IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 28, 2020, accepted April 13, 2020, date of publication April 24, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990170

Color Image Encryption Algorithm Based on Hyper-Chaos and DNA Computing

M. G. ABBAS MALIK[1], ZIA BASHIR[2], NADEEM IQBAL[3,4], AND MD. ATHAR IMTIAZ[5]

[1]School of Business and ICT, Universal College of Learning, Palmerston North 4442, New Zealand
[2]Department of Mathematics, Quaid-i-Azam University, Islamabad 45320, Pakistan
[3]Department of Computer Science, NCBA&E, Lahore 54660, Pakistan
[4]School of Computing and Information Sciences, Imperial College of Business Studies, Lahore 53720, Pakistan
[5]School of Fundamental Sciences, Massey University, Palmerston North 4442, New Zealand

Corresponding author: M. G. Abbas Malik (a.malik@ucol.ac.nz)

ABSTRACT In this study, a novel technique using a hyper chaotic dynamical system and DNA computing has been designed with high plaintext sensitivity. In order to reduce cost, a selection procedure using tent map has been employed for generating different key streams from the same chaotic data obtained from the iterations of chaotic dynamical system. After separating the three channels from the input color image, they are both confused and diffused. First of all, these channels are diffused on a decimal level. Then they are permuted. Further, DNA encoding is performed upon these channels. Moreover, DNA level diffusion is performed to further increase the degree of randomness in the image. Lastly, the DNA encoded image is converted into decimal to get the final cipher image. Both the experimental results and security analysis strongly demonstrate the robustness of the proposed scheme. A comparison of the proposed scheme has also been made with other recently developed schemes to show that this scheme outperforms the others in terms of computational cost, time and memory efficiency. Additionally, with the large key space, the proposed scheme can resist any brute force, plaintext and statistical attacks, therefore it is a good fit for the real world applications of the image security.

INDEX TERMS Image processing, chaos, encryption, decryption, DNA computing.

I. INTRODUCTION

With the digitalization of the world, cyber security of digital data that flows through the Internet in various forms is one of the main concern for the future of digitalization. Use of digital images is increasing in expressing and sharing sensitive information, whether in military, banks or the personal moments of an individual. Numerous image encryption schemes have been developed based on different chaos theories, maps and systems to hide plain images from unwanted eyes or unauthorized/illegal access [1]–[3]. Hackers are continuously trying to break these schemes. The battle between cryptosystem and cryptoanalysis is never ending. Thus, to improve and develop more secure schemes is essential and inevitable for image security.

Usually, the images have high correlation between adjacent pixels, large data capacities and high redundancy, making it difficult for conventional security encryption schemes like DES, IDEA, RSA and AES to maintain higher standards of security [4], [5]. Whereas, the encryption schemes based on chaotic maps and systems [6]–[13] are more up to the job due to their properties like high sensitivity to initial values and other system parameters, plaintext sensitivity, large key space, chaotic behaviour and ergodicity. But many encryption schemes have been successfully broken with use of classical attacks [14]–[16].

Man must learn from nature in order to solve his problems. Among natural structures, DNA, contains genetic code, is the book of nature containing instructions of life, determining what we are today. The DNA model is a good way to store and manipulate large amount of data. Therefore, many encryption schemes [17]–[21] based on DNA computing and chaos have been developed. But, there are certain drawbacks or rooms for improvements for the existing encryption algorithms. One aspect of these schemes is the dimension of chaotic maps/systems used in them, either low dimension (one or

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

two) or high dimension (more than 2). Each category has its own merits and demerits. Encryption algorithms with low dimension chaotic maps/systems [17], [18], [22] are easy to implement with simple structure and low computational cost, but unable to maintain high security standards. With growing computing power, their small key space is vulnerable against brute force and other attacks [23]–[25]. Although, the high dimensional chaotic maps/systems generate more complex and random chaotic sequences with a large keys space. Being more sensitive to the initial values, they enable us to develop more secure image encryption schemes [7], [19], but with a higher computational cost. Generally in various steps of encryption process, different chaotic data is used without repetition, adding more complexity to encryption scheme.

Apart from quality of chaotic maps/systems, there are other problems regarding the use of DNA model. There are eight rules to convert a decimal pixel value to a DNA sequence and visa versa. In [18], [26], [27], the same rule is used for all the pixels in encryption and decryption. In [22], [28]–[32], conversion rules are part of the secret key set. But with the static conversion rule and small keys space of conversion rules (1-8), these encryption schemes are vulnerable against the plaintext and brute force attacks [33]. Although, there are some improvements in recent encryption algorithms. For instance, Wu *et al.* [34] generated DNA conversion rules based on respective entropies of all the three red, green and blue channels separately, but used the same rule of conversion for all the pixels of a channel. In another example, Chen *et al.* [35] generated random conversion rules based on rand function, but with the problem that they did not depend on plain images. Different plain images have same conversion rules, making their scheme insecure against known plaintext attack, chosen plaintext attack and statistical attack. Furthermore, DNA operations like DNA addition, DNA subtraction, DNA *XOR*, etc. (key components of a DNA model), are used widely for the diffusion of plain images [21], [22], [28], [29], [31], [36], [37]. Again, there are eight rules for applying these operations among DNA sequences, but most of the time, same rule is used that makes outcome predictable and less random. Thus, a lot of encryption algorithms have been broken due to less amount of plaintext sensitivity. For instance, encryption scheme [38] was broken by Zhang [39] with the help of a chosen plaintext attack. Liu *et al.* [40] showed that encryption algorithm [41] is vulnerable against the chosen plaintext attack due to its non-sensitivity to plain image and key set. Recently, an encryption scheme [42] has been cryptanalysed by Akhavan *et al.* [43] finding that the cipher image did not reasonably depend upon plain image and key set, making it vulnerable against known plaintext attack. They demonstrated that with the use of two or four known images, the key set can be found. Thus, we need a more efficient, robust and secure encryption scheme to overcome the problems, discussed above.

According to our hypothesis, If the same chaotic data can be used multiple times to generate random key streams without compromising the data security, then the computational cost can be reduced significantly even with higher dimension. To address the problems of lower dimensional schemes, a four dimensional hyper chaotic dynamical system (CDS) [44] is used in the proposed novel RGB encryption algorithm. Whereas, to reduce the computational cost, Chaotic Dynamical State Variables Selection Procedure (CDSVSP) designed by Bashir *et al.* [1] is used to select cross states chaotic data of CDS. More precisely to encrypt a color image of size $M \times N$, usually more than $3MN$ chaotic variables are used in each step of an encryption process to maintain higher standards of security. For example, Wu *et al.* [21] generated $12MN + 2400$ chaotic variables for encryption and Xiuli *et al.* [7] generated $16MN + 3200$ chaotic variables for encryption. In our case, with the help of CDSVSP, we only need $MN + 300$ chaotic variables in the encryption process. Thus, reducing the computational and memory cost significantly and resulting into a much faster encryption process. The cross state data is more random and chaotic in comparison with using a single state to generate key streams or pseudo random numbers. Furthermore, to use full potential of DNA model, pseudo random numbers are generated based on the pixels of plain image that govern the rules of DNA operations, conversion between DNA sequence and decimal numbers. Each pixel is converted with a different rule that is totally random and dependent on the pixel value. Also, the outcomes of DNA operations are entirely different for different rules, this greatly improves the randomness of cypher image. Even a single pixel value change of a plain image will result into entirely different DNA sequences, thus making the proposed scheme more secure against the plaintext and statistical attacks. Following this idea, an image encryption scheme based on hyper chaotic dynamical system [44] and DNA computing is designed with the following novel highlights:

- To overcome the shortcomings of lower dimensional chaotic maps/systems, a four dimensional chaotic dynamical system is employed. Further, to reduce the computation cost (a major concern of using higher dimensional chaotic maps/systems), CDSVSP is used to generate multiple key streams from the same chaotic data obtained by the iterations of chaotic dynamical system. As a result, the size of chaotic data used in the proposed encryption scheme is much less, even less than one tenth of what existing encryption schemes are using.
- To use the full potential of DNA computing, based on key streams different sequences of pseudo random numbers are generated to be used as rules for DNA operations and conversion of decimal pixel values to DNA and vice versa. This improves greatly the unpredictability and randomness of resultant DNA sequences or cipher image.
- To diffuse plain image properly, first we diffuse it with mask images (key streams) at decimal level and then at DNA level, the three channels red, green and blue are inter blended with different combinations of DNA operations randomly.

- To confuse plain image properly, the three channels are combined as one dimension array and then permuted, so that the pixel values are randomly distributed in all three channels.
- To ensure the maintenance of high standards of security for the proposed novel image encryption scheme, pixel values of plain image are involved in all the steps of encryption algorithm, *i.e.*, generation of key streams (pseudo random numbers), DNA conversion, DNA operations and decimal conversion. Together with the large key space and hyper chaotic data being highly sensitive to secret keys, a small change in the plain image and key set would result into an entirely different cipher image, thus making the encryption scheme more secure against the known attacks.

In the light of above discussed features, proposed image encryption is highly efficient and secure for practical use with much less cost and higher security in comparison with existing encryption schemes. The outcomes of extensive security analysis and tests done in section 4 confirm our claim.

The paper is organised as follows. Section 2 presents preliminary theories about CDM, DNA computing and CDSVSP. Section 3 showcases the proposed novel encryption scheme for RGB images. In section 4, simulation and the ensuing security analyses of the proposed scheme have been performed. Finally, Section 5 concludes this study.

II. PRELIMINARIES

Baker map, Lorenz map and logistic map are a few of the many maps which generate random data used in different chaos based applications. Our choice is the chaotic tent map [45] known as:

$$f(a,\rho,x)=\begin{cases}\lceil\frac{\rho}{a}x\rceil, & \text{if } 0\leq x\leq a;\\ \lfloor\frac{\rho(\rho-x)}{\rho-a}x\rfloor+1, & \text{if } a<x\leq\rho,\end{cases}\tag{II.1}$$

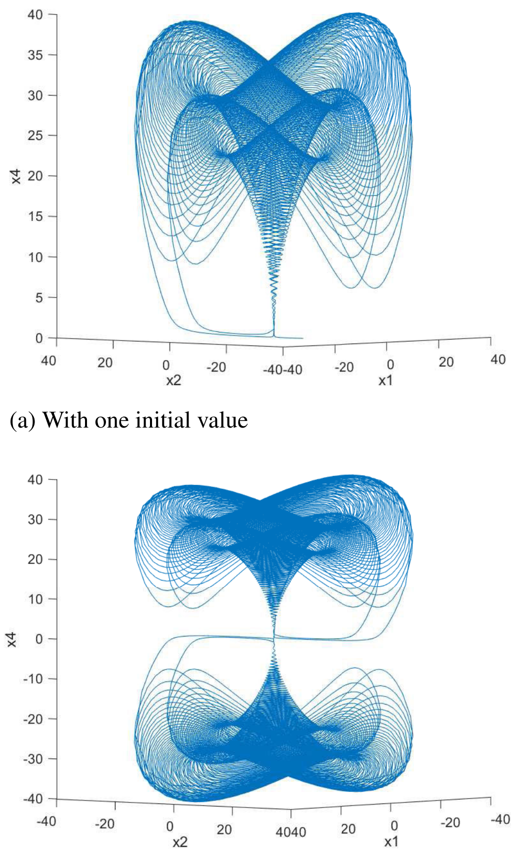
where $a\in(0,\rho)$ is an integer. It will be used in the proposed image encryption scheme.

A. CHAOTIC FOUR DIMENSIONAL DYNAMICAL SYSTEM

The chaotic dynamical systems are well known source to produce the chaotic data with desired properties. Lorenz and Rossler did the preliminary work by developing their respective three dimensional chaotic dynamical systems. Later, Yong and Yun-Qing [44] developed a much better four dimensional chaotic dynamical system (CDS) as follows.

$$\begin{aligned}\dot{x_1}&=ax_1-b_1x_1x_2x_3\\ \dot{x_2}&=bx_2-b_2x_1x_3x_4\\ \dot{x_3}&=cx_3-b_3x_1x_2x_4\\ \dot{x_4}&=dx_4-b_4x_1x_2x_3\end{aligned}\tag{II.2}$$

This new system is symmetrical, dissipative and possesses very rich dynamical structures. The Figure 1 shows the topology and chaotic behavior of the attractors of above chaotic dynamical system for $a=-10$, $b=3$, $c=-1$, $d=-2$, $b_1=1$, $b_2=-1$, $b_3=1$ and $b_4=1$.



(a) With one initial value

(b) With two initial values

FIGURE 1. With one initial value.

B. CHAOTIC DYNAMICAL STATE VARIABLES SELECTION PROCEDURE

For simplicity and better understanding the use of chaotic data generated from CDS II.2, we recalled the Chaotic Dynamical State Variable Selection Procedure (CDSVSP) developed by Bashir *et al.* [1] here as folllows:

Take a one dimensional array of 8 bit pixel values $P=\{P(0),P(1),P(2),\ldots,P(MN-1)\}$ of size MN, where M and N are natural numbers determining the dimension of plain image. The CDS II.2 is iterated $(M\times N)/4+n_0$ times with $n_0\geq 100$ for the initial values x_0, y_0, z_0 and w_0 that will be the part of secret keys. The CDSVSP is explained with the help of following definitions:

1) With each iteration of the CDS II.2, we will get the values of the four state variables, *i.e.*, X, Y, Z and W. Let $S=\{X_i,Y_j,Z_k,W_l\}$ where X_i, Y_j, Z_k, W_l are the states of X, Y, Z and W in i^{th}, j^{th}, k^{th} and l^{th} iteration, respectively and the values of i, j, k and l are not equal to each other.

2) A new variable say $slt(L)$ is defined as the selected variable in $\{X_i, Y_j, Z_k, W_l\}$.

$$slt(L) = \begin{cases} X_i & \text{if } index(L) = 0, \\ Y_j & \text{if } index(L) = 1, \\ Z_k & \text{if } index(L) = 2, \\ W_l & \text{if } index(L) = 3. \end{cases}$$

Through the usage of this variable, a key stream element for $P(L)$ will be generated. An indicator $index(L)$ will make the decision, defined as:

$$index(L) = mod(f(a, \rho, P(L-1)), 4)$$

In the above equation $f(a, \rho, x)$ is a tent map. a and ρ contribute to the secret keys. For the first pixel value, $P(-1)$ has been set as a seed value.

The CDSVSP works as follows:

Choose i_0, j_0, k_0, l_0 sufficiently large, different from each other and act as secret keys. Now, we create the initial state $S = \{X_{i_0}, Y_{j_0}, Z_{k_0}, W_{l_0}\}$, for the first pixel $P(0)$. Then, we calculate the selected variable $slt(0)$ for $P(0)$ by computing $index(0)$. System state is updated as follows:

$$S = \begin{cases} \{X_{i_0+1}, Y_{j_0}, Z_{k_0}, W_{l_0}\} & \text{if } index(0) = 0, \\ \{X_{i_0}, Y_{j_0+1}, Z_{k_0}, W_{l_0}\} & \text{if } index(0) = 1, \\ \{X_{i_0}, Y_{j_0}, Z_{k_0+1}, W_{l_0}\} & \text{if } index(0) = 2, \\ \{X_{i_0}, Y_{j_0}, Z_{k_0}, W_{l_0+1}\} & \text{if } index(0) = 3. \end{cases}$$

Again, calculate the selected variable $slt(1)$ from updated state S for $P(1)$ by computing $index(1)$ and update the system state. Inductively, get the updated state variable set S for $P(n)$ and select the state variable $slt(n)$ from S by computing $index(n)$. Suppose, we have the state $\{X_i, Y_j, Z_k, W_l\}$ for the $P(n)$ and without loss of generality, we assume that $index(n) = 0$. In this case, the state value X_i is chosen for ciphering $P(n)$, and the combination of state variables is $\{X_{i+1}, Y_j, Z_k, W_l\}$. Similarly, calculate $index(n+1)$, and let $index(n+1) = 1$. The state value Y_j will be selected to cipher $P(n+1)$, and the combination state variables transforms to $\{X_{i+1}, Y_{j+1}, Z_k, W_l\}$. Similarly, we supposed that $index(n+2) = 2$. The state value Z_k will be selected for ciphering $P(n+2)$. Then, state variables combination changes to $\{X_{i+1}, Y_{j+1}, Z_{k+1}, W_l\}$. Assume that $index(n+3) = 3$, without loss of generality. The state value W_l will be chosen for ciphering $P(n+3)$.

C. DNA MODEL

DNA encoding is done using four basic nucleic acids. These are called Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). These nucleotides complement themselves. For example, upon attributing '01' to C, '10' will be attributed to G. Analogously, upon attributing '00' to A, '11' will be attributed to T. There are total $4! = 24$ combinations out of which just 8 combinations comply with the Watson-Crick rule of complementary nucleotides. Eight rules of encoding have been shown in the Table 1.

TABLE 1. Eight kinds of encoding rules for DNA sequencing.

1	2	3	4	5	6	7	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. Addition, Subtraction and XOR operation on DNA nucleotides according to rule 1.

+	A	T	C	G	-	A	T	C	G	⊕	A	T	C	G
A	A	T	C	G	A	C	G	A	T	A	A	T	C	G
T	T	G	A	C	T	A	C	T	G	T	T	A	G	C
C	C	A	G	T	C	G	T	C	A	C	C	G	A	T
G	G	C	T	A	G	T	A	G	C	G	G	C	T	A

To use in encryption process, $DNA-Conv$ and $DEC-Conv$ are two conversion functions from 8-bit pixel value to DNA sequence of length four and vice versa based on rules Table 1, respectively. For example $DNA - Conv(89, 2) = GGCG$, $DNA - Conv(89, 4) = TTAT$ and $DNA - Conv(89, 5) = AATA$. Also, $DEC - Conv(ATCG, 2) = 57$, $DEC - Conv(ATCG, 5) = 108$ and $DEC - Conv(ATCG, 7) = 198$.

There exist other operations in DNA cryptography like addition, subtraction and XOR. These operations resemble the operations in the binary number system. Since we have eight kinds of encoding rules, in the same way, we have eight kinds of addition, subtraction and XOR operations. Table 2 presents these three operations according to rule 1. For further use, we denote $+$, $-$ and $\oplus$ for DNA addition, DNA subtraction and DNA XOR, respectively. We can these operations between two DNA sequences of length four based upon rules (1-8). For example, $+(ATGG, GCAT, 2) = GGGA$ and $+(ATGG, GCAT, 6) = AGAT$; Also, $-(ATGG, GCAT, 2) = GGGT$ and $-(ATGG, GCAT, 7) = TTTA$; Lastly, $\oplus(ATGG, GCAT, 1) = GGGC$ and $\oplus(ATGG, GCAT, 5) = GAGT$.

III. PROPOSED IMAGE ENCRYPTION SCHEME

The given color plain image of size $M \times N \times 3$ comprising of three channels (R, G, B) is first converted into three one dimensional arrays $I = (RO, GO, BO)$, by inserting one row after the other, respectively. Each of the one dimensional MN sized arrays RO, GO and BO consists of eight bit pixel values of red, green and blue channels, respectively. The CDS II.2 is iterated $(MN)/4 + n_0$ times with $n_0 \geq 100$ for the initial values x_0, y_0, z_0 and w_0 that will be part of secret keys.

A. KEY STREAM GENERATION PROCEDURE

The key stream generation procedure (KSGP) given below will be used in various steps of the encryption process to generate entirely different and random key streams from the same chaotic data obtained by the iterations of CDS II.2.

Let $I = (RO, GO, BO)$ be the three given one dimensional MN sized arrays of eight bit pixel values. In order to make sure that a change in a single pixel value in any channel results

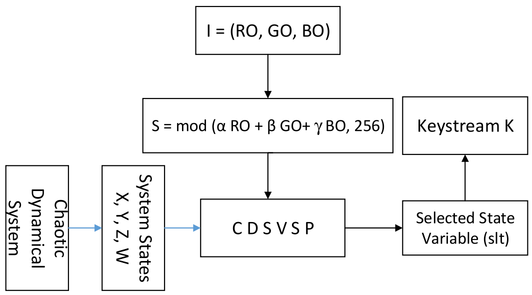


FIGURE 2. Key stream generation procedure.

into an entirely different cipher image, a linear combination of RO, GO and BO is obtained as follows.

$$S = \mathrm{mod}(\alpha RO + \beta GO + \gamma BO, 256)$$

where α, β, γ are integers and part of the secret keys. Thus, S is a one dimensional array of MN eight bit pixels values. Then, we apply CDSVSP to S and get the selected state variable slt that is used to generate the key stream denoted by K as follows:

$$K(i) = \mathrm{mod}(round((abs(slt(i)) - floor(abs(slt(i)))) \times 10^{15}), 256) \tag{III.1}$$

where $floor(x)$ means the nearest integer to x toward minus infinity, $abs(x)$ is the absolute value of x, $\mathrm{mod}(x, y)$ is the remainder when x is divided by y, $round(x)$ rounds the value of x and i varies from 0 to $MN - 1$.

Remark 1: The brief description of KSGP can be seen in the Figure 2. The CDSVSP [1] is quite efficient in generating entirely different sequences for selected variables even for the change of one pixel. Therefore, KSGP results into random key streams that will allow us to achieve high standards of security with minimal resources.

B. ENCRYPTION PROCESS

The proposed encryption procedure is presented as follows:

Step: 1 (Decimal diffusion)

KSGP is applied to the plain image $I = (RO, GO, BO)$ three times with different key sets to generate three key streams K_1, K_2 and K_3. Then, I will be diffused with these key streams to get $I_1 = (RO_1, GO_1, BO_1)$ as follows

$$\begin{cases} RO_1(i) = \mathrm{mod}(RO(i) + RO_1(i-1) + K_1(i) + K_2(i), 256), \\ GO_1(i) = \mathrm{mod}(GO(i) + GO_1(i-1) + K_2(i) + RO_1(i), 256), \\ BO_1(i) = \mathrm{mod}(BO(i) + BO_1(i-1) + K_3(i) + GO_1(i), 256), \end{cases} \tag{III.2}$$

where the seed values $RO_1(-1)$, $GO_1(-1)$ and $BO_1(-1)$ are part of secret keys.

Step: 2 (Permutation)

The three one dimensional arrays $I_1 = (RO_1, GO_1, BO_1)$ are combined as one dimensional $3MN$ sized array

$$T = \{RO_1(0), RO_1(1), \ldots, RO_1(MN-1), GO_1(0), GO_1(1), \ldots, GO_1(MN-1), BO_1(0), BO_1(1), \ldots, BO_1(MN-1)\}.$$

In order to scramble the pixel positions, the chaotic tent map $f(w, 3MN - 1, x)$ has been used h times. Here $w = \sum_{i=0}^{3MN-1} T(i) \bmod (3MN - 1)$ and $x = \{0, 1, 2, \ldots, 3MN - 1\}$, being the input variable, indexes each pixel of T. Since tent map works on the one-to-one principle, so it will render a permutation say $\sigma(x)$. As the permutation $\sigma(x)$ is applied on T, T' is obtained. In other words, positions of pixels or index values are changed in $\sigma(x)$, *i.e.*, $T'(\sigma(x)) = T(x)$. Then, T' is split into three one dimensional MN sized arrays $I_2 = (RO_2, GO_2, BO_2)$ as follows:

$$\begin{cases} RO_2 = \{T'(0), T'(1), \ldots, T'(MN - 1)\}, \\ GO_2 = \{T'(MN), T'(MN + 1), \ldots, T'(2MN - 1)\}, \\ BO_2 = \{T'(2MN), T'(2MN+1), \ldots, T'(3MN-1)\}. \end{cases} \tag{III.3}$$

Step: 3 (DNA conversion)

Once again, KSGP is applied to $I_2 = (RO_2, GO_2, BO_2)$ four times with different key sets to generate four key streams K_4, K_5, K_6 and K_7. The three one dimensional arrays RO_2, GO_2 and BO_2 are transformed into DNA nucleotides resulting into the arrays RO_3, GO_3 and BO_3 by applying DNA conversion rules based on the key streams K_4, K_5 and K_6, respectively, as follows:

$$\begin{cases} RO_3(i) = DNA - Conv(RO_2(i), rule_1(i)), \\ GO_3(i) = DNA - Conv(GO_2(i), rule_2(i)), \\ BO_3(i) = DNA - Conv(BO_2(i), rule_3(i)), \end{cases} \tag{III.4}$$

where

$$rule_{j-3}(i) = \mathrm{mod}(K_j(i), 8) + 1, \tag{III.5}$$

for $i = 0, 1, 2, \ldots, MN - 1$ and $j = 4, 5, 6$. Each of the three one-dimensional arrays in $I_3 = (RO_3, GO_3, BO_3)$ consists of MN DNA sequences of length 4.

Step: 4 (DNA level diffusion)

The DNA encoded arrays $I_3 = (RO_3, GO_3, BO_3)$ are inter diffused with the help of DNA operations to get DNA arrays $I_4 = (RO_4, GO_4, BO_4)$ according to the rules based on the key stream K_7 as follows:

if $r(i) = 0$

$$\begin{cases} RO_4(i) = \oplus(RO_3(i), GO_3(i), rule_4(i)), \\ GO_4(i) = +(GO_3(i), BO_3(i), rule_4(i)), \\ BO_4(i) = -(BO_3(i), RO_4(i), rule_4(i)), \end{cases} \tag{III.6}$$

if $r(i) = 1$

$$\begin{cases} RO_4(i) = -(RO_3(i), GO_3(i), rule_4(i)), \\ GO_4(i) = \oplus(GO_3(i), BO_3(i), rule_4(i)), \\ BO_4(i) = +(BO_3(i), RO_4(i), rule_4(i)), \end{cases} \tag{III.7}$$

if $r(i) = 2$

$$\begin{cases} RO_4(i) = +(RO_3(i), GO_3(i), rule_4(i)), \\ GO_4(i) = -(GO_3(i), BO_3(i), rule_4(i)), \\ BO_4(i) = \oplus(BO_3(i), RO_4(i), rule_4(i)), \end{cases} \tag{III.8}$$

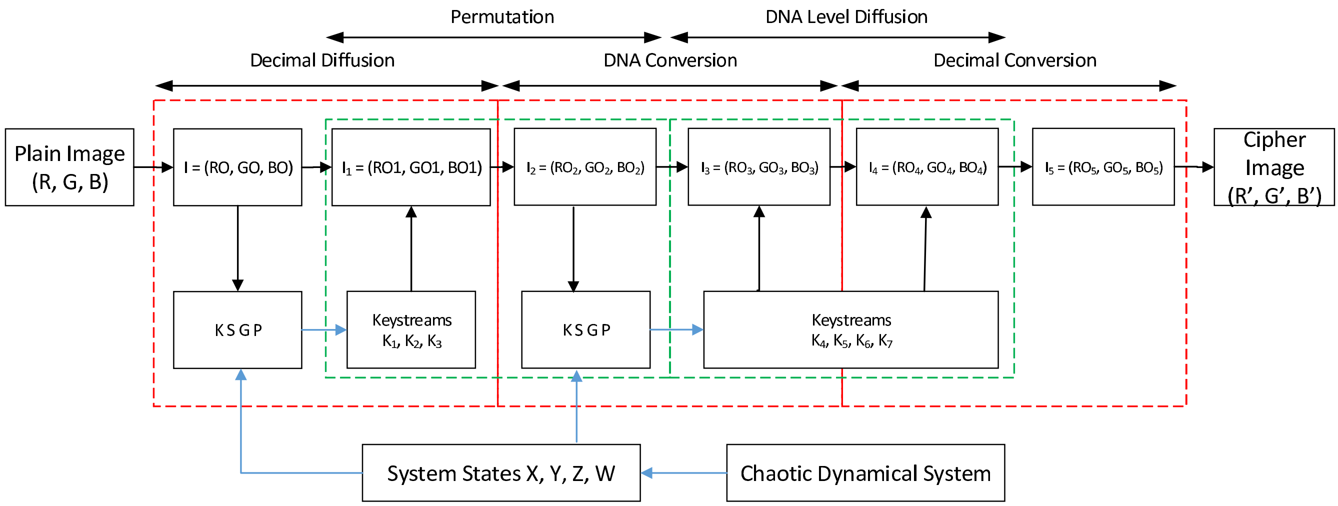


FIGURE 3. Encryption procedure.

where

$$r(i) = \mathrm{mod}(K_7(i), 3), \quad \text{(III.9)}$$

and

$$rule_4(i) = \mathrm{mod}(K_7(i), 8) + 1, \quad \text{(III.10)}$$

for $i = 0, 1, 2, \ldots, MN - 1$.

Step: 5 (Decimal conversion)

The DNA arrays $I_4 = (RO_4, GO_4, BO_4)$ are converted into decimal arrays $I_5 = (RO_5, GO_5, BO_5)$ as follows:

$$\begin{cases} RO_5(i) = DEC - Conv(RO_4(i), rule_5(i)), \\ GO_5(i) = DEC - Conv(GO_4(i), rule_6(i)), \\ BO_5(i) = DEC - Conv(BO_4(i), rule_7(i)), \end{cases} \quad \text{(III.11)}$$

where

$$\begin{cases} rule_5(i) = \mathrm{mod}(i + RO_5(i-1), 8) + 1, \\ rule_6(i) = \mathrm{mod}(i + GO_5(i-1), 8) + 1, \\ rule_7(i) = \mathrm{mod}(i + BO_5(i-1), 8) + 1, \end{cases} \quad \text{(III.12)}$$

for $i = 0, 1, 2, \ldots, MN - 1$ and $RO_5(-1)$, $GO_5(-1)$ and $BO_5(-1)$ are the seed values of DNA to decimal conversion process.

Step: 6 (Cipher image making)

The three one-dimensional MN sized arrays RO_5, GO_5 and BO_5 of eight bit pixel values are converted into $M \times N$ sized red channel R', green channel G' and blue channel B', respectively. Finally, from red, blue and green channels, a cipher image (R', G', B') is generated as an output of the proposed image encryption scheme.

The decryption is done in the reverse order of encryption. The flow chart of encryption process is expressed in Figure 3.

C. DISCUSSION

KSGP is used seven times to generate key streams $K_i(i = 1, 2, \ldots, 7)$ based on CDSVSP and linear combinations of all three channels red, green and blue. Therefore, one pixel change in any of the channels will trigger a different sequence of selected variables in CDSVSP resulting into an entirely different key stream. Also, the pixel values are permuted across channels, so pixel change in any position of the channels is equally affected in generating unpredictable key streams. Overall, key streams are directly linked with pixel values of the plain image.

The mask images (Key streams $K_i(i = 1, 2, 3)$) are diffused with $I = (RO, GO, BO)$ to change pixel values and it is apparent from Equations III.2 that they are also intermixed. The sequences of pseudo random numbers $rule_i(i = 1, 2, 3)$ are used to convert 8-bit decimal pixel values to DNA sequences of length 4. The use of pseudo random numbers being used as a conversion rule results into completely random and unpredictable DNA sequences as the same pixel value has different corresponding DNA sequences for different rule values (1-8).

Moreover, a pseudo random number r given in Equation III.9 determines different set of DNA operations III.6, III.7 and III.8 that are used to intermix red, green and blue channels. Further, the key stream K_7 is also used to decide on the DNA conversion rules (1-8) using $rule_4$ in Equation III.10 for these DNA operations. Thus, the resultant Diffused DNA sequences are totally random and it is not possible for hackers to figure out any thing useful by employing different chosen plain images attacks because, all this DNA diffusion process is dependent upon the pixel values of $I_2 = (RO_2, GO_2, BO_2)$.

Finally in the DNA to decimal conversion process, each value of $I_4 = (RO_4, GO_4, BO_4)$ is converted in 8-bit decimal pixel values with rules $rule_i(i = 5, 6, 7)$. This further randomizes the cipher image as different conversion rules for the same DNA sequence would result in entirely different decimal values. For example, $DEC - Conv(TGAC, 2) = 225$ and $DEC - Conv(TGAC, 7) = 75$.

TABLE 3. Key space.

	Keys	Size
CDS	$x_0, y_0, z_0, w_0, b_1, b_2, b_3, b_4$	10^{120}
KSGP	$a_i, \rho_i, \alpha_i, \beta_i, \gamma_i, S_i(-1)\ (i = 1, 2, ..., 7)$	10^{121}
Decimal Diffusion	$RO_1(-1), GO_1(-1), BO_1(-1)$	10^7
Decimal Conversion	$RO_5(-1), GO_5(-1), BO_5(-1)$	10^7
	Total	10^{215}

In light of the above discussion, it is clear that the proposed encryption algorithm has the potential to resist any attacks with much less resources as compared to the other existing schemes. The experimental and security analyses done in the next section also support our thesis.

IV. SIMULATION AND SECURITY ANALYSIS

In the realm of image cryptography, a plethora of attacks exist. Normally, all these attacks exploit the inherent vulnerabilities lying in the design of ciphers. The list of known attacks normally include chosen plaintext/ciphertext attack, statistical attack, differential attack, brute force attack, noise and data crop attack, entropy attack *etc.* One of the principal foci of any cipher is to make it defiant to these attacks.

In order to do the security and performance analysis of the proposed scheme, eight different color images have been taken, all of size 256 × 256. The images 'Lena', 'Baboon', 'Peppers', 'Tree', 'House', 'Beans', 'F16' and 'Girl' can be accessed in the USC-SIPI Image Database, available online at: http://sipi.usc.edu/database/). All the simulations are done in Matlab 2016 version with 64-bit double-precision according to the IEEE 754 Standard [46].

To simulate the proposed algorithm, we have taken these initial values and system parameters for the four-dimensional chaotic dynamical system:

$x_0 = 17$, $y_0 = 25$, $z_0 = 118$, $w_0 = 5$, $a = -10$, $b = 3$, $c = -1$, $d = -2$, $b_1 = 1$, $b_2 = -1$, $b_3 = 1$, $b_4 = 1$.

Furthermore, the step size is taken sufficiently small in solving CDS to avoid any unwanted behaviour [47] and degradation effects [48].

The Figure 4 shows the input plain-images, encrypted images and the corresponding decrypted images. After encryption, the cipher images have been completely changed and have left no clue to reach to the original images.

A. KEY SPACE ANALYSIS

Key space is a very important factor for developing an encryption scheme. No matter how much robust algorithm, one has developed, if the key space is low, then it can be broken through brute force attack. The proposed encryption scheme comprises of twelve different system states and parameters that are x_0, y_0, z_0, w_0, a, b, c, d, b_1, b_2, b_3, b_4. Given the computational precision of 10^{-15}, the key space of the proposed encryption scheme comes out to be $10^{215} \approx 2^{714}$ as shown in the Table 3.

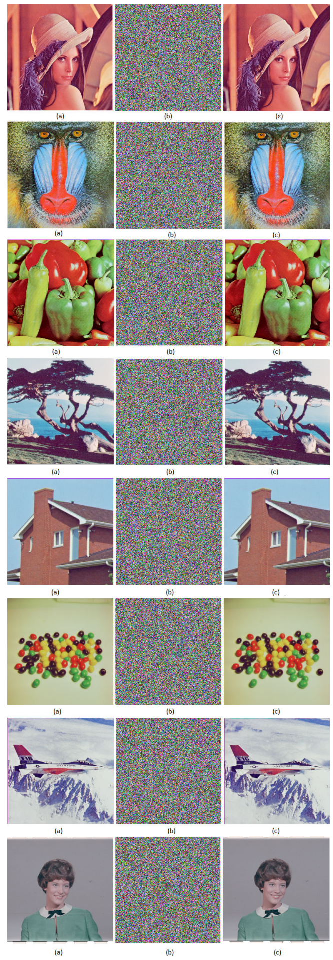


FIGURE 4. (a) Plain images (b) Encrypted images (c) Decrypted images (Lena, Baboon, Peppers, Tree, House, Beans, F16 and Girl).

TABLE 4. Key space comparison.

Scheme	Key space
Proposed	10^{215}
Ref. [7]	3.9402×10^{185}
Ref. [21]	10^{88}
Ref. [35]	10^{60}
Ref. [49]	2.4×10^{112}
Ref. [50]	2.9645×10^{149}
Ref. [51]	1.6777×10^{64}

A comparison of the key space of the algorithm with the other recent algorithms has been made in the Table 4. The key space is sufficiently large for the proposed image encryption algorithm as compared to other recent algorithms.

B. SENSITIVITY TO SECRET KEY

Extreme key sensitivity is a necessary and sufficient condition for a good cryptosystem. Key sensitivity means that a minimal change in any of the keys should results into a radically different output. It can be gauged in both encryption and decryption processes. During the encryption process, a tiny change in only one of the secret keys should result in a

TABLE 5. Difference rates between two images encrypted by slightly different keys.

Secret security keys	Difference rates(%)							
	Lena	Baboon	Peppers	Tree	House	Beans	F16	Girl
$Key_1(x_0{}' = x_0 + 10^{-14})$	99.6007	99.6124	99.6165	99.6073	99.6119	99.6170	99.6307	99.6073
$Key_2(y_0{}' = y_0 + 10^{-14})$	99.6182	99.6292	99.6414	99.6038	99.6140	99.6145	99.6229	99.6155
$Key_3(z_0{}' = z_0 + 10^{-14})$	99.6170	99.6156	99.6140	99.6028	99.6287	99.6195	99.6395	99.6272
$Key_4(w_0{}' = w_0 + 10^{-14})$	99.6267	99.6282	99.6299	99.6089	99.6007	99.6033	99.6251	99.6265
$Key_5(x_0{}' = x_0 - 10^{-14})$	99.6014	99.6170	99.6150	99.6160	99.6119	99.6201	99.6117	99.6282
$Key_6(y_0{}' = y_0 - 10^{-14})$	99.6323	99.6170	99.6140	99.6089	99.6292	99.6109	99.6209	99.6104
$Key_7(z_0{}' = z_0 - 10^{-14})$	99.6082	99.6063	99.6185	99.6053	99.6140	99.6145	99.6134	99.6053
$Key_8(w_0{}' = w_0 - 10^{-14})$	99.6262	99.6111	99.6063	99.6124	99.6121	99.6297	99.6141	99.6063

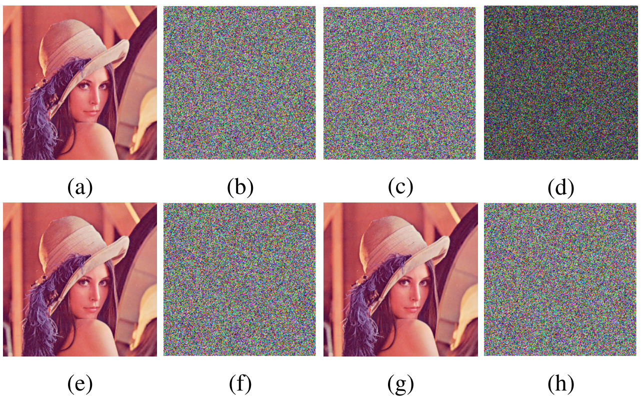


FIGURE 5. Key sensitivity analysis: (a) The plain Lena image; (b) The Lena image encrypted with key Key_0; (c) The Lena image encrypted with key Key_1; (d) The difference image between (b) and (c); (e) The decrypted image from (b) with the correct set of keys Key_0; (f) The decrypted image from (b) with the wrong set of keys Key_1; (g) The decrypted image from (c) with the correct set of keys Key_1; (h) The decrypted image from (c) with the wrong set of keys Key_0.

completely different cipher image as compared to the one obtained using a key without any change. For decryption, the original plain image should not be recovered from the cipher image unless 100% correct secret key set is used.

In order to measure the key sensitivity of the proposed encryption scheme, two key sets with very tiny difference have been used to encrypt the same input image. Suppose, $Key_0 = \{x_0, y_0, z_0, w_0, a, b, c, d, b_1, b_2, b_3, b_4\}$ is used to encrypt Lena plain image of the Figure 5a and as a result, we got the cipher image that is shown in Figure 5b. In order to demonstrate the key sensitivity, a very tiny change of 10^{-14} has been done in just a single variable x_0, *i.e.*, $x_0{}' = x_0 + 10^{-14}$ and we got a resulting new key set, Key_1. On encrypting the same Lena image (Figure 5a) using the new key set Key_1, the resultant cipher image is in the Figure 5c. Figure 5d shows an image obtained by taking the absolute value of the difference between the corresponding pixel intensity values in Figure 5b and 5c. An analysis of Figure 5b and 5c shows that a very small change of (10^{-14}) in only one variable has resulted in an approximately 99.62% differences among the encrypted images in terms of the intensities of pixels.

To demonstrate the key sensitivity in a more elaborate way, the rates of differences of pixel intensities between two encrypted images generated by Key_0 and $Key_t (t = 1, 2, \ldots 8)$ for all 8 test images (USC-SIPI Image Database) have been calculated by introducing a very slight difference in only one key between Key_0 and Key_t. The results have been listed in Table 5. It can be observed from the Table 5 that the average rate of difference between two encrypted images is more than 99.60%. Therefore, it can be safely concluded that a very minute change in the secret keys has resulted into the radically different encrypted images.

To demonstrate the key sensitivity for the decryption process, the keys Key_0 and Key_1 have been used to retrieve the original images from the encrypted images in Figures 5b and 5c respectively. The retrieved original images are shown in the Figures 5e-5h. The figures indicate that we can only recover the encrypted images successfully, if we employ the correct keys. Even a very tiny difference in the secret keys has an enormous impact on the decryption result and cannot provide the valid input image as demonstrated in Figure 5. Therefore, we can conclude that the proposed algorithm is high sensitive to a very minute key change during the encryption and decryption processes.

C. DIFFERENTIAL ATTACK (PLAINTEXT SENSITIVITY)

A potential hacker tries in every possible way to find the original image. One of the ways is to do a minor change in the input plain image, then encrypt both the plain images and find some meaningful relationship between both the plain as well as encrypted images. To tackle this situation, two measures Number of Pixels Change Rate (*NPCR*) and Unified Average Changing Intensity (*UACI*) have been developed to test the impact on the encrypted image after changing a single pixel in the plain-image. *NPCR* refers to the percentage of different intensities of pixels between the two images, *i.e.*, plain and cipher images. Whereas, *UACI* is the average intensity of the differences between the plain image and cipher image. They are formulated as follows:

$$NPCR = \frac{\sum_{i,j} D(a, b)}{M \times N} \times 100\% \tag{IV.1}$$

where M and N are the dimensions of the image. In the above formula, $D(a, b)$ is expressed as:

$$D(a, b) = \begin{cases} 1, & \text{if } C(a, b) \neq C'(a, b); \\ 0, & \text{if } C(a, b) = C'(a, b). \end{cases} \tag{IV.2}$$

TABLE 6. Average NPCR and UACI values for chosen images.

Images	NPCR(%)			UACI(%)		
	Red	Green	Blue	Red	Green	Blue
Lena	99.6216	99.6277	99.6189	33.4032	33.5397	33.4912
Baboon	99.6470	99.6372	99.6109	33.6047	33.4579	33.4247
Peppers	99.6872	99.6489	99.6117	33.4029	33.4851	33.5216
Tree	99.6185	99.6238	99.6395	33.6858	33.6383	33.5260
House	99.6155	99.6292	99.6267	33.4144	33.6090	33.4288
Beans	99.6143	99.6302	99.6140	33.4069	33.6900	33.4726
F16	99.6204	99.6155	99.6552	33.5097	33.5354	33.6907
Girl	99.6292	99.6597	99.6178	33.5893	33.4938	33.6882
Average	**99.6317**	**99.6340**	**99.6243**	**33.5021**	**33.5562**	**33.5305**

TABLE 7. A Comparison between the proposed algorithm and the others on the basis of average NPCR and UACI on the Lena image.

Algorithm	Average NPCR(%)	Average UACI(%)
Proposed Algorithm	99.6300	33.5296
Ref. [4]	99.60	33.48
Ref. [21]	99.2172	33.4055
Ref. [35]	99.6037	33.4463
Ref. [36]	99.59	33.41
Ref. [37]	99.6000	33.4316
Ref. [50]	99.5956	33.4588
Ref. [51]	99.6173	33.4249
Ref. [52]	99.5991	33.4650

$$UACI = \frac{1}{M \times N}\left[\sum_{a,b} \frac{|C(a,b) - C'(a,b)|}{255}\right] \times 100\% \quad \text{(IV.3)}$$

C and C' are respectively the cipher images before and after one pixel of the plain image is changed.

The values of the differential attack metrics of *NPCR* and *UACI* have been shown in the Table 6.

The average values of these metrics are 99.6300%($NPCR$) and 33.5296%($UACI$) clearly demonstrating the defiance of potential differential attacks. Further according to the Table 7, the proposed algorithm performs better than the algorithms described in [4], [21], [35]–[37], [50]–[52] as far as *NPCR* and *UACI* for the Lena image are concerned.

D. RANDOMNESS OF GENERATED KEYSTREAMS

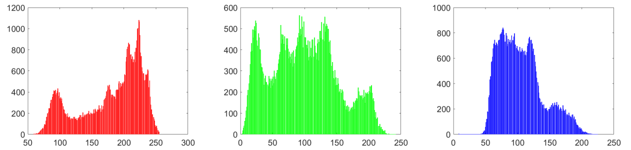
The proposed encryption scheme is efficient in a way that it uses less chaotic data. To see that generated keystreams (mask images) are totally random and have absolutely no relation with each other. NPCR and UACI are calculated between them. Table 8 shows that $NPCR > 99.6$ and Table 9 shows that $UACI > 33.4$. Thus, the minimal resources are used without compromising the security.

E. STATISTICAL ANALYSIS

Enduring the statistical attacks should be one of the built-in features of a good image encryption scheme. Histogram and correlation attacks are normally included in the statistical attacks.

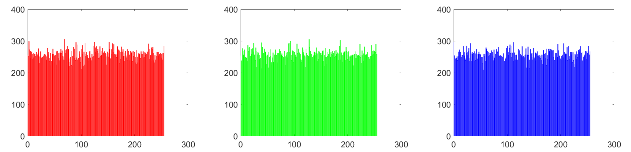
1) HISTOGRAM

An image's histogram depicts the pixel intensities distribution. A potential intruder can exploit the histogram of an image to get some meaningful information for his malicious intentions. So, after the encryption process, the histogram of a ciphered image should be as much uniform as possible. Figures 6 and 7 depict the histograms of Lena image for its plain and encrypted versions respectively. In the plain image, the bars are not uniform, rather they are fluctuating. Hence full of information for some potential adversary. Whereas for the ciphered image, the bars are very uniform giving no meaningful information to an adversary. These uniform bars give a tough time to a potential hackers in launching any statistical attack.



(a) (b) (c)

FIGURE 6. Plain Lena image Histograms in the (a) red, (b) green, (c) blue components.



(a) (b) (c)

FIGURE 7. Cipher Lena image histograms in the (a) red, (b) green, (c) blue components.

Variance is a measure of average spread of data/information around the mean value. A low the variance means a lower differences between data point in a distribution. Thus, a small variance value is desirable for a uniform histogram. Table 10 gives the variance values for the histograms of the encrypted test images (Lena, Baboon, Peppers, Tree, House, Beans, F16 and Girl).

The variance values in the first column of Table 10 have been calculated through the initial key set Key_0, while the ones in the remaining columns have been obtained by changing only one secret key of Key_0. These secret keys are $Key_t (t = 1, 2, \ldots 8)$ as discussed in the section IV-B. It is clear from the table that the variance values of the ciphered-images are about 800, which are less than those in [21], [36], [49], [53]. In contrast, variance values for the input plain images are about 86,488. Thus, the proposed scheme has a good encryption effect by reducing the variance values.

2) CORRELATION ANALYSIS

Generally, an image consists of a large number of pixels. Usually, the adjacent pixels are very similar, so they have very high correlation with each other. An ideal image encryption algorithm should break this inherent correlation between the adjacent pixels such that an adversary could not use this information to predict to the original image. Normally, the

TABLE 8. The NPCR between the keystreams.

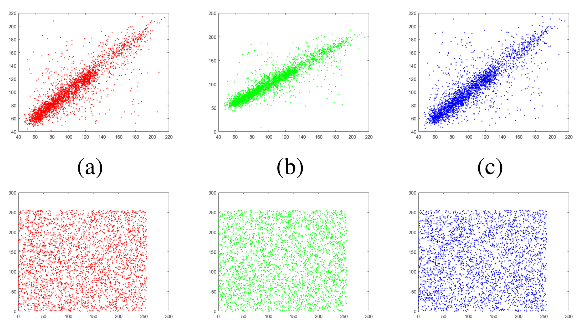
	K_1	K_2	K_3	K_4	K_5	K_6	K_7
K_1	-	99.6201	99.6322	99.6421	99.6529	99.6820	99.6724
K_2		-	99.6712	99.6923	99.6542	99.7851	99.7123
K_3			-	99.7143	99.7532	99.7620	99.7734
K_4				-	99.7834	99.6504	99.6930
K_5					-	99.7834	99.7043
K_6						-	99.7865
K_7							-

TABLE 9. The UACI between the keystreams.

	K_1	K_2	K_3	K_4	K_5	K_6	K_7
K_1	-	33.7271	33.4620	33.4320	33.5029	33.4891	33.6908
K_2		-	33.5443	33.4623	33.4942	33.4450	33.5523
K_3			-	33.5843	33.5332	33.5790	33.6134
K_4				-	33.5683	33.5144	33.4990
K_5					-	33.5622	33.6598
K_6						-	33.4799
K_7							-

TABLE 10. The variance values of histograms of the encrypted images by employing different keys.

Images	Key_0	Key_1	Key_2	Key_3	Key_4	Key_5	Key_6	Key_7	Key_8
Lena	799.1979	763.1589	786.3099	788.1797	722.3490	820.2604	800.8255	776.5391	811.4609
Baboon	814.5417	777.4557	779.8229	822.4635	753.3333	797.7526	751.2031	817.3125	802.1328
Peppers	741.2708	772.7552	748.7630	787.2109	737.8255	791.1589	775.4427	747.8698	838.9036
Tree	779.6563	798.2500	849.9740	799.5781	818.2083	756.3932	810.4792	784.1797	799.3047
House	765.2083	793.5885	818.3203	794.6901	812.7578	794.7708	835.7500	784.8307	776.6667
Beans	776.7370	778.5990	731.7344	834.3307	821.7109	798.7552	772.6510	754.9870	754.7344
F16	832.2891	815.5417	827.9115	858.2813	814.7135	795.4453	761.7969	783.3516	834.4063
Girl	813.5078	787.2422	742.3516	818.4714	774.5833	790.8776	766.5339	816.3438	757.4427
Average	**790.3011**	**787.2422**	**785.6485**	**812.9007**	**781.9352**	**793.1768**	**784.3353**	**783.1768**	**796.8815**



(a) (b) (c)

(d) (e) (f)

FIGURE 8. Correlation distribution of adjacent pixels: (a) Horizontally in the red channel of the plain Lena image; (b) Vertically in the green channel of the plain Lena image; (c) Diagonally in the blue channel of the plain Lena image; (d) Horizontally in the red channel of the cipher Lena image; (e) Vertically in the green channel of the cipher Lena image; (f) Diagonally in the blue channel of the cipher Lena image.

correlation between horizontal, vertical and diagonal pixels is calculated. The mathematical formula for calculating this correlation is given below [54]:

$$CC = \frac{N\sum_{j=1}^{N}(x_j \times y_j) - \sum_{j=1}^{N} x_j \times \sum_{j=1}^{N} y_j}{\sqrt{\left(N\sum_{j=1}^{N} x_j^2 - \left(\sum_{j=1}^{N} x_j\right)^2\right)\left(N\sum_{j=1}^{N} y_j^2 - \left(\sum_{j=1}^{N} y_j\right)^2\right)}}, \tag{IV.4}$$

TABLE 11. Correlation coefficient for the red, green and blue channels of the plain Lena image and its encrypted version.

Image	Component	Correlation direction		
		Horizontal	Vertical	Diagonal
Original Lena image	Red	0.9433	0.9647	0.9201
	Green	0.9205	0.9518	0.9040
	Blue	0.8733	0.9106	0.8487
Encrypted Lena image	Red	0.0071	0.0009	-0.0043
	Green	-0.0005	-0.0034	0.0026
	Blue	-0.0029	0.0045	0.0008

where x and y are the pixel intensity values of two adjacent pixels and the number of pixels has been represented by N.

Figure 8 depicts the correlation distribution of adjacent pixels in horizontal, vertical and diagonal orientations for the plain and ciphered Lena image.

The Correlation Coefficient (CC) between two adjacent pixels for plain and encrypted image of Lena have been shown in the Table 11. It can be seen from the Table 11 that the CC between adjacent pixels of the plain input image is nearly equal to 1 showing a high positive correlation between adjacent pixels. It is also clear from Figire 8a, 8b and 8c. Whereas, they are almost equal to 0 in case of ciphered image, showing no or very low correlation between adjacent pixels. Both the Table 11 and Figure 8 clearly demonstrate that the proposed encryption algorithm has broken the correlation among the adjacent pixels between the plain image and ciphered image and does not give even an iota of resemblance between the

TABLE 12. Comparison of correlation coefficients between Proposed scheme and other encryption schemes.

Image	Encryption algorithm	Correlation direction		
		Horizontal	Vertical	Diagonal
Original Lena image		0.9124	0.9424	0.8909
Encrypted Lena image	Proposed scheme	0.0012	0.0007	-0.0003
	Ref. [21]	-0.0082	-0.0128	-0.0012
	Ref. [37]	0.0019	-0.0035	-0.0013
	Ref. [50]	-0.0021	0.0009	0.0003
	Ref. [51]	0.0058	0.0033	0.0010
	Ref. [55]	0.0022	0.0001	-0.0017
	Ref. [56]	0.0014	0.0014	0.0045
	Ref. [57]	-0.0116	-0.0001	0.0035
	Ref. [58]	0.0023	0.0019	0.0011

two. In other words, this signals to a successful breakage of the correlation of adjacent pixels in the plain image. Table 12 has compared the correlation of the original Lena image with its different encrypted versions generated by different encryption algorithms. The proposed algorithm has given the better results as compared to the encryption algorithms reported in [21], [37], [50], [51], [55]–[58].

F. INFORMATION ENTROPY

Entropy is used to measure the degree of randomness and unpredictability of an information source. In 1949, Shannon [59] developed a mathematical formula to measure information entropy:

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) log \frac{1}{p(m_i)} \tag{IV.5}$$

In the above formula, $H(m)$ represents the information entropy of m that is an information source. $p(m_i)$ is the probability of the symbol m_i. The maximum value of the information entropy comes out to be 8, if we have an absolutely random image consisting of 256 gray values. The more close to 8 the value of entropy is, the better it always is. Table 13 shows the values of the entropy for our chosen images. The average of the values for all the images is nearly equal to the ideal value 8. Hence, the proposed scheme is very much resistant to any kind of entropy attack. Table 13 further gives a comparison with some other schemes. The proposed scheme performs better than those in [7], [21], [60], [61] as far as information entropy is concerned.

G. PEAK SIGNAL-TO-NOISE RATIO ANALYSIS

To create a maximum divergence between an input plain image and its encrypted version is the chronic idea of any image encryption scheme. To yardstick this measure, researchers have given the idea of Peak-Signal-to-Noise Ratio (PSNR). This metric measures the difference between the input image and its encrypted version. Mathematically it is defined as:

$$PSNR = 20log_{10}(\frac{255}{\sqrt{MSE}})dB \tag{IV.6}$$

$$MSE = \frac{1}{M \times N} \sum_{i,j} (P_0(i,j) - P_1(i,j))^2 \tag{IV.7}$$

where M and N refer to the width and height of the test image respectively. $P_0(i,j)$ and $P_1(i,j)$ are the intensity values of the pixels of the original and ciphered images respectively. Further, MSE is the mean squared error between the two images, *i.e.*, the original and the encrypted one. The larger value of MSE will generate a smaller value of $PSNR$, which in turn renders a better encryption security.

Table 14 gives the $PSNR$ values of the proposed and the other encryption techniques. It can be observed from the table that the $PSNR$ values between the plain and the decrypted images are (∞). It signals that there is no difference between the plain input image and its decrypted version owing to the fact that $MSE = 0$. This refers to the fact that the encryption and decryption algorithms do not lose any pixel intensity value. Results in Table 14 show that the $PSNR$ value given by the proposed scheme is the smallest when it is compared with some other schemes [62]–[64]. Thus, the proposed scheme has a better security.

H. NOISE AND DATA LOSS ATTACKS

Practically, the transmission of images may get polluted with some kind of noise during its transfer over the network or multi-media network. Furthermore, during the transmission, some portion of the image may be lost. A good encryption scheme should endure the noise and data loss attacks. Figures. 9a to 9c depict the encrypted images polluted by Pepper & Salt noise with various noise densities of 0.1, 0.2 and 0.3.

Figures. 9d to 9f show the decrypted images through the usage of the proposed algorithm. It is clear that the original images are easily discerned from the noisy ciphered images. Figure 10a plots the encrypted Baboon image.

TABLE 13. The comparison of information entropy analysis using different encryption schemes.

Encryption schemes	Images	Original			Encrypted		
		Red	Green	Blue	Red	Green	Blue
Proposed scheme	Lena	7.2507	7.5931	6.9659	7.9970	7.9970	7.9976
	Baboon	7.6942	7.4637	7.7443	7.9974	7.9968	7.9970
	Peppers	7.3402	7.4470	7.0569	7.9970	7.9974	7.9973
	Tree	7.2104	7.4136	6.9207	7.9970	7.9970	7.9976
	House	6.4311	6.5389	6.2320	7.9971	7.9974	7.9972
	Beans	5.7920	6.2195	6.7986	7.9973	7.9973	7.9967
	F16	6.7106	6.7962	6.2001	7.9966	7.9974	7.9972
	Girl	5.7150	5.3738	5.7117	7.9972	7.9968	7.9975
	Average	**6.7680**	**6.8557**	**6.7038**	**7.9971**	**7.9971**	**7.9973**
Ref. [7]	Lena				7.9973	7.9969	7.9971
Ref. [21]	Lena				7.9892	7.9896	7.9896
Ref. [37]	Lena				7.9971	7.9973	7.9973
Ref. [50]	Lena					7.9972	
Ref. [51]	Lena				7.9973	7.9975	7.9975
Ref. [60]	Lena				7.9895	7.9894	7.9894
Ref. [61]	Lena				7.9943	7.9943	7.9942

TABLE 14. A comparison of the PSNR results: 'O, C and D' represent the original, cipher and decrypted images.

		Lena	Baboon	Peppers	Tree	House	Beans	F16	Girl
Proposed	PSNR (O-D)	∞	∞	∞	∞	∞	∞	∞	∞
	PSNR (O-C)	7.8694	8.7855	9.0991	8.6982	9.8163	8.4897	8.1482	9.8564
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							

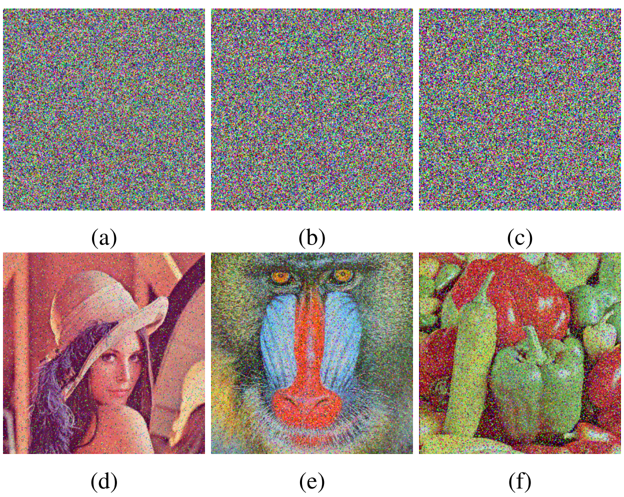


FIGURE 9. The Pepper and Salt noise attack by adding different densities: (a) Cipher Lena image with 0.1 density; (b) Cipher Baboon image with 0.2 density; (c) Cipher Pepper image with 0.3 density; (d) Decrypted Lena image from (a); (e) Decrypted Baboon image from (b); (f) Decrypted Pepper image from (c).

In Figure 10b, a block of data from the Baboon ciphered image with size 80 × 170 has been cropped out to depict the data loss. The ciphered image with partially lost data is

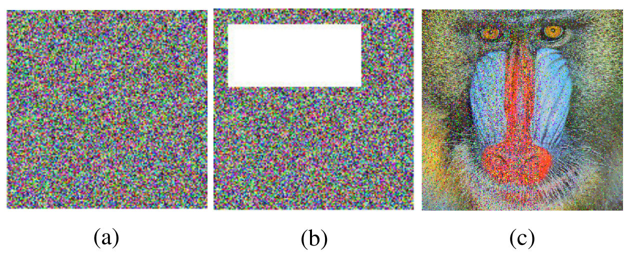


FIGURE 10. Data loss attack: (a) Cipher Baboon image; (b) Cipher Baboon image with a 80 × 170 data loss; (c) Decrypted Baboon image from (d).

decrypted using the proposed algorithm. Figure 10c shows the decrypted image. It is clear that the decrypted image still bears so much information that it may be easily recognized. Thus, we can conclude that the proposed scheme is impervious and resistant to any noise and data loss threat.

I. COMPUTATIONAL TIME AND COMPLEXITY

After security concerns, a good image cipher should have a fast speed for a real life application. Our algorithm has been coded and compiled under Intel(R) Core(TM) i5-4210U CPU @ 1.70 GHz 2.40 GHz, 8 GB memory, Windows 10, MATLAB R2016a. In the literature, there are two ways

TABLE 15. Encryption throughput of the proposed algorithm and some other schemes.

Algorithm	Speed (MBit/s)
Proposed	2.68
Ref. [7]	1.28
Ref. [52]	2.39
Ref. [53]	0.21
Ref. [65]	0.16

for judging the computational time of some algorithm, *i.e.*, empirical and theoretical. In empirical method, the computational time of some algorithm is physically measured through some gadget like stopwatch *etc.* We have calculated the encryption throughput (ET) for the proposed algorithm. ET is calculated by dividing the size of the given image in Megabits to the total time taken for its encryption. The average ET for the eight images is 2.68 MBit/s. The Table 15 shows that the proposed encryption procedure is better than [7], [52], [53], [65].

In theoretical analysis for calculating time-complexity, the mathematical theory of Asymptotics [66] is employed. For this purpose, two aspects will be considered, *i.e.*, generation of the chaotic data and the encryption algorithm. Chaotic data K is being generated by applying CDSVSP to the array S described in the section III. Its complexity comes out to be $\Theta(4MN)$ where (M, N) is the size of the given input image. The complexity of both the decimal diffusion and permutation operations is $\Theta(6MN)$. The complexity of DNA conversion and decimal conversion is $\Theta(6MN)$. Moreover, the complexity of DNA level diffusion is $\Theta(3MN)$. So, the total complexity comes out to be $\Theta(19MN)$, which is better than $\Theta(24MN)$ [21] and $O(64\ N^2)$ [67]. If the image taken is a square of size N, then it becomes $\Theta(19N^2)$.

V. CONCLUSION

The simulations and experimental test results show that the proposed encryption scheme is successful in maintaining high standards of security even though using much less resources in comparison with the existing encryption algorithms. The chaotic data obtained from CDS is of high quality and CDSVSP is quite efficient in generating entirely random selected sequences from it. Also, the pixels of plain image are directly involved in all the encryption steps like DNA conversion, diffusion in decimal, thus DNA sequences have high plaintext sensitivity, making it impossible to break by plaintext attacks and statistical attacks. The full potential of DNA model is demonstrated by using different conversion rules and DNA operations for each pixel that further randomise the ciphered image, improving the existing encryption algorithms based on DNA model. Thus, due to lower time complexity, higher efficiency and security, we are justified in claiming that the proposed algorithm is more suitable for real-time and real-life applications in the filed of image security, multi-media networks and on-line systems.

Funding This study is partially funded by Universal College of Learning, New Zealand (PD20/1323).

Conflict of interest The authors declare that they have no conflict of interest.

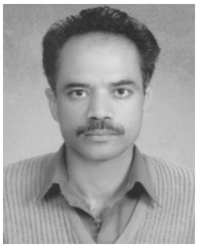
REFERENCES

[1] Z. Bashir, J. Wątróbski, T. Rashid, S. Zafar, and W. Sałabun, "Chaotic dynamical state variables selection procedure based image encryption scheme," *Symmetry*, vol. 9, no. 12, p. 312, 2017.
[2] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dyn.*, vol. 92, no. 2, pp. 305–313, Apr. 2018.
[3] G. Ye and J. Zhou, "A block chaotic image encryption scheme based on self-adaptive modelling," *Appl. Soft Comput.*, vol. 22, pp. 351–357, Sep. 2014.
[4] 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.
[5] 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.
[6] X.-Y. Wang and Z.-M. Li, "A color image encryption algorithm based on hopfield chaotic neural network," *Opt. Lasers Eng.*, vol. 115, pp. 107–118, Apr. 2019.
[7] 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.
[8] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-box," *Inf. Sci.*, vol. 450, pp. 361–377, Jun. 2018.
[9] Y. Zhou, Z. Hua, C.-M. Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2001–2012, Sep. 2015.
[10] Z. Liu, Y. Zhang, W. Liu, F. Meng, Q. Wu, and S. Liu, "Optical color image hiding scheme based on chaotic mapping and Hartley transform," *Opt. Lasers Eng.*, vol. 51, no. 8, pp. 967–972, Aug. 2013.
[11] M. H. Annaby, M. A. Rushdi, and E. A. Nehary, "Color image encryption using random transforms, phase retrieval, chaotic maps, and diffusion," *Opt. Lasers Eng.*, vol. 103, pp. 9–23, Apr. 2018.
[12] N. Zhou, Y. Hu, L. Gong, and G. Li, "Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations," *Quantum Inf. Process.*, vol. 16, no. 6, p. 164, Jun. 2017.
[13] N. Zhou, X. Yan, H. Liang, X. Tao, and G. Li, "Multi-image encryption scheme based on quantum 3D Arnold transform and scaled Zhongtang chaotic system," *Quantum Inf. Process.*, vol. 17, no. 12, p. 338, Dec. 2018.
[14] C. Li, D. Lin, and J. Lu, "Cryptanalyzing an image-scrambling encryption algorithm of pixel bits," *IEEE Multimedia Mag.*, vol. 24, no. 3, pp. 64–71, Aug. 2017.
[15] C. Li, "Cracking a hierarchical chaotic image encryption algorithm based on permutation," *Signal Process.*, vol. 118, pp. 203–210, Jan. 2016.
[16] 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.
[17] 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.
[18] 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.
[19] W. Liu, K. Sun, Y. He, and M. Yu, "Color image encryption using three-dimensional sine ICMIC modulation map and DNA sequence operations," *Int. J. Bifurcation Chaos*, vol. 27, no. 11, Oct. 2017, Art. no. 1750171.
[20] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1457–1466, May 2012.
[21] 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.
[22] Q. Zhang, L. Liu, and X. Wei, "Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps," *Int. J. Electron. Commun.*, vol. 68, no. 3, pp. 186–192, Mar. 2014.
[23] 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.
[24] 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.

[25] 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.
[26] A. Y. Niyat and M. H. Moattar, "Color image encryption based on hybrid chaotic system and DNA sequences," *Multimedia Tools Appl.*, vol. 79, nos. 1–2, pp. 1497–1518, Jan. 2020.
[27] Q. Zhang and X. Wei, "A novel couple images encryption algorithm based on DNA subsequence operation and chaotic system," *Optik*, vol. 124, no. 23, pp. 6276–6281, Dec. 2013.
[28] X.-Y. Wang, P. Li, Y.-Q. Zhang, L.-Y. Liu, H. Zhang, and X. Wang, "A novel color image encryption scheme using DNA permutation based on the Lorenz system," *Multimedia Tools Appl.*, vol. 77, no. 5, pp. 6243–6265, Mar. 2018.
[29] 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.
[30] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, "A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm SHA-2," *Nonlinear Dyn.*, vol. 83, no. 3, pp. 1123–1136, Feb. 2016.
[31] L.-M. Zhang, K.-H. Sun, W.-H. Liu, and S.-B. He, "A novel color image encryption scheme using fractional-order hyperchaotic system and DNA sequence operations," *Chin. Phys. B*, vol. 26, no. 10, Sep. 2017, Art. no. 100504.
[32] Y.-Q. Zhang, X.-Y. Wang, J. Liu, and Z.-L. Chi, "An image encryption scheme based on the MLNCML system using DNA sequences," *Opt. Lasers Eng.*, vol. 82, pp. 95–103, Jul. 2016.
[33] Y. Zhang, D. Xiao, W. Wen, and K.-W. Wong, "On the security of symmetric ciphers based on DNA coding," *Inf. Sci.*, vol. 289, pp. 254–261, Dec. 2014.
[34] X. Wu, K. Wang, X. Wang, and H. Kan, "Lossless chaotic color image cryptosystem based on DNA encryption and entropy," *Nonlinear Dyn.*, vol. 90, no. 2, pp. 855–875, Oct. 2017.
[35] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, and B.-Q. Yang, "Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption," *Signal Process.*, vol. 142, pp. 340–353, Jan. 2018.
[36] 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.
[37] B. Norouzi and S. Mirzakuchaki, "An image encryption algorithm based on DNA sequence operations and cellular neural network," *Multimedia Tools Appl.*, vol. 76, no. 11, pp. 13681–13701, Jun. 2017.
[38] 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.
[39] 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.
[40] Y. Liu, J. Tang, and T. Xie, "Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map," *Opt. Laser Technol.*, vol. 60, pp. 111–115, Aug. 2014.
[41] L. Liu, Q. Zhang, and X. Wei, "A RGB image encryption algorithm based on DNA encoding and chaos map," *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1240–1248, Sep. 2012.
[42] M. Kumar, A. Iqbal, and P. Kumar, "A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie–Hellman cryptography," *Signal Process.*, vol. 125, pp. 187–202, Aug. 2016.
[43] A. Akhavan, A. Samsudin, and A. Akhshani, "Cryptanalysis of an image encryption algorithm based on DNA encoding," *Opt. Laser Technol.*, vol. 95, pp. 94–99, Oct. 2017.
[44] C. Yong and Y. Yun-Qing, "A new four-dimensional chaotic system," *Chin. Phys. B*, vol. 19, no. 12, 2010, Art. no. 120510.
[45] N. Masuda and K. Aihara, "Cryptosystems with discretized chaotic maps," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 1, pp. 28–40, Jan. 2002.
[46] *IEEE Standard For Binary Floating-Point Arithmetic*, IEEE Standard 754-1985, Oct. 1985.
[47] I. Öztürk and R. Kiliç, "Cycle lengths and correlation properties of finite precision chaotic maps," *Int. J. Bifurcation Chaos*, vol. 24, no. 9, 2014, Art. no. 1450107.
[48] S. Li, G. Chen, and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," *Int. J. Bifurcation Chaos*, vol. 15, no. 10, pp. 3119–3151, Oct. 2005.
[49] A. 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. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0030402617311695
[50] 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.
[51] H. G. Mohamed, D. H. ElKamchouchi, and K. H. Moussa, "A novel color image encryption algorithm based on hyperchaotic maps and mitochondrial DNA sequences," *Entropy*, vol. 22, no. 2, p. 158, 2020.
[52] T. Hu, Y. Liu, L.-H. Gong, S.-F. Guo, and H.-M. Yuan, "Chaotic image cryptosystem using DNA deletion and DNA insertion," *Signal Process.*, vol. 134, pp. 234–243, May 2017.
[53] 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.
[54] 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.
[55] A. Y. Niyat, M. H. Moattar, and M. N. Torshiz, "Color image encryption based on hybrid hyper-chaotic system and cellular automata," *Opt. Lasers Eng.*, vol. 90, pp. 225–237, Mar. 2017.
[56] S. Rajagopalan, S. Sharma, S. Arumugham, H. N. Upadhyay, J. B. B. Rayappan, and R. Amirtharajan, "YRBS coding with logistic map–a novel Sanskrit aphorism and chaos for image encryption," *Multimedia Tools Appl.*, vol. 78, no. 8, pp. 10513–10541, Apr. 2019.
[57] 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, Nov. 2019.
[58] R. Enayatifar, A. H. Abdullah, I. F. Isnin, A. Altameem, and M. Lee, "Image encryption using a synchronous permutation-diffusion technique," *Opt. Lasers Eng.*, vol. 90, pp. 146–154, Mar. 2017.
[59] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.
[60] X. Wu, J. Kurths, and H. Kan, "A robust and lossless DNA encryption scheme for color images," *Multimedia Tools Appl.*, vol. 77, no. 10, pp. 12349–12376, May 2018.
[61] 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.
[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] S. Cai, L. Huang, X. Chen, and X. Xiong, "A symmetric plaintext-related color image encryption system based on bit permutation," *Entropy*, vol. 20, no. 4, p. 282, 2018.
[66] H. Joe, "Asymptotic efficiency of the two-stage estimation method for copula-based models," *J. Multivariate Anal.*, vol. 94, no. 2, pp. 401–419, Jun. 2005.
[67] 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, Oct. 2018.

M. G. ABBAS MALIK received the Ph.D. degree in computer science with specialization in artificial intelligence and statistical learning from Université Grenoble Aples (UGA), France, the master's degree in computational linguistics from Univesité de Paris 7, Denis Didrot, France, and the Master of Science degree in computer science from the University of Punjab, Pakistan. Since 2009, he has been serving in various universities in France, Pakistan, Saudi Arabia, and New Zealand. He is currently working with the Universal College of Learning, New Zealand. His primary research interests include artificial intelligence, machine learning, data analytics, natural language processing, computer vision, augmented reality, and virtual reality.

ZIA BASHIR is currently an Assistant Professor of mathematics with the Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan. His research interests include function spaces, fuzzy Analysis, decision making models, and cryptography.



NADEEM IQBAL received the M.Phil. degree in computational science and engineering from 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 bachelor's and master's students. His research interests include images cryptography, computer graphics, and philosophy of mathematics.

MD. ATHAR IMTIAZ received the master's and Engineering degrees in computing, and the Ph.D. degree in computer science from the University of Auckland. He is currently employed as a Lecturer with Massey University, Palmerston North. His research interests include HCI, machine learning, education technology, and software engineering. His area of specialization is human–computer interaction (HCI). He has industry work experience in some of the largest software multinational companies.