entropy

MDPI

Article

An Algorithm of Image Encryption Using Logistic and Two-Dimensional Chaotic Economic Maps

Sameh S. Askar [1,2,*], Abdel A. Karawia [2,3], Abdulrahman Al-Khedhairi [1] and Fatemah S. Al-Ammar [1]

[1] Department of Statistics and Operations Researches, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia; akhediri@ksu.edu.sa (A.A.-K.); sameh_asker@mans.edu.eg (F.S.A.-A.)

[2] Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt; abibka@mans.edu.eg

[3] Computer Science Unit, Deanship of Educational Services, Qassim University, P.O. Box 6595, Buraidah 51452, Saudi Arabia

* Correspondence: saskar@ksu.edu.sa; Tel.: +966-55-588-3742

Received: 26 November 2018; Accepted: 31 December 2018; Published: 9 January 2019

Abstract: In the literature, there are many image encryption algorithms that have been constructed based on different chaotic maps. However, those algorithms do well in the cryptographic process, but still, some developments need to be made in order to enhance the security level supported by them. This paper introduces a new cryptographic algorithm that depends on a logistic and two-dimensional chaotic economic map. The robustness of the introduced algorithm is shown by implementing it on several types of images. The implementation of the algorithm and its security are partially analyzed using some statistical analyses such as sensitivity to the key space, pixels correlation, the entropy process, and contrast analysis. The results given in this paper and the comparisons performed have led us to decide that the introduced algorithm is characterized by a large space of key security, sensitivity to the secret key, few coefficients of correlation, a high contrast, and accepted information of entropy. In addition, the results obtained in experiments show that our proposed algorithm resists statistical, differential, brute-force, and noise attacks.

Keywords: logistic map; chaotic economic map; image encryption; image decryption; security analysis

MSC: 68P25; 68U10; 94A60

1. Introduction

Converting secret information such as images from a decipherable form to an undecipherable one has been recently investigated to fulfill the demand of a secure route of image transmission through different communications channels such as Internet and wireless networks. Encryption schemes have been introduced based on chaotic models to handle that issue and to provide such secure routes [1,2]. The Data Encryption Standard (DES) algorithm is a traditional scheme and has been used for encrypting images [3]. It has some drawbacks, among which is that the algorithm's efficiency is low when dealing with large images [4,5]. The theory of chaos is an important theory that possesses some complex features such as sensitivity to the initial conditions, unpredictability, and bifurcation. It has been recently adopted by researchers in the process of encryption [1,2]. Chaos has presented some popular models, which include some complex properties that help provide a robust encryption scheme. Such models include the logistic map, Lorenz attractor, the Hénon map, and some chaotic economic systems [6]. Such complex properties of chaotic maps may be reflected in an analogy to certain properties of the cryptographic process of ideal ciphers like diffusion, confusion, balance, and avalanche.

The cipher schemes proposed based on chaotic characteristics have shown a robust and efficient way to tackle image broadcast very quickly with highly secured routes via different types of telecommunications. Since Matthews [7] published his first algorithm of encryption in 1989, which depended on some chaotic characteristics, a number of chaos-based image encryption schemes with have been developed in literature. For instance, in [8], the authors created a cipher image by dividing it into some blocks and used some kind of permutations applied to each block using the logistic map. A two-dimensional discrete chaotic system has been introduced in [9]. It used the Chebyshev chaotic sequences to scramble each pixel in the novel image. Increasing the key space is another aspect that makes the algorithm used for encryption very difficult to break, and this has been performed in [10]. A multi-chaotic system has been adopted in the process of encryption in [10]. In [11], the Rossler system has been included in an encryption algorithm and has been used to change the image's pixel values and position in order to make the ciphered images have high uncertainty. A logistic chaotic function has been used in [12] with the one-time pads in order to increase the encryption space, and hence, robust cipher images can be obtained. The shuffling approach has been introduced in [13], presenting a good scheme of encryption based on the cat map. It is a one-dimensional chaotic map with some advantages such as simplicity and high efficiency [14]. However, it has some weaknesses that may make it possible for the encryption scheme to be easily broken. This weakness includes a small key space and a weak security [15,16].

It is important here to highlight some of the recent and state-of-the-art algorithms in cryptography that will be used in comparison with our proposed algorithm. For example, in [17], Sivakumar et al. introduced a new image encryption algorithm based on pixel position permutation and a random key stream. In [18], the authors combined DNA sequences with a chaotic map to introduce an image algorithm of encryption. Based on discrete wavelet transform and multi-chaos, an image algorithm was proposed in [19]. The Arnold transformation has been used to encrypt images with high security in [20]. The idea in [20] depended on splitting the image randomly into several parts, and then, each part was coded by the Arnold transformation to increase the security level. Adopting the classical confusion-diffusion structure, an image encryption algorithm has been introduced by Hua et al. based on a 2D logistic-sine-coupling map [21]. In [22], a simple piecewise linear chaotic map has been utilized to construct a bit-level image encryption algorithm. Recently, a robust image encryption/decryption algorithm via 1D chaotic economic map was proposed in [1]. The advantage of that algorithm is that the periodic recovery of the plain image is removable.

Some of the disadvantages existing in those algorithms have led us to propose a new 2D chaotic algorithm for image encryption. The algorithm published in [1] is constructed based on a one-dimensional chaotic map, but this one depends on a two-dimensional chaotic system. The number of parameters in the current algorithm is more than that in [1], and therefore, the security level is higher than the one in [1]. The current paper is organized as follows. In Section 2, a brief introduction about the 2D chaotic map used in the algorithm is presented. In Section 3, the algorithm's steps are introduced in detail. In Section 4, we discuss our obtained results due to applying the algorithm on some popular images and make some comparisons with the results obtained by existing algorithms in the literature. In Section 5, we give some conclusions.

2. The Two-Dimensional Map

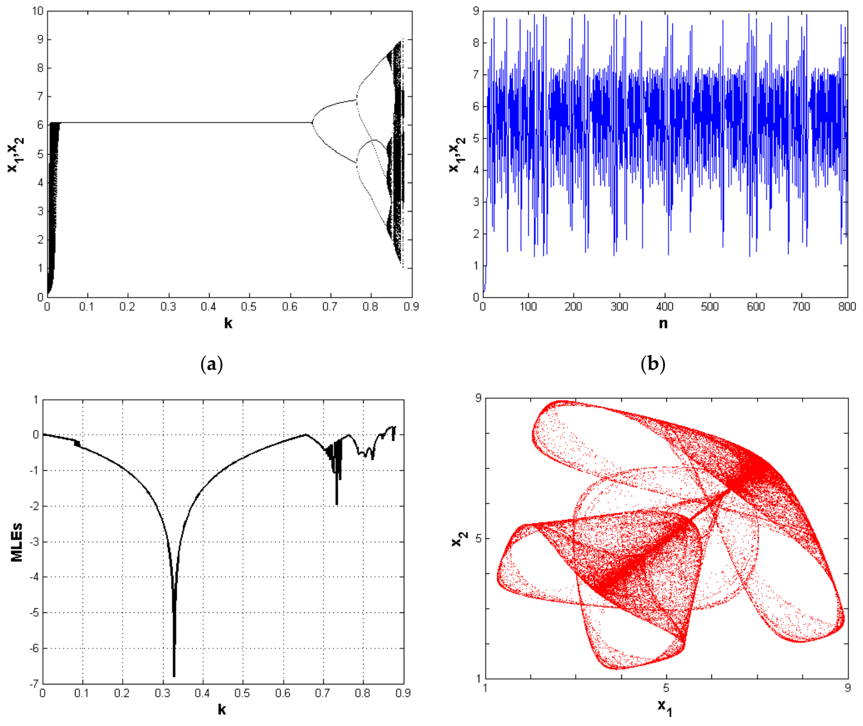
Recently, the adoption of chaotic dynamical systems to encrypt images has drawn researchers' attention. Such systems possess many parameters that have influences on the systems' behaviors and are suitable in the encryption scheme. Those parameters have some important usage, as they are used as security keys in the process of encryption. Small changes in the value of the system's parameters may make the system enter a region where chaos arises. The chaotic behavior of such systems is an important aspect when cryptographic algorithms are constructed. In this work, we adopt the chaotic economic map introduced in [23] that takes the following form:

$$\left.\begin{aligned} x_{1,n+1} &= x_{1,n} + kx_{1,n}^2\left(a - c - b\frac{x_{1,n}}{Q_n} - b\log(Q_n)\right), \\ x_{2,n+1} &= x_{2,n} + kx_{2,n}^2\left(a - c - b\frac{x_{2,n}}{Q_n} - b\log(Q_n)\right), \end{aligned}\right\} \tag{1}$$

where:

$$Q_n = x_{1,n} + x_{2,n}, \quad n = 0, 1, 2, \ldots$$

Equation (1) shows the chaotic behavior of a map with six distinct parameters. These parameters are important in the economy. First, a stands for the market size and should always be positive, while the parameter $b > 0$ represents the change of price within the market. Economically, the condition $a > b$ and $a > c$ should be always satisfied. The parameter $c \geq 0$ is called the cost of producing one unit of a good, and $k > 0$ is known as the speed adjustment. The chaotic behavior of the economic map (1) is detected at the parameter values: $x_{1,0} = 0.11, x_{2,0} = 0.12, a = 2, b = 0.5$, and $c = 0.5$. Figure 1a shows the bifurcation diagram of Equation (1) with respect to the parameter k for these parameters. Numerical simulation has shown that when the system (1) approaches the value $k = 0.6490$, a two-cycle period appears, and so on, for other different cycle periods until the system becomes chaotic. This is clear in the Lyapunov exponents plotted in Figure 1c at the same values of the parameters. Lyapunov exponents tell us the rate of divergence of nearby trajectories, a key component of chaotic dynamics. Figure 1b shows the time series for both variables $x_{1,n}$ and $x_{2,n}$. Finally, Figure 1d shows the complex chaotic behavior of the system (1).



(**a**)

(**b**)

(**c**)

(**d**)

Figure 1. (**a**) Bifurcation diagram with respect to the parameter k at $(x_{1,0}, x_{2,0}, a, b, c) = (0.11, 0.12, 2, 0.5, 0.5)$; (**b**) time series for the system's variables at $(x_{1,0}, x_{2,0}, k, a, b, c) = (0.11, 0.12, 0.87, 2, 0.5, 0.5)$; (**c**) the Lyapunov exponent with respect to the parameter k at $(x_{1,0}, x_{2,0}, a, b, c) = (0.11, 0.12, 2, 0.5, 0.5)$; (**d**) chaotic behavior of the system at the parameters $(x_{1,0}, x_{2,0}, k, a, b, c) = (0.11, 0.12, 0.87, 2, 0.5, 0.5)$.

3. The Algorithm

It is known that any original image has strong correlations among its pixels. In the encryption process, it is important to scatter the pixels to break their correlations. The well-known approach to do that is the shuffling process, which is applied to each array of the original image. Suppose that the original image has the dimension $M \times N$ and $\mathbf{P} = (p_{i,j})_{1 \le i \le M, 1 \le j \le N}$ is the array of its pixel positions. To produce the shuffling array, $\mathbf{Sh}^{rc}_{\mathbf{P}}$, we continue as follows:

1. (Shuffling of rows) In this step, we recall the logistic map $x_{n+1} = rx_n(1 - x_n)$. The logistic map is used to generate a set of random values in the interval $[1, M]$, say $i_1, i_2, ..., i_M,\ i_k \neq i_l\ \forall\ k \neq l$. The rows of the $\mathbf{P}$ array are changed according to those random values, and hence, a new array $\mathbf{Sh}^{r}_{\mathbf{P}}$ is obtained:

$$\mathbf{Sh}^{r}_{\mathbf{P}} = \begin{pmatrix} p_{i_1,1} & p_{i_1,2} & \cdots & \cdots & p_{i_1,N} \\ p_{i_2,1} & p_{i_2,2} & \cdots & \cdots & p_{i_2,N} \\ \vdots & \vdots & \ldots & \ldots & \vdots \\ \vdots & \vdots & \ldots & \ldots & \vdots \\ p_{i_M,1} & p_{i_M,2} & \cdots & \cdots & p_{i_M,N} \end{pmatrix}$$

2. (Shuffling of columns) Using a similar technique as in (1), we generate a new set of random values in the interval $[1, N]$, say $j_1, j_2, ..., j_N,\ j_k \neq j_l\ \forall\ k \neq l$. The columns of $\mathbf{Sh}^{r}_{\mathbf{P}}$ take the following form:

$$\mathbf{Sh}^{rc}_{\mathbf{P}} = \begin{pmatrix} p_{i_1,j_1} & p_{i_1,j_2} & \cdots & \cdots & p_{i_1,j_N} \\ p_{i_2,j_1} & p_{i_2,j_2} & \cdots & \cdots & p_{i_2,j_N} \\ \vdots & \vdots & \ldots & \ldots & \vdots \\ \vdots & \vdots & \ldots & \ldots & \vdots \\ p_{i_M,j_1} & p_{i_M,j_2} & \cdots & \cdots & p_{i_M,j_N} \end{pmatrix}$$

Now, the steps of our encryption algorithm are constructed as follows:

Step 1: Read the original image, then convert it to a gray image, say $\mathbf{P}$.
Step 2: Perform row and column shuffling to generate the shuffling array, say $\mathbf{Sh}^{rc}_{\mathbf{P}}$.
Step 3: Convert the pixel values of $\mathbf{Sh}^{rc}_{\mathbf{P}}$ from decimal to binary, $\mathbf{S} = \{S_1, S_2, \cdots, S_{MN}\}$.
Step 4: Use the two-dimensional chaotic economic map (1) to generate MN values $\mathbf{x} = \{x_1, x_2, \cdots, x_{MN}\}$ as follows:

(i) For $i = 2 : n$, Compute:

$$x_{1,i} = x_{1,(i-1)} + kx^2_{1,(i-1)}\left(a - c - b\frac{x_{1,(i-1)}}{Q_{i-1}} - b\log(Q_{i-1})\right)$$
$$x_{2,i} = x_{2,(i-1)} + kx^2_{2,(i-1)}\left(a - c - b\frac{x_{2,(i-1)}}{Q_{i-1}} - b\log(Q_{i-1})\right).$$

End.

(ii) Set $x_{1,0} = x_{1,n}$ and $x_{2,0} = x_{2,n}$ as initial values.

(iii) Generate MN values $\mathbf{x}_1 = \{x_{11}, x_{12}, \cdots, x_{1,MN}\}$ and $\mathbf{x}_2 = \{x_{21}, x_{22}, \cdots, x_{2,MN}\}$ using the two-dimensional chaotic economic map (1).

Step 5: Do the following preprocessing for the generated values in Step 4:

$$x_i = floor\left(mod\left(x_i \times 10^{14}\right), 256\right)$$

Step 6: Convert the preprocessing values in Step 5 from decimal to binary, $\mathbf{B}_1 = \{b_{11}, b_{12}, \cdots, b_{1,MN}\}$ and $\mathbf{B}_2 = \{b_{21}, b_{22}, \cdots, b_{2,MN}\}$ for $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively.
Step 7: Perform the bit-wise *XOR* between the values of $\mathbf{B}_1$ and $\mathbf{I}$, $\mathbf{e}_1 = bitxor(\mathbf{B}_1, \mathbf{I})$, where $\mathbf{I}$ equals the value of the array $\mathbf{P}$ after reshaping it to be a vector of size $1 \times MN$.
Step 8: Perform the bit-wise *XOR* between the values of $\mathbf{B}_2$ and $\mathbf{I}$, $\mathbf{e}_2 = bitxor(\mathbf{B}_2, \mathbf{I})$.

Step 9: Perform the bit-wise *XOR* between the values of $\mathbf{e}_1$ and $\mathbf{e}_2$, $\mathbf{e} = bitxor(\mathbf{e}_1, \mathbf{e}_2)$.
Step 10: Convert the values of **e** from binary to decimal, say **E**.
Step 11: The cipher pixel set is denoted by $\mathbf{E} = \{E_1, E_2, \cdots, E_{MN}\}$.
Step 12: Reshape the set **E** to be an array of size $M \times N$, say **C** as the cipher image.

In addition, the steps of our decryption algorithm are described as follows:

Step 1: Read the cipher image, **C** with size $M \times N$.
Step 2: Reshape **C** to the cipher pixel set $\mathbf{E} = \{E_1, E_2, \cdots, E_{MN}\}$.
Step 3: Convert the values of **E** from decimal to binary, say **e**.
Step 4: Repeat **Step 4** in the encryption algorithm to generate MN values for the two vectors, $\mathbf{x}_1 = \{x_{11}, x_{12}, \cdots, x_{1,MN}\}$ and $\mathbf{x}_2 = \{x_{21}, x_{22}, \cdots, x_{2,MN}\}$.
Step 5: Do the following preprocessing for the generated values in Step 4:

$$x_i = floor\left(mod\left(x_i \times 10^{14}\right), 256\right)$$

Step 6: Convert the preprocessing values in Step 5 from decimal to binary, $\mathbf{B}_1 = \{b_{11}, b_{12}, \cdots, b_{1,MN}\}$ and $\mathbf{B}_2 = \{b_{21}, b_{22}, \cdots, b_{2,MN}\}$ for $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively.
Step 7: Perform the bit-wise *XOR* between the values of $\mathbf{B}_1$ and **e**, $\mathbf{F}_1 = bitxor(\mathbf{B}_1, \mathbf{e})$.
Step 8: Perform the bit-wise *XOR* between the values of $\mathbf{B}_2$ and **e**, $\mathbf{F}_2 = bitxor(\mathbf{B}_2, \mathbf{e})$
Step 9: Perform the bit-wise *XOR* between the values of $\mathbf{F}_1$ and $\mathbf{F}_2$, $\mathbf{z} = bitxor(\mathbf{F}_1, \mathbf{F}_2)$
Step 10: Convert the values of **z** from binary to decimal, say **Z**.
Step 11: Reshape **Z** to be an array of size $M \times N$, say **Sh**.
Step 12: Perform row and column shuffling to get the decryption image, say **D**.

The Secret Key Generation

Assume that $\mathbf{P} = (p_{i,j})$, $i = 1, 2, ..., M, j = 1, 2, ..., N$, is the plain image, where $p_{i,j}$ refers to the value of the pixel at the position (i, j) and (M, N) is the size of the plain image **P**. The secret key will be calculated via the key mixing proportion factor as follows:

$$K_u = \frac{1}{256} mod\left(\sum_{i=\frac{(u-1)M}{10}+1}^{\frac{uM}{10}} \sum_{j=1}^{N} p_{i,j}, 256\right) \tag{2}$$

Then, we change the initial condition β_0 according to the following formula:

$$\beta_0 \leftarrow \frac{(\beta_0 + K)}{2}, \tag{3}$$

where $\beta_0 = x_{10}, r_{10}, x_{20}, r_{20}, x_{1,0}, x_{2,0}, a, b, c, k$ and $K = K_u, u = 1, 2, \cdots, 10$, respectively.

After that, take two initial values for the logistic map, x_{10}, x_{20}, two parameters for the logistic map, r_{10}, r_{20}, two initial values for the system, $x_{1,0}, x_{2,0}$, and four system parameters, a, b, c, k.

Remark 1. *It is not difficult to note that* $0 < K_u < 1$*. Therefore, the value of* β_0 *on the right-hand side of Equation (3) must satisfy the conditions of Map (1).*

4. Experimental Analysis

In the current section, we apply the introduced algorithm on several types of images to test its robustness. Ten well-known images, lena (128 × 128, 256 × 256, 512 × 512), barbara (256 × 256), airplane (512 × 512), boat (512 × 512), house (256 × 256), baboon (512 × 512), moon surface (256 × 256), and resolution chart (256 × 256) are tested as plain images. The plain images and their corresponding histograms are displayed in the first and second columns in Figure 2.

The security keys of our encryption algorithm are chosen as follows. In the logistic map, $r = 3.9985$, $x_0 = 0.02$ for row shuffling and $r = 3.9995$, $x_0 = 0.3$ for column shuffling. In the two-dimensional chaotic economic map (1), $x_{1,0} = 0.11, x_{2,0} = 0.12, a = 2, b = 0.5, c = 0.5$, and $k = 0.87$. Our experiments are carried out under MATLAB R2016a running on a laptop with the following features: Intel(R) Core(TM) i7-4700MQ 2.40 GHz, 12.0 GB memory, and 1.0 TB capacity.

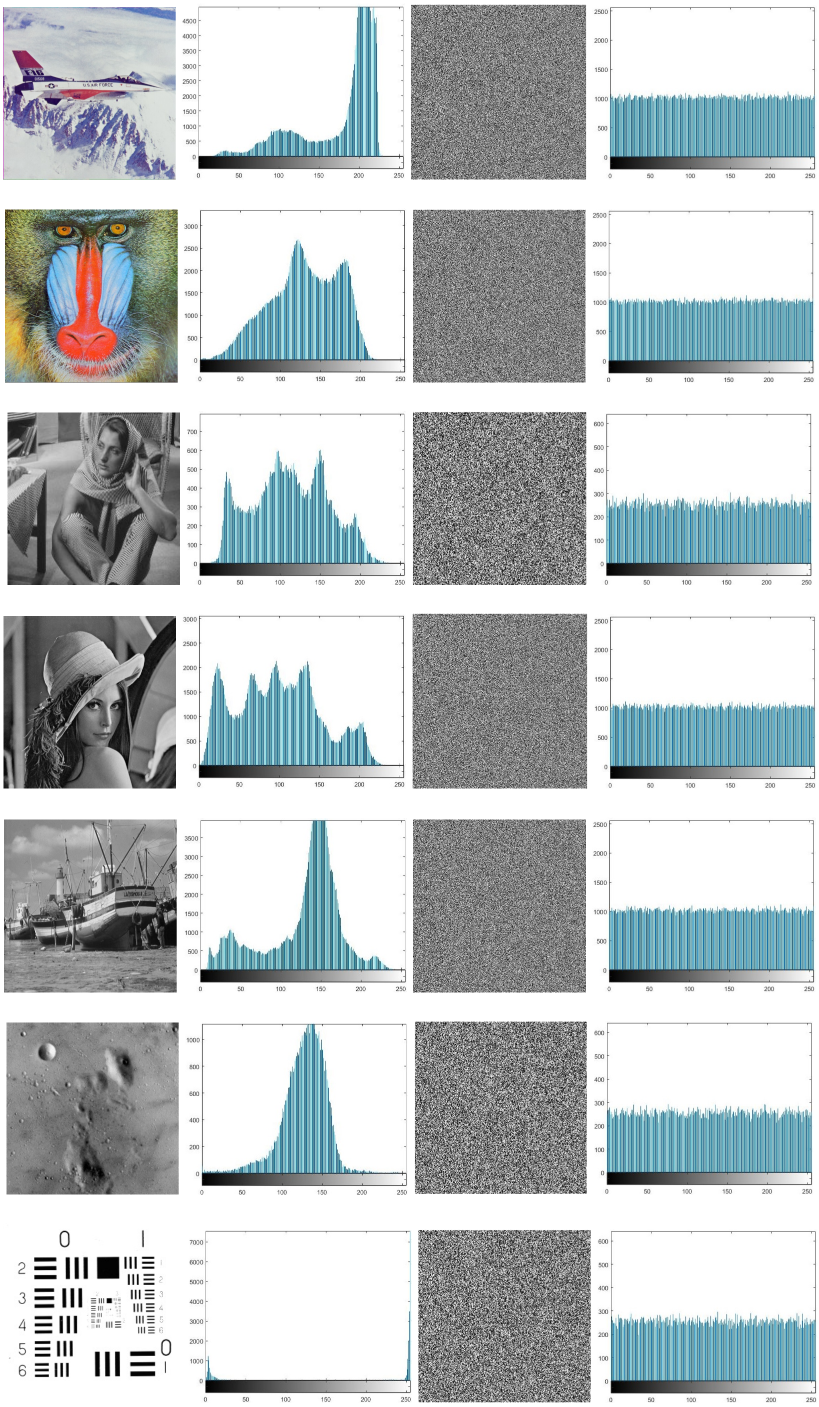


Figure 2. Histogram diagram for plain and cipher images at $a = 2, b = 0.5, c = 0.5, x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$.

5. Security Analysis

A good encryption algorithm possesses the following properties: (i) histograms of cipher images that are completely uniform; (ii) good information entropy; (iii) very low relationship between two adjoining pixels; (iv) resistance to statistical and differential attacks; (v) sensitivity to the secret keys;

(vi) huge key space; (vii) high contrast; (viii) resisting noise attacks. Those aspects are tested for the proposed algorithm.

5.1. Statistical Analysis

Statistically, plain and cipher images must be different and therefore similarity must be hidden in order to avoid the attempts of attackers to break the algorithm.

5.1.1. Analysis of the Histogram

The distribution of pixel values in the image is shown by a histogram analysis. The histogram of the cipher image should be uniformly distributed. Therefore, obtaining any useful statistical information will be very hard. The third and fourth columns of Figure 2 display cipher images and their corresponding histograms.

The histograms of the obtained encrypted images are approximately uniform, are different from those histograms of the plain images, and show that the proposed algorithm is secure enough for image encryption. The cipher images have histograms that have an approximately uniform distribution. The chi-square test [24] is used to check this uniformity using the following formula:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - e_i)^2}{e_i}, \tag{4}$$

where the gray level number is $k = 256$ and O_i and e_i represent the observed and the expected occurrence frequencies of each gray level (0–255). Using a level of significance that $\alpha = 0.05$, the chi-square values of the cipher images are given in Table 1.

It is easy to see that $\chi^2(k-1, \alpha) = \chi^2(255, 0.05) = 293$. Therefore, our null hypothesis is accepted with this level of significance, and the cipher histograms have a uniform distribution. It is shown in Table 1 that the value of χ^2 is less than the tabulated value $\chi^2(255, 0.05)$.

Table 1. χ^2 values of plain and cipher images at $a = 2, b = 0.5, c = 0.5, x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$.

Image	χ^2	
	Plain Image	**Cipher Image**
Lena	4.0659×10^4	255.4371
Cameraman	1.1097×10^5	255.9152
Barbara	3.5245×10^4	258.9021

5.1.2. Entropy Process of Information

For a source with symbol S, the process of entropy is denoted by H and is calculated by the following formula [25].

$$H(S) = -\sum_{i=0}^{N-1} p(s_i) log_2 p(s_i) \tag{5}$$

where $p(s_i)$ refers to the probability of symbol s_i and the bits are used for entropy. The theoretical result of entropy $H(S)$ is eight if $N = 256 = 2^8$ gray values for an image with equal probability. The information entropy shows the distribution of gray values. The information entropy for the cipher images is exhibited in Table 2. The obtained values are near the theoretical value of eight. Recently, the authors in [26] proposed a statistical test for the block entropy and introduced qualitative and quantitative results. We have selected 100 non-overlapped blocks of size 16 × 16 randomly from each cipher image. We have calculated the information entropy for each block based on Equation (5), and the average entropy was recorded. From Table 2, all actual block entropy values are passed the theoretical block entropy values at $\alpha = 0.01$ and $\alpha = 0.05$. Therefore, we may conclude that the cipher images is random-like after applying our proposed encryption algorithm.

Table 2. Information entropy tests for cipher images at $a = 2, b = 0.5, c = 0.5, x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$.

Image	**Global Entropy**	**Actual Block Entropy**	**Theoretical Block Entropy** $\alpha = 0.01$ **7.16276745**	$\alpha = 0.05$ **7.16634107**
lena (256 × 256)	7.9981	7.1921	Pass	Pass
lena (512 × 512)	7.9994	7.1862	Pass	Pass
barbara (256 × 256)	7.9973	7.1801	Pass	Pass
airplane (512 × 512)	7.9996	7.1817	Pass	Pass
boat (512 × 512)	7.9995	7.1849	Pass	Pass
house (256 × 256)	7.9984	7.1829	Pass	Pass
baboon (512 × 512)	7.9995	7.1827	Pass	Pass
moon surface (256 × 256)	7.9982	7.1819	Pass	Pass
resolution chart (256 × 256)	7.9980	7.1812	Pass	Pass

5.1.3. Examination of Correlation

In this subsection, the vertical, horizontal, and diagonal correlations between two adjacent pixels in a cipher image are evaluated. This examination can be carried out by selecting randomly 5000 pairs of two neighboring pixels from the original and cipher images. The coefficient of correlation for each pair is obtained using [4,27]:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{6}$$

where:

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

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

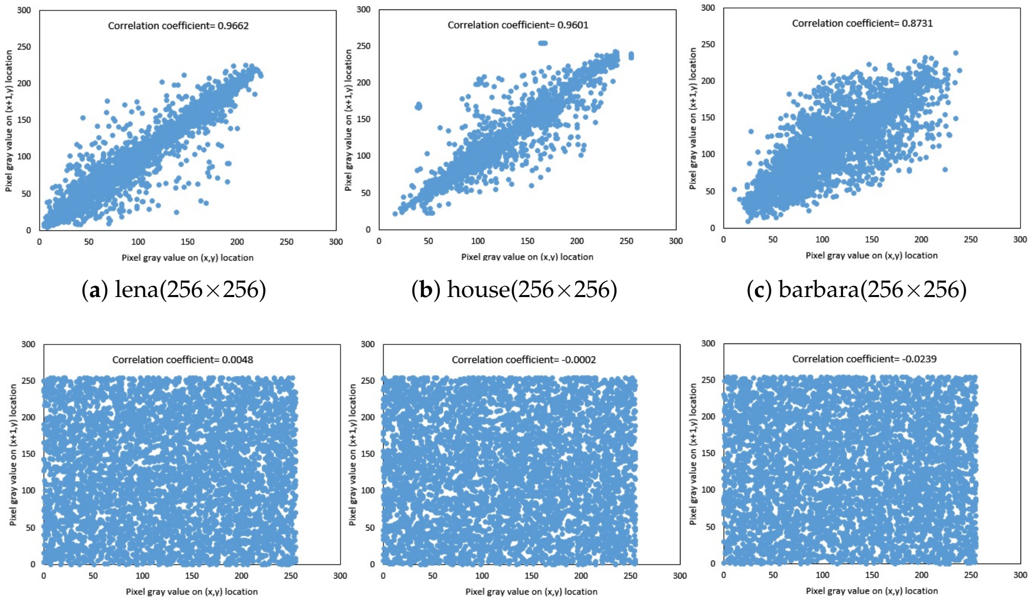
and:

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

where the gray values of any two neighboring pixels of an image are denoted by x and y. Figure 3 demonstrates the horizontal distribution of the relationship between two neighboring pixels in the original image and its corresponding ciphered image. Table 3 shows the relationship of two neighboring pixels in the original and cipher images. Figure 3 presents those correlations for the plain and the cipher one. The correlation coefficients of the cipher image for our proposed algorithm and other algorithms are shown in Table 4. The correlation coefficients of two adjacent pixels of the cipher image are close to zero.

Table 3. Correlation coefficients of two adjacent pixels of the plain image and cipher image at a = 2, $b = 0.5,\ c = 0.5,\ x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$.

Image		**Plain Image**	**Cipher Image**
lena (128 × 128)	H	0.8906	−0.0102
	V	0.9518	−0.0230
	D	0.8512	0.0119
lena (256 × 256)	H	0.9351	0.0005
	V	0.9692	0.0017
	D	0.9181	−0.0025
lena (512 × 512)	H	0.9659	−0.0091
	V	0.9837	−0.0198
	D	0.9552	−0.0062
barbara (256 × 256)	H	0.8105	−0.0088
	V	0.8797	−0.0179
	D	0.8335	−0.0054
airplane (512 × 512)	H	0.9566	0.0096
	V	0.9600	0.0053
	D	0.9237	0.0063
boat (512 × 512)	H	0.9385	−0.0025
	V	0.9669	0.0186
	D	0.9225	−0.0111
house (256 × 256)	H	0.9812	0.0003
	V	0.9660	−0.0178
	D	0.9402	0.0050
baboon (512 × 512)	H	0.9121	0.0110
	V	0.8634	0.0019
	D	0.8282	0.0118
moon surface (256 × 256)	H	0.8859	−0.0026
	V	0.9316	0.0100
	D	0.8988	−0.0029
resolution chart (256 × 256)	H	0.8828	0.0007
	V	0.8793	0.0106
	D	0.7524	0.0159



(**a**) lena(256×256) (**b**) house(256×256) (**c**) barbara(256×256)

(**d**) lena(256×256) (**e**) house(256×256) (**f**) barbara(256× 256)

Figure 3. (**a**–**c**) Correlation analysis of plain images and (**d**–**f**) correlation analysis of cipher images at $a = 2,\ b = 0.5,\ c = 0.5,\ x_{1,0} = 0.11,\ x_{2,0} = 0.12$, and $k = 0.87$.

Table 4. Correlation coefficients of two adjacent pixels of the cipher image for the proposed algorithm at $a = 2,\ b = 0.5,\ c = 0.5,\ x_{1,0} = 0.11,\ x_{2,0} = 0.12$, and $k = 0.87$ and other algorithms.

Image		**Cipher Image**						
		Proposed	[6]			[1]	[21]	[22]
			Case I	**Case II**	**Case III**			
lena (256 × 256)	H	0.0005	0.0122	0.0075	−0.0038	0.0077	-	−0.0230
	V	0.0017	−0.0456	−0.0079	0.0093	0.0168	-	0.0019
	D	−0.0025	−0.0188	−0.0093	−0.0189	0.0104	-	−0.0034

5.2. Sensitivity Analysis

One of the important features of a good encryption algorithm is the sensitivity to the security key and the plain image. Therefore, any small change in the security key or the plain image tends to a completely different cipher image [28].

5.2.1. Differential Attack

In order to get some insights about the influence of changing one pixel in the cipher image, some measures such as *NPCR* (Number of Pixels Change Rate) and *UACI* (Unified Average Changing Intensity) are used. They have the following formulae:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%, \tag{7}$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|I_1(i,j) - I_2(i,j)|}{255}\right] \times 100\%. \tag{8}$$

where:

$$D(i,j) = \begin{cases} 1 & \text{if } I_1(i,j) \neq I_2(i,j), \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The width and height of the two ciphered images I_1 and I_2 are denoted by W and H.

In Table 5, each gray level is coded in such a way that only eight bits are used in the coding. The obtained calculations demonstrate that the percentage of the *NPCR* is acceptable, as it is close to 99.6%, while the percentage of the *UACI* is close to 33.4% is also acceptable for the assigned images. Therefore, we can conclude that our encryption algorithm can survive against differential attack.

Table 5. Comparison of the Number of Pixels Change Rate (*NPCR*) and Unified Average Changing Intensity (*UACI*) of the lena, barbara, and house images with size 256 × 256 using our proposed algorithm and other algorithms.

Cipher Algorithm	***NPCR***	***UACI***
Theoretical expected value	99.61%	33.46%
Proposed (lena)	99.60%	33.43%
Proposed (barbara)	99.62%	33.45%
Proposed (house)	99.60%	33.40%
[1] (lena)	99.59%	30.63%
[21] (lena)	99.60%	33.46%
[22] (lena)	99.62%	33.51%

5.2.2. Key Sensitivity Test

Some key sensitivity tests are carried out in this subsection in order to check our proposed algorithm. Figure 4a–c shows the decrypted images of the ciphered images of lena, barbara, and house using the proper values of the secret key. In the secret key, if the value of a is changed to

1.99999999999999 and the other values of the secret key are unchanged, the results give the decrypted images shown in Figure 4d–f, which are completely unreadable and different from the original ones. Figure 4g–i shows the decrypted images of the ciphered images with a similar security key except $c = 0.49999999999999$. Based on those obtained results, one can conclude that any small change in one of the security keys leads to a bad image being produced, and then, the original plain image cannot be retrieved. Therefore, our algorithm possesses strong security keys.

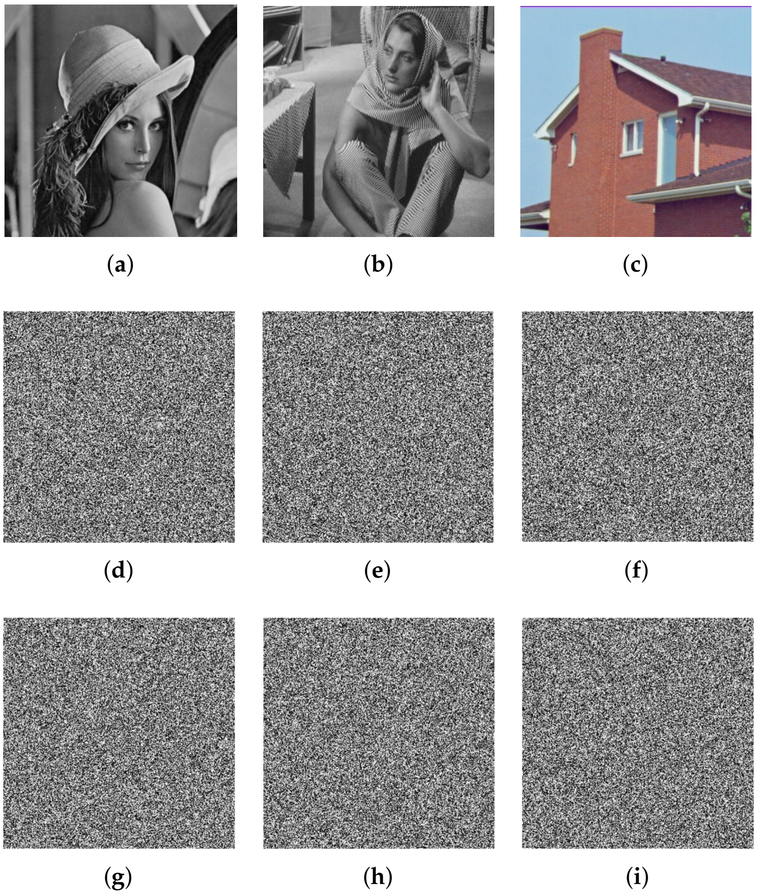


Figure 4. Decrypted images of lena, barbara, and house for the proper secret key (**a**–**c**), (**d**–**f**) decrypted images at $a = 1.99999999999999, b = 0.5,\ c = 0.5, x_1(0) = 0.11, x_2(0) = 0.12$, and $k = 0.87$, and (**g**–**i**) decrypted images at $a = 2, b = 0.5, c = 0.49999999999999, x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$.

5.3. The Analysis of Our Key Space

A strong encryption algorithm has a large enough key space. For high security, the key space should be greater than 2^{100} [2,29]. The proposed algorithm uses two initial values for x_0 and two initial values for r_0. The logistic map is utilized for row shuffling and column shuffling. In addition, there are six initial values, $x_{1,0}, x_{2,0}, a, b, c$, and k, which are used in the two-dimensional chaotic economic map. Therefore, our security keys consist of ten values. Therefore, using a precision of 10^{-14} [2], the size of the proposed algorithm key space is 10^{140}, and it is more than 2^{100}. Table 6 shows that our proposed algorithm has a key space larger than the key spaces in [1,6,21,22]. Therefore, the key space of our algorithm is large enough to resist brute-force attacks. This makes our proposed algorithm sufficiently robust to oppose a wide range of brute-force attacks.

Table 6. Key space comparison.

Algorithm	**Ours**	[1]	[6]	[21]	[22]
Key Space	10^{140}	10^{140}	10^{84}	$2^{256} \approx 1.16 \times 10^{77}$	$2^{210} \approx 1.65 \times 10^{63}$

5.4. Noise Attacks

The security of our algorithm against noise is tested. Two famous types of noises are incorporated into the encrypted images. The first is Gaussian noise with variances of 0.01 and 0.1. The second is salt and pepper with densities of 0.05 and 0.1. The Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are used for measuring.

Assume that X and Y denote the plain image and decrypted image for the ciphered image after the noise is incorporated. MSE and PSNR are calculated by the following equations:

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(X_{ij} - Y_{ij}\right)^2 \tag{10}$$

and:

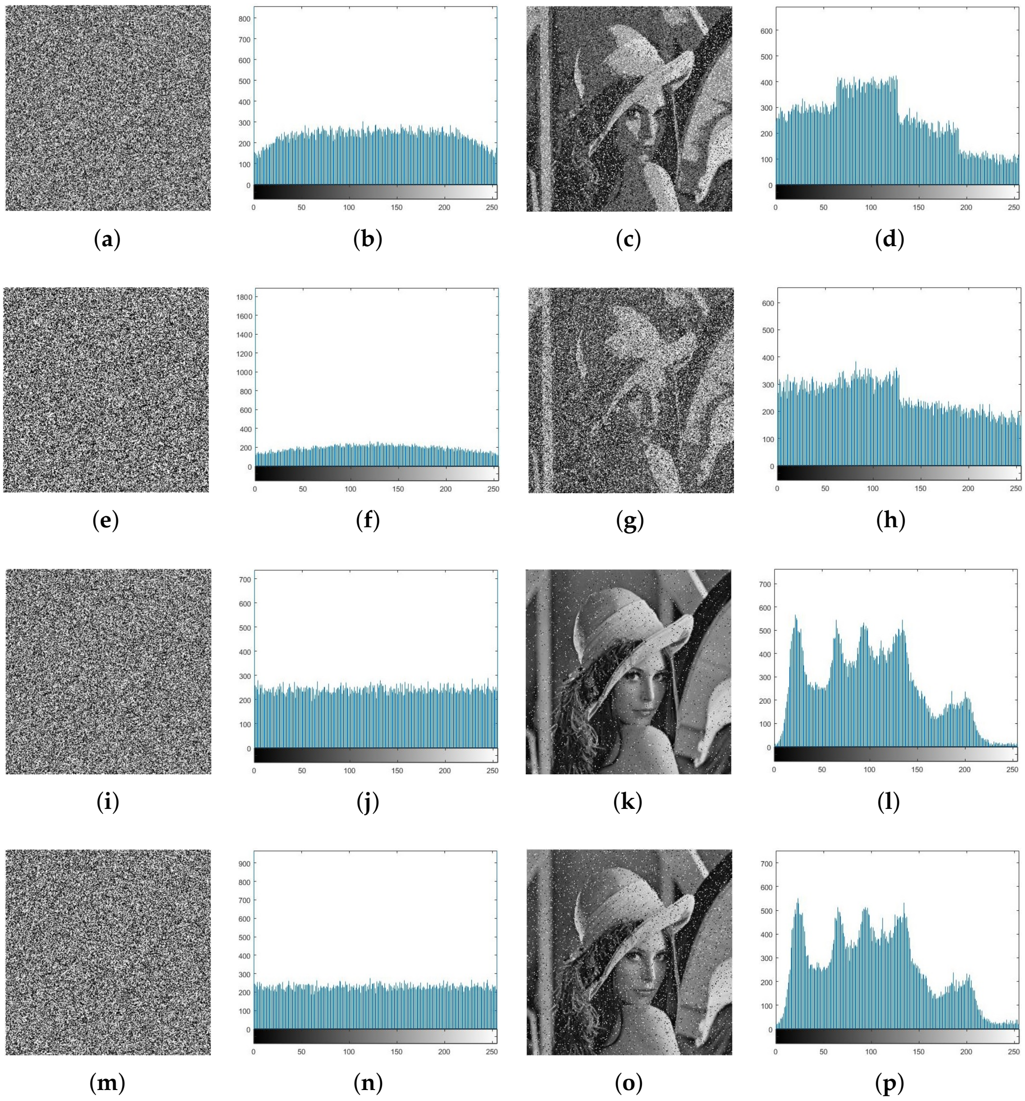
$$\text{PSNR} = 10 \times \log_{10} \left(\frac{I^2}{MSE} \right) \tag{11}$$

where I is the maximum possible pixel value of an image.

The high similarity between the plain image and the decrypted image has been measured using the PSNR. Table 7 and Figure 5 show the results. The results show that the images with salt and pepper noise are better than the images with Gaussian noise because the PSNR is greater than 18.6. In addition, the histograms of the decrypted images and the plain images are approximately similar.

Table 7. Noise attack results.

Noise	**MSE (Our Algorithm)**	**MSE [30]**	**PSNR (Our Algorithm)**	**PSNR [30]**
Gaussian noise with variance = 0.01 and mean = 0	2321.4	4410.1	14.5	11.7
Gaussian noise with variance = 0.1 and mean = 0	5201.2	5631.4	11.0	10.6
Salt and pepper noise with density 0.05	437.9	869.9	21.7	18.7
Salt and pepper noise with density 0.1	893.1	1829.6	18.6	15.5



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

(**i**) (**j**) (**k**) (**l**)

(**m**) (**n**) (**o**) (**p**)

Figure 5. (**a**,**e**,**i**,**m**) Encrypted images after adding Gaussian and salt and pepper, (**b**,**f**,**j**,**n**) corresponding histograms of the encrypted images, (**c**,**g**,**k**,**o**) decrypted images at $a = 2, b = 0.5, c = 0.5, x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$, and (**d**,**h**,**l**,**p**) corresponding histograms of decrypted images.

5.5. Contrast Analysis

The difference investigation of the image permits the observer to entirely understand the texture of images. The ciphered image has high contrast levels as a result of the high randomness given by the map (1) in the process of encryption. We have evaluated the contrast of the ciphered image and the effectiveness of Map (1) in image encryption applications. This can be carried out by the following formula:

$$C = \sum_{i,j} |i - j|^2 p(i, j) \tag{12}$$

where $p(i, j)$ is used to refer to the number of gray levels in the co-occurrence matrices.

The contrast analyses of both the plain and cipher images for lena (256 × 256), barbara (256 × 256), and house (256 × 256) are shown in Table 8. Table 9 shows that the contrast analysis of the introduced algorithm overcomes the algorithm in [6]. It has the best result.

Table 8. Contrast analysis of plain and cipher images at $a = 2, b = 0.5, c = 0.5, x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$.

Image	Contrast	
	Plain Image	Cipher Image
lena (256 × 256)	0.3563	10.6201
house (256 × 256)	0.1741	10.5431
barbara (256 × 256)	0.9463	10.6986

Table 9. Contrast analysis of the plain (lena (256 × 256)) and cipher image using the proposed algorithm at $a = 2, b = 0.5, c = 0.5, x_{1,0} = 0.11, x_{2,0} = 0.12$, and $k = 0.87$ and other algorithms.

Image	Contrast							
		Cipher Image						
	Plain Image	Proposed Algorithm	[6]			[21]	[22]	[1]
			Case I	Case II	Case III			
lena	0.3563	10.6201	10.1655	10.3909	10.3971	10.5034	10.4723	10.4767

5.6. Gray Value Degree Analysis

Measuring the gray difference of a pixel requires one to know its four neighbors. This can be carried out using the following formula:

$$G(x,y) = \frac{\sum [I(x,y) - I(\acute{x},\acute{y})]^2}{4}, (\acute{x},\acute{y}) = \begin{cases} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{cases} \tag{13}$$

where $I(x,y)$ refers to the value of a pixel at location (x,y) and $I(\acute{x} - \acute{y})$ denotes its four neighboring pixels. Now, the average difference of the whole image is measured by the following:

$$\bar{I}(x,y) = \frac{\sum_{x=2}^{M-1} \sum_{y=2}^{N-1} G(x,y)}{(M-2) \times (N-2)} \tag{14}$$

where M and N refer to the row and column numbers in an image. Using (11) and (12), the Gray Value Degree (GVD) is defined as,

$$\text{GVD} = \frac{\acute{\bar{I}}(x,y) - \bar{I}(x,y)}{\acute{\bar{I}}(x,y) + \bar{I}(x,y)} \tag{15}$$

where the average neighborhood gray difference of an original image is denoted by $\acute{\bar{I}}$, while $\bar{I}(x,y)$ refers to those of the encrypted image. Table 10 presents the degree values of the computed gray value degree for distinct images using the proposed algorithm. One can see from Table 10 that our obtained degree values are close to the ideal value, which is one. Table 11 shows some validations of the proposed algorithm versus existing algorithms in the literature.

Table 10. Gray Value Degree (GVD for different test images.

Image	GVD Value
lena	0.9653
barbara	0.9571
house	0.9755
moon surface	0.9756
resolution chart	0.9482

Table 11. GDV analysis of the proposed algorithm with other algorithms.

Image	**GVD Value**			
	Proposed Algorithm	**Arnold's**	**[31]**	**[32]**
lena	0.9653	0.8900	0.9540	0.9624

5.7. Peak Signal to Noise Ratio Analysis

PSNR is used to carry out the evaluation of the encryption algorithms. It is applied to both images, the original image and the encrypted one. It considers the original image as a signal, while the encrypted image as a noise. PSNR is calculated by using Equation (11). Table 12 shows the PSNR values of some different images. It is shown that those values are low, which means it is difficult to retrieve the original image from the encrypted one in the case of hacker attacks.

Table 12. PSNR values for different test images.

Image	**PSNR Value**
lena	8.5557
barbara	9.1158
house	9.2541
moon surface	10.1982
resolution chart	4.9300

5.8. Computational Complexity

We make a comparison of the computational complexity between the traditional DES, AES, and other algorithms in the literature including our proposed algorithm. The results appear in Table 13. Based on Table 13, we can concluded that the time of our algorithm is less than the traditional DES and AES algorithms and the other algorithms in the literature and is completely acceptable for encryption of an image.

Table 13. The speed of encryption for each algorithm. DES, Data Encryption Standard.

Algorithm	**Encryption Time (in Seconds)**
Proposed Algorithm	0.1309
DES	0.6305
AES	0.2173
[21]	0.1493
[22]	0.2021
[1]	0.5463

6. Conclusions

The current paper introduced a new cryptographic algorithm based on a two-dimensional chaotic economic map and logical operator bit-wise *XOR* . The size of the key space for our proposed algorithm is 10^{140}, which allows our algorithm to survive against attacks. The numerical experiments have shown that the algorithm is very sensitive to the security keys, and hence, any bit change or random guessing about the proper value of the security keys fails to retrieve the original image. The obtained results by the proposed algorithm have shown that the ciphered image is random-like since the information entropy of the cipher image of size 256 × 256 is very close to the theoretical value of eight, the *NPCR* is close to 99.6%, and *UACI* is close to 33.4%. Therefore, our algorithm is extremely sensitive to any small change in the original image. According to the obtained results, we may conclude that the proposed algorithm owns a high level of security that allows it to prevent any attempts of attack by hackers.

Author Contributions: Conceptualization, methodology, and writing, original draft preparation, S.S.A. and A.A.K.; software, validation, formal analysis, and investigation, A.A.K. and F.S.A.-A.; writing, review and editing, S.S.A., A.A.K. and A.A.-K.

Funding: This research was funded by Deanship of Scientific Research at King Saud University for funding this research group number (RG-1435-054).

Acknowledgments: The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for funding this research group number (RG-1435-054).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Askar, S.S.; Karawia, A.A.; Alammar, F. Cryptographic algorithm based on pixel shuffling and dynamical chaotic economic map. *IET Image Process.* **2018**, *12*, 158–167. [CrossRef]
2. Zhu, C. A novel image encryption scheme based on improved hyperchaotic sequences. *Opt. Commun.* **2012**, *285*, 29–37. [CrossRef]
3. Abomhara, M.; Zakaria, O.; Khalifa, O. An Overview of Video Encryption Techniques. *Int. J. Comput. Theory Eng.* **2010**, *2*, 1793–8201. [CrossRef]
4. Chen, G.; Mao, Y.; Chui, C. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761. [CrossRef]
5. Chiaraluce, F.; Ciccarelli, L.; Gambi, E.; Pierleoni, P.; Reginelli, M. A new chaotic algorithm for video encryption. *IEEE Trans. Consum. Electron.* **2002**, *48*, 838–843. [CrossRef]
6. Askar, S.S.; Karawia, A.A.; Alshamrani, A. Image encryption algorithm based on chaotic economic model. *Math. Probl. Eng.* **2015**, *2015*, 341729. [CrossRef]
7. Matthews, R. On the derivation of a chaotic encryption algorithm. *Cryptologia* **1989**, *13*, 29–42. [CrossRef]
8. Xu, E.; Shao, L.; Cao, G.; Ren, Y.; Qu, T. A New Method of Information Encryption. In Proceedings of the International Colloquium on Computing, Communication, Control, and Management, Sanya, China, 8–9 August 2009; Volume 4, pp. 583–586.
9. Zhang, D.; Gu, Q.; Pan, Y.; Zhang, X. Discrete Chaotic Encryption and Decryption of Digital Images. In Proceedings of the International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; Volume 3, pp. 849–852.
10. Nien, H.; Huang, W.; Hung, C.; Chen, S.; Wu, S.; Huang, C.; Hsu, Y. Hybrid image encryption using multi-chaos-system. In Proceedings of the International Conference on Information, Communications and Signal Processing (ICICS), Macau, China, 8–10 December 2009; pp. 1–5.
11. Cao, Y.; Fu, C. An image encryption scheme based on high dimension chaos system. In Proceedings of the International Conference on Intelligent Computation Technology and Automation, Changsha, China, 20–22 October 2008; Volume 2, pp. 104–108.
12. Jeyamala, J.; GrpiGranesh, S.; Raman, S. An image encryption scheme based on one time pads—A chaotic approach. In Proceedings of the International Conference on Computing, Communication and Networking Technologies, Karur, India, 29–31 July 2010; pp. 1–6.
13. Zhu, W.; Shen, Y. Encryption Algorithms Using Chaos and CAT Methodology. In Proceedings of the International Conference of Anti-Counterfeiting Security and Identification in Communication (ASID), Chengdu, China, 18–20 July 2010; pp. 20–23.
14. Elnashaie, S.; Abashar, M. On the chaotic behavior of forced fluidized bed catalytic reactors. *Chaos Solitons Fractals* **1995**, *5*, 797–831. [CrossRef]
15. Kocarev, L. Chaos-based cryptography: A brief overview. *IEEE Circuits Syst. Mag.* **2001**, *1*, 6–21. [CrossRef]
16. Ponomarenko, V.; Prokhorov, M. Extracting information masked by the chaotic signal of a time-delay system. *Phys. Rev. E* **2002**, *66*, 1–7. [CrossRef]
17. Sivakumar, T.; Venkatesan, R. Image Encryption Based on Pixel Shuffling and Random Key Stream. *Int. J. Comput. Inf. Technol.* **2014**, *3*, 1468–1476.
18. Zhang, J.; Fang, D.; Ren, H. Image Encryption Algorithm Based on DNA Encoding and Chaotic Maps. *Math. Probl. Eng.* **2014**, *2014*, 917147. [CrossRef]
19. Wang, W.; Tan, H.; Pang, Y.; Li, Z.; Ran, P.; Wu, J. Novel Encryption Algorithm Based on DWT and Multichaos Mapping. *J. Sens.* **2016**, *2016*, 2646205. [CrossRef]

20. Zou, J.; Weng, T. A New Image Encryption Instant Communication Method Based On Matrix Transformation. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing. Smart Innovation, Systems and Technologies*; Pan, J.S., Tsai, P.W., Huang, H.C., Eds.; Springer: Berlin, Germany, 2017; Volume 63, pp. 321–329.
21. Hua, Z.; Jin, F.; Xu, B.; Huang, H. 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **2018**, *149*, 148–161. [CrossRef]
22. Xu, L.; Li, Z.; Li, J.; Hua, W. A novel bit-level image encryption algorithm based on chaotic maps. *Opt. Lasers Eng.* **2016**, *78*, 17–25. [CrossRef]
23. Askar, S.S. Complex dynamic properties of Cournot duopoly games with convex and log-concave demand function. *Oper. Res. Lett.* **2014**, *42*, 85–90. [CrossRef]
24. Kwok, H.; Tang, W. A fast image encryption system based on chaotic maps with finite precision representation. *Chaos Solitons Fractals* **2007**, *32*, 1518–1529. [CrossRef]
25. Sobhy, M.; Shehata, A. Methods of attacking chaotic encryption and countermeasures. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, USA, 7–11 May 2001; Volume 2, pp. 1001–1004.
26. Tao, X.; Liao, X.; Tang, G. A novel block cryptosystem based on iterating a chaotic map. *Phys. Lett. A* **2006**, *349*, 109–115.
27. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. [CrossRef]
28. Pareek, N.; Patidar, V.; Sud, K. Image encryption using chaotic logistic map. *Image Vis. Comput.* **2006**, *24*, 926–934. [CrossRef]
29. Song, C.; Qiao, Y.; Zhang, X. An image encryption scheme based on new spatiotemporal chaos. *Opt.—Int. J. Light Electron Opt.* **2013**, *124*, 3329–3334. [CrossRef]
30. Parvin, Z.; Seyedarabi, H.; Shamsi, M. A new secure and sensitive image encryption scheme based on new substitution with chaotic function. *Multimedia Tools Appl.* **2016**, *75*, 10631–10648. [CrossRef]
31. Hanchinamani, G.; Kulakarni, L. Image Encryption Based on 2-D Zaslavskii Chaotic Map and Pseudo Hadmard Transform. *Int. J. Hybrid Inf. Technol.* **2014**, *7*, 185–200. [CrossRef]
32. Rhouma, R.; Solak, E.; Belghith, S. Cryptanalysis of a new substitution-diffusion based image cipher. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 1887–1892. [CrossRef]