electronics

MDPI

Article

New Image Encryption Algorithm Using Hyperchaotic System and Fibonacci Q-Matrix

Khalid M. Hosny [1,*], Sara T. Kamal [2], Mohamed M. Darwish [2] and George A. Papakostas [3]

[1] Information Technology Department, Zagazig University, Zagazig 44519, Egypt
[2] Computer Science Department, Assiut University, Assiut 71515, Egypt; saratarik@aun.edu.eg (S.T.K.); mohamed_darwish@aun.edu.eg (M.M.D.)
[3] HUMAIN-Lab, Department of Computer Science, International Hellenic University, 65404 Kavala, Greece; gpapak@cs.ihu.gr
* Correspondence: k_hosny@yahoo.com; Tel.: +20-1024455680

Abstract: In the age of Information Technology, the day-life required transmitting millions of images between users. Securing these images is essential. Digital image encryption is a well-known technique used in securing image content. In image encryption techniques, digital images are converted into noise images using secret keys, where restoring them to their originals required the same keys. Most image encryption techniques depend on two steps: confusion and diffusion. In this work, a new algorithm presented for image encryption using a hyperchaotic system and Fibonacci Q-matrix. The original image is confused in this algorithm, utilizing randomly generated numbers by the six-dimension hyperchaotic system. Then, the permutated image diffused using the Fibonacci Q-matrix. The proposed image encryption algorithm tested using noise and data cut attacks, histograms, keyspace, and sensitivity. Moreover, the proposed algorithm's performance compared with several existing algorithms using entropy, correlation coefficients, and robustness against attack. The proposed algorithm achieved an excellent security level and outperformed the existing image encryption algorithms.

Keywords: image encryption; hyperchaotic system; fibonacci Q-matrix; attacks

check for updates

Citation: Hosny, K.M.; Kamal, S.T.; Darwish, M.M.; Papakostas, G.A. New Image Encryption Algorithm Using Hyperchaotic System and Fibonacci Q-Matrix. *Electronics* **2021**, *10*, 1066. https://doi.org/10.3390/electronics10091066

Academic Editors: Byung Cheol Song and Manohar Das

Received: 23 March 2021
Accepted: 27 April 2021
Published: 30 April 2021

1. Introduction

The transmission of digital images through various networks is a routine process where thousands of digital images are transmitted every moment. In social networks, users do not want others to access their images. In healthcare networks, medical images are sensitive where any misuse of these images may lead to wrong diagnoses and inaccurate medical decisions. Transmission of the military images via different networks requires high-security levels to prevent intruders from getting them. Generally, owners of digital images do not want others to access their images without permission. For these reasons, securing images' contents has become an important issue. Several security approaches are used to achieve image confidentiality, so an unauthorized user cannot access image content.

Image security approaches are divided into three main categories: data hiding [1,2], image watermarking [3–7], and encryption [8–11]. In data hiding techniques, a secrete message is embedded in the cover image so that it is not detectable. In image watermarking techniques, pieces of digital data inserted in the image where the original and watermarked images' perceptibility are similar. In image encryption techniques, the digital input image converted to a noise image using a key, which is not understood or predicting its content. Users cannot restore the encrypted image without knowing the key.

There are several techniques used in digital image encryption, such as the theory of chaos [12–14], DNA [15–17], the method of quantum [18,19], and compressive sensing [20,21]. Image encryption techniques depend on two significant steps. The first step is confusion in which pixel arrangements changed. Diffusion is the second step, which depends on changing the values of pixels. Chaotic-based methods possess intrinsic properties such as non-periodicity, random behavior, and sensitivity to control parameters and initial conditions [22]. These properties enable the successful utilization of chaotic-based methods in the encryption of images.

Chai et al. [23] pointed out that digital images' chaotic-based encryption systems are classified into two main categories. The first category includes low-dimensional systems such as 1D chaotic maps. The second one is the high-dimensional systems, such as hyperchaotic systems. The low-dimensional chaotic maps friendly applicable due to their simple structures. Despite these intrinsic properties, these maps have a small keyspace and achieve low-security levels [24].

Several numbers of chaos-based encryption exist, such as [25–31]. Chen and Hu [32] proposed a medical image encryption method using a logistic-sine map for the confusing process. The scrambled image is divided into blocks where a hyperchaotic system is used for diffusing the image blocks. Chai et al. [33] utilized a memristive chaotic system in image encryption, which improved its ability to resist the differential attack. Chai et al. [34] presented a new image encryption algorithm based on the parameter-varying chaotic system, elementary cellular automata (ECA), and block compressive sensing (BCS). Tsafack et al. [35] designed a new 4D chaotic circuit and applied it in image encryption. In [36], Ramasamy et al. proposed a new algorithm that depends on Block Scrambling and Modified Zigzag Transformation to scramble the plain image, and then the key was generated based on Enhanced Logistic–Tent Map (ELTM) to diffuse the scrambled image. Zheng and Liu [37] designed a new scheme for encrypting gray images. First, a new 2D chaotic map system (2D-LSMM) was introduced, which is based on both logistic and sine maps. Then, the encryption scheme was based on DNA, where the encoding and operation rules of DNA sequences were determined by 2D-LSMM chaotic sequences. In [38], Kari et al. introduced a novel image encryption technique based on chaotic maps. In this algorithm, pixel positions were changed in the confusion phase by using Arnold's cat map. Additionally, the contents of pixels were updated in the diffusion phase that is controlled by the extension of the plain image matrix, XOR operation, and exchange operation. The authors in [39] presented a fast image encryption technique based on simultaneous permutation and diffusion operation (SPDO). The values of the pixels are permuted and diffused simultaneously using a SineSine map.

Liu et al. [40] utilized a coupled hyperchaotic system in pathological image encryption. Yu et al. [41] used Chen's hyperchaotic system with fractional Fourier transform to encrypt images. Hyperchaotic methods are used as alternatives to the low-dimensional chaotic systems to overcome their limitations. The hyperchaotic methods outperformed the low-dimension chaotic methods in terms of randomness, unpredictability, nonlinearity, and initial conditions. The hyperchaotic methods produced key sequences that have a large keyspace. Generally, the utilization of hyper-chaotic systems improves the level of security. However, image encryption algorithms that used hyperchaotic methods have weaknesses against different attacks. Moreover, the encrypted image histogram is not uniform for some algorithms.

Related works have some limitations that can be summarized as follows:

1. Low keyspace and less sensitivity to the initial conditions.
2. The initial condition of the chaotic map does not depend on the plain image that leads to weaknesses in resisting differential attacks.
3. When the encrypted image is attacked with noise and data cut, some of the encryption algorithms failed to retrieve the plain image.
4. Some of these algorithms cannot resist statistical attacks as the histogram of the encrypted image is not flat.

These weaknesses motivated the authors to propose a new algorithm for encrypting images. The proposed algorithm utilized a six-dimension (6D) hyperchaotic system and Fibonacci Q-matrix to encrypt grayscale images through two main steps. First, the pixels' positions in the original image scrambled using the 6D hyperchaotic system. Only three sequences from this 6D hyperchaotic system were randomly selected to permit the original image. Second, the Fibonacci Q-matrix is used in the diffusion process, where this process is performed on a confused image's sub-blocks. Based on performed experiments, the proposed image encryption algorithm successfully encrypts gray images with excellent performance. The contributions of this work are summarized as:

1. The first utilization of the Fibonacci Q-matrix in image encryption.
2. Using 6D hyperchaotic system in image encryption for the first time.
3. Integration of the 6D hyperchaotic system and Fibonacci Q-matrix assure high-security level.
4. The large keyspace of the proposed algorithm leads to good resistance to brute force attacks.
5. The proposed image encryption algorithm has super robustness to most attacks.
6. Analysis of the obtained results shows the excellent performance of the proposed algorithm.

The following sections are: The mathematical foundations of the 6D hyperchaotic system and the Fibonacci Q-matrix presented in Section 2. The proposed algorithm is presented in Section 3. Tests and results are discussed in Section 4. The conclusion is presented in Section 5.

2. Mathematical Foundations

2.1. Six-Dimensional Hyperchaotic System

Generally, mathematical analysis shows that chaotic functions are nonlinear with dynamic behavior. Therefore, their responses are unpredictable. Previous studies show that the hyperchaotic functions' dynamical behavior is much complicated than the corresponding one of the low-dimension chaotic functions. A hyperchaotic system should have at least four dimensions. Moreover, low-dimension chaotic functions contain only one positive Lyapunov exponent, while the hyperchaotic systems have at least two.

Wang and Yu [42] defined the 6D hyperchaotic system as:

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) + x_4 - x_5 - x_6 \\ \dot{x}_2 = cx_1 - x_2 - x_1x_3 \\ \dot{x}_3 = -bx_3 + x_1x_2 \\ \dot{x}_4 = dx_4 - x_2x_3 \\ \dot{x}_5 = ex_6 + x_3x_2 \\ \dot{x}_6 = rx_1 \end{cases} \tag{1}$$

where a, b, c, d, e, and r are constants; $x_1, x_2, x_3, x_4, x_5,$ and x_6 refer to state variables of the 6D hyperchaotic system. In this paper, the constant values selected are $a = 10$, $b = \frac{8}{3}$, c = 28, d = −1, e = 8, and $r = 3$. This selection ensures that the system has two positive Lyapunov exponents that achieve the condition (sum of all exponents is negative).

2.2. Fibonacci Q-matrix

The elements of the Fibonacci sequence, F_n, are [43]:

$$F_n = F_{n-1} + F_{n-2}\ ,\ n > 1 \tag{2}$$

where $F_1 = F_2 = 1$.

The Fibonacci Q matrix is given by:

$$Q = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \tag{3}$$

The nth power of the Fibonacci Q matrix is the matrix defined by:

$$Q^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} \tag{4}$$

where F_n is the Fibonacci number, and the determinants of the Fibonacci Q-matrix is:

$$Det(Q^n) = F_{n+1}F_{n-1} - F_n^2 = (-1)^n \tag{5}$$

The inverse matrix Q^{-n} has the following form:

$$Q^{-n} = \begin{bmatrix} F_{n-1} & -F_n \\ -F_n & F_{n+1} \end{bmatrix} \tag{6}$$

3. The Proposed Algorithm

The new algorithm utilized a 6D hyperchaotic system and Fibonacci Q-matrix to encrypt the input image. Since the 6D hyperchaotic system has complex high-dynamic behaviors and two positive Lyapunov exponents, its utilization improves the encryption performance and increases security level. Fibonacci Q-matrix is very simple, fast, and able to diffuse the scrambled image. A flowchart of the proposed encryption-decryption algorithm is shown in Figure 1.

3.1. Encryption

The encryption depends on two steps: confusion and diffusion. The pixels' arrangements and values are modified in these processes, respectively. The confusion step is based on the 6D hyperchaotic system. First, we calculate the initial condition of the system that is based on the plain image. Then a new vector is obtained by iterating the hyper chaotic system, and then we select three sequences ($x_1, x_3,$ and x_5). This vector is sorted, and the position of the sorted numbers is used to confuse the plain image. After confusing the plain image, the diffusion step is performed to obtain the encrypted image. In our algorithm, the diffusion is based on the Fibonacci Q-matrix. The scrambled image is divided into blocks, each with size 2 × 2, and then each block is diffused using the Fibonacci Q-matrix. Two rounds of confusion and diffusion steps are performed to get the encrypted image. Algorithm 1 describes the encryption steps.

3.2. Decryption

The decryption steps are the reverse of the encryption steps. The plain image can be retrieved from the encrypted image by doing the following steps:

1. The encrypted image (C) is divided into blocks, each with size 2 × 2, and then the diffusion equation with Q^{-10} is applied to image blocks by using the following equation:

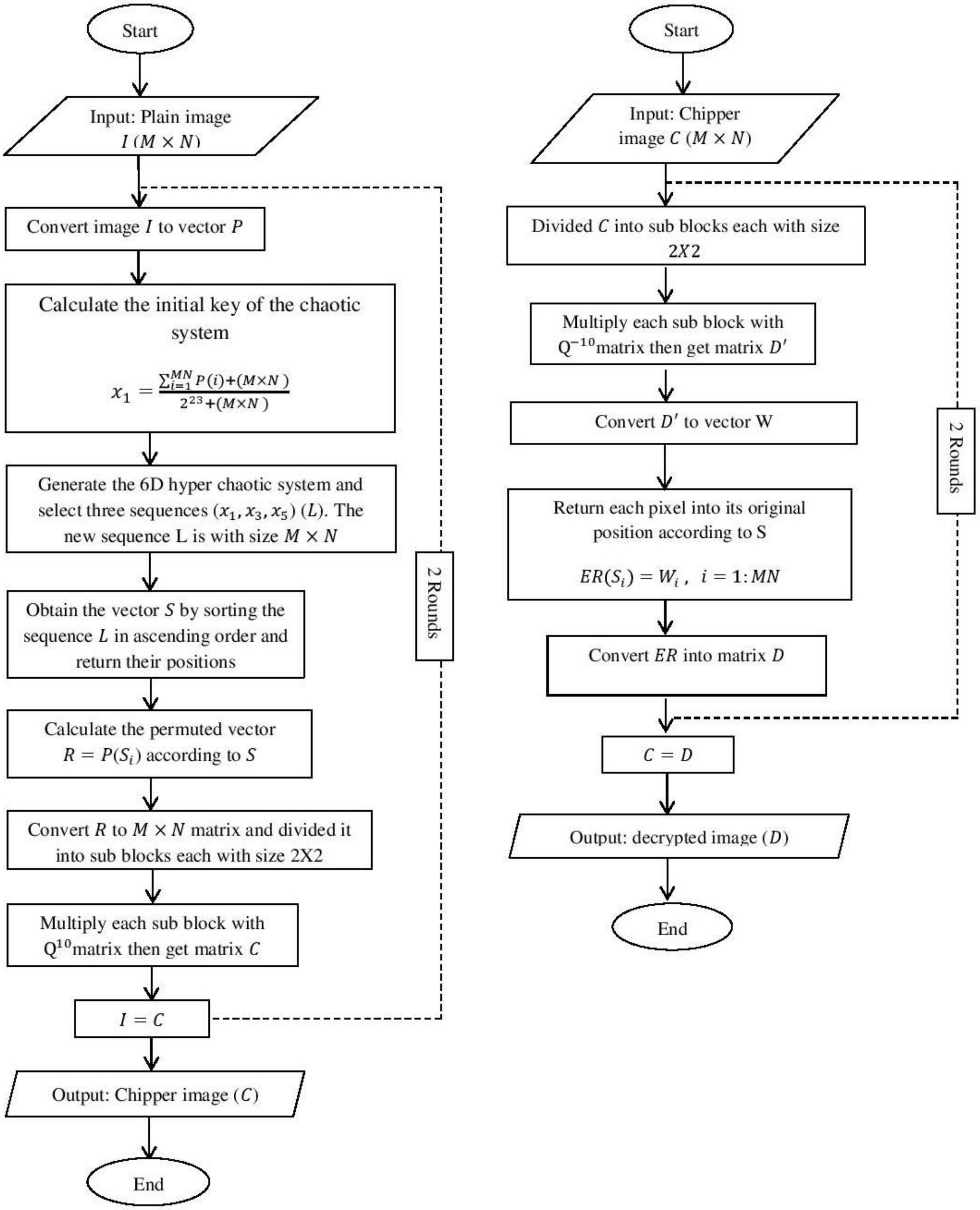
$$\begin{bmatrix} D'_{i,j} & D'_{i,j+1} \\ D'_{i+1,\ j} & D'_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} C_{i,j} & C_{i,j+1} \\ C_{i+1,j} & C_{i+1,j+1} \end{bmatrix} \begin{bmatrix} 34 & -55 \\ -55 & 89 \end{bmatrix} \bmod 256 \tag{11}$$

where $i = 1:3:5\ldots\ldots\ldots:M; j = 1:3:5\ldots\ldots\ldots:N.$

2. The scrambled image (D') obtained from the previous step is converted into vector W.
3. The vector S generated in the encryption step is used to return each pixel to its original position by the following equation:

$$ER(S_i) = W_i\ ,\ i = 1:MN \tag{12}$$

4. Convert the vector ER in to matrix to obtain the decrypted image (D).
5. Two rounds of decryption steps are performed to get the decrypted image.


Start
Input: Plain image $I\ (M \times N)$
Convert image I to vector P
Calculate the initial key of the chaotic system
$x_1 = \frac{\sum_{i=1}^{MN} P(i)+(M\times N)}{2^{23}+(M\times N)}$
Generate the 6D hyper chaotic system and select three sequences (x_1, x_3, x_5) (L). The new sequence L is with size $M \times N$
Obtain the vector S by sorting the sequence L in ascending order and return their positions
Calculate the permuted vector $R = P(S_i)$ according to S
Convert R to $M \times N$ matrix and divided it into sub blocks each with size 2X2
Multiply each sub block with Q^{10}matrix then get matrix C
$I = C$
2 Rounds
Output: Chipper image (C)
End
Start
Input: Chipper image $C\ (M \times N)$
Divided C into sub blocks each with size $2X2$
Multiply each sub block with Q^{-10}matrix then get matrix D'
Convert D' to vector W
Return each pixel into its original position according to S
$ER(S_i) = W_i\ ,\ \ i = 1{:}MN$
Convert ER into matrix D
2 Rounds
$C = D$
Output: decrypted image (D)
End

(**a**) Encryption Process

(**b**) Decryption process

Figure 1. Flow chart of the proposed algorithm.

Algorithm 1 The image encryption algorithm.

```
1:  $i = 1$
2:  Transform the image array to a vector $P$.
3:  Calculate the initial key of the hyperchaotic system as follows:
    $x_1 = \frac{\sum_{i=1}^{MN} P(i)+(M\times N)}{2^{23}+(M\times N)}$    (7)
    $x_i = \mathrm{mod}\left(x_{i-1} \times 10^6, 1\right)\ i = 2,\ 3,\ ..,\ 6$    (8)
    With the initial conditions; $x_1, x_2, \ldots, x_6$.
4:  Iterate the hyperchaotic system in (1) $N_0 + MN/3$ times then discard the $N_0$ values to make
    a new sequence $L$ with size $M \times N$. (we select three sequences ($x_1, x_3,$ and $x_5$) from the
    system in (1)).
5:  Sort $L$ in ascending order and return their positions in vector $S$.
6:  Permit the image vector $P$ to generate newly shuffled sequence $R$ as follows:
    $R_i = P(S_i), i = 1 : MN$    (9)
7:  Convert the sequence $R$ into the matrix $R'$ and divide it into sub-blocks, each with size $2 \times 2$.
8:  Get the Chipper image $C$ by multiplying each $2 \times 2$ sub-block in $R'$ with the Fibonacci Q
    matrix ($Q^{10}$):
    $\begin{bmatrix} C_{i,j} & C_{i,j+1} \\ C_{i+1,j} & C_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} R'_{i,j} & R'_{i,j+1} \\ R'_{i+1,j} & R'_{i+1,j+1} \end{bmatrix} \begin{bmatrix} 89 & 55 \\ 55 & 34 \end{bmatrix} \mathrm{mod}\ 256$    (10)
    with $i = 1 : 3 : \ldots\ldots : M, j = 1 : 3 : \ldots\ldots : N$.
9:  Let $I = C$ then $i = i + 1$.
10: Replicates steps 2 TO 8 for $i <= 2$.
```

4. Tests and Results

The proposed algorithm's effectiveness was tested using different standard grayscale images (Baboon, Pepper, Boat, Airplane, and Lena) with sizes 512 × 512 and 256 × 256. Additionally, the proposed algorithm compared with existing algorithms for image encryption. All performed experiments executed using MATLAB (R2015a) with a Laptop computer equipped with Core i5-2430M 2.4GH CPU and 4 GB RAM.

Eight experiments were performed to evaluate the proposed encryption algorithm using entropy, correlation coefficients, differential attack, noise and data cut attacks, histograms, keyspace, key sensitivity, and NIST Statistical Test.

4.1. Entropy

The image randomness measured by entropy can be defined by:

$$H(m) = \sum_{i=1}^{2^w-1} P(m_i) log_2 \frac{1}{P(m_i)} \tag{13}$$

where the occurrence probability of m_i is $P(m_i)$; the number 2^w refers to the total number of m_i, where the total number of image pixels is represented by the integer w. An ideal value of entropy for gray images is 8. The entropy of a few gray images encrypted using the new and existing algorithms [44–48] shown in Tables 1 and 2. Our proposed method records the highest average entropy value. Additionally, our proposed algorithm is tested on 10 images of the size 512 × 512, and 10 images of the size 256 × 256 are selected from SIPI datasets. The average of entropy values for each image size obtained using our proposed algorithm is listed in Table 3. Then, the results are compared with methods [44–48]. All entropy values for the chipper images that encrypted with the new method approached 8. The chipper images encrypted using the proposed encryption method have the highest randomness.

Table 1. Entropy values of images with size 512 × 512 with our algorithm and other encryption algorithms.

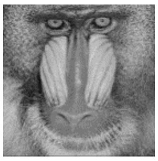
						**Average**
Plain image	7.3579	7.5585	7.1914	6.6776	7.4451	-
Proposed	7.999295	7.999267	7.999284	7.999312	7.999333	7.9993
Hua et al. [44]	7.9991	7.9993	7.9993	7.9993	7.9992	7.9992
Wu et al. [45]	7.9992	7.9993	7.9994	7.9992	7.9994	7.9993
Li et al. [46]	7.9922	7.9921	7.9924	7.9925	7.9924	7.9923
Niyat et al. [47]	7.9990	7.9990	7.9992	7.9991	7.9995	7.9992
Enayatifar et al. [48]	7.9981	7.9983	7.9988	7.9991	7.9994	7.9987

Table 2. Entropy values of images with size 256 × 256 with our algorithm and other encryption algorithms.

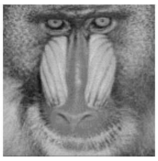
						**Average**
Plain image	7.2279	7.5625	7.1583	6.7045	7.4311	-
Proposed	7.9975	7.9970	7.9976	7.9972	7.9972	7.9973
Hua et al. [44]	7.9974	7.9971	7.9974	7.9971	7.9976	7.9973
Wu et al. [45]	7.9971	7.9974	7.9971	7.9970	7.9976	7.9972
Li et al. [46]	7.9912	7.9909	7.9907	7.9912	7.9913	7.9911
Niyat et al. [47]	7.9970	7.9972	7.9973	7.9970	7.9974	7.9972
Enayatifar et al. [48]	7.9938	7.9958	7.9941	7.9974	7.9975	7.9957

Table 3. Comparison of average entropy values between our algorithm and other encryption algorithms.

Image Size	**Proposed**	**Hua et al. [44]**	**Wu et al. [45]**	**Li et al. [46]**	**Niyat et al. [47]**	**Enayatifar et al. [48]**
512 × 512	7.9992	7.9992	7.9993	7.992	7.9991	7.9984
256 × 256	7.9973	7.9973	7.9973	7.9911	7.9972	7.9954

4.2. Correlation Coefficient

Generally, the input images' adjacent pixels have a high correlation in the diagonal, horizontal, and vertical directions. A successful encryption algorithm must minimize this correlation. Any two neighboring pixels, x and y, have the following correlation coefficient:

$$r_{x,y} = \frac{E((y - E(y))(x - E(x)))}{\sqrt{D(y)D(x)}} \tag{14}$$

$$E(x) = \frac{1}{T}\sum_{i=1}^{T} x_i \tag{15}$$

$$D(x) = \frac{1}{T}\sum_{i=1}^{T} (x_i - E(x))^2 \tag{16}$$

where the integer T refers to the total number of adjoining pixels; $D(x)$ and $E(x)$ are the variance and expectation of x, respectively. In the successfully encrypted image, the correlation between adjoining pixels should approach 0.

In this experiment, nearby pixels are grouped in pairs, where 40,000 of these pairs are randomly selected, then the correlation coefficients computed for the three directions.

Tables 4 and 5 shows the encrypted images' calculated correlation coefficients' absolute values using the new and existing image encryption algorithms [44–48]. The average coefficient correlations for the new encryption algorithm are very close to 0. All the results confirm that our proposed algorithm can remove the correlation between adjacent pixels in the encrypted image.

Table 4. Correlation coefficients in three directions: Horizontal (H), Vertical (V), and Diagonal (D) for images with the size of 512 × 512.

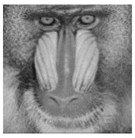
Method							**Average**
Proposed	H	0.0251	0.0044	0.0041	0.0269	0.0019	0.0125
	V	0.0040	0.0077	0.0011	0.0196	0.0069	0.0079
	D	0.0231	0.0067	0.0113	0.0260	0.0200	0.0174
Hua et al. [44]	H	0.0132	0.0100	0.0214	0.0154	0.0250	0.017
	V	0.0150	0.0038	0.0330	0.0089	0.0116	0.0145
	D	0.0309	0.0321	0.0213	0.0031	0.0025	0.01742
Wu et al. [45]	H	0.0029	0.0006	0.0003	0.0025	0.0032	0.0019
	V	0.0033	0.0038	0.0034	0.0050	0.0016	0.0034
	D	0.0062	0.0010	0.0011	0.0012	0.0023	0.0024
Li et al. [46]	H	0.0065	0.0021	0.0078	0.0033	0.0018	0.0043
	V	0.0047	0.0143	0.0095	0.0067	0.0026	0.0076
	D	0.0017	0.0029	0.024	0.0015	0.0022	0.0065
Niyat et al. [47]	H	0.0027	0.0046	0.0018	0.0024	0.0018	0.0027
	V	0.0021	0.0025	0.0083	0.0104	0.0112	0.0069
	D	0.0042	0.0034	0.0014	0.0015	0.0054	0.0032
Enayatifaret al. [48]	H	0.0016	0.0053	0.0071	0.0037	0.0008	0.0037
	V	0.0048	0.0138	0.0095	0.0014	0.0021	0.0063
	D	0.0024	0.0019	0.014	0.0008	0.0005	0.0039

Table 5. Correlation coefficients in three directions: Horizontal (H), Vertical (V), and Diagonal (D) for images with size 256 × 256.

Method							**Average**
Proposed	H	0.0065	0.0211	0.0138	0.0229	0.0069	0.0142
	V	0.0337	0.0129	0.0093	0.0103	0.0479	0.0228
	D	0.0244	0.0013	3.4412×10^{-6}	0.0100	0.0075	0.0088
Hua et al. [44]	H	0.0113	0.0196	0.0014	0.0055	0.0074	0.009
	V	0.00051	0.0165	0.0181	0.0014	0.0096	0.00922
	D	0.0136	0.0210	0.0066	0.0083	0.0193	0.0138
Wu et al. [45]	H	0.0026	0.0016	0.0001	0.0028	0.0056	0.0025
	V	0.0009	0.0059	0.0031	0.0041	0.0037	0.0035
	D	0.0052	0.0034	0.0015	0.0010	0.0032	0.0029
Li et al. [46]	H	0.0055	0.0021	0.0073	0.0075	0.0041	0.0053
	V	0.0015	0.0218	0.0216	0.0084	0.0021	0.0111
	D	0.0041	0.0096	0.0035	0.0011	0.0009	0.0038
Niyat et al. [47]	H	0.0060	0.0049	0.0085	0.0054	0.0061	0.0062
	V	0.0058	0.0031	0.0092	0.0089	0.0116	0.0077
	D	0.0016	0.0079	0.0024	0.0021	0.0018	0.0032
Enayatifaret al. [48]	H	0.0059	0.0037	0.0073	0.0062	0.0023	0.0051
	V	0.0041	0.0258	0.0109	0.0074	0.0019	0.01
	D	0.0028	0.0079	0.0016	0.0009	0.0011	0.0029

4.3. Differential Attack

In this attack, the attacker aims to decrypt the encrypted images without using the key through determining the relation between original and encrypted images. Therefore, small pixel changes in the original image significantly affect the encrypted image, making it more difficult for attackers to crack the encrypted image. Successful algorithms for image encryption must resist this attack. Robustness to this attack based on the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI):

$$\mathrm{NPCR} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \mathrm{DIF}(i,j) \times 100(\%) \tag{17}$$

$$\mathrm{UACI} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_2(i,j) - C_1(i,j)|}{255} \times 100(\%) \tag{18}$$

with

$$\mathrm{DIF}(i,j) = \begin{cases} 0\,, & C_2(i,j) = C_1(i,j), \\ 1\,, & C_2(i,j) \neq C_1(i,j), \end{cases} \tag{19}$$

The symbol C_2 refers to the chipper image that encrypted from the original image by changing only one pixel, while C_1 refers to the chipper image encrypted from the same plain image.

Table 6 shows the computed values of the five gray images encrypted using the proposed and the existing image encryption algorithms [44–48]. In addition, the average values of NPCR and UACI of the images selected from SIPI datasets are presented in Table 7. To confirm the efficiency of our algorithm, the results are compared with other methods [44–48].

Table 6. Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) of the encrypted image using different encryption algorithms.

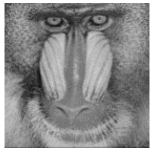
Size	**Method**							**Pass Rate**
512 × 512	Proposed	NPCR	99.6075	**99.5876**	99.6101	99.6082	99.6174	19/20
		UACI	33.4742	33.4012	33.4688	33.4585	33.4322	
256 × 256		NPCR	99.5941	99.6033	99.6078	99.6017	99.6246	
		UACI	33.4610	33.4274	33.4188	33.5053	33.4226	
512 × 512	Hua et al. [44]	NPCR	99.5995	99.6128	99.6029	99.6181	99.5960	-
		UACI	33.5250	33.5513	33.4745	33.4384	33.4858	18/20
256 × 256		NPCR	99.6307	99.6231	**99.5682**	99.6231	99.5850	-
		UACI	33.4534	**33.6805**	33.3633	33.4665	33.5582	-
512 × 512	Wu et al. [45]	NPCR	99.5903	99.6112	99.6124	99.6261	99.6002	17/20
		UACI	33.5281	33.5265	**33.5891**	**33.5782**	33.5079	-
256 × 256		NPCR	99.5925	99.6078	99.6170	99.6231	99.6200	-
		UACI	33.3822	33.4953	**33.6609**	33.6358	33.4169	-
256 × 256	Li et al. [46]	NPCR	$\mathbf{0.0862 \times 10^{-4}}$	$\mathbf{0.0862 \times 10^{-4}}$	$\mathbf{0.0862 \times 10^{-4}}$	$\mathbf{0.0862 \times 10^{-4}}$	$\mathbf{0.0862 \times 10^{-4}}$	-
		UACI	$\mathbf{1.9946 \times 10^{-6}}$	$\mathbf{1.9946 \times 10^{-6}}$	$\mathbf{1.9946 \times 10^{-6}}$	$\mathbf{1.9946 \times 10^{-6}}$	$\mathbf{1.9946 \times 10^{-6}}$	-
512 × 512	Niyat et al. [47]	NPCR	99.5966	99.6202	99.6057	**99.4350**	99.6152	-
		UACI	33.5016	33.5323	33.4726	33.5317	33.5024	19/20
256 × 256		NPCR	99.6081	99.6601	99.6154	99.6532	99.6217	-
		UACI	33.4125	33.4415	33.5057	33.5098	33.4159	-
512 × 512	Enayatifar et al. [48]	**NPCR**	**99.2394**	**99.3017**	**99.2918**	**99.4883**	99.6304	-
		UACI	**33.3144**	**33.0026**	**32.4162**	**33.3562**	**33.5989**	3/20
256 × 256		**NPCR**	**99.1051**	**98.4975**	**99.25**	**99.4176**	**99.5193**	-
		UACI	**33.2517**	**32.9483**	33.3928	33.5254	33.5851	-

Table 7. Comparison of average values of NPCR and UACI.

	Image Size	Proposed	Hua et al. [44]	Wu et al. [45]	Niyat et al. [47]	Enayatifar et al. [48]
NPCR	512 × 512	99.6087	99.6162	99.6081	**99.58553**	**99.35947**
	256 × 256	99.6124	99.5925	99.61238	99.63142	**99.13223**
UACI	512 × 512	33.4678	33.4696	33.54787	33.49192	**33.18788**
	256 × 256	33.4797	33.4321	33.52045	33.4595	33.30098

As mentioned in [49], the critical values of NPCR and UACI are N_α^* and u_α, respectively, which are calculated as follows:

$$N_\alpha^* = \frac{\left(G -^{-1}(\alpha)\sqrt{\frac{G}{MN}}\right)}{G+1} \tag{20}$$

$$u_\alpha^{*-} = \mu_u -^{-1}\left(\frac{\alpha}{2}\right)\sigma_u \tag{21}$$

$$u_\alpha^{*+} = \mu_u +^{-1}\left(\frac{\alpha}{2}\right)\sigma_u \tag{22}$$

$$\mu_u = \frac{G+2}{3G+3} \tag{23}$$

$$\sigma_u = \frac{(G+2)\left(G^2+2G+3\right)}{18(G+1)^2 GR} \tag{24}$$

To resist the differential attacks, the value of NPCR for the encrypted image should be larger than N_α^*, and the value of UACI should be in the range of $(u_\alpha^{*-}, u_\alpha^{*+})$. When significant level α = 0.05, then $N_\alpha^* = 99.5693\%$ and $(u_\alpha^{*-}, u_\alpha^{*+}) = (33.2824\%\ ,\ 33.6447\%)$ for the image with size 256 × 256. However, when the size of the image is 512 × 512, the N_α^* is 99.5893% and $(u_\alpha^{*-}, u_\alpha^{*+}) = (33.3730\%,\ 33.5541\%)$. In Tables 6 and 7, the values that did not pass the test are displayed in bold. Our proposed algorithm achieves the highest pass rate compared to other methods, reflecting excellent robustness of the differential attack.

4.4. Noise and Data Cut Attacks

When images are transmitted over the network, they are vulnerable to noise or cropping (data cut). Successful image encryption algorithms should have robustness against noise and cropping attacks. The well-known measure, PSNR (peak signal to noise ratio), is used to evaluate the decrypted image quality. Mathematically, for original and decrypted images, I_O and I_D, the PSNR is:

$$\text{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\text{MSE}}\right)\ (\text{db}), \tag{25}$$

where MSE refers to the mean square error:

$$\text{MSE} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|I_O(i,j) - I_D(i,j)\right|^2 \tag{26}$$

A higher PSNR value reflects high image quality. For a PSNR $>$ 35, original and decrypted images are indistinguishable.

This experiment was performed to test robustness against noise and data cut attacks. In this experiment, an encrypted image is contaminated with "salt and peppers" noise of 2 different levels, 0.002 and 0.005, decrypted using the new method. The encrypted images were also attacked by a data cut of 128 × 128 and 64 × 64 and then decrypted using the new algorithm. The PSNR for the five tested images with noise and data cut with a size of 512 × 512 is shown in Table 8.

Table 8. Peak signal to noise ratio (PSNR) (dB.) values for noise and data cut attacks.

Standard Grayscale Images	Lena	Baboon	Peppers	Boat	Airplane
Salt and Pepper with noise level 0.002	28.2751	30.5936	29.6619	30.4091	29.3654
Salt and Pepper with noise level 0.005	24.4812	26.6862	25.5269	26.5629	25.0544
Data cut with block size 128 × 128	16.7418	18.5936	17.5580	18.3569	17.1245
Data cut with block size 64 × 64	22.7112	24.5162	23.6008	24.4023	23.1438

The new algorithm is robust against "salt and peppers" noise with density 0.002, where all values of PSNR are approaching 30db. When the level of noise increased to 0.005, the average value of PSNR decreased to 25.6db. For the data cut off size 64 × 64, the PSNR values are around 24db, and the decrypted image's content is visible. Moreover, when the encrypted image is attacked with the data cut off size 128 × 128, a relatively big cut off (i.e., the encrypted image lost 1/8 information), the PSNR is decreased to 18dB. Despite the reduction in PSNR values, the decrypted image is recognizable.

Figure 2 shows the noise and data cut attacks for an encrypted image, demonstrating that the reader can easily recognize the decrypted images' content in different cases (i.e., noise, data cut). Therefore, the new algorithm is durable and resistant to these attacks.

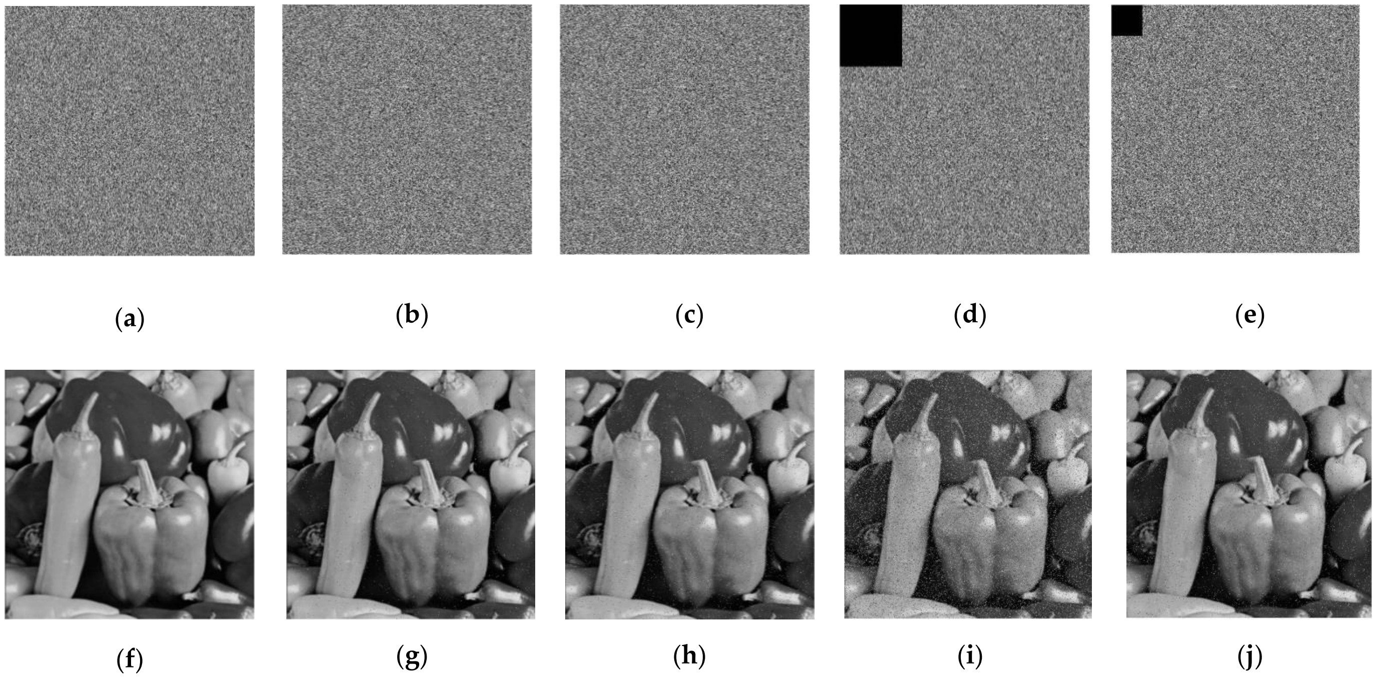


Figure 2. (**a**) The encrypted image, (**b**) noisy encrypted image with 0.002, (**c**) noisy encrypted image with 0.005 and (**d**) encrypted image with 128 × 128 data cut. (**e**) Encrypted image with 64 × 64 data cut. (**f**–**j**) Decrypted images of (**a**–**e**).

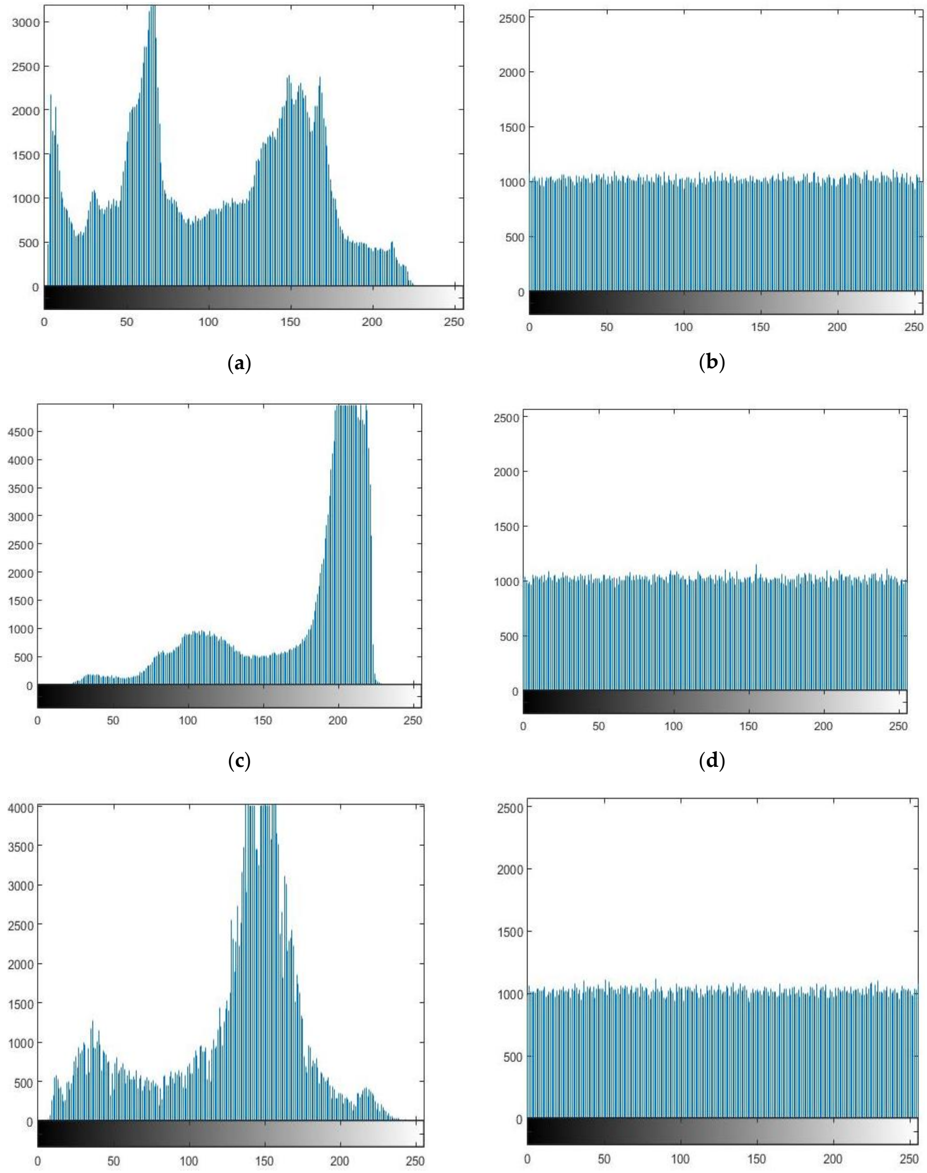
4.5. Histograms

Visual representation of image pixels distribution is called "Image Histogram," used to evaluate image encryption algorithms. A successful algorithm for image encryption must generate a flat histogram for the encrypted image.

Three standard gray images, Peppers, Airplane, and Boat, encrypted using the new algorithm. The histogram of the original and encrypted images displayed in Figure 3. Based on the distinguishable contents of the original images, their histograms are different. On the other side, the encrypted images have very similar and uniform histograms. Attackers are not able to recover the original images from encrypted image histograms. To ensure the uniform distribution of the histogram, the chi-square test is calculated by the following equation:

$$\chi^2 = \sum_{i=1}^{256} \frac{(O_i - EV)^2}{EV} \tag{27}$$

where O_i refers to the recurrence rate of the grey value i; $EV = O/256$ is the expected frequency of each grey value. Assume a significant level of 0.05, $\chi^2(255.05) = 293.2478$. The histogram of the encrypted image is considered to be uniform if the value of χ^2 is less than 293. Here we calculate the χ^2 for the encrypted images and record the results in Table 9. All values in Table 9 are less than 293, so the histograms of images encrypted using the proposed algorithm have uniform distribution. These results ensure the efficiency of the new algorithm.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 3. Histograms of original and encrypted images: (**a**) original "Peppers," (**b**) encrypted "Peppers," (**c**) original "Airplane," (**d**) encrypted "Airplane," (**e**) original "Boat," and (**f**) encrypted "Boat".

Table 9. Chi-square test.

	Lena	Baboon	Peppers	Boat	Airplane
512 × 512	242.9590	255.6563	266.0371	259.4941	250.1230
256 × 256	264.8750	224.2578	268.4766	219.9297	253.9063

4.6. Keyspace

The keyspace size is crucial in the encryption process. The encryption algorithm is robust to brute force attacks if its keyspace size >2^{100}. The proposed encryption algorithm has different security keys: x_1, x_2, x_3, x_4, x_5, x_6, N_0, a, b, c, d, e, and r. If we assume the accuracy of the initial value equals to 10^{16}, then the total keyspace is larger than $N_0 \times 10^{96}$, which shows robustness to the brute force attack.

4.7. Key Sensitivity

Successful image encryption algorithms must show high sensitivity to the secrete keys, which results in a noticeable change in a decrypted image with minimal modifications in initial conditions of the utilized secrete key used in the encryption process. An experiment was performed to test the key sensitivity of the new algorithm. The original image of "Lena" encrypted using the initial conditions (0.1, 0.1, 0.1, 0.1, 0.1, and 0.1). Figure 4a,b show the original and encrypted images of Lena.

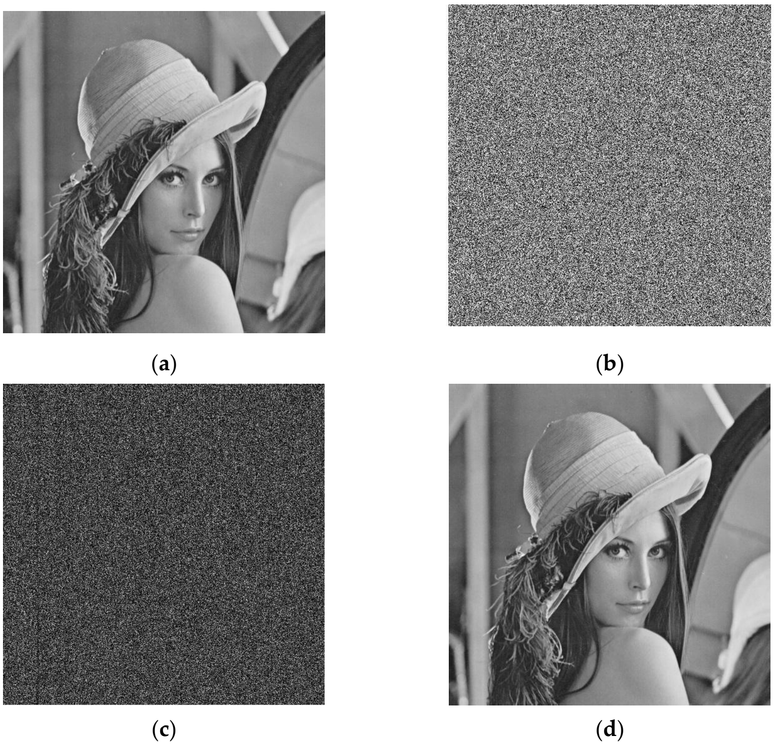

(**a**)
(**b**)
(**c**)
(**d**)

Figure 4. Key sensitivity: (**a**) original "Lena," (**b**) encrypted "Lena" with the original initial conditions, (**c**) decrypted "Lena" with the modified key, and (**d**) decrypted "Lena" with the original key.

The key is modified with only one-bit difference (0.1, 0.1, 0.1, 0.1, 0.1, and 0.1000001). The decryption process with the modified key failed to restore the original image, as shown in Figure 4c. On the other side, decryption using the original secret key successfully recovered the original image, as displayed in Figure 4d.

4.8. NIST Statistical Test

A good encryption algorithm should produce an encrypted image with high randomness. The NIST statistical test suite provides statistical tests to respect the randomness of the sequence generated with the encryption algorithm. The significance level is set to 0.01 for all tests in NIST. In this experiment, we calculate p-values for encrypted peppers image of a size 512 × 512, which is being changed into a binary sequence. Then we record the results for different statistical tests in Table 10. The p-values ≥ 0.01 and indicates the randomness of the binary sequence. From the results, we can see that the sequence generated using the proposed algorithm passed all tests, which assures the randomness of the binary sequence.

Table 10. NIST statistical test.

Test Name	p-Value	Conclusion
Frequency	0.4718	Random
Block-frequency	0.2090	Random
Runs	0.9161	Random
Longest run	0.1318	Random
Discrete Fourier Transform Test	0.8831	Random
Non-overlapping template	0.7616	Random
Cumulative sums (forward)	0.3469	Random
Cumulative sums (reverse)	0.7083	Random

4.9. Computational Complexity

The steps required to perform the encryption process are used to measure the computational complexity of the algorithm. For the plain image of size $M \times N$, the time complexity of the confusion steps in the proposed algorithm is $O(M \times N)$. Regarding the diffusion step, the time complexity is $O((M \times N)/Bs)$, where Bs is the number of blocks in the image. Therefore, the total time complexity of the proposed algorithm is $O(M \times N)$.

5. Conclusions

The authors proposed a new algorithm for gray image encryption. In this algorithm, the Fibonacci Q-matrix is integrated with a 6D hyperchaotic system. First, we generate random sequences using a 6D hyperchaotic system, and we select three of these sequences to change the pixel position. Then, we use the Fibonacci Q-matrix with $n = 10$ to change the pixels value for each sub-block (size(2 × 2)) of the shuffled image. Double confusion/diffusion operations are applied to increase the security level.

The new algorithm is sensitive to minimal modifications in pixel distribution, and the secret key, where an entirely different encrypted image, is obtained. Therefore, the proposed algorithm successfully resists the differential attack. The new algorithm resists a brute force attack where the keyspace size is large enough. Moreover, the new algorithm's security performance was evaluated using information entropy, correlation coefficients, noise, and data cut attack and histogram. The new algorithm can encrypt gray images with high-security levels. In the future, we will study the effectiveness of our algorithm in encrypting color images.

Author Contributions: Conceptualization, K.M.H.; methodology, K.M.H. and M.M.D.; software, S.T.K. and M.M.D.; validation, S.T.K., M.M.D. and G.A.P.; investigation, K.M.H., S.T.K., M.M.D. and G.A.P.; data curation, S.T.K. and M.M.D.; writing—original draft preparation, K.M.H., S.T.K. and M.M.D.; writing—review and editing, K.M.H., M.M.D. and G.A.P.; visualization, S.T.K.; supervision, K.M.H. and G.A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the MPhil program "Advanced Technologies in Informatics and Computers," hosted by the Department of Computer Science, International Hellenic University, Kavala, Greece.

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

References

1. Abdel-Aziz, M.M.; Hosny, K.M.; Lashin, N.A. Improved data hiding method for securing color images. *Multimed. Tools Appl.* **2021**, *80*, 12641–12670. [CrossRef]
2. Li, N.; Huang, F. Reversible data hiding for JPEG images based on pairwise nonzero AC coefficient expansion. *Signal Process.* **2020**, *171*, 107476. [CrossRef]
3. Hosny, K.M.; Darwish, M.M.; Li, K.; Salah, A. Parallel Multi-Core CPU and GPU for Fast and Robust Medical Image Watermarking. *IEEE Access* **2018**, *6*, 77212–77225. [CrossRef]
4. Hosny, K.M.; Darwish, M.M. Robust color image watermarking using invariant quaternion Legendre-Fourier moments. *Multimed. Tools Appl.* **2018**, *77*, 24727–24750. [CrossRef]
5. Hosny, K.M.; Darwish, M.M. Invariant image watermarking using accurate Polar Harmonic Transforms. *Comput. Electr. Eng.* **2017**, *62*, 429–447. [CrossRef]
6. Hosny, K.M.; Darwish, M.M. Resilient Color Image Watermarking Using Accurate Quaternion Radial Substituted Chebyshev Moments. *ACM Trans. Multimed. Comput. Commun. Appl.* **2019**, *15*, 1–25. [CrossRef]
7. Molina-Garcia, J.; Garcia-Salgado, B.P.; Ponomaryov, V.; Reyes-Reyes, R.; Sadovnychiy, S.; Cruz-Ramos, C. An effective fragile watermarking scheme for color image tampering detection and self-recovery. *Signal Process. Image Commun.* **2020**, *81*, 115725. [CrossRef]
8. Laiphrakpam, D.S.; Khumanthem, M.S. Medical image encryption based on improved ElGamal encryption technique. *Optik* **2017**, *147*, 88–102. [CrossRef]
9. Li, Y.; Yu, H.; Song, B.; Chen, J. Image encryption based on a single-round dictionary and chaotic sequences in cloud computing. *Concurr. Comput. Pract. Exp.* **2021**, *33*, 1. [CrossRef]
10. Artiles, J.A.; Chaves, D.P.; Pimentel, C. Image encryption using block cipher and chaotic sequences. *Signal Process. Image Commun.* **2019**, *79*, 24–31. [CrossRef]
11. Wang, X.; Gao, S. Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. *Inf. Sci.* **2020**, *507*, 16–36. [CrossRef]
12. Luo, Y.; Zhou, R.; Liu, J.; Cao, Y.; Ding, X. A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map. *Nonlinear Dyn.* **2018**, *93*, 1165–1181. [CrossRef]
13. He, Y.; Zhang, Y.-Q.; Wang, X.-Y. A new image encryption algorithm based on two-dimensional spatiotemporal chaotic system. *Neural Comput. Appl.* **2018**, *32*, 247–260. [CrossRef]
14. Irani, B.Y.; Ayubi, P.; Jabalkandi, F.A.; Valandar, M.Y.; Barani, M.J. Digital image scrambling based on a new one-dimensional coupled Sine map. *Nonlinear Dyn.* **2019**, *97*, 2693–2721. [CrossRef]
15. Zhang, Y. The image encryption algorithm based on chaos and DNA computing. *Multimed. Tools Appl.* **2018**, *77*, 21589–21615. [CrossRef]
16. Chai, X.; Gan, Z.; Yuan, K.; Chen, Y.; Liu, X. A novel image encryption scheme based on DNA sequence operations and chaotic systems. *Neural Comput. Appl.* **2019**, *31*, 219–237. [CrossRef]
17. Xuejing, K.; Zihui, G. A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. *Signal Process. Image Commun.* **2020**, *80*, 115670. [CrossRef]
18. Zhou, N.; Hu, Y.; Gong, L.; Li, G. Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations. *Quantum Inf. Process.* **2017**, *16*, 164. [CrossRef]
19. El-Latif, A.A.A.; Abd-El-Atty, B.; Abou-Nassar, E.M.; Venegas-Andraca, S.E. Controlled alternate quantum walks based privacy preserving healthcare images in Internet of Things. *Opt. Laser Technol.* **2020**, *124*, 105942. [CrossRef]
20. Zhang, D.; Liao, X.; Yang, B.; Zhang, Y. A fast and efficient approach to color-image encryption based on compressive sensing and fractional Fourier transform. *Multimed. Tools Appl.* **2018**, *77*, 2191–2208. [CrossRef]
21. Ye, G.; Pan, C.; Dong, Y.; Shi, Y.; Huang, X. Image encryption and hiding algorithm based on compressive sensing and random numbers insertion. *Signal Process.* **2020**, *172*, 107563. [CrossRef]
22. Li, Y.; Wang, C.; Chen, H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Opt. Lasers Eng.* **2017**, *90*, 238–246. [CrossRef]
23. Chai, X.; Fu, X.; Gan, Z.; Lu, Y.; Chen, Y. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process.* **2019**, *155*, 44–62. [CrossRef]
24. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [CrossRef]
25. Ullah, A.; Jamal, S.S.; Shah, T. A novel scheme for image encryption using substitution box and chaotic system. *Nonlinear Dyn.* **2018**, *91*, 359–370. [CrossRef]
26. Pak, C.; An, K.; Jang, P.; Kim, J.; Kim, S. A novel bit-level color image encryption using improved 1D chaotic map. *Multimed. Tools Appl.* **2018**, *78*, 12027–12042. [CrossRef]
27. Cao, C.; Sun, K.; Liu, W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. *Signal Process.* **2018**, *143*, 122–133. [CrossRef]
28. Li, Z.; Peng, C.; Li, L.; Zhu, X. A novel plaintext-related image encryption scheme using hyper-chaotic system. *Nonlinear Dyn.* **2018**, *94*, 1319–1333. [CrossRef]

29. Gong, L.; Deng, C.; Pan, S.; Zhou, N. Image compression-encryption algorithms by combining hyper-chaotic system with discrete fractional random transform. *Opt. Laser Technol.* **2018**, *103*, 48–58. [CrossRef]
30. Li, M.; Wang, P.; Liu, Y.; Fan, H. Cryptanalysis of a Novel Bit-Level Color Image Encryption Using Improved 1D Chaotic Map. *IEEE Access* **2019**, *7*, 145798–145806. [CrossRef]
31. Hu, X.; Wei, L.; Chen, W.; Chen, Q.; Guo, Y. Color Image Encryption Algorithm Based on Dynamic Chaos and Matrix Convolution. *IEEE Access* **2020**, *8*, 12452–12466. [CrossRef]
32. Chen, X.; Hu, C.-J. Adaptive medical image encryption algorithm based on multiple chaotic mapping. *Saudi J. Biol. Sci.* **2017**, *24*, 1821–1827. [CrossRef] [PubMed]
33. Chai, X.; Zheng, X.; Gan, Z.; Han, D.; Chen, Y. An image encryption algorithm based on chaotic system and compressive sensing. *Signal Process.* **2018**, *148*, 124–144. [CrossRef]
34. Chai, X.; Fu, X.; Gan, Z.; Zhang, Y.; Lu, Y.; Chen, Y. An efficient chaos-based image compression and encryption scheme using block compressive sensing and elementary cellular automata. *Neural Comput. Appl.* **2018**, *32*, 4961–4988. [CrossRef]
35. Tsafack, N.; Kengne, J.; Abd-El-Atty, B.; Iliyasu, A.M.; Hirota, K.; El-Latif, A.A.A. Design and implementation of a simple dynamical 4-D chaotic circuit with applications in image encryption. *Inf. Sci.* **2020**, *515*, 191–217. [CrossRef]
36. Ramasamy, P.; Ranganathan, V.; Kadry, S.; Damaševičius, R.; Blažauskas, T. An Image Encryption Scheme Based on Block Scrambling, Modified Zigzag Transformation and Key Generation Using Enhanced Logistic—Tent Map. *Entropy* **2019**, *21*, 656. [CrossRef] [PubMed]
37. Zheng, J.; Liu, L. Novel image encryption by combining dynamic DNA sequence encryption and the improved 2D logistic sine map. *IET Image Process.* **2020**, *14*, 2310–2320. [CrossRef]
38. Kari, A.P.; Navin, A.H.; Bidgoli, A.M.; Mirnia, M. A new image encryption scheme based on hybrid chaotic maps. *Multimed. Tools Appl.* **2021**, *80*, 2753–2772. [CrossRef]
39. Liu, L.; Lei, Y.; Wang, D. A Fast Chaotic Image Encryption Scheme with Simultaneous Permutation-Diffusion Operation. *IEEE Access* **2020**, *8*, 27361–27374. [CrossRef]
40. Liu, H.; Kadir, A.; Liu, J. Color pathological image encryption algorithm using arithmetic over Galois field and coupled hyper chaotic system. *Opt. Lasers Eng.* **2019**, *122*, 123–133. [CrossRef]
41. Yu, S.-S.; Zhou, N.-R.; Gong, L.-H.; Nie, Z. Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system. *Opt. Lasers Eng.* **2020**, *124*, 105816. [CrossRef]
42. Wang, J.; Yu, W.; Wang, J.; Zhao, Y.; Zhang, J.; Jiang, D. A new six-dimensional hyperchaotic system and its secure communication circuit implementation. *Int. J. Circuit Theory Appl.* **2019**, *47*, 702–717. [CrossRef]
43. Zhou, T.; Shen, J.; Li, X.; Wang, C.; Tan, H. Logarithmic encryption scheme for cyber–physical systems employing Fibonacci Q-matrix. *Future Gener. Comput. Syst.* **2020**, *108*, 1307–1313. [CrossRef]
44. Hua, Z.; Zhou, Y.; Huang, H. Cosine-transform-based chaotic system for image encryption. *Inf. Sci.* **2019**, *480*, 403–419. [CrossRef]
45. Wu, J.; Liao, X.; Yang, B. Image encryption using 2D Hénon-Sine map and DNA approach. *Signal Process.* **2018**, *153*, 11–23. [CrossRef]
46. Li, C.; Luo, G.; Qin, K.; Li, C. An image encryption scheme based on chaotic tent map. *Nonlinear Dyn.* **2017**, *87*, 127–133. [CrossRef]
47. Niyat, A.Y.; Moattar, M.H.; Torshiz, M.N. Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Opt. Lasers Eng.* **2017**, *90*, 225–237. [CrossRef]
48. Enayatifar, R.; Abdullah, A.H.; Isnin, I.F.; Altameem, A.; Lee, M. Image encryption using a synchronous permutation-diffusion technique. *Opt. Lasers Eng.* **2017**, *90*, 146–154. [CrossRef]
49. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. JSAT* **2011**, *1*, 31–38.