IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 10, 2021, accepted March 31, 2021, date of publication April 7, 2021, date of current version April 23, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3071535*

# Noise-Resistant Image Encryption Scheme for Medical Images in the Chaos and Wavelet Domain

**ARSLAN SHAFIQUE[1], JAMEEL AHMED[1], (Member, IEEE), MUJEEB UR REHMAN[1], AND MOHAMMAD MAZYAD HAZZAZI[2]**

[1]Department of Electrical Engineering, Riphah International University, Islamabad 46000, Pakistan
[2]Department of Mathematics, College of Science, King Khalid University, Abha 61413, Saudi Arabia

Corresponding author: Arslan Shafique (arslan.shafique@riphah.edu.pk)

This work was supported by the Deanship of Scientific Research at King Khalid University through the Research Groups Program under Grant R. G. P. 1/77/42.

**ABSTRACT** In this paper, a noise-resistant image encryption scheme is proposed. We have used a cubic-logistic map, Discrete Wavelet Transform (DWT), and bit-plane extraction method to encrypt the medical images at the bit-level rather than pixel-level. The proposed work is divided into three sections; In the first and the last section, the image is encrypted in the spatial domain. While the middle section of the proposed algorithm is devoted to the frequency domain encryption in which DWT is incorporated. As the frequency domain encryption section is a sandwich between the two spatial domain encryption sections, we called it a "sandwich encryption." The proposed algorithm is lossless because it can decrypt the exact pixel values of an image. Along with this, we have also gauge the proposed scheme's performance using statistical analysis such as entropy, correlation, and contrast. The entropy values of the cipher images generated from the proposed encryption scheme are more remarkable than 7.99, while correlation values are very close to zero. Furthermore, the number of pixel change rate (NPCR) and unified average change intensity (UACI) for the proposed encryption scheme is higher than 99.4% and 33, respectively. We have also tested the proposed algorithm by performing attacks such as cropping and noise attacks on enciphered images, and we found that the proposed algorithm can decrypt the plaintext image with little loss of information, but the content of the original image is visible.

**INDEX TERMS** Discrete wavelet transform (DWT), chaotic map, medical images, bit-plane decomposition, security analysis of medical images.

## I. INTRODUCTION

In In the past few decades, information security has become a very demanding research area. With the fascinating evolution in digital information security such as image, video and audio, secure transmission over an insecure channel (Internet) is not convenient [1], [2]. Compare with the textual information; images have more redundancy, the bulk of data and high correlation in which traditional cryptosystems such as Advanced Encryption Standard (AES), Data Encryption Standard (DES) [3], [4] are not suitable. Although these algorithms are very secure, the computational time is high and not suitable for real-time applications due to the number of rounds. In the present era, the secure transmission of medical images and ensuring the integrity, authenticity, availability, and confidentiality are very influential. Therefore, most of the researchers are trying to develop an appropriate image encryption scheme with the extensive variety of non-linear systems and transforms such as Fourier Transform [5], [6], Discrete Wavelet Transform [7], SCAN [8], compressive sensing [9] and chaos [10], [11]. Among all these approaches, it is observed that the chaos-based image encryption schemes are the significantly secure encryption algorithms due to its tremendous properties such as the complexity of structures, highly sensitive to initial condition and generating of highly random sequences [12].

Chaotic systems are often used to generate random key streams by selecting appropriate initial conditions and state variables, which are also known as key parameters or seeds and then encrypted images can be produced by employing the generated key streams [13]. A slight change in the seed values may cause completely different key streams and thus a significant difference can occur in the output image (encrypted image). Nowadays, chaotic properties have

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Hui Yeh.

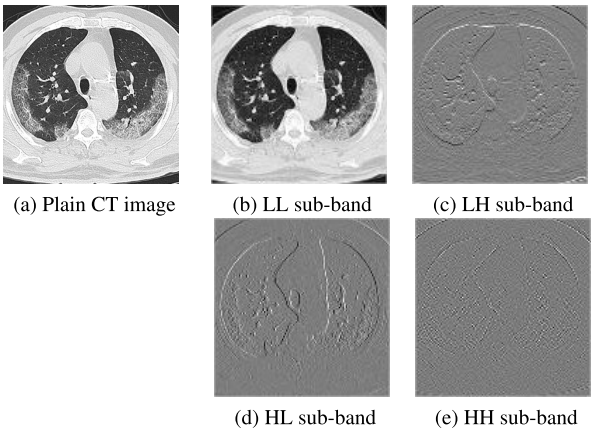



**FIGURE 1.** $1^{st}$ level decomposition of plain image using DWT.

drawn the attention of many researchers in chaos-based image encryption [14]–[20]. In most cases, confusion-diffusion based schemes are considered a secure encryption algorithm and the idea was given by Shannon [21]. In confusion-diffusion based encryption schemes, confusion is associated with the pixel scrambling in which permutation can be done in several ways such as direct pixel scrambling [22], row scrambling and column scrambling [23]. While the diffusion process alters the pixel values through some transformation methods or by logical methods [24]. In an image, the pixels are the biggest parameter and it can be composed of a different number of bits. For instance, in an eight-bit image, there are 8-bits in each pixel.

Any encryption scheme can be considered more secure if it encrypts the digital information at the smallest level [25]. Bit-level encryption may produce better-encrypted images as it encrypts the image at bit-level, which is the smallest element in an image. The bit-level encryption, chaos-based and confusion-diffusion based encryption are fallen into one of the broad categories of encryption called spatial domain encryption [26]. In spatial domain encryption, pixel values manipulates directly via different mathematical operations such as permutation [27], substitution [28], [29], cyclic shift operation [30] and by mean of other logical operations [31]. In contrast to the spatial domain encryption, frequency domain encryption is also taken into account. One cannot directly manipulate the pixels before converting the real pixel numbers into different frequencies during the frequency domain encryption. For example, discrete wavelet transform converts the plain image into four different frequency bands such as LL sub-band, LH sub-band, HL sub-band and HH sub-band. Figure 1 and Figure 2 show the $1^{st}$ and $3^{rd}$ level decomposition of plain CT image using DWT.

In the wavelet decomposition, one thing is noticeable: in every level of decomposition using DWT, the sub-bands image's size becomes half of the previous sub-band. For example, if anyone takes DWT of an image of size M × N, then after the $1^{st}$ level decomposition, the size of the sub-bands (LL, LH, HL and HH) become $(\frac{M}{2} \times \frac{N}{2})$. Similarly, the size of the sub-bands at $2^{nd}$ level decomposition decreases

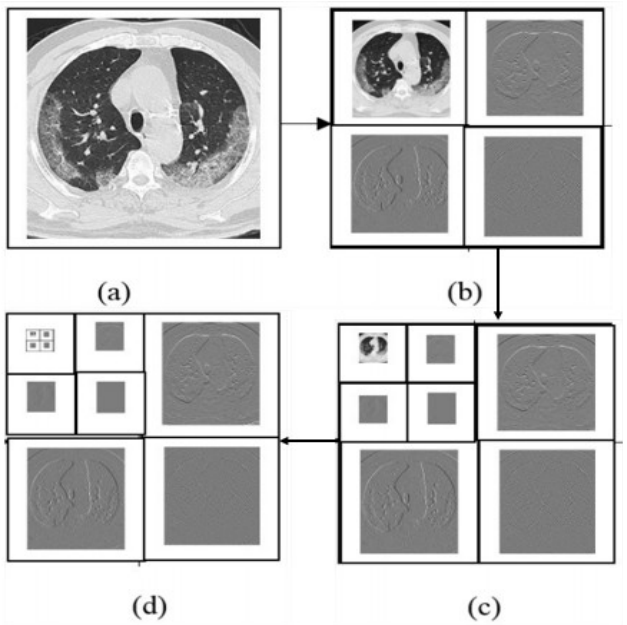



**FIGURE 2.** Decomposition at $3^{rd}$ Level using DWT (a) Plain CT image (b) $1^{st}$ Level decomposition (c) $2^{nd}$ Level decomposition (d) $3^{rd}$ Level decomposition.

accordingly. All the sub-bands have a different kind of frequency components. LL sub-band consists of low frequencies, which means most of the information of the plain image present in the LL sub-band while very little information lies in the other three sub-bands (LH, HL, HH). Because these sub-bands contain high-frequency components, the LL and LH sub-bands are so powerful, if we ignore the HL and HH sub-bands while taking the IDWT of LL and LH sub-bands, the original image can still recover with little loss of information, but perceptually there will be no difference between the plain and the recovered image.

### A. CONTRIBUTIONS OF THIS WORK

As most of the part of the world is suffering from security issues. So, keeping in mind the security issues, we have made an effort to secure the digital data from adversaries. The contributions of this work are as under:

- In this paper, we have examined the past encryption algorithms and pointed out some security issues, which are presented in section II. To provide security to the medical images, we have developed a cryptosystem that can generate three different RGB cipher images corresponding to one plain medical image. To retrieve the information of the original image, the receiver must have all three cipher images.
  In the proposed encryption scheme, we have deployed spatial and frequency domain encryption in which the frequency-domain encryption section lies in between the other two sections of the spatial domain encryption; thus, we called it sandwich encryption. While designing the proposed encryption algorithm, encryption computational time, and security level, both these parameters are considered.

- To reduce the computational time, we have considered only a low-frequency band (LL sub-band), which is extracted by taking Discrete Wavelet Transform (DWT) of the plain image. The reason for considering only the LL sub-band is that most of the plain image information present in the LL sub-band. So it is more important to secure the low-frequency sub-bands than the high-frequency sub-bands, which may cause an increase in the encryption computational time by applying some complex mathematical operations sub-bands (LH, HL and HH).
- We have figure out the proposed encryption scheme by using statistical analysis such as entropy, correlation, energy, PSNR, MSE to show the effectiveness of the proposed work. The proposed algorithm is also tested against different attacks such as cropping attack, noise attack and brute force to prove that our encryption scheme is robust against different attacks.

## II. RELATED WORK AND PRELIMINARIES

In recent years, a bulk of security work on the security of the digital data has been successfully carried out by different researchers [32]–[38], But most of the existing work still has some issues which need to be improved remarkably. This section provides the previous work related to digital data security and some drawbacks of the existing encryption schemes.

In [39], Wang *et al.* put his effort to achieve the confusion and diffusion simultaneously to reduce encryption computational time. Although the authors were successful in reducing the processing time, but the security was compromised a bit. In [40], Zhou *et al.* presented his work to secure the digital images using complex mathematical functions of chaos. A new chaotic map by combining different chaotic structures was also proposed and then used in his proposed encryption scheme. The encryption scheme is consists of 5 rounds rather than single encryption round to enhance the security. In real-time applications, we required low processing time. There is always a tradeoff between security and encryption computational time. Obviously, as we increase the number of rounds, the security of the encryption scheme enhances. Nevertheless, with the enhancement of security by increases the number of rounds, we must compromise on the processing time, which is not suitable for real-time applications. In contrast to [40], Shafique *et al.* Proposed a bit-level encryption scheme (IEC-BPMC) [41], in which the author(s) applied the permutation function only on the most significant bit-planes to reduce the computational time. The reason is that most of the information is present in the MSBs bit-planes. The information content gradually decreases as we go from the MSBs to the LSBs bit-planes. Although the author(s) has designed the encryption schemes for real-time applications, security was compromised and it was a break in [42]. In [42], Wen *et al.* has done cryptanalysis on the encryption scheme which was proposed in [41]. The author used a chosen-plaintext attack methodology to break the IEC-BPMC. A plain image can be recovered in the bit-plane encryption methods by combining only four MSBs bit-planes with little information loss. By adopting this methodology, one can achieve high security with less processing time, but that encryption scheme would be lossy, which is not suitable for those applications where we required the exact recovery of information. In 2014, Wang *et al.* [43] claimed that the encryption scheme proposed in [44] is insecure against chosen-plaintext attack. To remove the flaws of the encryption scheme proposed in [44], Zhang and Xiao [45] proposed a cryptosystem which is based on bit-level permutation and diffusion only mechanism. According to Shannon's theory, any cryptosystem becomes stronger if it contains a confusion-diffusion mechanism. While the encryption scheme proposed in [45] was based only on the diffusion mechanism, and it may consider an insecure scheme because this scheme does not satisfy Shannon's criteria. In [46], the author(s) has presented an encryption scheme which also does not satisfy the theory presented in [21]. In [47] Xu et. al proposed a novel bit-level permutation encryption in which the author(s) utilized "permutation-diffusion" mechanism. Later on, In [48], a bit-level and pixel-level substitution-based encryption scheme was proposed. Pixel level substitution may be strong in some cases i.e. multiple substitution box(S-box) encryption [49]. But in the case of bit-level substitution, it cannot be effective because bit-planes consist of only two values 0 and 1. The problems of the pixel substitution are addressed in [49]. In [49], Anees *et al.* highlighted that the single S-box encryption is not suitable for those images which contain higher as well as a lower number of gray levels. The author(s) presented an encryption scheme to solve these issues by addressing the single S-box encryption problem. In his proposed work, multiple S-boxes were used to improve the security of the encryption algorithm. Apart from using multiple S-boxes, the strength of the S-box plays a vital role in substitution based cryptosystems. In [50], Shafique *et al.* proposed a new methodology to construct a robust S-box to enhance the substitution based encryption schemes. The substitution based encryption has gained much attention from the researcher. The substitution-based encryption schemes are well improved nowadays, but there is a major flaw of high processing time.

Apart from symmetric-key cryptosystems, asymmetric algorithms also have their importance. In [51], a public key was used to create random sequences which utilizes elliptic curve cryptography (ECC). The proposed scheme through ECC is a very time-consuming cryptosystem because of several numbers of rounds. It can be a time-efficient encryption scheme by deploying some transformation techniques with ECC rather than using more than one number of encryption rounds. In [52], a homomorphic cryptosystem is used to build a new encryption scheme. For better security, the author(s) has used cloud storage for the sake of data storage. Although the proposed algorithm's security was quite better, it does not provide high computational speed. For further improving the security and reducing the computational time, In [53],

the homomorphic cryptosystem is incorporated with Ant lion Optimization (ALO) encryption. In [54], an encryption scheme was proposed based on the Goldreich Goldwasser Hallevi (GGH) algorithm. The only flaw in this scheme was, it cannot resist the chosen-Cipher attack. In [55], a genetic algorithm (GA) was incorporated to generate a different number of cipher images. In the GA algorithm, there were some interpretation problems of function and selection of ciphering. In [56], an encryption scheme was proposed based on a chaotic map. Different chaotic structures like a logistic map, gauss iterated and sine map has some mapping drawbacks.

### A. CUBIC-LOGISTIC MAP (C-LM)

C-LM is a one-dimensional modified version of a chaotic logistic map and can be used for various purposes such as random number generation, key image generation and keystream generation. We have used C-LM in our proposed encryption algorithm. Using C-LM over logistic map [57] is that C-LM is the modified form of a logistic map and gives better results than the chaotic logistic map. The C-LM is given as: [58]:

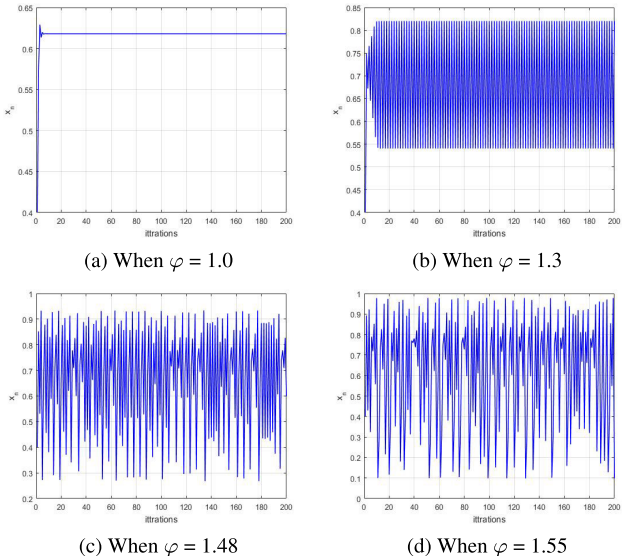
$$x_{n+1} = \varphi x_n(1 - x_n)(2 + x_n) \tag{1}$$

Different chaotic maps have different chaotic ranges. While in some intervals, the chaotic system generates constant values. Those intervals in which the same number of values generates are always avoidable. So it is essential to choose the right interval for any chaotic system. Right interval means in which the system shows random behavior. For equation 1, there are two seed values $x_0$ and $\varphi$. These initial values should lie in the following intervals:

$$x_0 \in (01)$$
$$\varphi \in [1.421.60)$$

The intervals mentioned above are those in which the system 1 comes into the chaotic state and can generate the values ranges (0 1]. These initial values, which do not lie in the given intervals, are not suitable for generating random sequences. We will see these effects one by one by taking different initial values from the intervals and from out of the intervals.

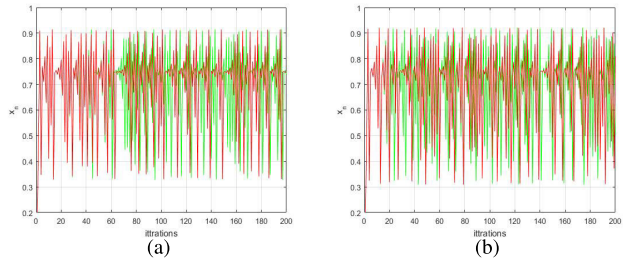
- The C-LM is highly dependent on the parameter $\varphi$. By changing the value of $\varphi$, the system shows different behavior. First, we take two different initial values of $\varphi$ which are from out of the given chaotic interval and plotted the generated sequences one by one as shown in Figure 3(a) and (b). Figure 3(a) shows that the C-LM initially shows some randomness, but it becomes steady after some iterations and generates all constant values. While in Figure 3(b) shows that the generated values are repeating after a short period which means the selected values are not suitable for the chaotic system, which is given in equation 1.
- After that, we have chosen two more different values of $\varphi$, which lies in the given interval. From Figure 3(c) and (d), it can be seen that the CLM is now able to generate more random values.




(a) When $\varphi = 1.0$ (b) When $\varphi = 1.3$

(c) When $\varphi = 1.48$ (d) When $\varphi = 1.55$

**FIGURE 3.** (a-d) Graph of different chaotic sequences for different values of $\varphi$.




(a) (b)

**FIGURE 4.** Graph of sensitivity of initial conditions: (a) Green signal is generated when $\varphi = 1.450000$ and red signal is generated when $\varphi = 1.450001$ while $x_0$ is constant for both the signals i.e $x_0 = 0.2$ (b) Green signal is generated when $x_0 = 0.2000$ and red signal is generated when $x = 0.2001$ while $\varphi$ is constant for both the signals i.e $\varphi = 0.1.46$.

The initial values should also be sensitive. Sensitivity to initial values means that if we slightly change these values, there must be a significant change in the output stream. We have chosen slightly different values of $\varphi$ such that $\varphi = 1.450000$ and $\varphi = 1.450001$ and by keeping the values of $x_0$ constant such that $x_0 = 0.2$, we have plotted both the signals for $\varphi = 1.450000$ and $\varphi = 1.450001$. Figure 4(a) shows that, after minor changes in the value of $\varphi$, the same values are generated until the $23^{rd}$ iteration. After that, they separate till the $200th$.

Similarly, we have taken slightly different values of $x_0$, such that $x_0 = 0.2000$ and $x_0 = 0.2001$. while $\varphi$ is constant, i.e. $\varphi = 1.46$, both the generated signals corresponding to the selected values of $x_0$ shows the different behavior as it can be seen in Figure 4(b), which means that C-LM is highly sensitive to both the initial conditions.

The interval in which the system shows the chaotic behavior can also be analyzed by the bifurcation diagram. The bifurcation diagram of C-LM is shown in Figure 5. The

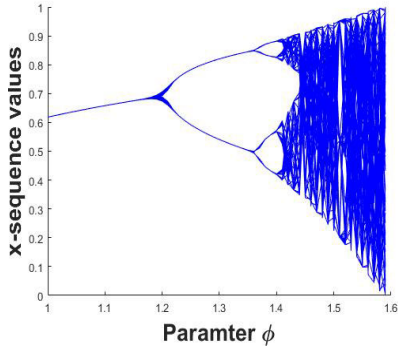



**FIGURE 5.** Bifurcation diagram of C-LM.

parameter $\varphi$ is on the horizontal axis, while the values generated corresponding to each value of $\varphi$ is mentioned on the vertical axis. When the value of $\varphi$ is from 1 to 1.2, the generated stream is constant in which the one value is repeating again and again. While increasing the value of $\varphi$ from 1.2 to 1.3, two possible values are generated, which are repeating. After $\varphi = 1.38$, the system bifurcates into four different trajectories. It means that by setting the values, $\varphi = 1.38$, the system can generate only four different repeating values. At $\varphi = 1.4$, the system generates eight different repeated values. This kind of behavior shows that the number of different generated values increases by a factor of two. By increasing the value of $\varphi$, the system enters into the chaotic region in which the system can generate so many different random values.

### B. BIT-PLANE EXTRACTION

In the process of bit-plane extractions, the plain image divides into eight bit-planes in which every bit-plane consists of two different values i.e., 0 and 1. The percentage of information in each bit-plane is different. Among all the eight bit-planes, the MSB bit-plane ($8^{th}$ bit-plane) occupies the highest information while the least information present in the LSB bit-plane ($1^{st}$ bit-plane). The information of the plain image gradually decreases from the last MSB bit-plane to the first LSB bit-plane. This effect can be seen in Figure 6.

A 256 gray levels plain image is chosen which can be represented as 2.

$$\begin{aligned} I_{a,b} = {} & I^n_{a,b} \times 2^6 + I^n_{a,b} \times 2^5 + I^n_{a,b} \times 2^4 \\ & + I^n_{a,b} \times 2^3 + I^n_{a,b} \times 2^2 + I^n_{a,b} \times 2^n \\ & + I^1_{a,b} \times 2^n + I^0_{a,b} \times 2^n \end{aligned} \tag{2}$$

or we can simply write in a close form as given in 3:

$$I_{a,b} = \sum_{n=0}^{7} I^n_{a,b} \times 2^n \tag{3}$$

where; $n = 0, 1, 2 \ldots. 7$. Percentage information can be calculated by equation 4 [59] and the statistics of information division in each plane is given in Table 1.

$$I_p = \frac{2^{p-1}}{\sum_{p=0}^{7} 2^{p-1}} \tag{4}$$

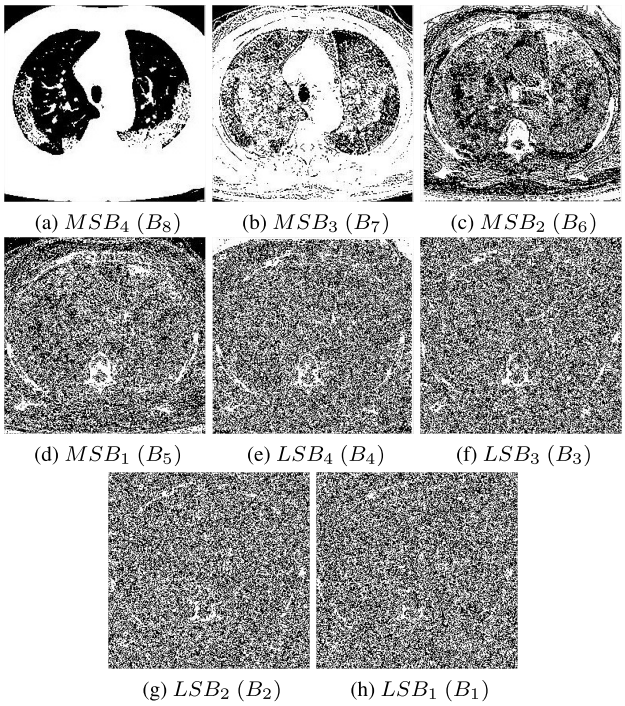



**FIGURE 6.** Bit-plane decomposition of CT plain image(a-d) 4 MSBs bit-planes (e-h) 4 LSBs bit-planes.

**TABLE 1.** Information division in each bit-plane.

| Bit-plane position $(B_i)$, [i = 0, 1, ...7] | percentage information in each bit-plane |
|---|---|
| 0 | 0.3000 |
| 1 | 0.7900 |
| 2 | 1.4200 |
| 3 | 3.1200 |
| 4 | 6.2500 |
| 5 | 12.2300 |
| 6 | 25.7000 |
| 7 | 50.2000 |

From Table 1, it can be seen that the MSB bit-plane ($8^{th}$ bit-plane) contained more than 50% of the information of the plain image, which is the highest information occupied bit-plane among all other extracted bit-planes. The least information present in the LSB is bit-plane ($1^{st}$ bit-plane), which is less than 1%. For the extraction of the bit-planes, equation 5 can be considered:

$$I_n = mod(floor(\frac{I_{a,b}}{2^n}), 2) \tag{5}$$

Again $n = 0, 1,2 \ldots 7$. The elements of all the bit-planes $I_n$ lies in [0, 1]. Here, one thing can be noted that if we extract the bit-planes by using equation 5, equation 2 can be used to composition the plain image from the eight extracted bit-planes.

## III. OVERVIEW OF THE PROPOSED CRYPTOSYSTEM

The proposed encryption scheme is composed of three major sections, as shown in Figure 7 and the generalized block diagram of the proposed algorithm is shown in Figure 8.

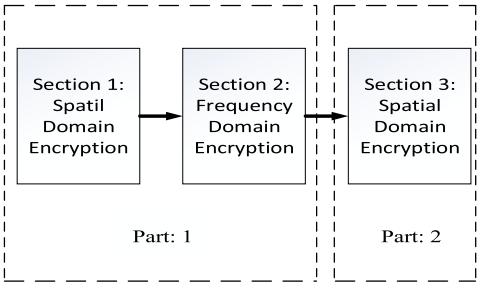



**FIGURE 7.** Three sections of the proposed algorithm.

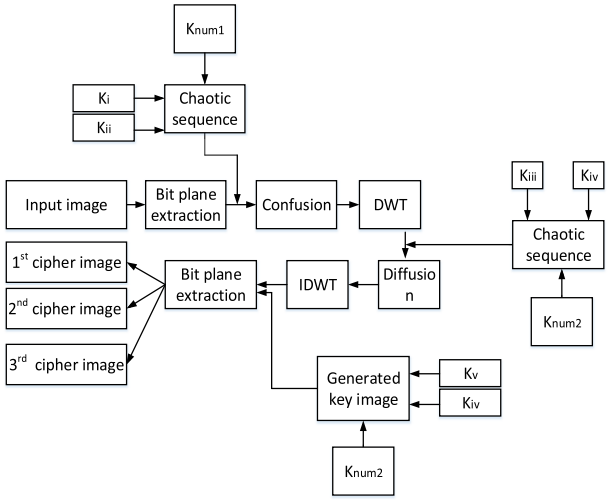



**FIGURE 8.** Generalized block diagram of the proposed encryption algorithm.

The first section of the proposed scheme is devoted to spatial domain encryption in which we directly manipulate the pixel values by extracting the bit-planes from the plain image. We have adopted the bit-pane extraction in our proposed algorithm because, according to [60], image encryption could be more strong if we encrypt the image at the smallest level-element. As the smallest element in the gray-scale images is the bit, when we convert the image into bit-planes. Furthermore, a 1-D random signal is generated using a chaotic structure in which the initial values of C-LM are taken as keys ($K_i$ and $K_{ii}$). Nevertheless, here we have included one extra key (a key number $K_{num1}$), which amplifies the values generated after iterating the chaotic map *mn* times. This 1-D random generated key-stream is used for the permutation of only MSB bit-planes. In comparison, the other LSB bit-planes proceed as it is in the next section of the proposed cryptosystem. The reason is that the LSB bit-planes contain less than 6 % information of the plaintext image, as shown in Table 1. So there is no need to do several many mathematical operations on the LSB bit-planes. It may result in increasing the processing time. However, we have manipulated the LSB bit-planes in the next section of the proposed encryption algorithm in which the frequency domain encryption is done by using a Discrete Wavelet Transform (DWT). By taking the DWT of the modified and other unchanged bit-planes ($BP_4$, $BP_3$, $BP_2$, $BP_1$), bit-plane elements are converted into frequency components.

DWT transforms any 2-D signal into four sub-bands (LL, LH, HL and HH). As the LL sub-band consists of low frequencies, most of the information present in the LL sub-band can be seen in Figure 1. Due to the maximum information present in the LL sub-band, we have only manipulated the frequencies of such sub-bands with a 2-D signal generated using C-LM with different key parameters. This 2-D signal is then used to perform the logical operation on all the LL sub-bands one by one.

After manipulating LL sub-bands' frequencies, we take Inverse Discrete Wavelet Transform (IDWT) to convert the frequency components into real pixel values.

The two sections, as mentioned above of the proposed encryption scheme are shown in Figure 9; which is the first part of the proposed algorithm. While the second part of the algorithm is shown in Figure 10, which is the third section of the proposed encryption scheme, and it is dedicated to the spatial domain encryption.

After the frequency domain encryption part, we have eight different images known as pre-cipher images. Now again, extract the eight bit-planes from each pre-cipher image. It will give us 64 bit-planes in which 32 are the MSB bit-planes and the other 32 are the LSB bit-planes. Now make two groups of the first 48 bit-planes in which each group will have 24 bit-planes. Combine $1^{st}$, $9^{th}$ and $17^{th}$ bit-planes ($A_8$, $B_8$ and $C_8$) to create the $8^{th}$ pre-RGB cipher image as shown in Figure 10. Now combine $2^{nd}$, $10^{th}$ and $18^{th}$ bit-planes ($A_7$, $B_7$ and $C_7$) to create the $7^{th}$ Pre-RGB cipher image and so on. Form the first group of the bit-planes, we get eight pre-RGB cipher images which will then combine using equation (1) to produce $1^{st}$ RGB cipher image. The combining process of respective bit-planes is illustrated in Figure 10. Similarly, generate $2^{nd}$ RGB cipher image from the $2^{nd}$ group of the bit-planes. The remaining 16 bit-planes are placed in the $3^{rd}$ group in which eight more bit-planes are included, which are extracted from the cover image that is generated by using a chaotic structure with different key values and generate $3^{rd}$ RGB cipher image in the same way as we generated $1^{st}$ and $2^{nd}$ RGB cipher image.

### A. PROPOSED ENCRYPTION SCHEME

All three sections of the proposed scheme are explained in detail in this section. The sequence of the mathematical and the logical operations to encrypt the plain image using the proposed encryption algorithm are as follows:

**Section 1: Spatial domain encryption**

- Choose an 8-bit grayscale CT image of size MN i.e $I_{a,b}$(M×N. where M and N are the number of rows and columns in the input image, respectively.
- Extract 8 bit-planes from the input image. Each element in the bit-plane will be in the range [0, 1]. The extracted bit-planes are given as:

$$I'^{7}_{a,b} \times 2^7, I'^{6}_{a,b} \times 2^6, I'^{5}_{a,b} \times 2^5, I'^{4}_{a,b} \times 2^4,$$
$$I'^{3}_{a,b} \times 2^3, I'^{2}_{a,b} \times 2^2, I'^{1}_{a,b} \times 2^1, I'^{0}_{a,b} \times 2^0,$$

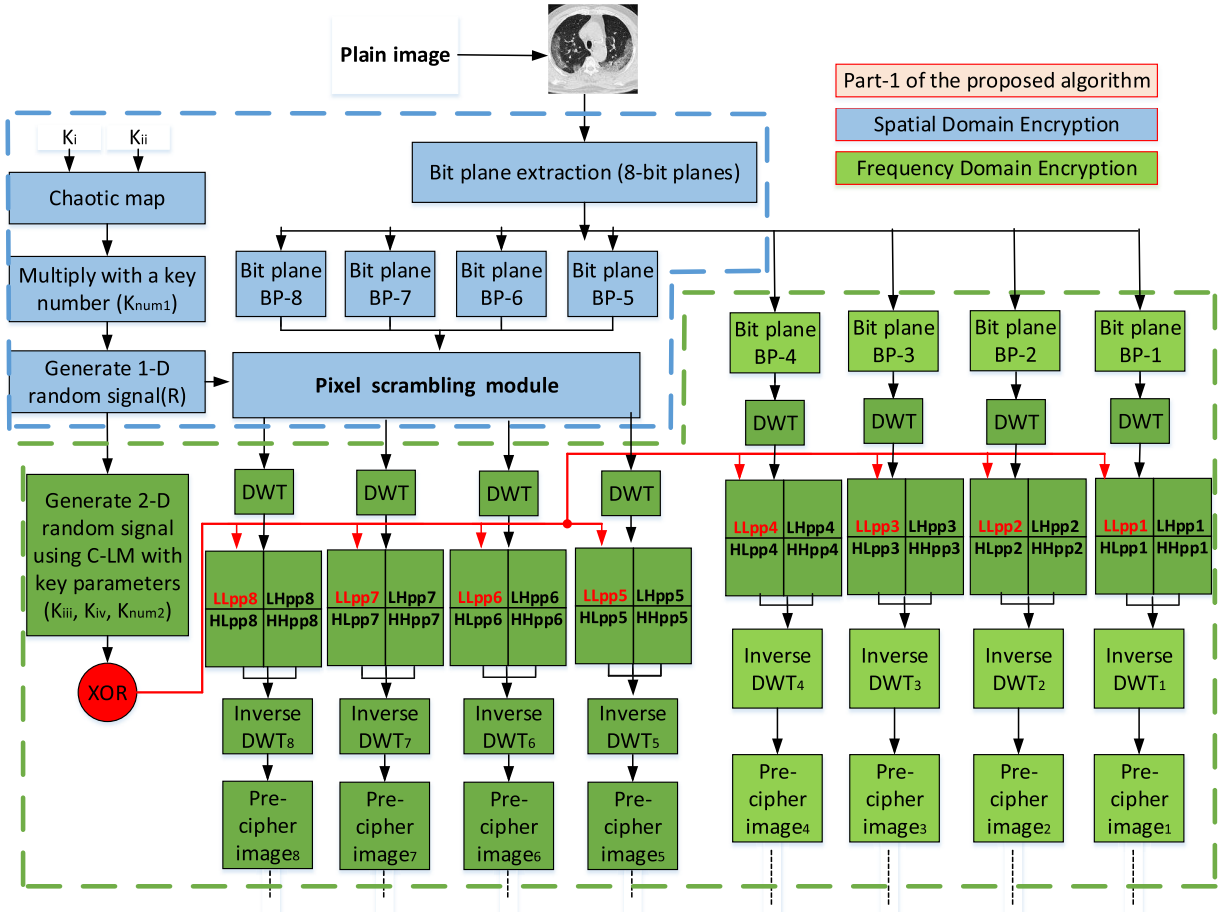



**FIGURE 9.** Block diagram of the first two sections of the proposed algorithm, (Part-1): Spatial and frequency domain encryption.

In general, the bit planes can be expressed as:

$$I'^{n}_{a,b} \times 2^{n} \rightarrow \{\mathrm{n} \mid \mathrm{n} \in W \wedge 0 \leq \mathrm{n} \leq 7\}$$

Where: $I'_{a,b}$ and $n$ represents the bit-pane and its position respectively. While $2^n$ represents that how much information is present in the specific bit-plane. $2^n$ can be treated as an information control factor. For instance, from Figure 6(a), which can be represented as: : $I'^{7}_{a,b} \times 2^7$, it can be noted that due to the highest information control factor ($2^7$), most of the information of the plain image present in that bit-plane ($B_8$). As we decrease the value of the information control factor, the content of the input image decreases accordingly in the bit-planes. Due to the lower information present in the LSB bit-planes, we have scrambled only MSB bit-planes to reduce the computational time. A certain portion of one of the bit-plane (B) is shown given below:

$$B = \begin{bmatrix} 1&0&0&0&1&0&0&0&0 \\ 1&1&0&0&0&0&1&0&1 \\ 0&1&0&0&0&0&1&1&1 \\ 0&1&0&0&0&0&0&1&1 \\ 0&1&1&1&1&0&0&0&0 \\ 0&1&1&1&0&0&0&0&1 \\ 1&1&1&1&1&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0 \\ 1&0&0&0&1&0&1&0&0 \end{bmatrix}$$

- Generate the random sequence $R$ using the C-LM by selecting the suitable keys ($K_i$ and $K_{ii}$). The length of the generated will be $MN$. i.e R = $r_1$, $r_2$, $r_3$.... $r_{MN}$. All the values of the generated key-streams will be in the range [0 1]. So, to amplify these values, we have introduced another key, which is a key number ($K_{num1}$) which is placed at the appropriate position as given in the equation (5). Now map the generated key-stream as follows:

$$\begin{aligned} key - stream &= uint8(mod(floor((stream) \\ &\quad *K_{num}), MN)); K_{num} \geq 258. \\ \text{key-stream} &\in [0(\mathrm{M} * \mathrm{N}) - 1]; \end{aligned} \tag{6}$$

This stream ($key-stream$) is used as a permutation function by which we have scrambled the bits of only MSB bit-planes. The impact of pixels permutation is shown in Figure 12. 12(a) shows the smooth pattern of the pixels of an image, while Figure 12(b) (which is generated after applying the permutation function on Figure 12(a)) shows the permuted version of Figure 12(a). After completing the scrambling process, the processed pixel values will pass to the frequency domain encryption module.

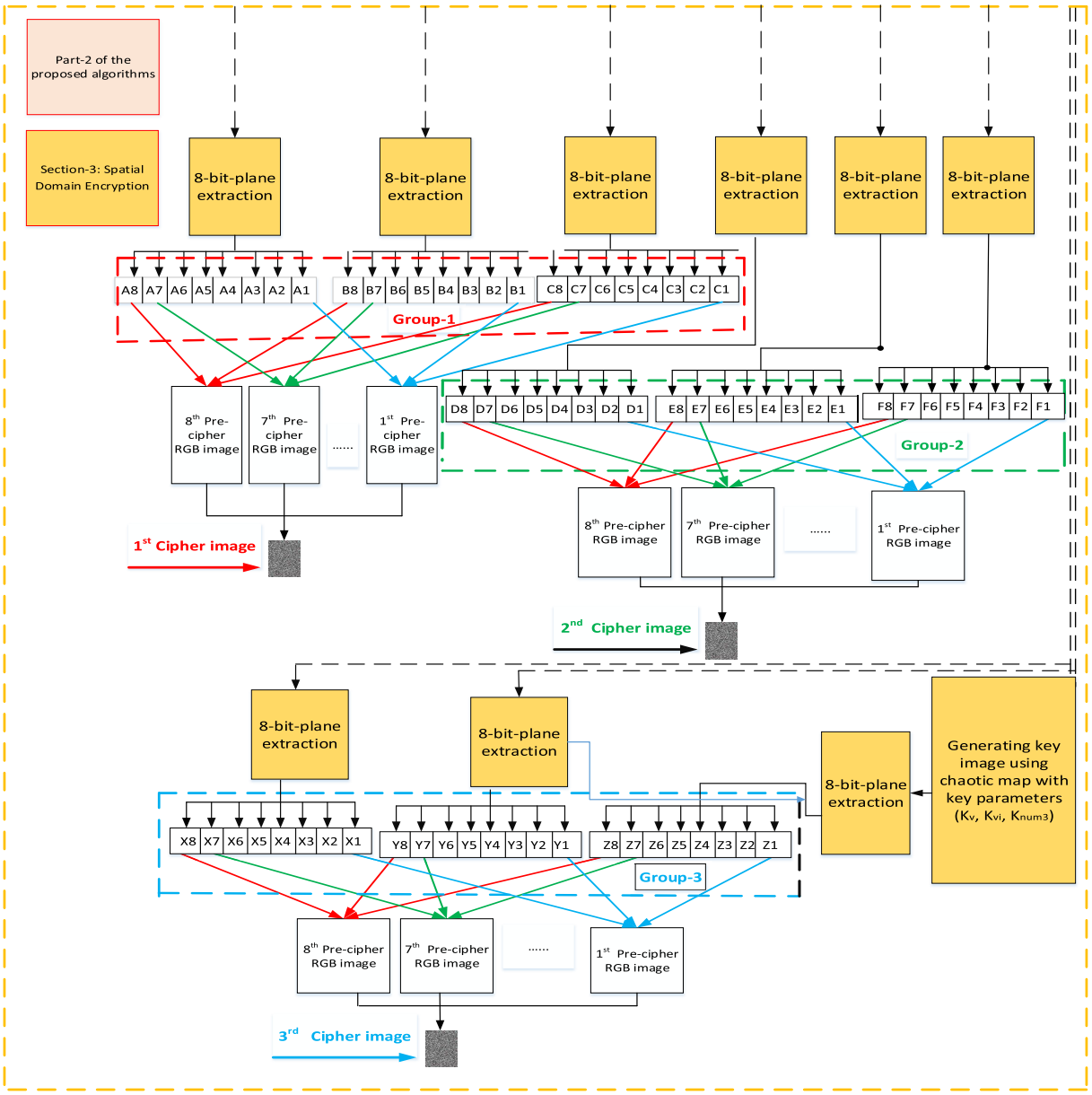
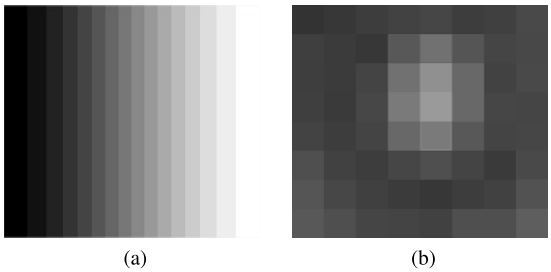



**FIGURE 10.** Block diagram of the 3$^{rd}$ section of the proposed algorithm, (Part-2): Spatial domain encryption.



(a) (b)

**FIGURE 11.** Visualization of impact permutation function: (a) smooth pattern of an image (b) permutation version of image (a).

### Section 2: Frequency domain encryption

- Take the discrete wavelet transform (DWT) of all the scrambled and non-scrambled bit-planes. We will get four frequency bands (LL, LH, HL, HH) corresponding to each scrambled and non-scrambled bit-plane. Which will give us 32 permuted frequency bands in which eight permuted bands are LL sub-bands ($LL_{p1}$, $LL_{p2}$, $LL_{p3}$, ……, $LL_{p8}$) are the highest information occupied sub-bands (low-frequency sub-bands). While the other 24 sub-bands contain the least information (high-frequency sub-bands). The size of each frequency band will be $\frac{M}{2} \times \frac{N}{2}$.
- Generate the mask of size $\frac{M}{2} \times \frac{N}{2}$ by using the C-LM by selecting different seed values ($K_{iii}$, $K_{iv}$, $K_{num2}$). The purpose of using different seed values is to avoid the continuation of identical keys. The mask is then applied only on the low frequency permuted

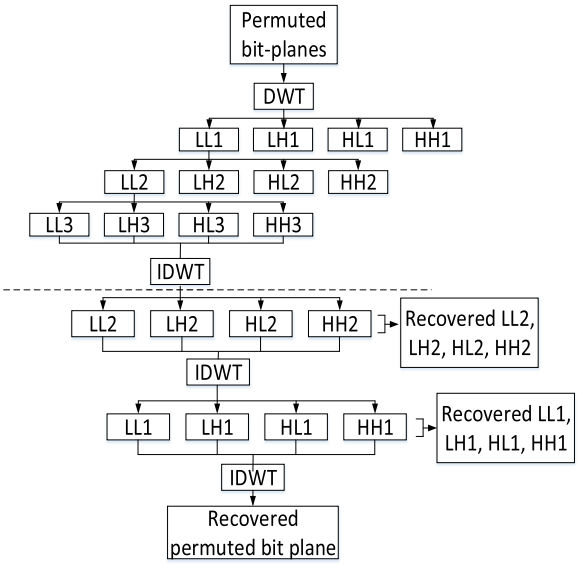



**FIGURE 12.** $3^{rd}$ level DWT and its IDWT: A Tree.

sub-bands($LL_{p1}$, $LL_{p2}$, $LL_{p3}$, ……, $LL_{p8}$ and save as $M_1, M_2, M_3, \ldots . M_8$. Mask is applied by the logical XOR operation, which is given as:

$$key - stream_2 = \text{uint8}(\text{mod}(\text{floor}((stream_2)*K_{num2}), 256))$$
$$\text{Mask} = \text{reshape}\ (key - stream_2, \text{M/2, N/2})$$
$$M_1 = LL_{p1} \oplus \text{Mask}$$
$$M_2 = LL_{p2} \oplus \text{Mask}$$
$$M_3 = LL_{p3} \oplus \text{Mask}$$
$$\vdots$$
$$M_8 = LL_{p8} \oplus \text{Mask}$$

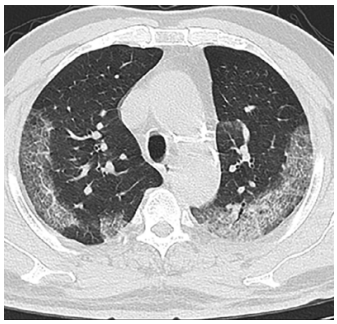
- Now takes the IDWT of each set of the respective four sub-bands. After taking the IDWT, the modified frequencies of the frequency bands are now converted into the real pixel values. Here we will have eight altered images. These values will be passed through the spatial domain encryption module after converting the frequency values into real pixel values.
  A short tree of DWT and IDWT is given in Figure 12 in which $3^{rd}$level decomposition is presented to explain the converting and recovering from spatial to frequency and frequency to the spatial domain, respectively.
  **Section 3: Spatial domain encryption**
- Eight modified images which are the Inverses of DWT are the inputs to the spatial domain section as shown in figure 10. Now extract the eight bit-planes from the eight input images. Here, we will have 64 bit-planes and the size of each bit-plane will be $MN$.
- Generate noisy cover-image by using equation (5), the keys ($K_v$, $K_{vi}$, $K_{num3}$) used to generate the noisy cover-image are different from the keys which are used to generate the $key - stream$.
- Extract 8 bit-planes from the noisy cover-image. The purpose of extracting bit-planes from the noisy cover-image is that we need to make three groups of bit-planes in which each group must contain 24 bit-planes, which are further divided into three sub-groups and each group contain eight bit-planes as shown in Figure 10. Combine the bit-planes, which are at the same positions, to generate eight pre-RGB cipher images. After that, for generating the $1^{st}$ cipher image, use equation 1 to combine eight pre-RGB cipher images. Similarly, we have generated $2^{nd}$ and $3^{rd}$ cipher image. The receiver must have all of these RGB cipher images to reconstruct the original image. The proposed encryption scheme is suitable for gray scale as well as binary images. Two test CT plain images (gray and binary images), their corresponding cipher images and histograms which are generated using the proposed encryption scheme are shown in Figures 13 and 14 respectively. Moreover, we have tested the proposed algorithm by encrypting the plain white and plain black image having a single gray level. From Figures 15 and 16, it can be seen that the proposed encryption algorithm is fully capable of encrypting the single gray level images properly.

### B. DECRYPTION ALGORITHM

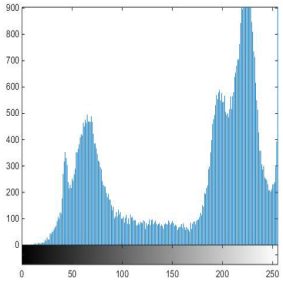
To reconstruct the original image, the receiver must have all the three cipher images generated using the proposed algorithm. To decrypt the plain image, one must follow the encryption steps in the reverse order, which are as under:

- Extract the RGB components from all the three cipher images.
- Now Extract the bit-planes from all the RGB components using equation (4).
- Take DWT to convert the bit-planes into frequency components and perform the XOR operations only on the low-frequency bands with the generated 2-D random signal using the appropriate keys.
- Take IDWT to construct the permuted bit-planes from the frequency sub-bands.
- In the last, perform inverse permutation using the generated 1-D random signal with correct keys to reconstruct the eight bit-planes of the plain image. Now combine the bit-planes using equation 1 to recover the original image. The proposed algorithm is a lossless encryption algorithm because it can recover every single bit from the cipher image. I.e., when we subtract the decrypted image from the plain image, all the entries in the resultant difference matrix ($D_{mat}$) become zero. There is also no perceptual difference between the plain and cipher image as shown in Figure 18(a) and Figure 17(a).

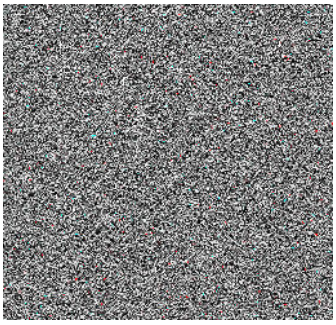
$$D_{mat} = \begin{bmatrix} P_{1,1} = 0 & P_{1,2} = 0 \ldots . P_{1,N} = 0 \\ P_{2,1} = 0. & . \\ P_{3,1} = 0 & . \\ . & . \\ . & . \\ P_{M-1,1} = 0 & P_{M-1,2} = 0 \ldots . P_{M-1,N} = 0 \\ P_{M,1} = 0 & P_{M,2} = 0 \ldots . P_{M,N} = 0 \end{bmatrix}$$



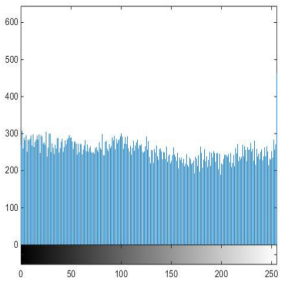
(a) plain CT image



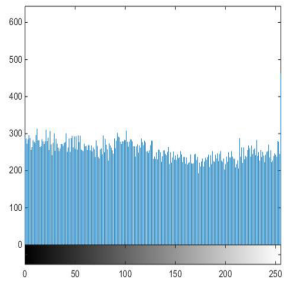
(b) Histogram of plain image



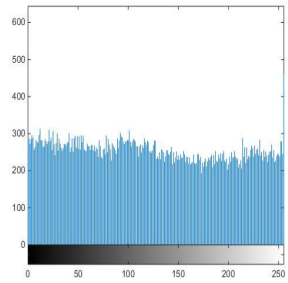
(c) $1^{st}$ cipher image



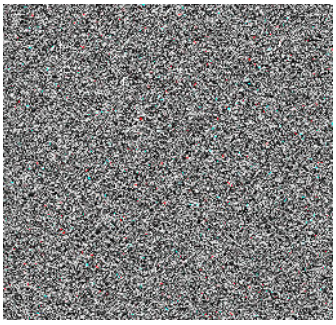
(d) Histogram of R component of $1^{st}$ cipher image



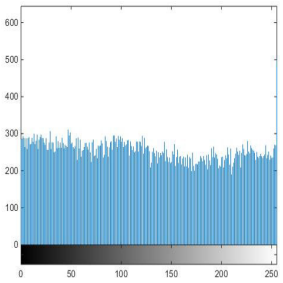
(e) Histogram of G component of $1^{st}$ cipher image



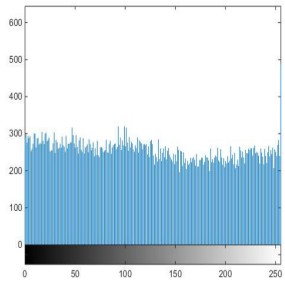
(f) Histogram of B component of $1^{st}$ cipher image



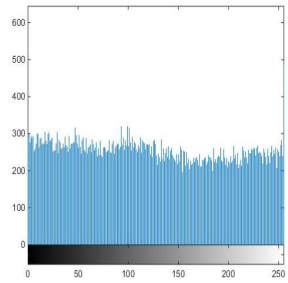
(g) $2^{nd}$ cipher image



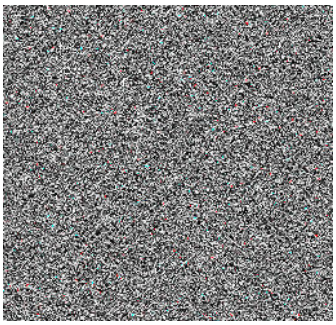
(h) Histogram of R component of $2^{nd}$ cipher image



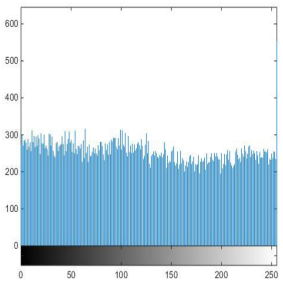
(i) Histogram of G component of $2^{nd}$ cipher image



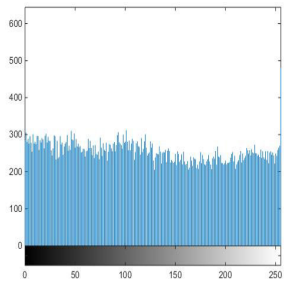
(j) Histogram of B component of $2^{nd}$ cipher image



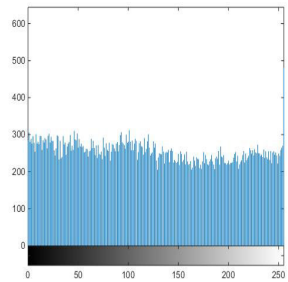
(k) $3^{rd}$ cipher image



(l) Histogram of R component of $3^{rd}$ cipher image



(m) Histogram of G component of $3^{rd}$ cipher image

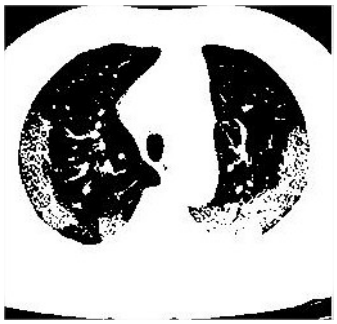


(n) Histogram of B component of $3^{rd}$ cipher image

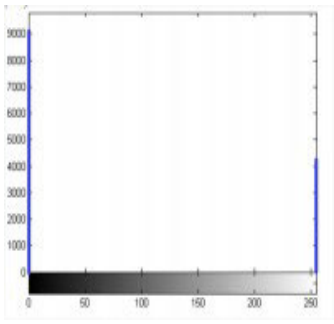
**FIGURE 13.** **A 256 gray level CT image, Cipher image and their corresponding Histograms.**

## IV. PERFORMANCE AND STATISTICAL SECURITY ANALYSIS

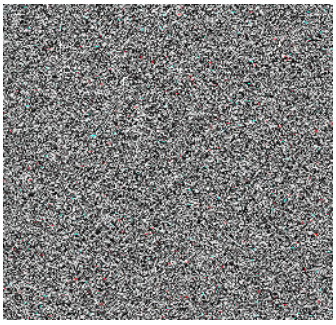
To evaluate the performance and the security of the encryption algorithm, several security analysis are present in the literature [22], [61]–[63]. Different kinds of experiments and analyses such as keys sensitivity analysis, key-space analysis, entropy, correlation, energy, contrast, histogram analysis, lossless analysis, cropping attack analysis, UACI and NPCR



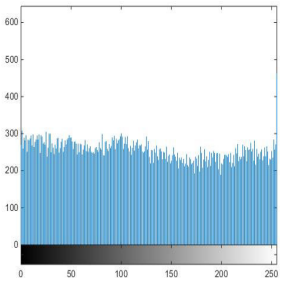
(a) plain CT image



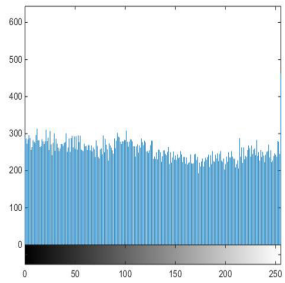
(b) Histogram of plain image



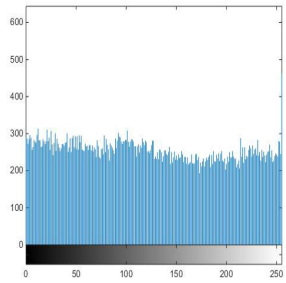
(c) $1^{st}$ cipher image



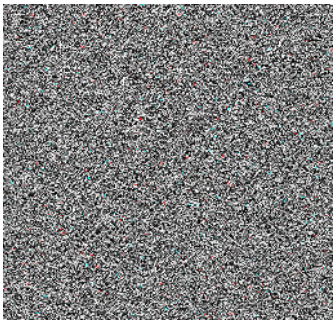
(d) Histogram of R component of $1^{st}$ cipher image



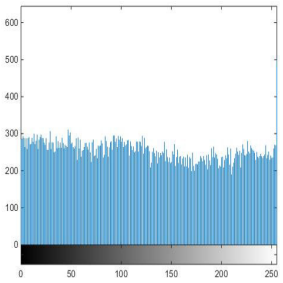
(e) Histogram of G component of $1^{st}$ cipher image



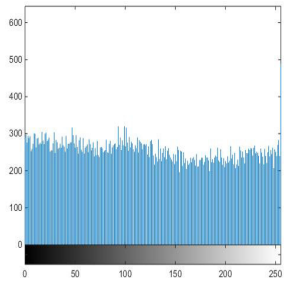
(f) Histogram of B component of $1^{st}$ cipher image



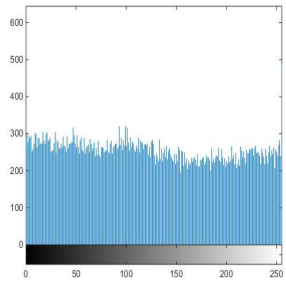
(g) $2^{nd}$ cipher image



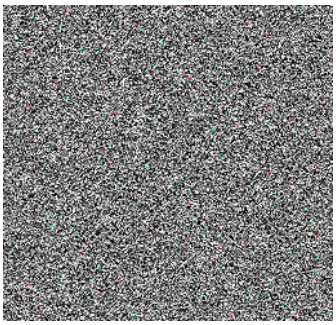
(h) Histogram of R component of $2^{nd}$ cipher image



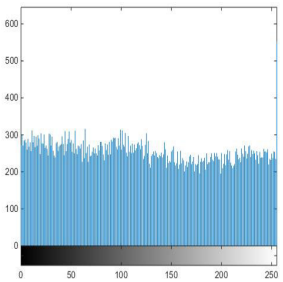
(i) Histogram of G component of $2^{nd}$ cipher image



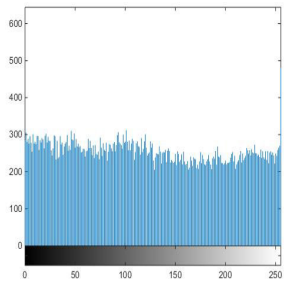
(j) Histogram of B component of $2^{nd}$ cipher image



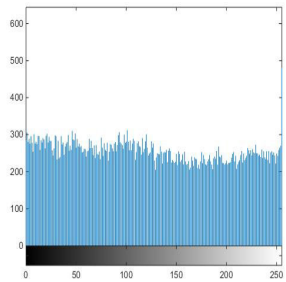
(k) $3^{rd}$ cipher image



(l) Histogram of R component of $3^{rd}$ cipher image



(m) Histogram of G component of $3^{rd}$ cipher image



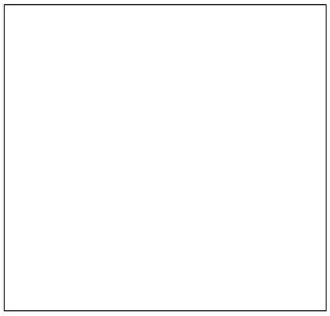
(n) Histogram of B component of $3^{rd}$ cipher image

**FIGURE 14.** A 2 gray level CT Binary image, Cipher image and their corresponding Histograms.

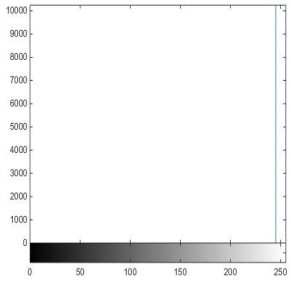
are carried out to gauge the performance of the proposed encryption algorithm. All these simulations are performed on MATLAB 14 i3-31 110M CPU @ 2.400GHZ and 8GB RAM.

### A. KEY SENSITIVITY ANALYSIS

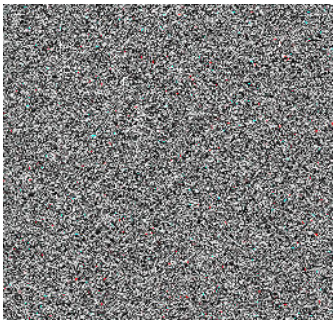
For enhancing the security of the encryption algorithm, the sensitivity of the key should be under-considered. The



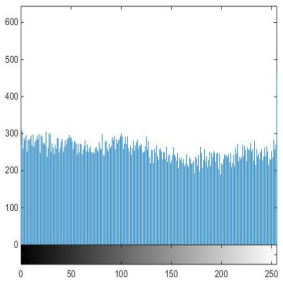
(a) All white plain image



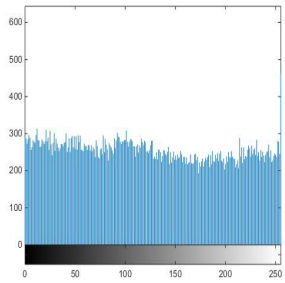
(b) Histogram of plain image



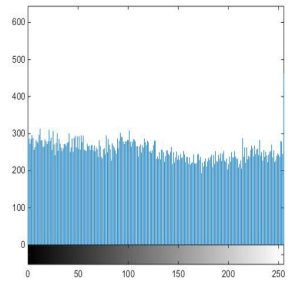
(c) $1^{st}$ cipher image



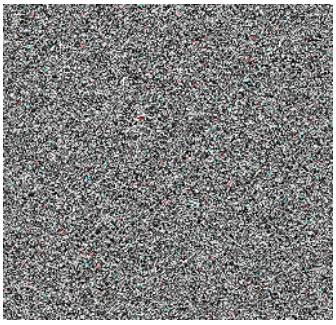
(d) Histogram of R component of $1^{st}$ cipher image



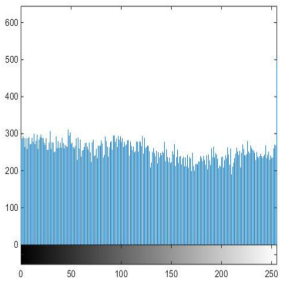
(e) Histogram of G component of $1^{st}$ cipher image



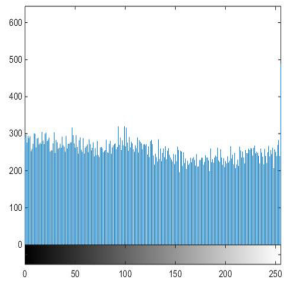
(f) Histogram of B component of $1^{st}$ cipher image



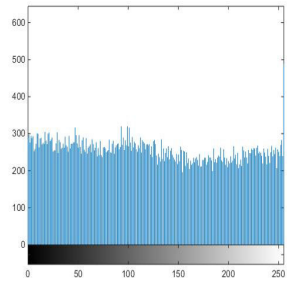
(g) $2^{nd}$ cipher image



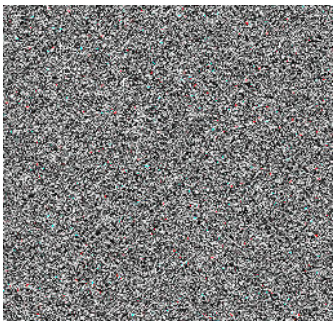
(h) Histogram of R component of $2^{nd}$ cipher image



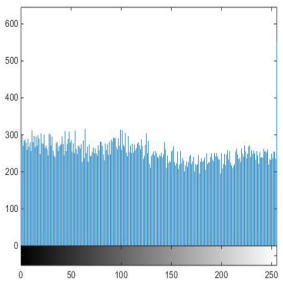
(i) Histogram of G component of $2^{nd}$ cipher image



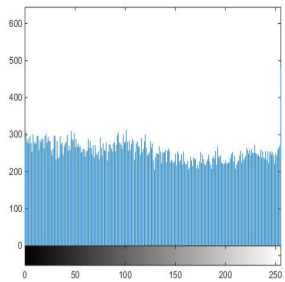
(j) Histogram of B component of $2^{nd}$ cipher image



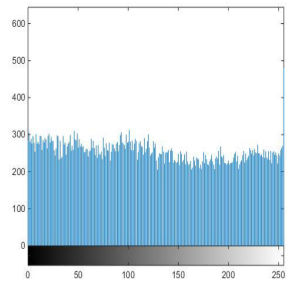
(k) $3^{rd}$ cipher image



(l) Histogram of R component of $3^{rd}$ cipher image



(m) Histogram of G component of $3^{rd}$ cipher image

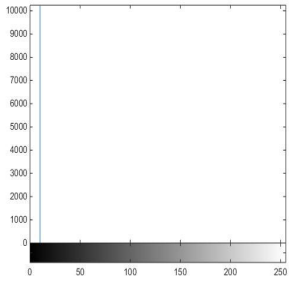


(n) Histogram of B component of $3^{rd}$ cipher image

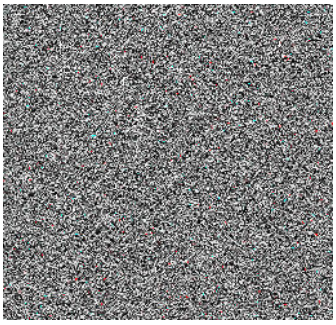
**FIGURE 15.** **A single gray level white image, Cipher image and their corresponding Histograms.**

sensitivity of the key is referred to the minor change in the original keys may result in a drastic change in the cipher images. For instance, we have added a minute number ($10^{-15}$) in each key. i.e. $K_i = 1.490000000000001$, $K_{ii} = 0.232000000000001$, $K_{num1} = 99.000000000000001$, $K_{iii} = 1.510000000000001$, $K_{iv} = 0.216000000000001$,

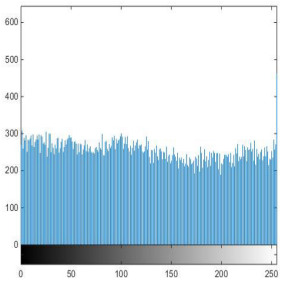
(a) All black plain image



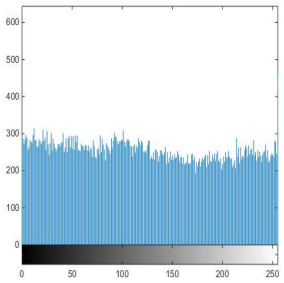
(b) Histogram of plain image



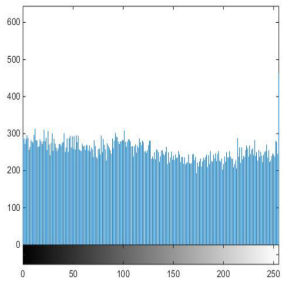
(c) $1^{st}$ cipher image



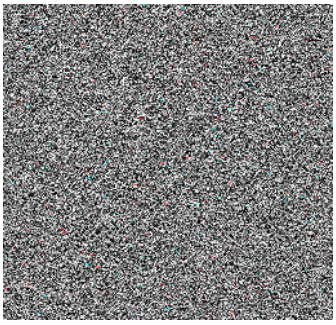
(d) Histogram of R component of $1^{st}$ cipher image



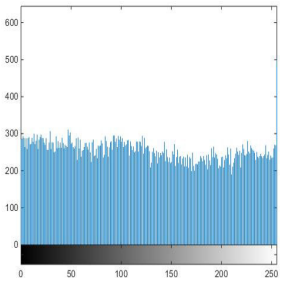
(e) Histogram of G component of $1^{st}$ cipher image



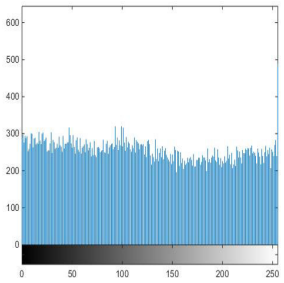
(f) Histogram of B component of $1^{st}$ cipher image



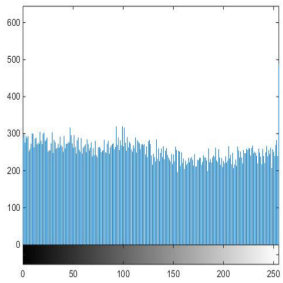
(g) $2^{nd}$ cipher image



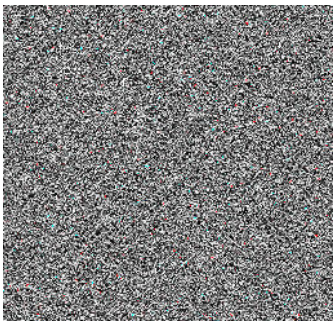
(h) Histogram of R component of $2^{nd}$ cipher image



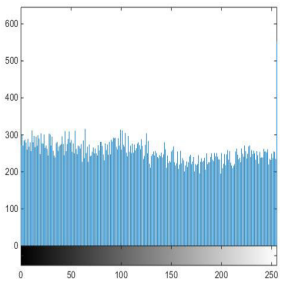
(i) Histogram of G component of $2^{nd}$ cipher image



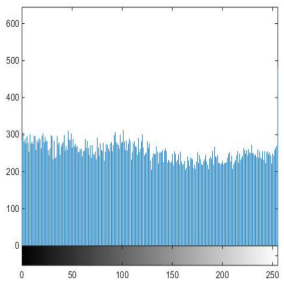
(j) Histogram of B component of $2^{nd}$ cipher image



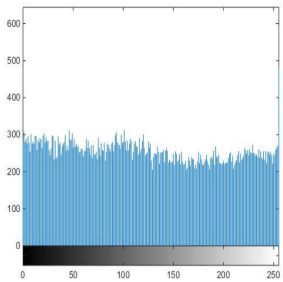
(k) $3^{rd}$ cipher image



(l) Histogram of R component of $3^{rd}$ cipher image



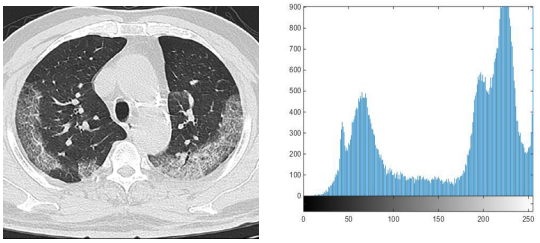
(m) Histogram of G component of $3^{rd}$ cipher image



(n) Histogram of B component of $3^{rd}$ cipher image

**FIGURE 16.** **A single gray level black plain image, Cipher image and their corresponding Histograms.**

$K_{num2} = 97.000000000000001$, $K_v = 1.470000000000001$, $K_{vi} = 0.258000000000001$, $K_{num3} = 93.000000000000001$, the cipher images which are generated after making a minor change in the original keys are completely different from the cipher images generated by employing the correct keys. We have also estimated the similarity of cipher images, which



(a) Decrypted CT image (b) Histogram of decrypted CT image

**FIGURE 17.** Reconstruction of plain image.

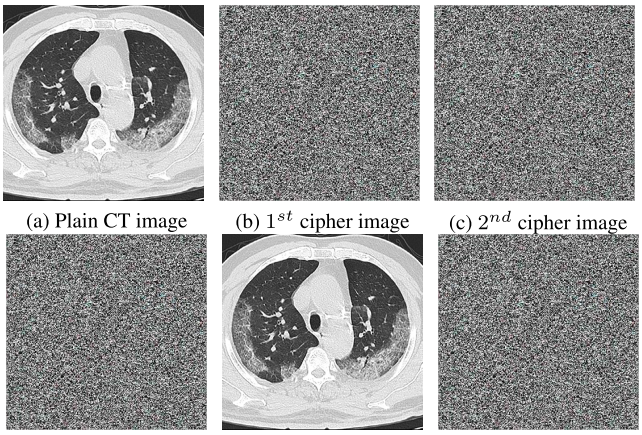
**TABLE 2.** Percentage change in cipher images.

| Original Images and the existing schemes | Cipher images | Percentage change occurred in cipher images after a minor changes made in the original keys |
|---|---|---|
| **CT image**$_{(1)}$ | $1^{st}$ cipher image | 99.6974 |
| | $2^{nd}$ cipher image | 99.3684 |
| | $3^{rd}$ cipher image | 99.8431 |
| **CT image**$_{(2)}$ | $1^{st}$ cipher image | 99.7316 |
| | $2^{nd}$ cipher image | 99.8735 |
| | $3^{rd}$ cipher image | 99.8793 |
| **CT image**$_{(3)}$ | $1^{st}$ cipher image | 99.3671 |
| | $2^{nd}$ cipher image | 99.8241 |
| | $3^{rd}$ cipher image | 99.3791 |
| **Ref [64]** | $1^{st}$ cipher image | 99.7103 |
| | $2^{nd}$ cipher image | 99.8036 |
| | $3^{rd}$ cipher image | 99.3708 |
| **Ref [65]** | $1^{st}$ cipher image | 99.3730 |
| | $2^{nd}$ cipher image | 99.7013 |
| | $3^{rd}$ cipher image | 99.6301 |
| **Ref [66]** | $1^{st}$ cipher image | 99.7301 |
| | $2^{nd}$ cipher image | 99.9324 |
| | $3^{rd}$ cipher image | 99.9350 |
| **Black image** | $1^{st}$ cipher image | 99.3021 |
| | $2^{nd}$ cipher image | 99.5301 |
| | $3^{rd}$ cipher image | 99.7310 |
| **White image** | $1^{st}$ cipher image | 99.7983 |
| | $2^{nd}$ cipher image | 99.7610 |
| | $3^{rd}$ cipher image | 99.8340 |

are generated with original and with slightly different keys, by subtracting the corresponding cipher images. After subtracting the cipher images, we have calculated the number of zeros in the resultant difference matrix, and we found that there are very few zeros. This means that the similarity between the cipher image is very less. The percentage difference between each group of cipher images is shown in Table 2.

### B. MORE ABOUT THE SENSITIVITY OF KEY

We have done key sensitivity analysis in another way as well. First, we encrypt the plain image using original keys and then made some minor changes as explained in section IV-A. These slightly different keys are then used to decrypt the original image from the cipher images. The resultant decrypted image occurs completely different from the original image. This effect can be shown in Figure 18.



(a) Plain CT image (b) $1^{st}$ cipher image (c) $2^{nd}$ cipher image

(d) $3^{rd}$ cipher image (e) Decrypted image with original keys (f) Decrypted image with slightly different keys

**FIGURE 18.** Key sensitivity analysis.

### C. KEY SPACE ANALYSIS

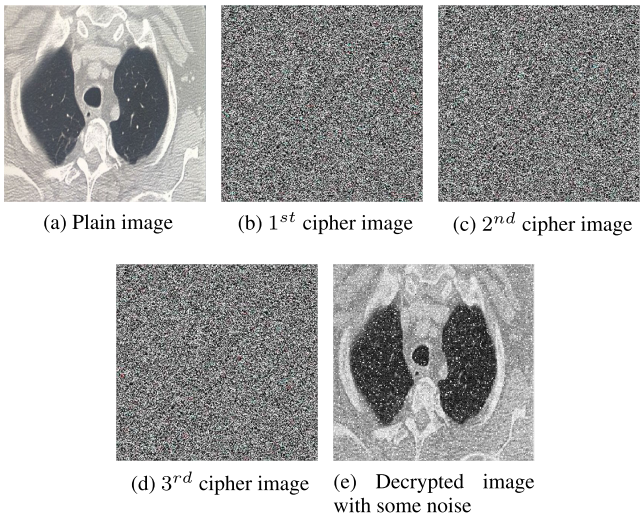
The requirements of resisting the brute force attack must be satisfied. Which is possible by enlarging the key-space. In the proposed encryption algorithm, there are nine different key parameters are used which are $K_i$, $K_{ii}$, $K_{num1}$, $K_{iii}$, $K_{iv}$, $K_{num2}$, $K_v$, $K_{vi}$ and $K_{num3}$. The numeric values of these keys are as under:

$$K_i = 1.490000000000001 \quad K_{ii} = 0.232000000000001$$
$$K_{num1} = 99.000000000000001 \quad K_{iii} = 1.510000000000001$$
$$K_{iv} = 0.216000000000001 \quad K_{num2} = 97.000000000000001$$
$$K_v = 1.470000000000001 \quad K_{vi} = 0.258000000000001$$
$$K_{num3} = 93.000000000000001$$

As the sensitivity of each key is $10^{-15}$, which means that each key has a space of $10^{15}$. According to the number of keys used in the proposed encryption algorithm, the total key-space will is $10^{15*9}$. This means the total number of possible keys will be approximately equal to $2^{135}$.

### D. NOISE ATTACK ANALYSIS

We have added some noise in all three generated cipher images corresponding to one plain image to perform the noise attack analysis. In other words, we can say that any adversary has contaminated the cipher images with some noise to destroy the meaning information. It is essential for any encryption scheme that if the encrypted image is corrupted with the noise, the encryption algorithm must be able to decrypt the original image. Obviously, after adding the noise in the cipher image, the decrypted image will not be exactly the replica of the plain image. However, the information should be visible. The addition of noise is performed as follows:



(a) Plain image (b) $1^{st}$ cipher image (c) $2^{nd}$ cipher image

(d) $3^{rd}$ cipher image (e) Decrypted image with some noise

**FIGURE 19.** Noise attack analysis.

Let the three cipher images are ($C_1$, $C_2$ and $C_3$). whereas the noise is expressed as a small integer number $N_{noise}$. we have performed a logical operation for the addition of noise in the cipher images as follows:

$$C1_{noise}(\text{i,j}) = C_1(\text{i,j}) \oplus N_{noise}$$
$$C2_{noise}(\text{i,j}) = C_2(\text{i,j}) \oplus N_{noise}$$
$$C3_{noise}(\text{i,j}) = C_3(\text{i,j}) \oplus N_{noise}$$

The decryption process is then applied on the noisy cipher images ($C1_{noise}$(i,j), $C2_{noise}$(i,j) and $C3_{noise}$(i,j)) with the original keys. The decrypted image is almost the same as the plain image. Form Figure 19, it can be seen that the information in the decrypted image can be visible. Moreover, the percentage of recovered information in different decrypted images are listed in Table 3.

### E. CROPPING ATTACK ANALYSIS

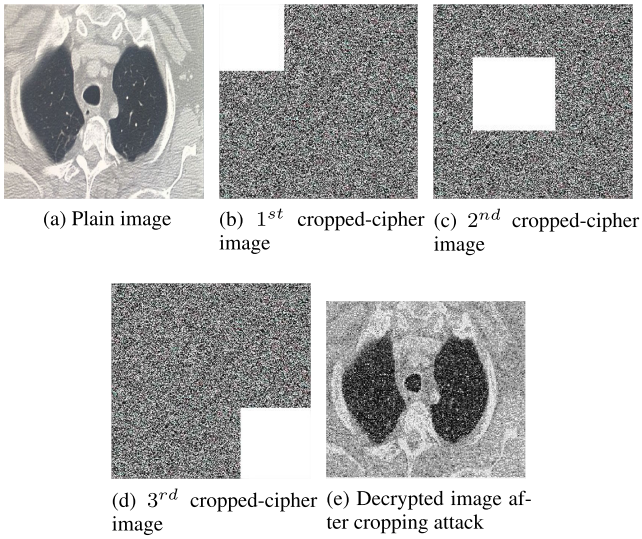
Apart from adding noise in the cipher image, the enemy can also attack by cropping the transmitted image. The image cropping may result in the loss of data. To attempt the cropping attack analysis, we have cropped a block of pixels as shown in Figure 20(b), (c) and (d) and then applied the reverse process of the proposed scheme on the cropped cipher images. The resultant decrypted image is very much similar to the plain image. Like in the noise attack analysis, after decryption, some meaningful information is also lost in this particular analysis, but the information can still easily be visualized. Figure 20 reflects the cropping attack analysis in which Figure 20(d) is a decrypted image after cropping some portion of the cipher images. The percentage loss of information after the cropping attack is given in Table 4. The procedure of evaluation of percentage loss is the same as we adopted in section IV-A

### F. SENSITIVITY ANALYSIS

For the sensitivity analysis, the Number of Pixel Change Rate (NPCR) and Unified Average Change Intensity (UACI) are

**TABLE 3.** Recovered information percentage after adding the noise in cipher text images.

| Original Images | Cipher images | Recovered information percentage |
|---|---|---|
| **CT image$_1$** | $1^{st}$ cipher image | 94.6870 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **CT image$_2$** | $1^{st}$ cipher image | 93.8361 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **CT image$_3$** | $1^{st}$ cipher image | 93.6385 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Ref [64]** | $1^{st}$ cipher image | 91.9873 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Ref [65]** | $1^{st}$ cipher image | 93.5983 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Ref [66]** | $1^{st}$ cipher image | 92.3789 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Black image** | $1^{st}$ cipher image | 91.6852 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **White image** | $1^{st}$ cipher image | 94.1368 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |



(a) Plain image (b) $1^{st}$ cropped-cipher image (c) $2^{nd}$ cropped-cipher image

(d) $3^{rd}$ cropped-cipher image (e) Decrypted image after cropping attack

**FIGURE 20.** Cropping attack analysis.

frequently used. These two measures evaluate the change in the cipher images after changes a single pixel value in the plain image. Sensitivity analysis provides evidence that a slight change in the original image can significantly change the cipher image. NPCR and UACI can be calculated as follow [67]:

$$NPCR = \frac{\sum_{a,b} D(a,b)}{MN} \times 100\% \quad (7)$$

$$UACI = \frac{1}{mn}\left[\sum_{a,b} \frac{|C_1(a,b) - C_2(a,b)|}{2^K - 1}\right] \times 100\% \quad (8)$$

where MN is the total number of pixels present in an image and $C_1$ and $C_2$ are the two different cipher images. $C_2$ is

**TABLE 4.** Percentage loss of information after the cropping attack on cipher text images.

| Original Images | Cipher images | Information loss percentage |
|---|---|---|
| **CT image$_1$** | $1^{st}$ cipher image | 10.6321 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **CT image$_2$** | $1^{st}$ cipher image | 10.8035 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Ref [64]** | $1^{st}$ cipher image | 11.0153 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Ref [65]** | $1^{st}$ cipher image | 9.99368 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Ref [66]** | $1^{st}$ cipher image | 10.3650 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **Black image** | $1^{st}$ cipher image | 10.8321 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |
| **White image** | $1^{st}$ cipher image | 10.6310 |
| | $2^{nd}$ cipher image | |
| | $3^{rd}$ cipher image | |

**TABLE 5.** NPCR analysis.

| Plain CT images | Ref [68] | Ref [69] | Ref [70] | Ref [71] | proposed |
|---|---|---|---|---|---|
| **image$_{(1)}$** | 99.5107 | 99.3662 | 99.6558 | 99.6831 | 99.6958 |
| **image$_{(2)}$** | 99.2077 | 99.6511 | 99.5926 | 99.4501 | 99.6882 |
| **image$_{(3)}$** | 99.7162 | 99.8565 | 99.5505 | 99.3878 | 99.6103 |
| **image$_{(4)}$** | 99.3281 | 99.4332 | 99.5912 | 99.4799 | 99.5930 |
| **image$_{(5)}$** | 99.5218 | 99.5227 | 99.6109 | 99.3929 | 99.6132 |
| **image$_{(6)}$** | 99.4146 | 99.5331 | 99.6399 | 99.5078 | 99.6712 |
| **Black image** | 99.1218 | 99.6010 | 99.5892 | 99.5916 | 99.7123 |
| **White image** | 99.3131 | 99.5425 | 99.5780 | 99.6785 | 99.6085 |

generated after changing one pixel in the plain image. Moreover, D(a,b) will produce two different results, either 1 or 0. It depends on the conditions on the pixels of $C_1$ and $C_2$. If the respective pixels of $C_1$(a,b) $C_2$(a,b) are same then D(a,b) will be 0 otherwise it will be 1. In equation 8, K represents the total number of bits in a pixel. i.e, for an eight-bit gray-scale image, every pixel will have 8 bits. We can evaluate the expected values of UACI and NPCR by 9 and 10.

$$Exp - NPCR = \left[1 - \frac{1}{2^K}\right] \times 100\% \tag{9}$$

$$Exp - UACI = \frac{1}{2^{2K}}\left[\frac{\sum_{Y=1}^{2^b-1} Y(Y+1)}{2^Y - 1}\right] \times 100\% \tag{10}$$

For a robust encryption algorithm, the expected values for NPCR and UACI must be nearly equal to 99.6073% and 33.4351%, respectively. We have calculated and compared NPCR and UACI values for different images with other encryption algorithms, as shown in Table 5 and 6. We found that the proposed encryption algorithm can generate the values of NPCR and UACI are above 99.6073% and 33.4351% respectively and higher than the compared schemes.

**TABLE 6.** UACI analysis.

| Plain CT images | Ref [68] | Ref [69] | Ref [70] | Ref [71] | proposed |
|---|---|---|---|---|---|
| **image$_{(1)}$** | 33.5050 | 33.3972 | 33.3125 | 33.4831 | 33.6254 |
| **image$_{(2)}$** | 33.2744 | 33.3584 | 33.4420 | 33.5949 | 33.5021 |
| **image$_{(3)}$** | 33.6049 | 33.3791 | 33.6562 | 33.5326 | 33.7015 |
| **image$_{(4)}$** | 33.8333 | 33.3358 | 33.6493 | 33.4363 | 32.6930 |
| **image$_{(5)}$** | 33.6145 | 33.4376 | 33.6118 | 33.4275 | 33.6745 |
| **Black image** | 33.4046 | 33.4075 | 33.6040 | 33.4987 | 33.6802 |
| **White image** | 33.4632 | 33.3516 | 33.6283 | 33.5261 | 33.6127 |

### G. LOSSLESS ANALYSIS

Lossless analysis refers to the information loss after the decryption of the plain image. For instance, the encryption algorithms proposed in [72], [73] are the lossy algorithms, which means these algorithms cannot reconstruct the exact pixel values of an original image, but it does not mean that these algorithms cannot decrypt the content of the plain image. However, these existing schemes are suitable for image encryption but not suitable for text encryption because in text encryption, every bit should be recovered to decrypt the corrected text. For the information loss analysis, peak signal to noise ratio (PSNR) and mean square error (MSE) are normally used. These two paraments can be calculated using 10 and 11, respectively [74], [75].

$$PSNR = 20log_{10}\left(\frac{MAX_h}{\sqrt{MSE}}\right) \tag{11}$$

where: $MAX_h$ is the highest pixel value of an original image.

$$MSE = \frac{1}{KL}\sum_{i=1}^{K}\sum_{j=1}^{L}(I(i,j) - C(i,j))^2 \tag{12}$$

where $K$ and $L$ represent the rows and columns of the plain image, respectively. Whereas, $I(i,j)$ is the original image and $C(i,j)$ is a transformed image. There is an inverse relationship between the MSE and PSNR as it is given as:

$$PSNR \propto \frac{1}{MSE} \tag{13}$$

Mean square error is an error between any two images, whether it can be a pair of plain and cipher image or plain and decrypted image. If someone performs only MSE and PSNR analysis, it always requires high values of MSE and lower values of PSNR. However, in this section, information loss analysis is presented. So we will consider plain image and decrypted images. We have encrypted several plain images and then decrypt those encrypted images using the proposed encryption scheme. After decryption, we have subtracted the plain images from the corresponding decrypted images; all the numeric values in the resultant matrix are zero, which shows that all the information is reconstructed. It means if there is no difference between the original and decrypted image, the MSE between the original and decrypted images will be zero and then, according to equation 12, PSNR should be infinite. We evaluated lossless analysis for different plain

**TABLE 7.** Lossless analysis.

| Plain CT images | Proposed algorithm | | Ref [76] | | Ref [77] | |
|---|---|---|---|---|---|---|
| | MSE | PSNR | MSE | PSNR | MSE | PSNR |
| $image_{(1)}$ | 0 | $\infty$ | 8.2486 | 39.0235 | 0.4034 | 51.0721 |
| $image_{(2)}$ | 0 | $\infty$ | 11.6899 | 37.4986 | 0.2041 | 49.3236 |
| $image_{(3)}$ | 0 | $\infty$ | 7.2465 | 39.4701 | 0.8800 | 56.2168 |
| Black image | 0 | $\infty$ | 26.6350 | 33.8652 | 0.8551 | 59.7601 |
| White image | 0 | $\infty$ | 21.7312 | 34.7823 | 1.6999 | 61.9310 |

images and reported the proposed encryption scheme statistics compared with other schemes. From Table 7, it can be seen that the MSE and PSNR values of the proposed algorithm for all the images are zero and infinity respectively. In contrast, the compared encryption algorithms have values other than zero and infinity.

### H. ENTROPY ANALYSIS

Entropy is used to measure the pixel disorder of an image. To calculate the randomness of the cipher image, the following mathematical formula is used [78]:

$$Ent(Y) = -\sum_{m=1}^{N} p(y_m) log_2(p(y_m)) \tag{14}$$

where p($y_m$) shows the probability occurrence of the symbol $y_m$ and N is the number of pixels present in an image. In an 8-bit gray-scale image, there must be $2^8$ different symbols having the range [0 255]. The maximum possible value of entropy depends on the number of bits i.e. if an image is 8-bit, the value of the entropy will never increase by 8. Similarly, if the image is 1-bit (binary image), the maximum entropy value will be 1. Here we are analyzing the 8-bit images, which means that, for strong encryption, the entropy value should be very close to 8 to create the maximum possible randomness. We have evaluated the entropy values of different plain and cipher images and compared with the entropy values of the other encryption algorithms. Table 8 presents the entropy values in which we can see that our encryption scheme has an average entropy is approximately 7.9983, which is higher than the other comparable schemes. From these analyses, one can conclude that the proposed encryption algorithm is robust against the entropy attacks.

### I. SPEED ANALYSIS AND COMPUTATION COMPLEXITY

Security is always the priority of any encryption scheme, but we cannot ignore the processing time analysis. The computational time for every algorithm may differ depending upon the computer specifications such as RAM, CPU and operating system. We cannot compare the execution time of different algorithms until the nominees are tested on the computer having the same specification. So, we have implemented various encryption schemes listed in Table 9 on the same computer and we found that the proposed algorithm is a bit faster than the other algorithms.

### J. HISTOGRAM ANALYSIS

The histogram of an image shows the frequency distribution of the gray values of an image. To resist the statistical analysis, the cipher image should have nearly uniform and it must be completely different from the histogram of the plaintext image [86]. We have analyzed histograms of some CT plain and their corresponding cipher images and found that the histogram of the cipher images generated from the proposed encryption algorithm is fairly uniform and completely different from the histogram of the original images. This effect can be seen in Figures 13-16, which shows the histogram of the plaintext images and their corresponding histogram of *RGB* component of the cipher images.

Histogram analysis can also be performed numerically by calculating the maximum and minimum curve values, the range, and the variance of the enciphered image. Mathematically, it can be calculated using equation 15 [87]. The statistical values of these parameters are displayed in Table 10. From Table 10, it can be seen that the variance values of the enciphered image are minimal when compares to the plaintext image, which ultimately reveals the uniformity of the histograms of the enciphered image.

$$V_{ar} = \frac{1}{n^2} \sum_{a=1}^{n} \sum_{b=1}^{n} \frac{1}{2} (Y_a - Y_b)^2 \tag{15}$$

where Y is the sequence of the histogram values i.e.(Y = $y_1$, $y_2$,$y_3$ ….. $y_n$) and n is the total number is values in the histogram. Other parameters such as maximum deviation ($M_D$), deviation from the uniform histogram ($D_p$) and the irregular deviation ($I_D$) can also be used to analyze the uniformity of the histograms. The maximum deviation is used to evaluate the encryption algorithm's quality by calculating the rate of change in the pixel values of the plaintext image concerning the pixels of the enciphered image. Maximum deviation values reveal a great change between the pixel values of the plaintext and enciphered image. Also, it declares that the encryption scheme is highly secure. $M_D$ can be calculated as [88]:

$$M_D = \frac{a_0 + a_{255}}{2} + \sum_{i=1}^{255} a_i \tag{16}$$

where $a_0$ and $a_{255}$ is the value difference in the histogram at position 0 and 255.

The irregular deviation is based on how much numerical histogram distribution is close to the uni from distribution. The smaller the values show that $I_D$ is close to the uniform distribution. It can be calculated as:

1) Absolute difference between the values before and after encryption:

$$D_{abs} = |P - C| \tag{17}$$

2) Calculate histogram of $D_{abs}$ = histogram (D)

**TABLE 8.** Entropy analysis.

| Images | Corresponding Cipher | Components | Original image | Ref [79] | *Proposed* |
|---|---|---|---|---|---|
| **CT image**$_{(1)}$ | $1^{st}$ cipher image | R-Component | $Image_{(1)}$ 7.1849 | $Image_{(1)}$: 7.9821 | 7.9979 |
| | | G-Component | | | 7.9971 |
| | | B-Component | | | 7.9921 |
| | | **Average Entropy**$_{(1)}$ | | | 7.9957 |
| | $2^{nd}$ cipher image | R-Component | $Image_{(2)}$: 7.2240 | $Image_{(2)}$ 7.9861 | 7.9972 |
| | | G-Component | | | 7.9971 |
| | | B-Component | | | 7.9969 |
| | | **Average Entropy**$_{(2)}$ | | | 7.9970 |
| | $3^{rd}$ cipher image | R-Component | $Image_{(3)}$: 7.4822 | $Image_{(3)}$ 7.9853 | 7.9966 |
| | | G-Component | | | 7.9974 |
| | | B-Component | | | 7.9976 |
| | | **Average Entropy**$_{(3)}$ | | | 7.9972 |
| | | **Average Entropy**$_{(net)}$ | 7.2970 | 7.9845 | 7.9966 |
| **Black image** | $1^{st}$ cipher image | | | **Ref** [80] | |
| | | R-Component | $Image_{(4)}$: 7.4730 | $Image_{(4)}$ 7.9965 | 7.9971 |
| | | G-Component | | | 7.9976 |
| | | B-Component | | | 7.9972 |
| | | **Average Entropy**$_{(1)}$ | | | 7.9973 |
| | $2^{nd}$ cipher image | R-Component | $Image_{(5)}$: 7.3821 | $Image_{(5)}$ 7.9968 | 7.9979 |
| | | G-Component | | | 7.9970 |
| | | B-Component | | | 7.9969 |
| | | **Average Entropy**$_{(2)}$ | | | 7.9972 |
| | $3^{rd}$ cipher image | R-Component | $Image_{(6)}$: 7.4932 | $Image_{(6)}$ 7.9899 | 7.9978 |
| | | G-Component | | | 7.9976 |
| | | B-Component | | | 7.9977 |
| | | **Average Entropy**$_{(3)}$ | | | 7.9977 |
| | | **Average Entropy**$_{(net)}$ | 7.4494 | 7.9984 | 7.9974 |
| **White image** | $1^{st}$ cipher image | | | **Ref** [81] | |
| | | R-Component | $Image_{(7)}$: 7.3682 | $Image_{(7)}$ 7.9893 | 7.9978 |
| | | G-Component | | | 7.9972 |
| | | B-Component | | | 7.9973 |
| | | **Average Entropy**$_{(1)}$ | | | 7.9974 |
| | $2^{nd}$ cipher image | R-Component | $Image_{(8)}$: 7.3680 | $Image_{(8)}$ 7.9973 | 7.9978 |
| | | G-Component | | | 7.9970 |
| | | B-Component | | | 7.9979 |
| | | **Average Entropy**$_{(2)}$ | | | 7.9979 |
| | $3^{rd}$ cipher image | R-Component | $Image_{(9)}$: 7.2683 | $Image_{(9)}$ 7.9886 | 7.9968 |
| | | G-Component | | | 7.9971 |
| | | B-Component | | | 7.9970 |
| | | **Average Entropy**$_{(3)}$ | | | 7.9969 |
| | | **Average Entropy**$_{(net)}$ | 7.3348 | 7.9917 | 7.9974 |

**TABLE 9.** Processing time analysis.

| Encryption schemes | **Ref** [82] | **Ref** [83] | **Ref** [84] | **Ref** [85] | **Ref** [32] | Proposed |
|---|---|---|---|---|---|---|
| **Computational time(sec)** | 17.6 | 3.54 | 6 .6063 | 5.926 s – 6.018 s | 3.5781 | 3 .3021 |

3) Find the average values, that how many pixels are deviated from each deviation:

$$M_H = \frac{1}{255}\sum_{i=1}^{255} h_i \tag{18}$$

where $h_i$ is the amplitude of histogram

4) Subtract $h_i$ and $M_H$ and take the absolute values as:

$$H_{D_i} = |h_i - M_H| \tag{19}$$

5) Now from the equation 17, 18 and 19, we can calculate $I_D$ deviation as follows [89]:

$$I_D = \sum_{i=1}^{255} H_{D_i} \tag{20}$$

$D_p$ measures the deviation of the histogram of the enciphered image from the ideal histogram. For the highly secure encryption scheme, it is always required minimum values of $D_p$. It can be calculated using equation 10 [90].

$$D_p = \frac{\sum_{C_i=0}^{255} |H_{C_i} - H_C|}{M \times N} \tag{21}$$

$H_C$ and $H_{C_i}$ are the histogram of the enciphered image and ideally enciphered image. $H_{C_i}$ can be calculated as:

$$H_{C_i} = \begin{cases} \dfrac{M \times N}{256} & 0 \le C_i \le 255 \\ 0 & elsewhere \end{cases}$$

From Table 10, it can be analyzed that the values of $I_D$, $M_D$ and $D_p$ are comparatively better than the existing schemes.

### K. ENERGY ANALYSIS

The energy of an image is referred to as the quantity of the information present in an image. More energy will result in more information. Plain images always contain more energy

**TABLE 10.** Variance and encryption quality analysis.

| | Variance analysis | | | |
|---|---|---|---|---|
| | Min | Max | Range | Variance |
| **Encryption schemes** | | **Plaintext image** | | |
| | 0 | 560 | 560 | $2.8501e^{+04}$ |
| | | **Enciphered image** | | |
| **Ref [91]** | 220 | 301 | 81 | 241.1983 |
| **Ref [64]** | 225 | 327 | 102 | 290.2106 |
| **Ref [65]** | 245 | 356 | 111 | 285.3021 |
| **Ref [66]** | 201 | 288 | 87 | 283.3710 |
| **Proposed** | 198 | 295 | 82 | 239.7301 |
| | **Encryption quality analysis** | | | |
| | $\mathbf{M}_D$ | $\mathbf{I}_D$ | $\mathbf{D}_P$ | |
| **Ref [91]** | 37,981 | 39542 | 0.0551 | |
| **Ref [64]** | 36,483 | 39708 | 0.0518 | |
| **Ref [65]** | 39,183 | 37297 | 0.0526 | |
| **Ref [66]** | 35,790 | 37483 | 0.0556 | |
| **Proposed** | 40,837 | 37152 | 0.0510 | |

than cipher images. The relation between energy and information is given as:

$$\text{Energy} \propto \text{information}$$

For strong encryption, energy values must be closer to zero. Form Table 11, it can be seen that the energy values of the plain image are high because of more information. For the cipher images generated through the proposed scheme, the energy values are much closer to zero and lower than the other encryption schemes. This shows that the proposed algorithm can significantly reduce the energy value of the cipher image. The following mathematical formula can be used to calculate the energy value of the image [92].

$$E = \sum_{N=1}^{M} P(i,j)^2 \tag{22}$$

where: M is the total number of pixels in an image and p(i,j) is the position of the pixel at $i^{th}$ row and $j^{th}$ column.

### L. PERMUTATION PROCESS EVALUATION

We have considered MATLAB to implement the encryption processes proposed in [93]–[96] and generated the permuted images corresponding to each permutation process. To evaluate the proposed and existing permutation processes, we have performed some security analysis such as entropy, contrast, PSNR and MSE on the permuted images generated through the permutation process proposed in [93]–[96]. The statistical values for the proposed and existing permutation process are listed in Table 12. From Table 12, it can be analyzed that after applying the proposed and the permutation process proposed in [93]–[96], the proposed permutation process can generate more secure and random permuted images.

### M. CHI-SQUARE ANALYSIS

Apart from the visualized demonstration of the pixel values for the ciphertext images, a chi-square test is frequently used. It can be calculated by the equation 23.

$$W_{test}^2 = \sum_{i=1}^{L} \frac{(P_i - Q_i)^2}{Q_i} \tag{23}$$

where L is the total number of gray values in the image, $p_i$ and $Q_i$ represent the frequency occurrence of each pixel value obtained from the extermination and the expected frequency occurrence of each pixel value respectively. The values of chi-square analysis for the proposed encryption algorithm corresponding to the different ciphertext images are given in Table 13. It can be seen from Table 13 that the chi-square values obtained for the proposed algorithm are all less than the theoretical value 293.24783. Therefore, the gray values in the ciphertext image obtained through the proposed algorithm are uniformly distributed and pass the chi-square test.

### N. NIST SP 800-22 TEST

The NIST SP 800-22 test comprises fifteen statistical test suite for PNGs (pseudo-random number generators) and RNGs (random number generators). NIST tests produce different p-values, which could be helpful to determine whether the generated random sequence is accepted or not.

P-vales depend on the condition; if the p-value is assumed 0.001 as condition, it must be 0.001 < P-value < 0.01 for the successful bit test. Table 14 shows the NIST test values and it can be seen that the random sequence generated in the proposed work has passed all NIST tests.

### O. CLASSICAL ATTACKS

We have performed resistance attack analysis on our proposed encryption scheme to show the robustness against the four classical types of attack such as Ciphertext only, known-plaintext, chosen Ciphertext and chosen-plaintext attack. Among all these attacks, the chosen-ciphertext attack is the most powerful attack. If the encryption algorithm can resist this type of attack, it can resist other attacks as well [97].

#### 1) CIPHERTEXT ONLY ATTACK

In this case, the adversary has access to a set of cipher images. The attackers attempt to decrypt the known cipher images using a decryption algorithm [97]. The objective is to find the correct key so that the attacker can be able to decrypt the other messages as well. To prevent the plaintext message from this kind of attack, a confusion-diffusion network plays a vital role. With the permutation using chaos and diffusion using XOR operation and discrete wavelet transform, it is nearly impossible to execute this kind of attack in the proposed encryption algorithm.

#### 2) KNOWN PLAINTEXT ATTACK

In this case, the adversary has access to the collection of different plaintext images. Using the known-plaintext images, the adversary encrypts those images using the encryption algorithm and tries to find the correct key [97]. From the key space analysis presented earlier, it is clear that the adversary

**TABLE 11.** Energy analysis.

| Images | Corresponding Cipher Images | Components | Original image | Ref [79] | *Proposed* |
|---|---|---|---|---|---|
| **CT image**$_{(1)}$ | $1^{st}$ cipher image | R-Component | $Image_{(1)}$: 0.1535 | $Image_{(1)}$ 0.0193 | 0.0158 |
| | | G-Component | | | 0.0158 |
| | | B-Component | | | 0.0158 |
| | | **Average Energy**$_{(1)}$ | | | 0.0158 |
| | $2^{nd}$ cipher image | R-Component | $Image_{(2)}$: 0.1512 | $Image_{(2)}$ 0.0159 | 0.0155 |
| | | G-Component | | | 0.0158 |
| | | B-Component | | | 0.0154 |
| | | **Average Energy**$_{(2)}$ | | | 0.0155 |
| | $3^{rd}$ cipher image | R-Component | $Image_{(3)}$: 0.1610 | $Image_{(3)}$ 0.0157 | 0.0156 |
| | | G-Component | | | 0.0157 |
| | | B-Component | | | 0.0154 |
| | | **Average Energy**$_{(3)}$ | | | 0.0155 |
| | | **Average Energy**$_{(net)}$ | 0.01552 | 0.0169 | 0.0155 |
| **Black image** | $1^{st}$ cipher image | | | **Ref** [80] | |
| | | R-Component | $Image_{(4)}$: 0.1598 | $Image_{(4)}$ 0.0156 | 0.0158 |
| | | G-Component | | | 0.0158 |
| | | B-Component | | | 0.0159 |
| | | **Average Energy**$_{(1)}$ | | | 0.0158 |
| | $2^{nd}$ cipher image | R-Component | $Image_{(5)}$: 0.1681 | $Image_{(5)}$ 0.0155 | 0.0153 |
| | | G-Component | | | 0.0156 |
| | | B-Component | | | 0.0158 |
| | | **Average Energy**$_{(2)}$ | | | 0.0155 |
| | $3^{rd}$ cipher image | R-Component | $Image_{(6)}$: 0.1455 | $Image_{(6)}$ 0.0159 | 0.0155 |
| | | G-Component | | | 0.0157 |
| | | B-Component | | | 0.0158 |
| | | **Average Energy**$_{(3)}$ | | | 0.0156 |
| | | **Average Energy**$_{(net)}$ | 0.0156 | 0.0156 | 0.0156 |
| **White image** | $1^{st}$ cipher image | | | **Ref** [81] | |
| | | R-Component | $Image_{(7)}$: 0.1148 | $Image_{(7)}$ 0.0156 | 0.0152 |
| | | G-Component | | | 0.0156 |
| | | B-Component | | | 0.0152 |
| | | **Average Energy**$_{(1)}$ | | | 0.0153 |
| | $2^{nd}$ cipher image | R-Component | $Image_{(8)}$: 0.1241 | $Image_{(8)}$ 0.0158 | 0.0155 |
| | | G-Component | | | 0.0153 |
| | | B-Component | | | 0.0158 |
| | | **Average Energy**$_{(2)}$ | | | 0.0153 |
| | $3^{rd}$ cipher image | R-Component | $Image_{(9)}$: 0.1157 | $Image_{(9)}$ 0.0159 | 0.0155 |
| | | G-Component | | | 0.0155 |
| | | B-Component | | | 0.0158 |
| | | **Average Energy**$_{(3)}$ | | | 0.0156 |
| | | **Average Energy**$_{(net)}$ | 0.1182 | 0.0157 | 0.0154 |

**TABLE 12.** Permutation process evaluation.

| **Security** parameters | **Ref** [82] | **Ref** [83] | **Ref** [84] | **Ref** [85] | Proposed |
|---|---|---|---|---|---|
| Entropy | 7.9861 | 7.9763 | 7.9874 | 7.9726 | 7.9968 |
| Contrast | 9.9876 | 9.9964 | 9.9934 | 10.7681 | 10.8671 |
| PSNRs | 20.9755 | 25.6781 | 27.3715 | 29.3781 | 18.9756 |
| MSE | 23.6710 | 24.9781 | 24.9783 | 23.9961 | 25.9782 |
| Energy | 0.0167 | 0.0168 | 0.0162 | 0.0159 | 0.0157 |

**TABLE 13.** Chi-square analysis.

| **Images** | $W^2{}_{th}$ | $W^2{}_{proposed}$ | Result |
|---|---|---|---|
| **Lena** | 293.24783 | 253.6781 | Pass |
| **Baboon** | 293.24783 | 261.3571 | Pass |
| **Cameraman** | 293.24783 | 264.3468 | Pass |
| **Boat** | 293.24783 | 259.3601 | Pass |

**TABLE 14.** NIST test analysis.

| Statistical results | P-value | Result |
|---|---|---|
| Cusum-Forward | 0.427234 | Success |
| Frequency | 0.873571 | Success |
| Block Frequency (m = 128) | 0.256211 | Success |
| Long Runs of Ones | 0.799356 | Success |
| Cusum-Reverse | 0.417224 | Success |
| Runs | 0.749556 | Success |
| Non Overlapping Templates (m = 9, B = 000000001) | 0.534410 | Success |
| Rank | 0.125426 | Success |
| Spectral DFT | 0.729347 | Success |
| Approximate Entropy (m = 10) | 0.827492 | Success |
| Overlapping Templates (m = 9) | 0.534549 | Success |
| Universal | 0.314426 | Success |
| Random Excursions Variant (x = 1) | 0.216310 | Success |
| Approximate Entropy (m = 10) | 0.827492 | Success |
| Random Excursions (x = +1) | 0.454652 | Success |
| Serial (m = 16) | 0.513420 | Success |

is nearly unable to find the correct key in a meaningful time slot.

### 3) CHOSEN CIPHERTEXT ATTACK

In this case, an attacker chooses a piece of cipher image and gathers the information using the ciphertext image and generating their corresponding decryption under the unknown key [97].In the proposed encryption algorithm, the keystream relies on the private key. Hence, the keystream

restored with one chosen-plaintext cannot decipher other messages, which implies that our algorithm can resist the chosen ciphertext attack.

#### 4) CHOSEN PLAINTEXT ATTACK

In this case, an attacker chooses a piece of a plain image and gather the information using the ciphertext image and generating their corresponding encryption under the unknown key [97]. The proposed encryption algorithms can resist the chosen plaintext attack in two aspects. First, the random sequence generator generates the scrambling sequence which scrambles the pixel values of the plaintext image. The scrambling sequence for different images can be different by using different inputs to the random sequence generator. Secondly, in the pixel diffusion phase, each pixel is diffused by deploying DWT and XOR operations, making the proposed encryption algorithm resist the attack of selecting plaintext.

## V. CONCLUSION

In this paper, efficient image encryption is proposed to produce three RGB cipher images corresponding to one gray-scale image. The proposed algorithm is suitable for both image and text encryption because, during the decryption process, every single bit can be recover. The proposed scheme uses bit-plane extraction, Discrete Wavelet Transform and chaos to encrypt the plain image. It has three different sections, based on which the security of proposed encryption is enhanced significantly. We have used spatial domain encryption in the first and last sections, while the frequency domain encryption is in the middle of the other two sections. The purpose of using the spatial and frequency domain encryption in a single encryption algorithm is to enhance security and reduce the processing time of the proposed encryption scheme. Moreover, for the security measurements, we have done a number of security analysis in detail to show the effectiveness of the proposed algorithm.

## VI. FUTURE WORK

In future work, we aim to introduce an intelligent scheme using machine learning to enhance the security and encryption speed of the proposed work. In this scheme, different features will extract from the plain image, and encrypt only region of interests to make it more fast and robust for practical and real-rime cryptographic applications.

### Conflict of Interest

- The authors declare no conflict of interest.

## REFERENCES

[1] A. Roy, A. P. Misra, and S. Banerjee, "Chaos-based image encryption using vertical-cavity surface-emitting lasers," *Optik*, vol. 176, pp. 119–131, Jan. 2019.
[2] J. S. Khan and J. Ahmad, "Chaos based efficient selective image encryption," *Multidimensional Syst. Signal Process.*, vol. 30, no. 2, pp. 943–961, Apr. 2019.
[3] D. Selent, "Advanced encryption standard," *Rivier Acad. J.*, vol. 6, no. 2, pp. 1–14, 2010.
[4] D. Coppersmith, "The data encryption standard (DES) and its strength against attacks," *IBM J. Res. Develop.*, vol. 38, no. 3, pp. 243–250, May 1994.
[5] B. M. Hennelly and J. T. Sheridan, "Image encryption and the fractional Fourier transform," *Optik*, vol. 114, no. 6, pp. 251–265, 2003.
[6] J. B. Lima and L. F. G. Novaes, "Image encryption based on the fractional Fourier transform over finite fields," *Signal Process.*, vol. 94, pp. 521–530, Jan. 2014.
[7] G. Bhatnagar, Q. M. J. Wu, and B. Raman, "Discrete fractional wavelet transform and its application to multiple encryption," *Inf. Sci.*, vol. 223, pp. 297–316, Feb. 2013.
[8] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognit.*, vol. 37, no. 4, pp. 725–737, Apr. 2004.
[9] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.
[10] I. Hussain, A. Anees, A. H. AlKhaldi, A. Algarni, and M. Aslam, "Construction of chaotic quantum magnets and matrix Lorenz systems S-boxes and their applications," *Chin. J. Phys.*, vol. 56, no. 4, pp. 1609–1621, Aug. 2018.
[11] R. Ge, G. Yang, J. Wu, Y. Chen, G. Coatrieux, and L. Luo, "A novel chaos-based symmetric image encryption using bit-pair level process," *IEEE Access*, vol. 7, pp. 99470–99480, 2019.
[12] A. Anees and I. Hussain, "A novel method to identify initial values of chaotic maps in cybersecurity," *Symmetry*, vol. 11, no. 2, p. 140, Jan. 2019.
[13] I. Hussain, F. Ahmed, U. M. Khokhar, and A. Anees, "Applied cryptography and noise resistant data security," *Secur. Commun. Netw.*, vol. 2018, pp. 1–2, Dec. 2018.
[14] X. Zhang, L. Wang, Z. Zhou, and Y. Niu, "A chaos-based image encryption technique utilizing Hilbert curves and H-fractals," *IEEE Access*, vol. 7, pp. 74734–74746, 2019.
[15] A. Anees and Y.-P. P. Chen, "Designing secure substitution boxes based on permutation of symmetric group," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 7045–7056, 2020.
[16] Q. Lu, C. Zhu, and X. Deng, "An efficient image encryption scheme based on the LSS chaotic map and single S-box," *IEEE Access*, vol. 8, pp. 25664–25678, 2020.
[17] A. Kanso and M. Ghebleh, "A novel image encryption algorithm based on a 3D chaotic map," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 7, pp. 2943–2959, Jul. 2012.
[18] X. Wu, B. Zhu, Y. Hu, and Y. Ran, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access*, vol. 5, pp. 6429–6436, 2017.
[19] I. Hussain, A. Anees, T. Al-Maadeed, and M. Mustafa, "Construction of S-box based on chaotic map and algebraic structures," *Symmetry*, vol. 11, no. 3, p. 351, Mar. 2019.
[20] A. Anees and Y.-P.-P. Chen, "Discriminative binary feature learning and quantization in biometric key generation," *Pattern Recognit.*, vol. 77, pp. 289–305, May 2018.
[21] N. H. Beebe, "A complete bibliography of the bell system technical journal, 1940–1949," 2020.
[22] F. Ahmed, A. Anees, V. U. Abbas, and M. Y. Siyal, "A noisy channel tolerant image encryption scheme," *Wireless Pers. Commun.*, vol. 77, no. 4, pp. 2771–2791, Aug. 2014.
[23] A. Anees and M. A. Gondal, "Construction of nonlinear component for block cipher based on one-dimensional chaotic map," *3D Res.*, vol. 6, no. 2, p. 17, Jun. 2015.
[24] X. Wang and S. Gao, "Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory," *Inf. Sci.*, vol. 507, pp. 16–36, Jan. 2020.
[25] C. Pak, K. An, P. Jang, J. Kim, and S. Kim, "A novel bit-level color image encryption using improved 1D chaotic map," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 12027–12042, May 2019.
[26] S. L. P. Ching and F. Yunos, "Effect of self-invertible matrix on cipher hexagraphic polyfunction," *Cryptography*, vol. 3, no. 2, p. 15, Jun. 2019.
[27] Z.-H. Gan, X.-L. Chai, D.-J. Han, and Y.-R. Chen, "A chaotic image encryption algorithm based on 3-D bit-plane permutation," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7111–7130, Nov. 2019.
[28] A. Anees and Z. Ahmed, "A technique for designing substitution box based on van der Pol oscillator," *Wireless Pers. Commun.*, vol. 82, no. 3, pp. 1497–1503, Jun. 2015.
[29] A. Shafique and J. Ahmed, "Dynamic substitution based encryption algorithm for highly correlated data," *Multidimensional Syst. Signal Process.*, vol. 32, no. 1, pp. 91–114, Jan. 2021.

[30] X. Wang, X. Zhu, X. Wu, and Y. Zhang, "Image encryption algorithm based on multiple mixed hash functions and cyclic shift," *Opt. Lasers Eng.*, vol. 107, pp. 370–379, Aug. 2018.

[31] P. Kumar, A. Fatima, and N. K. Nishchal, "Image encryption using phase-encoded exclusive-OR operations with incoherent illumination," *J. Opt.*, vol. 21, no. 6, Jun. 2019, Art. no. 065701.

[32] I. Hussain, A. Anees, M. Aslam, R. Ahmed, and N. Siddiqui, "A noise resistant symmetric key cryptosystem based on $S_8$ S-boxes and chaotic maps," *Eur. Phys. J. Plus*, vol. 133, no. 4, pp. 1–23, Apr. 2018.

[33] S. S. Jamal, A. Anees, M. Ahmad, M. F. Khan, and I. Hussain, "Construction of cryptographic S-boxes based on Mobius transformation and chaotic tent-sine system," *IEEE Access*, vol. 7, pp. 173273–173285, 2019.

[34] A. Anees, W. A. Khan, M. A. Gondal, and I. Hussain, "Application of mean of absolute deviation method for the selection of best nonlinear component based on video encryption," *Zeitschrift für Naturforschung A*, vol. 68, nos. 6–7, pp. 479–482, Jul. 2013.

[35] W. Major, W. J. Buchanan, and J. Ahmad, "An authentication protocol based on chaos and zero knowledge proof," *Nonlinear Dyn.*, vol. 99, pp. 1–23, Jan. 2020.

[36] S. E. El-Khamy, N. O. Korany, and A. G. Mohamed, "A new fuzzy-DNA image encryption and steganography technique," *IEEE Access*, vol. 8, pp. 148935–148951, 2020.

[37] H. Huang, "Novel scheme for image encryption combining 2D logistic-sine-cosine map and double random-phase encoding," *IEEE Access*, vol. 7, pp. 177988–177996, 2019.

[38] A. Anees and A. M. Siddiqui, "A technique for digital watermarking in combined spatial and transform domains using chaotic maps," in *Proc. 2nd Nat. Conf. Inf. Assurance (NCIA)*, Dec. 2013, pp. 119–124.

[39] X. Wang, Q. Wang, and Y. Zhang, "A fast image algorithm based on rows and columns switch," *Nonlinear Dyn.*, vol. 79, no. 2, pp. 1141–1149, Jan. 2015.

[40] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, pp. 172–182, Apr. 2014.

[41] A. Shafique and J. Shahid, "Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps," *Eur. Phys. J. Plus*, vol. 133, no. 8, p. 331, Aug. 2018.

[42] H. Wen and S. Yu, "Cryptanalysis of an image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps," *Eur. Phys. J. Plus*, vol. 134, no. 7, p. 337, Jul. 2019.

[43] X.-Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dyn.*, vol. 62, no. 3, pp. 615–621, Nov. 2010.

[44] W. Zhang, K.-W. Wong, H. Yu, and Z.-L. Zhu, "An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 8, pp. 2066–2080, Aug. 2013.

[45] Y. Zhang and D. Xiao, "An image encryption scheme based on rotation matrix bit-level permutation and block diffusion," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 1, pp. 74–82, Jan. 2014.

[46] A.-V. Diaconu, "Circular inter–intra pixels bit-level permutation and chaos-based image encryption," *Inf. Sci.*, vols. 355–356, pp. 314–327, Aug. 2016.

[47] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Opt. Lasers Eng.*, vol. 78, pp. 17–25, Mar. 2016.

[48] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Opt. Lasers Eng.*, vol. 90, pp. 238–246, Mar. 2017.

[49] A. Anees, A. M. Siddiqui, and F. Ahmed, "Chaotic substitution for highly autocorrelated data in encryption algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 9, pp. 3106–3118, Sep. 2014.

[50] A. Shafique, "A new algorithm for the construction of substitution box by using chaotic map," *Eur. Phys. J. Plus*, vol. 135, no. 2, pp. 1–13, Feb. 2020.

[51] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal Process.*, vol. 141, pp. 217–227, Dec. 2017.

[52] A. M. Vengadapurvaja *et al.*, "An efficient homomorphic medical image encryption algorithm for cloud storage security," *Procedia Comput. Sci.*, vol. 115, pp. 643–650, 2017.

[53] K. Shankar and S. K. Lakshmanaprabu, "Optimal key based homomorphic encryption for color image security aid of ant lion optimization algorithm," *Int. J. Eng. Technol.*, vol. 7, no. 9, pp. 22–27, 2018.

[54] M. Sokouti, A. Zakerolhosseini, and B. Sokouti, "Medical image encryption: An application for improved padding based GGH encryption algorithm," *Open Med. Inform. J.*, vol. 10, p. 11, 2016.

[55] H. Nematzadeh, R. Enayatifar, H. Motameni, F. G. Guimarães, and V. N. Coelho, "Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices," *Opt. Lasers Eng.*, vol. 110, pp. 24–32, Nov. 2018.

[56] R. Parvaz and M. Zarebnia, "A combination chaotic system and application in color image encryption," *Opt. Laser Technol.*, vol. 101, pp. 30–41, May 2018.

[57] A. Anees, I. Hussain, A. Algarni, and M. Aslam, "A robust watermarking scheme for online multimedia copyright protection using new chaotic map," *Secur. Commun. Netw.*, vol. 2018, Jun. 2018, Art. no. 1840207.

[58] N. F. Elabady, H. M. Abdalkader, M. I. Moussa, and S. F. Sabbeh, "Image encryption based on new one-dimensional chaotic map," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Apr. 2014, pp. 1–6.

[59] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, Mar. 2011.

[60] Y. Liu, Z. Qin, and J. Wu, "Cryptanalysis and enhancement of an image encryption scheme based on bit-plane extraction and multiple chaotic maps," *IEEE Access*, vol. 7, pp. 74070–74080, 2019.

[61] I. Hussain, A. Anees, A. H. Alkhaldi, M. Aslam, N. Siddiqui, and R. Ahmed, "Image encryption based on Chebyshev chaotic map and $S_8$ S-boxes," *Optica Applicata*, vol. 49, no. 2, pp. 317–330, 2019.

[62] A. Shafique and F. Ahmed, "Image encryption using dynamic S-box substitution in the wavelet domain," *Wireless Pers. Commun.*, vol. 115, no. 3, pp. 2243–2268, 2020.

[63] A. Anees, "An image encryption scheme based on lorenz system for low profile applications," *3D Res.*, vol. 6, no. 3, p. 24, Sep. 2015.

[64] A. Firdous, A. U. Rehman, and M. M. S. Missen, "A highly efficient color image encryption based on linear transformation using chaos theory and SHA-2," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 24809–24835, Sep. 2019.

[65] F. Musanna and S. Kumar, "A novel fractional order chaos-based image encryption using Fisher Yates algorithm and 3-D cat map," *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 14867–14895, Jun. 2019.

[66] Y. Zhang, "The fast image encryption algorithm based on lifting scheme and chaos," *Inf. Sci.*, vol. 520, pp. 177–194, May 2020.

[67] F. Ozkaynak, "Role of NPCR and UACI tests in security problems of chaos based image encryption algorithms and possible solution proposals," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 621–624.

[68] S. Noshadian, A. Ebrahimzade, and S. J. Kazemitabar, "Optimizing chaos based image encryption," *Multimedia Tools Appl.*, vol. 77, no. 19, pp. 25569–25590, Oct. 2018.

[69] Y. Luo, R. Zhou, J. Liu, S. Qiu, and Y. Cao, "An efficient and self-adapting colour-image encryption algorithm based on chaos and interactions among multiple layers," *Multimedia Tools Appl.*, vol. 77, no. 20, pp. 26191–26217, Oct. 2018.

[70] Y. P. K. Nkandeu and A. Tiedeu, "An image encryption algorithm based on substitution technique and chaos mixing," *Multimedia Tools Appl.*, vol. 78, no. 8, pp. 10013–10034, Apr. 2019.

[71] Z. Li, C. Peng, L. Li, and X. Zhu, "A novel plaintext-related image encryption scheme using hyper-chaotic system," *Nonlinear Dyn.*, vol. 94, no. 2, pp. 1319–1333, Oct. 2018.

[72] K. Gupta and S. Silakari, "Novel approach for fast compressed hybrid color image cryptosystem," *Adv. Eng. Softw.*, vol. 49, pp. 29–42, Jul. 2012.

[73] C. K. Huang and H. H. Nien, "Multi chaotic systems based pixel shuffle for image encryption," *Opt. Commun.*, vol. 282, no. 11, pp. 2123–2127, Jun. 2009.

[74] S. S. Jamal, M. U. Khan, and T. Shah, "A watermarking technique with chaotic fractional S-box transformation," *Wireless Pers. Commun.*, vol. 90, no. 4, pp. 2033–2049, Oct. 2016.

[75] F. A. Khan, J. Ahmed, J. S. Khan, J. Ahmad, and M. A. Khan, "A novel image encryption based on lorenz equation, Gingerbreadman chaotic map and $S_8$ permutation," *J. Intell. Fuzzy Syst.*, vol. 33, no. 6, pp. 3753–3765, Nov. 2017.

[76] T. S. Ali and R. Ali, "A new chaos based color image encryption algorithm using permutation substitution and Boolean operation," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 19853–19873, Jul. 2020.

[77] L. Zhang and X. Zhang, "Multiple-image encryption algorithm based on bit planes and chaos," *Multimedia Tools Appl.*, vol. 79, nos. 29–30, pp. 20753–20771, Aug. 2020.

[78] A. Anees, A. M. Siddiqui, J. Ahmed, and I. Hussain, "A technique for digital steganography using chaotic maps," *Nonlinear Dyn.*, vol. 75, no. 4, pp. 807–816, Mar. 2014.
[79] Ü. Çavuşoğlu, S. Kaçar, A. Zengin, and I. Pehlivan, "A novel hybrid encryption algorithm based on chaos and S-AES algorithm," *Nonlinear Dyn.*, vol. 92, no. 4, pp. 1745–1759, Jun. 2018.
[80] Y. Liu, Z. Jiang, X. Xu, F. Zhang, and J. Xu, "Optical image encryption algorithm based on hyper-chaos and public-key cryptography," *Opt. Laser Technol.*, vol. 127, Jul. 2020, Art. no. 106171.
[81] L. Ding and Q. Ding, "A novel image encryption scheme based on 2D fractional chaotic map, DWT and 4D hyper-chaos," *Electronics*, vol. 9, no. 8, p. 1280, Aug. 2020.
[82] A. Gupta, D. Singh, and M. Kaur, "An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 3, pp. 1309–1324, Mar. 2020.
[83] Z. Hua, S. Yi, and Y. Zhou, "Medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Process.*, vol. 144, pp. 134–144, Mar. 2018.
[84] M. Z. Talhaoui, X. Wang, and A. Talhaoui, "A new one-dimensional chaotic map and its application in a novel permutation-less image encryption scheme," *Vis. Comput.*, pp. 1–12, Aug. 2020.
[85] K. A. K. Patro, B. Acharya, and V. Nath, "Secure, lossless, and noise-resistive image encryption using chaos, hyper-chaos, and DNA sequence operation," *IETE Tech. Rev.*, vol. 37, no. 3, pp. 223–245, May 2020.
[86] L. Gong, K. Qiu, C. Deng, and N. Zhou, "An image compression and encryption algorithm based on chaotic system and compressive sensing," *Opt. Laser Technol.*, vol. 115, pp. 257–267, Jul. 2019.
[87] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice," *Inf. Sci.*, vol. 273, pp. 329–351, Jul. 2014.
[88] N. F. El Fishawy and O. M. A. Zaid, "Quality of encryption measurement of bitmap images with rc6, mrc6, and rijndael block cipher algorithms," *IJ Netw. Secur.*, vol. 5, no. 3, pp. 241–251, 2007.
[89] H. M. Elkamchouchi and M. A. Makar, "Measuring encryption quality for bitmap images encrypted with Rijndael and Kamkar block ciphers," in *Proc. 22nd Nat. Radio Sci. Conf. (NRSC)*, Mar. 2005, pp. 277–284.
[90] S. Vaidyanathan, "Analysis, control and synchronization of hyperchaotic Zhou system via adaptive control," in *Advances in Computing and Information Technology*. Berlin, Germany: Springer, 2013, pp. 1–10.
[91] M. Alawida, A. Samsudin, J. S. Teh, and R. S. Alkhawaldeh, "A new hybrid digital chaotic system with applications in image encryption," *Signal Process.*, vol. 160, pp. 45–58, Jul. 2019.
[92] S. I. Batool and H. M. Waseem, "A novel image encryption scheme based on Arnold scrambling and Lucas series," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27611–27637, Oct. 2019.
[93] K. A. K. Patro, B. Acharya, and V. Nath, "Various dimensional colour image encryption based on non-overlapping block-level diffusion operation," *Microsyst. Technol.*, vol. 26, pp. 1–12, Nov. 2019.
[94] D. Sravanthi, K. A. K. Patro, B. Acharya, and S. Majumder, "A secure chaotic image encryption based on bit-plane operation," in *Soft Computing in Data Analytics*, Singapore: Springer, pp. 717–726, 2019.
[95] K. A. K. Patro and B. Acharya, "A simple, secure, and time-efficient bit-plane operated bit-level image encryption scheme using 1-D chaotic maps," in *Innovations in Soft Computing and Information Technology*, Singapore: Springer, 2019, pp. 261–278.
[96] K. A. K. Patro, M. P. J. Babu, K. P. Kumar, and B. Acharya, "Dual-layer DNA-encoding–decoding operation based image encryption using one-dimensional chaotic map," in *Advances in Data and Information Sciences*, Singapore: Springer, pp. 67–80, 2020.
[97] S. Bruce, *Applied Cryptography*, 2nd ed. Hoboken, NJ, USA: Wiley, 1996.

**ARSLAN SHAFIQUE** received the B.E. degree in mechatronics engineering from the Wah Engineering College, Wah Cantonment, in 2014, and the M.S. degree in electrical engineering from Heavy Industries Taxila Education City (HITEC) University, Taxila, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Engineering and Applied Sciences, Riphah International University, Islamabad, Pakistan. He has been serving as a Research Associate with the Faculty of Engineering and Applied Sciences, Riphah International University. He has five journal publications with a cumulative impact factor of 14.54. His research interests include cryptography, secure communication, and machine learning.

**JAMEEL AHMED** (Member, IEEE) received the B.E. degree in electronic engineering from the NED University of Engineering and Technology, Karachi, the M.S. degree in electrical engineering from the National University of Science and Technology, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore. Subsequently, he has carried out a Postdoctoral Fellowship twice with NTU. He is actively involved in teaching and research for the last 25 years. He is currently a Professor and the Dean of the Faculty of Engineering and Applied Sciences, Riphah International University, Islamabad. He has published more than 50 national and international research publications. In addition, he has authored four international and one national book. He is a member of NCRC and HEC, and an elected member of the Governing body of Pakistan Engineering Council.

**MUJEEB UR REHMAN** received the B.E. and M.S. (Hons.) degrees in electrical engineering from Riphah International University (RIU), Islamabad, Pakistan, in 2014 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Faculty of Engineering and Applied Sciences. He has been serving as a Lecturer with the Faculty of Engineering and Applied Sciences, RIU. His research interests include cryptography, secure communication, and machine learning. He is a certified Professional Engineer (Pakistan Engineering Council).

**MOHAMMAD MAZYAD HAZZAZI** received the Ph.D. degree in mathematics from the University of Sussex, Brighton, U.K. He is currently working as an Assistant Professor with the Department of Mathematics, King Khalid University, Abha, Saudi Arabia. His research interests include coding theory, cryptography, finite geometry, algebraic geometry, and group theory.