynamical keys for the substitution-permutation layers. To do this, the plain image and the secret hash key are used as input for the hash function, and then the hash function produces the secret key of the chaotic generator, that provides the dynamic keys at each round of the cryptosystem. After that, for each plain block, first we use the CBC (cipher-block chaining) mode [12] (bit-wise XOR operation between the current plain block and the previous ciphered one, while in the first block the previous ciphered block is the initial random block IV (initialization vector)) and then we apply the substitution layer *rs* times and the permutation layer for rp times. Finally, these processes (CBC, substitution, permutation) are repeated *r* rounds and for each round a new dynamic keys are generated from the chaotic generator to be used in both layers, and so on.

Figure 4 shows the decryption part of the proposed cryptosystem. The decryption process is similar to the encryption one; the differences are in the inverse substitution and the reverse permutation layers, first of all, the dynamic keys are used in reverse order in both layers. Second, the permutation layer is starting the recover process from the last byte of the current block until the first one. Third, all counters will be starting in reverse order. Finally, the CBC mode function is used at the end of decryption of each block. The inverse substitution layer is starting as normal from the first to the last byte. The receiver uses the decrypted image to test the authentication source, and to test the integrity of the message.

The decryption and authentication processes suppose that the secret hash key and the secret key of the chaotic generator are known (transmitted in secret manner by the sender to the receiver.

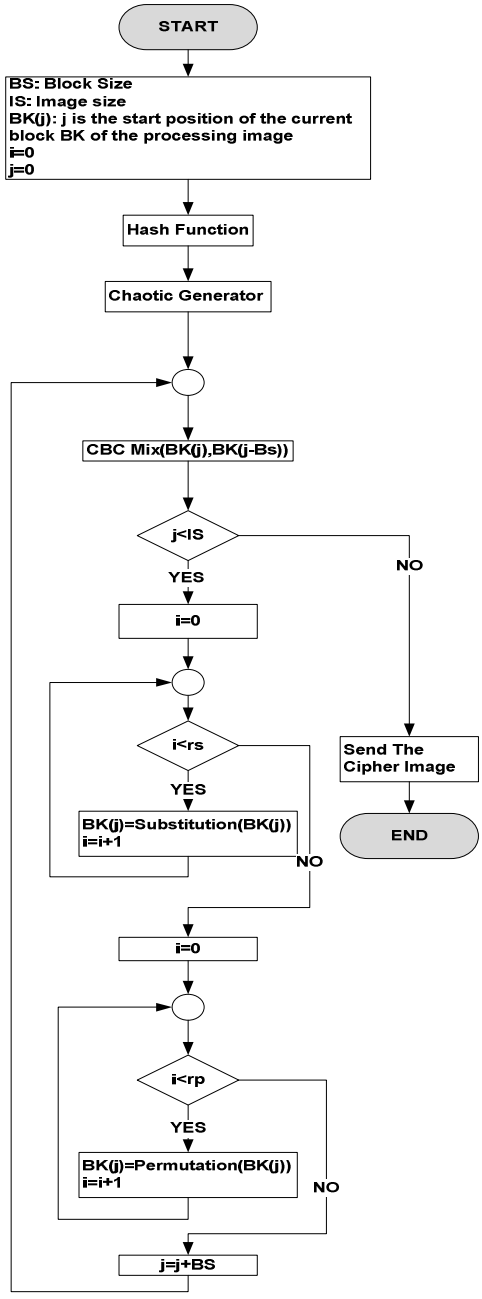


Figure 3. Encryption Components of the Cryptosystem

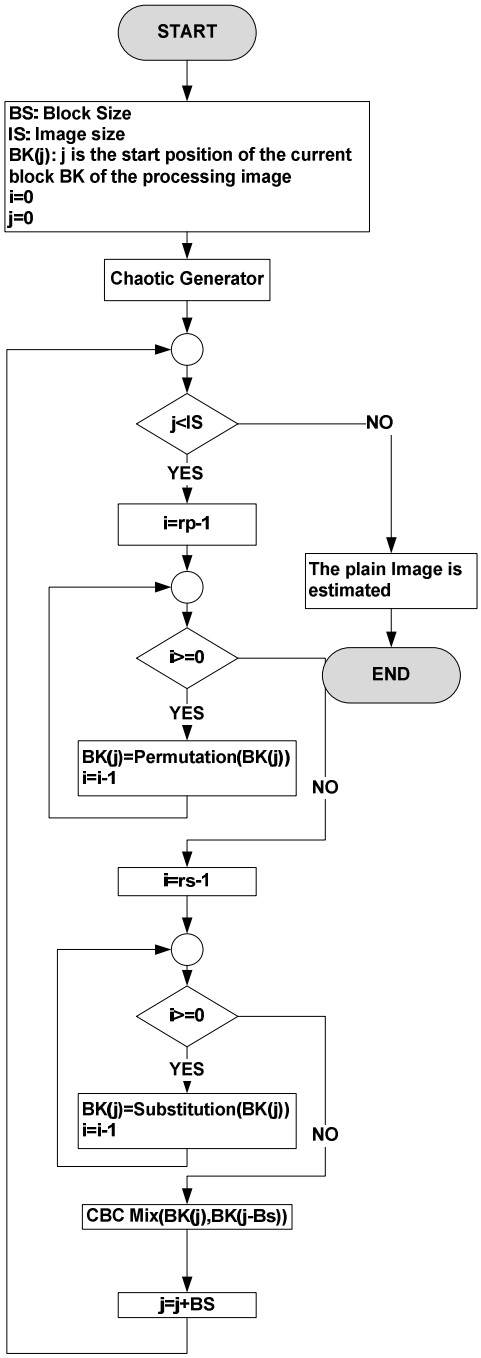


Figure 4. Decryption Components of the Cryptosystem

A. *Substitution Layer*

The substitution layer of this cryptosystem is based on the FSTM (Finite State Skew Tent Map). This layer is making some nonlinear transformation in the image data; the nonlinear step implies the cryptosystem to be more resistance against differential cryptanalysis attacks. This layer works on byte-by-byte transformation and it changes the value of the byte depending on a chaotic parameter a , which comes from a robust chaotic generator. The mathematical model of the encrypted part of FSTM is [5].

$$Y = S_a(X) = \begin{cases} \left\lceil \frac{Q}{a} \times X \right\rceil & ,0 \le X \le a \\ \left\lfloor \frac{Q}{Q-a} \times (Q-X) \right\rfloor + 1 & ,a \prec X \prec Q \end{cases} \quad (1)$$

Since FSTM function is a bijective one, it means that the FSTM function is invertible one, the inverse equations at the decryption part are:

$$X = S_a^{-1}(Y) = \begin{cases} \xi_1 & ,\theta(Y) = Y and \frac{\xi_1}{a} \succ \frac{Q-\xi_2}{Q-a} \\ \xi_2 & ,\theta(Y) = Y and \frac{\xi_1}{a} \le \frac{Q-\xi_2}{Q-a} \\ \xi_1 & ,\theta(Y) = Y+1 \end{cases} \quad (2)$$

Where

$$\xi_1 = \left\lfloor \frac{a}{Q} \times Y \right\rfloor \quad (3)$$

$$\xi_2 = \left\lceil \left(\frac{a}{Q} - 1 \right) \times Y + Q \right\rceil \quad (4)$$

$$\xi_3 = \left\lceil \frac{a}{Q} \times Y \right\rceil \quad (5)$$

$$\theta(Y) = Y + \xi_1 - \xi_3 + 1 \quad (6)$$

The structure of the dynamic key during the substitution process is:

$$K_s = \left[K_{s_0} \| K_{s_1} \| K_{s_2} \| K_{s_3} \| .. K_{s_{r-1}} \right] \quad (7)$$

$$K_{s_j} = a_j \quad (8)$$

Where r is the number of rounds of each block, and inside each round the substitution layer is repeated *rs* times.

Q=substitution processing unit-1

We chose the substitution processing unit to be 256 bits

$1 \le a_j \le Q \qquad j = 0,1,2,..r-1$

The total number of blocks in the plain image is $NB = L \times C \times P / Block\ size$, here L, C and P are the number of lines, the number of columns and the number of plains. The $Block\ size$ variable is used to save the size of the block in bytes (here 256 bytes).

The simplified version of the El Assad et al chaotic generator produces a 32-bit samples each calling time, we chose to set the substitution layer key to be 8-bit (image data is represent in one byte). Each sample of the chaotic generator will produce 32 bits, so we have two options:

1- Take the first 8 bits from each sample to be used as the dynamic key and skip the remaining 24 bits, if the first 8 bits are zero, then we will take the next 8 bits and so on.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-298-1

2- Divide the 32 bits into 4 bytes and make xor operation between the 4 bytes to produce the required dynamic key.

We chose to work according to the first option, since the dynamic key must be greater than zero and it is the best case for this condition in terms of the time and the probability of getting zero output.

Since (1) and (2) are time consuming and the range values of the input, the output and the dynamic keys of those equations is limited in [0-255], we use them to produce a lookup tables in order to reduce the execution time of the substitution and inverse substitution operations. For comparison, the both versions (substitution, inverse substitution by equations and by lookup tables) are implemented.

B. Permutation Layer

The permutation process changes the pixel position on the block under the test without changing its value. In our proposed cryptosystem, the permutation process is based on the modified version of the basic 2D Cat Map [8]:

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = Mod\left(\begin{bmatrix} 1 & u \\ v & 1+u\times v \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} r_i + r_j \\ r_j \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right) \quad (9)$$

Where (i_n, j_n) are the new row and column position of the old (i, j) byte position inside the *block* of size $M \times M$ bytes after applying the 2D cat map. u , v , r_i and r_j are the system parameters in the range of $[0, M-1]$. The last two parameters are added to the basic model to overcome the fixed point problem of the basic 2D cat map model.

The structure of the dynamic keys during the permutation process is:

$$K_p = \left\lfloor K_{p_0} \| K_{p_1} \| K_{p_2} \| K_{p_3} \| .. K_{p_{r-1}} \right\rfloor \quad (10)$$

$$K_{p_j} = \left[u_j \parallel v_j \parallel rl_j \parallel rc_j \right] \quad (11)$$

It is clear from (9) that the determinant of the Jacobean matrix of this model is 1, and so the 2D cat map process is a bijective. From the modulo operation we can conclude that the 2D cat map function is non-invertible one, but it is reversible, by applying the permutation loop in reverse order, that means beginning of the decryption process from the last byte inside the encrypted block, so the dynamic keys are used also in reverse order.

The 2D cat map is one-to-one function (bijective), which means every point of the square matrix can be transferred to exactly one unique point, from this fact we can replace the swap operation of (9) by the copy operation during the permutation layer to reduce the execution time.

The required dynamic keys bits for the permutation layer are giving by the following equation:

$$q = \log(M) \quad (12)$$

Where q is the number of required bits for each sub dynamic key, so, we need for each block $4 \times q \times r$ bits from the chaotic generator. During the permutation layer it is not necessary to have all sub-keys greater than zero, at least one of them is sufficient to be different of zero, because the probability to have the four sub-keys equal to zero is zero.

III. Time and Security Analysis

Time and security analysis is the most important part of any chaos based cryptosystem. Performance analysis of the proposed cryptosystem is provided in the first subsection, while in the next subsections we present the experimental results of testing different attacks.

A. Time Analysis

We applied time test for our proposed algorithm based on both versions (with/without lookup table) and AES algorithm using a C compiler of 3.1 GHz Intel processor Core ™ i3-2100 CPU, 4GB RAM, and Windows 7 32-Bit Operating System. The proposed algorithm was applied to an image file Boat.bmp of size $256 \times 256 \times 3$. Table 1 presents the average time of applying the two versions of the proposed algorithm for $(r = 1, rp = 1, rs = 1)$ and for the AES algorithm. We observe that, the proposed algorithm with lookup table version at least (in average that means without warm up time that happen on the first execution) is 10 times faster than AES algorithm that we applied.

Remark: we have used the AES algorithm given by the following website:

https://code.google.com/p/rikiglue/source/browse/src/frame/aes.cpp?spec=svn9239a0474d811daae909075568688a46134858c6&r=9239a0474d811daae909075568688a46134858c6.

TABLE I. Encryption and decryption Time of the proposed algorithm vs. AES algorithm in seconds

Algorithm Name	Encryption	Decryption
Proposed Algorithm Based Lookup table	0.0060	0.0058
Proposed Algorithm	0.0097	0.0316
AES	0.0643	0.0668

Also, from our knowledge, the proposed scheme is faster than many chaos based encryption algorithms of the literature. Specifically, we compare the obtained results with Yang cryptosystem results [13]. The security level of both cryptosystems is almost the same, while our proposed one is approximately twice faster than Yang scheme.

B. Plain Text Sensitivity Attacks

One bit change on the original plain text P_1 it becomes P_2, then, encrypts the both plain text using the same secret key will produce cipher texts C_1 and C_2 , if we calculate the number of different bits between C_1 and C_2 , then divide the result by the total number of bits, this is called the hamming distance value, in mathematical form is:

$$d_{Hamming}(C_1, C_2) = \sum_{K=1}^{L \times C \times P \times 8} C_1(K) \oplus C_2(K) \quad (13)$$

We executed our proposed algorithm on Boat.bmp of size $256 \times 256 \times 3$ for 1000 different secret keys to take the

average value of the hamming distance, which is *0.500009* and this value is the optimal value that can be reached for plaintext sensitivity attack.

C. Key Sensitivity Attack

In a similar procedure, we can apply the key sensitivity test by changing one bit on the secret key and encrypts P_1 using the original secret key, encrypts P_2 using the key after one bit change, this also produces C_1 and C_2. Then, we calculate three parameters: HD (Hamming distance), the NPCR (Number of Pixels Change Rate) and the UACI (Unified Average Changing Intensity) [14]. The following mathematical forms are used to calculate the last two parameters:

$$NPCR(C_1,C_2)=\frac{1}{L\times C\times P}\sum_{K=1}^{L\times C\times P}D(K)\times 100 \quad (14)$$

Where

$$D(K)=\begin{cases}1 & if\ C_1(K)\neq C_2(K)\\ 0 & f\ C_1(K)=C_2(K)\end{cases} \quad (15)$$

$$UACI(C_1,C_2)=\frac{1}{L\times C\times P\times 255}\sum_{K=1}^{L\times C\times P}|C_1(K)-C_2(K)|\times 100 \quad (16)$$

Notice that UACI and NPCR are calculated in bytes level while HD is calculated in bit level. We executed our proposed algorithm for 1000 different secret keys on the same image. The following table presents the results of the key sensitivity attack. As, we can see these values are near the optimal values.

TABLE II. NPCR, UACI AND HD VALUES FOR THE KEY SENSITIVITY ATTACK TEST

Test Name	Test Value
HD	0.50002
NPCR	99.6098
UACI	33.4667

D. Histogram Analysis

The histogram of the ciphered image of any crypto-system should be uniform; to test the uniformity distribution of the ciphered image we applied the chi-square test.

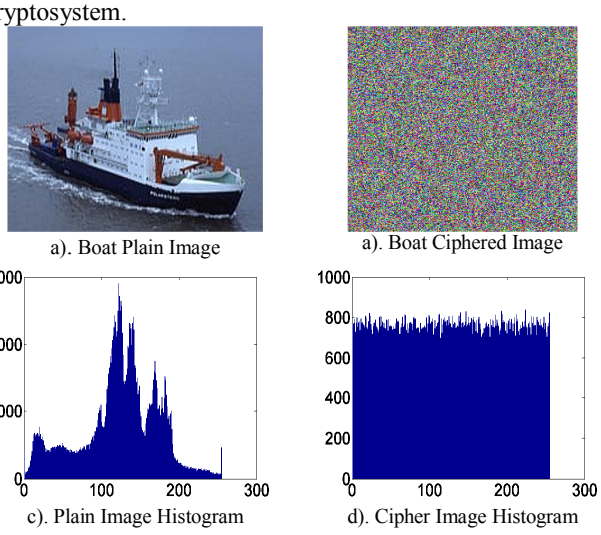
$$\chi^2_{\exp}\sum_{i=0}^{Ny-1}\frac{(O_i-E_i)^2}{E_i} \quad (17)$$

This test was applied on three different nature's images (Boat, Cameraman, and Jet) of size $256\times256\times3$. For a secure cryptosystem the experimental value must be less than the theoretical one, which is 293 in case of alpha=0.05 and num intervals=256. Table 3, presents the calculated chi-square value on each image.

TABLE III. CHI SQUARE TEST RESULTS ON THREE DIFFERENT IMAGES

Image Name	Chi-square Value
Boat	255.431
Cameraman	255.737
Jet	253.787

The histogram of the plain image and its ciphered one are presented in Figure 5, it is clear that the pixel values show a pattern in part c) of the figure which presents the histogram of the plain image, while in part d) the distribution of the pixel values are almost uniform and significantly different from the histogram of the plain image. As a result the statistical analysis of the histogram will be useless or most difficult for cryptanalysis of the ciphered images of this cryptosystem.



a). Boat Plain Image a). Boat Ciphered Image

c). Plain Image Histogram d). Cipher Image Histogram

Figure 5. Histograms of the Boat Plain Image and its Ciphered One

E. Correlation Analysis

One of the most difficult properties of the image encryption algorithms comes from the high correlation between adjacent pixels. To test the security of the proposed algorithm we randomly selected N = 10000 pairs of adjacent pixels in vertical, horizontal, and diagonal directions from the plain image and its ciphered one. Then we calculated the correlation coefficient according to the following equation [8]:

$$\rho_{xy}=\frac{\mathrm{cov}(x,y)}{\sqrt{D(x)\times D(y)}} \quad (18)$$

Where:

$$\mathrm{cov}(x,y)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))\times(y_i-E(y)) \quad (19)$$

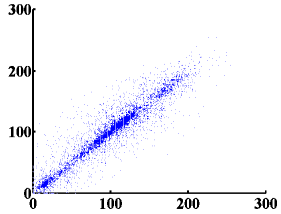
$$D(x)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))^2 \quad (20)$$

$$E(x)=\frac{1}{N}\sum_{i=1}^{N}x_i \quad (21)$$

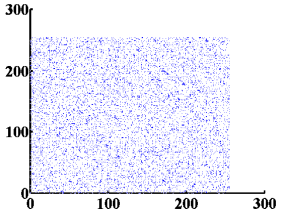
In the above equations, x_i and y_i are the values of the two adjacent pixels in the plain image and the corresponding ciphered image. Figure 6 shows the correlation coefficients of the adjacent pixels in vertical direction for both Boat plain image of size $256\times256\times3$ and the corresponding ciphered image; we omitted the figures of diagonal and horizontal directions since they are similar for vertical one. Table 4 presents the correlation values in the three directions for

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-298-1

Boat, Cameraman and Jet images of the same size $256 \times 256 \times 3$.



a-) Correlation of adjacent pixels in vertical from the Plain image



b-) Correlation of adjacent pixels in vertical from the Cipher image

Figure 6. Correlation Analysis of the Plain and Ciphered Images in Horizontal Direction

TABLE IV. CORRELATION COEFFICIENTS OF PLAIN AND CIPHER IMAGES

Direction	Plain image	Cipher Image	Image Name
Vertical	0.944191	0.008648	Boat
Horizontal	0.936227	0.008683	
Diagonal	0.892431	0.008371	
Vertical	0.869266	0.008267	Airplane
Horizontal	0.929387	0.007203	
Diagonal	0.900825	0.007230	
Vertical	0.980595	0.008538	Cameraman
Horizontal	0.970803	0.008434	
Diagonal	0.951154	0.009412	

It is clear from Figure 6, and table 4, that the correlation coefficients of the adjacent pixels in the plain and the cipher images are far apart, that means the cryptosystem success to convert the high correlation coefficients values from the plain image into very little correlation coefficients between adjacent pixels in the cipher image.

IV. CONCLUSION AND FUTURE WORK

In this paper, we designed and tested an efficient image encryption and authentication scheme based on chaotic sequences. The proposed cryptosystem uses the high sensitivity features of the chaotic systems for initial values by producing the secret key of the chaotic generator from the secret hash key and the plain image. This mechanism permits to achieve the required diffusion and confusion effects. Simulation results and performance analysis show that our proposed cryptosystem at least is ten times faster than AES algorithm and also better than most of the chaos based cryptosystems of the literature, for both security level and time consuming. Our future work will focus on designing and testing chaos-based hash functions.

REFERENCES

[1] A. Akhshan, A. Akhavan, S.C. Lim, and Z. Hassan, "An Image Encryption Scheme Based on Quantum Logistic Map," Communications in Nonlinear Science and Numerical Simulation, vol. 17, no. 12, 2012, pp. 4653-4661.

[2] S. El Assad, "Chaos Based Information Hiding and Security," in 7th International Conference for Internet Technology and Secured Transactions, IEEE, London, United Kingdom, 10-12 Dec. 2012, pp. 67- 72. *Invited paper.

[3] M. Chetto, H. Noura, S. El Assad, and M. Farajallah,"How to Guarantee Secured Transactions With QoS and Real-Time Constraints", in 7th International Conference for Internet Technology and Secured Transactions, IEEE, London, United Kingdom, 10-12 Dec. 2012, pp. 40-44.

[4] S. Rakesh, A. Ajitkumar, B, Shadakshari, and B, Annappa, "Image Encryption Using Block Based Uniform Scrambling and Chaotic Logistic Mapping," International Journal on Cryptography and Information Security –IJCIS, vol. 2, no. 1, 2012, pp. 49-57.

[5] F. Chiaraluce, L. Ciccarelli, E. Gambi, P. Pierleoni, and M. Reginelli, "A New Chaotic Algorithm for Video Encryption," IEEE Transactions on Consumer Electronics, vol. 48, no. 4, 2002, pp. 838–844.

[6] G. Chen, Y. Mao, and C.K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," Chaos, Solitons & Fractals, vol. 21, no. 3, pp. 749-761, 2004.

[7] J. Fridrich, "Symmetric Ciphers Based no Two-Dimensional Chaotic Maps," International Journal of Bifurcation and Chaos, vol. 8, no. 6, 1998, pp. 1259-1284.

[8] C.Y. Song, Y.L. Qiao, and X.Z. Zhang, "An Image Encryption Scheme Based on New Spatiotemporal Chaos," Optik, October 2012.

[9] D. Socek, S. Li, S. Magliveras, and B. Furht, "Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption," in First IEEE International Conference on Security and Privacy for Emerging Areas in Communications Networks, Athens, Greece, 2005, pp. 406 - 413.

[10] S. El Assad (85%), and H. Noura (15%), Generator of Chaotic Sequences and Corresponding Generating System WO Patent WO/2011/121,218, 2011.

[11] Secure Hash Standard, Standard Publication #180-1 (addendum to [1]), 1995, United States Department of Commerce, National Institute of Standards and Technology, Federal Information Processing.

[12] J. Katz and L. Yehuda, Introduction to Modern Cryptography. USA: CRC-Press, 2007.

[13] H. Yang, KW. Wong, X liao, W. Zhang, and P. Wei, "A Fast Image Encryption and Authentication Scheme Based on Chaotic Maps", Communications in Nonlinear Science and Numerical Simulation, vol. 15, no. 11, 2010, pp. 3507-3517.

[14] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," in Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology, Santa Barbara, California, USA, 1990, pp. 2-21.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-298-1