Journal of Information Security and Applications manuscript No.
(will be inserted by the editor)

A Lightweight and Efficient Digital Image Encryption Using Hybrid Chaotic Systems for Wireless Network Applications

Islam T. Almalkawi[a,1], **Rami Halloush**[b,2], **Ayoub Alsarhan**[c,1], **Ahmed Al-Dubai**[d,3], **Jamal N. Al-karaki**[e,1,4]

[1]Hashemite University, Zarqa, Jordan
[2]Yarmouk University, Irbid, Jordan
[3]Edinburgh Napier University, Edinburgh, UK
[4]Abu Dhabi Polytechnic, Abu Dhabi, UAE

Received: date / Accepted: date

Abstract Due to limited processing capabilities and other constraints of most wireless networks, many existing security algorithms do not consider the network efficiency. This is because most of these security solutions exhibit intolerable overhead and consider only securing scalar data, which are not suitable for other data types such as digital images, hence affecting the provided security level and network performance. Thus, in this paper, we propose a lightweight and efficient security scheme based on chaotic algorithms to efficiently encrypt digital images. Our proposed algorithm handles digital images in two phases: Firstly, digital images are split into blocks and compressed by processing them in frequency domain instead of Red-Green-Blue (RGB) domain. The ultimate goal is to reduce their sizes to speed up the encryption process and to break the correlation among image pixel values. Secondly, 2D Logistic chaotic map is deployed in key generation, permutation, and substitution stages for image pixel shuffling and transposition. In addition, 2D Henon chaotic map is deployed to change the pixel values in the diffusion stage in order to enhance the required level of security and resist various security attacks. Security performance analysis based on standard test images shows that our proposed scheme overcomes the performance of other existing techniques.

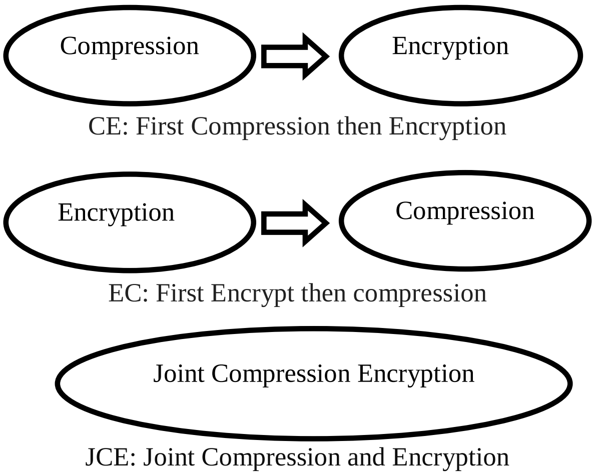
Keywords

Secure Wireless Networks, Image Encryption, Chaotic Map, Henon Map, 2D Logistic Map, Diffusion, Permutation, and Substitution.

1 Introduction

Communications through wireless channels are vulnerable to several security challenges associated with transmitting data over open and shared mediums. In wireless networks, a plethora of applications deal with transmitting different data types, such as scalar data, still images, videos, and real-time streaming multimedia data. These types of data have different characteristics and need different requirements for efficient transmission. One of these requirements is to provide an efficient and lightweight mechanism for security and privacy. While efficiency here refers to the capability of providing strong security level against most types of attacks, lightweight indicates reasonable computational and communication incurred overhead that do not affect the overall network performance. Thus, any proposed security scheme should protect the network against the different types of attacks while considering the various constraints and limitations of the wireless network in terms of key elements including bandwidth, data rate, packet size, processing capability, time delivery, energy consumption, etc [1]. Besides the characteristics and limitations of wireless networks, the advancement of the camera sensing technology and the widespread of social-media applications over wireless networks have increased volumes and data types of exchanged data. One of these data types is digital images that have different characteristics compared to normal scalar data in terms of high volume (bulk capacity), high correlation among neighboring pixels, and time delivery constraints for real-time applications. These characteristics raise the issue of using image compression schemes to reduce the exchanged data size. In addition, they highlight the need for efficient encryption techniques to replace the existing traditional scalar data encryption techniques that are not suitable for image encryption in modern wireless networks [2], especially in terms of confidentiality, integrity, and authenticity of digital image

[a]e-mail: eslam.malkawi@hu.edu.jo
[b]e-mail: rami.h@yu.edu.jo
[c]e-mail: ayoubm@hu.edu.jo
[d]e-mail: A.Al-Dubai@napier.ac.uk
[e]e-mail: Jamal.alkaraki@adpoly.ac.ae



CE: First Compression then Encryption

EC: First Encrypt then compression

JCE: Joint Compression and Encryption

Fig. 1: Classification of Compression and Encryption Schemes.

transmission. Moreover, the image encryption techniques have to be lightweight without noticeable degradation of network performance in terms of processing, energy, and time consumption.

In theory, the concept of reducing data size by compression, and the concept of encrypting the data to achieve confidentiality and integrity may look contradictory [3]. However, the two concepts can be used together to maximize the benefits and obtain best results in exchanging large volumes of data safely and efficiently. In general, the combination of compression and encryption can be achieved using three techniques, as shown in Figure 1: 1) First a compression scheme is used, then an encryption scheme (compression-encryption). 2) First an encryption scheme is used, then a compression scheme (encryption-compression). 3) Both schemes are employed together in a single process (hybrid compression-encryption).

The CE technique is proposed in this paper where a compression scheme is used first to reduce image size in order to accelerate the encryption process, and second to break the high redundancy among the neighboring pixels in order to increase the efficiency of the encryption scheme. Then the technique is followed by the proposed encryption method.

1.1 Image Compression

Image compression, as shown in Figure 2, can be understood as a process of producing a shorter representation in terms of number of bits, compared with the original image without perceivable information loss. Compression schemes are classified into two main groups based on reversibility: lossless and lossy [4]. In the lossless compression schemes, the original image can be fully recovered from the compressed data without losing any bits and this scheme is suitable for text and scientific images. On the other hand, reconstructed images from lossy compression schemes will not be identical to the original images, however, the lost data is not significant in most cases.

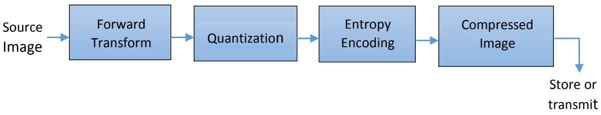


Fig. 2: Image Compression Process.

There are many compression techniques based on different transforms in literature, such as the Karhunen-Loeve Transform (KLT), Discrete Fourier Transformation (DFT), Discrete Wavelet Transformation (DWT), Discrete Cosine Transformation (DCT) [5–7]. Although KLT compression performance is optimal [8], it is not widely well known and still not used in the standard image compression schemes. In comparison to other techniques [9], the DCT scheme reduces the blocking effects due to DCT homogenization extension properties and because it uses real number based calculations. This makes the DCT algorithm much simpler, faster, and more commonly used in standard digital image compression schemes such as JPEG, MPEG, H.261. These features of DCT make it more suitable to meet the requirements of wireless applications.

1.2 Image Encryption

A wide range of cryptographic techniques and methods have been developed to secure transmitted data [10]. However, most of these algorithms have disadvantages when dealing with image data in terms of small key space, low level of security and high computational complexity [11, 12]. To tackle these challenges in image protection, many chaos-based cryptographic algorithms have been suggested [12–15]. These chaos-based security methods have many useful security properties: aperiodicity, unpredictable and non-linearity, high sensitivity to initial conditions and parameters, ergodicity, pseudo-randomness, and other high permutation and diffusion properties [16]. One of the well-known chaotic maps used in image cryptography is the two-dimensional Logistic Map [17] due to the aforementioned properties and for its fast computation, which make it very suitable for image encryption in wireless network communications [18]. 2D Logistic map is expressed in below equations:

$$\begin{aligned} x_{i+1} &= r \times (3y_i + 1) \times x_i(1 - x_i) \qquad (1) \\ y_{i+1} &= r \times (3x_i + 1) \times y_i(1 - y_i) \end{aligned}$$

Where x_0 and y_0 are the initial state, and r is the system parameter. If the system parameter $r \in$[1.11, 1.19], then 2D Logistic map behaves chaotic. We refer the reader to [17] for more details about the generated value representations.

1.3 Paper Contribution

In this paper, we propose an efficient and lightweight image encryption scheme based on hybrid chaotic systems. Our proposed security scheme is implemented in two phases: image compression, and image encryption. DCT based compression is used to decrease the image size, and therefore speeding the encryption process with less computation and communication requirements. Moreover, DCT transforms the image into frequency domain rather than RGB and hence removes the similarities among the neighboring pixels to increase the security efficiency of the proposed encryption scheme.

The image encryption phase is composed of four steps. We first generate a secret key based on 2D Logistic map to be used for the image encryption processes. The generated key is 256 bit long to provide a large key space and resist against most type of attacks. Then, we use 2D Logistic map in the permutation stage to shuffle only pixel's rows and columns, not every pixel in the image to further reduce the computational overhead. In this stage, the plain image is considered as a matrix of 2D array (M columns and N rows). Using the 2D logistic map, two chaotic sequences are generated of M and N length respectively to perform row/column shuffling and obtain the permuted image. Then in the diffusion stage, we use another type of chaotic map to increase the security level, which is the 2D Henon Map [19], to change the pixel values of the permuted image. This is done by first dividing the image into $S \times S$ blocks, where S is determined based on plain image format (4×4 for gray scale or RGB image, and 32×32 for binary images). Then pixels of each block will be mixed with random sequence matrix generated by Henon map. At the end, a substitution stage is added also to shift each diffused image's pixel using a reference matrix generated by 2D Logistic map.

The simulation for performance evaluation shows that our proposed scheme meets the security requirements of image encryption and has several advantages including large key space , high resistance to statistical and differential attacks measured by NPCR, UACI, information entropy, and correlation of adjacent pixel values and it has high sensitivity to any change in the generated secret key.

The rest of this paper is organized as follows: Section 2 reviews related work using chaotic maps in securing digital images. Section 3 presents our proposed security scheme and analyzes its properties. Section 4 evaluates the security performance of our proposed image encryption scheme, and Section 5 outlines conclusions of the study.

2 Related Work

Ankita et al. [20] proposes a multiple-image encryption technique based on the chaotic confusion-diffusion strategy and chaotic Discrete Fractional Random Transform (DFRNT). The proposed scheme starts deriving a composite image by combining the even and odd bits of three images. Then, chaotic confusion-diffusion strategy using four chaotic maps, namely, Arnold Cat Map (ACM), Logistic-Tent Map (LTM), Logistic-Sine Map (LSM) and Tent-Sine Map (TSM) is applied to scramble and diffuse the pixel values of the composite image. Although the authors show that their scheme obtained good security results, it still needs extra storage requirements to store three images to process them. In addition, their confusion-diffusion scheme processes the composite image pixel by pixel using four chaotic maps which leads to increase the complexity of the system.

To resist against chosen-plaintext and chosen-ciphertext attack, Yueping et al. [21] proposes a hyper-chaos-based image encryption algorithm that adopts a 5-D multi-wing hyperchaotic system. In this scheme, the secret key is generated by hyper-chaotic system with relation to the original image. Then, pixel-level and bit-level permutations are employed and then followed by a diffusion operation to change pixel values. However, this bit-level algorithm is not efficient and needs a large number of chaotic iterations for the bit-level decomposition. In addition, the initial condition of chaotic map is fixed on secret key or plaintext which may cause a risk of resistance to known plaintext and chosen plaintext attack, as explained in Zhang et al. [22]. An image encryption technique is proposed by Bhaskar and Tarni [23] based on cross coupled chaotic logistic map with two sets of keys. In this scheme, two pseudo random number sequences are generated from the logistic map: the first sequence is used for permuting the plain image whereas the second sequence is used for generating random DeoxyriboNucleic Acid (DNA) sequence. Next, the plain image is permuted by the first random sequence and encrypted by DNA computation. However, the proposed technique uses multi-round iteration process in addition to floating-point calculations and conversion operations, which introduces a considerable overhead in terms of latency and required processing capabilities.

An improved color image encryption scheme has been proposed by Yashasvee et al. [24] based on 2D Logistic map and advanced encryption standard (AES). However, AES, which is a symmetric block cipher, is not a lightweight scheme for image encryption, hence is not suitable for wireless communication in terms of complexity and time constraints. This was clearly depicted from their time analysis of AES encryption. In the scheme proposed by Prusty et al. [25], the encryption and the decryption of an image is performed by shuffling the image pixel locations using Arnold Cat Map and generating a pseudo-random number (security key) by using Henon map. After that, an XOR (eXelusive OR) operation is performed on the pixel value and the key value generated by the Henon map. The shuffling process in this scheme is done for every pixel location of an image, which leads to consume much time especially for large image size. An image encryption algorithm based on Ikeda and Henon map is presented by Sekertekin and Atan [26], where firstly rows and columns of the image are shuffled by Ikeda map and then pixel values are changed by Henon map. Then, image pixels are shuffled again using Ikeda map. The security key used in this scheme is only generated from the parameters and initial conditions of both chaotic maps. This indicates that the key space is small which lowers the security level of the scheme.

A chaos-based image encryption scheme is proposed by Gopalakrishnan et al. [27] with permutation and diffusion structure. The proposed encryption scheme undergoes three phases: mixing process, permutation process, and diffusion process. The image quality is degraded in the mixing phase using the logistic map. In the permutation stage, the plain image pixels are shuffled many rounds using the Tent map. In the diffusion process, the permuted image pixels are changed and processed using XOR with a random binary sequence. Rani and Kumar [28] proposed an image security scheme using chaotic logistic map for pixel shuffling and modified RC4 stream cipher for image encryption. A hybrid encryption technique is proposed by Kester et al. [29] for digital image encryption between Diffie-Hellman public key cryptography and pixel shuffling algorithm. In this proposed scheme, Diffie-Hellman key exchange algorithm is firstly used to exchange security keys between the sender and receiver. Then, a shared secret key is generated from the private key of the first party and the public key of the second party, and hashed using the MD5 algorithm. This generated key then is used to encrypt the image pixels. Finally, the encrypted image is further confused by pixel shuffling and displacement algorithm. In the work of [27–29], we notice that the encryption process is also done in pixel-wise manner which requires extra calculations and processing time.

An image encryption scheme based on high level chaotic map and improved gravity model is proposed by Majid et al. [30]. In this proposed scheme, the sum of the whole image pixel values is calculated first and used for initial state condition of Sinus Power Logistic chaotic map. Then, the original image is permuted by using cross map for pixel shuffling. After that, the image is encrypted using bit-XOR with the permuted image pixels and gravity model. However, this scheme involves every image pixel in the permutation stage and depends also on bitwise XOR operations for each pixel of the permuted image to generate the encrypted image. As proposed by Xie et al. [31], a three-dimensional (3D) optical chaotic Arnold‘s cat map is used to shuffle the position of image pixels because of its higher security and faster speed compared with 1D and/or 2D chaotic maps. The permuted image is then confused by changing the value of the pixels using the logistic map and XOR operation. This scheme is based on optics chaos and requires additional hardware like optical lasers where they are only used to generate the key. In addition, the shuffling process in this system is done at the image pixel level. A wavelet based partial image encryption is proposed by Hazarika et al. [32]. In this proposed scheme, the digital image is encrypted partially by using discrete wavelet transformation (DWT) technique, pixel shuffling, and changing of pixel values. DWT is used to decompose the image into sub-bands, then only one band (LL) is used for encryption since it contains most of the information. Then chaotic logistic map is used for pixel shuffling and modification. Although DWT retains the most of the quality of the processed image, it adds much computation complexity and needs more processing time comparing with other techniques such as DCT [33].

3 The Proposed Encryption Scheme

Most of the cryptographic algorithms for securing images are based on the combination of value change and pixel shuffling. In this paper, we propose a novel lightweight and efficient chaotic-based image encryption scheme suitable for wireless network applications. The proposed scheme has two phases: Compression and Encryption. We first utilize the benefits of image compression technique using the DCT transformation, to reduce image size and speed up the process. Then, we use four stages of image encryption based on 2D Logistic map and 2D Henon map to increase the security level of the proposed scheme. In the first step of the encryption phase, we generate a secret key to be used in the other stages of image encryption: Permutation, Diffusion, and Substitution. Figure 3 shows the block diagram of our proposed image encryption scheme.

3.1 Image Compression

In order to speed up the performance time of the proposed encryption scheme and enhance its security efficiency, we choose to use DCT technique for image compression because of its desirable features. For example, DCT has better compression ratio and needs less computation processing capability comparing with other compression techniques, especially when using its approximation implementation [34, 35]. In addition, DCT transforms the image into frequency domain rather than RGB and hence removes the correlations among the neighboring pixels, hence increasing the security efficiency of the proposed encryption scheme. Moreover, comparing with other techniques [9], DCT scheme reduces the blocking effects due to DCT homogenization extension

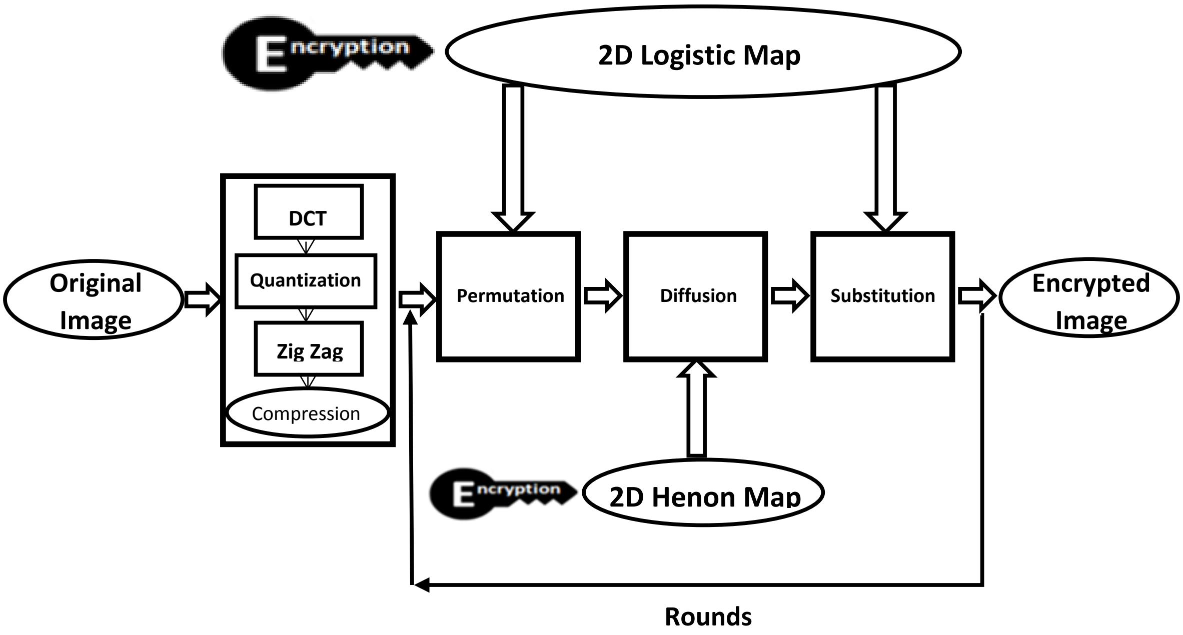


Fig. 3: The Block Diagram of the Proposed Image Encryption Scheme.

properties and because it uses real number based calculations. This makes DCT algorithm much simpler, faster, and more commonly used in standard digital image compression schemes such as JPEG, MPEG, H.261.

DCT belongs to a family of 16 trigonometric transformations and it is used here to change the representation of image pixel information to make the data more amenable to compression. The type-2 DCT (or 2D DCT) transforms the spatial representation of pixel information into to a spatial-frequency domain. An image block of size $M \times N$ whose pixel intensities $f(x,y)$ is transformed into a block of coefficients $F(u,v)$, according to the following equation:

$$F(u,v) = \frac{2}{\sqrt{M \times N}} C_u C_v \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \\ cos\left(\frac{\Pi(2y+1)v}{2N}\right) cos\left(\frac{\Pi(2x+1)u}{2M}\right) \quad (2)$$

Where x & u = 0,1, ..., M-1 and y & v = 0,1, ..., N-1

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & , u = 0 \\ 1 & , otherwise \end{cases} \quad \text{and} \quad C_v = \begin{cases} \frac{1}{\sqrt{2}} & , v = 0 \\ 1 & , otherwise \end{cases}$$

We use Equation 2 to convert a block in the uncompressed original image to matrix $F(u,v)$, which has the same size of the block of the original image $M \times N$, and here the two conversion parameters u and v point to the spatial frequencies. The DCT method is used for large compression applications and it is computed usually by taking 8×8 blocks from the original image (gray scale or from each color plane), as shown in Figure 4.

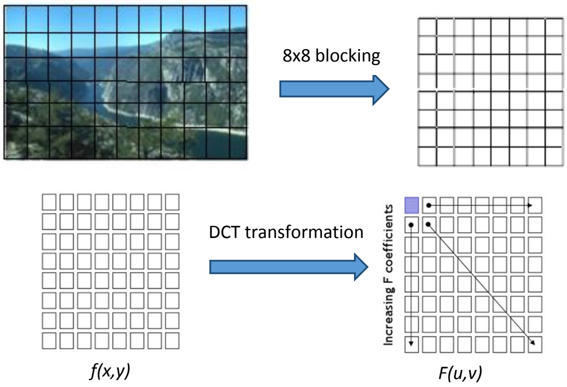
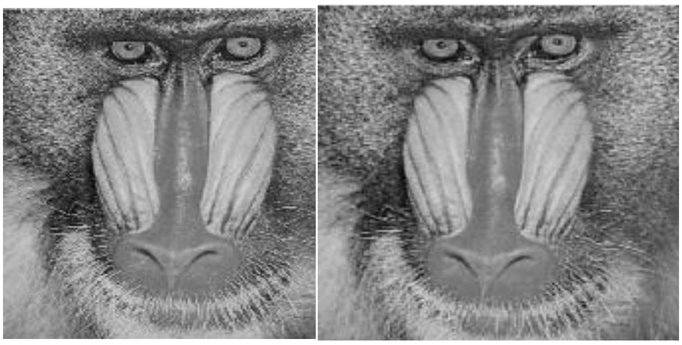


Fig. 4: DCT Concept.

Having applied the DCT transformation over an 8×8 block, there are 64 coefficients: the DC term and low AC frequencies, which are the most important, being in the upper left corner followed by the remaining high AC coefficients, which are less important to the human visual system. Next, a quantization step is used to retain the important coefficients and discard others. There are many quantization tables and values that can be used based on the desired quality and compression ratio. Figure 5 shows the original image



(a)

(b)

Size Original	Size Compress	Comp. DCT Ratio	MSE DCT	PSNR DCT
256.5918	29.3906	8.7304	135.5297	26.8105

(c)

Fig. 5: a) The Original Image, b) The Compressed Image, c) Image Information.

Secret Key (*K*) =

x_0	y_0	r	T	R
52bit	52bit	52bit	52bit	48bit

Fig. 6: Secret Key Format.

before and after compression. The naked eye cannot notice the difference, even though the compressed image size is less by 8 times. This because we preserve the basic structure and important information.

3.2 Secret Key Sequence Generator

Our proposed security scheme uses symmetric secret keys for image encryption and authentication. Thus, only the authorized nodes that share the same secret keys can exchange and reveal the original images correctly. Here, the secret key generation is based on the 2D Logistic map and Henon map because of their many attractive security features mentioned earlier. One of them is the high key sensitivity to chaotic map parameters and initial conditions changes. It also provides a large key space, to make it difficult to be broken by different attacks and to resist against brute-force attack [36]. The size of the secret key (K) is 256 bit and it is divided into five parts: x_0, y_0, r, T, and R, as shown in Figure 6. Notice that $(x_0, y_0) \in (0,1)$ are the initial states of 2D Logistic map and Henon map, and $r \in (1.11,1.19)$ is the system parameter of 2D Logistic map.

In order to increase the security level of our proposed scheme, information from the original image $f(i, j)$ is also used in generating the secret key. Therefore, we merge the control parameter T and the iteration number R with the secret key, and they are calculated as follows:

$$T = (\frac{Sum}{L}) \times x_0 \quad mod \quad 1 \tag{3}$$

$$R = \left(\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)\ mod \quad 3.1\right) + 2 \tag{4}$$

Where $Sum = \sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)$ and L is a system constant. The fraction decimal numbers of x_0, y_0, r, and T are converted to binary format of size 52 bit per each number, and R is represented in 48 bit number. After finding the parameters T and R, initial states can be found in each iteration by using the following equations:

$$X_0^i = (T + x_0 \times i)\ mod \quad 1 \tag{5}$$

$$Y_0^i = (T + y_0 \times i)\ mod \quad 1 \tag{6}$$

$$r^i = (T \times i \times r)\ mod\ 0.09 + 1.11 \tag{7}$$

Where X_0^i, Y_0^i and r^i are the initial values and parameters of 2D Logistic/Henon map in the i^{th} iteration, $1 \le i \le R$.

Now, the pseudo random number sequences generated by the 2D logistic map are controlled by the secret key and ready to be used in the next stages of image encryption. Notice that the same secret key should be used at the receiver side to decrypt the image and retrieve the original information. The generation procedure of the secret key (K) is shown in Algorithm 1.

Algorithm 1: The generation of the secret key *K*

```
Input: Initial states ($x_0$, $y_0$, $r$, $T$, $R$).
Output: Secret key $K$ with length of 256 bits.
$x_0 = \left(\sum_{i=1}^{52} K[i\,]x\,2^{i-1}\right) / 2^{52}$;
$y_0 = \left(\sum_{i=53}^{104} K[i\,]x\,2^{i-53}\right) / 2^{52}$;
$r = \left(\left(\sum_{i=105}^{156} K[i\,]x\,2^{i-105}\right) / 2^{52}\right)$;
$T = \left(\sum_{i=157}^{208} K[i\,]x\,2^{i-157}\right) / 2^{52}$;
$R = \left(\sum_{i=209}^{256} K[i\,]x\,2^{i-209}\right) / 2^{48}$;
for $i$ = 1 to $R$ do
    find $x_0^i$ by applying equation (5)
    find $y_0^i$ by applying equation (6)
    find $r^i$by applying equation (7)
end for
```

Algorithm 2: Image Permutation Process

```
Input: 2D Compressed Image $CI$, column permutation
       matrix $X_{seq}$, and row permutation matrix $Y_{seq}$
Output: Permuted Image $PI$
1. for $c = 1$ to M do
2.     $PCI_{r,c} = CI_{r,c}$ permute with $X_{seq}$
       (Permuted Column Image along x axis)
3. end for
4. for $r = 1$ to N do
5.     $PI_{r,c} = PCI_{r,c}$ permute with $Y_{seq}$
       (row permutation along y axis)
6. end for
```

3.3 Image Permutation Process

In order to make it harder for any hacker to predict image pixel values, we break the high correlation between the adjacent pixels by using the image permutation process . In this permutation stage, we only shuffle the position of rows and columns of the compressed image in order to reduce the computational overhead, by which any pixel can be moved anywhere in the compressed image in a random way. This is done by using 2D Logistic map with the established key to generate two random sequences: X_{seq}, and Y_{seq}. Where X_{seq} is the x coordinate random sequence of the 2D Logistic map with size of M (image width), and Y_{seq} is the y coordinate of the generated random sequence with size N (image height). X_{seq} and Y_{seq} are then converted to unique integer numbers:

$$X_{seq} = (X_1, X_2, ..., X_M) \ mod \ M + 1 \tag{8}$$

$$Y_{seq} = (Y_1, Y_2, ..., Y_N) \ mod \ N + 1 \tag{9}$$

If we assume that the size of the compressed image is $M \times N$, where N and M are the number of pixel rows and columns respectively, then we use Y_{seq} to sort the rows of the compressed image and X_{seq} to shuffle the compressed image columns, as explained in Algorithm 2:

3.4 Image Diffusion Process

In order to further increase the security level of our proposed image encryption scheme, we implement an image diffusion process, in which the image pixel values are changed. We employ 2D Henon chaotic map in the diffusion stage because it gives a strong encryption system with the use of a secret key that is composed of the initial values and system parameters of the chaotic map. Without knowing this secret key, it will be very hard to get the plain image pixels from the encrypted values, which appear to be random numbers. Henon chaotic map introduces discrete time dynamical and 2D invertible map represented with an attractor. We use this map to generate a random sequence which is used to encrypt the shuffled image pixels from the permutation stage. The 2D Henon map is defined as follows:

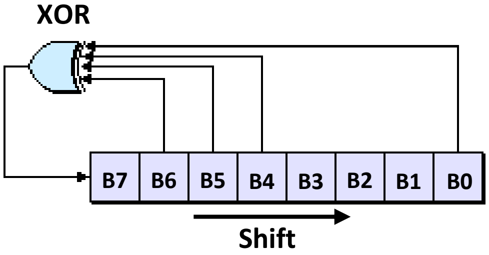


Fig. 7: 8-bit LFBSR.

$$X_{i+1} = 1 - a \times X_i^2 + Y_i \quad and \quad Y_{i+1} = b \times Y_i \tag{10}$$

where $i = 0, 1, 2, \ldots$ and (x_0, y_0) are the initial condition values. (a, b) are the system parameters with 1.4 and 0.3 values respectively for having chaotic behavior. We generate two pseudo random numbers (X) and (Y) of $S \times S$ matrix, where S is the size of processing image block and depends on the image format (equal to 4 if the image is gray-scale or RGB color, 32 if is binary image). These two matrices are then multiplied to construct a one matrix (Z). Then, by using Equation 11, the decimal random number values of Z matrix are converted to integer numbers $(K1)$ in range (0-255) and then translate them to binary representation of 8 bits each.

$$K1_{ij} = \lfloor Z_{ij} \times 10^{10} \rfloor \ mod \ 256 \tag{11}$$

Then, we generate K_2 matrix of $S \times S$ size using 8-bit Linear Feedback Shift Register (LFBSR) with an agreed initial seed like 10111101, as shown in Figure 7. Afterword, we do bit-by-bit XOR operation between K_1 and K_2 to get the final random sequence (W) matrix.

$$W_{ij} = K1_{ij} \oplus K2_{ij} \tag{12}$$

Finally, $S \times S$ block of the gray-scale permuted image is masked by the generated random sequence by XOR operation to generate the diffused image (DI). This process is repeated with all image blocks and iterated again for several times.

$$DI_{ij} = PI\,(i, j) \oplus W_{ij} \tag{13}$$

If the digital image is colored, then this diffusion process is applied on each image plane (RGB). If the size of the permuted image $M \times N$ is not divisible by S×S processing block size, then we apply this diffusion process with respect to the region $S\lfloor M/S \rfloor \times S\lfloor N/S \rfloor$.

3.5 Image Pixel Substitution

In order to strengthen the encryption property of our proposed scheme and diffuse the correlation between plain and cipher image, we add another process called the image pixel substitution. In the image pixel substitution stage, we changes the pixel values randomly, based on the reference matrix (I) that depends on the two random sequences generated in the permutation stage using the 2D Logistic map, X_{seq} and Y_{seq}. These two random sequences are added together to create a one random sequence, $Z_{seq} = X_{seq} + Y_{seq}$. We arrange the random numbers of Z_{seq} in a matrix Z, and then each $S \times S$ block of Z matrix is converted to integer random numbers using block function I, as explained in Equation 14, where g_N, g_R, g_S, g_D are defined in Equation 15.

$$I = \begin{bmatrix} g_N(S_{1,1}) & g_R(S_{1,2}) & g_S(S_{1,3}) & g_D(S_{1,4}) \\ g_R(S_{2,1}) & g_S(S_{2,2}) & g_D(S_{2,3}) & g_N(S_{2,4}) \\ g_S(S_{3,1}) & g_D(S_{3,2}) & g_N(S_{3,3}) & g_R(S_{3,4}) \\ g_D(S_{4,1}) & g_N(S_{4,2}) & g_R(S_{4,3}) & g_S(S_{4,4}) \end{bmatrix} \tag{14}$$

$$\begin{aligned} g_N(d) &= \tau(2d)\ mod\ F \\ g_R(d) &= \tau\left(\sqrt[3]{d}\right)\ mod\ F \\ g_S(d) &= \tau\left(d^3\right)\ mod\ F \\ g_D(d) &= \tau(4d)\ mod\ F \end{aligned} \tag{15}$$

Where $\tau(d)$ converts a decimal number to integer number by truncating 8 digits from the fraction number, and F represents the number of intensity scales of the plaintext image (F=2 if it is binary, F=256 if it is 8-bit gray-scale).

Now, substitution process can start by shifting each pixel in the diffused image with the value of the random number from matrix I over the integer space of [0, $F-1$] to obtain the substituted image (SI), as shown in Equation 16:

$$SI = (DI + I)\ mod\ F \tag{16}$$

4 Simulation Results and Performance Evaluation

To evaluate the performance of our proposed image encryption scheme and check its security efficiency, we conduct several simulation experiments using the Matlab platform. We use Matlab R2012b under Windows 8 professional on machine with Intel(R) Core(TM) i7 1.6 GHz and 4 GB RAM. We use in the simulation different standard test images from CMU and USC-SIPI databases in order to compare our results with other recently proposed algorithms [26, 27, 30, 37–39].

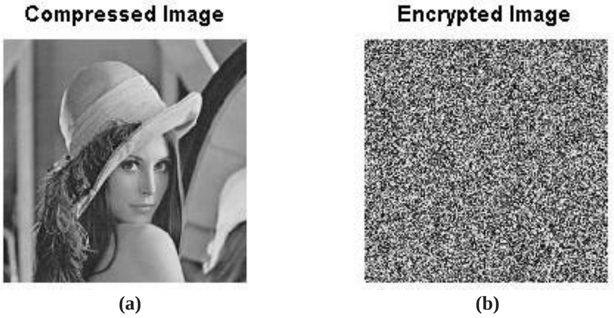


Fig. 8: a) Compressed plain Image, b) Cipher Image.

Simulation measurements show that our proposed scheme encrypts the images to random-like cipher images, as shown in Figure 8, using the chosen secret key and resists against most of the cryptanalytic attacks. Moreover, experimental results show that our proposed scheme outperforms other existing chaotic image encryption schemes under various security analysis. More specifically, we test our algorithm using different experimental and security analysis tools such as: statistical analysis (histogram and correlation of adjacent pixels), key space and key sensitivity, information entropy test, and differential attack analysis.

4.1 Histogram Analysis

One of the straightforward methods for testing the encryption strength of the proposed security scheme is the Histogram Analysis that measures the resistance of the encrypted image against statistical attacks. A good secure image encryption system should encrypt the image to random-like cipher image with uniformly distributed pixel values. The histogram graph shows the relationship between the pixel intensity values and their number of occurrences, or in other words, how the pixels are distributed in the image. We calculate the histogram of both plain compressed images and encrypted images by our proposed encryption scheme, as shown in Figure 9. As seen from the histogram figures, the histogram of the plain images has tilted structure similar to the original images, but the histogram of the encrypted images is flat and has no clue about original images. Hence, statistical cryptanalytic attacks using histogram information are impossible.

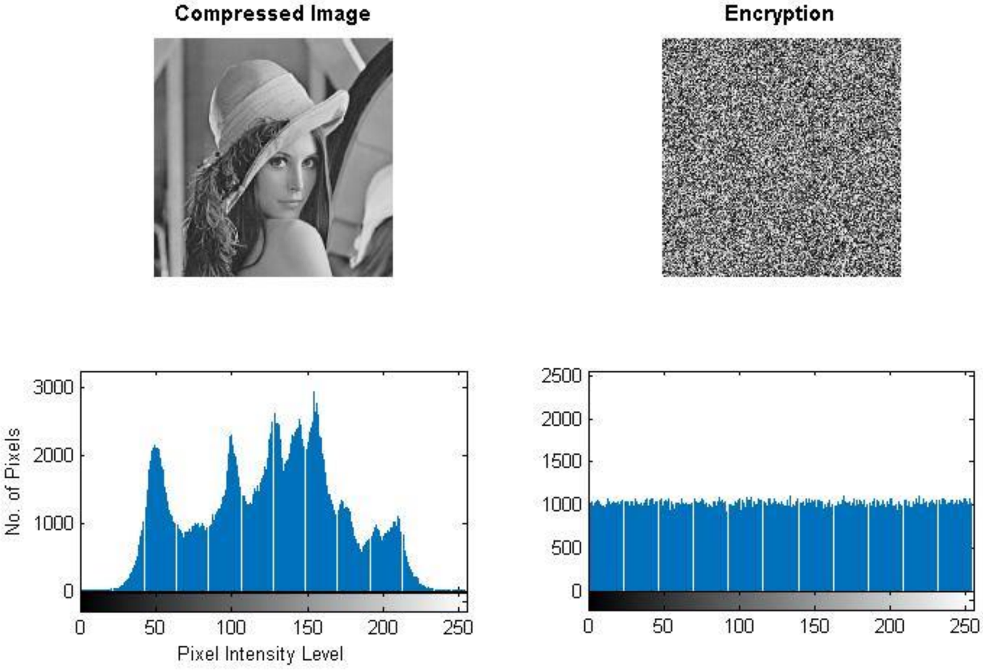


Fig. 9: The Histogram Analysis for Lina Image.

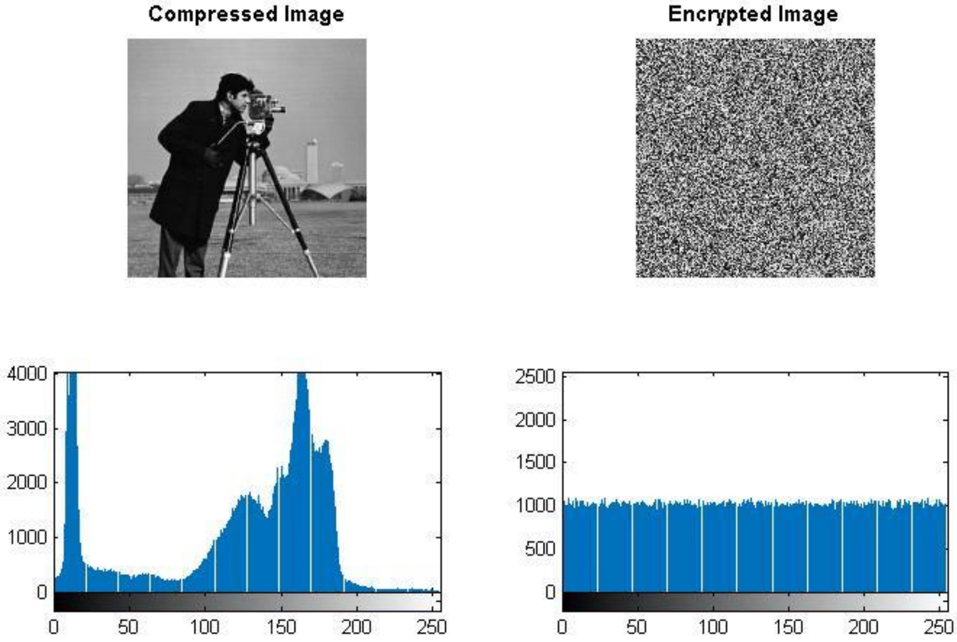


Fig. 10: The Histogram Analysis for Camera Man Image.

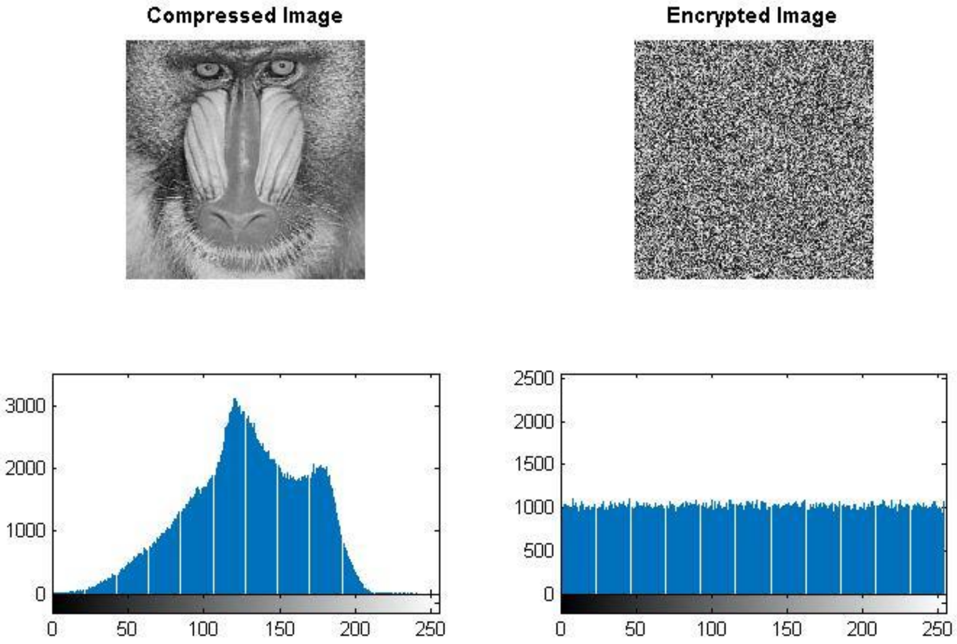


Fig. 11: The Histogram Analysis for Baboon Image.

4.2 Correlation of Adjacent Pixels

One of the characteristics of a plain digital image is the high information redundancy. Because of that there are high spatial correlations between adjacent pixels at horizontal, vertical and diagonal directions. We already utilized the high information redundancy in the image in the compression phase to reduce its size. Furthermore, a good image encryption algorithm should break these pixel correlations in all directions in the encrypted image in order to resist against statistical attacks. The correlation coefficient is calculated by the following equations:

$$E\left(X\right) = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{17}$$

$$D\left(X\right) = \frac{1}{N}\sum_{i=1}^{N} \left(X_i - E\left(X\right)\right)^2$$

$$cov\left(X,Y\right) = \frac{1}{N}\sum_{i=1}^{N} \left(\left(X_i - E\left(X\right)\right)\left(Y_i - E\left(Y\right)\right)\right)$$

$$r_{xy} = \frac{cov\left(X,Y\right)}{\sqrt{D\left(X\right)} \times \sqrt{D\left(Y\right)}}$$

Where X and Y represent the two adjacent pixel values, N is number of pixels, $E(X)$ and $D(X)$ represent the expectation and variable variance, and r_{XY} is the correlation coefficient. We use this analysis to show the correlation among randomly selected pairs of pixels in both plain compressed image and cipher image. The analysis is carried out by following the equations in 4.2 on randomly selected 3,000 pairs of adjacent pixels in the horizontal, vertical, and diagonal directions. Figure 12 shows the correlation distribution of the adjacent pixels in the three directions of the Lena image before and after encryption using our proposed scheme, where the *x-axis* represents the intensity value of randomly-selected pixel and *y-axis* represents the intensity value of the corresponding adjacent pixel. Similar results were given, shown numerically in Table 1, using other test images.

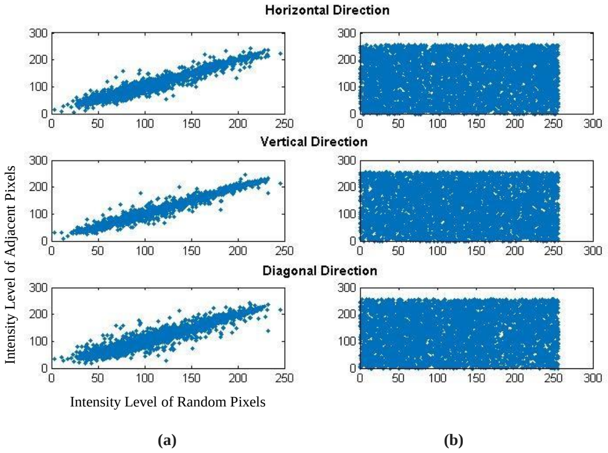


Fig. 12: Correlation Analysis Test for Lina Image: a) Plain, b) Encrypted.

Table 1: Numerical Results of Correlation Analysis of Different Test Images.

(a) Plain Images

Image name	Rxy_V	Rxy_H	Rxy_D
Baboon_gray	0.9039	0.8948	0.8214
Lena_gray_256	0.9855	0.9692	0.9539
Lena_gray_512	0.9914	0.9827	0.9734
Camera_Man	0.9918	0.9869	0.9784

(b) Encrypted Images

Image name	Rxy_V	Rxy_H	Rxy_D
Baboon_gray	-0.0054	-0.0059	-0.0068
Lena_gray_256	-0.0052	0.0132	-0.0124
Lena_gray_512	-3.0213e-04	0.0053	0.0032
Camera_Man	0.0339	0.0002	0.0124

Table 2: Comparison Results of Correlation Analysis.

Encryption Scheme	Rxy_V	Rxy_H	Rxy_D
Proposed Scheme	-0.00032	0.0053	0.0032
YeterSekertekin et al. [26]	0.0022	0.0042	-0.0041
M.Majid et al. [30]	0.0065	0.0129	0.0013
Gopalakrishnan et al. [37]	0.0023	-0.0021	0.0132
Yushu Zhang et al. [38]	0.0011	0.0018	-0.0012

Table 3: Entropy Test of Different Test Images.

Image name	Entropy_plain	Entropy_Encrypted
Baboon	7.6030	7.9993
Lena_gray_256	7.4522	7.9992
Lena_gray_512	7.4533	7.9994
Camera man	7.0880	7.9993

Table 4: Comparison Results of Entropy Analysis.

Encryption Scheme	Entropy
Proposed Scheme	7.9994
Gopalakrishnan et al. [37]	7.9972
M. Majid et al. [30]	7.9973
Yushu Zhang et al. [38]	7.9994
BehrouzFathi et al. [39]	7.9992

As seen from Figure 12, it is clear that the adjacent pixels correlation of the Lena encrypted image is totally broken in all directions, which makes it hard to predict by statistical attacks. This means that the distribution of the adjacent pixels of the ciphered image using our proposed scheme is highly dispersed and has a very low correlation with the plain image. Table 2 shows the comparison results with other proposed algorithms for Lena image. From the comparison table, we can notice that our scheme has very good security efficiency - outperforms in many cases- compared with other recent works but with a fast and lightweight implementation.

4.3 Information Entropy Test

In information theory, Entropy is a measure of disorder/unpredictability of a source of information, and hence, this measure can be used to test the randomness of pixel values in an encrypted image. Image entropy test, $H(S)$, is defined as follows (in bits):

$$H\left(S\right) = \sum_{i=1}^{m} P\left(X_i\right) log_2 \frac{1}{P\left(X_i\right)} \tag{18}$$

where m is the number of symbols (pixel gray levels) emitted by the source, and $P(Xi)$ represents the probability of symbol X_i. A good image encryption scheme should obtain a maximum value of $H(S)$ when all the symbols (gray levels) are uniformly distributed, i.e. have equal probability. Thus, when the entropy test gives less than the maximum, symbol values can be predictable and may raise security issue. If we assume that a pixel value is represented by 8 bits, then the number of gray levels are $2^8 = 256$ and maximum entropy is $H(S) = 8$. Table 3 shows the entropy values of different test images before and after encrypting them using our proposed security scheme:

To measure the image entropy value, we use the block entropy test explained in [40], where 100 non-overlapped 16×16 blocks are randomly selected from each encrypted image. Then, the information block entropy is calculated using Equation 18, and the final entropy value is the average calculated block entropy. Table 4 shows comparisons of the Entropy analysis with other proposed algorithms. The Entropy result of our scheme has a very good value and better than most of other performances.

4.4 Differential Attack Analysis

In order to test the security performance of our proposed image encryption scheme against differential attacks, we use two additional analysis, the Number of Pixel Changing Rate (NPCR) and the Unified Averaged Changed Intensity (UACI) [41]. NPCR tests between two cipher images, C_1 and C_2, whose plain images are slightly different. Also, UACI measures between the two cipher images, C_1 and C_2. The difference between the two tests is that NPCR focuses on the absolute number of changing pixels values in differential attacks, while UACI concentrates on the average difference between the two cipher images, C_1 and C_2 [41]. NPCR and UACI are

Table 5: NPCR and UACI Analysis.

Image name	NPCR	UACI
Baboon	0.9963	0.3369
Lena_gray_256	0.9966	0.3356
Lena_gray_512	0.9964	0.3345
Camera man	0.9963	0.3334

Table 6: Comparison Results for Differential Attack Analysis.

Encryption Scheme	NPCR	UACI
Proposed Scheme	0.9966	0.3356
Gopalakrishnan et al. [37]	0.9961	0.3340
M.Majid et al. [30]	0.9968	0.3340
Yushu Zhang et al. [38]	0.9962	0.3343
BehrouzFathi et al. [39]	0.9962	0.3356

calculated using the following equations:

$$D(i,j) = \begin{cases} 0 \ , \ C_1(i,j) = C_2(i,j) \\ 1 \ , \ C_1(i,j) \neq C_2(i,j) \end{cases}$$

$$NPCR = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{D(i,j)}{B} \times 100\% \qquad (19)$$

$$UACI = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{L \times B} \times 100\% \qquad (20)$$

Where $D(i,j)$ is the difference function and denotes whether two pixels from the two cipher images C_1 and C_2 at the same location are equal or not. B is the number of pixels in the cipher image (i.e. $M \times N$) and P is the largest allowed pixel intensity (i.e. 255 in 8-bit pixel value).

A good image encryption scheme should be sensitive to any change in the plain image (even a 1 bit) and should achieve a NPCR value of approximately 99% and UACI value above 15%. Table 5 lists the NPCR and UACI scores of different encrypted test images using our proposed scheme. As seen from Table 5, our scheme satisfies the both performance metrics and can resist against differential attacks. Table 6 shows comparison results with other proposed algorithms for differential attack analysis using 256 × 256 Lena test image. It is clearly shown that the proposed algorithm is more sensitive to any change in the plain image and outperforms the other proposed algorithms with one or both NPCR and UACI metrics.

4.5 Key Space Analysis

In order to protect the exchanged images against brute-force attacks, which exhausts all the possible keys until finding the correct one, it is reported in [36] that the size of secret key used in encryption process should be greater than 100 bits. It means that a good image encryption scheme should have a key space (the total number of different secret keys) larger than 2^{100}. It is clear that our proposed image encryption scheme satisfies this security requirement as the employed secret key has the size of 256 bits. This is because our secret key are composed of 5 parts, as explained before in Figure 6: The two initial condition values x_0 and Y_0, the system parameter r, the control parameter T, and the number of rounds R. As a result, the secret key used in our proposed scheme is large enough to resist against brute-force attacks.

4.6 Key Sensitivity Analysis

Key sensitivity analysis means that how our system is sensitive to any little change in the secret key, and measures the consequences of using a different key with only one bit in the encryption or decryption process. In the encryption process, using two secret keys with only one different bit should yield two completely different cipher images of a same plain image. For testing the key sensitivity of our proposed image encryption scheme, we follow these steps:

1. First, we use two slightly different keys, K_1 and K_2 (the least significant bit is only changed in K_2):
   K_1 = DE 8E 59 CA 76 B1 3F CD 19 E9 8B 1E 81 12 DB B5 97 4A 8D AA EA 2D B6 1C 08 56 38 E2 99 34 36 85
   K_2 = DE 8E 59 CA 76 B1 3F CD 19 E9 8B 1E 81 12 DB B5 97 4A 8D AA EA 2D B6 1C 08 56 38 E2 99 34 36 84
2. Then, we encrypt the original test image (P) using the two secret keys K_1 and K_2 to produce two ciphertext images: $C_1 = Enc(P, K_1)$ and $C_2 = Enc(P, K_2)$.
3. Finally, we compare the two resulted encrypted images, $|C_1 - C_2|$.

Figure 13 shows the key sensitivity analysis for the encryption process using the camera man test image with two secret keys (K_1 and K_2) that are different only in the rightmost bit. Notice that the difference between the two same plain images is zero because they are identical, but the two ciphertext images (C_1 and C_2) produced by the two secret keys are totally different.

Also in the decryption process, using two slightly different keys with only one bit with the same cipher image should produce two totally different decrypted images. A good encryption algorithm scheme can only encrypt/decrypt correctly using the exact intended secret key, otherwise, rubbish data is obtained. For testing the key sensitivity of our proposed scheme in the decryption process, we follow these steps:

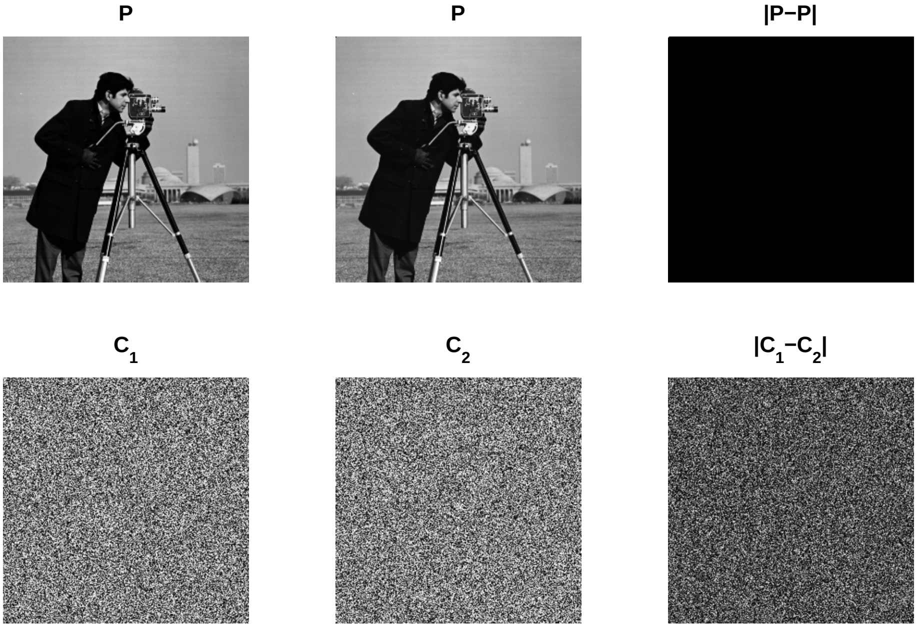


Fig. 13: Key Sensitivity with Only One Bit Different in Encryption Process.

1. First, we encrypt the plain test image (P) using the secret key K_1 to produce the ciphertext image C_1, $C_1 = Enc(P, K_1)$.
2. Then, we decrypt the ciphertext image C_1 using the two slightly different secret keys (K_1 and K_2): $D_1 = Dec(C_1, K_1)$ and $D_2 = Dec(C_1, K_2)$.
3. Finally, we compare the two generated decrypted images, $|D_1 - D_2|$.

Figure 14 shows the key sensitivity analysis of our proposed scheme in the decryption process, where the plain test image is encrypted first using the intended key K_1 to produce the ciphertext image C_1. Then, the ciphertext image is decrypted once by K_1 and another time by K_2. Notice that only the correct exact secret key can decrypt and obtain the original image, otherwise the decryption process fails completely.

4.7 Complexity and Time Analysis

Computation complexity and time analysis are also important measurements of the execution efficiency of our proposed encryption scheme as targeted processing huge-volume data (images) for wireless network applications. Therefore, one of the essential requirements of our proposed image encryption is to be a lightweight and efficient in terms of time consumption and processing complexity. To accomplish this task while satisfying the security requirements for a strong encryption mechanism, we propose the follows:

First, we adopt in our scheme partial image encryption focusing only on the image perceptually sensitive information. This is done by using DCT compression before encryption in order to decrease the image size and speed up the encryption process. Different DCT quantization tables can be used to achieve the desired image quality and compression ratio. Most of existing image encryption techniques use full image encryption without compression. Although they obtain good security measurements with full image quality, their schemes require much time processing and extensive computations. Thus, by using 2D DCT compression, the size of $N \times N$ image is reduced by a factor (F) based on the selected compression ratio to $(N \times N)/F$. For example, a 16:1 compression ratio reduces the image size from N^2 to $N^2/16$.

Second, we use chaotic-based image encryption algorithms in our proposed scheme, mainly 2D Logistic and Henon maps, for their fast computation and strong security properties comparing with other traditional cryptographic techniques such as DES, AES, TwoFish, and BlueFish. These traditional block-wise/one-dimensional bit stream ciphers sacrifice the two-dimensional nature of the image and add noticeable processing complexity dealing with the high information redundancy [40].

Finally, we use 2D Logistic map in the permutation stage to shuffle only pixel's rows and columns, not every pixel in the image to further reduce the computational overhead.

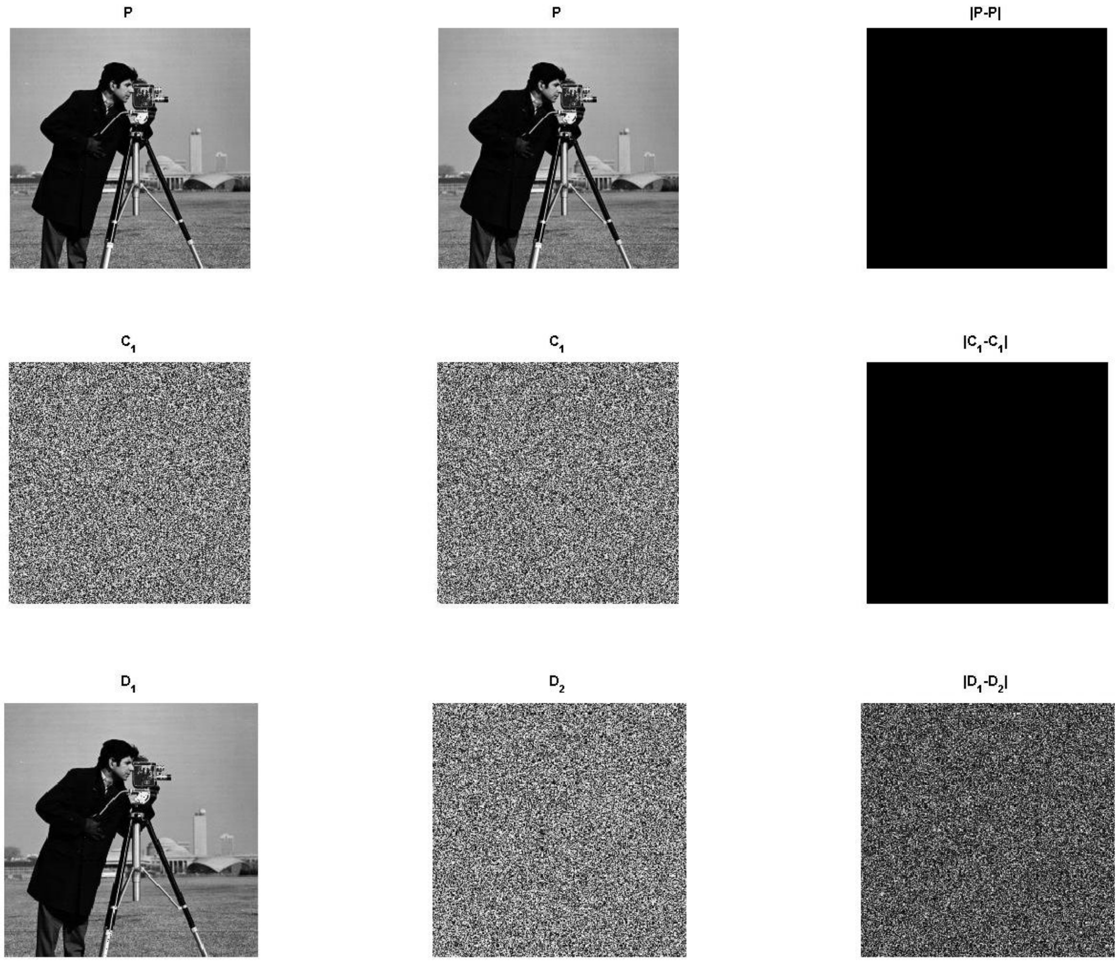


Fig. 14: Key Sensitivity with Only One Bit Different in Decryption Process.

Shuffling only the image's rows and columns reduces the computation complexity from $N \times N$ to $N+N$.

In our proposed algorithm, the computation complexity equals to $N^2 logN$ for the 2D DCT compression, $2N^2/F$ for the key generation, $2N/F$ for the permutation process, $8N^2/F$ for the diffusion stage, and N^2/F for the substitution process. Table 7 lists the time analysis comparisons with other encryption schemes for encrypting a 512×512 test image. However, the actual execution time performance of each scheme depends on many factor such as: the number of iterations, the type of the programing approach and the coding efficiency, as well as the platform specifications.

Table 7: Time Analysis Comparisons.

Encryption Scheme	Enc. Time (sec)
Proposed Scheme	0.1532
Hazarika et al. [32]	15.2563
Yashasvee et al. [24]	130.6902
Gopalakrishnan et al. [27]	0.2428

5 Conclusions

This study proposes a lightweight and efficient digital image security scheme based on chaotic theory for wireless networks. Our proposed scheme encrypts efficiently the images and resists against most attack types, and adds minimal impact on node computational overhead and overall network communication performance. The proposed scheme is implemented in two phases: image compression, and image encryption. In the compression phase, images are compressed to reduce their sizes using DCT transformation by utilizing the high redundancy in pixel information. The advantage of using the compression phase is to speed up and maximize the performance of the encryption process, to effectively deal with the node and network limited resources, and to transfer images rapidly. In the encryption phase, images are secured in four stages using hybrid chaotic maps.

Firstly, a synchronous secret key is generated of 256 bit long to provide a large key space and resist against most attack types. Secondly, in the image permutation, the image pixels are shuffled randomly using 2D Logistic map with the shared secret key. Thirdly, in the image diffusion, the image pixels are altered using 2D Henon map controlled with the secret key. Finally, in the image substitution, the image pixels are shifted randomly by using 2D Logistic map. These encryption steps are repeated in many rounds to obtain better confusion and diffusion properties and get strong cipher images. Simulation results and security analysis confirm that our proposed chaotic image security scheme provides random-like and complex cipher images satisfying the needed security requirements, and resisting against most existing cryptanalysis and cryptography attacks such as statistical attacks and differential attacks. Moreover, the security level provided by our scheme has been examined using various test images under different security analysis, such as statistical analysis (Histogram and correlation of adjacent pixels), key space and key sensitivity, information entropy test, and differential attack analysis.

Acknowledgement

This research paper was supported and funded by the Hashemite University, Zarqa – Jordan.

References

1. I. T. Almalkawi, M. Guerrero Zapata, J. N. Al-Karaki, and J. Morillo-Pozo, "Wireless multimedia sensor networks: Current trends and future directions," *Sensors*, vol. 10, no. 7, pp. 6662–6717, 2010.
2. C. Li, S. Li, M. Asim, J. Nunez, G. Álvarez, and G. Chen, "On the security defects of an image encryption scheme," *Image and Vision Computing*, vol. 27, no. 9, pp. 1371–1381, 2016.
3. E. Setyaningsih and R. Wardoyo, "Review of image compression and encryption techniques," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, 2017.
4. T. Ma, M. Hempel, D. Peng, and H. Sharif, "A survey of energy-efficient compression and communication techniques for multimedia in resource constrained systems," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 963–972, 2013.
5. W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.
6. G. K. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, 1992.
7. A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727 – 752, 2010.
8. M. Kramm, "Compression of image clusters using Karhunen Loeve transformations," in *Human Vision and Electronic Imaging XII*, vol. 6492, Feb. 2007, p. 64920G.
9. A. Katharotiya, S. Patel, and M. Goyani, "Comparative analysis between dct and dwt techniques of image compression," *Journal of Information Engineering and Applications*, vol. 1, no. 1, pp. 9 – 17, 2011.
10. W. Stallings, *Cryptography and Network Security: Principles and Practice*, 5th ed. Prentice Hall Press, 2010.
11. A. Soleymani, Z. M. Ali, and M. J. Nordin, "A survey on principal aspects of secure image transmission," in *Proceedings of World Academy of Science, Engineering and Technology*, 2012, p. 247–254.
12. S. Draksharam, D. Katravulapalli, K. R. Krishna, and V. Thanikaiselvan, "Analysis of hybrid chaotic image encryption," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 697–703.
13. P. R. Sankpal and P. A. Vijaya, "Image encryption using chaotic maps: A survey," in *2014 Fifth International Conference on Signal and Image Processing*, 2014, pp. 102–107.
14. E. M. Assistant, J. A. Joy, and N. A. Vasanthi, "Survey of chaos based image encryption and decryption techniques," in *AICWIC'13 Amrita International Conference of Women in Computing*, 2013.
15. P. Praveenkumar, T. K, J. Bosco Bala, and R. Amirtharajan, "Inbuilt image encryption and steganography security solutions for wireless systems: A survey," *Research Journal of Information Technology*, vol. 9, pp. 46–63, 2017.
16. G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 347 – 354, 2010.
17. D. F. Prunaret and R. L. Ruiz, "Basin bifurcations in a two-dimensional logistic map," *in Eprint-arXiv:nlin/0304059*, 2003.
18. I. T. Almalkawi, J. N. Al-Karaki, A. Alsarhan, R. Abu-Ajamiyah, and D. Al-Mughrabi, "An efficient digital image encryption using pixel shuffling and substitution for wireless networks," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, April 2019, pp. 266–271.

19. M. Henon, “A two-dimensional mapping with a strange attractor,” *Comm. Math. Phys.*, vol. 50, no. 1, pp. 69–77, 1976.
20. A. Bisht, M. Dua, and S. Dua, “A novel approach to encrypt multiple images using multiple chaotic maps and chaotic discrete fractional random transform,” *Journal of Ambient Intelligence and Humanized Computing*, 10 2018.
21. Y. Li, C. Wang, and H. Chen, “A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation,” *Optics and Lasers in Engineering*, vol. 90, pp. 238 – 246, 2017.
22. Y.-Q. Zhang and X.-Y. Wang, “Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation,” *Nonlinear Dynamics*, vol. 77, no. 3, pp. 687–698, 2014.
23. B. Mondal and T. Mandal, “A light weight secure image encryption scheme based on chaos and dna computing,” *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 4, pp. 499 – 504, 2017.
24. Y. Jha, K. Kaur, and C. Pradhan, “Improving image encryption using two-dimensional logistic map and aes,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 0177–0180.
25. A. K. Prusty, A. Pattanaik, and S. Mishra, “An image encryption and decryption approach based on pixel shuffling using arnold cat map and henon map,” in *2013 International Conference on Advanced Computing and Communication Systems*, 2013, pp. 1–6.
26. Y. Sekertekin and O. Atan, “An image encryption algorithm using ikeda and henon chaotic maps,” in *2016 24th Telecommunications Forum (TELFOR)*, 2016, pp. 1–4.
27. T. Gopalakrishnan, S. Ramakrishnan, and M. Balakumar, “An image encryption using chaotic permutation and diffusion,” in *2014 International Conference on Recent Trends in Information Technology*, 2014, pp. 1–5.
28. M. Rani and S. Kumar, “A novel and efficient approach to encrypt images using chaotic logistic map and stream cipher,” in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015, pp. 1442–1447.
29. Q. Kester, L. Nana, A. C. Pascu, and S. Gire, “A new encryption cipher for securing digital images of video surveillance devices using diffie-hellman-md5 algorithm and rgb pixel shuffling,” in *2013 European Modelling Symposium*, 2013, pp. 305–311.
30. M. Majid, S. Amir, H. Moein, and N. Mahboubeh, “An improved method for image encryption based on high level chaotic maps and improved gravity model,” in *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, 2015, pp. 253–259.
31. Y. Xie, J. Li, Z. Kong, Y. Zhang, X. Liao, and Y. Liu, “Exploiting optics chaos for image encryption-then-transmission,” *Journal of Lightwave Technology*, vol. 34, no. 22, pp. 5101–5109, 2016.
32. N. Hazarika, S. Borah, and M. Saikia, “A wavelet based partial image encryption using chaotic logistic map,” in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, 2014, pp. 1471–1475.
33. K. Bindu, A. Ganpati, and A. Kumar Sharma, “A comparative study of image compression algorithms,” *International Journal of Research in Computer Science*, vol. 2, pp. 37–42, 2012.
34. R. Cintra, F. Bayer, and C. Tablada, “Low-complexity 8-point dct approximations based on integer functions,” *Signal Processing*, vol. 99, pp. 201 – 214, 2014.
35. M. Jridi and P. K. Meher, “Scalable approximate dct architectures for efficient hevc-compliant video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1815–1825, 2017.
36. G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” *International Journal of Bifurcation and Chaos*, vol. 16, no. 08, pp. 2129–2151, 2006.
37. T. V. Gopalakrishnan and S. Ramakrishnan, “Image encryption in block-wise with multiple chaotic maps for permutation and diffusion,” *ICTAT Journal on Image and Video Processing*, vol. 6, no. 03, pp. 1220–1227, 2016.
38. Y. Zhang and D. Xiao, “An image encryption scheme based on rotation matrix bit-level permutation and block diffusion,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 1, pp. 74 – 82, 2014.
39. B. Fathi-Vajargah, M. Kanafchian, and V. Alexandrov, “Image encryption based on permutation and substitution using clifford chaotic system and logistic map,” *ournal of Computers*, vol. 13, no. 3, pp. 309–326, 2018.
40. Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, “Local shannon entropy measure with statistical tests for image randomness,” *Information Sciences*, vol. 222, pp. 323 – 342, 2013.
41. Y. Wu, J. Noonan, and S. Agaian, “*npcr* and *uaci* randomness tests for image encryption,” *in Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications*, vol. 2, pp. 31–38, 2011.