Generation of Chaotic Signal for Scrambling Matrix Content

Naziha Khlif, Ahmed Ghorbel, Walid Aydi, and Nouri Masmoudi
Laboratory of Electronics and Information Technologies, Sfax University, Tunisia

Abstract: *Very well evolved, information technology made so easy the transfer of all types of data over public channels. For this reason, ensuring data security is certainly a necessary requirement. Scrambling data is one solution to hide information from non authorized users. Presenting matrix content, image scrambling can be made by only adding a mask to the real content. A user, having the appropriate mask, can recognize the image content by only subtracting it. Chaotic function is recently used for image encryption. In this paper, an algorithm of image scrambling based on three logistic chaotic functions is proposed. Defined by its initial condition and parameter, each chaotic function will generate a random signal. The set of initial conditions and parameters is the encryption key. The performance of this technique is ensured for two great reasons. First, using masks on the image makes unintelligible its content. Second, using three successive encryption processes makes so difficult attacks. This point reflects, in one hand, a sufficient key length to resist to brute force attack. In the other hand, it reflects the random aspect of the pixel distribution in the scrambled image. That means, the randomness in one mask minimizes the correlations really existent between neighboring pixels. That makes our proposed approach resistant to known attacks and suitable for applications requiring secure data transfer such as medical image exchanged between doctors*.

Keywords: *Chaotic signal, scrambling, image encryption.*

Received April 16, 2017; accepted October 2, 2018
https://doi.org/10.34028/iajit/17/4/13

1. Introduction

One data type needing protection is the image since it presents very significant information. Therefore, scrambling an image becomes a necessity to preserve its security. Chaos is frequently used because a chaotic signal is sensitive to initial conditions change representing the encryption key and also because the chaotic function is irreversible. In other words, it is extremely difficult to know the key with chaotic sequences. Many researchers have recently proposed schemes based on chaotic functions to scramble the images [1, 2, 3, 4, 7, 10, 13, 16]. In this paper, we proposed an encryption scheme relying on three chaotic functions to scramble the image content. The same function used is the logistic map [8]. We only changed the three initial conditions in order to have three different chaotic signals. The remainder of this paper was organized as follows: In section 2, we surveyed such image encryption algorithms. Section 3 was devoted to describing our scheme. The experimental results according to different encryption effect analyses and the cryptographic security of our algorithm were discussed in section 4. Finally, in section 5, we drew our conclusions and suggested some future perspectives of our study.

2. Related Works

Several schemes have been recently proposed in the literature to hide image content. Tedmori and Al-Najdawi [12] proposed a lossless encryption algorithm based on the discrete cosine transform to protect images. Winquing *et al*. [3] conceived a scheme to encrypt image based on mixed chaotic sequences. These sequences are used to produce the corresponding permutation matrix and offset matrix. The matrix is used in the wavelet domain to scramble pixels. Dong *et al*. [13] proposed an algorithm to scramble digital images based on chaotic sequences and decomposition and recombination of pixel values. Both position and pixel values are changed. Logistic map is the chaotic function used for the two works [3, 13]. Rathore and Suryavanshi [10] proposed to double encrypt images using two different chaotic maps which are logistic and Barker maps. Chakraborty *et al*. [2] conceived an image encryption method using Substitution and the xor operation with chaotic sequences generated by logistic map. Qi *et al*. [9] proposed a scheme to encrypt images. In fact, the xor operation is run between the one dimensional converted image and chaotic sequence generated by the Lorenz map. Apeksha *et al*. [1] used an encryption algorithm based on two stages permutation and diffusion. Pixels are shuffled according to a chaotic sequence generated by an Arnold cat map in the permutation stage. Whereas, the diffusion stage consists of calculating the xor operation with first, the current plain pixel, then the key stream element after that with output cipher-pixel and finally with the previous cipher-pixel. Mondal and Mandal [7] used an encryption algorithm which is divided into three stages. The three

stages are the random number generation process based on a chaotic standard map, the image permutation process and the substitution process. In the work [4], authors conceived an algorithm based on multiple chaotic mapping to encrypt medical images. A Logistic-sine chaos mapping was firstly used to scramble the plain image. Then, divided into 2-by-2 sub blocks, the scrambled image was adaptively encrypted sub block by sub block using the hyper-chaotic system. As mentioned above, many chaotic functions can be used as a random number generator of random sequences used for image encryption. All the schemes used the same principle to encrypt images with different manners. In this paper, we adopted, as recently used, a chaotic encryption algorithm to scramble the image content.

3. Proposed Method

Our purpose is to conceive a crypto-system used for image protection. For this aim, we used a chaotic encryption algorithm. The principle of the chaotic cryptography is represented in Figure 1 as described in the work [5].

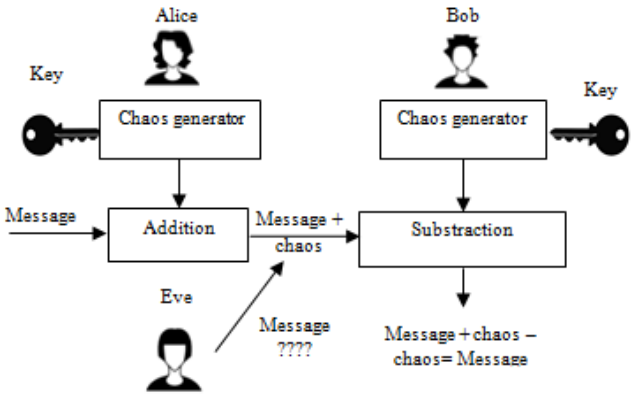

Figure 1. Chaotic cryptography principle.

Chaos is a noise, when added to the message, it produces a scrambled effect. We used the chaotic signal defined by the following logistic function [8]:

$$X_{n+1} = \mu X_n (1 - X_n) \quad (1)$$

Where μ is a parameter in the interval [0, 4], X_n is a point in the interval [0, 1]. Depending on the value of μ, we obtained a convergent sequence, a sequence subject to oscillations or chaotic signal. Chaos sets in μ = 3.57. For μ = 4, the behavior of the chaos is optimal, because in this case, the amplitude X_n covers all the dynamics [0, 1] of the map. The encryption key is the pair (μ, X_0), where X_0 is the initial condition. Our encryption algorithm is based on three logistic functions having the same parameter μ=4 and three different initial conditions. Each logistic function generates a chaotic matrix with the same image size. We obtained this random matrix by concatenating chaotic value generated. Each chaotic value X_n is transformed into a three-digit integer. The last one is equal to thousand times the approximation to 10^{-3} near the random chaotic value X_n. The encryption process is the addition of the three chaotic matrices to the image content. It is like an addition of three masks to the image. The decryption process is only to subtract the same generated matrix from the encrypted image. Our scheme is divided into two steps: the random matrix generation similar for the sending and the reception and the encryption/decryption processes.

3.1. Random Matrix Generation

The first stage of our scheme is the generation of three random matrices. A schematic representation of the random matrix generation is given in Figure 2.

For all the chaotic generators, we used the same parameter μ=4 because the behavior of the chaos is optimized for this value. We only change the initial condition for each generator. Both initial condition and all values generated are in the interval [0, 1] with double precision. Each value generated will be used for the next generation. At the same time, it is passed to the mathematic processing module. This operation consists of multiplying by 10^3 the approximation to 10^{-3} near this value generated. Then, being a three-digit number, it passes to the scheduling module to preserve its position in the random matrix. If the image to be encrypted has a size of M*N, the chaotic generator should be run M*N times to obtain (after scheduling module) an M*N matrix used as a mask. In our proposed scheme, the first step of our algorithm called the random matrix generation leads to three different masks. After that, these masks are added to the original image to have a scrambled one. This step is carried out identically in the reception part while the decryption process.

3.2. Encryption/Decryption Processes

The encryption process consists of adding the three random matrices to the original image. Having the same encryption key, the receiver is able to generate the same three random matrices used for the encryption, to decrypt the received encrypted image. The decryption operation is just a subtraction operation between the encrypted image and the three random matrices. Figure 3 explains our scheme. We give, in Figure 4, an example of the encryption/decryption processes of the first 4×4 block size of the 128×128 resolution Lena BMP image. After the encryption process, the modulo 256 operation is applied in order to have the same image structure.

.

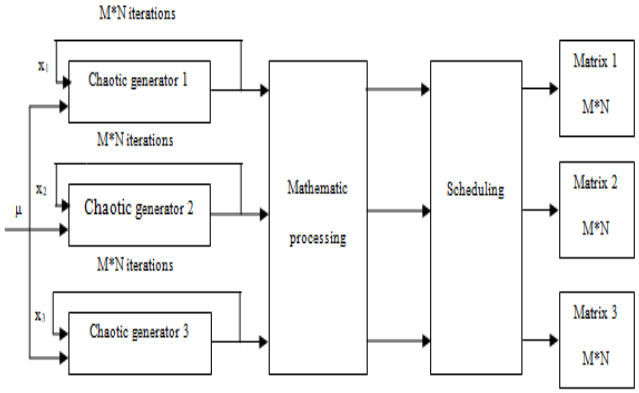

Figure 2. Schematic representation of random matrix generation

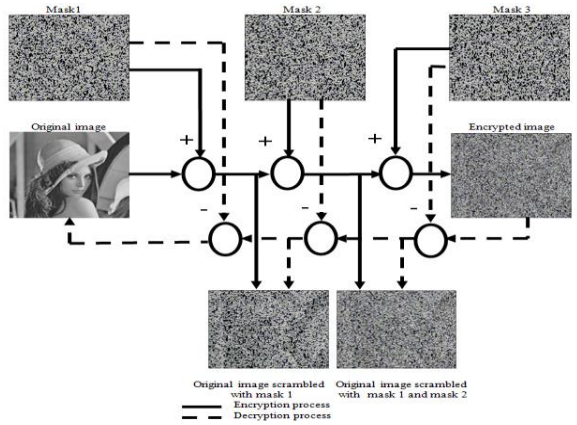

Figure 3. The proposed encryption/decryption scheme.

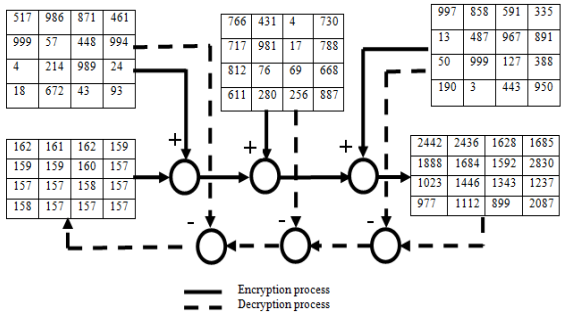

Figure 4. The first 4×4 encrypted/decrypted examples block size of the 128×128 resolution Lena BMP image.

4. Results and Discussions

4.1. Experimental Conditions

Our encryption and decryption algorithms are tested on four bmp images (Lena, Barbara, Baboon and Parrots) with three different sizes each one (128×128, 256×256 and 512×512). Our algorithms are implemented with the Matlab 2008. We discussed the perceptual and cryptographic security of our scheme. The execution time of the encryption algorithm is computed by an Intel (R), Core i3, 2.3 GHz machine.

4.2. Perceptual Effect

We collect in Figure 5 the snapshots of both Lena and Barbara images with different sizes. It is clearly noticeable proving that the two image's contents are well hidden which reflects the strong scrambling effect of our scheme. To better conclude about the perceptual security of our scheme we calculate the Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) values of the encrypted images. These metrics are frequently used to determine the visual quality of the compressed images. The PSNR allows the measurement of distortion in a digital image relevant on the image compression. The PSNR is given by the following formula [9]:

$$\mathrm{PSNR} = 10\log_{10}\frac{R^2}{\mathrm{MSE}} \quad (2)$$

Where R is the maximum value of a pixel, equal to 255 for an 8-bit pixel representation. The Mean Square Error (MSE), given by the following Equation:

$$\mathrm{MSE} = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[I_1(m,n) - I_2(m,n)]^2 \quad (3)$$

Where M and N are the width and height in pixels of an image.

m and n are the coordinates of the considered pixel. I_1 and I_2 are the pixel values for the original and encrypted images, respectively. Unlike the PSNR which calculates the difference pixel by pixel, SSIM measures the structural similarity between the two images. This metric is calculated on several windows of an image. The SSIM between two windows x and y of the same size is given by the following Equation [14]:

$$\mathrm{SSIM} = \frac{(2\mu_x\mu_y + c_1)(2\mathrm{cov}_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\theta_x^2 + \theta_y^2 + c_2)} \quad (4)$$

Where μ_x is the x mean

μ_y is the y mean

θ_x^2 is the x variance

θ_y^2 is the y variance

cov_{xy} is the x and y covariance

$C_1=(K_1L)^2$

$C_2=(K_1L)^2$

$c_2 = (K_2L)^2$ where C_1 and C_2 are two variables to stabilize the division when the denominator is too low.

L is the pixel values dynamic which is equal to 256 for images coded on 8 bits.

K_1 = 0.01 and K_2 = 0.03 by default.

In Table 1, we give the PSNR and SSIM distortions of encrypted images with three different sizes. From this table, we prove the strong scrambling effect of our encryption algorithm. This conclusion comes from the lowest PSNR and SSIM values of encrypted images. It should be remembered that for two identical images, the PSNR value is close to 99 dB and the SSIM absolute value is close to 1. For a good encryption, the PSNR value should be as low as possible and the SSIM value must be close to 0. This is the case of the PSNR and SSIM values obtained from the encrypted images at different sizes.

4.3. Statistical Encryption Effect

In Table 2, we give the percentage of the Encryption Ratio (ER). In which, we compute the number of pixels changed between the original and encrypted images. From this table, it is noticeable that almost 99% of pixel are changed. This result with the SSIM values which are close to 0 (Table 2) reflects the non understandability of the encrypted image content.

4.4. Cryptographic Security

A good encryption effect of an algorithm should satisfy some conditions that make it resistant to attacks. We verified here the same criteria adopted in the work [6]. We prove in this part the resistance of our encryption algorithm to brute force and differential attacks.

1. Key space analysis: In cryptography, a key symmetric encryption algorithm should resist to a brute force attack where the security is limited to 2^{128} [11]. In our scheme, the secret key includes three floating-point numbers X_{00}, X_{01} and X_{02}, representing the three initial conditions combination of the three chaotic used functions, with a precision of 10^{15} each one and 3 bits to represent $\mu = 4$. Thus, our key space is $(3+10^{45})$ which is approximately equal to 2^{153} making this key sufficient to resist a brute force attack.
2. Differential analysis: One differential analysis for an encryption scheme is to check the Number of Pixel Change Rate (NPCR) [15] between two different images with the same size.

The NPCR is defined as follows:

$$NPCR(\%) = \sum_{i,j} \frac{D(i,j)}{T} \times 100 \quad (5)$$

Where $D(i, j)$ is computed as:

$$D(i,j) = \begin{cases} 0 \ if \ C^1(i,j) = C^2(i,j) \\ 1 \ if \ C^1(i,j) \neq C^2(i,j) \end{cases} \quad (6)$$

Where C^1 and C^2 are ciphertext images before and after one pixel change in a plain text image, respectively. $C^1(i, j)$ and $C^2(i, j)$ are pixel values on the (i, j) grid. T denotes the total pixel number in the ciphertext.

For a good encryption, the NPCR should be close to 100% as possible. Here, we change 1 LSB in the sub key X_{00} or X_{01} or X_{02} and we compute NPCR between the two encrypted images with original and 1 bit changed keys. In Table 3 we give the different keys used. Table 4 shows the NPCR's variation of encrypted images with three different sizes. While comparing encrypted images, the NPCR value is almost equal to 99%. This reflects that the slightest change in the encryption key leads to a great change in the ciphertext. Let's proceed conversely. That means let's try to decrypt, using the same key with 1 bit changed, the same encrypted image using the original key.

In other words, could this inform us about the image content after the decryption process? Figure 6 shows snapshots of decrypted, Lena encrypted image using the original key, with 1 bit changed in X_{00}, X_{01} and X_{02}. From this figure (Figure 6), it is noticeable that the decrypted image is totally unintelligible even when only 1 bit is changed in the decryption key. Thus, any simple change in the key makes the decryption impossible.

4.5. Time Cost

A verification of the computing time speed of the encryption algorithm is always necessary to know the efficiency of a scheme. In Table 5 we give the time cost of our algorithm for three different sizes of the considered images. It is noticeable that the encryption process is very speedy. It has not exceeded a few milliseconds. That proves the rapidity of our scheme. But we should compare it with other schemes to better conclude its efficiency.

Table 1. PSNR and SSIM distortion of encrypted images with three different sizes.

	128×128		**256×256**		**512×512**	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Lena.bmp	11.1058	0.0552	9.2798	0.0283	12.6593	0.0309
Barbara.bmp	10.2240	0.0577	10.9871	0.0304	11.2999	0.0294
Baboon.bmp	10.4521	0.0615	10.6219	0.0493	10.8391	0.0361
Parrots.bmp	10.1082	0.0481	9.8271	0.0395	10.4311	0.0299

Table 2. Encryption Rate (ER%) of encrypted images with three different sizes.

	ER (%)		
	128×128	256×256	512×512
Lena.bmp	99.4202	99.5819	99.3858
Barbara.bmp	99.5300	99.4507	99.4480
Baboon.bmp	99.5220	99.5476	99.5301
Parrots.bmp	99.5460	99.4924	99.5187

Table 3. Different keys used.

	Original key	**1st key**	**2nd key**	**3rd key**
1st sub key (X_{00})	0.8476235917246 83	0.8476235917246 82	0.8476235917246 83	0.847623591 724683
2nd sub key (X_{01})	0.2581397246193 58	0.2581397246193 58	0.2581397246193 59	0.258139724 619358
3rd sub key (X_{02})	0.5281496372859 62	0.5281496372859 62	0.5281496372859 62	0.528149637 285963

Table 4. NPCR's variation of encrypted images with three different sizes.

NPCR (%)				
Image	The LSB changed in	128×128	256×256	512×512
Lena.bmp	1st sub key	99.04	99.15	99.12
	2nd sub key	99.10	99.07	99.12
	3rd sub key	99.40	98.98	98.09
Barbara.bmp	1st sub key	99.34	99.34	98.38
	2nd sub key	99.13	98.69	98.76
	3rd sub key	98.84	99.03	99.27
Baboon.bmp	1st sub key	99.43	98.95	98.94
	2nd sub key	99.25	98.97	98.93
	3rd sub key	99.18	99.01	99.00
Parrots.bmp	1st sub key	99.15	99.27	99.37
	2nd sub key	99.18	99.51	99.47
	3rd sub key	99.07	99.43	99.52

Table 5. Cost at encryption time computed with encrypted images considering three different sizes. **NPCR (%).**

	Encryption time (ms)		
Encrypted image	128×128	256×256	512×512
Lena.bmp	1.4	3.5	16.9
Barbara.bmp	1.7	2.3	16.2
Baboon.bmp	1.5	2.5	16.0
Parrots.bmp	1.7	3.2	16.7

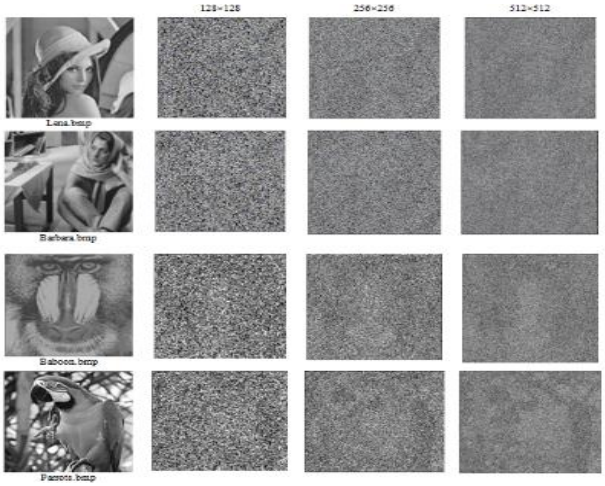

Figure 5. Snapshots of encrypted images with three different sizes.

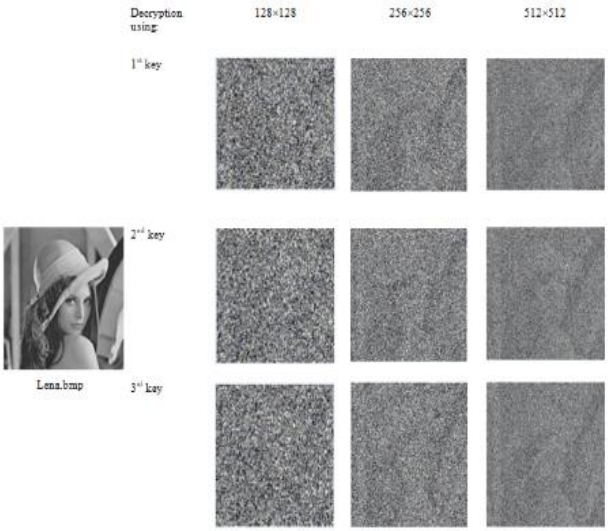

Figure 6. Snapshots of encrypted images with three different sizes.

4.6. Scheme Evaluation

Our scheme is based on three chaotic sequences to hide image content. It provides a very scrambling effect. PSNR and SSIM values are very distorted, which improve the perceptual scrambling of our scheme. It is also proven quick and resistant to brute force and differential attacks. In order to evaluate objectively our encryption algorithm, we compared in Table 6 our results with other works.

It is noticeable that we evaluate our scheme based on many tests. From Table 6, it is shown that most of the tests were not reported in the considered works which are the key space, encryption rate, SSIM, NPCR for the work [10] and time cost for the work [7]. The work [4] considered NPCR test and key space analysis. We tried in this table to compare some of our results with the mentioned tests found in the literature. Other tests are not available (symbol NA in the table).

We obtained the low PSNR value compared to the works [7, 10] and at low cost in time compared to the work [10]. For the NPCR value, both our scheme and the two works [4, 7] reached almost the 99%. The PSNR and time cost's values reflect that our scheme seems to be more efficient.

5. Conclusions

In this paper, we proposed an image encryption scheme based on the addition of chaotic sequences to the plaintext. The 153 bits key is used to assure the algorithm resistance to brute force attack. Our scheme is proven quick and resistant to differential attack. It has also a very strong perceptual security. In fact, encrypted images are unintelligible and have low PSNR and SSIM values compared to the original ones. Our scheme can be adopted with any matrix content whatever its length. To this end, we intend to apply our algorithm in pattern recognition. This will be studied and proved in our future work.

Table 6. Comparison of the proposed scheme with other works.

Schemes	**Method**	**Encryption process**	**Considered tests**					
			Key space (bits)	ER (%)	PSNR (dB)	SSIM	NPCR (%)	Time cost (s)
Rathore and Suryavanshi [10]	**2 chaotic maps (1 Logistic and 1 Barker)**	**Dual encryption**	NA	NA	≈ 40	NA	NA	0.169
Mondal and Mandal [7]	**3 chaotic standard maps**	**Permutations and substitution processes**	NA	NA	-39.589	NA	≈ 99	NA
Chen and Hu [4]	**Logistic sine map and hyper chaotic system**	**multiple chaotic mapping**	$10^{64} \times 4^{256}$	NA	NA	NA	99.59	NA
Proposed scheme	**3chaotic logistic maps**	**Addition of 3 random masks**	153	≈ 99	≈ 10	≈ 0.038	≈ 99	≈ 0.007

The missing values are represented by the symbol NA (Not Available.

References

[1] Apeksha W., Abhishek B., Abhishek S., and Sanuj K., "Chaos Based Image Encryption and Decryption," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 64-68, 2016.

[2] Chakraborty SH., Seal A., Roy M. and Mali K., "A Novel Lossless Image Encryption Method using DNA Substitution and Chaotic Logistic Map," *International Journal of Security and Its Applications*, vol. 10, no. 2, pp. 205-216, 2016.

[3] Chen W., Wang T., and Wang B., "Design of Digital Image Encryption Algorithm Based on Mixed Chaotic Sequences," *International Journal on Smart Sensing and Intelligent Systems*, vol. 7, no. 4, pp. 1453-1469, 2014.

[4] Chen X. and Hu C., "Adaptive Medical Image Encryption Algorithm Based on Multiple Chaotic Mapping," *Saudi Journal of Biological Sciences*, vol. 24, no. 8, pp. 1821-1827, 2017.
[5] Khlif N., Damak T., Kammoun F., and Masmoudi N., "Joint Selective Encryption of CAVLC and Signs of Motion Vectors for H.264/AVC," *Journal of Testing and Evaluation*, vol. 44, no. 1, pp. 160-174, 2016.
[6] Khlif N., Masmoudi A., Kammoun F., and Masmoudi N., "Secure Chaotic Dual Encryption Scheme for H.264/AVC Video Conferencing Protection," *IET Image Processing*, vol. 12, no. 1, pp. 42-52, 2018.
[7] Mondal B. and Mandal T., "A Nobel Chaos based Secure Image Encryption Algorithm," *International Journal of Applied Engineering Research*, vol. 11, no. 5, pp. 3120-3127, 2016.
[8] May R., "Simple Mathematical Models with Very Complicated Dynamics," *Nature*, vol. 261, pp. 459-467, 1976.
[9] Poobathy D. and Chezian R., "Edge Detection Operators: Peak Signal to Noise Ratio Based Comparison," *International Journal of Image, Graphics and Signal Processing*, vol. 6, no. 10, pp. 55-61, 2014.
[10] Rathore D. and Suryavanshi A., "A proficient Image Encryption using Chaotic Map Approach," *International Journal of Computer Applications*, vol. 134, no. 10, pp. 20-24, 2016.
[11] Schneier B., *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Wiley Computer Publishing, 1996.
[12] Tedmori S. and Al-Najdawi N., "Lossless Image Cryptography Algorithm Based on Discrete Cosine Transform" *The International Arab Journal of Information Technology*, vol. 9, no 5, pp. 471-478, 2012.
[13] Wang D., Chang C., Liu Y., Song G., and Liu Y., "Digital Image Scrambling Algorithm Based on Chaotic Sequence and Decomposition and Recombination of Pixel Values," *International Journal of Network Security*, vol. 17, no. 3, pp. 322-327, 2015.
[14] Wang Z., Bovik A., Sheikh H., and Simoncelli E., "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 1-14, 2004.
[15] Wu Y., Noonan J., and Agaian S., "NPCR and UACI Randomness Tests for Image Encryption," *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications JSAT*, vol. 1, no. 2, pp. 31-38, 2011.
[16] Zhang Q., Guo Y., Li W., and Ding Q., "Image Encryption Method Based on Discrete Lorenz Chaotic Sequences," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 3, pp. 576-586, 2016.

Naziha Khlif received her diploma in Electrical Engineering in 2006, the master degree in Electronics in 2011 and the PhD in Electrical Engineering in 2016 from the National Engineering School of Sfax University of Sfax. She is currently a postdoctoral research fellow at Sfax University. Her research interests include image processing, video coding, cryptography and data security.

Ahmed Ghorbel received his diploma in Electrical and Automation Engineering in 2012 and the master degree in Automation and intelligent technology in 2013 from the National Engineering School of Gabes University of Gabes. He is currently a PhD student at the National Engineering School of Sfax, Sfax University since September 2014. His research interests include face recognition.

Walid Aydi received the degree of Engineering in Electrical Engineering in 2008, Masters Degree in Engineering in the major of electronics and telecommunication 2009, and a Ph.D degree in Electronic Engineering in 2013, from the National Engineering School of Sfax. From 2014 to 2017, he held the position of Assistant Professor in the Higher Institute of Computer Science and Multimedia, Gabes, Tunisia. He has been employed as an assistant professor in The Prince Sattam bin Abdullaziz University in Saudi Arabia since 2017. His research interests focus on image processing, pattern recognition, and computer vision.

Nouri Masmoudi received his electrical engineering degree from the Faculty of Sciences and Techniques Sfax University in 1982, the DEA degree from the National Institute of Applied Sciences Claude Bernard University Lyon in 1984. He received his PhD degree from the National Engineering School of Tunis (ENIT) in 1990. He is currently a professor in the electrical engineering department in the National Engineering School of Sfax. His research activities include Design, Telecommunication, Embedded Systems, Information Technology, Video Coding and Image Processing.