A New Algorithm for Medical Color Images Encryption Using Chaotic Systems

Seyed Shahabeddin Moafimadani *, Yucheng Chen and Chunming Tang *

School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China; yuchengchen@e.gzhu.edu.cn
* Correspondence: ctang@gzhu.edu.cn (C.T.); Shahabmadani@e.gzhu.cn (S.S.M.)

Received: 11 May 2019; Accepted: 4 June 2019; Published: 10 June 2019

Abstract: In this paper, we present a new algorithm based on chaotic systems to protect medical images against attacks. The proposed algorithm has two main parts: A high-speed permutation process and adaptive diffusion. After the implementation of the algorithm in the MATLAB software, it is observed that the algorithm is effective and appropriate. Also, to quantitatively evaluate the uniformity of the histogram, the chi-square test is done. Key sensitivity analysis demonstrates that images cannot be decrypted whenever a small change happens in the key, which indicates that the algorithm is suitable. Clearly, part of special images is selected to test the selected plain-text, like an all-white image and an all-black image. Entropy results obtained from the implementation of the algorithm on this type of images show that the proposed method is suitable for this particular type of images. In addition, the obtained results from noise and occlusion attacks analysis show that the proposed algorithm can withstand against these types of attacks. Moreover, it can be seen that the images after encryption and decryption are of good quality; the measures such as the correlation coefficients, the entropy, the number of pixel change rate (NPCR), and the uniform average change intensity (UACI) have suitable values; and the method is better than previous methods.

Keywords: image encryption; medical color images; RGB; chaotic system

1. Introduction

The confidentiality of patient information is one of the vital security aspects of electronic health services. For example, the confidentiality of patients' medical records is necessary. In addition, the methods of protection should be improved due to the rapid advancement of technologies for accessing the personal information of individuals. The security and privacy of medical image transferring is one of the acute subjects that should be seriously considered in telecare medical information systems (TMIS). In the past years, medical images were grayscale, but today, color images have entered the medical arena, and they can show more accurate information about body conditions. Color images that are acquired by new scanners using the Medipix3RX chip technology are very important in the medical arena. Image-data transferring from a position to another via an unsafe network are usually determined in qualifications of privacy, validity, totality, and confidentiality. Therefore, more significance should be given to the security of the sensitive data that are included in medical images by DICOM (digital imaging communication in medicine). In this textuality, many problems using various cryptographic techniques have been proposed in the literature to overcome this problem [1–4]. In their paper, Abdel-Nabi and Al-Haj proposed a hybrid encryption algorithm using watermarking that offers high embedding capabilities for medical images. The proposed algorithm is a combination of reversible data-caching techniques with standard encryption techniques for ensuring the security requirements for transferred and stored medical images [5]. In their paper, Abdmouleh et al. presented a partial cryptographic approach that was based on the digital wavelet transform (DWT) and was
JPEG2000 compliant to ensure the safe transfer and storage of medical images [6]. Lakshmi et al. presented a similar algorithm using a discrete wavelet transform (DWT), with the difference being that they used a fuzzy chaotic map for the watermarking [7]. In their paper, Cao et al. presented a medical image encryption algorithm using edge maps that were derived from a source image. The algorithm consists of three parts: Bit-plane decomposition, a random-sequence generator, and permutation [8]. Ismail et al. presented the double-humped (DH) logistic map to produce pseudorandom numbers keys (PRNG) in their paper. The generalized parameter that is added to the map provides more control on the map chaotic range [9]. Jeong et al. proposed a new medical image encrypting method using a 2D chaotic map and C-MLCA in their article. The 2D chaotic map is a construction with self-guarding attributes, which moves the location of the pixel and encrypts the image [10]. In their method, Ke et al. offered an encryption algorithm that was based on reversible data using the MSB-based prediction [11]. In their study, Nematzadeh et al. proposed an encryption method for medical images based on a hybrid pattern of the improved genetic algorithm (IGA) and paired map lattices. First, the assumed way employs a paired map lattice to produce some of secure cipher-images as a primitive population of the IGA. Then, it exerts the IGA to both increase the entropy of the cipher images and reduce the algorithm’s calculations time [12]. In their paper, Singh et al. proposed a medical image encryption scheme algorithm using the improved ElGamal encryption technique. Their proposed method made a new contribution since the necessity for separate calculations for encoding plain messages to elliptic curve coordinates was removed. The algorithm using the improved version of the ElGamal encryption scheme is designed to encrypt medical images [13]. Suganya and Amudha’s proposed method uses two encryption algorithms, namely, RC4 and AES, which are the stream cipher and block cipher algorithms, respectively. The main objective of this method is to provide integrity control for medical images, although they are encrypted. Experimental security analysis is conducted using 8-bit ultrasound images and 16-bit positron emission tomography (PET) images [14]. The chaos systems that have become popular today have been used in many types of research. For example: Liang and Qi investigated mechanical analysis of generates the Chen chaotic system to the extensible Kolmogorov system. In Hu et al.’s research, the Chen chaotic system is designed as a pseudorandom sequence producer. In their research, Wang et al. used the memristor chaotic systems (MCSs). Gong et al. provided a method for image encryption based on hyper-chaotic system and discrete fractional random transform. Michail et al.’s research was based on chaotic systems and hash functions to implement totally self-checking (TSC). James et al., in their research based on chaotic systems and hash functions, discussed the performance of SHA-3 256-bit core. Ahmad and Das, based on chaos and hash algorithms, discussed Hardware performance analysis of SHA-256 and SHA-512 algorithms on FPGAs. In their research, Xu et al. improved chaotic cryptosystem based on circular bit shift and XOR operations. Pareek and Patidar, in their research, designed medical image protection based on genetic algorithm and chaotic system. Hua et al., in their study, designed medical image encryption using hash algorithm high-speed scrambling [15–24]. Also, Chai et al. designed a color image encryption method based on dynamic DNA encryption and chaotic system [25] and Ma et al. provided a new method of plaintext-related and chaos-based image encryption [26]. Niyat et al. offered an image encryption algorithm with the rule of cellular automata (CA). CA is a self-establishing construction with a group of cells in which any cell is updated based on certain regulation that are to depend on a limited number of neighboring cells [27]. Chen et al. designed an image encryption method based on hyperchaotic system in the turner transfigure domain. The RGB ingredients of the main color image are encrypted into 1D circulation. [28]. The method in [5] focused on the embedding capacity, but no results are given with respect to other criteria, such as the correlation coefficients entropy, the number of pixel change rate (NPCR), or the uniform average change intensity (UACI). The methods in [6,7] are only suitable for the NPCR, and other values are not suitable. In the method in [9], the values that are obtained are appropriate, but they are only suitable for grayscale images. The results of the methods
in [10–12] are only suitable with respect to their correlation coefficients, but there are no improvements with respect other criteria. The results that are obtained in [13,14] are only suitable with respect to entropy. The methods in [27,28] perform well in color images encrypting and can be used to encrypt medical images. Nevertheless, our method is better in terms of the correlation coefficients, the entropy, the NPCR, and the UACI.

Our aim in this paper is to provide an algorithm that protects medical color images based on chaotic systems and SHA-256 systems. The algorithm is composed of two parts: A high-speed permutation process and adaptive diffusion. For this reason, in Section 2, we will present the basic concepts of chaotic systems and SHA-256 systems. In Section 3, we will describe the proposed algorithm’s equations, and in Section 4, the empirical results from the implementation of the proposed algorithm that is simulated in the MATLAB software will be given. In Section 5, we compare the results of the proposed method with previous methods, and in Section 6, we will explain the quality and appropriateness of this method.

2. Preliminary Work

2.1. Chaotic Systems

Chaos theory is a chapter of mathematics centralization on the action of dynamical systems that are highly sensitive to initial situation. “Chaos” is a notion denoting that within the obvious accidentalness of chaotic systems, there are basically models, stable feedback rings, iteration, self-likeness, fractals, self-formation, and dependence on programming at the initial part, which is known to have sensitive to depend on initial situation.

Little differences in initial situation, like those owing to rounding errors in numerical calculations, output widely in different outcomes for dynamical systems, thus, generally interpretation of the long-term oracle of their action impossible. This action is known as certain chaos, or simply chaos. The theory was tabloid by Edward Lorenz as follows:

Chaos: When the design specifies the future, but the proximate present does not proximately specify the future.

In 1963, Lorenz studied chaotic systems using a nonlinear differential equation, which is one of the first examples of algebraic chaotic systems in dissipative systems. Chaotic systems are very abundant in nature and they are used in many branches of science, such as the physics of dynamics and photonics, medical sciences, chemistry, and demography. In recent years, much research has been done on chaotic systems by scientists and more practical systems have been introduced, such as Chen’s system, Lu’s system, and Qi’s 3D four-wing chaotic system [15–17]. The chaotic system that is used in this algorithm is the Chen-based hyper-chaotic system. The Chen-based hyper-chaotic system in [18] is described as follows:

\[
\begin{align*}
\dot{s} &= a(t - s) \\
\dot{i} &= ds - su + ct - v \\
\dot{u} &= st - bu \\
\dot{v} &= s + w
\end{align*}
\]

where s, t, u, and v are the fixed variables and a, b, c, d, and w are the controlling parameters of the system. The dynamical cycle will be hyper-chaotic when \(a = 36\), \(b = 3\), \(c = 28\), \(d = -16\), and \(-0.7 < w < 0.7\).

2.2. SHA-256 (Secure Hash Algorithm 256)

A cryptographic hash (sometimes called a “digest”) is a type of ‘signature’ for a text or data file. The SHA-256 produces an almost-unique 256-bit (32-byte) signature for a text. The SHA-256 is a type of the deputy hash functions to the SHA-1 (referential to as SHA-2) and is one of the existing powerfulness hash functions. The SHA-256 is not much more complicated to code than the SHA-1 and has not yet been agreement in any path. The 256-bit key makes it a good common-function for the
AES. It is explained in the NIST (National Institute of Standards and Technology) standard ‘FIPS 180-4’. The NIST also prepared a number of test vectors to investigate the validity of its execution [19–21].

To the SHA-256, the message is decomposed to \( n \) blocks with 512 bits, and at the end of its the final block, bit ‘1’ is added to be followed by \( k \) zero bits, where \( k \) is the least nonnegative the answer path of the equation:

\[
l + 1 + k = 448 \mod 512.
\]

Next, a 64-bit binary block that is equivalent to \( l \) is added. Clearly, a “1” followed by \( k \) “0” s that is followed by 64 bits are added at the end of \( M \) to generate a crooked message of length 512 * \( n \) bits. For instance, the 8-bit ASCII message “abc” has a length of

\[
l = 8 \times 3 = 24.
\]

Therefore, the message is padded with a one bit, then 448 – (24 + 1) = 423 zero, and then the 64 bits of the length of message \((11000)_2 = (24)_{10}\). Then, one message plan carries out on the blocks of \( M \), generating the \( W_t \) amount, any of which is to the corresponding \( t \)-th repetition of the transmutation. The transmutation takes \( W_t \), a fixed value, \( K_t \) and the primary amounts \( H(0) \) (in the repetition one) or the values generated in the past repetition; carries out the transmutation procedure; and produces a series of hash values via a number of repetitions. The last produced hash value is considered as the message digest, \( h \) [19]. The SHA-256 needs 64 repetition to generate its message digest. The round contains additives and rational functions that are set to generate the round’s output values. The included NLFs are shown in Equation (2):

\[
\begin{align*}
\text{Ch}(x, y, z) &= (x \cdot y) \oplus (\overline{x} \cdot y) \\
\text{Maj}(x, y, z) &= (x \cdot y) \oplus (x \cdot z) \oplus (x \cdot z)
\end{align*}
\]

\[
\begin{align*}
\sum_{0}^{256} (x) &= \text{ROTR}^2(x) \oplus \text{ROTR}^{13}(x) \oplus \text{ROTR}^{22}(x) \\
\sum_{1}^{256} (x) &= \text{ROTR}^6(x) \oplus \text{ROTR}^{11}(x) \oplus \text{ROTR}^{25}(x)
\end{align*}
\]

where \( \oplus, \cdot, \) and \( \overline{\cdot} \) denote the XOR, AND, and NOT bitwise rational functions, respectively, and \( x, y, \) and \( z \) are 32-bit words. \( \text{ROTR}^X \) Shows \( x \) right round bit spin. According to the 64 \( W_t \) values that are necessary, the first 16 are organized by the 512-bit input block whereas the remaining 48 \( W_t \) amounts are calculated using Equation (3). The functions \( \sigma_0 \) and \( \sigma_1 \) are computed using Equation (4).

\[
W_t = \sigma_1^{(256)}(W_{t-2}) + W_{t-7} + \sigma_0^{(256)}(W_{t-15}) + W_{t-16} | 16 \leq t \leq 63
\]

\[
\begin{align*}
\sigma_1^{(256)} &= \text{ROTR}^7(x) \oplus \text{ROTR}^{18}(x) \oplus \text{SHR}^3(x) \\
\sigma_0^{(256)} &= \text{ROTR}^{17}(x) \oplus \text{ROTR}^{19}(x) \oplus \text{SHR}^{10}(x)
\end{align*}
\]

Here, \( \text{SHR}^X \) stands for the right bit shift. The SHA-256 base transformation rounds are shown in Figure 1.
2.3. SHA-256 Architectures

The performance of the SHA-256 construction’s transformation round is shown in Figure 1. It takes, as inputs, eight 32-bit characters, \((a_{t-1} - h_{t-1})\), the value \(W_{t-1}\), and the stable value \(K_{t-1}\), performs the calculations shown in Figure 1, and generates the values \((a_t - h_t)\) after 64 repetitions [19].

3. Proposed Algorithm

The proposed algorithm has two parts: A high-speed permutation process and adaptive diffusion.

3.1. High-Speed Permutation Process

Step 1. First, the plain image \(P\) of size \(M \times N \times D\) is used as the input; it has the initial state values of \(a_0, b_0, c_0\), and \(d_0\); and uses the SHA-256 function, which is constructed according to the plain image. We consider \(DM = D \times M\) and \(s = \text{sum}(sha256)/(64 \times 256)\), which reshape matrix \(P\) into the 2D matrix \(P_1\). Then, the new values of the chaos system \(a_0, b_0, c_0\), and \(d_0\) are generated using Equation (4) as follows:

\[
\begin{align*}
    a_0 &= s + a_0 - s + a_0 \\
    b_0 &= s + b_0 - s + b_0 \\
    c_0 &= s + c_0 - s + c_0 \\
    d_0 &= s + d_0 - s + d_0
\end{align*}
\]

(5)

Step 2. Then, the initial values and parameters are used to iterate the chaotic systems to obtain the vectors \(a_1, a_2, a_3,\) and \(a_4\) and quantize to generate four different vectors \(PR1, PC1, PR2,\) and \(PC2\), which are as follows:

\[
\begin{align*}
    PR1 &= (|a1| - a1) \times 10^{15} \mod N + 1 \\
    PC1 &= (|a2| - a2) \times 10^{15} \mod DM + 1 \\
    PR2 &= (|a3| - a3) \times 10^{15} \mod N + 1 \\
    PC2 &= (|a4| - a4) \times 10^{15} \mod DM + 1
\end{align*}
\]

(6)

Here, we have to use the \(\text{circshift}\) (shift array circularly) rule and its definition is as follows: If \(A\) and \(B\) are matrixes, \(B = \text{circshift}(A, \text{shiftsize})\) circularly shifts the values in the array \(A\) using the shift size elements [22].
Step 3. We consider $PR, PC \in \mathbb{N}^{DM \times N}$ and for $i = 1$ to $DM$, if $i$ is odd, then we get the following:

$$
\begin{cases}
PR(:, i) = \text{circshift } P1(:, i) \text{ by step } PR1(i) \\
\text{and else}
\end{cases}
$$

(7)

Step 4. For $j = 1$ to $N$, if $j$ is odd, then we get the following:

$$
\begin{cases}
PC(:, j) = \text{circshift } IR(:, j) \text{ by step } IC1(j) \\
\text{and else}
\end{cases}
$$

(8)

Now, $P2 = PC$ is the permutated image.

This Process explained in Algorithm 1 with the title: High-speed Permutation Process.

**Algorithm 1 High-speed Permutation Process**

Input: Image $P$ of size $M \times N \times D$, Initial state: $a0, b0, c0, d0$, and Sha256 value of $P$

Output: Permutated Image

1. Let $DM = D \times M$, $s = \text{sum (sha256)}/(64 \times 256)$, and reshape $P$ to 2-dimension matrix $P1$.

2. Generate a new initial value of the chaotic system: $a0, b0, c0, d0$.

3. Use the initial value and parameters to iterate the chaotic system to get the vectors: $a1, a2, a3, a4$, and quantize to generate four different vectors: $PR1, PC1, PR2, PC2$.

4. Set $PR, PC \in \mathbb{N}^{DM \times N}$

5. for $i = 1$ to $DM$ do

6. if $i$ is odd then

7. $PR(:, i) = \text{circshift } P1(:, i) \text{ by step } PR1(i)$

8. else

9. $PR(:, i) = \text{circshift } P1(:, i) \text{ by step } -PR2(i)$

10. end if

11. end for

12. for $j = 1$ to $N$ do

13. if $j$ is odd then

14. $PC(:, j) = \text{circshift } PR(:, j) \text{ by step } PC1(j)$

15. else

16. $PC(:, j) = \text{circshift } PR(:, j) \text{ by step } -PC2(j)$

17. end if

18. end for

19. Let $PC$ be the permutated imagere

20. turn Permutated Image

3.2. Adaptive Diffusion

Step 1. First, the other initial values and parameters are used to iterate the chaotic systems to obtain the vectors $a110, b110, c110, and d110$, and they are quantized to generate four different vectors $a11, b11, c11, and d11$.

We set $N0$ as a random number, $N00 = N0 + 1$ and $nn = (DM \times N)/2$.

We set $N0$ as a random number, $N00 = N0 + 1$ and $nn = (DM \times N)/2$.

$$
\begin{cases}
a11 = \lfloor (a110 + b110) \times 10^{15} \rfloor \mod 2^3 + 1 \\
b11 = \lfloor (b110 - a110) \times 10^{15} \rfloor \mod 2^3 + 1 \\
c11 = \lfloor (c110 + d110) \times 10^{15} \rfloor \mod 2^8 \\
d11 = \lfloor (c110 + d110) \times 10^{15} \rfloor \mod 2^8
\end{cases}
$$

(9)
Step 2. Set $A, B \in \mathbb{N}^{DM \times N}$ and $i = 1$ to $DM$. If $i \geq 1$ and $i \leq DM/2$, then we get the following:

$$
\begin{align*}
A(i,:) &= a_{11}(((i-1) \times N + 1) : (i \times N),:) \\
B(i,:) &= c_{11}(((i-1) \times N + 1) : (i \times N),:).
\end{align*}
$$

(10)

Otherwise,

$$
\begin{align*}
A(i,:) &= b_{11}(((i-1 - (DM/2)) \times N + 1) : ((i - (DM/2)) \times N),:) \\
B(i,:) &= d_{11}(((i-1 - (DM/2)) \times N + 1) : ((i - (DM/2)) \times N),:).
\end{align*}
$$

(11)

Step 3. Here, we have to apply bitcirshift rule, which is an action that is done on all of the bits of a binary amount, in which they are transformed by a determined number of locations to the left or right. Bitcirshift is used when the operand is being used as a series of bits relatively than generally. In other words, the operand behaves as single bits that show something and are not values [22].

Let $P2 = PC$, and $P3 \in \mathbb{N}^{DM \times N}$. If $i = 1$ to $DM$ and $j = 1$ to $N$, then we get the following:

$$
P3(i, j) = \text{bitcirshift} P2(i, j) \text{by step } A(i, j).
$$

(12)

Step 4. Let $key_{r0} = (a_{110}(1 : N) + b_{110}(1 : N))' / 2$ and $key_{c0} = (c_{110}(1 : DM) + d_{110}(1 : DM)) / 2$, and quantize them as $key_r$ and $key_c$, respectively, in $[0, 255]$.

Let $P2 = PC$, and $P3 \in \mathbb{N}^{DM \times N}$. If $i = 1$ to $DM$, $r1 = \text{cirshift}(B(i,:), [0, 1])$ and $i = 1$, then we get the following:

$$
\begin{align*}
P4R(i,:) &= ((P3(i,:) + r1) \text{mod} \text{256}) \oplus key_r, \\
\text{and else}
\end{align*}
$$

(14)

Step 5. Set $P4R, P4C \in \mathbb{N}^{DM \times N}$. If $i = 1$ to $DM$, $r1 = \text{cirshift}(B(i,:), [0, 1])$ and $i = 1$, then we get the following:

$$
\begin{align*}
P4R(i,:) &= ((P3(i,:) + r1) \text{mod} \text{256}) \oplus key_r, \\
P4R(i,:) &= ((P3(i,:) + r1) \text{mod} \text{256}) \oplus P4R((i-1,:),:).
\end{align*}
$$

(15)

Step 6. If $j = 1$ to $N$, $c1 = \text{cirshift}(B(j,:), [j, 0])$ and $j = 1$, then we get the following:

$$
\begin{align*}
P4C(:, j) &= ((P4R(:, j) + c1) \text{mod} \text{256}) \oplus key_c, \\
\text{and else}
\end{align*}
$$

P4C, the final image, is encrypted.

This Process explained in Algorithm 2 with the title: Adaptive diffusion.

Our proposed algorithm is a symmetric algorithm. The decryption procedure is the opposite of the encryption method and decryption is done using the encryption method’s formulas. This is shown in Figure 2.

Remark 1. The proposed algorithm is suitable for all color images (RGB). Because medical images are very important in today’s technological world, we decided to use the proposed algorithm for medical images.
Algorithm 2  Adaptive diffusion

\textbf{Input:} Input data from permutation procession  
\textbf{Output:} Encrypted Image

1: Use other initial values and parameters to iterate the chaotic system again to get the vectors: $a_{110}$, $b_{110}$, $c_{110}$, $d_{110}$, and quantize them to generate four different vectors: $a_{11}$, $b_{11}$, $c_{11}$, $d_{11}$.

2: Set $A, B \in \mathbb{N}^{DM \times N}$

3: for $i = 1$ to $DM$ do
4: \hspace{1em} if $i \geq 1$ and $i \leq DM/2$ then
5: \hspace{2em} $A(i,:) = a_{11}(((i - 1) \times N + 1) : (i \times N),:)$
6: \hspace{2em} $B(i,:) = c_{11}(((i - 1) \times N + 1) : (i \times N),:)$
7: \hspace{1em} else
8: \hspace{2em} $A(i,:) = b_{11}(((i - 1 - (DM/2)) \times N + 1) : ((i - (DM/2)) \times N + 1) : (((i - (DM/2)) \times N),:)$
9: \hspace{1em} end if
10: end for
11: Let $P_2 = P_C$, and set $P_3 \in \mathbb{N}^{DM \times N}$
12: for $i = 1$ to $DM$ do
13: \hspace{1em} for $j = 1$ to $N$ do
14: \hspace{2em} $P_3(i,j) = \text{bitcircshift} P_2(i,j)$ by step $A(i,j)$
15: \hspace{1em} end for
16: end for
17: Let $key_{r0} = (a_{110}(1 : N) + b_{110}(1 : N))'/2$, $key_{c0} = (c_{110}(1 : DM) + d_{110}(1 : DM))/2$, and quantize them $key_r, key_c$ in $[0, 255]$
18: Set $P_4R, P_4C \in \mathbb{N}^{DM \times N}$
19: for $i = 1$ to $DM$ do
20: \hspace{1em} $r_1 = \text{circshift} (B(i,:),[0,i])$
21: \hspace{1em} if $i = 1$ then
22: \hspace{2em} $P_4R(i,:) = \text{bitxor}(\text{mod}((P_3(i,:) + r_1), 256), key_r)$
23: \hspace{1em} else
24: \hspace{2em} $P_4R(i,:) = \text{bitxor}(\text{mod}((P_3(i,:) + r_1), 256), P_4R((i-1,:),:))$
25: \hspace{1em} end if
26: end for
27: for $j = 1$ to $N$ do
28: \hspace{1em} $c_1 = \text{circshift} (B(:,j),[j,0])$
29: \hspace{1em} if $j = 1$ then
30: \hspace{2em} $P_4C(:,j) = \text{bitxor}(\text{mod}((P_4R(:,j) + c_1), 256), key_c)$
31: \hspace{1em} else
32: \hspace{2em} $P_4C(:,j) = \text{bitxor}(\text{mod}((P_4R(:,j) + c_1), 256), P_4C(:,(j-1)))$
33: \hspace{1em} end if
34: end for
35: Let $P_4C$ be the final encrypted image
36: return Encrypted Image

The decryption process is inverse encryption process.

\textbf{Input:} Input data from permutation procession  
\textbf{Output:} Encrypted Image
Figure 2. Schematic of the proposed encryption algorithm (a), and schematic of the proposed decryption algorithm (b).

4. Experiment Result and Security Analysis

In this section, we implemented the proposed algorithm on two medical color images using the MATLAB 2017a software environment (in personal computer with core i7, 3.4GHz, RAM 16GB). As we stated in the introduction, the medical color images are obtained using the Medipix3RX chip technology that is used in today’s imaging devices [1–4].

For example, four color images 256x256 in size have been selected as the plain images. In Figure 3, images b, e, h, and k are images that are encrypted by the proposed algorithm for the plain images a, d, g, and j, respectively; and images c, f, i, and l are the decrypted images.
Figure 3. (a,d,g,j): Plain images. (b,e,h,k): Respective encrypted images. (c,f,i,l): Respective decrypted images. Initial values for all images: (a0 = 0.1314, b0 = 0.5214, c0 = 0.3698, and d0 = 0.8419). Values of the chaotic system: Image (a): sha256 = '9ADBBFB88CFD90C23CE114E47402054E6DDC4182510E80980EA7151CD11E6D18', image (d): sha256 = '8BF6A886E4B58D2B530749EE9BAB54A3C360D406DC5B901CC169D78709FA3CA09', image (g): sha256 = '49A22186DB65786789CD1391CDE4D9737039E758F39A4SC59D8338DE05353337', and image (j): sha256 = '6EB1A45F27A6709A25265835F05BC11E057255DA81359299631F4724936C8'.

4.1. Security Analysis

As seen, it is not possible to visually compare the plain images with images that were obtained from the decryption process, and the measures such as the correlation coefficients of two adjacent
4.1. Security Analysis

As seen, it is not possible to visually compare the plain images with images that were obtained from the decryption process, and the measures such as the correlation coefficients of two adjacent pixels in the plain image and the cipher image, the entropy, the NPCR (number of pixel change rate), and the UACI (uniform average change intensity) should be mathematically examined. We consider an example of a baby’s image.

4.2. Histogram Analysis

Color images include three main color channels (red, green, and blue), and these images are called RGB images. Figure 4 shows the histograms of these three channels that are observed for the baby’s image.

![Figure 4](image)

**Figure 4.** (a): Plain image baby, and (b–d): R, G, and B histograms, respectively.

In Figure 5, we can see the baby’s decrypted image from the three-channel color histograms.

![Figure 5](image)

**Figure 5.** (a): Baby’s decrypted image, and (b–d): R, G, and B histograms, respectively.
Chi-square Analysis

Statistical analysis is a type of the commonplace cryptology procedures. The monotony of the histogram of cipher demonstrates the strength of the encryption path to statistical analysis. The ocular effect of the histogram is not sufficient to verify the accident of a cipher image’s pixel values [26]. To quantitatively measure the monotony of the histogram, we use the chi-square test as a metric. The description of the chi-square is as follows:

$$\chi^2_{\text{exp}} = \sum_{i=1}^{Q} \frac{(O_i - E_i)^2}{E_i},$$

where $O_i$ is the observed incidence frequency of each rate on the histogram of the ciphered image, $E_i$ is the envisage incidence frequency of the uniform distribution, and $M \times N$ is the length of an image trail. For an ideal image encryption system, the empirical chi-square value should be less than the theoretic amount. With the importance level of 0.05, the theoretic chi-square value is 293 [26]. The chi-square test conclusions and transition rates are listed in Tables 1 and 2. All the test images transition the test, which shows that our plan has a satisfying encryption effect.

Table 1. Chi-square test results (part 1).

| Images | Channels | R      | G      | B      | R      | G      | B      |
|--------|----------|--------|--------|--------|--------|--------|--------|
|        | $x^2_{\text{test}}$ | 247.0762 | 204.9082 | 220.0742 | 251.8379 | 260.2695 | 256.4063 |
|        | $x^2_{255,0.05}$  | 293     | 293     | 293     | 293     | 293     | 293     |
| Pass or not | Yes       | Yes     | Yes     | Yes     | Yes     | Yes     | Yes     |

Table 2. Chi-square test results (part 2).

| Images | Channels | R      | G      | B      | R      | G      | B      |
|--------|----------|--------|--------|--------|--------|--------|--------|
|        | $x^2_{\text{test}}$ | 285.8359 | 261.9980 | 247.2793 | 250.0836 | 210.7622 | 231.1039 |
|        | $x^2_{255,0.05}$  | 293     | 293     | 293     | 293     | 293     | 293     |
| Pass or not | Yes       | Yes     | Yes     | Yes     | Yes     | Yes     | Yes     |

4.3. Correlation Analysis

The correlation coefficient of two adjacent pixels in the plain image and the cipher image is one of the important factors in determining the quality of image encryption algorithms [23]. In Figure 6, we can see the correlation histograms for the plain image and cipher image, and the correlation histograms are shown in three directions: Horizontal, vertical, and diagonal.
Figure 6. Correlation histograms. (a–c): For the plain image; and (d–f): For the cipher image.

In Table 3, the numerical values of the correlation for the plain image and the cipher image are given, and the values in the table are calculated for three directions: Horizontal, vertical, and diagonal. We can specifically see that the correlation coefficients of the plain image are near to 1, however the correlation coefficients of the cipher-image are about equal 0, which may explain why the designed encrypted algorithm has a powerful resistance to possible statistical attacks. The table specifies that the proposed algorithm has the required quality. The correlation coefficient of two adjacent pixels in the plain image and the cipher image is obtained as follows:

$$r_{xy} = \frac{E((x_i - E(x))(y_i - E(y)))}{\sqrt{D(x)D(y)}}$$

(17)

where

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

$$D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2.$$ 

$$E((x_i - E(x))(y_i - E(y))) = \text{cov}(x, y), E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i$$ is the expected value, $$N$$ is the number of image pixels, and $$D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2$$ is the variance. $$x$$ and $$y$$ are the gray values of two adjacent pixels, and $$N$$ is the total number of pixels that are chosen from the image.

Table 3. Correlation coefficients in the plain image and the cipher image.

| Image | Channel | H   | V   | D   | H   | V   | D   |
|-------|---------|-----|-----|-----|-----|-----|-----|
| a     | R       | 0.9952 | 0.9978 | 0.9897 | -0.0115 | 0.0048 | -0.0026 |
|       | G       | 0.9825 | 0.9881 | 0.9688 | 0.0109 | 0.0097 | -0.0161 |
|       | B       | 0.9833 | 0.9803 | 0.9615 | -0.0224 | -0.0091 | 0.0062 |
|       | R       | 0.9217 | 0.8673 | 0.8580 | 0.0097 | -0.0091 | 0.0094 |
| d     | G       | 0.8575 | 0.7647 | 0.7328 | 0.0015 | 0.0075 | -0.0055 |
|       | B       | 0.9140 | 0.8966 | 0.8626 | 0.0137 | 0.0051 | 0.0065 |
4.4. Entropy Analysis

The entropy randomly measures the data sequence and is defined as follows [24]:

\[
H(S) = \sum_{i=0}^{2^N-1} P(s_i) \log\left(\frac{1}{P(s_i)}\right)
\]

where \( N \) is the number of grayscale levels in an image, and \( P(s_i) \) is the incidence possibility of grayscale “I” in the image. The entropy amount will be 8 for images that are wholly accidentally generated. The nearer the entropy of an encryption method is to 8, the less foreseeable it is, and thus, the more secure the plan. The entropies for the designed encryption method have been measured for a sample image and the conclusions are shown in Table 4.

| Image Channel | Plain-Text H | V | D | Cipher-Text H | V | D |
|---------------|-------------|---|---|----------------|---|---|
| R             | 0.9942      | 0.9968 | 0.9887 | -0.0125 | 0.0047 | -0.0025 |
| G             | 0.9815      | 0.9871 | 0.9678 | 0.0109 | 0.0095 | -0.0160 |
| B             | 0.9823      | 0.9793 | 0.9605 | -0.0214 | -0.0093 | 0.0063 |
| R             | 0.9217      | 0.8663 | 0.8570 | 0.0095 | -0.0092 | -0.0093 |
| G             | 0.8575      | 0.7637 | 0.7318 | 0.0014 | 0.0073 | -0.0056 |
| B             | 0.9140      | 0.8956 | 0.8616 | 0.0135 | 0.0052 | 0.0066 |

4.5. NPCR (Number of Pixel Change Rate) and UACI (Uniform Average Change Intensity)

In a differential attack, a little variation is built to the plain image, and the designed algorithm is employed to encrypt the plain image before and after this variation. These two encrypted images have been evaluated to detect any possible connection between the plain image and the cipher image. The (UACI) and the (NPCR) are two indicator that are regularly used by researchers to test the differential attack resistor of any image encryption method [12].

Suppose that \( C_1 \) and \( C_2 \) are two cipher images that are encrypted from two plain images with only one-bit difference. The NPCR and UACI are defined as follows:

\[
NPCR(C_1, C_2) = \sum_{i,j} \frac{H(i, j)}{M} \times 100\%
\]

(19)

and

\[
UACI(C_1, C_2) = \sum_{i,j} \frac{|C_1(i, j) - C_2(i, j)|}{(S - 1) \times M} \times 100\%
\]

(20)

where \( M \) shows the total number of pixels in any cipher-image, \( S \) illustrates the number of allowed pixels, and \( H(i, j) \) demonstrates the difference between \( C_1 \) and \( C_2 \), which is specified as follows.
\[ H(i, j) = \begin{cases} 
0, & \text{if } C_1(i, j) = C_2(i, j) \\
1, & \text{if } C_1(i, j) \neq C_2(i, j) 
\end{cases} . \tag{21} \]

The larger the NPCR and UACI are, the better the quality of the algorithm. For four randomly selected points, the NPCRs and UACIs are listed in Table 5.

| Position     | (12,34) | (34,56) | (56,78) | (78,90) |
|--------------|---------|---------|---------|---------|
| NPCR         | 99.6232 | 99.6215 | 99.6170 | 99.5971 |
| UACI         | 33.4574 | 33.4952 | 33.5326 | 33.4755 |

4.6. Key Space

The key space for encryption algorithms should be large enough to withstand potential attacks. The minimum key space should be \(2^{100}\). The input values of \((x_0, y_0, z_0, h_0, SHA256)\) act as a secret key, and, based on this, the secret key space is \(10^{14} \times 10^{14} \times 10^{14} \times 10^{14} \times 2^{128} = 10^{56} \times 2^{128}\). This indicates that the designed method has good key space.

4.7. Key Sensibility Analysis

A safe encryption system must be sensitive to the key; for example, the little change of encryption keys can lead to very different cipher image, and a small change of the decryption keys cannot decrypt the image. Several key sensitivity tests are performed. Figure 7 shows the encrypted images of the baby (plain image b). Figure 3a shows the encrypted image using user keys with a 1-bit difference. The plain encrypted image is indicated in Figure 3b. When the keys of the initial state \(x_0, y_0, z_0, h_0,\) and \(SHA256\) are changed by one bit (i.e., \(10^{-14}\) for \(x_0, y_0, z_0,\) and \(h_0\) and \(2^{-128}\) for \(SHA256\)), the five new encrypted images are obtained and shown in Figure 7a–e. We compare them with the image in Figure 7e, and the five differential images are shown in Figure 7f–j. This shows that there are very big differences between the images in Figure 7e,f–j.

In addition, to experience the capability of the designed method to resist the cipher text attack, the keys \(x_0, y_0, z_0,\) and \(h_0\) will be modified by \(10^{-14}\) and \(SHA256\) will be changed by \(2^{-128}\) to decrypt the plain encrypted image. The decrypted images are indicated in Figure 7, which are wholly different from the plain image. Therefore, it can be seen that the cipher text cannot be suitably decrypted without the correct keys, which shows that the proposed method can effectively hamper the cipher text merely attack.
Table 5. Number of pixel change rates (NPCRs) and uniform average change intensities (UACIs) of different positions (%).

| Position | NPCR      | UACI      |
|----------|-----------|-----------|
| (12,34)  | 99.6232   | 33.4574   |
| (34,56)  | 99.6215   | 33.4952   |
| (56,78)  | 99.6170   | 33.5326   |
| (78,90)  | 99.5971   | 33.4755   |

4.6. Key Space

The key space for encryption algorithms should be large enough to withstand potential attacks. The minimum key space should be $2^{100}$. The input values of $(x_0, y_0, z_0, h_0, SHA256)$ act as a secret key, and, based on this, the secret key space is $10^{14} \times 10^{14} \times 10^{14} \times 10^{14} \times 2^{128} = 10^{56} \times 2^{128}$. This indicates that the designed method has good key space.

4.7. Key Sensibility Analysis

A safe encryption system must be sensitive to the key; for example, the little change of encryption keys can lead to very different cipher images, and a small change of the decryption keys cannot decrypt the image. Several key sensitivity tests are performed. Figure 7 shows the encrypted images of the baby (plain image b). Figure 3a shows the encrypted image using user keys with a 1-bit difference. The plain encrypted image is indicated in Figure 3b. When the keys of the initial state $x_0, y_0, z_0, h_0, and SHA256$ are changed by one bit (i.e., $10^{-14}$ for $x_0, y_0, z_0, and h_0$ and $2^{-128}$ for SHA256), the five new encrypted images are obtained and shown in Figure 7a–e. We compare them with the image in Figure 7e, and the five differential images are shown in Figure 7f–j. This shows that there are (a), (b), (c), (d), (e).

Figure 7. Cipher-images within authentic primary keys and the differences between them and the plain encrypted images: (a–e) Five new encrypted images with the keys $x_0 + 10^{-14}, y_0 + 10^{-14}, z_0 + 10^{-14}, h_0 + 10^{-14}, and SHA256+2^{-128}$, respectively; and (f–j) differences between the unauthentic encrypted images and the plain image.

4.8. Known-Plain Image and Chosen-Plain Image Analysis

Clearly, some specific images are selected to test the selected plain-text attack, like a full-white image in Figure 8a and a full-black image in Figure 8d. The results are shown in Figure 8, which indicate that the cryptology is appropriate for these specific images and can resist the chosen-plain-text attack. In Table 6, we can see the entropy values for all-white and all-black images. It can be seen that all the values that are obtained are close to 8, indicating the suitability of the proposed algorithm.
Figure 7. Cipher-images within authentic primary keys and the differences between them and the plain encrypted images: (a)–(e) Five new encrypted images with the keys $x_0 + 10^{-14}$, $y_0 + 10^{-14}$, $z_0 + 10^{-14}$, $h_0 + 10^{-14}$, and SHA256$^2 - 128$, respectively; and (f)–(j) differences between the unauthentic encrypted images and the plain image. In addition, to experience the capability of the designed method to resist the cipher text attack, the keys $x_0$, $y_0$, $z_0$, and $h_0$ will be modified by $10^{-14}$ and SHA256 will be changed by $2^{128}$ to decrypt the plain encrypted image. The decrypted images are indicated in Figure 7, which are wholly different from the plain image. Therefore, it can be seen that the cipher text cannot be suitably decrypted without the correct keys, which shows that the proposed method can effectively hamper the cipher text merely attack.

4.8. Known-Plain Image and Chosen-Plain Image Analysis

Clearly, some specific images are selected to test the selected plain-text attack, like a full-white image in Figure 8a and a full-black image in Figure 8d. The results are shown in Figure 8, which indicate that the cryptology is appropriate for these specific images and can resist the chosen-plain-text attack. In Table 6, we can see the entropy values for all-white and all-black images. It can be seen that all the

Figure 8. Cont.
**Figure 8.** Selected plain-image test for white and black images, display (a) the full-white image, (b) the cipher image of panel (a), (c) the histogram of channel R (b), (d) the full-black image, (e) the cipher image of panel (d), and (f) the histogram of channel R.

**Table 6.** Information entropy of the full-black image and full-white image.

| Image | R    | G    | B    |
|-------|------|------|------|
| Black | 7.9993 | 7.9994 | 7.9993 |
| White | 7.9994 | 7.9994 | 7.9993 |

**4.9. Noise Attack and Occlusion Attack**

In the digital world, the images will unexpectedly experience noise and occlude attacks in the transition process, and an effective cryptology must be strong against them. The baby image is used as the test image. Figure 9 shows the noisy cipher images that are contaminated by Gaussian noise (GN), salt and pepper noise (SPN), and speckle noise (SN) with different noise compression and their decrypted images. As seen from Figure 10, the most information of the plain image can be intuitively identified from the decrypted image’s presentation of the cipher images with different occlusion effects and their corresponding retrieved images. Specifically, the decrypted images still include most data of the baby image. The PSNR (peak signal-to-noise-ratio) is employed to compute the condition of the decrypted image after a possible attack. For a gray image, the PSNR and MSE can be computed as follows:
where MSE shows the mean square error between the cipher image \(I_1\) and the plain image \(I_2\). The results are explained in Tables 7 and 8. 

\[
PSNR = 10 \times \log_{10} \left(\frac{255 \times 255}{MSE}\right)(db) \tag{22}
\]

\[
MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} ||I_1(i, j) - I_2(i, j)||^2
\]

\[
MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} ||I_1(i, j) - I_2(i, j)||^2 \tag{23}
\]

where MSE shows the mean square error between the cipher image \(I_1\) and the plain image \(I_2\), and \(m\) and \(n\) are the width and height, respectively [25]. The results are explained in Tables 7 and 8.

![Figure 9. Noise attack test results. (a,b) Gaussian noise (GN); (c,d) salt and pepper noise (SPN); (e,f) speckle noise (SN).](image)

| Item         | PSNR  |
|--------------|-------|
|              | R     | G     | B     |
| Salt and Pepper | 34.2863 | 33.6124 | 33.3711 |
| Gaussian     | 30.6104 | 29.9219 | 29.7102 |
| Speckle      | 30.6023 | 29.8951 | 29.7059 |

![Figure 10. Cont.](image)
5. Comparison

In this section, we will compare the results that were obtained from the proposed algorithm with previous algorithms. The results that were obtained from the designed method are measured with the methods in References [10,24] with respect to their correlation, entropy, NPCR, and UACI, and the key space was compared with references [9,12,24]. The results of the comparison are shown in Tables 9–14.

Table 9. Correlation coefficients results in the original image and the encrypted image for the proposed method and the two methods that were presented in existing methods.

| Image | Methods | Channel | Plain Image | Cipher Image |
|-------|---------|---------|-------------|--------------|
|       |         | H       | V           | D            | H            | V           | D            |
| Image a | proposed | R       | 0.9952      | 0.9978       | 0.9897       | -0.0115     | 0.0048       | -0.0026      |
|        |         | G       | 0.9825      | 0.9881       | 0.9688       | 0.0109       | 0.0097       | -0.0161      |
|        |         | B       | 0.9833      | 0.9803       | 0.9615       | -0.0224      | -0.0091      | 0.0062       |
|        | Algorithm [10] | R       | 0.9952      | 0.9978       | 0.9897       | -0.0122      | -0.0117      | -0.0238      |
|        |         | G       | 0.9825      | 0.9881       | 0.9688       | -0.0113      | 0.0079       | -0.0230      |
|        |         | B       | 0.9833      | 0.9803       | 0.9615       | -0.0099      | 0.0149       | 0.0092       |
|        | Algorithm [24] | R       | 0.9952      | 0.9978       | 0.9897       | -0.0114      | -0.0110      | -0.0174      |
|        |         | G       | 0.9825      | 0.9881       | 0.9688       | -0.0206      | -0.0071      | 0.0180       |
|        |         | B       | 0.9833      | 0.9803       | 0.9615       | -0.0134      | -0.0106      | -0.00194     |
### Table 9. Cont.

| Image | Methods | Channel | Plain Image | Cipher Image |
|-------|---------|---------|-------------|--------------|
|       |         |         | H  V  D     | H  V  D      |
|       |         |         | R 0.9217 0.8673 0.8580 | 0.0097 | -0.0091 | -0.0094 |
|       |         |         | G 0.8575 0.7647 0.7328 | 0.0115 | 0.0075 | -0.0035 |
|       |         |         | B 0.9140 0.8966 0.8626 | 0.0137 | 0.0051 | 0.0065 |
| Image d | Algorithm [10] |         | R 0.9217 0.8673 0.8580 | -0.0133 | -0.0095 | -0.0070 |
|       |         |         | G 0.8575 0.7647 0.7328 | -0.0182 | 0.0230 | -0.0056 |
|       |         |         | B 0.9140 0.8966 0.8626 | -0.0282 | -7.3230 × 10^{-4} | -0.0073 |
|       | Algorithm [24] |         | R 0.8575 0.7647 0.7328 | -0.0059 | 0.0109 | 1.7711 × 10^{-4} |
|       |         |         | B 0.9140 0.8966 0.8626 | -0.0216 | -0.0089 | 0.0077 |
| Image g | Algorithm [10] |         | R 0.9942 0.9968 0.9887 | -0.0125 | 0.0047 | -0.0025 |
|       |         |         | G 0.9815 0.9871 0.9678 | 0.0109 | 0.0095 | -0.0160 |
|       |         |         | B 0.9823 0.9793 0.9605 | -0.0214 | -0.0093 | 0.0063 |
|       | Algorithm [24] |         | R 0.9942 0.9968 0.9887 | -0.0136 | 0.0066 | -0.0035 |
|       |         |         | G 0.9815 0.9871 0.9678 | 0.0113 | 0.0103 | -0.0190 |
|       |         |         | B 0.9823 0.9793 0.9605 | -0.0237 | -0.0098 | 0.0073 |
| Image j | Algorithm [10] |         | R 0.9942 0.9968 0.9887 | -0.0129 | 0.0049 | -0.0076 |
|       |         |         | G 0.9815 0.9871 0.9678 | 0.0111 | 0.0101 | -0.0171 |
|       |         |         | B 0.9823 0.9793 0.9605 | -0.0231 | -0.0156 | 0.0067 |
|       | Algorithm [24] |         | R 0.9217 0.8663 0.8570 | 0.0095 | -0.0092 | -0.0093 |
|       |         |         | G 0.8575 0.7637 0.7318 | 0.0014 | 0.0073 | -0.0056 |
|       |         |         | B 0.9140 0.8956 0.8616 | 0.0135 | 0.0052 | 0.0066 |
|       | Algorithm [10] |         | R 0.9217 0.8663 0.8570 | 0.0115 | -0.0102 | -0.0105 |
|       |         |         | G 0.8575 0.7637 0.7318 | 0.0084 | 0.0094 | -0.0083 |
|       |         |         | B 0.9140 0.8956 0.8616 | 0.0196 | 0.0067 | 0.0089 |
|       | Algorithm [24] |         | R 0.9217 0.8663 0.8570 | 0.0115 | -0.0111 | -0.0106 |
|       |         |         | G 0.8575 0.7637 0.7318 | 0.0082 | 0.0103 | -0.0074 |
|       |         |         | B 0.9140 0.8956 0.8616 | 0.0161 | 0.0083 | 0.0090 |

### Table 10. Information entropies of the cipher images and plain images, part 1.

| Image | Channel | Proposed Algorithm [10] | Method [24] |
|-------|---------|-------------------------|-------------|
|       |         | Proposed Algorithm d    | Method a     |
| Plainimage | R 6.9581 | 6.981 6.9581 7.7047 | 7.7047 7.7047 |
|       | G 6.8945 | 6.8945 6.8945 7.4724 | 7.4724 7.4724 |
|       | B 6.1365 | 6.1365 6.1365 7.7502 | 7.7502 7.7502 |
| RGB   | R 7.2528 | 7.2528 7.2528 7.7604 | 7.7604 7.7604 |
|       | G 7.9993 | 7.9997 7.9997 7.9997 | 7.9997 7.9997 |
|       | B 7.9993 | 7.9997 7.9997 7.9997 | 7.9997 7.9997 |
|       | RGB 7.9997 | 7.9997 7.9997 7.9997 | 7.9997 7.9997 |
Table 11. Information entropies of the cipher images and plain images, part 2.

| Image | Channel | Proposed Algorithm Image g | Method [10] Image g | Method [24] Image g | Proposed Algorithm Image j | Method [10] Image j | Method [24] Image j |
|-------|---------|----------------------------|-------------------|-------------------|----------------------------|-------------------|-------------------|
| Plainimage | R | 6.9571 | 6.9571 | 6.9571 | 7.7067 | 7.7067 | 7.7067 |
| | G | 6.8955 | 6.8955 | 6.8955 | 7.4734 | 7.4734 | 7.4734 |
| | B | 6.1355 | 6.1355 | 6.1355 | 7.7512 | 7.7512 | 7.7512 |
| | RGB | 7.2548 | 7.2548 | 7.2548 | 7.7614 | 7.7614 | 7.7614 |
| Cipherimage | R | 7.9982 | 7.9982 | 7.9982 | 7.9993 | 7.9993 | 7.9993 |
| | G | 7.9983 | 7.9979 | 7.9982 | 7.9995 | 7.9995 | 7.9994 |
| | B | 7.9994 | 7.9985 | 7.9994 | 7.9981 | 7.9980 | 7.9979 |
| | RGB | 7.9996 | 7.9995 | 7.9996 | 7.9994 | 7.9993 | 7.9994 |

Table 12. NPCRs and UACIs at different positions (%), Part 1.

| Position | Proposed Algorithm Position (12, 34) | Method [10] Position (12, 34) | Method [24] Position (12, 34) | Proposed Algorithm Position (34, 56) | Method [10] Position (34, 56) | Method [24] Position (34, 56) |
|----------|--------------------------------------|-------------------------------|-------------------------------|--------------------------------------|-------------------------------|-------------------------------|
| NPCR     | 99.6232                             | 99.6222                       | 99.6218                       | 99.5015                             | 99.6187                       |
| UACI     | 33.4574                             | 33.4504                       | 33.4767                       | 33.4952                             | 33.3952                       | 33.4850                       |

Table 13. NPCRs and UACIs at different positions (%), Part 2.

| Position | Proposed Algorithm Position (36, 78) | Method [10] Position (36, 78) | Method [24] Position (36, 78) | Proposed Algorithm Position (78, 90) | Method [10] Position (78, 90) | Method [24] Position (78, 90) |
|----------|--------------------------------------|-------------------------------|-------------------------------|--------------------------------------|-------------------------------|-------------------------------|
| NPCR     | 99.6170                             | 99.4170                       | 99.6123                       | 99.5971                             | 99.1971                       | 99.6223                       |
| UACI     | 33.5326                             | 33.2326                       | 33.3979                       | 33.4755                             | 33.3755                       | 33.4759                       |

Table 14. Comparison of the proposed algorithm’s key space with other algorithms.

| Algorithm | Proposed | [9] | [12] | [24] |
|-----------|----------|-----|------|------|
| Key space | $10^{56} \times 2^{128}$ | $2^{192}$ | $2^{128}$ | $2^{256}$ |

6. Conclusions

In this paper, we present a new algorithm based on chaotic systems to protect these images against attacks. The proposed algorithm has two main parts: A high-speed permutation process and adaptive diffusion, which lead to a very efficient and reliable approach in this regard. By examining the results that were obtained from the implementation of the proposed algorithm in the MATLAB software environment and comparing these results with existing methods, it is observed that the designed method is better than those algorithms with respect to the important factors that are mentioned. Such that, to quantitatively evaluate the uniformity of the histogram, the chi-square test is done and the obtained results are desirable. Also, key sensitivity analysis shows that the image is not decrypted with a small change in the key, which indicates that the algorithm is suitable. Clearly, particular images are selected to experiment the selected plain-text, such as a full-white image and a full-black image. Entropy results obtained from the implementation of the algorithm on this type of images show that the proposed method is suitable for this particular type of images. In the real world, the images will inevitably experience noise and occlude attacks while shifting, and an effective cryptosystem must be powerful versus them. The obtained results from noise and occlusion attacks analyses show that the proposed algorithm can withstand against these types of attacks. It is also observed that the values that are obtained with respect to the entropy, NPCR, and UACI are better than those from the methods in existing papers. As we have already mentioned, the key space should be large enough (at least $2^{100}$).
Compared to the old methods, we observe that the key space of our method is very large and more resistant than other methods in dealing with attacks.

**Author Contributions:** For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “conceptualization, S.S.M., Y.C. and C.T; methodology, S.S.M., Y.C. and C.T; software, S.S.M., Y.C. and C.T; validation, S.S.M., Y.C. and C.T; formal analysis, S.S.M., Y.C. and C.T; investigation, S.S.M., Y.C. and C.T; resources, S.S.M., Y.C. and C.T; data curation, S.S.M., Y.C. and C.T; writing—original draft preparation, S.S.M.; writing—review and editing, S.S.M. and C.T; visualization, S.S.M., Y.C. and C.T; supervision, C.T; project administration, C.T; funding acquisition, C.T.

**Funding:** This research was funded by the Foundation of National Natural Science of China (No. 61772147), Guangdong Province Natural Science Foundation of major basic research and Cultivation project (No. 2015A030308016), Project of Ordinary University Innovation Team Construction of Guangdong Province (No. 2015KCXTD014), Collaborative Innovation Major Projects of Bureau of Education of Guangzhou City (No. 1201610005) and National Cryptography Development Fund (No. MMJJ20170117).

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

**References**

1. Bensalah, M.; Rhaskali, Y.; Azzaz, M.S. Medical Images Encryption Based on Elliptic Curve Cryptography and Chaos Theory. In Proceedings of the 2018 International Conference on Smart Communications in Network Technologies (SaCoNeT), El Oued, Algeria, 27–31 October 2018; pp. 222–226.

2. Mohamed Parvees, M.Y.; Abdul Samath, J.; Parameswaran Bose, B. Medical Images are Safe—An Enhanced Chaotic Scrambling Approach. *J. Med. Syst.* 2017, 41, 167. [CrossRef] [PubMed]

3. Rinkel, J.; Magalhães, D.; Wagner, F.; Frojdh, E.; Sune, R.B. Equalization method for Medipix3RX. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc.Equip.* 2015, 801, 1–6. [CrossRef]

4. Gimenez, E.N.; Astrømskas, V.; Horswell, I.; Omar, D.; Spiers, J.; Tartoni, N. Development of a Schottky CdTe Medipix3RX hybrid photon counting detector with spatial and energy resolving capabilities. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip.* 2016, 824, 101–103. [CrossRef]

5. Abdel-Nabi, H.; Al-Haj, A. Medical imaging security using partial encryption and histogram shifting watermarking. In Proceedings of the 2017 8th International Conference on Information Technology (ICITT), Amman, Jordan, 17–18 May 2017; pp. 802–807.

6. Abdmouleh, M.K.; Khalifallah, A.; Bouhlel, M.S. A Novel Selective Encryption DWT-Based Algorithm for Medical Images. In Proceedings of the 2017 14th International Conference on Computer Graphics, Imaging and Visualization, Marrakesh, Morocco, 23–25 May 2017; pp. 79–84.

7. Lakshmi, C.; Thenmozhi, K.; Rayappan, J.B.B.; Amirtharajan, R. Encryption and watermark-treated medical image against hacking disease—An immune convention in spatial and frequency domains. *Comput. Methods Progr. Biomed.* 2018, 159, 11–21. [CrossRef] [PubMed]

8. Cao, W.; Zhou, Y.; Chen, C.L.P.; Xia, L. Medical image encryption using edge maps. *Signal Process.* 2017, 132, 96–109. [CrossRef]

9. Ismail, S.M.; Said, L.A.; Radwan, A.G.; Madian, A.H.; Abu-Elyazeed, M.F. Generalized double-humped logistic map-based medical image encryption. *J. Adv. Res.* 2018, 10, 85–98. [CrossRef] [PubMed]

10. Jeong, H.; Park, K.; Cho, S.; Kim, S. Color Medical Image Encryption Using Two-Dimensional Chaotic Map and C-MLCA. In Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUNFN), Prague, Czech Republic, 3–6 July 2018; pp. 501–504.

11. Ke, G.; Wang, H.; Zhou, S.; Zhang, H. Encryption of medical image with most significant bit and high capacity in piecewise linear chaos graphics. *Measurement* 2019, 135, 385–391. [CrossRef]

12. Nematzadeh, H.; Enayatifar, R.; Motameni, H.; Guimaraes, F.G.; Coelho, V.N. Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices. *Opt. Lasers Eng.* 2018, 110, 24–32. [CrossRef]

13. Laiphrakpam, D.S.; Kumanthom, M.S. Medical image encryption based on improved ElGamal encryption technique. *Optik* 2017, 147, 88–102. [CrossRef]

14. Suganya, G.; Amudha, K. Medical image integrity control using joint encryption and watermarking techniques. In Proceedings of the 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, India, 6–8 March 2014; pp. 1–5.
15. Liang, X.; Qi, G. Mechanical analysis of Chen chaotic system. *Chaos Solitons Fractals* **2017**, *98*, 173–177. [CrossRef]

16. Hu, H.; Liu, L.; Ding, N. Pseudorandom sequence generator based on the Chen chaotic system. *Comput. Phys. Commun.* **2013**, *184*, 765–768. [CrossRef]

17. Wang, L.; Dong, T.; Ge, M.-F. Finite-time synchronization of memristor chaotic systems and its application in image encryption. *Appl. Math. Comput.* **2019**, *347*, 293–305. [CrossRef]

18. 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]

19. Michail, H.E.; Athanasiou, G.S.; Theodoridis, G.; Gregoriades, A.; Goutis, C.E. Design and implementation of totally-self checking SHA-1 and SHA-256 hash functions’ architectures. *Microprocess. Microsyst.* **2016**, *45*, 227–240. [CrossRef]

20. James, J.; Karthika, R.; Nandakumar, R. Design & Characterization of SHA 3- 256 Bit IP Core. *Procedia Technol.* **2016**, *24*, 918–924.

21. Ahmad, I.; Shoba Das, A. Hardware implementation analysis of SHA-256 and SHA-512 algorithms on FPGAs. *Comput. Electr. Eng.* **2005**, *31*, 345–360. [CrossRef]

22. Xu, S.-J.; Chen, X.-B.; Zhang, R.; Yang, Y.-X.; Guo, Y.-C. An improved chaotic cryptosystem based on circular bit shift and XOR operations. *Phys. Lett. A* **2012**, *376*, 1003–1010. [CrossRef]

23. Pareek, N.K.; Patidar, V. Medical image protection using genetic algorithm operations. *Soft Comput.* **2016**, *20*, 763–772. [CrossRef]

24. Hua, Z.; Yi, S.; Zhou, Y. Medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process.* **2018**, *144*, 134–144. [CrossRef]

25. 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]

26. Ma, S.; Zhang, Y.; Yang, Z.; Hu, J.; Lei, X. A New Plaintext-Related Image Encryption Scheme Based on Chaotic Sequence. *IEEE Access* **2019**, *7*, 30344–30360. [CrossRef]

27. 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]

28. Chen, H.; Tanougast, C.; Liu, Z.; Hao, B. Securing color image by using hyperchaotic system in gyrator transform domains. *Opt. Quantum Electron.* **2016**, *48*, 396. [CrossRef]