IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 25, 2020, accepted September 12, 2020, date of publication September 21, 2020, date of current version October 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025241

DNA Strands Level Scrambling Based Color Image Encryption Scheme

NADEEM IQBAL[1,2], MUHAMMAD HANIF[3], SAGHEER ABBAS[1], MUHAMMAD ADNAN KHAN[4], SULTAN H. ALMOTIRI[5], AND MOHAMMED A. AL GHAMDI[5], (Associate Member, IEEE)

[1]Department of Computer Science, National College of Business Administration and Economics, Lahore 54660, Pakistan
[2]School of Computing and Information Sciences, Imperial College of Business Studies, Lahore 53720, Pakistan
[3]Department of Computer Science, Bahria University, Lahore Campus, Lahore 44000, Pakistan
[4]Department of Computer Science, Lahore Garrision University, Lahore 54810, Pakistan
[5]Computer Science Department, Umm Al-Qura University, Makkah City 21421, Saudi Arabia

Corresponding author: Muhammad Adnan Khan (madnankhan@lgu.edu.pk)

This work was supported by the Data and Artificial Intelligence Scientific Chair at Umm AlQura University.

ABSTRACT In the past, many image encryption schemes have been developed through the swapping operations at the different levels of granularity. These levels span bits, Deoxyribonucleic acid (DNA) molecules, pixels, blocks of pixels. In this study, a new scheme for the encryption of color images based on the DNA strands level scrambling (DNASLS) and chaotic system has been proposed. After a color image is input, it is decomposed into the red, green and blue components. After it, these components are merged to form a big single image. Intertwining logistic map (ILM) has been used for the random data which generates three streams of random numbers. These streams have been further manipulated in such a way that nine streams are spawned out of them. One stream out of these nine streams has been used for the generation of a key image. Two streams have been used to DNA-encode the big single image and the key image. Afterwards, the strands of DNA-encoded single image are swapped with each other. Four streams determine the addresses of two DNA encoded pixels for the selection. A yet another stream is being used to select a particular strand from the DNA strands. To create the diffusion effects, an XOR operation has been done between the DNA encoded image after the swapping of strands and the DNA encoded key image. Finally, the last and ninth stream has been used to decode the DNA-encoded pixels into the decimal form. Purely random numbers with no inter-dependence have been employed in the entire encryption process. The effects of plaintext sensitivity have been achieved through the incorporation of Secure Hash Algorithm-256 or SHA-256 hash codes. In the end, the experiment and the security analysis have been performed. The results of the validation metrics like information entropy(7.9973), average key sensitivity(99.61%) and mean absolute error(84.7158) demonstrate the security, defiance to the number of attacks and a potential for real world application of the proposed image cipher.

INDEX TERMS Encryption, decryption, DNA, strands, swapping, image processing.

I. INTRODUCTION

With the widespread use of information processing gadgets and communication channels, the entire way of living of people has been changed. With this new culture, the privacy of the people is at stake. We frequently store the precious images on different gadgets like hard disks, USBs, flash drives, PDAs, cloud servers *etc*. These images have crept in the entire spectrum of human existence spannig business, research, medicine, traffic, health, government, social life to name a few. Besides, these images are also commonly shared with our associates and other acquaintances using the information highway like the internet which is not secured. Sometimes, the image to be stored or shared with someone is very sensitive and important, the design of a new missile, the picture of some spy in military or espionage setting, for instance. So, it may have grave implications if such image is hacked during its storage or transmission.

The associate editor coordinating the review of this manuscript and approving it for publication was Yonghong Peng.

Hence, measures must be taken to avoid any such untoward situation. Classically, the security of data has been ensured through the ciphers like Rivest, Shamir, and Adleman (RSA), Data Encryption Standard(DES), Advanced Encryption Standard(AES), International Data Encryption Algorithm(IDEA) *etc*. But in the context of images, unluckily, these ciphers can not be used because they were designed to encrypt textual data [1], [2]. Diametrically opposite to text, images have different characteristics like high inter-pixel relationship, high volume and redundancy. Chaotic maps/systems have proved very handy in generating the random and chaotic data for the images encryption technology. These maps have excellent qualities like extreme dependence upon the initial values and the systems parameters, mixing, ergodicity, aperiodicity *etc*. Dozens of image cryptosystems have been produced in the past using these maps [3], [3]–[26]. Although, chaotic maps have excellent properties as described above, even then many image ciphers were broken due to having different weaknesses and vulnerabilities in their design principles [27]–[29]. So more cautious and corrective measures need to be taken to avert these attacks.

A host of image cryptosystems have been produced through the combination of chaotic maps and some another instrument like 15-puzzle [30], Josephus Problem [31], knight [32], cellular automaton [33] Latin squares [34] and Latin cubes [35] to name a few. Many image ciphers were developed using the permutation and swapping operations at the different levels of granularity like bit level scrambling [18]–[23], [25], [36], pixels level scrambling [19], [24]–[26], DNA level scrambling [3], [9]–[14], block level scrambling [15]–[17]. As far as we know, no image encryption scheme based on the scrambling/swapping at DNA strand level has been tried in the literature so far.

Two types of chaotic maps exist, *i.e.*, low-dimensional and high-dimensional. Both have their pros and cons. In particular, the low-dimensional maps [37]–[39], due to their simple structure, are easy in their implementation in some programming setup but they are plagued with low chaoticity and randomness, small key space and hence poor security [40], [41] which is not suitable for the secured cryptosystems. For instance, Xiao et al [42] cracked an image cryptosystem [43] by chosen plaintext attack. In contrast to them, high-dimensional maps [10], [24], [30] render more random and chaotic data which is in line with the security demands of the ciphers, but they are hard for implementation. So, a proper balance is required to strike between these two competing needs.

DNA computing has been successfully incorporated in the realm of images cryptography due to its excellent characteristics of extra-ordinary information density, ultra-low energy consumption and massive parallelism [38], [44]. Researchers have produced a lot of image ciphers by combining the chaotic systems and DNA computing. But some image ciphers contain different defects in their design. Som *et al.* [45] wrote an RGB image cryptosystem using chaotic map and DNA encoding, for instance. In this particular image cipher, to permute the pixels'positions, an Arnold cat map was used. This map has a defect of limited number of iterations [46]. Further, the plain image must be square for the Arnold cat map which is not suitable for the rectangular images. So, a suitable map addressing both the square and rectangular images should be used. Moreover, few encryption schemes have low plaintext sensitivity in their design which led to their breakage due to the launch of chosen-plaintext and known-plaintext attacks upon them. For instance, the cipher [47] was cryptanalyzed through the launch of a chosen-plaintext attack [48]. The reason was that the chaotic data generated were not dependent upon the input plain image. To put this in other words, this algorithm suffered from the poor plaintext sensitivity which ultimately could not endure the chosen-plaintext attack. So, new ciphers with rich plaintext sensitivity are required.

Recently, an image cryptosystem through the usage of DNA computing and coupled map lattices was written by Wang *et al.* [49]. They used SHA-256 hash algorithm to create the plaintext sensitivity. In a yet another study conducted recently by [50], a color image cipher was developed through the usage of DNA operations and spatiotemporal chaotic system. In this particular scheme, the color planes were DNA encoded according to the random DNA encoding rules. These color planes were combined to make a new matrix and a DNA level scrambling was performed with the help of the matrix generated through the mixed nonlinear coupled map lattices system.

By considering the above discussion, a better level of granularity has been tried to introduce in the current study, *i.e.*, DNA strands (A, T, C, G) for the basic building block of permutation and swapping. Whatsoever ciphers we happen to explore from the literature, the DNA level confusion/scrambling/swapping was realized by making the DNA encoded pixel as the basic building block. The given cipher has more sophistication and randomization as far as the operations of confusion and diffusion are concerned since the effects of both the operations have been realized through a single action of swapping the DNA strands. Moreover, the selection of DNA encoded pixels and the DNA strands come directly from the streams of chaotic data. These streams depend upon the given plaintext image. So the proposed cipher has the high plaintext sensitivity to defy any potential differential attack which is also sometimes called as known plaintext/chosen plaintext attack.

Following lines describe the remaining portion of the paper. Section 2 discusses the basic principles upon which the current study rests, *i.e.*, the theory of chaotic system and the DNA computing. Section 3 deals with the generation of the random data and the algorithm for encryption which have been developed for the proposed cipher. Section 4 is for the experimentation of the potential encryption scheme. Security analysis has been performed in the section 5. The paper has been wrapped up in the last section 6 along with the necessary remarks of conclusion.

II. BASIC PRINCIPLES

In this section, the basic principles for the proposed image cipher will discussed a little bit.

A. CHAOTIC SYSTEM

Theory of chaos probes into the conduct of dynamical systems that are too much sensitive to the chaotic system's initial states and system parameters. On the surface, this behavior seems random, but there is a strict deterministic pattern generated by this system. According to this theory, a very minute change in either the initial states or the system parameters cause a very marked change in the output. Strictly complying with this philosophy, a plethora of chaotic systems/maps have been designed. These systems are the mathematical functions giving the stream of random numbers. We have used intertwining logistic map for this purpose [51].

$$\begin{cases} x_{n+1} = [\,\mu \times k_1 \times y_n \times (1 - x_n) + z_n]\,\mathrm{mod}1 \\ y_{n+1} = [\,\mu \times k_2 \times y_n + z_n \times 1/1 + x_{n+1}^2]\,\mathrm{mod}1 \\ z_{n+1} = [\,\mu \times (x_{n+1} + y_{n+1} + k_3) \times \sin z_n]\,\mathrm{mod}1 \end{cases} \tag{1}$$

In the above equation, $0 < \mu \leq 3.999$, $|k_1| > 33.5$, $|k_2| > 37.9$, $|k_3| > 35.7$. When this map is compared to its predecessor logistic map, it has better behavior of chaoticity. Apart from that, it has much even distribution and contains no blank windows [52].

B. DNA COMPUTING

In DNA theory, each DNA molecule comprises of four bases/strands A(Adenine), C(Cytosine), G(Guanine) and T(Thymine). There exists a complementary relationship with each other pair-wise for these bases. Specifically, if '00' is attached to C then '11' will be attached to G. In the same way, if '10' is attached to A then '01' will be attached to T. Since 4!=24, there come out a total of 24 kinds of encoding. After checking all these 24 encodings, there exist 8 such encodings which comply with the Watson-Crick complementary rules (Table 1). Apart from that, according to the research conducted in [53], some binary operations like addition, subtraction and XOR have been reported. Of these operations, this research has only used the XOR operation (Table 2). In order to incorporate DNA computing in the proposed image cipher, two conversion functions, i.e., *DNA_Encoding* and *DNA_Decoding* have been used in the algorithm. The first function will convert the given 8-bit pixel intensity value and some rule number to its DNA molecule analogue of length four and the second one will do the reverse process of the first one. In each conversion, some specific rule taken from the Table 1 will be used. For instance, $DNA\_Encoding(182, 1) = GTCG$, $DNA\_Encoding(182, 4) = AGTA$ and $DNA\_Encoding(182, 7) = GACG$. Besides, the $DNA\_Decoding(CGTA, 1) = 108$, $DNA\_Decoding(CGTA, 4) = 54$ and $DNA\_Decoding(CGTA, 7) = 99$. Apart from that, $\oplus(CATG, TCGA) = GCCG$, for DNA XOR operation.

TABLE 1. DNA rules for encoding.

Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
00:A	00:A	00:C	00:C	00:G	00:G	00:T	00:T
01:C	01:G	01:A	01:T	01:A	01:T	01:C	01:G
10:G	10:C	10:T	10:A	10:T	10:A	10:G	10:C
11:T	11:T	11:G	11:G	11:C	11:C	11:A	11:A

TABLE 2. XOR operation on DNA strands.

-	A	T	C	G
A	A	T	C	G
T	T	A	G	C
C	C	G	A	T
G	G	C	T	A

III. PROPOSED IMAGE ENCRYPTION SCHEME

This scheme encrypts an RGB image *img* with arbitrary size say $m \times n \times 3$ by using the DNA strands level scrambling. The flowchart diagram characterizing the algorithm's flow has been depicted in the Figure 1.

After the color/RGB image is input, it is broken into its three color components each of size $m \times n$. They are joined together for making a single grayscale image with size $m \times 3n$. This act of merging/concatenation bears more randomization as the DNA strands made from the pixels will be inter-blended freely facing no watertight compartments of the specific components thus boosting the security effects. SHA-256 hash codes for each given input color image has been utilized to embed the plaintext sensitivity in the proposed image cryptosystem. Each input image will have unique hash codes which will counter any potential known and chosen plaintext attack. A single pixel's change in the input image would generate a categorically different encrypted image. The hash codes generated through the SHA-256 hash function will update the variables of the chaotic map being used in the cipher's development.

The three streams of random numbers, *i.e.*, u, v and w have been generated by iterating the ILM. Through these three streams, nine keystreams have been developed due to the peculiar requirement of the algorithm. These nine streams are $row1$, $column1$, $row2$, $column2$, $selector$, $key - image$, $rule1$, $rule2$ and $rule3$. To start the encryption process, first of all, the input image img has been DNA-encoded by using the keystream $rule1$ to get the image $img1$. Further, keystream $rule2$ is used to DNA-encode key image $key - image$ for getting key image $key-image1$. The DNA-encoded image $img1$ along with the streams $row1$, $column1$, $row2$, $column2$ and $selector$ are fed into the DNA strands level scrambler (DNASLS). DNASLS randomly selects two pixels of the image $img1$ at the addresses $(row1, column1)$ and $(row2, column2)$. The keystream $selector$ selects two particular strands from these two pixels and swaps them. It is to be noted that $(row1, column1)$ and $(row2, column2)$ will select entirely two different pixels from the DNA encoded pixels. They have no relation with each other. DNASLS will iterate this process for $3mn$ times and will yield the DNA strands

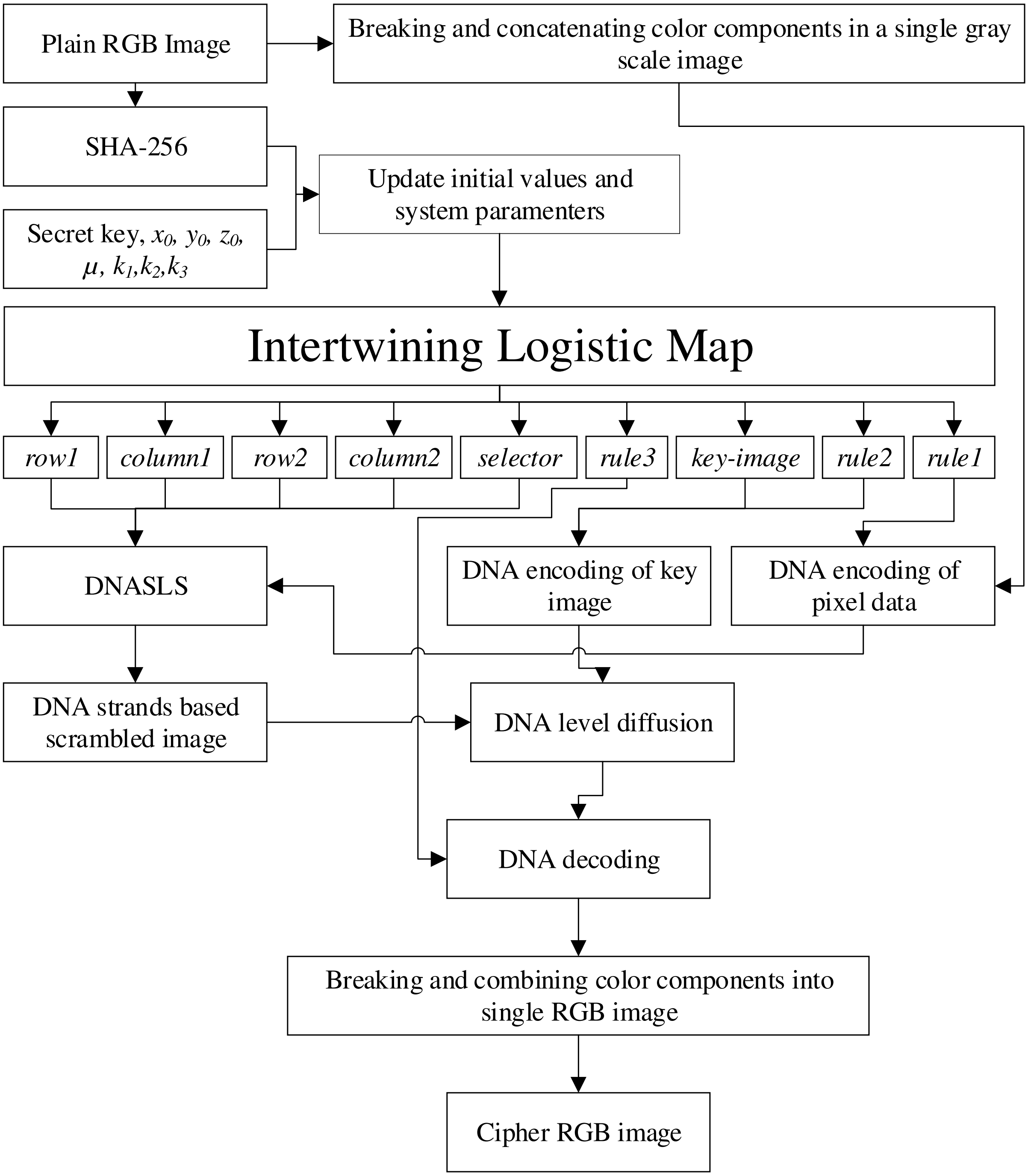

Plain RGB Image
Breaking and concatenating color components in a single gray scale image
SHA-256
Secret key, x_0, y_0, z_0, μ, k_1,k_2,k_3
Update initial values and system paramenters
Intertwining Logistic Map
row1
column1
row2
column2
selector
rule3
key-image
rule2
rule1
DNASLS
DNA encoding of key image
DNA encoding of pixel data
DNA strands based scrambled image
DNA level diffusion
DNA decoding
Breaking and combining color components into single RGB image
Cipher RGB image

FIGURE 1. **DNASLS-based encryption scheme.**

based scrambled image *img*2. Next, to produce the diffusion effects, *img*2 and DNA encoded key image $key - image1$ have been XORed as described earlier to get the image *img*3. After it, DNA encoded image *img*3 has been translated into the decimal form by using the keystream *rule*3 to get the *img*4. In the next stage, This image is broken into the three components, merged together to obtain the cryptic image *img*5. The size of *img*5 will be $m \times n \times 3$ which is same as of the original input image.

A. KEY STREAM GENERATION PROCEDURE

Plaintext sensitivity has been embedded in the potential image cipher. For this purpose, through the hashing of SHA-256 function of given plaintext image, a 256-bit secret key is obtained. In this way, for each input plain image, a different 256-bit secret key will be obtained. Through this act, a strong nexus is built between the input image and the corresponding 256-bit secret key which boosts the security effects of the potential image cipher. The slightest tempering in any of the bit of the input image will cause to produce completely different hash codes. 8-bit blocks $k_1, k_2, \ldots, k_{32}$ have been created out of the 256-bit secret key K.

Step 1: Through the following equations (2) - (8), the secret key has been tempered as follows:

$$x_0 = x_0' + \frac{((k_1 \oplus k_2) + (k_3 \oplus k_4))}{2^{12}} \tag{2}$$

$$y_0 = y_0' + \frac{((k_5 \oplus k_6) + (k_7 \oplus k_8))}{2^{12}} \tag{3}$$

$$z_0 = z_0' + \frac{((k_9 \oplus k_{10}) + (k_{11} \oplus k_{12}))}{2^{12}} \tag{4}$$

$$\mu = \mu_0' + \frac{((k_{13} \oplus k_{14}) + (k_{15} \oplus k_{16}))}{2^{12}} \tag{5}$$

$$k_1 = k_1' + \frac{((k_{17} \oplus k_{18}) + (k_{19} \oplus k_{20}))}{2^{12}} \tag{6}$$

$$k_2 = k_2' + \frac{((k_{21} \oplus k_{22}) + (k_{23} \oplus k_{24}) + (k_{25} \oplus k_{26}))}{2^{12}} \tag{7}$$

$$k_3 = k_3' + \frac{((k_{27} \oplus k_{28}) + (k_{29} \oplus k_{30}) + (k_{31} \oplus k_{32}))}{2^{12}} \tag{8}$$

In the above equations, $\oplus$ refers to the XOR operation. Further, $x_0', y_0', z_0', \mu_0', k_1', k_2', x_3'$ are the variables of the chaotic map before the sensitivity and $x_0, y_0, z_0, \mu_0, k_1, k_2, x_3$ are the variables of the chaotic map after the sensitivity.

Step 2: Through the iteration of the chaotic map 1 a number of times, *i.e.*, (n_0+3mn), we have got these streams of random numbers $u = [u_1 : u_{3mn+n0}]$, $v = [v_1 : v_{3mn+n0}]$ and $w = [w_1 : w_{3mn+n0}]$. The value of n_0 is 500 here. The first n_0 values are ignored and the remaining values have been used to avoid any kind of transient effects of the chaotic system.

Step 3: The step 2 yields chaotic sequences u, v and w which are in "raw" form. To make them useful, they are normally refined so that they may be used in the algorithm. These sequences along with the dimension of the input image (m, n) have been sent to the Algorithm 1 to generate the nine keystreams of random numbers, *i.e.*, $row1$, $column1$, $row2$, $column2$, $key-image$, $selector$, $rule1$, $rule2$ and $rule3$. Of course, these streams will be used in the algorithmic logic we have written for this image cipher.

Algorithm 1 Generation of keystreams

```
Input: u, v, w, m, n
Output: row1, column1, row2, column2, selector,
    key-image, rule1, rule2, rule3
for $i = 1$ to $3mn$ do
    $row1(i) = mod(floor(u(i) \times (10^{14}), m) + 1$
    $column1(i) = mod(floor(v(i) \times 10^{14}), 3n) + 1$
    $row2(i) = mod(floor(w(i) \times (10^{14}), m) + 1$
    $column2(i) = mod(floor(u(i) \times v(i) \times (10^{14}), 3n)$
    $+1$
    $selector(i) = mod(floor((u(i) + w(i)) \times (10^{14}), 4)$
    $+1$
    $key-image(i) = mod(floor(u(i) \times w(i) \times 10^{14}), 256)$
    $rule1(i) = mod(floor(v(i) \times w(i) \times (10^{14}), 8) + 1$
    $rule2(i) = mod(floor((u(i) + w(i)) \times (10^{14}), 8) + 1$
    $rule3(i) = mod(floor((u(i)+v(i)+w(i)) \times (10^{14}), 8)$
    $+1$
end for
```

B. IMAGE ENCRYPTION PROCEDURE

Assume the size of the input color/RGB image say img is $m \times n \times 3$. The following steps explain the potential encryption procedure in detail.

Step 1:(Breaking RGB image and concatenating components)

Break this color image img into its constituent color channels, *i.e.*, red, green and blue and concatenate these components to obtain the single grayscale image $img1$. The size of this grayscale image is $m \times 3n$.

Step 2:(DNA encoding of pixels and key image)

Reshape the pixel data $img1$ and chaotic streams $key - image$, $rule1$ and $rule2$ to $m \times 3n$. Convert $img1$ and $key - image$ into the DNA strands according to the keystreams $rule1$ and $rule2$ respectively as follows:

$$\begin{cases} img2(x, y) = DNA\_Encoding(img1(x, y), rule1(x, y)), \\ key - image1(x, y) = DNA\_Encoding(key - image(x, y) \\ \qquad\qquad , rule2(x, y)), \end{cases} \tag{9}$$

Here $x = [1 : m]$ and $y = [1 : 3n]$. Further $img2$ and $key - image1$ are the pixel data and key image in the form of DNA strands.

Step 3:(DNA strands level scrambling)

An algorithm called DNA strands scrambler has been called to randomly scramble the DNA strands as shown in the Algorithm 2. As can be seen from this algorithm, The strands $img1(row1(i), column1(i))(selector(i))$ and $img1(row2(i), column2(i))(selector(i))$ have been swapped by using the variable $temp$ for $i = 1, 2, 3, \ldots, 3mn$. Lastly, the line 6 assigns the value of $img1$ to $img2$ and returns it.

Algorithm 2 DNA strands scrambler

```
Input: img1, row1, column1, row2, column2, selector,
    m, n
Output: img2
1: for $i = 1$ to $3mn$ do
2:     $temp = img1(row1(i), column1(i))(selector(i))$
3:     $img1(row1(i), column1(i))(selector(i)) =$
       $img1(row2(i), column2(i))(selector(i))$
4:     $img1(row2(i), column2(i))(selector(i)) = temp$
5: end for
6: $img2 = img1$
```

Step 4:(DNA level diffusion)

Carry out an XOR operation between the $img2$ and $key - image1$ to get the $img3$ as follows:

$$img3(x, y) = img2(x, y) \oplus key - image1(x, y) \tag{10}$$

for $x = [1 : m]$ and $y = [1 : 3n]$. $\oplus$ represents the XOR operation.

Step 5:(Conversion of DNA strands into decimal)

Reshape the keystream *rule*3 to $m \times 3n$ and convert the DNA strands into the decimal form to get *img*4 as below.

$$img4(i, j) = DNA\_Decoding(img3(x, y), rule3(x, y)) \quad (11)$$

for $x = [1 : m]$ and $y = [1 : 3n]$.

Step 6:(Breaking and merging the image into the color image)

Finally, break the *img*4 image into individual components. Further, they are joined with each other to get the cipher color image *img*5 whose size is $m \times n \times 3$.

Because the potential cryptosystem for the images has been built upon the principle of private key, so the decryption algorithm will be an exact inverse of the steps of the encryption algorithm.

IV. EXPERIMENT

Eight color images are selected from the http://sipi.usc.edu/database/ which is called the USC-SIPI Image Database to prove the utility and the do-ability of the proposed RGB image encryption scheme. All the related experiments and simulations have been carried out in the environment of MATLAB whose specifications are: version = 2016, double-precision with 64-bit and the IEEE [54] standard 754. The above-mentioned color images(each of size of 256×256) are Lena, Baboon, Brain, Pigeon, Starfish, Tree, House and F16.

To generate the chaotic data, we have used Intertwining Logistic Map. For the proposed cipher, we have chosen these values for the initial states and the system parameters: $x_0 = 0.36$, $y_0 = 0.25$, $z_0 = 0.78$, $\mu = 1.5$, $k_1 = 35.5$, $k_2 = 38.2$, $k_3 = 36.0$. Figures 2, 3 and 4 show the original input images, ciphered images and the retrieved(decrypted) images respectively. Upon the application of the encryption algorithm, the plain images have been transformed into a very noisy form giving no hint or clue to the original information. This indicates the success of the proposed cipher scheme.

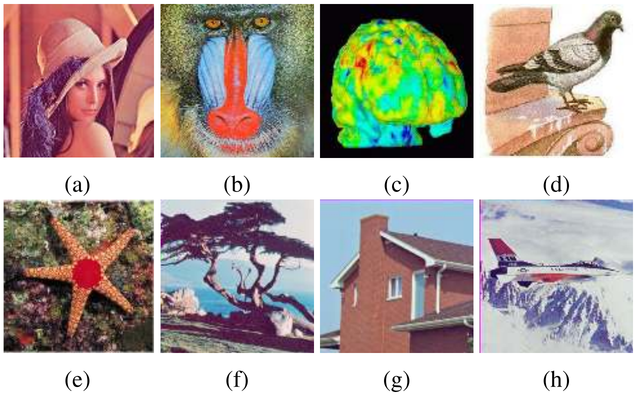


FIGURE 2. The original RGB plain images:(a) Lena; (b); Baboon; (c) Brain; (d) Pigeon; (e) Starfish; (f) Tree; (g) House; (h) F16.

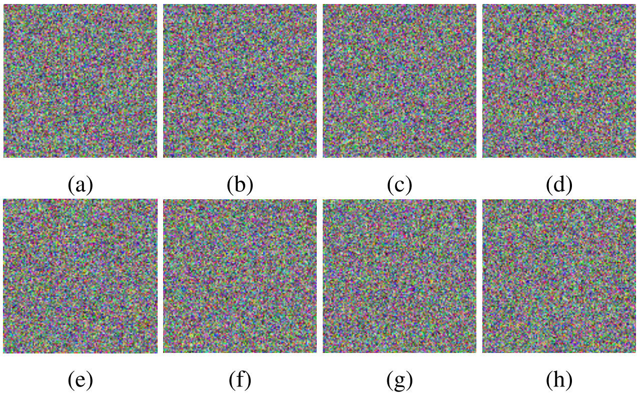


FIGURE 3. The cipher RGB images:(a) Lena; (b) Baboon; (c) Brain; (d) Pigeon; (e) Starfish; (f) Tree; (g) House; (h) F16.

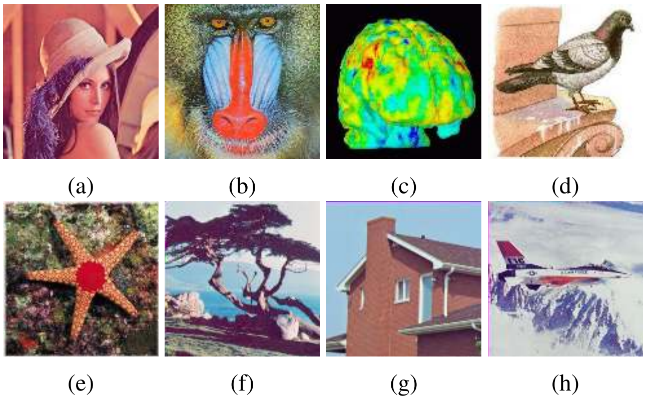


FIGURE 4. The decrypted RGB images:(a) Lena; (b); Baboon; (c) Brain; (d) Pigeon; (e) Starfish; (f) Tree; (g) House; (h) F16.

V. SECURITY AND THE PERFORMANCE ANALYSES

This section is concerned with a comprehensive security and performance analyses of this new proposed image cryptosystem.

A. KEY SPACE

In the literature, many attacks exist to reach to the original data. Brute-force attack is one of them. In this attack, hackers try to exhaust all the possible keys so that the relevant one may be discovered. This attack can be countered by enlarging the set of possible keys called key space. In principle, all the ciphers can be broken through this attack. But the key space is usually kept so large that, all the possible keys can not be checked in a practical time. The minimum threshold for the key space proposed by the scientists is 2^{100} [3]. x_0, y_0, z_0 and μ, k_1, k_2, k_3 constitute the secret key of the potential image cryptosystem. The total key space gets calculated to be $10^{105} \approx 2^{348}$ if 10^{-15} is assumed to be the computer precision of the platform in which the research work was carried out. We have taken these latest researches [3], [10], [26], [49], [55] on the image cryptography using the chaotic systems and DNA computing for the sake of comparison. Table 3 conducts a comparative analysis of the potential

TABLE 3. Key space of the potential scheme for encryption and some other existing schemes.

Algorithm	Key space
Proposed	$10^{105} \approx 2^{348}$
Ref. [49]	2.9645×10^{149}
Ref. [3]	2^{256}
Ref. [26]	3.9402×10^{185}
Ref. [55]	1.6777×10^{64}
Ref. [10]	10^{88}

encryption algorithm with the researches we just mentioned. One can see that the proposed algorithm beats [3], [10], [55] regarding the key space.

B. KEY SENSITIVITY

Any good cryptosystem is assumed to have an extreme sensitivity to the key it has. Extreme sensitivity to a key refers to the fact that upon doing a very faint tempering in one of the parameters of the key, the output should have a huge change. There are two modes of a cryptosystem, *i.e.*, encryption and decryption. This sensitivity is checked through both the modes. In the first mode, a very minute change is made in one of the parameters of the key. Then two outputs, *i.e.*, the output without making a change in the key and the output with making a change in the key are analyzed. These two outputs should be drastically different with each other for a good cryptosystem. In the decryption mode, a very small tempering is made in the secret key and the output produced should again be very different from the one which was obtained without making any change in the key.

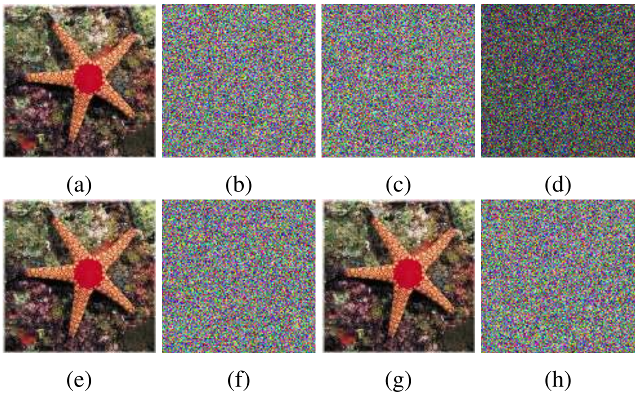


FIGURE 5. Test of key sensitivity on a selected Starfish image:(a) Input image; (b) Ciphered image through the key Key_0; (c) Ciphered image through the key Key_1; (d) Difference or differential image between (b) and (c); (e) Restored image from (b) through correct key set Key_0; (f) Restored image from (b) through wrong key set Key_1; (g) Restored image from (c) through the correct key set Key_1; (h) Restored image from (c) through wrong key set Key_0.

Figure 5 demonstrates the key sensitivity on a chosen image of Starfish for both the modes of encryption and decryption of the developed cryptosystem. Suppose the initial key set be $Key_0 = \{x_0, y_0, z_0, \mu, k_1, k_2, k_3\}$. By using this normal key, the Starfish image in the Figure 5a has been encrypted and the image so formed has been drawn in the Figure 5b. For the sake of demonstrating the key sensitivity, a microscopic tempering of 10^{-14} say (Δ) is done in just one parameter $x_0(x_0{}' = x_0 + \Delta)$ of the key Key_0. Let the new key set $\{x_0{}', y_0, z_0, \mu, k_1, k_2, k_3\}$ formed is Key_1. Through the usage of this key Key_1, the new encrypted image has been shown in the Figure 5c. Apart from the outputs of the crytosystems with minutely changed keys, Figure 5d draws the differential picture got by the two encrypted images of Figures 5b and 5c. Besides, with just a minute change of 10^{-14} in one of the parameters of the keys, there is 99.5855% difference in the pixels of these two images of Figure 5b and Figure 5c. The impact of changing the remaining parameters in the key upon the output has been shown in the Table 4. Entire results have been obtained by using the two keys, *i.e.*, Key_0 and $Key_t(t = 1, 2, \ldots 14)$. The table depicts that 99.5778% is the least rate of difference between any two cipher images which is better than [2], [56], [57]. Apart from that, the average key sensitivity is 99.61 which is equal to the [58] and better than [56], [57]. These statistics depict that the potential image cipher has better security effects.

The key sensitivity for the decryption mode has been demonstrated through the cipher images in the Figures 5b and 5c. An attempt is made to decrypt these ciphered images in the Figures 5e-5h. These figures along with their description clearly indicate that the cipher image is obtained by employing only the correct key. So, these illustrations say that the potential scheme for the image encryption enjoys a high sensitivity of the key for its encryption and decryption machineries.

C. STATISTICAL ANALYSIS

1) HISTOGRAM

Histogram of an image gives a pictorial representation of the pixels' intensity values. Histogram of any natural image has slanting/curved bar over it. This bar is replete with the information. The hacker can exploit this bar to fulfill his/her malicious intentions. So, this is the job of the cipher to refashion the pixel intensity values in such a way that the resulting histogram may have a uniform bar over it. This uniformity of the bar of a histogram is a great deterrent to any kind of leakage of the information contained by the image. Further, It is very daunting to the hackers and make their histogram attack unsuccessful. Figures 6 and 7 depict the histograms of the plain and cipher images of Lena. It can be seen from the figures that histograms have a slating bar in case image is plain, whereas they have a uniform bar for the cipher/cryptic images.

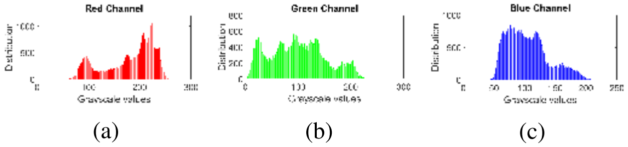
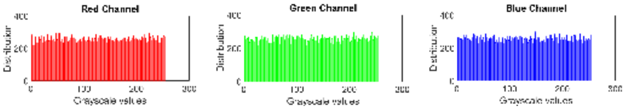


FIGURE 6. Lena image's histogram plots (a) red, (b) green, (c) blue channels.

TABLE 4. Rates of difference between two images encrypted by slightly different keys.

Secret security keys	Rates of difference(%)							
	Lena	Baboon	Brain	Pigeon	Star Fish	Tree	House	F16
$Key_1(x_0{}' = x_0 + \Delta)$	99.6043	99.6333	99.6007	99.6119	99.5855	99.6068	99.6053	99.6078
$Key_2(y_0{}' = y_0 + \Delta)$	99.6089	99.6251	99.6134	99.5900	99.6272	99.6312	99.6236	99.5962
$Key_3(z_0{}' = z_0 + \Delta)$	99.6099	99.6170	99.6134	99.6251	99.6272	99.6129	99.6023	99.5931
$Key_4(\mu = \mu + \Delta)$	99.6246	99.5962	99.6114	99.5946	99.6134	99.6414	99.6017	99.6038
$Key_5(k_1{}' = k_1 + \Delta)$	99.6226	99.5972	99.6119	99.6185	99.6063	99.5855	99.6216	99.6068
$Key_6(k_2{}' = k_2 + \Delta)$	99.6109	99.6399	99.6272	99.6185	99.5987	99.5895	99.6033	99.6129
$Key_7(k_3{}' = k_3 + \Delta)$	99.6063	99.6241	99.6460	99.5865	99.6028	99.5875	99.5987	99.5906
$Key_8(x_0{}' = x_0 - \Delta)$	99.6297	99.5936	99.5778	99.5916	99.6409	99.6236	99.6078	99.5997
$Key_9(y_0{}' = y_0 - \Delta)$	99.6058	99.6267	99.6119	99.6109	99.6134	99.5931	99.5997	99.6084
$Key_{10}(z_0{}' = z_0 - \Delta)$	99.6028	99.6109	99.5834	99.6241	99.6140	99.5875	99.5946	99.6104
$Key_{11}(\mu = \mu - \Delta)$	99.6134	99.5885	99.6068	99.5951	99.6063	99.5992	99.5911	99.5992
$Key_{12}(k_1{}' = k_1 - \Delta)$	99.6043	99.6292	99.6155	99.5941	99.5931	99.6094	99.6201	99.5911
$Key_{13}(k_2{}' = k_2 - \Delta)$	99.6206	99.5987	99.5890	99.6109	99.6053	99.6048	99.6211	99.6267
$Key_{14}(k_3{}' = k_3 - \Delta)$	99.5956	99.6043	99.6170	99.6058	99.6195	99.6206	99.6129	99.6348
Average	**99.6114**	**99.6132**	**99.6090**	**99.6055**	**99.6110**	**99.6066**	**99.6074**	**99.6058**
Average of all	**99.61**	-	-	-	-	-	-	-



(a) (b) (c)

FIGURE 7. Cipher image's histograms of Lena (a) red, (b) green, (c) blue channels.

Scientists have introduced a metric called variance to compute the uniformity of the histograms. Its formula is

$$var(X) = \frac{1}{N^2} \sum_{a=1}^{N} \sum_{b=1}^{N} \frac{1}{2}(x_a - x_b)^2 \qquad (12)$$

In the above equation, $X = \{x_0, x_1, x_2, \ldots, x_{255}\}$ refers to the vector of the histogram values. Besides, x_a and x_b are the numbers of pixels whose gray values are equal to a and b respectively. N represents the number of gray levels. The ciphers producing images with a low variance values of the histograms are better [59], [60]. Table 5 shows the variances of the histograms of the cryptic Lena, Baboon, Brain, Pigeon, Starfish, Tree, House and F16 images. The first row of the table shows the results of the variances through the usage of key set Key_0. Further, the keys Key_t $(t = 1, 2, \ldots 14)$ defined in the Section V-B have been used to calculate the variance values in the other rows. The average variance value has come out to be 254.8626, while for the encrypted Lena image, it is 238.4063 which is better than 264.37 [26]. Whereas the variance for the histogram of the plain Lena image is 59,640.

2) CORRELATION ANALYSIS

This analysis is normally done for the second statistical attack. In normal images, the pixels are highly interrelated with each other. This is the reason that these normal images are sensible to the human eye. One of the main tasks of any image cryptosystem is to disturb this inter-pixel relationship so that the resultant image may lose this sense. Upto what extent, the objective has been achieved after the cipher algorithm get applied to the normal image is checked through this metric. The following mathematical formula is utilized to find the correlation coefficient among the pixels(CC) [30]:

$$CC = \frac{Z\sum_{l=1}^{Z}(r_l \times s_l) - \sum_{l=1}^{Z} r_l \times \sum_{l=1}^{Z} s_l}{\sqrt{\left(Z\sum_{l=1}^{Z} r_l^2 - \left(\sum_{l=1}^{Z} r_l\right)^2\right)\left(Z\sum_{l=1}^{Z} s_l^2 - \left(\sum_{l=1}^{Z} s_l\right)^2\right)}} \qquad (13)$$

This equation contains some variables. Here we will describe them. r and s are the color values of two consecutive/adjacent pixels and Z represents the pixels' numbers in the given image. Besides, the consecutive pixels form a correlation distribution in the three directions. These distributions have been demonstrated through the Figure 8. These directions include the horizontal, vertical and diagonal for both the original/input image and the cryptic Lena image.

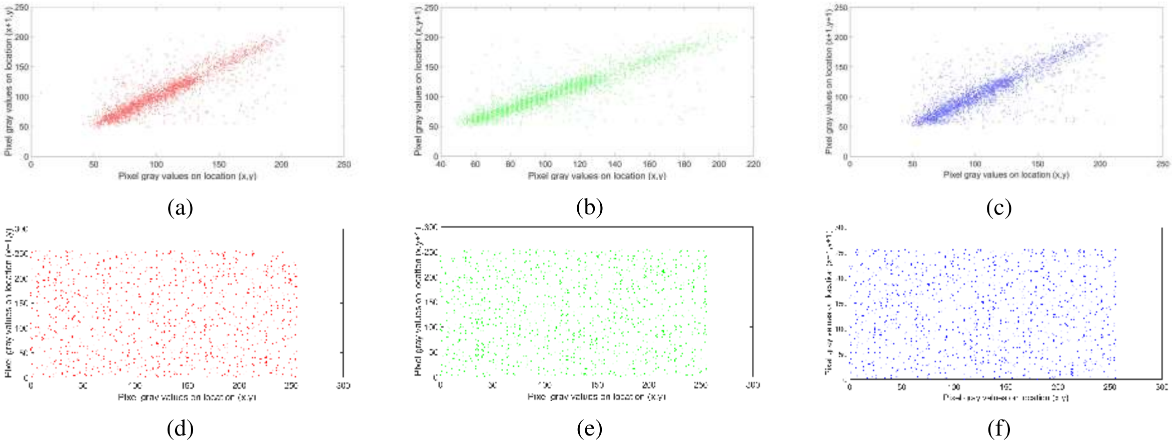
The correlation coefficients between two adjacent/ consecutive pixels for the plain Lena image and its corresponding ciphered version are shown in the Table 6. The 6 shows that the value of CC for the plain image is near to 1, while this value is near to 0 in case of the cipher image. Further Table 6 and Figure 8 jointly depict that the potential relationing between the adjacent/pixel positions become marginalized showing the success of the proposed cipher. Besides, a comparison has also been done between the proposed cipher and other related ciphers in the Table 7.

D. INFORMATION ENTROPY ANALYSIS

Information entropy mathematically characterizes the randomness and unpredictability of some signal. Its formula was

TABLE 5. Histograms' variance values for the cipher-images through the usage of different keys.

Secret security keys	Difference rates(%)									
	Lena	Baboon	Brain	Pigeon	Star Fish	Tree	House	F16	**Average**	**Average of all**
Key_0	238.4063	250.7630	249.6589	252.9609	235.9219	245.3516	265.5677	236.5391	**246.8962**	**254.8626**
Key_1	257.5651	241.0911	257.0339	249.5339	255.0990	234.7630	280.1589	241.5104	**252.0944**	
Key_2	254.5339	242.9948	254.5130	244.4688	270.1563	230.5521	262.7057	252.2839	**251.5261**	
Key_3	268.5521	251.0807	265.5156	250.8646	267.8958	255.7813	244.9948	245.8724	**256.3197**	
Key_4	279.3906	258.8776	247.8880	211.7057	232.3438	270.4323	253.0234	250.6016	**250.5329**	
Key_5	260.7344	246.8359	253.3854	276.1016	243.0625	262.7578	245.9974	260.2630	**256.1423**	
Key_6	244.7578	265.0286	270.1667	254.0911	270.8047	257.2083	244.5286	250.8906	**257.1846**	
Key_7	251.3958	232.0052	245.3125	260.4063	260.1432	272.3542	282.0339	245.6016	**256.1566**	
Key_8	243.2266	242.2422	227.4401	253.5417	264.2188	239.5599	249.6380	246.8958	**245.8454**	
Key_9	259.5938	266.6198	275.7786	267.5026	244.2266	257.1693	267.3464	249.5755	**260.9766**	
Key_{10}	270.8516	259.3073	258.5391	241.3229	253.3880	284.5208	264.6380	255.7109	**261.0348**	
Key_{11}	273.0703	262.7448	258.2552	273.9427	271.2630	247.9505	263.0078	261.1354	**263.9212**	
Key_{12}	267.4635	256.5130	266.6641	261.0469	234.3620	246.3958	269.2161	232.5703	**254.2790**	
Key_{13}	236.7135	234.7969	263.8932	258.0182	240.9089	270.7161	247.0729	252.8620	**250.6227**	
Key_{14}	252.5339	246.7682	266.1953	281.8750	282.5443	248.3125	250.5990	246.4193	**259.4059**	



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 8. Correlation distribution of consecutive pixels for the Lena image(direction, component, image): (a) horizontal, red component, plain image; (b) vertical, green component, plain image; (c) diagonal, blue component, plain image; (d) horizontal, red component, cipher image ;(e) vertical, green component, cipher image; (f) diagonal, blue component, plain image.

TABLE 6. Correlation coefficient for input Lena image and its encrypted version.

Image	Correlation orientation/direction		
	Horizontal	Vertical	Diagonal
Original Lena image	0.9172	0.9516	0.8941
Encrypted Lena image	0.0042	0.0087	-0.0031

developed by an information theorist Shannon [61] in 1949.

$$IE(q) = \sum_{b=0}^{2^n-1} p(q_b) log \frac{1}{p(q_b)} \tag{14}$$

where IE is the information entropy. The above equation calculates the information entropy when q is given as an information source. Further, $p(q_b)$ gives the probability of q_b. The maximum value of the entropy for the given 256 gray levels comes out to be 8. So this serves as an ideal value for the information entropy. The more close the value of information entropy to 8 is, it is always better. Table 8 gives the entropies' values for the pictures of the potential scheme. One can see that the average value is very near to 8, which is an ideal value in this context. Therefore, the proposed scheme is robust against the entropy attack. A comparison has been drawn in the Table 8. One can see that the proposed cryptosystem does better than [3], [10], [26], [49], [55] as far as entropy is considered for the Lena image.

E. DIFFERENTIAL ATTACK

Hackers have a lot of schemes in their ''ammunition'' to crack the image cipher. Differential attack is one of them. In this particular attack, two samples of cipher images are obtained. The first one through a straightforward way whereas the second one by changing a intensity value for some randomly selected pixel in the given input image. Through an ingenious manipulation, the potential relationship between the plain and ciphered images can be figured out which ultimately lead to the revelation of the key. To combat this attack, two measures are normally used by the research community. The first measure is called number of pixels change rate (NPCR), while the second measure is usually called as unified average changing intensity(UACI). These two measures are calculated on the plain input image and the cipher output image.

TABLE 7. Correlation coefficients of the proposed method for the adjacent pixels and others.

Image	Encryption scheme	Correlation orientation/direction		
		Horizontal	Vertical	Diagonal
Original Lena image		0.9172	0.9516	0.8941
Encrypted Lena image	the proposed scheme	0.0042	0.0087	-0.0031
Lena	Ref. [49]	-0.0021	0.0009	0.0003
Lena	Ref. [3]	0.0012	0.0031	0.0060
Lena	Ref. [26]	-0.0027	0.0033	-0.0035
Lena	Ref. [55]	0.0082	-0.0032	-0.0025
Lena	Ref. [10]	-0.0082	-0.0128	-0.0012

TABLE 8. The information entropy analysis' results.

Encryption schemes	Images	Size	Plain	Ciphered
Proposed algorithm	Lena	256 × 256	7.5954	7.9974
	Baboon	256 × 256	6.9730	7.9972
	Brain	256 × 256	7.0097	7.9972
	Pigeon	256 × 256	6.4523	7.9972
	Starfish	256 × 256	6.7093	7.9974
	Tree	256 × 256	6.7057	7.9973
	House	256 × 256	7.3118	7.9971
	F16	256 × 256	7.3424	7.9974
	Average		**7.0125**	**7.9973**
Ref. [49]	Lena	256 × 256	7.5788	7.9972
Ref. [3]	Lena	256 × 256		7.9972
Ref. [26]	Lena	256 × 256		7.9971
Ref. [55]	Lena	256 × 256		7.9880
Ref. [10]	Lena	256 × 256		7.9896

Their mathematical formulas are

$$NPCR = \frac{\sum_{p,q} S(p,q)}{M \times N} \times 100\% \tag{15}$$

where M and N represent the width and height of the image respectively. $S(p,q)$ can be defined by:

$$S(p,q) = \begin{cases} 1, & \text{if } \ E(p,q) \neq E'(p,q); \\ 0, & \text{if } \ E(p,q) = E'(p,q). \end{cases} \tag{16}$$

$$UACI = \frac{1}{M \times N}\left[\sum_{p,q} \frac{|E(p,q) - E'(p,q)|}{255}\right] \times 100\% \tag{17}$$

E and E' are respectively the cryptic/output images before and after one pixel of the plain image is changed.

Both the $NPCR$ and $UACI$ values of the chosen eight images have been shown in Table 9. The averages of $NPCR$ and $UACI$ for the selected images are 99.6027% and 33.4908% respectively which expressly prove that the potential encryption scheme is sufficiently potent to withstand the attacks of $NPCR$ and $UACI$ which are differential in character. Further Table 10 draws a comparison of the metrics of the proposed scheme for the Lena image with some other algorithms [3], [10], [26], [49], [55]. Clearly, the proposed method has the better values of $NPCR$ and $UACI$ than [49] and [3], [49], [55] respectively.

F. PEAK SIGNAL-TO-NOISE RATIO ANALYSIS

A lot of metrics exist to gauge the efficiency and performance of the ciphers. Peak signal-to-noise ratio is one of them.

TABLE 9. Average NPCR and UACI values for different images.

Images	NPCR(%)	UACI(%)
Lena	99.6063	33.4681
Baboon	99.6017	33.4731
Brain	99.5870	33.3495
Pigeon	99.6023	33.5240
Starfish	99.5921	33.5188
Tree	99.6134	33.5107
House	99.6073	33.5202
F16	99.6114	33.5617
Average	**99.6027**	**33.4908**

TABLE 10. The average values of NPCR and UACI on the Lena image for the potential algorithm and some others.

Algorithm	Image	NPCR(%)	UACI(%)
Proposed algorithm	Lena	99.6063	33.4681
Ref. [49]	Lena	99.5956	33.4588
Ref. [3]	Lena	99.6101	30.3489
Ref. [26]	Lena	99.6067	33.5000
Ref. [55]	Lena	99.6150	33.4205
Ref. [10]	Lena	99.6090	33.4727

It aims at measuring the difference/discrepancy between the original plain image and the encrypted output image. Mathematically it is defined as

$$\begin{cases} PSNR = 20log_{10}(255/\sqrt{MSE})dB \\ MSE = \dfrac{1}{M \times N} \sum_{p=1}^{M} \sum_{q=1}^{N} (P_0(p,q) - P_1(p,q))^2 \end{cases} \tag{18}$$

where (M, N) is the dimensions of the test image. $P_0(p,q)$ and $P_1(p,q)$ are the pixel values of the plain/original and encrypted images respectively. Further, MSE is the mean squared error between the original image and the encrypted image. The larger value of MSE is always desirable. Since MSE and $PSNR$ are reciprocally interrelated, so the larger value of MSE will lead to the smaller value of $PSNR$. Table 11 shows the PSNR value obtained by the proposed cipher and some of the chosen ones. One can see the infinite (∞) value in the first row of the table. This infinite value signals that there is not a difference of single pixel intensity value between the two images. These two images are plain image and the decrypted image. This caused due to the value of zero for MSE lying in the denominator for the formula of $PSNR$. These phenomena also refer that the proposed image cryptosystem is lossless. The $PSNR$ value for the image of Lena given by the proposed algorithm is better than [62]–[64] as given by the Table 11.

TABLE 11. The Peak signal-to-noise ratio results between the original images and corresponding ciphered/decrypted images: Here 'O-C': original and ciphered images, and 'O-D': the original and decrypted images.

		Lena	Baboon	Brain	Pigeon	Starfish	Tree	House	F16	**Average**
Proposed algorithm	PSNR (O-D)	∞	∞	∞	∞	∞	∞	∞	∞	
	PSNR (O-C)	8.6696	8.7890	5.9946	6.9128	8.0443	8.1740	8.9554	7.9830	**7.9403**
Ref. [62]	PSNR (O-D)	96.2956								
	PSNR (O-C)	9.0348								
Ref. [63]	PSNR (O-C)	8.6878								
Ref. [64]	PSNR (O-C)	9.0486								
Ref. [10]	PSNR (O-C)	8.1300								7.8720

TABLE 12. The *MAE* results.

Image	MAE		
	Red	Green	Blue
Lena	84.1188	78.2947	70.5952
Baboon	76.2810	72.9998	79.8142
Brain	104.6149	107.0394	107.9894
Pigeon	102.9192	91.4679	89.5854
Starfish	80.6161	82.5385	84.5347
Tree	77.0134	86.9337	80.7303
House	69.6062	76.2653	79.9348
F16	81.3689	84.3369	83.5803
Average for each component	**84.5673**	**84.9845**	**84.5955**
Average for all images		**84.7158**	
Ref. [26]		89.4033	
Ref. [65]		80.2	
Ref. [63]		82.8419	
Ref. [10]		84.4922	

G. MEAN ABSOLUTE ERROR (MAE)

Creating a maximum discrepancy between the plain input image and the ciphered output image is one of the recurrent themes of any image cryptosystem. To measure it, a test metric called Mean absolute error (MAE) is utilized in this regard. Its mathematical formulation is:

$$MAE_{R,G,B} = \frac{1}{M \times N} \sum_{p=1}^{M} \sum_{q=1}^{N} |C_{R,G,B}(p,q) - P_{R,G,B}(p,q)| \tag{19}$$

In the above equation, P is the plain image and C is the cipher image. (M, N) is the dimension of the image. The relatively larger value of MAE is desirable for the security effects. Table 12 depicts the results of this metricMAE given by the proposed algorithm. Besides, this table also compares our result with some other researches. The proposed cipher performed better than [10], [63], [65].

H. NOISE AND DATA CROP ATTACKS

Things are not as straightforward as they sound. Sometimes, either during storage or during transmission, the encrypted image may subject to noise attack due to which it is corrupted. Now the decryption algorithm should be robust enough to restore the original image in a way that it may be easily recognized. Further, occasionally, some portion of the ciphered image may be damaged due to the crop/data loss attack on it. Again, the decryption algorithm should decode the image in such a way that the original input image may be recognized. Figures 9a to 9d show the encrypted images contaminated by Pepper & Salt noise with varying noise densities, *i.e.*, 0.1, 0.2, 0.3 and 0.4 and Figures 9e to 9h draw the corresponding decrypted images through the usage of the proposed scheme. The original visual content can be easily recognized.

Besides, Figures 10a to 10b draw the ciphered image of Lena and Baboon with data loss attacks of 170 × 80 and $\frac{1}{2} \times 128 \times 128$ respectively. Later on, the decryption algorithm has been implemented to these damaged ciphered images. Figures 10c to 10d plot the corresponding decrypted images. Obviously, the restored images can be easily discerned. Given these results, we are justified to say that the proposed scheme is robust enough to withstand the noise and data loss attacks and bears promise for some real world setting.

I. CONTRAST AND ENERGY ANALYSIS

Generally speaking, contrast analysis renders a measure of local intensity variations existing in an image. Put in other words, through this metric, heterogeneity of the pixels of an image is gauged. The relatively higher values of the contrast mirror that the image has abundantly different gray levels which is a good security effect. Contrast of an image is defined as [66]

$$C = \sum_{a,b} |i-j|^2 \times p(a,b) \tag{20}$$

In the above equation, $p(a, b)$ is the number of gray-level co-occurrence matrices (GLCM). Contrast of both the plain image and the cipher image has been calculated and the corresponding results have been displayed in the Table 13. The value of the contrast for the Lena image is 10.5112(average of red, green and blue channels) which is better than 10.4511 [67] and 8.4156 [66]. These signs indicate the relatively better security effects of the proposed cipher as compared to the existing ones.

Apart from that, energy of an image gives the sum of the squared elements in the gray level co-occurrence matrix given as [66]

$$E = \sum_{a,b} p(a,b)^2 \tag{21}$$

where $p(a, b)$ is the number of gray-level co-occurrence matrices. Table 14 portrays the energy analyses results for both the plain and cipher images. One can see that the results

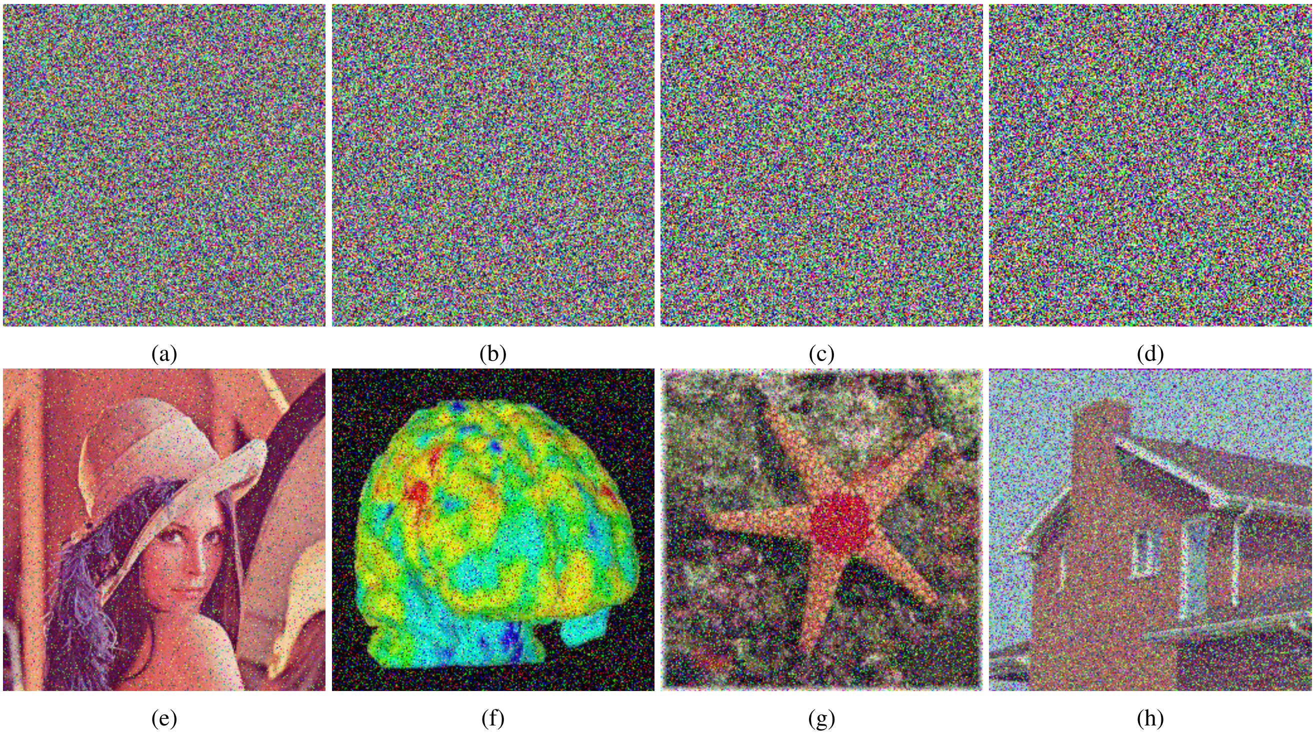


FIGURE 9. Pepper & Salt noise attack with image type and noise density:(a) (Ciphered Lena image, 0.1); (b) (Ciphered Brain image, 0.2); (c) (Ciphered Starfish image, 0.3); (d) (Ciphered House image, 0.4); (e) (Decrypted image from (a,0.1); (f) (Decrypted image from (b, 0.2); (g) (Decrypted image from (c, 0.3); (h) (Decrypted image from (d, 0.4).

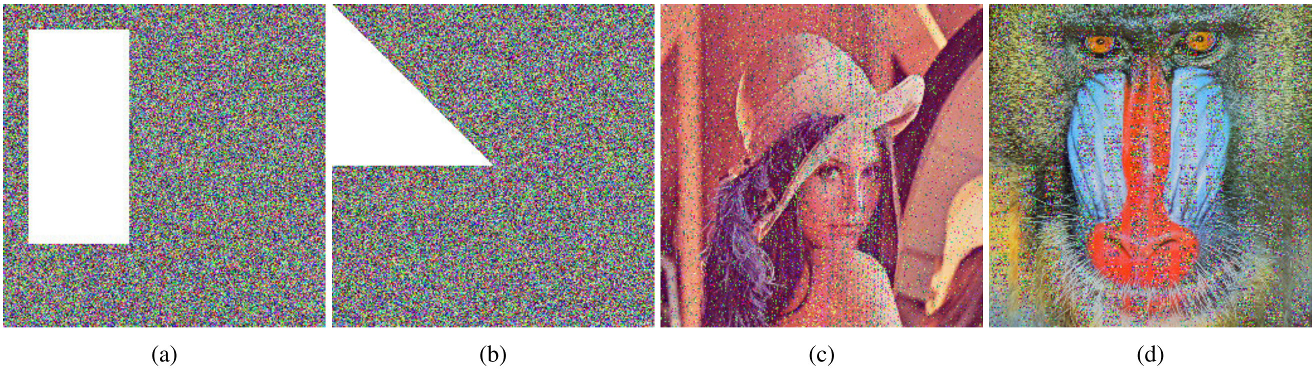


FIGURE 10. Data loss attack with image and data loss dimension: (a)(Encrypted Lena image, 170 × 80); (b) (Encrypted Baboon image, $\frac{1}{2}$ × 128 × 128); (c) Decrypted Lena image from (a); (d) Decrypted Baboon image from (b).

TABLE 13. The contrast analysis results.

Image	Plaintext			Ciphertext		
	Red	Green	Blue	Red	Green	Blue
Lena	0.7594	1.0527	0.6394	10.5239	10.4356	10.5741
Baboon	1.3652	1.9110	1.9198	10.5064	10.5076	10.3981
Brain	0.6941	0.6156	0.4913	10.5798	10.5520	10.5036
Pigeon	0.7241	0.8274	0.8688	10.5151	10.5145	10.5491
Starfish	1.3574	1.3693	1.3128	10.6238	10.5214	10.4815
Tree	0.6706	1.0472	0.6888	10.4105	10.5094	10.5149
House	0.3440	0.4337	0.4886	10.5276	10.5142	10.5366
F16	0.7556	1.0502	0.3518	10.5793	10.4686	10.5625
Average for each component	**0.8338**	**1.0384**	**0.8452**	**10.5333**	**10.5029**	**10.5151**
Average for all images		**0.9058**			**10.5171**	

TABLE 14. The energy analysis results.

Image	Plaintext			Ciphertext		
	Red	Green	Blue	Red	Green	Blue
Lena	0.1166	0.0699	0.1402	0.0156	0.0156	0.0156
Baboon	0.0477	0.0464	0.0412	0.0156	0.0156	0.0156
Brain	0.3286	0.2192	0.4219	0.0156	0.0156	0.0156
Pigeon	0.2599	0.1342	0.1175	0.0156	0.0156	0.0156
Starfish	0.0410	0.0549	0.0756	0.0156	0.0156	0.0156
Tree	0.1559	0.0989	0.1872	0.0156	0.0156	0.0156
House	0.1723	0.1971	0.1687	0.0156	0.0156	0.0156
F16	0.2910	0.3157	0.4341	0.0156	0.0156	0.0156
Average for each component	**0.1766**	**0.1420**	**0.1983**	**0.0156**	**0.0156**	**0.0156**
Average for all images		**0.1723**			**0.0156**	

TABLE 15. Speed analysis of the proposed algorithm.

Algorithm	Image	Speed(sec)
Proposed	Lena	5.9328
	Baboon	5.6520
	Brain	5.4667
	Pigeon	5.8088
	Starfish	5.7061
	Tree	5.8412
	House	5.9912
	F16	5.8042
	Average	**5.7754**
Ref. [68]	Lena	6.3849
Ref. [70]	Lena	0.1635
Ref. [71]	Lena	2.2149
Ref. [69]	Lena	20.9158

of the plaintext images are relatively higher than their ciphertext counterparts. Besides the average energy value for the three components for all the images is 0.0156 which is better than 0.0176 [66].

J. SPEED AND COMPLEXITY ANALYSIS

The proposed algorithms of encryption and decryption have been written and compiled under the Intel(R) Core(TM) i5-4210U CPU @ 1.70 GHz 2.40 GHz, 8 GB memory. Besides, the platform selected for this work is Windows 10 and MATLAB R2016a. Complexity analysis refers to the computational time an algorithm takes. Normally two types of analyses exist, *i.e.*, the empirical analysis and the theoretical analysis. The former analysis directly measures the time taken by the algorithm through some gadget like stopwatch. Table 15 gives the time taken for encryption by the different images. The time taken by the Lena image and the average for all the eight images is 5.9328 sec and 5.7754 sec respectively which is better than the [68] and [69].

As far as the theoretical analysis is concerned, a number of factors like input, compiler, software, hardware *etc*. eclipse its universality. The latter analysis forgoes these limitations and carries us to such a setting which renders a pure computational performance of the given algorithm. Inspired by this fact, we have resorted to theoretical analysis only. Asymptotics [72] - a mathematical theory, is normally employed to carry out this analysis. The complexity for iterating the ILM is $\Theta(9MN)$ where (M, N) is the dimension of the input image. The complexity of DNA encoding for the pixel data and the key image is $\Theta(3MN)$ each. The complexity for the swapping operations for the DNA strands is $\Theta(3MN)$. Morever $\Theta(3MN)$ is the complexity for the DNA diffusion. Lastly, the complexity for DNA to decimal conversion is $\Theta(3MN)$. The total complexity comes out to be $\Theta(24MN)$ which is equal to [10], [26] and better than $O(64N^2)$ [73].

VI. CONCLUSION

By treating the single DNA strand as a basic unit for scrambling, a novel image encryption scheme has been developed in this study. Plethora of image ciphers exist based on the theories of chaos and DNA computing. The novelty of the proposed scheme lies in the fact that a greater degree of sophistication and randomness has been thrown in the proposed image cipher by lowering the level of granularity to the DNA strand. In the previous ciphers, the basic unit of scrambling has been a whole DNA molecule which is relatively bigger than a DNA strand. Apart from that, SHA-256 hash codes have been used to embed the plaintext sensitivity in the cipher. Four dedicated and independent streams of random numbers are for the selection of two DNA encoded molecules. The fifth independent stream of random number is for the selection of a particular DNA strand which is to swapped between these two selected DNA molecules. This act of swapping served both the purposes of confusion and diffusion. Besides, only a single round of XOR operation between the pixel data after the swapping of the strands and the DNA encoded key image has been carried out. Both the experimentation and the security analyses of the proposed cipher gave very promising results which indicate the robustness, good security effects, defiance to the different threats and a potential for a real world application of the proposed scheme.

REFERENCES

[1] 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.

[2] Z. Parvin, H. Seyedarabi, and M. Shamsi, "A new secure and sensitive image encryption scheme based on new substitution with chaotic function," *Multimedia Tools Appl.*, vol. 75, no. 17, pp. 10631–10648, 2016.

[3] S. Suri and R. Vijay, "A synchronous intertwining logistic map-DNA approach for color image encryption," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 6, pp. 2277–2290, Jun. 2019.

[4] 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.

[5] X. Wang and L. Liu, "Image encryption based on hash table scrambling and dna substitution," *IEEE Access*, vol. 8, pp. 68533–68547, 2020.

[6] F. Masood, J. Ahmad, S. A. Shah, S. S. Jamal, and I. Hussain, "A novel hybrid secure image encryption based on Julia set of fractals and 3D Lorenz chaotic map," *Entropy*, vol. 22, no. 3, p. 274, Feb. 2020.

[7] J. Ahmad, F. Masood, S. A. Shah, S. S. Jamal, and I. Hussain, "A novel secure occupancy monitoring scheme based on multi-chaos mapping," *Symmetry*, vol. 12, no. 3, p. 350, Mar. 2020.

[8] J. Ahmad, A. Tahir, J. S. Khan, M. A. Khan, F. A. Khan, Arshad, and Z. Habib, "A partial ligt-weight image encryption scheme," in *Proc. U.K./China Emerg. Technol. (UCET)*, Aug. 2019, pp. 1–3.

[9] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu, "A novel image encryption scheme based on DNA sequence operations and chaotic systems," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 219–237, Jan. 2019.

[10] X. Wu, K. Wang, X. Wang, H. Kan, and J. Kurths, "Color image DNA encryption using NCA map-based CML and one-time keys," *Signal Process.*, vol. 148, pp. 272–287, Jul. 2018.

[11] T. Hu, Y. Liu, L.-H. Gong, and C.-J. Ouyang, "An image encryption scheme combining chaos with cycle operation for DNA sequences," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 51–66, Jan. 2017.

[12] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, Jan. 2017.

[13] X.-Y. Wang, Y.-Q. Zhang, and X.-M. Bao, "A novel chaotic image encryption scheme using DNA sequence operations," *Opt. Lasers Eng.*, vol. 73, pp. 53–61, Oct. 2015.

[14] Q. Zhang, X. Xue, and X. Wei, "A novel image encryption algorithm based on DNA subsequence operation," *Sci. World J.*, vol. 2012, pp. 1–10, Jan. 2012.

[15] L. Xu, X. Gou, Z. Li, and J. Li, "A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion," *Opt. Lasers Eng.*, vol. 91, pp. 41–52, Apr. 2017.

[16] A. ur Rehman, D. Xiao, A. Kulsoom, M. A. Hashmi, and S. A. Abbas, "Block mode image encryption technique using two-fold operations based on chaos, MD5 and DNA rules," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 9355–9382, Apr. 2019.

[17] X. Chai, Z. Gan, and M. Zhang, "A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion," *Multimedia Tools Appl.*, vol. 76, no. 14, pp. 15561–15585, 2017.

[18] L. Teng, X. Wang, and J. Meng, "A chaotic color image encryption using integrated bit-level permutation," *Multimedia Tools Appl.*, vol. 77, no. 6, pp. 6883–6896, Mar. 2018.

[19] S. Sun, "A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling," *IEEE Photon. J.*, vol. 10, no. 2, pp. 1–14, Apr. 2018.

[20] F. Li, H. Wu, G. Zhou, and W. Wei, "Robust real-time image encryption with aperiodic chaotic map and random-cycling bit shift," *J. Real-Time Image Process.*, vol. 16, no. 3, pp. 775–790, Jun. 2019.

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

[22] X. Zhang, F. Han, and Y. Niu, "Chaotic image encryption algorithm based on bit permutation and dynamic DNA encoding," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–11, Aug. 2017.

[23] S. F. Raza and V. Satpute, "A novel bit permutation-based image encryption algorithm," *Nonlinear Dyn.*, vol. 95, no. 2, pp. 859–873, Jan. 2019.

[24] H. R. Amani and M. Yaghoobi, "A new approach in adaptive encryption algorithm for color images based on dna sequence operation and hyper-chaotic system," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 21537–21556, 2019.

[25] 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.

[26] X. Chai, X. Fu, Z. Gan, Y. Lu, and Y. Chen, "A color image cryptosystem based on dynamic DNA encryption and chaos," *Signal Process.*, vol. 155, pp. 44–62, Feb. 2019.

[27] C. Li, "Cracking a hierarchical chaotic image encryption algorithm based on permutation," *Signal Process.*, vol. 118, pp. 203–210, Jan. 2016.

[28] C. Li, D. Lin, and J. Lu, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE MultimediaMag.*, vol. 24, no. 3, pp. 64–71, Aug. 2017.

[29] Y. Liu, H. Fan, E. Y. Xie, G. Cheng, and C. Li, "Deciphering an image cipher based on mixed transformed logistic maps," *Int. J. Bifurcation Chaos*, vol. 25, no. 13, Dec. 2015, Art. no. 1550188.

[30] N. Iqbal, S. Abbas, M. A. Khan, T. Alyas, A. Fatima, and A. Ahmad, "An RGB image cipher using chaotic systems, 15-puzzle problem and dna computing," *IEEE Access*, vol. 7, pp. 174051–174071, 2019.

[31] Z. Hua, B. Xu, F. Jin, and H. Huang, "Image encryption using Josephus problem and filtering diffusion," *IEEE Access*, vol. 7, pp. 8660–8674, 2019.

[32] J. Xiaoyong, B. Sen, Z. Guibin, and Y. Bing, "Image encryption and compression based on the generalized knight's tour, discrete cosine transform and chaotic maps," *Multimedia Tools Appl.*, vol. 76, no. 10, pp. 12965–12979, 2017.

[33] A. Babaei, H. Motameni, and R. Enayatifar, "A new permutation-diffusion-based image encryption technique using cellular automata and DNA sequence," *Optik*, vol. 203, Feb. 2020, Art. no. 164000.

[34] M. Xu and Z. Tian, "A novel image encryption algorithm based on self-orthogonal Latin squares," *Optik*, vol. 171, pp. 891–903, Oct. 2018.

[35] M. Xu and Z. Tian, "A novel image cipher based on 3D bit matrix and Latin cubes," *Inf. Sci.*, vol. 478, pp. 1–14, Apr. 2019.

[36] M. Hanif, R. A. Naqvi, S. Abbas, M. A. Khan, and N. Iqbal, "A novel and efficient 3D multiple images encryption scheme based on chaotic systems and swapping operations," *IEEE Access*, vol. 8, pp. 123536–123555, 2020.

[37] A. Jain and N. Rajpal, "A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps," *Multimedia Tools Appl.*, vol. 75, no. 10, pp. 5455–5472, May 2016.

[38] M. Babaei, "A novel text and image encryption method based on chaos theory and DNA computing," *Natural Comput.*, vol. 12, no. 1, pp. 101–107, Mar. 2013.

[39] Q. Zhang, L. Liu, and X. Wei, "Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps," *AEU-Int. J. Electron. Commun.*, vol. 68, no. 3, pp. 186–192, Mar. 2014.

[40] R. Enayatifar, A. H. Abdullah, and I. F. Isnin, "Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence," *Opt. Lasers Eng.*, vol. 56, pp. 83–93, May 2014.

[41] X. Wang and C. Liu, "A novel and effective image encryption algorithm based on chaos and DNA encoding," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6229–6245, Mar. 2017.

[42] H. Wang, D. Xiao, X. Chen, and H. Huang, "Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map," *Signal Process.*, vol. 144, pp. 444–452, Mar. 2018.

[43] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," *Signal Process.*, vol. 138, pp. 129–137, Sep. 2017.

[44] L. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, Nov. 1994.

[45] S. Som, A. Kotal, A. Chatterjee, S. Dey, and S. Palit, "A colour image encryption based on DNA coding and chaotic sequences," in *Proc. 1st Int. Conf. Emerg. Trends Appl. Comput. Sci.*, Sep. 2013, pp. 108–114.

[46] X. Wu, H. Kan, and J. Kurths, "A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps," *Appl. Soft Comput.*, vol. 37, pp. 24–39, Dec. 2015.

[47] Q. Zhang, L. Guo, and X. Wei, "A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system," *Optik-Int. J. Light Electron Opt.*, vol. 124, no. 18, pp. 3596–3600, Sep. 2013.

[48] Y. Zhang, "Cryptanalysis of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system," *Optik*, vol. 126, no. 2, pp. 223–229, Jan. 2015.

[49] X. Wang, Y. Wang, X. Zhu, and C. Luo, "A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level," *Opt. Lasers Eng.*, vol. 125, Feb. 2020, Art. no. 105851.

[50] K. Xuejing and G. Zihui, "A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115670.

[51] M. A. Khan, J. Ahmad, Q. Javaid, and N. A. Saqib, "An efficient and secure partial image encryption for wireless multimedia sensor networks using discrete wavelet transform, chaotic maps and substitution box," *J. Mod. Opt.*, vol. 64, no. 5, pp. 531–540, Mar. 2017.

[52] I. Shatheesh Sam, P. Devaraj, and R. S. Bhuvaneswaran, "An intertwining chaotic maps based image encryption scheme," *Nonlinear Dyn.*, vol. 69, no. 4, pp. 1995–2007, Sep. 2012.

[53] O. D. King and P. Gaborit, "Binary templates for comma-free DNA codes," *Discrete Appl. Math.*, vol. 155, nos. 6–7, pp. 831–839, Apr. 2007.

[54] *IEEE Standard 754-1985 Standard for Binary Floating-Point Arithmetic*, IEEE Standard 754-1985, I.C.S.S. Committee and A.N.S. Institute, 1985.

[55] D. H. ElKamchouchi, H. G. Mohamed, and K. H. Moussa, "A bijective image encryption system based on hybrid chaotic map diffusion and DNA confusion," *Entropy*, vol. 22, no. 2, p. 180, Feb. 2020.

[56] A. Kulsoom, D. Xiao, Aqeel-ur-Rehman, and S. A. Abbas, "An efficient and noise resistive selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules," *Multimedia Tools Appl.*, vol. 75, no. 1, pp. 1–23, Jan. 2016.

[57] Aqeel-ur-Rehman, X. Liao, A. Kulsoom, and S. Ullah, "A modified (dual) fusion technique for image encryption using SHA-256 hash and multiple chaotic maps," *Multimedia Tools Appl.*, vol. 75, no. 18, pp. 11241–11266, 2016.

[58] Aqeel-ur-Rehman, X. Liao, M. A. Hahsmi, and R. Haider, "An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using DNA and chaos," *Optik-Int. J. Light Electron Opt.*, vol. 153, pp. 117–134, Jan. 2018.

[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.-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.

[61] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[62] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Opt. Commun.*, vol. 285, no. 1, pp. 29–37, Jan. 2012.

[63] B. Norouzi and S. Mirzakuchaki, "A fast color image encryption algorithm based on hyper-chaotic systems," *Nonlinear Dyn.*, vol. 78, no. 2, pp. 995–1015, Oct. 2014.

[64] N. Taneja, B. Raman, and I. Gupta, "Combinational domain encryption for still visual data," *Multimedia Tools Appl.*, vol. 59, no. 3, pp. 775–793, Aug. 2012.

[65] B. Norouzi, S. M. Seyedzadeh, S. Mirzakuchaki, and M. R. Mosavi, "A novel image encryption based on hash function with only two-round diffusion process," *Multimedia Syst.*, vol. 20, no. 1, pp. 45–64, Feb. 2014.

[66] 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, 2014.

[67] J. Ahmad and S. O. Hwang, "A secure image encryption scheme based on chaotic maps and affine transformation," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13951–13976, 2016.

[68] Z. Hua and Y. Zhou, "Image encryption using 2D logistic-adjusted-sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.

[69] Z. Xiong, Y. Wu, C. Ye, X. Zhang, and F. Xu, "Color image chaos encryption algorithm combining CRC and nine palace map," *Multimedia Tools Appl.*, vol. 78, no. 22, pp. 31035–31055, 2019.

[70] L. Huang, S. Cai, X. Xiong, and M. Xiao, "On symmetric color image encryption system with permutation-diffusion simultaneous operation," *Opt. Lasers Eng.*, vol. 115, pp. 7–20, Apr. 2019.

[71] H. Diab, "An efficient chaotic image cryptosystem based on simultaneous permutation and diffusion operations," *IEEE Access*, vol. 6, pp. 42227–42244, 2018.

[72] H. Joe, "Asymptotic efficiency of the two-stage estimation method for copula-based models," *J. Multivariate Anal.*, vol. 94, no. 2, pp. 401–419, 2005.

[73] A. Girdhar and V. Kumar, "A RGB image encryption technique using Lorenz and Rossler chaotic system on dna sequences," *Multimedia Tools Appl.*, vol. 77, no. 20, pp. 27017–27039, 2018.

NADEEM IQBAL received the M.Phil. degree in computational science and engineering from the NUST, Islamabad, Pakistan. He is currently working as an Assistant Professor with the School of Computing and Information Sciences, Imperial College of Business Studies (ICBS), Lahore, Pakistan. Prior to joining the ICBS, he worked in various academic institutions and has guided numerous undergrad and masters students. His research interests span images cryptography, computer graphics and philosophy of mathematics.

MUHAMMAD HANIF received the B.S. degree in information technology from the University of Malakand, Pakistan, and the M.S. degree in information technology from SEECS, NUST, Islamabad, Pakistan. He is currently pursuing the Ph.D. degree in computer science with NCBA&E Lahore, Pakistan. He is currently working as an Assistant Professor with the Department of Computer Science, Bahria University, Lahore Campus, Pakistan. He was with various academic institutions, and has supervised numerous bachelor's and master's students from past eight years. His current research interests include images cryptography, computer graphics, networking, cloud computing, and the Internet of Things.

SAGHEER ABBAS received the M.Phil. degree in computer science and the Ph.D. degree from the School of Computer Science, NCBA&E, Lahore, Pakistan. He has been teaching graduate and undergraduate students in computer science and engineering for the past eight years. He is currently an Assistant Professor with the School of Computer Science, NCBA&E. He has published about 60 research articles in international journals and reputed international conferences. His current research interests include cloud computing, the IoT, intelligent agents, image processing, and cognitive machines with various publications in international journals and conferences.

MUHAMMAD ADNAN KHAN received the B.S. and M.Phil. degrees from the International Islamic University, Islamabad, Pakistan, by obtaining scholarship award from the Punjab Information and Technology Board, Govt of Punjab, Pakistan, and the Ph.D. degree from ISRA University, Pakistan, by obtaining a scholarship award from the Higher Education Commission, Islamabad, Pakistan. He is currently working as an Assistant Professor with the Department of Computer Science, Lahore Garrison University, Lahore, Pakistan. Before joining the Lahore Garrison University, he has worked in various academic and industrial roles in Pakistan. He has been teaching graduate and undergraduate students in computer science and engineering for the past 12 years. He is currently guiding four Ph.D. scholars and six M.Phil. Scholars. He has published more than 150 research articles in International Journals as well as reputed International Conferences. His research interests primarily include machine learning, MUD, image processing and medical diagnosis, channel estimation in multi-carrier communication systems using soft computing with various publications in journals and conferences of international repute.

SULTAN H. ALMOTIRI received the B.Sc. degree (Hons.) in computer science from King Abdulaziz University, Saudi Arabia, in 2003, and the M.Sc. degree in internet, computer, and system security and the Ph.D. degree in wireless security from Bradford University, U.K., in 2006. He was the Chairman of the Computer Science Department, Umm AlQura University, Saudi Arabia, and the Vice Dean of eLearning and distance Education with Umm AlQura University, Saudi Arabia. He is currently an Assistance Professor with the Computer Science Department, Faculty of Computer and Information Systems, Umm AlQura University. His research interests including cyber security, cryptography, AI, machine learning, eHealth, eLearning, IoT, RFID and wireless sensors, and image processing.

MOHAMMED A. AL GHAMDI (Associate Member, IEEE) received the B.Sc. degree (Hons.) in computer science from King Abdulaziz University, Saudi Arabia, in 2004, the master's degree in internet software systems from the University of Birmingham, in 2007, and the Ph.D. degree in computer science from the University of Warwick, U.K, in 2012. He is currently an Associate Professor with the Computer Science Department, Umm Al-Qura University, Saudi Arabia. He has published a number of good quality journal papers in data analysis, AI, cloud computer, cyber security and machine learning.