electronics

MDPI

Article

Correcting Errors in Color Image Encryption Algorithm Based on Fault Tolerance Technique

Heba G. Mohamed [1,2,*], Fadwa Alrowais [3] and Dalia H. ElKamchouchi [2,4]

[1] Electrical Department, College of Engineering, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[2] Electrical Department, College of Engineering, Alexandria Higher Institute of Engineering and Technology, Alexandria 21421, Egypt; dhelkamchouchi@pnu.edu.sa
[3] Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 84428, Saudi Arabia; fmalrowais@pnu.edu.sa
[4] Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
* Correspondence: hegmohamed@pnu.edu.sa

Abstract: Security standards have been raised through modern multimedia communications technology, which allows for enormous progress in security. Modern multimedia communication technologies are concerned with fault tolerance technique and information security. As a primary method, there is widespread use of image encryption to protect image information security. Over the past few years, image encryption has paid more attention to combining DNA technologies in order to increase security. The objective here is to provide a new method for correcting color image encryption errors due to the uncertainty of DNA computing by using the fractional order hyperchaotic Lorenz system. To increase randomness, the proposed cryptosystem is applied to the three plain image channels: Red, Green, and Blue. Several methods were compared including the following: entropy, correlation, key sensitivity, key space, data loss attacks, speed computation, Number of Pixel changing rate (NPCR), and Unified Average Change Intensity randomness (UACI) tests. Consequently, the proposed scheme is very secure against a variety of cryptographic attacks.

Keywords: image encryption; information security; fault tolerance; fractional order chaotic map; multimedia; cryptography; DNA computing

Citation: Mohamed, H.G.; Alrowais, F.; ElKamchouchi, D.H. Correcting Errors in Color Image Encryption Algorithm Based on Fault Tolerance Technique. *Electronics* **2021**, *10*, 2890. https://doi.org/10.3390/electronics10232890

Academic Editors: Christos Volos, Lazaros Moysis, Denis Butusov and Ahmed Radwan

Received: 29 October 2021
Accepted: 19 November 2021
Published: 23 November 2021

1. Introduction

Nowadays, digital image encryption is an evolving technique in the digital communication network arena. Security concerns arise when these images are required to be transmitted or stored over numerous networks. Additionally, digital images are crucial concerns in many uses, such as military information, telemedicine applications and cloud computing. Therefore, ensuring the security, honesty and strength of digital medical imaging has become an imperative topic for numerous researchers [1–3]. Distinctive properties between image and text, such as large data space and strong interconnection within pixels, make some conventional encryption techniques inconvenient for digital image encryption. Images are not well suited to encode algorithms used for text data, and their encryption algorithms can lead to problems such as leakage of contour information, low efficiency, etc. [4,5].

The chaotic system has been famous for the sensitivity of primary conditions and parameters, pseudo randomness, ergodicity and reproduction, which was first proposed by Lorenz [6]; it has also been appropriate for digital image encryption. Several chaos-based on image encryption have been introduced since Matthews introduced the first algorithm for image encryption based entirely on chaos [7]. Chaotic systems are generally used to generate a pseudorandom key from a key or plaintext. When generated from

a pseudorandom key, a stream cipher is produced, while when generated from plain text, a block cipher is produced. Images are encrypted chaotically through confusion and diffusion. Pixels of the image are scrambled in the confusion level using a secret key according to the control parameters. While in the diffusion stage, chaotically generated sequences are used to change pixels' values. These techniques make chaotic encryption extremely secure. In recent years, numerous chaotic image encryption schemes using the integral order chaotic systems have been overtly (publicly) proposed [8–13].

Fractional-Order Hyperchaotic Systems (FOHS) exhibit higher nonlinearity and degrees due to their geometrical interpretation of fractional derivatives included in the expressions for nonlocal effects in either space or time [14,15]. This means that this kind of chaotic system has great ability to protect sensitive information. As a counterpart to stream ciphers (RC4, Spritz, Salsa, etc.) and in addition to pseudo randomness, FOHS exhibits extreme sensitivity to primary values and parameter settings, as well as ergodicity and unpredictability, making it ideal for image encryption. FOHS presents many advantages over stream ciphers since the substitution and diffusion primitives of chaotic maps change based on initial conditions. Thus, fractional order hyperchaotic systems can play an important role in information security.

Wang et al. [15] applied the FOHS for securing color images by embedding the system parameters as well as the derivative order into the system. In [16,17] Wu et al. and Zhao et al., respectively applied 3D FOHS and Chen chaotic systems in ciphering their color images. While Huang et al. applied 4D FOHS based on neural network scheme for encrypting color images, where the measurements proved the efficacy of the scheme [18].

In DNA, there is enormous parallelism and an extremely high information density, which make DNA cryptography an excellent tool for securing end-to-end communication. Adleman [19] completed a DNA computing experiment. The DNA cryptography algorithm has been incorporated into numerous image encryption systems. DNA cryptography have extensive use of the advances of the DNA molecules, as extreme-high storage bulk, extreme-low energy consumption, and the potential of ultra-large-scale parallel computing to achieve the cryptographic functions of information encryption. Gehani et al. [20] set the DNA cryptography foundation using molecular approach and one-time pad concept, which has perfect privacy. Later then, after Gehani approach, numerous image encryption algorithms based on DNA cryptography emerged among the public [21–26]. Zhang et al. [27] proposed an algorithm, which disturb the locations and values of the pixels using chaotic system and applying DNA cryptography, where the pseudo-DNA operations are controlled by the quaternary chaotic sequences. Based on DNA computing, Xie et al. [28] concluded that an image encryption system would not be secure if only a scramble processing step was implemented. Liu et al. [29] broke the encryption system based on DNA computing by applying chosen plain-images, and their cryptosystem-retrieved cipher-images have to be capable of resisting differential attack.

Secure protocols and standards based on cryptography need various computations and transmissions. As a result, faults are inevitable. Previous studies have proposed various strategies to detect and correct one or several errors. In particular, research focuses on the fault-tolerant techniques of block ciphers and public-key cryptography. Depending on the structures of the algorithms, the Algorithm Based Fault Tolerant (ABFT) technique provides a common process for designing fault tolerant structures by altering the algorithm computation in order to achieve additional data for discovering and correcting error [30–33]. The ABFT use of the same arithmetic operations of the targeted system and gives a theoretical approach to designing a fault tolerant form of the system. It does not require an additional arithmetical logic unit and has relatively low overhead. ABFT can be effectively integrated into stream ciphers. Zhang, Lee and Tsai have proposed two efficient fault-tolerant schemes based on the RSA cryptosystem, respectively, in 1999 and 2003 [34,35]. The vulnerability of Zhang's scheme was pointed out by Iuon-Chang Lei et al. [36]. Using RSA-based transpose matrix, Shreenath Acharya, Sunaina Kotekar, and Seema S Joshi enhanced Iuon-Chang Lei et al.'s scheme with an extra level of security [37].

In 2016, H Elkamchouchi et al. proposed a method for improving digital signature scheme based on fault tolerance to help speed up decryption, as well as to overcome several common attacks [38]. Furthermore, it enhances security by converting the original message into a transposed matrix. In the same year, a new key agreement protocol based on factoring and discrete logarithms [39].For each matrix of 3*3; the scheme can correct four errors at most.

In this article, a new secured cryptosystem using the fractional order hyperchaotic Lorenz system based on fault tolerance technique is presented to encrypt color image. The system is carried out through different forms of permutation to improve security level, as well as up to three errors that can be discovered and corrected through the performance of fault tolerance technique. The article is planned as follows: In Section 2, a brief explanation of the fractional order hyperchaotic system, DNA encoding, and fault tolerance is provided. Section 3 discusses the proposed algorithm for image encryption, while Section 4 delivers numerical simulation results. Section 5 examines the proposed scheme's performance. This article is concluded in Section 6.

2. Preliminaries

2.1. Fractional-Order Hyperchaotic System

Recent years have seen an increase in interest in hyperchaotic systems. According to general definitions, a hyperchaotic system is a chaotic system with more than one positive Lyapunov exponent, as well as more complex dynamical behaviors than chaotic systems. An illustration of this is the fractional order hyperchaotic Chen's system [40] and the hyperchaotic Lorenz system [41,42]. In addition to the fact that the *FOHCL* system has good complex dynamics [42,43], several previous studies have demonstrated its effectiveness in encrypting images [15,44,45]. As a result, we generate a chaotic sequence for our algorithm [15,43] using a four-dimensional *FOHCL* system. The *FOHCL* consists of the following elements:

$$\begin{aligned} \frac{d^{q^1}x}{dt^{q^1}} &= a(y-x)+w, \\ \frac{d^{q^2}y}{dt^{q^2}} &= bx-y-xz, \\ \frac{d^{q^3}z}{dt^{q^3}} &= xy-cz, \\ \frac{d^{q^4}w}{dt^{q^4}} &= yz+dw, \end{aligned} \tag{1}$$

where a, b, c, d and $q_i \;\; i = (1, 2, 3, 4)$ are the control parameters of the fractional order hyper chaotic system. It provide hyperchaotic behavior when the control parameters are $a = 10$, $b = 8/3$, $c = 28$, $d = -1$ with initial values $x^0 = 12$, $y^0 = 22$, $z^0 = 31$ and $w^0 = 4$ and the Lyapunov exponents of the system are $\lambda_1 = 0.3362$, $\lambda_2 = 0.1568$, $\lambda_3 = 0$, $\lambda_4 = -15.1724$. A hyperchaotic behavior is observed with two positive values among all four Lyapunov exponents [46].

2.2. DNA Encoding

Deoxyribonucleic Acid (DNA) stores genetic information of all living organisms. In other words, it is a carrier of information made up of many small units referred to as nucleotides. There were three components in these nucleotides: Nitrogenous base, five carbon sugar, and phosphate group. The Nitrogenous base consists of four bases: Adenine, Thymine, Cytosine, and Guanine (A, T, C, G). All the complex information about an organism is stored in the combination of these bases. Adenine and Guanine are named purines, while Thymine and Cytosine are named pyrimidines [47]. The eight encoding rules for DNA nucleotides are shown in Table 1. Based on these rules, the value of a pixel is converted into its corresponding DNA sequence, which are used in cryptography. In DNA cryptography, the four bases A, T, C and G are used to capture the information.

Table 1. DNA nucleotides encoding rules [47].

Rule	DNA Nucleotides			
	A	**T**	**G**	**C**
R#1	00	11	01	10
R#2	00	11	10	01
R#3	11	00	01	10
R#4	11	00	10	01
R#5	10	01	11	00
R#6	01	10	11	00
R#7	10	01	00	11
R#8	01	10	00	11

2.3. Fault Tolerance Technique

Security protocols based on cryptography demand extensive computation and communication to attain a certain level of security. A critical function of cryptography is to make confidential data secure, reliable, and encrypted when communicating over unreliable channels. Fault tolerance techniques are used to correct errors during transmission. Figure 1 shows a conventional fault tolerance network.

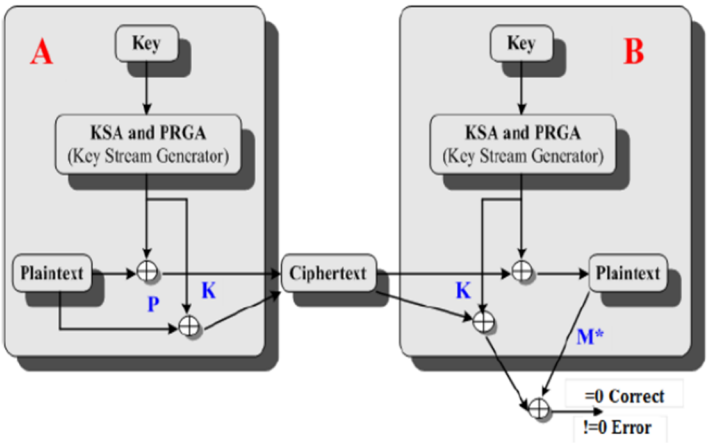


Figure 1. Conventional Fault Tolerance Technique.

3. Proposed Cryptosystem

The block diagram of the proposed cryptosystem illustrated in Figure 2 consists of four main phases. In the first phase, the input color image is diffused through pixel level encryption stage, which is based on the generated fractional order hybrid chaotic map discussed in the first part. For more randomness to increase the efficiency of the encryption, the second phase, bit level permutation diffuses each bit from the output of binary conversion. In the third phase, DNA level encryption, as part of the hyperchaotic sequence, the bit stream of the image is encoded as DNA sequence. Then apply DNA mutations and 3-dimensional permutations to enhance the security of the shuffled information. Finally, the fourth phase is fault tolerance technique, which detects error through check sum insertion and digital signature and corrects it.

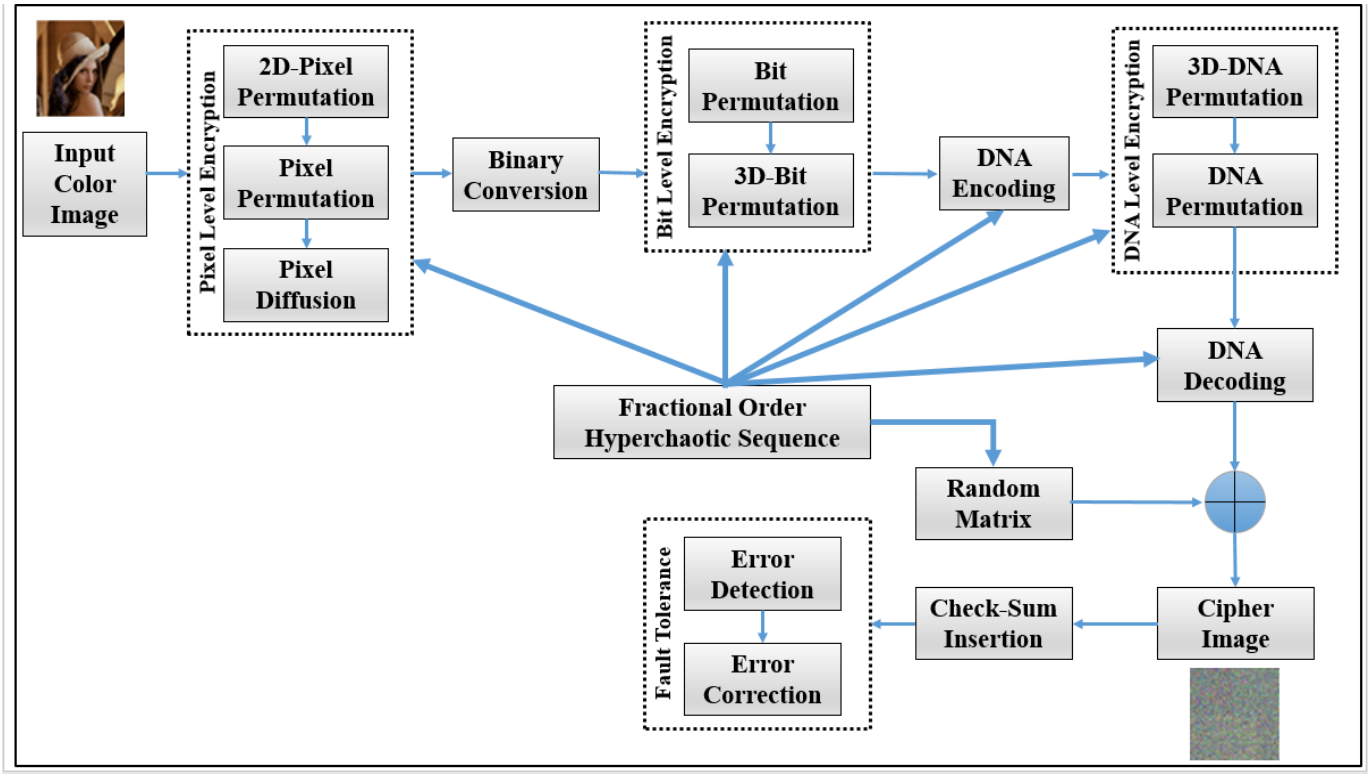


Figure 2. Proposed cryptosystem.

3.1. Synthesis of the Hyperchaptic Sequence

Considering that hyperchaotic systems are well suited in Section 2.1, we use the 4-D chaotic system for image encryption. The sequence consists of three steps that are carried out according to the given algorithm.

Step 1. The *FOHCL* structure is first iterated N_0 times, then the generated sequence is removed to prevent adverse effects.

Step 2. This process continues to iterate N = Ceil [40 (H × W) + 4 (H + W) + 14)/4] times where, H and W define as the size of the image. For the jth iteration, the *FOHCL* system generates four state values obtained from (1) which denoted by $s^j = \left\{x_1^j, x_2^j, x_3^j, x_4^j\right\}$, $(j = 1, 2, 3 \ldots N)$.

Step 3. As a result of this iteration, fractional order hyperchaotic sequences can be achieved by concatenating all the above states as

$$\begin{aligned} K_{FOHCL} &= \left\{s^1, s^2, s^3 \ldots s^N\right\} \\ &= \left\{x_1^1, x_2^1, x_3^1, x_4^1 \ldots x_1^N, x_2^N, x_3^N, x_4^N\right\} \\ &= \left\{k^1, k^2, k^3 \ldots k^{4N-2}, k^{4N-1}, k^{4N}\right\} \end{aligned} \tag{2}$$

To enable encryption, sort the sequence K_{FOHCL} into subsequences and serve it for two purposes: (1) implement permutations; (2) manipulate images for diffusion. Our scheme shows that we use original K_{FOHCL} values for the first purpose, but that we map the hyperchaotic sequence with n values to range of [0, 255] for the second purpose.

$$S^i = \mathrm{mod}[Floor(\mathrm{mod}((|k_i| - Floor(|k_i|) \times 10^{15}, 10^8)), 256) \qquad i = 1, 2, 3, \ldots n \tag{3}$$

where S^i is the ith integer sequence, mod is the modulo operation, $|\cdot|$ is the absolute value operation [29].

3.2. Pixel Level Encryption

Step 1: Let M is the input image with size $H \times W$. By using Equation (1), create a hyperchaotic sequence K_{FOHCL}.

Step 2: Perform 2D pixel permutation, which means by row and column on M to obtain M_0 by extracting the first entries from K_{FOHCL} with size $H + W$.

Step 3: In order to obtain M_1, we need to extract the next items from K_{FOHCL} with size $H \times W$ to perform pixel permutation as follows:

1. Organize pixels, bits, and acid bases into a one-dimensional vector V with $L = H \times W/H \times W \times 8/H \times W \times 4$.
2. Extract a subsequence from K_{FOHCL} with the length of vector V and sort in ascending order to obtain i^x, $x = 1, 2, \ldots, L$.
3. A new vector V' is created by rearranging V according to i^x as follows:

$$V'_x = V_{i^x} \tag{4}$$

Step 4: To obtain sequence S_1, first extract the next items from K_{FOHCL} with size $H \times W + 1$ to form a new sequence S_0. Next, map S_0 to the integer interval [0, 255] from Equation (3) to produce S_1. Assuming M_1 has been converted globally by performing pixel diffusion; M_2 can be obtained by taking S_1 as the primary value and the remainder as the key. The geometrical image on pixels is diffused in two stages in our scheme.

The first stage diffusion can occur as:

$$\begin{aligned} D^1 &= s^1 \otimes \bmod(C^0 + k^1, 256), \\ D^i &= s^i \otimes \bmod(D^{i-1} + k^i, 256) \end{aligned} \tag{5}$$

where $S = \{s^i\}$, $i = 1, 2, \ldots, L$ is the $1D$ pixel sequence of the input image with length L, C^0 is the initial key k is the key sequence which defined by $k^i \in [0, 255]$.

The second stage diffusion can occur as:

$$\begin{aligned} D^1 &= D^1 \otimes \bmod(\left|D^L - k^1\right|, 256) \\ D^i &= D^i \otimes \bmod(\left|D^{i-1} - k^i\right|, 256) \end{aligned} \tag{6}$$

Then obtain D as a total pixel diffusion by applying XOR operation between (5) and (6).

Step 5: Encode M_2 to a bit sequence M_B to be with size $H \times W \times 8$.

3.3. Bit Level Encryption

Step 1: Three-dimensional permutation is a method of permuting planes in three dimensions in different directions. We will only provide the operation in direction of width here to simplify things due to the similarity of the operations in each direction as follows:

1. Create a subsequence of the chaotic sequence K_{FOHCL} of length $L = H$
2. Reorder the plane p to get p' by ascending the index sequence i^x. $x = 1, 2, \ldots L$

$$p'_x = p_{i^x} \qquad x = 1, 2, 3 \ldots L \tag{7}$$

Step 2: Basically, this is step 3 in pixel level encryption, but we will use a subsequence with a length of $L = H \times W \times 8$

Step 3: M_{B1} can be obtained by performing bit permutation on M_B using the next items from K_{FOHCL} with size $H \times W \times 8$. For the next items with length $H + W + 8$ implement 3D-permutation on M_{B1} to get M_{B2}

3.4. DNA Encoding and Level Encryption

In order to encode the bit stream of the input image following the rules decided by hyperchaotic sequence, the bit stream is encoded as a DNA sequence as follows:

Step 1: Calculate the sequence S_2 by converting the next items of K_{FOHCL} with size $H \times W \times 4$ to the integer interval of [0, 255] using Equation (3).

Step 2: In order to acquire M_D, encode the bits in M_{B2} with the DNA nucleotide rule as follows:

$$\text{Rule} = \text{mod}\,(S^{i}{}_{2}, 8) + 1 \tag{8}$$

Step 3: Apply 3D permutation on M_D using the next items from K_{FOHCL} with size $H + W + 4$ to acquire M_{D1}

3.5. DNA Decoding

Step 1: Step 1 in the DNA encoding process is identical in order to obtain the S_3 sequence.

Step 2: Using the DNA rule established in (9) for M_{D1}, encode the i^{th} strand of DNA to obtain the binary sequence M_{B3}:

$$\text{Rule} = \text{mod}\,(S^{i}{}_{3}, 8) + 1 \tag{9}$$

Step 3: C_1 is the cipher image generated from the binary sequence M_{B3}.

Step 4: Obtain S_4 with the same manner for obtaining S_2 and S_3

Step 5: From $S4$, generate random matrix R with size $H \times W$ then XOR the random matrix with the cipher matrix C_1 to obtain a new cipher image C

3.6. Fault Tolerance

Step 1: Generate two prime vectors P_{v1} and P_{v2} with length W and H, respectively

Step 2: Creates a matrix C' with length $(W + 1) \times (H + 1)$ matrix as follows:

$$C' = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1H} & C_1 \\ c_{21} & c_{12} & \cdots & c_{1H} & C_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{W1} & c_{W2} & \cdots & c_{WH} & C_W \\ C_1 & C_2 & \cdots & C_H & h \end{pmatrix} \tag{10}$$

where $C_i = \prod_{j=1}^{H} c_{ij} * P_{v1i} \mod 256$, $for\ 1 \le i \le W$, $C_j = \prod_{i=1}^{W} c_{ij} * P_{v2i} \mod 256$, $for\ 1 \le j \le H$, and $h = H(C_1, C_2 \ldots C_W, C^1, C^2, \ldots C^H)$, where $H(.)$ is the hash value.

Step 3: In order to check the validity of the recipient's signature, the transmitter can now check:

$$C_i = \prod_{j=1}^{H} c_{ij} * P_{v1j} \,\text{mod}256\ ,\ for\ 1 \le i \le W \tag{11}$$

$$C_j = \prod_{i=1}^{W} c_{ij} * P_{v2j} \,\text{mod}256\ ,\ for\ 1 \le j \le H\ , \tag{12}$$

Upon determining that they are true, the transmitter computes $h = H(C_1, C_2 \ldots C_W, C^1, C^2, \ldots C^H)$ and checks the receiver's signature. Upon receiving the valid signature, he sends the securely encrypted data to the recipient. Otherwise, errors have occurred in either the calculation phase or the transmission phase.

Step 4: A location where the error was discovered, and data corrections were needed can be determined

Case 1: If $C_k \neq \prod_{j=1}^{H} c_{kj} * P_{v1j} \quad k = 1,2,3 \ldots W$ and $C_l = \prod_{i=1}^{W} c_{il} * P_{v2l} \quad l = 1,2,3 \ldots H$, the pixel value at s-row and l-column is false and the correct one should be

$$C_k - \prod_{j=1, j\neq l}^{H} c_{kj} * P_{v1j} \tag{13}$$

Case 2: If $C_{k_1} \neq \prod_{j=1}^{H} c_{k_1 j} * P_{v1j}, C_{k_2} \neq \prod_{j=1}^{H} c_{k_2 j} * P_{v1j}, k_1, k_2 = 1,2 \ldots W$ and $k_1 \neq k_2$, $C_l \neq \prod_{i=1}^{W} c_{il} * P_{v2l}$ two pixels data at (k_1, l) and (k_2, l) are wrong and the correct ones should be

$$\begin{aligned} c_{k_1 l} &= C_{k_2} - \prod_{j=1, j\neq l}^{H} c_{k_1 j} * P_{v1j} \\ c_{k_2 l} &= C_{k_2} - \prod_{j=1, j\neq l}^{H} c_{k_2 j} * P_{v1j} \end{aligned} \tag{14}$$

Case 3: If $C_k \neq \prod_{j=1}^{H} c_{kj}, C_{l_1} \neq \prod_{i=1}^{W} c_{il_1} * P_{v2l_1}, C_{l_2} \neq \prod_{i=1}^{W} c_{il_2} * P_{v2l_2}$ and $C_{l_3} \neq \prod_{i=1}^{W} c_{il_3} * P_{v2l_3}$ three errors occur at (k, l_1), (k, l_2) and (k, l_3). The correct ones should be the following:

$$\begin{aligned} c_{kl_1} &= C_{l_1} - \prod_{i=1, i\neq l_1}^{H} c_{l_1 i} * P_{iv2} \\ c_{kl_2} &= C_{l_2} - \prod_{i=1, i\neq l_2}^{H} c_{l_2 i} * P_{iv2} \\ c_{kl_3} &= C_{l_3} - \prod_{i=1, i\neq l_3}^{H} c_{l_3 i} * P_{iv2} \end{aligned} \tag{15}$$

Up to three errors can be discovered and corrected.

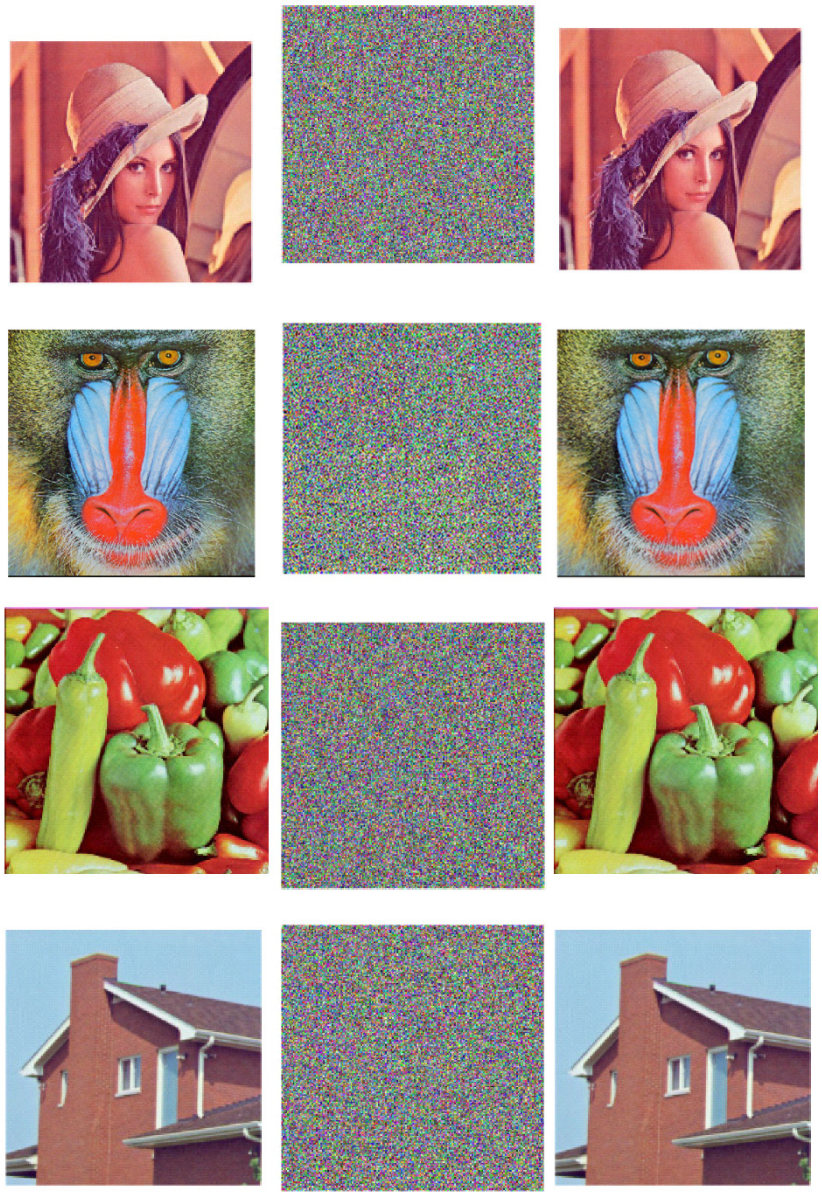
4. Simulation Results

We compare the proposed scheme with existing schemes in order to assess its performance. Comparative analysis was performed on a variety of attacks, which included key sensitivity, plaintext sensitivity, differential attacks, brute force attacks, data crop attacks, and entropy attacks. *FOHCL* initialized its initial values $x^0 = 12, y^0 = 22, z^0 = 31$ and $w^0 = 4$ with 10,000 iterations. In addition, a fixed value of 0.98 is provided for all the fractional orders q_i, i = (1, 2, 3, 4). As shown in Table 2, four images "Lena", "Baboon", "Peppers", and "House" are used for testing the proposed algorithm. Matlab (R2015a) (Mathworks, Natick, MA, USA) is used to carry out all simulations on a 64-bit Windows 7 (Microsoft, Redmond, WA, USA) with 64 GB memory.

Table 2. Proposed images.

Images	Size (W × H)
Lena	256 × 256
Baboon	512 × 512
Peppers	280 × 270
House	360 × 344

Figure 3 is a comparison of original images, ciphered images, and recovered images that was utilized in order to assess the performance of the proposed cryptosystem. Since the cipher images have been changed completely, it is impossible to determine their origin.



(a) (b) (c)

Figure 3. Simulation results. (**a**) Original images, (**b**) Encrypted images and (**c**) Recovered images.

5. Security and Performance Analysis

5.1. Key Size

It can be described in basic terms that the proposed scheme uses four initial values, which are x^0, y^0, z^0 and w^0. The keyspace size is $10^{16\times4} = 10^{64} \approx 2^{212}$, depending on the precision of the initial values. A larger key space than 2^{100} increases the possibility of achieving high-level security [1,42]. As a result, the proposed scheme has a large key size to resist brute-force attacks. Further enhancing the key size can also be achieved by using the fractional orders of the *FOHCL*.

5.2. Key Sensitivity

Cryptosystems with extreme key sensitivity are necessary and sufficient for their performance. An important characteristic of keys is their sensitivity, which means a minor alteration in any key should result in drastically different results. The sensitivity can be analyzed when encrypting or decrypting data. The image produced by using a key without

any changes during the encryption process should be completely different than the one obtained by using a key with no changes during the process. An incorrect secret key set will prevent recovery of the plain image from the cipher image.

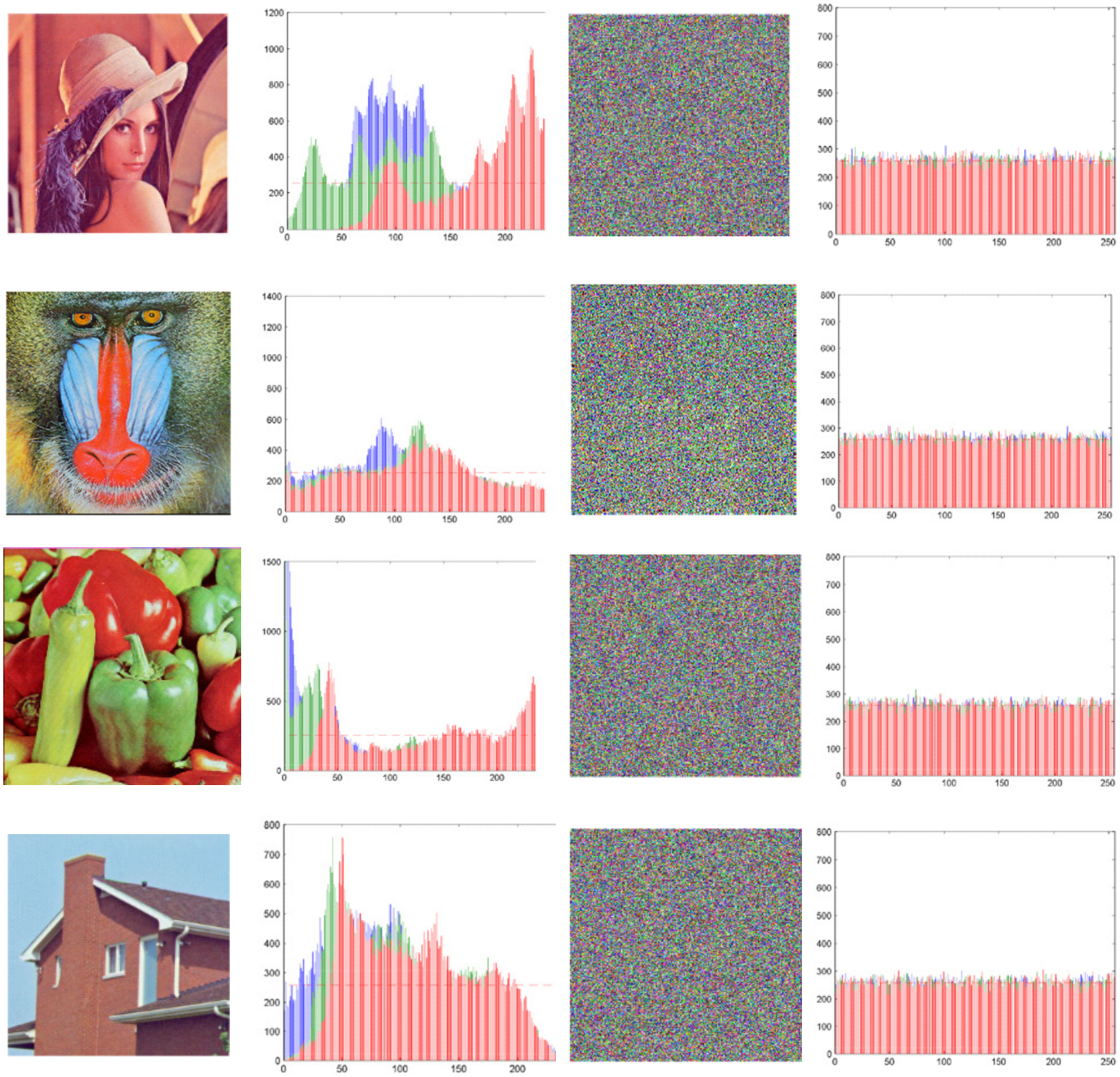
By decrypting the cipher images twice, we demonstrate the effectiveness of the proposed algorithm based on its secret keys. Decrypting the cipher images starts with the correct secret keys $x^0 = 12, y^0 = 22, z^0 = 31$ and $w^0 = 4$, whereas the second time the cipher images was decrypted with slightly different keys $x^0 = 12 + 10^{-15}, y^0 = 22, z^0 = 31$ and $w^0 = 4$. In Figure 4, we demonstrate the results of the experiment using the Lena image. Our comparison clearly displays that a little change in the secret keys affects in a totally different way the decrypted images, proving that the proposed cryptosystem is very sensitive to secret keys.

Figure 4. Key sensitivity: (**a**) Lena image; (**b**) Encrypted Lena with correct key; (**c**) Encrypted Lena with wrong key. (**d**) The difference between (**b**) and (**c**); (**e**) Decrypted Lena with the correct key from (**b**). (**f**) Decrypted Lena with the wrong key from (**b**); (**g**) Decrypted Lena with the key from (**c**); (**h**) Decrypted Lena with the wrong key from (**c**).

5.3. Histogram Analysis

Image encryption usually involves measuring the spreading of pixel values between an original image and a cipher image using a histogram. Original images do not always have a uniform distribution of their histograms, but cipher images with a good encryption scheme tend to have a uniform distribution. Alternatively, the flatter the histogram of the cipher image, the more effective the encryption scheme.

Figure 5 shows the histograms of the input images and their cipher images. The histograms of plain images are not uniformly distributed. Unlike cipher images, cipher images are uniformly distributed. These results suggest that the cryptosystem is resistant to histogram attacks.



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

Figure 5. Histograms of colored plain images. (**a**) Plain images. (**b**) Histograms of plain images. (**c**) Cipher images. (**d**) Histograms of cipher images.

5.4. Correlation Factor Analysis

Natural images usually have high correlation between adjacent pixels. It is important to use a cryptographic algorithm that reduces such correlation. In the following formula, correlation coefficient *CC* can be expressed as follows to measure correlation:

$$CC = \frac{E((p_1 - E(p_1)) - (p_2 - E(p_2)))}{\sqrt{E(p_1 - E(p_1))^2} \times \sqrt{E(p_2 - E(p_2))^2}},$$
$$E(p_1) = \frac{1}{m}\sum_{i=1}^{m} p_{1i}, E(p_2) = \frac{1}{m}\sum_{i=1}^{m} p_{2i} \quad (16)$$

where p_1 and p_2 are two neighboring pixels and $m = 1000$. Each of the 1000 pairs of adjacent pixels in the original and cipher images was randomly selected and analyzed horizontally, vertically and diagonally. Figure 6 shows the correlation distribution of adjacent pixels for both plain and cipher Lena images, and Table 3 shows their results. Additionally, a comparison is shown in Table 4 between the proposed cryptosystem and related works. According to Table 4, the correlation coefficients for the original images are close to 1, but those for the ciphered images are very close to 0, which means the correlativeness shared between adjacent pixels is extremely rare.

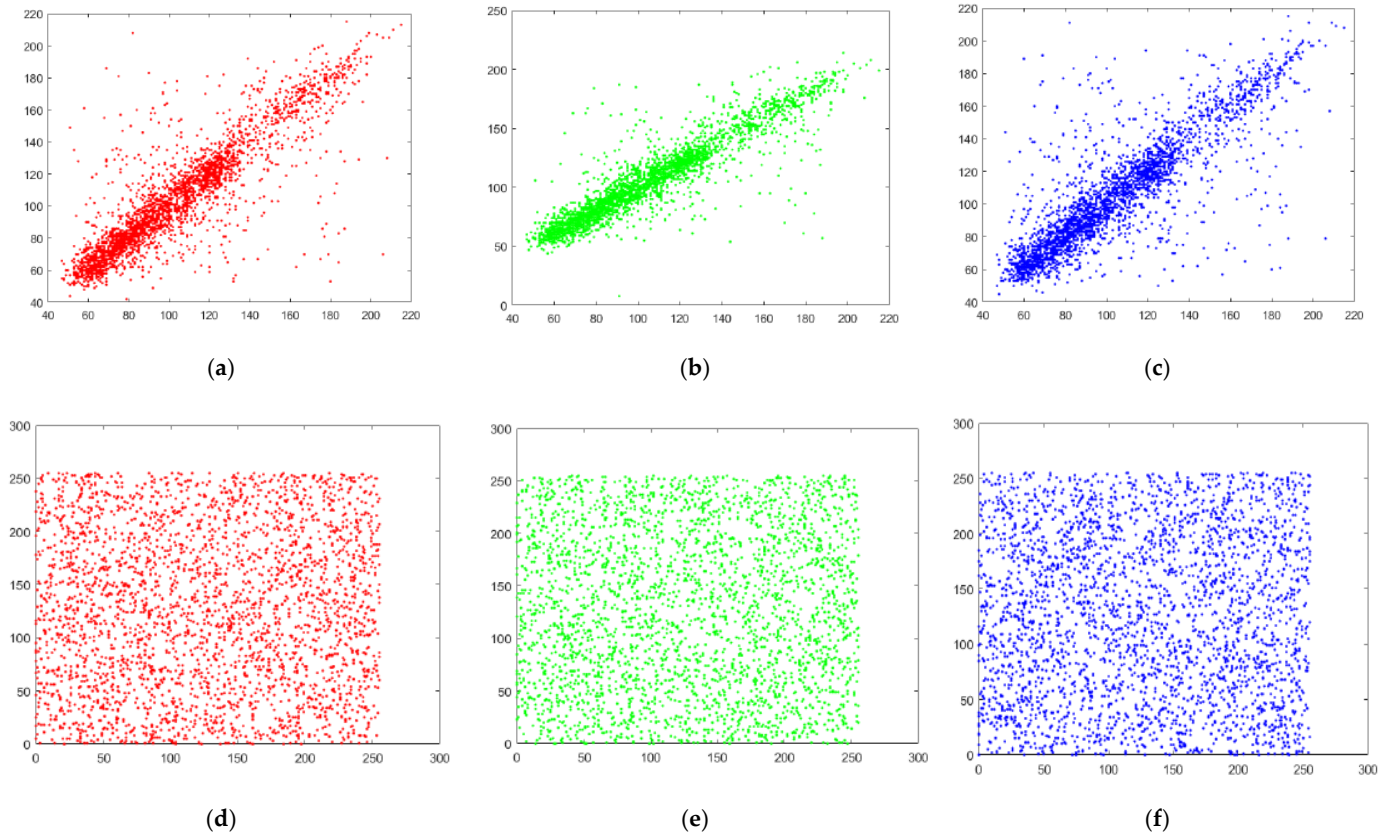


Figure 6. Correlation distribution: (**a**,**d**) Horizontal distribution in the red channel of the plain and cipher images; (**b**,**e**) Vertical distribution in the green channel of the plain and cipher images; (**c**,**f**) Diagonal distribution in the blue channel of the plain and cipher images.

Table 3. The various calculated correlation factors on Lena image.

Correlation Factors	Components	Direction of Adjacent Pixels		
		Horizontal	Vertical	Diagonal
Plain	R	0.9429	0.9741	0.9592
	G	0.9352	0.9633	0.9410
	B	0.9179	0.9504	0.9278
Cipher	R	0.0085	0.0015	−0.0021
	G	−0.0047	−0.0043	0.0035
	B	−0.0013	0.0025	0.0008

Table 4. Correlation coefficients between various encryption schemes and the proposed scheme.

Algorithm	Direction of Adjacent Pixels		
	Horizontal	Vertical	Diagonal
Proposed	0.0012	0.0009	−0.0003
[48]	−0.0082	0.0118	−0.0012
[49]	0.0019	0.0014	−0.0028
[50]	0.0032	0.0015	−0.0018
[51]	−0.0022	−0.0010	0.0005
[52]	−0.0031	0.0027	0.00011

5.5. Plaintext Senstivity

A potential hacker attempts to locate the original image through every means available. In order to do this, we must first modify the original image, followed by encrypting both plain images, and finally finding some correlation between them. For evaluating the impact of changing one pixel in a plain image on an encrypted image, two measures have been developed: NPCR and UACI. The NPCR represents the difference in intensity between plain and cipher images. In contrast, UACI denotes the average intensity of differences between plain and cipher images. They are calculated as follows:

$$\text{NPCR} = \frac{1}{W \times H}\sum_{i,j} D(a,b) \times 100\% \tag{17}$$

$$D(a,b) = \left\{ \begin{array}{ll} 1 & if \;\; E(a,b) \neq E'(a,b) \\ 0 & if \;\; E(a,b) = E'(a,b) \end{array} \right\} \tag{18}$$

$$\text{UACI} = \frac{1}{W \times H}\sum_{i,j} \frac{E(a,b) - E'(a,b)}{255} \times 100\% \tag{19}$$

Both E and E' are the cipher images before and after the plain image has been altered by a single pixel. The values of NPCR and UACI's differential attack metrics are shown in Table 5. According to Table 6, the proposed algorithm is also better than other algorithms for NPCR and UACI for Lena images.

Table 5. Performance of plaintext sensitivity.

Images	NPCR (%)			UACI (%)		
	R	G	B	R	G	B
Lena	99.6243	99.6173	99.6292	33.4932	33.5781	33.6019
Baboon	99.6182	99.6191	99.6023	33.5354	33.4995	33.9971
Peppers	99.6205	99.6301	99.6183	33.6009	33.6053	33.4982
House	99.6394	99.6297	99.6136	33.9737	33.5864	33.8654

Table 6. Plaintext sensitivity performance compared with previous work.

Algorithms	Average NPCR	Average UACI
Proposed	99.6387	33.5498
[48]	99.6051	33.4294
[49]	99.6218	33.4809
[50]	99.6164	33.4650
[51]	99.2975	33.4999
[52]	99.1507	33.4380

5.6. Information Entropy

Information entropies are judged based on their randomness and predictability according to their entropy. The concept of entropy in information was developed by Shannon as follow

$$IE(m) = -\sum_{i=0}^{2^n-1} P(m_i) \log \frac{1}{P(m_i)} \tag{20}$$

where $IE(m)$ represents the information entropy of m, $P(m_i)$ denotes by the probability of message m_i. When 256 gray values are assigned to an absolutely random image, the maximum information entropy is 8. Entropy always performs better when it is closer to 8 than other values. The values of entropy for our selected images are shown in Table 7. Lena and Pepper images have an average value close to 8, which is close to the ideal value. Therefore, the proposed scheme is well protected from any entropy attack. Additionally, Table 7 illustrates that under comparable conditions, the proposed scheme performs better results.

Table 7. Information Entropy.

Images	Lena Image			Pepper Image		
	R	**G**	**B**	**R**	**G**	**B**
Proposed Cryptosystem	7.9993	7.9982	7.9975	7.9976	7.9987	7.9991
[48]	7.9981	7.9979	7.9970	7.9962	7.9926	7.9960
[49]	7.9919	7.9914	7.9892	7.9818	7.9971	7.9836
[50]	7.9973	7.9967	7.9873	7.9946	7.9945	7.9966

5.7. Data Loss Attacks

It is possible that part of the image will be lost during the transmission. Effective encryption should be able to cope with attacks involving data loss. The data loss from Figure 7a can be observed in the cropped portion of the Baboon ciphered image with size 70×180. This algorithm can be used to decrypt ciphertext images with partial loss of data. The decrypted image can be seen in Figure 7c. Clearly, the decrypted image is still so rich in information that it may be easily recognized. This means that the proposed scheme is highly protected against data loss.

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

Figure 7. Data loss attack for Baboon image: (**a**) Cipher image; (**b**) Cipher image with a 70×180 data loss; (**c**) Decrypted Baboon image from (**b**).

5.8. Computational Speed

The most important factor for an image cipher is its speed. The proposed cryptosystem has been developed under Intel (R) Core (TM) i5 CPU 2.7 GHz, 8 GB memory, Windows 10, MATLAB R2018a. A comparison of computational speed of the proposed cryptosystem and different encryption schemes appears in Table 8. The results demonstrate that the proposed system meets real-time performance requirements while being sufficiently fast compared to other schemes.

Table 8. Computational Speed.

Algorithms	Proposed System	[48]	[49]	[50]	[51]
Speed	0.16364	2.48231	1.53217	1.24786	0.98517

6. Conclusions

In this article, we present a new cryptosystem for correcting errors in color images based on *FOHCL* and DNA computing through the use of fault tolerance technique. A pseudorandom sequence is generated by the *FOHCL* that is used throughout the encryption process. It is possible to extent the small variation in one pixel to all other pixels by using simple pixel diffusion. Different types of transformations are performed on different levels of data. Data correction and digital signature are combined to create a fault-tolerant scheme that allows the recipient to verify the sender's signature while also correcting up to three errors. A comprehensive experiment and a security analysis have revealed that the proposed scheme is highly sensitive to the secret key, has a large key space, as well as the proposed scheme that can resist a number of known attacks, such as brute-force attacks, statistical attacks, differential attacks, data loss attacks and high-speed performance. The results show that a slight change in the secret keys result in very different ciphered images, the correlation coefficient is near to 1 for plain images and close to 0 for cipher images, the plaintext sensitivity is near to the ideal values of NPCR (99.61%) and UACI (33.46%), information entropy achieve ideal values which are close to 8. The decrypted image is still so rich in information even though a part of the image is lost during the transmission, as well as the system being fast in comparison to other schemes. These characteristics indicate that the proposed system is a promising one for image encryption.

Author Contributions: Conceptualization, H.G.M. and F.A.; methodology, H.G.M. and D.H.E.; software, H.G.M.; validation, H.G.M., F.A. and D.H.E.; formal analysis, F.A.; resources, D.H.E.; data curation, D.H.E.; writing—original draft preparation, H.G.M.; writing—review and editing, H.G.M.; visualization, F.A.; supervision, H.G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research project was funded by the Deanship of Scientific Research, Princess Nourah bint Abdulrahman University, through the Program of Research Project Funding after Publication, grant No (41-PRFA-P-26).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Ding, L.; Ding, Q. A Novel Image Encryption Scheme Based on 2D Fractional Chaotic Map, DWT and 4D Hyper-chaos. *Electronics* **2020**, *9*, 1280. [CrossRef]
2. Dagadu, J.C.; Li, J.; Aboagye, E.O.; Ge, X. Chaotic medical image encryption based on Arnold transformation and pseudorandomly enhanced logistic map. *Structure* **2017**, *4*, 8096–8103.
3. Chai, X.; Bi, J.; Gan, Z.; Liu, X.; Zhang, Y.; Chen, Y. Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. *Signal Process.* **2020**, *176*, 107684107684. [CrossRef]
4. Ding, M.; Jing, F. Digital image encryption algorithm based on improved Arnold transform. In Proceedings of the 2010 International Forum on Information Technology and Applications, Kunming, China, 16–18 July 2010; pp. 174–176.
5. Hou, W.B.; Wu, C.M. Image encryption and sharing based on Arnold transform. *J. Comput. Appl.* **2011**, *10*, 2682–2686.
6. Lorenz, E. Deterministic Non-period Flows. *J. Atmos. Sci.* **1972**, *20*, 130–141. [CrossRef]
7. Matthews, R. On the derivation of a "chaotic" encryption algorithm. *Cryptologia* **1989**, *13*, 29–42. [CrossRef]
8. Li, Z.; Peng, C.; Li, L.; Zhu, X. A novel plaintext-related image encryption scheme using hyper-chaotic system. *Nonliear Dyn.* **2018**, *94*, 1319–1333. [CrossRef]
9. Zhang, Y. The unified image encryption algorithm based on chaos and cubic S-Box. *Inf. Sci.* **2018**, *450*, 361–377. [CrossRef]

10. Batool, S.I.; Waseem, H.M. A novel image encryption scheme based on Arnold scrambling and Lucas series. *Multimed. Tools Appl.* **2019**, *78*, 27611–27637. [CrossRef]
11. Wang, X.; Çavuşoğlu, Ü.; Kacar, S.; Akgul, A.; Pham, V.-T.; Jafari, S.; Alsaadi, F.E.; Nguyen, X.Q. S-box based image encryption application using a chaotic system without equilibrium. *Appl. Sci.* **2019**, *9*, 781. [CrossRef]
12. Zhang, Y.; Chen, A.; Tang, Y.; Dang, J.; Wang, G. Plaintext-related image encryption algorithm based on perceptron-like network. *Inf. Sci.* **2020**, *526*, 180–202. [CrossRef]
13. Li, Z.; Peng, C.; Tan, W.; Li, L. A Novel Chaos-Based Color Image Encryption Scheme Using Bit-Level Permutation. *Symmetry* **2020**, *12*, 1497. [CrossRef]
14. Podlubny, I.; Petráš, I.; Vinagre, B.M.; O'Leary, P.; Dorčák, Ľ. Analogue realizations of fractional-order controllers. *Nonlinear Dyn.* **2002**, *29*, 281–296. [CrossRef]
15. Wang, Z.; Huang, X.; Li, Y.-X.; Song, X.-N. A new image encryption algorithm based on the fractional-order hyperchaotic Lorenz system. *Chin. Phys. B* **2013**, *22*, 010504. [CrossRef]
16. Wu, X.; Kan, H.; Kurths, J. A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps. *Appl. Soft Comput.* **2015**, *37*, 24–39. [CrossRef]
17. Zhao, J.; Wang, S.; Chang, Y.; Li, X. A novel image encryption scheme based on an improper fractional-order chaotic system. *Nonlinear Dyn.* **2015**, *80*, 1721–1729. [CrossRef]
18. Huang, X.; Sun, T.; Li, Y.; Liang, J. A color image encryption algorithm based on a fractional-order hyperchaotic system. *Entropy* **2014**, *17*, 28–38. [CrossRef]
19. Adleman, L.M. Molecular computation of solutions to combinatorial problems. *Science* **1994**, *266*, 1021–1024. [CrossRef]
20. Gehani, A.; LaBean, T.; Reif, J. DNA-based cryptography. In *Aspects of Molecular Computing*; Springer: New York, NY, USA, 2003; pp. 167–188.
21. Wang, X.; Wang, Y.; Zhu, X.; Luo, C. A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level. *Opt. Lasers Eng.* **2020**, *125*, 105851. [CrossRef]
22. Zheng, J.; Luo, Z.; Zeng, Q. An efficient image encryption algorithm based on multi chaotic system and random DNA coding. *Multimed. Tools Appl.* **2020**, *79*, 29901–29921. [CrossRef]
23. Siddartha, B.; Ravikumar, G. An efficient data masking for securing medical data using DNA encoding and chaotic system. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 6008–6018.
24. Wang, T.; Wang, M.H. Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. *Opt. Laser Technol.* **2020**, *132*, 106355. [CrossRef]
25. Zefreh, E.Z. An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. *Multimed. Tools Appl.* **2020**, *79*, 24993–25022. [CrossRef]
26. Babaei, A.; Motameni, H.; Enayatifar, R. A new permutation-diffusion-based image encryption technique using cellular automata and DNA sequence. *Optik* **2020**, *203*, 164000. [CrossRef]
27. Zhang, Q.; Liu, L.; Wei, X. Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps. *AEU Int. J. Electron. Commun.* **2014**, *68*, 186–192. [CrossRef]
28. Xie, T.; Liu, Y.; Tang, J. Breaking a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Optik* **2014**, *125*, 7166–7169. [CrossRef]
29. Liu, Y.; Tang, J.; Xie, T. Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map. *Opt. Laser Technol.* **2014**, *60*, 111–115. [CrossRef]
30. Patel, J.H.; Fung, L.Y. Concurrent error detection in ALU's by recomputing with shifted operands. *IEEE Trans. Comput.* **1982**, *C-31*, 589–595. [CrossRef]
31. Gulati, R.K.; Reddy, S.M. Concurrent error detection in VLSI array structures. In Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors, (ICCD), Austin, TX, USA, 7–9 October 1986; pp. 488–491.
32. Kuhn, R.H. *Yield Enchancement by Fault-Tolerant Systolic Arrays in VLSI and Modern Signal Processing*; Prentice-Hall: Hoboken, NJ, USA, 1985; pp. 178–184.
33. Al-Yamani, A.A.; Oh, N.; McCluskey, E.J. Performance Evaluation of Checksum Based ABFT. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, San Francisco, CA, USA, 24–26 October 2001; p. 461.
34. Zhang, C.N. Integrated Approach for Fault Tolerance and Digital Signature in RSA. *IEEE Proc. Comput. Digit. Tech.* **1999**, *146*, 151–159. [CrossRef]
35. Lee, N.; Tsai, W. Efficient Fault-tolerant Scheme basd on the RSA system. *IEEE Proc. Comput. Digit. Tech.* **2003**, *150*, 17–20. [CrossRef]
36. Lin, I.-C.; Wang, H.-L. An Improved Digital Signature Scheme with Fault Tolerance in RSA. In Proceedings of the Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Darmstadt, Germany, 15–17 October 2010.
37. Acharya, S.; Kotekar, S.; Joshi, S.S.; Shetty, S.; Lobo, S. Implementing Digital Signature based Secured Card System for Online Transactions. *Int. J. Comput. Appl.* **2013**, *65*, 27–32.
38. Elkamchouchi, H.; Mohamed, H.G.; Ahmed, F.; ElKamchouchi, D.H. A Secure Digital Signature Scheme with Fault Tolerance Based on the Improved RSA System. In Proceedings of the Fifth International Conference on Cryptography and Information Security (CRYPIS-2016), Sydney, Australia, 28–29 May 2016; pp. 35–44.

39. Elkamchouchi, H.; Mohamed, H.G.; Ahmed, F.; ElKamchouchi, D.H. New Secure Proxy Signature Scheme with Fault Tolerance Based on Factoring and Discrete Logarithm. *Int. J. Sci. Technol. Res. Eng. (IJSTRE)* **2016**, *1*, 106–113.
40. Li, Y.; Tang, W.K.S.; Chen, G.R. Generating hyperchaos via state feedback control. *Int. J. Bifurc. Chaos Appl. Sci. Eng.* **2005**, *15*, 3367–3375. [CrossRef]
41. Wang, X.Y.; Wang, M.J. A hyperchaos generated from Lorenz system. *Physica* **2008**, *387*, 3751–3758. [CrossRef]
42. Wang, X.Y.; Song, J.M. Synchronization of the fractional order hyperchaos Lorenz systems with activation feedback control. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 3351–3357. [CrossRef]
43. Wang, S.; Wu, R. Dynamic analysis of a 5D fractional order hyperchaotic system. *Int. J. Control. Autom. Syst.* **2017**, *15*, 1003–1010. [CrossRef]
44. He, J.; Yu, S.; Cai, J. A method for image encryption based on fractional-order hyperchaotic systems. *J. Appl. Anal. Comput.* **2015**, *5*, 197–209.
45. Wu, X. A color image encryption algorithm using the fractional-order hyperchaotic systems. In Proceedings of the 5th International Workshop on Chaos-Fractals Theories and Applications, IWCFTA (12), Liaoning, China, 18–21 October 2012; pp. 196–201.
46. Sebastian, A.; Delson, T. Secure magnetic resonance image transmission and tumor detection techniques. In Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 18–19 March 2016; pp. 1–5.
47. Wang, Q.; Zhang, Q.; Zhou, C. A multilevel image encryption algorithm based on chaos and DNA coding. In Proceedings of the 2009 Fourth International on Conference on Bio-Inspired Computing (BICTA 09), Beijing, China, 16–19 October 2009; pp. 70–74.
48. Zhang, Y.; He, Y.; Li, P.; Wang, X. A new color image encryption scheme based on 2dnlcml system and genetic operations. *Opt. Lasers Eng.* **2020**, *128*, 106040. [CrossRef]
49. Tariq, S.; Khan, M.; Alghafis, A.; Amin, M. A novel hybrid encryption scheme based on chaotic lorenz system and logarithmic key generation. *Multimed. Tools Appl.* **2020**, *79*, 23507–23529. [CrossRef]
50. Alghafis, A.; Munir, N.; Khan, M.; Hussain, I. An encryption scheme based on discrete quantum map and continuous chaotic system. *Int. J. Theor. Phys.* **2020**, *59*, 1227–1240. [CrossRef]
51. Munir, N.; Khan, M.; Wei, Z.; Akgul, A.; Amin, M.; Hussain, I. Circuit implementation of 3d chaotic self-exciting single-disk homopolar dynamo and its application in digital image confidentiality. *Wirel. Netw.* **2020**, 1–18. [CrossRef]
52. Kang, X.; Guo, Z. A new color image encryption scheme based on dna encoding and spatiotemporal chaotic system. *Signal Process. Image Commun.* **2020**, *80*, 115670. [CrossRef]