IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 2, 2021, accepted August 12, 2021, date of publication August 18, 2021, date of current version August 31, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106028

On the Image Encryption Algorithm Based on the Chaotic System, DNA Encoding, and Castle

NADEEM IQBAL[1], RIZWAN ALI NAQVI[2], MUHAMMAD ATIF[1], MUHAMMAD ADNAN KHAN[3], MUHAMMAD HANIF[4], SAGHEER ABBAS[5], AND DILDAR HUSSAIN[6]

[1]Department of Computer Science & IT, The University of Lahore, Lahore 54590, Pakistan
[2]Department of Unmanned Vehicle Engineering, Sejong University, Seoul 05006, South Korea
[3]Pattern Recognition and Machine Learning Laboratory, Department of Software, Gachon University, Seongnam 13557, South Korea
[4]Department of Computer Science, Bahria University, Lahore Campus, Lahore 54600, Pakistan
[5]Department of Computer Science, National College of Business Administration & Economics, Lahore 54660, Pakistan
[6]School of Computational Sciences, Korea Institute for Advanced Study (KIAS), Seoul 02455, Republic of Korea

Corresponding authors: Dildar Hussain (hussain@kias.re.kr) and Muhammad Adnan Khan (adnan@gachon.ac.kr)

This work was supported in part by Korea Institute for Advanced Study under Grant CG076601, and in part by Sejong University and Gachon University Faculty Research Fund.

ABSTRACT Hundreds of image encryption algorithms have been developed for the security and integrity of images through the combination of DNA computing and chaotic maps. This combination of the two instruments is not sufficient enough to thwart the potential threats from the cryptanalysis community as the literature review suggests. To inject more robustness and security stuff, a novel image encryption scheme has been written in this research by fusing the chaotic system, DNA computing and Castle —a chess piece. As the plain image is input, its pixels are shifted to the scrambled image at the randomly chosen pixel addresses. This scrambling has been realized through the routine called Image Scrambler using Castle (ISUC). Castle randomly moves on the hypothetical large chessboard. Pixels taken from the plain image are shifted to the addresses of the scrambled image, where Castle lands in each iteration. After the plain image is scrambled, it is DNA encoded. Two mask images are also DNA encoded. Then to throw the diffusion effects in the cipher, DNA Addition and DNA XOR operations between the DNA encoded pixels data and the DNA encoded mask images have been conducted. Next, the pixels data are converted back into their decimal equivalents. Four dimensional chaotic system has been used to get the chaotic vectors. The hash codes given by the SHA-256 function have been used in the cipher to introduce the plaintext sensitivity in its design. We got an information entropy of 7.9974. Simulation carried out through the machine, and the thorough security analyses demonstrate the good security effects, defiance to the varied attacks from the cryptanalysis community, and the bright prospects for some real world application of the proposed image cipher.

INDEX TERMS Castle, chaotic system, chess, decryption, DNA encoding, encryption.

I. INTRODUCTION

The science and art of cryptography is as old as the human civilization itself. As the societies grew, tribes of human beings started to be organized into different kingdoms. Naturally, all these developments led to the new ideas and phenomena like politics, superiority, power and battles *etc*. These advancements propelled people to communicate with each other secretly and with the selective recipients. Not only the science of cryptography got founded but at the same time continued to be evolved. The roots of cryptography are found in the ancient Roman and Egyptian civilizations. Hundreds of cryptosystems have been developed since then. Whatever the cryptosystem is, the basic idea stays the same, *i.e.*, to communicate secretly in the presence of an eavesdropper.

In history, certain zeitgeists have always been characterizing the cultures all over the world. No doubt, today's is information and communication technologies. Images have assumed a central place in all of this. Be it academia, industry, research and science facilities, diplomatic and government settings, art, design, medicine, social life to name a few, these images have infiltrated in the entire cross sections of the societies. People from all over the world routinely store these images on different gadgets like hard disks, tablets,

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

PDAs *etc*., or transmit them to their acquaintances through the public networks like Internet. Sometimes, these images happen to be extremely secret and confidential, a picture of some spy, blueprint of some new industrial product, for instance. So, to safeguard them from the ubiquitous hackers and antagonists, foolproof steps must be adopted. The current study is a step forward in this direction. One will find plethora of ciphers like AES, DES, RSA *etc.* in the literature. Unfortunately, they can't be applied over the images since their area of application was text data [1], [2]. Since the images have radically different characteristics like bulky volume, high redundancy and strong inter-pixel bonding *etc.*, this is the reason that these ciphers can't be applied over them. So, an entirely different setting is required for the solution of this problem. Fortunately, the chaotic maps have proved very handy for the enterprise of image encryption. Many ciphers have been developed through the fusion of these maps with DNA computing. Some image ciphers used chaotic maps with other instruments and concepts like knight [3], king [4], 15-puzzle [5], fractals [6], Latin cubes [7], Nine palace [8], swapping operations [9], circular shift operations [10] *etc.* The current study has tried to seek the harmonious complementarities existing among the three giants, *i.e.*, chaos, DNA computing and chess piece Castle in order to develop a more secured cipher. We describe them one by one in the following.

Chaos is a mathematical theory [11], which investigates the behavior of dynamical systems, being highly dependent to the system parameters and the initial conditions. It's an interdisciplinary enterprise, which states that given the apparent random behavior of the chaotic systems, there exist strictly deterministic patterns in the form of constant feedback loops, fractals and self-organization. This theory maintains that a very minute change in one of the parameters and states can bring a huge difference in the later states. It means there exists a sensitive dependence on the system parameters and the initial conditions. There is a very interesting metaphor, the *butterfly effect*, regarding chaos that goes like this, "*if a butterfly flaps its wing in Brazil, it can cause a hurricane in the Texas.*" This theory was given by Edward Lorenz [12]. Strictly complying with the idea of chaos theory, mathematicians have given a large number of chaotic maps/systems. These maps are basically the set of mathematical equations which provide the random numbers. These maps have proved very useful in developing a large number of image cryptosystems due to their excellent inbuilt properties as described earlier. Recently, a 2D modular chaotification system has been developed in order to improve the chaos complexity [13]. In the reported work, the authors have contended that the existing chaotic maps may be plagued with defects like incomplete output distributions and discontinuous chaotic ranges. Such defects may disrupt the working of the potential chaos-based applications. Owing to the fact that the modular operation is a bounded transform, the map introduced in [13] may spawn the behaviors of chaoticity in a relatively wider parameters ranges whereas the existing maps do not demonstrate such behavior. In a yet another research work [14], a new 2D chaotic system has been developed called 2D-LSM. This new 2D chaotic system has wide and continuous chaotic range than many of the existing 2D chaotic maps. This map has been employed to devise an image cipher which exploits the properties of Latin square and color image in their entirety.

DNA stands for deoxyribonucleic acid. This is the germ plasm of all life styles [15]. DNA is basically a type of biological macromolecule and is comprised of nucleotides. These nucleotides contain a single base and there are four types of bases, *i.e.*, A(Adenine), C(Cytosine), G(Guanine) and T(Thymine). A DNA is a double-stranded molecule occurring in nature. The two DNA strands complementing each other are held with each other to make a double-helix structure by hydrogen bonds between the complementary bases of A and T (or C and G). Watson and Crick, the two scientists, discovered this double-helix structure. Therefore, this structure is termed as Watson-Crick complementarity [16].

Chess is a game whom history has made synonymous with the ingenuity. This has become an epitome of ruse, collusion, plot, maneuvering, rigging, planning, strategy and other terms of such character. We believe that the chess has an immense potential for the enterprise of cryptography. Each player/piece of chess moves in its unique way. Knight, a chess piece, has already been used in the different image ciphers for image scrambling [3]. One of the drawbacks of using knight for scrambling is that it moves in two and half boxes of the chess, so, the scrambling effects will be localized in the given input plain image. To address this problem, we have conceived the idea of using Castle, another chess piece, for the project of scrambling. Castle moves in the entire chessboard in one go both horizontally and vertically. In this way, scrambling effects will spread all over the image leading to the greater security.

As described earlier, through the fusion of chaotic systems and DNA computing, many image encryption algorithms have been produced. We will describe here few of them. An RGB image encryption scheme was proposed by [5]. The authors of this scheme used chaotic tent map, the intertwining logistic map, DNA computing and the game of 15-puzzle in their proposed scheme. Scrambling was carried out by the 15-puzzle game and the diffusion effects were embedded through the DNA encoding. One more scheme for the image encryption using the Henon-sine map and DNA encoding was given by [17]. In the confusion process, the permutation-diffusion architecture was employed. Besides, the DNA encoding and XOR operation were used in the diffusion process. The substitution boxes were used in diffusion process to make DNA level encryption more complicated and hence more secured. The results showed that their scheme was secured enough to avert the varied attacks. An another algorithm for the images encryption based on fractional Fourier transformation, chaotic map and DNA was given by [18]. Lorenz map chaotic stream was utilized in the random phase masks generation. The plain image was transformed into DNA matrix, which was then joined with the random phase mask using the fractional Fourier transform.

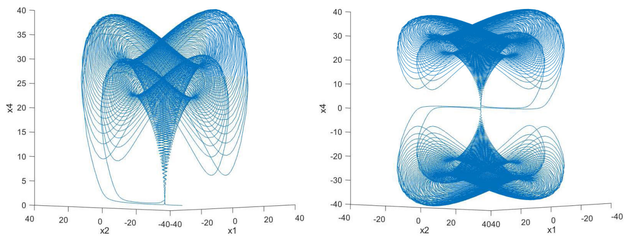
The results showed that their proposed scheme could safeguard against different attacks. A multiple images encryption algorithm using the strands of DNA was written in [19]. The given images were merged into a single image and the single image was resized into a one dimensional array. The confusion process was performed using the indexes of the half array. These indexes were also used in the diffusion process to diffuse the pixels using DNA sequences. The results showed that their scheme was not only secured against different types of attacks but was also fast. The DNA encoding scheme was also used in another image encryption scheme proposed by [20]. In this scheme, the binary search tree (BST) was utilized. Based on the size of the given input image, a candidate BST was determined and was converted into the DNA sequences along with the plain image and XOR operation was performed between them to get the DNA based diffused image. Results showed that the scheme was robust against the different attacks. Besides, an image cipher based on DNA, 3D latin cubes and pixel-level dynamic filtering was given in [21]. The 5D hyperchaotic system was used for the random number sequences. The dynamic filtering was applied over each pixel value to get the diffusion effects over the image. Then the DNA encoding was implemented to further diffuse the image. Obtained image was converted into the 3D DNA level cubes. The latin cuber operation was performed over them and then it was converted back into the 2D cipher image. The results obtained were quite good. A yet another novel approach in image encryption was developed by [22]. The pixel level filtering and DNA level diffusion were used in this scheme using the hyperchaotic map. After the generation of random sequences, the dynamic filtering was applied over the pixels value to alter them. The bit level scrambling was then implemented to perform the permutation process. Next, the DNA level encoding was carried out to perform the diffusion process. The simulation results demonstrated that the scheme has the desirable security effects. Recently, a novel image encryption algorithm has been developed by using the keys derived from plaintext image and DNA [23]. The proposed scheme has employed the idea of chaotic visual selective encryption of the given image data. To guarantee an immunity against the potential threats, the initial values of the chaotic map have been made dependent over the plain image through SHA-512 hash function and the random DNA sequence. The proposed scheme used three 1D chaotic maps. Diffusion operation has been introduced in the plain image through the selection of blocks that have greater correlation. Further, an XOR operation has been carried out with the random matrix. Remaining two chaotic maps cracked the inherent inter-pixel tight correlation through the confusion operation. As the cipher image is sliced into the Least Significant Bit (LSBs) and the Most Significant Bits (MSBs), the host image underwent the lifting wavelet transformation. Simulation and security analysis indicated the robustness and real world application of the image cipher. Moreover, in [24] an image encryption scheme was developed using a novel permutation and DNA operations. For plaintext sensitivity, DNA hamming distances and SHA-256 hash codes have been employed to temper the initial conditions of the 4D chaotic map. A novel scrambling scheme was used through the instrument of improved balanced binary tree. Afterwards, dynamic block coding rules have been embedded in which distinct coding rules were used from the distinct image blocks. In order to realize the diffusion effects, a novel algorithm through inter-block and intra-block has been introduced which carried out the DNA operations on the key matrix and the intermediate encryption result. Experimental results indicated the robustness and immunity of the cipher against the varied attacks from the hackers' community. In a work [25], a new image encryption algorithm has been given through chaos, DNA strands and multi-objective particle swarm optimization (PSO). In order to reach the best cipher image, linear differential descent strategy and chaotic initial particle swarm optimization were employed to avoid in getting stuck the local optima. DNA mask image has been created using the logistic map and DNA encoding. Position of a particle was made to correspond to position of the plain image. Further, the iterative PSO algorithm was applied to the correlation coefficient and the information entropy. Performance analysis indicated that the entropy and correlation coefficient attained very good values. Moreover, the scheme had the ability to defy the typical attacks. Some researchers employed compression as well to their encryption algorithms like [26], [27] to raise the communication speed.

Many works have been cracked due to the different loopholes in the design principles of the image ciphers as literature suggests. Low plaintext sensitivity, for instance, is the cause of cryptanalysis by the known-plaintext and chosen-plaintext attacks upon them. As an example, scheme given in [28] was cracked by the chosen-plaintext attack [29]. Cause of this breakage was that the chaotic vectors produced were independent of the input plain image for encryption. To put this in other words, the scheme given in [28] had not the plaintext sensitivity. Hence, novel image encryption algorithms are needed with suitable plaintext sensitivity in their designs. In [30], some rationality and practicability problems have been figured out about the image cipher reported by the name of 2D logistic-adjusted-sine-map-based image encryption scheme [31]. This scheme was cryptanalyzed by the chosen plaintext attack described in [30]. The experimentations suggested that the plain images could be recovered without having any access to the secret key. Later on, some improvements were suggested to avoid such kinds of attacks in the future. In another cryptanalysis work [32], an image cipher consisting of chaotic system and DNA technology [33] was broken. This scheme was again cracked by the chosen plaintext attack over it. To improve the scheme, some remedial measures were suggested by the cryptanalysts. In a yet another cryptanalysis task [34], a scheme given in [35] has been successfully broken by the chosen plaintext attack. Few tests were conducted through the proposed attack algorithm and the moduli employed in the ICS-IE algorithm [35]. The simulations indicated that the proposed attack algorithm was correct in successfully recovering the plain images without having any know-how about the secret key. Lastly in this work [36], an image cipher based on the DNA level diffusion and the pixel level filtering was analyzed for the possible loopholes in [22]. Through the launch of plaintext attack, the scheme [22] was broken by the [36].

In this research study, after taking inspiration from the above discussion, we undertake to engineer a yet another image encryption algorithm which can be characterized by the following features.

- Castle —a chess piece will be used for the project of scrambling. Previously, chess pieces knight and king have used for scrambling. Although, the given images got scrambled through them but their drawback was that the effects remained localized. Whereas, the scrambling through Castle puts greater scrambling effects and hence render more security. The reason is that Castle can sweep the entire chessboard in one move.
- Four dimensional dynamical chaotic system has been used for the generation of the random data. This dynamic system has rich dynamical structures.
- Fuller potential of DNA encoding has been reaped. The DNA encoding rules have been dynamically applied over the pixels. These rules have been figured out from the pixels data instead of the random vectors given by the chaotic map. This act makes the cipher more defiant from the differential attacks.
- The plaintext sensitivity has been increased by making the starting address of the Castle to initiate its scrambling project from any of the pixel address on the chessboard. This act caused to create a richer plaintext sensitivity. Besides, as a byproduct, key space also got increased.

Rest of the paper has been formatted as follows. Section II discusses the basic theories upon which the study rests, *i.e.*, the theory of chaotic system, DNA computing and chess piece Castle. Section III describes the way, the random data has been spawned, and the algorithm for encryption. Section IV is for the simulation of the encryption scheme. Security analysis has been performed in the Section V. The paper has been wrapped up in the last Section VI along with the necessary remarks of conclusion.



(a) One initial value

(b) Two initial values

FIGURE 1. Attractors of FDCM.

II. BUILDING BLOCKS

This section is a brief discussion about the building blocks which our study has used.

A. FOUR DIMENSIONAL CHAOTIC MAP(FDCM)

The chaotic systems are the popular source for the generation of the random data. Lorenz and Rossler developed their 3D chaotic maps. Afterwards, Yong and Yun-Qing [37] came up with a better 4D chaotic dynamical system which can be described as follows

$$\begin{aligned}
\dot{x_1} &= ax_1 - b_1x_1x_2x_3\\
\dot{x_2} &= bx_2 - b_2x_1x_3x_4\\
\dot{x_3} &= cx_3 - b_3x_1x_2x_4\\
\dot{x_4} &= dx_4 - b_4x_1x_2x_3 \qquad (1)
\end{aligned}$$

This map is dissipative, symmetrical and enjoys the rich dynamical structures. Figure 1 depicts topology and the chaotic behavior of attractors of this chaotic map for $a = -10$, $b = 3$, $c = -1$, $d = -2$, $b_1 = 1$, $b_2 = -1$, $b_3 = 1$ and $b_4 = 1$.

B. DNA COMPUTING

As described earlier, four nucleotides, *i.e.*, A(Adenine), C(Cytosine), G(Guanine) and T(Thymine) complement each other (Figure 2). For instance, if ‘01’ is connected to A then ‘10’ will be connected to T. In the same way, if ‘11’ is connected to C then ‘00’ will be connected to G. There are a total of 4! or 24 kinds of encoding possibilities, in which 8 comply with the complementary rules of Watson-Crick (Table 1). In the realm of DNA technology and in particular DNA computing, research [38] reports some operations over the DNA strands. These operations include XOR, addition and subtraction binary operations. Table 2 shows these operations.

To use A, T, C and G in the proposed encryption algorithm, two conversion functions $DNA - Conv$ and $DEC - Conv$ have been defined which take 8-bit pixel value and convert it into its DNA sequence of length four and vice versa using the (1-8) rules. For example $DNA - Conv(89, 1) = GGCG$, $DNA - Conv(89, 5) = TTAT$ and $DNA - Conv(89, 8) = AATA$. Also, $DEC - Conv(ATCG, 1) = 57$, $DEC - Conv(ATCG, 4) = 198$ and $DEC - Conv(ATCG, 8) = 108$.

FIGURE 2. A DNA.

TABLE 1. Encoding rules for the 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

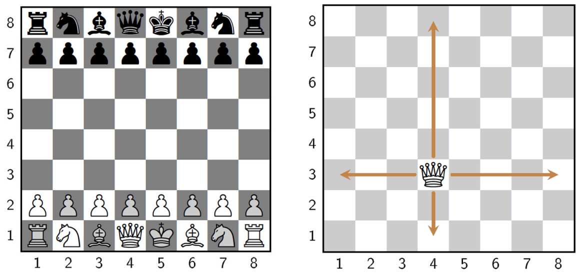
Other operations like addition, subtraction and XOR are also used in the realm of DNA cryptography. These operations are like the popular operations which are carried out in the binary number system. As 8 kinds of encoding rules are there, similarly, there exist 8 kinds of addition, subtraction and XOR operations. All these three operations have been depicted in the Table 2 according to the rule 1. Normally, the symbols $+$, $-$ and $\oplus$ for DNA addition, DNA subtraction and DNA XOR are used. These operations will be carried out between the two DNA sequences of length four based upon the eight rules. For example $+(GTAC, GCCT, 5) = GACC$ and $+(GTAC, GCCT, 1) = AATG$. Also, $-(GTAC, GCCT, 8) = CCTA$ and $-(GTAC, GCCT, 3) = TAGC$. Lastly, $\oplus(GTAC, GCCT, 4) = GCAA$ and $\oplus(GTAC, GCCT, 7) = TGAC$.

TABLE 2. XOR, addition and subtraction operations on the DNA strands/nucleotides.

+	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

C. CASTLE

Castle is a piece of chess —a classical, strategical and a board-based game which is usually played between the two persons [39]. Chessboard is an 8 × 8 grid shown in the Figure 3a. Both the rows and columns are labeled with the integers from 1 to 8. There are 32 pieces in the game in which each player has 16 pieces. The names of the pieces are King, Queen, Bishop, Knight, Castle (aka as Rook) and Pawn. Each piece plays in a distinct way from the most weak (Pawn) to the most powerful (Queen). There are different colored pieces for each player, normally in the black and white colors lying on the first two rows of the chessboard. The particular instance of the Figure 3a shows that the black colored pieces have been placed on the rows 7 and 8, whereas the white colored pieces are lying on the rows 1 and 2. Each square of the chessboard has a unique address. For example, the initial addresses of the Castle with the black colors are (1, 8) and (8, 8). Aim of game is to defeat the opponent. In this research, only Castle has been used for the project of scrambling/confusion of the pixels. Castle moves only horizontally and vertically for an arbitrary steps as depicted in the Figure 3b. To make it a good fit, two deviations have been made from the game's rules: first, the chessboard will have the same size as that of the input image, second, if Castle goes past the edge, it will appear from the opposite edge. Random numbers will govern the Castle for different movements in arbitrary directions.



(a) Chess board with pieces in it (b) Chess board with moves of Castle

FIGURE 3. Game of chess.

III. PROPOSED IMAGE ENCRYPTION ALGORITHM

The proposed algorithm uses chaos theory, DNA computing and random movement of Castle in one setting to encrypt the given input image. It is to be noted that this scheme is for the gray scale image of the arbitrary size $m \times n$. The algorithm is based on some stages (Figure 4). In the first stage, a gray scale image is input. SHA-256 hash codes for this image are generated to realize the effects of plaintext sensitivity. This sensitivity assures to foil the chosen plaintext/ciphertext attacks on the cipher. Each different input image will cause to spawn unique random data since the hash codes generated modify the starting values of the map. Different initial values cause different random numbers. In the second stage, since the chaotic data generated is in the "raw" form, this data is normally customized corresponding to the peculiar logic of the algorithm. Specifically, key streams formed are *direction*, $dist1$, $dist2$, $mask-one$ and $mask-two$. Of course, these five streams of random numbers will serve the different purposes of the cipher. In the third stage, scrambling has been performed. We assert that the main contribution of the current study is the usage of random movement of Castle for the project of image scrambling. Castle moves randomly on the vast chessboard of the size $m \times n$ for the number of pixels times, *i.e.*, mn for given input image. The random variable *direction* decides whether the Castle has to move in the vertical direction or in the horizontal direction. If horizontal direction is decided, the random variable $dist1$ provides the distance with which the Castle has to move. In the same way, if the direction decided is of vertical, the random variable $dist2$ provides the distance with which the Castle has to move. The Castle moves in the range of $(-(m-1), (m-1))$ and $(-(n-1), (n-1))$ for the vertical and horizontal directions respectively. In case, the number given by the $dist1$ or $dist2$ is greater than the distance between the current position of the Castle on the chessboard and the edge, the Castle has been made to move cyclically and to make it appear from the opposite edge of the board. It is to be noted that the random streams $dist1$ or $dist2$ may also give a zero value. In this case, no movement of the Castle would take place.

A scrambled image with the size of given input gray scale image is formed, *i.e.*, $m \times n$. Reshape the input image to the size of $1 \times mn$. We have linked ingeniously the shifting of pixels from the input gray scale image to the scrambled image with the random movement of the Castle.

Here we will describe our procedure. We have three entities at our disposal, *i.e*, pixels of input image, scrambled empty image, random movement of the Castle. Starting from the given particular address at the chessboard, *i.e.*, $(startx, starty)$, the Castle will move to a particular address at the chessboard depending upon the value of variables *direction*, $dist1$ and $dist2$. The first pixel taken from the input image will be shifted to the same address on the scrambled image as the address on which the Castle is currently residing in its first iteration. In the second iteration, the Castle will move again to a particular address on the chessboard, in the same way, the second pixel will be shifted from the input image to the same address on the scrambled image as the address where the Castle has landed on the chessboard. This process will be repeated mn times. Sometimes, Castle lands to an address on the chessboard which has already been occupied by some pixel in the scrambled image, in that case, no shifting would be carried out. In the last, the remaining pixels from the input image would be shifted to the empty addresses of the scrambled image. The scrambled image, key streams $mask - one$ and $mask - two$ are encoded into the DNA strands in the fourth stage. It is to be noted that the rules of conversion have been taken directly from the pixels data instead of the key streams to make the cipher more secure and robust from the differential attacks. In the fifth stage, in order to throw the diffusion effects, DNA Addition and DNA XOR operations have been carried out between the DNA encoded pixel data and DNA encoded $mask - two$ and $mask - one$ respectively. In the sixth and last stage, this DNA encoded strands of the pixel data are translated to their decimal equivalents to obtain the final gray scale cipher image.

A. PROCEDURE FOR KEY STREAM GENERATION

To create the plaintext sensitivity, 256-bit hash code has been employed in the proposed encryption scheme. Generated hash code has been used to update the starting values of the chaotic map. For every plain image, a unique hash code K is generated. Afterwards, it is sliced into 8-bit blocks. In this way, we get 32 hexadecimal digits $K = k_1, k_2, \ldots, k_{32}$. The following steps explain the updates of the values.

Step 1: By the following equations, the variables of the chaotic map/system are being upgraded as (2)–(5), shown at the bottom of the page, where x_0', y_0', z_0', w_0' are the initial values of the chaotic system before the addition of the plaintext sensitivity and x_0, y_0, z_0, w_0 are the starting values of the chaotic map after adding plaintext sensitivity. $\oplus$, $\vee$ and $\wedge$, respectively denote the logical XOR, OR and AND operations in the binary.

Step 2: The looping of the chaotic map (1) $(mn + n_0)$ times rendered the four chaotic sequences $x = \{x_t\}_{t=1}^{mn+n_0}$, $y = \{y_t\}_{t=1}^{mn+n_0}$, $z = \{z_t\}_{t=1}^{mn+n_0}$, $w = \{w_t\}_{t=1}^{mn+n_0}$, the variables m and n correspond to the size of input image. Further, value of n_0 is greater than 500. The first n_0 values of the map would be ignored to avoid the transient effects.

Step 3: Sequences x, y, z and w have been passed through the following equations (6) to obtain the sequences $dist1$, $dist2$, *direction*, $mask - one$ and $mask - two$.

$$\begin{cases} dist1(i) = floor(mod(abs(x(i)) - floor(abs(x(i))) \\ \quad \times 10^{14}, 2n - 1)) - (n - 1), \\ dist2(i) = floor(mod(abs(x(i) \times y(i)) - floor(abs(x(i) \\ \quad \times y(i)) \times 10^{14}, 2m - 1)) - (m - 1), \\ direction(i) = floor(mod(abs(y(i)) - floor(abs(y(i))) \\ \quad \times 10^{14}, 2)), \\ mask - one(i) = floor(mod(abs(z(i)) - floor(abs(z(i))) \\ \quad \times 10^{14}, 256)), \\ mask - two(i) = floor(mod(abs(w(i)) - floor(abs(w(i))) \\ \quad \times 10^{14}, 256)) \end{cases} \tag{6}$$

where x_i, y_i, z_i and w_i are the elements of x, y, z and w respectively. The remainder obtained when s is divided by t has been shown by $mod(s, t)$. $i = 1, 2, \ldots, mn$. Possible values of $dist1$ and $dist2$ are $(-(n - 1), (n - 1))$ inclusive, and $(-(m - 1), (m - 1))$ inclusive in a respective way.

B. ALGORITHM FOR IMAGE ENCRYPTION

Figure 4 shows the encryption algorithm. Steps below explain the procedure.

Step 1: Gray image img is input in the system. Reshape it to the size $1 \times mn$ and call it $img1$. Initialize the starting address

$$x_0 = x_0' + \frac{mod((k_1 \oplus k_2) + (k_3 \oplus k_4) \vee (k_5 \oplus k_6) \wedge (k_7 \oplus k_8), 256)}{256} \tag{2}$$

$$y_0 = y_0' + \frac{mod((k_9 \oplus k_{10}) \vee (k_{11} \oplus k_{12}) \wedge (k_{13} \oplus k_{14}) + (k_{15} \oplus k_{16}), 256)}{256} \tag{3}$$

$$z_0 = z_0' + \frac{mod((k_{17} \oplus k_{18}) + (k_{19} \oplus k_{20}) \vee (k_{21} \oplus k_{22}) + (k_{23} \oplus k_{24}), 256)}{256} \tag{4}$$

$$w_0 = w_0' + \frac{mod((k_{25} \oplus k_{26}) \vee (k_{27} \oplus k_{28}) + (k_{29} \oplus k_{30}) \wedge (k_{31} \oplus k_{32}), 256)}{256} \tag{5}$$

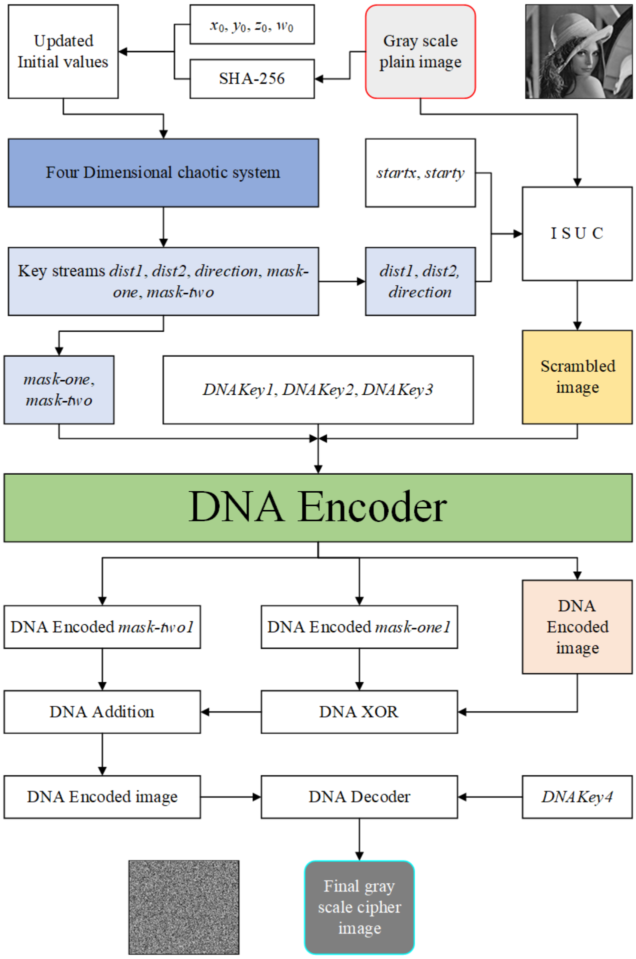


FIGURE 4. ISUC-based encryption scheme.

Algorithm 1 Image Scrambler Using Castle (ISUC)

Input: *img, startx, starty, direction, dist*1*, dist*2*, m, n*

Output: *img*2

```
1:  for $i \leftarrow 1$ to $m$ do
2:      for $j \leftarrow 1$ to $n$ do
            $D(i, j) \leftarrow -1$
3:      end for
4:  end for
5:  $i \leftarrow startx$
6:  $j \leftarrow starty$
7:  for $index \leftarrow 1$ to $mn$ do
8:      switch $(direction(index))$
9:      case 0:
10:         if $dist1(index) > 0$ then
11:             if $i - dist1(index) \geq 1$ then
12:                 $i \leftarrow i - dist1(index)$
13:             else
14:                 $i \leftarrow n - (dist1(index) - i)$
15:             end if
16:         else if $dist1(index) < 0$ then
17:             if $i - dist1(index) \leq n$ then
18:                 $i \leftarrow i - dist1(index)$
19:             else
20:                 $i \leftarrow -dist1(index) - (n - i)$
21:             end if
22:         end if
23:         case 1:
24:         if $dist2(index) > 0$ then
25:             if $i - dist2(index) \geq 1$ then
26:                 $i \leftarrow i - dist2(index)$
27:             else
28:                 $i \leftarrow m - (dist2(index) - i)$
29:             end if
30:         else if $dist2(index) < 0$ then
31:             if $i - dist2(index) \leq m$ then
32:                 $i \leftarrow i - dist2(index)$
33:             else
34:                 $i \leftarrow -dist2(index) - (m - i)$
35:             end if
36:         end if
37:     end switch
38: end for
39: if $D(i, j) = -1$ then
40:     $D(i, j) \leftarrow img(index)$
41:     $img(index) \leftarrow -1$
42: end if
43: $img2 \leftarrow RemainingPixels(img, D, m, n)$
```

of the Castle, *i.e.*, $(startx, starty)$. Invoke the Algorithm 1 ISUC with the parameters $img1$, $startx$, $starty$, $direcction$, $dist1$, $dist2$, m, n to get the scrambled image $img2$. Algorithm 1 further calls Algorithm 2.

In the following, we will explain the Algorithms 1 and 2 turn by turn.

1) Lines (1-4) initialize the scrambled image D with the value -1.
2) Lines (5-6) initialize the starting address (i, j) of the Castle with $(startx, starty)$.
3) Line 7 corresponds to the entire random movement of the Castle for mn times.
4) Line 8 is the *switch* header. Case 0 and Case 1 correspond to the horizontal and vertical movements of the Castle respectively.
5) If the condition of line 10 is true, then the condition of line 11 further checks whether the Castle moves past the edge of the chessboard? If it doesn't, then the value of i is being updated on the line 12. If it does, then in a cyclic fashion, the value of i is being updated on the line 14.
6) If the condition of line 16 is true, then the condition of line 17 further checks whether the Castle moves past the edge of the chessboard?. If it doesn't, then the value of i is being updated on the line 18. If it does, then in a cyclic fashion, the value of i is being updated on the line 20.
7) Case 1 spanning the lines (23-36), for the vertical movement of the Castle, can be understood analogously.

Algorithm 2 RemainingPixels

```
Input: img1, D, m, n
Output: SI
 1: $k \leftarrow 0$
 2: for $i \leftarrow 1$ to $m$ do
 3:     for $j \leftarrow 1$ to $n$ do
 4:         if $D(i,j) = -1$ then
 5:             while $img1(k+1) = -1$ do
 6:                 $k \leftarrow k+1$
 7:             end while
 8:             $D(i,j) \leftarrow img1(k+1)$
 9:             $k \leftarrow k+1$
10:         end if
11:     end for
12: end for
13: $SI \leftarrow D$
```

8) Whatever value of (i,j) is selected, pixel from $img(index)$ gets copied to the scrambled image D at (i,j) (line 40). It is to be noted that the *if* condition on line 39 checks whether the slot is empty. If it is, then the act of copying is carried out, otherwise, copying has not been carried out. It is to be further noted that the line at 41 assigns -1 to the $img(index)$ so that a track can be kept for the pixels which have been copied.
9) Line 43 calls the Algorithm 2 to put the remaining pixels in the scrambled image D.
10) The nested loops of lines (2-3) provide all the addresses $(i = 1 : m, j = 1 : n)$ for scrambled image D. If the condition of line 4 is true, then the *while* loop of lines (5-6) check for a non-empty slot. Upon finding the one, the pixel residing on that non-empty slot gets copied to the scrambled image D at the address (i,j). The index k is updated on the line 9 for the next pixel.

Step 2: DNA-encode the pixels data $img2$ and the mask images $mask-one$ and $mask-two$ as follows.

$$\begin{cases} img3(i) = DNA - Conv(img2(i), DNAkey1), \quad if\ i = 1 \\ img3(i) = DNA - Conv(img2(i), \\ mod(img2(i-1), 8) + 1), \quad if\ i > 1 \\ mask - one1(i) = DNA - Conv(mask - one(i), \\ \qquad DNAkey2), \quad if\ i = 1 \\ mask - one1(i) = DNA - Conv(mask - one(i), \\ \qquad mod(mask - one(i-1), 8) + 1), \quad if\ i > 1 \\ mask - two1(i) = DNA - Conv(mask - two(i), \\ \qquad DNAkey3), \quad if\ i = 1 \\ mask - two1(i) = DNA - Conv(mask - two(i), \\ \qquad mod(mask - two(i-1), 8) + 1), \quad if\ i > 1 \end{cases} \tag{7}$$

where $i = 1, 2, \ldots, mn$. Each array $img3$, $mask-one1$ and $mask-two1$ comprises of mn DNA sequences of length 4.

Step 3: Do the DNA XOR and DNA Addition operations (Table 2) between DNA image $img3$ and DNA key images $mask-one1$ and $mask-two1$ for the creation of effects of diffusion as follows:

$$\begin{cases} img4(i) = \oplus(img3(i), mask - one1(i)), \\ img5(i) = +(img4(i), mask - two1(i)) \end{cases} \tag{8}$$

where $i = 1, 2, 3, \ldots, mn$. $img5$ is the resultant image after creating the diffusion effects at DNA level.

Step 4: Lastly convert the DNA strands back into their decimal equivalents as (9), shown at the bottom of the page, where $i = 1, 2, \ldots, mn$. $img6$ is the resultant cipher image.

Since the proposed image cipher has been fashioned along the lines of private key (symmetric key), therefore, its decryption version will be trivial. This version can be obtained by inversing the steps of the encryption procedure.

IV. SIMULATION RESULTS

To show the practical effectiveness and do-ability of the proposed image encryption, we will simulate it in this section. For this purpose, eight gray scale test images Lena, Baboon, Moon, Clock, Camera man, Airplane, Aerial, Chemical plant have been selected from the USC-SIPI Image Database. Each of these images have the size of 256 × 256. MATLAB 2016 version with 64-bit double-precision and standard of IEEE 754 [40] has been used to carry out the computer experiments of the proposed encryption/decryption algorithms.

To ignite the chaotic system for the generation of the streams of random numbers, these system parameters and initial values have been given to it: $x_0 = 17$, $y_0 = 25$, $z_0 = 118$, $w_0 = 5$, $a = -10$, $b = 3$, $c = -1$, $d = -2$, $b_1 = 1$, $b_2 = -1$, $b_3 = 1$ and $b_4 = 1$. Apart from that, values given to the initial address of the Castle to initiate its trajectory of scrambling are $(startx, starty) = (50, 50)$. Moreover, $DNAKey1 = 100$, $DNAKey2 = 200$, $DNAKey3 = 50$ and $DNAKey4 = 60$ are the values of the DNA keys. Figures 5, 6 and 7 draw original plain images, encrypted images obtained after application of proposed cipher upon them and the decrypted images after applying the proposed decryption algorithm on the encrypted images. One can see that the proposed cipher has radically turned the test plain images into very unrecognizable format. Further, the plain images have been successfully retrieved after applying the decryption algorithm over the encrypted images. This vividly demonstrates the workability of the proposed algorithm.

V. PERFORMANCE AND SECURITY ANALYSES

Just visual inspection doesn't suffice the security requirements of any cipher as was done in the previous section.

$$\begin{cases} img6(i) = DEC - Conv(img5(i), DNAkey4), & if\ i = 1 \\ img6(i) = DNA - Conv(img5(i), mod(img6(i-1), 8) + 1), & if\ i > 1 \end{cases} \tag{9}$$

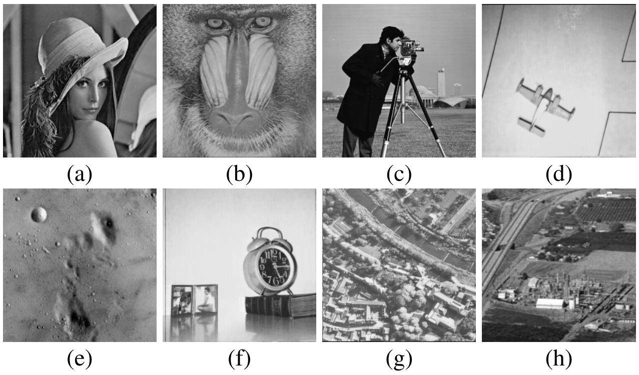


FIGURE 5. Eight test plain images: (a) Lena (a) Baboon (b) Camera man (c) Airplane (b) Moon (d) Clock (e) Aerial (f) Chemical plant.

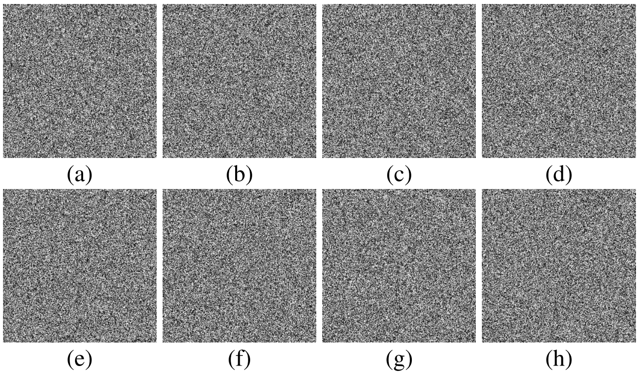


FIGURE 6. Cipher images: (a) Lena (b) Baboon (c) Camera man (d) Airplane (e) Moon (f) Clock (g) Aerial (h) Chemical plant.

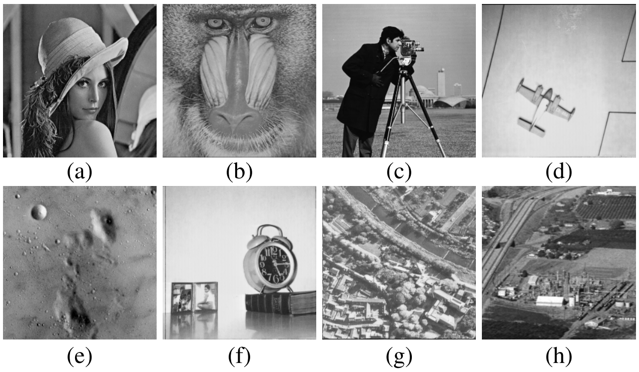


FIGURE 7. Decrypted images: (a) Lena (b) Baboon (c) Camera man (d) Airplane (e) Moon (f) Clock (g) Aerial (h) Chemical plant.

Researchers have developed a plethora of objective yardsticks to analyze the different security facets of any cipher. This section is for carrying out the performance and security analyses of the suggested image encryption using those yardsticks often called the validation metrics. Different metrics analyze the different aspects of the security of any cipher. These metrics normally include key space, key sensitivity, histogram and its variance, correlation coefficient, information entropy, differential attack, mean absolute error, peak signal-to-noise ratio, floating frequency, noise & data loss attacks, computational complexity and encryption throughput *etc*. Further, we have taken these studies [41]–[44] from the published works for the sake of comparison based on the different validation metrics.

A. KEY SPACE

Set of all possible secret keys of a cipher constitutes its key space. Sufficiently large key space serves as a great immunity to the potential brute-force attacks. Hackers try all the possible values of the secret key over the cipher until the concerned key is not found in this attack. To foil this attack, key space should be so much large that trying all the possible secret keys upon the cipher should not be feasible in the practical time. The researchers have evolved a consensus that 2^{100} [44] is the minimum key space to counter the brute-force attack. So, to ensure security from this attack, key space of algorithms should be minimum of this value 2^{100}. The secret key of the proposed cipher consists of the initial values x_0, y_0, z_0, w_0 and the system parameters a, b, c, d, b_1, b_2, b_3, b_4 of the chaotic map being used. If the computer precision of 10^{-14} is assumed then key space comes out to be $10^{14\times 12} = 10^{168}$. Apart from that, upon including the values of initial address of the Castle to start its journey on the chessboard *i.e.*, $(startx, starty)$ and DNA keys, *i.e.*, $DNAKey1$, $DNAKey2$, $DNAKey3$ and $DNAKey4$ in the key space, it becomes $10^{168} \times 2^{48} \approx 2.81 \times 10^{182}$, we are assuming here a 256 × 256 chessboard. Key space of 2.81×10^{182} is far more greater than the minimum threshold of 2^{100}. So the proposed cipher is immune from the threat of brute-force attack. Moreover, the Table 3 draws a comparison of the proposed cipher with some other published works. One can see that the proposed scheme beats these schemes [41], [44] regarding the key space.

TABLE 3. Key space comparison.

Algorithm	Key space
Proposed	2.81×10^{182}
Ref. [41]	10^{88}
Ref. [42]	3.9402×10^{185}
Ref. [43]	10^{195}
Ref. [44]	2.9645×10^{149}

B. FLOATING FREQUENCY

As the images are confused and diffused, the good ciphers are expected to render the encrypted images in such a way that the chaotic data has been distributed uniformly in all its sections (rows and columns). This is what the floating frequency instrument does [45]. Contrarily, if both rich and poor sections of encryption exist in the cipher images, they will be caught through this powerful security parameter. As far as the procedure of this parameter is concerned, windows of 256 elements (based on row and column) taken from the plain and cipher images are examined. During this examination, it is appraised that how many intensity values of the pixels are distinct with each other. Normally, two parameters

(*RFF*) and (*CFF*) are employed. They stand for row floating frequency and the column floating frequency respectively. Following is the three step mechanism for the evaluation of this parameter.

1) Proceeding in a systematic fashion, windows of 256 pixels are taken from rows and columns of the given image.
2) Against each selected window, the *RFF* and *CFF* for the pixel intensities are determined.
3) Draw both *RFF* and *CFF* after finding their mean values.

The *CFF* and *RFF* can be seen in the Figure 8 for the Lena's plain and cipher images. Specifically, in the Figure 8a, for the columns ranging from 1 to 256 of the plain image of Lena, the graph of floating frequency has been drawn. With the same pixels' intensity values, there are relatively more pixels. Hence, the values of *CFF* are low against all the selected windows. In contrast to that, relatively less pixels with the same intensity values exist in the Figure 8b. To put this in other words, there are more pixels with the different intensity values in each column. Such kind of phenomenon corresponds to the better ciphers. It further explains the high values of *CFF*. It is to be further noted that 128 and 162 respectively are the values of the means for the Lena's plain and encrypted images. In the percent form, this corresponds to $\frac{128}{256} \times 100\% = 50\%$ and 63% respectively. A greater percentage of *CFF* in the cipher image is indicative to the better security effects. In the same way, Figures 8c and 8d show the *RFF* for the Lena plain image and its encrypted version with the mean values of 107 and 161 respectively. 42% and 63% come out to be the their values in the percent form. Undoubtedly, more percentage of *RFF* in the encrypted versions of the plain images is better for the image ciphers.

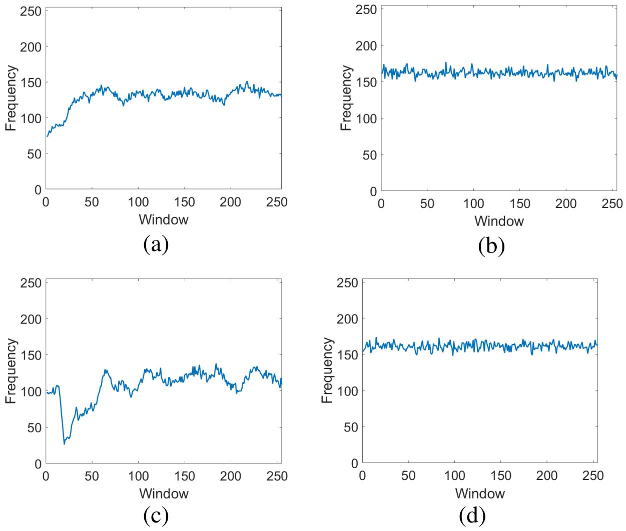


FIGURE 8. Column and row floating frequency for the plain and cryptic Lena images and their mean values: (a) Column floating frequency of plain image, 128; (b) Column floating frequency of encrypted image, 162; (c) Row floating frequency of plain image, 107; (d) Row floating frequency of encrypted image, 161.

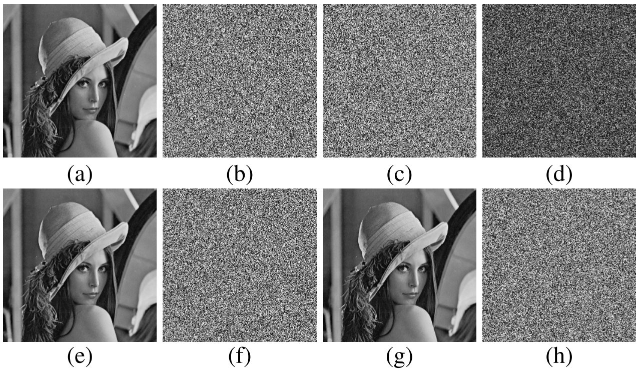


FIGURE 9. A demo of key sensitivity test on image of Lena:a) Original image; (b) Encrypted image using K_0; (c) Encrypted image using K_1; (d) Differential image between (e) and (f); (g) Decrypted image from (h) using correct key K_0; (i) Decrypted image from (j) using wrong key K_1; (k) Decrypted image from (l) using correct key K_1; (m) Decrypted image from (n) using wrong key K_0.

C. KEY SENSITIVITY

The image ciphers furnished with the extreme key sensitivity are more secured. So this characteristic should not be ignored at all, while engineering any cryptographic product. This sensitivity is demonstrated using both the encryption and decryption algorithms of the cipher. In the encryption algorithm, a faint modification is introduced in the secret key. Encrypted image obtained ought to be totally distinct from the one which was obtained without introducing any change in the key. Same should hold true in the decryption algorithm, *i.e.*, a very little change is made in the key and decryption algorithm is invoked. The original plain image should not be retrieved unless the 'verbatim' key is not employed.

The key sensitivity test for the proposed encryption algorithm has been performed by using the two slightly different keys for encrypting the single plain image. Suppose the initial key set is $x_0, y_0, z_0, w_0, a, b, c, d, b_1, b_2, b_3, b_4$ and say this as K_0. By using this key, the encryption algorithm has been applied on the plain image of Lena (Figure 9a) and cipher image of Lena has been obtained and drawn in the Figure 9b. Now, a very minute tempering of 10^{-14} is introduced in the variable x_0 of the secret key, *i.e.*, $x_0{}' = x_0 + 10^{-14}$. Other keys have been remained unchanged. In this way, a second set of keys has been obtained, *i.e.*, $x_0{}', y_0, z_0, w_0, a, b, c, d, b_1, b_2, b_3$ and b_4 and call it as K_1. Next, by using K_1, the same Lena image of Figure 9a is encrypted and cipher image obtained is drawn in the Figure 9c. Besides, Figure 9d shows the differential image (obtained by pixel-to-pixel difference) between these two images. Apart from that, even tiny change of (10^{-14}) exists in keys, encrypted image in Figure 9b has 99.6384% differences from the one in Figure 9c regarding the pixel values.

For checking this sensitivity in a more sophisticated manner, rates of difference have been found between two cipher-images rendered by K_0 and $K_t (t = 1, 2, \ldots 24)$. Keys K_0 and $K_t (t = 1, 2, \ldots 24)$ are slightly different. Table 4 lists the results obtained. The Table 4 shows that the least difference rate between any two encrypted images is 99.5809% which is slightly greater than [46]. Average value of this metric calculates to be 99.62% which is better than [47], [48]. These metrics depict that the proposed image encryption is better than the others.

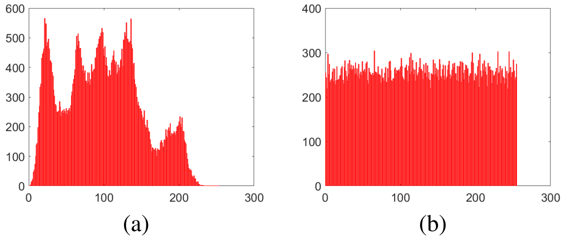
To check this sensitivity regarding second case, K_0 and K_1 are used for decrypting encrypted images of the Figures 9b and 9c respectively. Resultant decrypted images are drawn in Figures 9e-9h. These figures vividly demonstrate the feature of the proposed image cipher that encrypted images can be recovered only by employing the correct keys. Even a very faint change in any variable of the key set causes to end with the entirely different image. Hence, we can assert that the important feature of key sensitivity runs through the very fabric of both the encryption and decryption machineries of the algorithm.

D. STATISTICAL ANALYSIS

In statistical analysis, normally, histogram and correlation analyses are covered.

1) ANALYSIS OF HISTOGRAM

An image is basically a peculiar combination of pixels with distinct intensity values. Histogram shows pixel values' distribution in a pictorial form. Histograms of normal images have a very curved and slanting bar which is replete with a lot of information about the image. The way, its bar goes up and down, is very attractive and fascinating to the cryptanalysis savvy. So the principal job of any image cipher is to restructure pixels' intensity values in a way that resultant histogram made through it contains a very uniform bar. This uniformity is a great immunity for the potential histogram attack. Histograms of Lena's plain and encrypted images have been drawn in the Figures 10a and 10b respectively. We can see that the histogram bar for the plain input image is going up and down, whereas the bar of cipher image is uniform. Uniform bars of the histogram act as a great barrier to the histogram attack over the ciphers.



(a) (b)

FIGURE 10. Histogram analysis of Lena images (a) Plain image, (b) Encrypted image.

Set aside the visual results of the histograms regarding their bars, there exists an objective criterion for the quantification of the uniformity of a histogram, called variance. The relatively lesser values of this metric correspond to the higher uniformity of the histograms and vice versa [49], [50]. The mathematical formula for the calculation of variance is [51]

$$var(P) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(p_i - pj)^2 \tag{10}$$

where P is the 1D array of the values of histogram and $P = \{p_1, p_2, \ldots, p_{256}\}$. Here p_i and p_j refer to the frequency of pixels whose gray values are equal to i and j respectively. Variance values for the histograms of the cipher images of Lena, Baboon, Camera man, Airplane, Moon, Clock, Aerial, Chemical plant images can be seen in the Table 5. Variances in the first row of the table have been computed by initial key set K_0, whereas, the ones in the remaining rows are obtained by modifying key which is K_t $(t = 1, 2, \ldots 24)$ defined in the Section V-C. Table 5 shows that the average variance value for the selected images is 265.3581. Further, the variance for the Lena cipher image is 232.5076 which is better than [42]. Hence the proposed cipher is better.

2) CORRELATION ANALYSIS

Pixels of normal images are fashioned in such a way that a strong correlation exists between the neighboring pixels. Any two neighboring pixels residing in the horizontal, vertical or diagonal way are termed as adjacent/consecutive pixels. The prime focus of any image encryption algorithm is to dismantle this strong nexus between the consecutive pixels. Once these adjacent pixels are dismantled, the correlation among the adjacent pixels drops down steeply and the normal image becomes a noise-like image which can not be recognized. No correlation exists for an ideally randomized image. To measure this metric, we have chosen arbitrarily 3,000 pairs of adjacent pixels. These pixels have been chosen for the plain and cipher images of Lena. Further, these pairs have been taken for the vertical, horizontal and diagonal directions. Mathematical formulation (correlation coefficient CC) for the computation of this metric is [49]:

$$CC = \frac{N\sum_{j=1}^{N}(x_j \times y_j) - \sum_{j=1}^{N}x_j \times \sum_{j=1}^{N}y_j}{\sqrt{\left(N\sum_{j=1}^{N}x_j^2 - \left(\sum_{j=1}^{N}x_j\right)^2\right)\left(N\sum_{j=1}^{N}y_j^2 - \left(\sum_{j=1}^{N}y_j\right)^2\right)}} \tag{11}$$

In the above equation, N is the total number of pixels whereas, x and y are the pixel intensity values of two adjacent pixels. Correlation distributions drawn in the horizontal, vertical and diagonal orientations have been depicted in the Figure 11. Further, this figure covers the plain and encrypted images of Lena.

Moreover, the values of correlation coefficients have been written in the Table 6. The values of this metric are close to 1 for the plain image whereas, they are close to 0 if the image is cipher. The Table 6 and Figure 11 jointly demonstrate that the relation between the plain image and cipher image has been reduced dramatically. To put this phenomenon in

TABLE 4. Difference rates between two images encrypted by slightly different keys.

Secret security keys	Difference rates(%)							
	Lena	Baboon	Camera man	Airplane	Moon	Clock	Aerial	Chemical plant
$Key_1(x_0{}' = x_0 + 10^{-14})$	99.6490	99.6353	99.5987	99.6140	99.6506	99.5834	99.6217	99.6128
$Key_2(y_0{}' = y_0 + 10^{-14})$	99.5911	99.5972	99.5941	99.6109	99.5850	99.6246	99.6009	99.6632
$Key_3(z_0{}' = z_0 + 10^{-14})$	99.6384	99.6185	99.6429	99.6140	99.6445	99.5880	99.6295	99.6192
$Key_4(w_0{}' = w_0 + 10^{-14})$	99.6155	99.6368	99.6429	99.5819	99.5758	99.6368	99.6451	99.6965
$Key_5(a' = a + 10^{-14})$	99.6257	99.6269	99.6521	99.5911	99.5959	99.6269	99.6300	99.6273
$Key_6(b' = b + 10^{-14})$	99.6254	99.6277	99.6176	99.5909	99.5951	99.6243	99.6376	99.6209
$Key_7(c' = c + 10^{-14})$	99.6350	99.6265	99.6054	99.5912	99.5898	99.6323	99.6358	99.6366
$Key_8(d' = d + 10^{-14})$	99.6287	99.6312	99.6298	99.5932	99.5812	99.6368	99.6511	99.6052
$Key_9(b_1{}' = b_1 + 10^{-14})$	99.6525	99.6531	99.6415	99.5932	99.5958	99.6300	99.6408	99.6215
$Key_{10}(b_2{}' = b_2 + 10^{-14})$	99.6525	99.6321	99.6219	99.5899	99.5812	99.6368	99.6511	99.6512
$Key_{11}(b_3{}' = b_3 + 10^{-14})$	99.6545	99.6618	99.6229	99.5919	99.5921	99.6334	99.6259	99.6234
$Key_{12}(b_4{}' = b_4 + 10^{-14})$	99.6243	99.6312	99.6497	99.5923	99.5812	99.6618	99.6541	99.6615
$Key_{13}(x_0{}' = x_0 - 10^{-14})$	99.6384	99.5911	99.5926	99.5819	99.6368	99.5987	99.6113	99.6488
$Key_{14}(y_0{}' = y_0 - 10^{-14})$	99.5850	99.5819	99.5926	99.6155	99.5834	99.6002	99.6309	99.6507
$Key_{15}(z_0{}' = z_0 - 10^{-14})$	99.6445	99.6338	99.6185	99.6078	99.6460	99.6399	99.6297	99.6153
$Key_{16}(w_0{}' = w_0 - 10^{-14})$	99.6338	99.6109	**99.5809**	99.6155	99.6277	99.5956	99.6341	99.6663
$Key_{17}(a' = a - 10^{-14})$	99.6155	99.6368	99.6429	99.5819	99.5958	99.6368	99.6451	99.6965
$Key_{18}(b' = b - 10^{-14})$	99.6051	99.6269	99.6120	99.5810	99.6052	99.6258	99.6209	99.6162
$Key_{19}(c' = c - 10^{-14})$	99.6054	99.6261	99.6319	99.5899	99.5881	99.6208	99.6257	99.6061
$Key_{20}(d' = d - 10^{-14})$	99.6121	99.6260	99.6216	99.5919	99.5957	99.6261	99.6350	99.6165
$Key_{21}(b_1{}' = b_1 - 10^{-14})$	99.6111	99.6266	99.6299	99.5899	99.5987	99.6268	99.6211	99.6244
$Key_{22}(b_2{}' = b_2 - 10^{-14})$	99.6253	99.6265	99.6287	99.5898	99.5958	99.6368	99.6357	99.6265
$Key_{23}(b_3{}' = b_3 - 10^{-14})$	99.6158	99.6269	99.6299	99.5910	99.6258	99.6368	99.6257	99.5961
$Key_{24}(b_4{}' = b_4 - 10^{-14})$	99.6257	99.6269	99.6277	99.5919	99.5959	99.6277	99.6455	99.6165
Average	**99.63**	**99.63**	**99.62**	**99.60**	**99.60**	**99.62**	**99.63**	**99.63**
Average of all	**99.62**	-	-	-	-	-	-	-

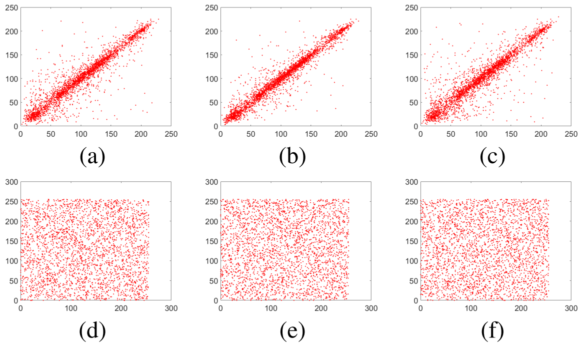
TABLE 5. The variance values of histograms of the cipher-images using different keys.

Secret security keys	Difference rates(%)									
	Lena	Baboon	Camera man	Airplane	Moon	Clock	Aerial	Chemical plant	**Average**	**Average of all**
Key_0	232.5076	269.2113	235.0123	236.5821	264.3124	251.6128	236.1412	281.8212	**250.9001**	**265.3581**
Key_1	253.3312	281.2767	267.9121	279.5123	252.3122	257.1541	270.1512	271.8612	**266.6889**	
Key_2	256.5111	237.2312	261.5112	256.0612	259.1212	271.6112	286.0912	261.6276	**261.2207**	
Key_3	271.6112	272.6121	270.6229	271.6112	240.6112	240.8212	255.9812	264.9212	**261.0990**	
Key_4	273.5812	255.6312	261.3329	274.8912	255.0212	263.6712	281.6922	273.6521	**267.4342**	
Key_5	254.6112	256.1541	256.2632	271.5312	265.0812	277.6112	261.6012	276.0039	**264.8572**	
Key_6	252.6003	263.1137	273.6445	255.2912	271.5433	266.4712	263.4300	261.1222	**263.4021**	
Key_7	264.4324	253.8812	245.3437	274.3320	255.6912	244.2912	274.3128	254.4912	**258.3470**	
Key_8	248.8811	279.6911	288.4811	281.9117	281.8800	269.7117	271.2423	271.1243	**274.1154**	
Key_9	277.8311	274.2312	261.8821	267.5112	281.4543	289.6216	271.4309	283.4233	**275.9232**	
Key_{10}	261.3328	264.8712	271.6198	266.3311	261.3454	273.6311	287.9210	254.3434	**267.6745**	
Key_{11}	284.6436	265.2711	264.9012	271.2243	286.4521	245.9330	281.1543	271.2232	**271.3504**	
Key_{12}	251.6322	269.6112	283.6843	254.8421	271.5400	261.6243	271.3232	261.0043	**265.6577**	
Key_{13}	271.2232	281.6812	271.2005	261.1432	271.3431	281.9212	271.0321	231.7632	**267.6635**	
Key_{14}	256.9512	281.6623	271.5312	276.6823	251.0436	271.5923	281.6347	261.6332	**269.0914**	
Key_{15}	281.1512	261.3354	271.9123	231.8923	271.8743	263.6435	245.3954	233.9329	**257.6422**	
Key_{16}	276.6098	281.8323	261.4320	281.8821	264.6832	281.2443	271.8922	257.2923	**272.1085**	
Key_{17}	261.8332	256.8337	281.6999	271.0039	261.3332	239.9323	271.6763	261.0432	**263.1695**	
Key_{18}	281.3322	261.0126	271.3987	261.9438	271.6523	281.6236	282.1612	254.2212	**270.6682**	
Key_{19}	281.2254	271.8712	271.6723	259.5121	257.0043	281.6512	271.3223	271.6923	**270.7439**	
Key_{20}	271.8223	271.3723	271.9543	271.5222	252.4762	265.6799	274.2431	281.5434	**270.0767**	
Key_{21}	271.6212	281.0123	249.5426	281.1519	241.1823	251.6312	281.5235	271.0523	**266.0897**	
Key_{22}	251.2323	261.7435	231.1122	264.4340	281.6435	251.6412	261.9778	244.3342	**256.0148**	
Key_{23}	281.5532	271.6823	251.8223	251.1523	231.3334	271.2256	251.9623	242.1243	**256.6070**	
Key_{24}	251.2223	239.5238	279.2123	279.5812	269.8723	281.1223	271.1023	251.6211	**265.4072**	

the other words, the pixels of the plain image have undergone an across the board breakage. Table 7 has further drawn a comparison of this metric between the proposed algorithm and the ones published in the literature [41]–[44]. The majority of the results of the proposed algorithm are competitive.

E. INFORMATION ENTROPY ANALYSIS

Entropy connotes disorder, unpredictability, randomness and other words of such character. After the application of an image cipher over the plain image, its pixels are disarrayed significantly. To measure this disorder in the pixels of the image, naturally, we require such a formula which may guage



(a) (b) (c)

(d) (e) (f)

FIGURE 11. Correlation distribution of adjacent pixels of Lena plain and cipher images (direction, type of image): (a) Horizontal, plain; (b) Vertical, plain image; (c) Diagonal, plain; (d) Horizontal, cipher;(e) Vertical, ciphere; (f) Diagonal, cipher.

TABLE 6. Correlation coefficient security parameter values between the plain and encrypted images of Lena.

Type of image	Correlation direction		
	Horizontal	Vertical	Diagonal
Plain image	0.9172	0.9516	0.8941
Cipher image	-0.0061	0.0067	-0.0018

TABLE 7. Correlation coefficients' comparison of various encryption methods.

Type of image	Algorithm	Correlation direction		
		Horizontal	Vertical	Diagonal
Plain Lena image		0.9172	0.9516	0.8941
Encrypted Lena image	Proposed	-0.0061	0.0067	-0.0018
	Ref. [41]	-0.0082	-0.0128	-0.0012
	Ref. [42]	-0.0029	0.0013	-0.0026
	Ref. [43]	-0.0063	0.0065	-0.0016
	Ref. [44]	-0.0021	0.0009	0.0003

TABLE 8. The information entropy results.

Schemes	Names of images	Plain	Cipher
Proposed	Lena	7.5954	7.9974
	Baboon	6.9730	7.9969
	Camera man	7.0097	7.9972
	Airplane	6.4523	7.9972
	Moon	6.7093	7.9974
	Clock	6.7057	7.9970
	Aerial	7.3118	7.9972
	Chemical plant	7.3424	7.9971
	Average	**7.0125**	**7.9972**
Ref. [41]	Lena	7.3200	7.9896
Ref. [42]	Lena	7.3003	7.9971
Ref. [43]	Lena	7.5954	7.9978
Ref. [44]	Lena	7.5788	7.9972

this disorder. Luckily, the formula given by Shannon [52], in 1949, does this job:

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) log \frac{1}{p(m_i)} \tag{12}$$

The entropy value for the information source m has been denoted by $H(m)$ in the above equation. Further, the probability of the symbol m_i has been referred to by $p(m_i)$. If some image with 256 gray values has been randomized to the idealistic proportions then the value of this metric calculates to be 8. The ciphers having value nearer to 8 bear greater security effects. Table 8 gives the entropy values for the various images. According to this table, the average measure of this metric for the images is very close to 8, the ideal value. Hence, the proposed algorithm is defiant to the entropy attack. Besides, this table also draws a comparison of this metric with some other algorithms in the published works. Proposed algorithm is superior than those in [41], [42], [44] for the Lena image regarding the validation metric of information entropy.

F. LOCAL SHANNON ENTROPY

A good image cipher is expected to distribute the pixels of the given image randomly to defy the varied attacks. The concept of local Shannon entropy (LSE) represents a relatively stricter expression of the randomness of the image pixels [53]. For the given image img, if k non-overlapping blocks S_1, S_2,..., S_k with T_B pixels are selected randomly, then the LSE is defined mathematically as

$$\overline{H_{k,T_B}(img)} = \sum_{i=1}^{k} \frac{H(S_i)}{k} \tag{13}$$

where $H(S_i)$ is the Shannon entropy of image block S_i and can be defined as

$$H(S_i) = \sum_{l=1}^{L} p(l) log \frac{1}{p(l)} \tag{14}$$

where L is the total number of pixel values and $p(l)$ is the probability of l^{th} value.

TABLE 9. Local Shannon entropy values for the given images.

Names of images	Local Shannon entropy	Result
Lena	7.902304	Passed
Baboon	7.902256	Passed
Camera man	7.903104	Failed
Airplane	7.902677	Passed
Moon	7.902641	Passed
Clock	7.902292	Passed
Aerial	7.902801	Passed
Chemical plant	7.902744	Passed

According to the set of recommendations given in [53], the parameters (k, T_B) are set as (30, 1936). Moreover, for the significance $\alpha = 0.05$, the ideal value of LSE comes out to be 7.902469317 and an image is considered to pass the test if $7.901901305 \leq LSE \leq 7.903037329$. Table 9 shows LSE values for the given encrypted images. Except one image, all the other images have passed the test of LSE. We can infer that the proposed image cipher has the capability to encrypt the plain images into their encrypted versions with requisite randomness for majority of the images.

G. PLAINTEXT SENSITIVITY ANALYSIS (DIFFERENTIAL ATTACK)

Hackers exploit every opportunity to have an access over the original plain images. One opportunity is to make a very minor corruption in the input image; encrypt the two images, *i.e.*, one without the minor change and the other with a minor change, and find some meaningful relationing between the cipher and plain images. This explains the term "Differential attack." To cope with this scenario, two security parameters, *i.e.*, *NPCR*: Number of Pixels Change Rate and *UACI*: Unified Average Changing Intensity have been put forward by the researchers to observe the implications of modifying the intensity value of a single pixel in the plain image on the resultant cipher image. Their mathematical formulations are

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \tag{15}$$

where M and N represent the width and height of the image respectively. $D(i,j)$ can be defined by:

$$D(i,j) = \begin{cases} 1, & \text{if} \quad C(i,j) \neq C'(i,j); \\ 0, & \text{if} \quad C(i,j) = C'(i,j). \end{cases} \tag{16}$$

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

where C: cipher image with no change in the intensity value of the pixels and C': cipher image obtained after the modification of the single pixel value.

TABLE 10. Average values for the metrics NPCR and UACI using the chosen images in percentage.

Names of images	NPCR	UACI
Lena	99.6140	33.4001
Baboon	99.6353	33.5160
Camera man	99.6048	33.6377
Airplane	99.6277	33.5800
Moon	99.5880	33.5259
Clock	99.6063	33.3111
Aerial	99.5850	33.4253
Chemical plant	99.5880	33.6194
Average	**99.6061**	**33.5019**

TABLE 11. Comparison for the values of NPCR and UACI based on encryption schemes.

Scheme	Image	NPCR(%)	UACI(%)
Proposed	Lena	99.6140	33.4001
Ref. [41]	Lena	99.6090	33.4727
Ref. [42]	Lena	99.6067	33.5000
Ref. [43]	Lena	99.5804	33.4533
Ref. [44]	Lena	99.5956	33.4588

These values of the selected images have been written in Table 10. The average values of *NPCR* and *UACI* are 99.6061% and 33.5019% respectively which vividly prove that the suggested encryption scheme is sufficiently potent to withstand the differential attacks of *NPCR* and *UACI*. Besides, Table 11 compares our values of *NPCR* and *UACI* for the Lena image with some published works [41]–[44]. The proposed method has the better values of *NPCR* as compared to the ones given in [41]–[44]. Unfortunately, *UACI* values of the proposed scheme are not competitive.

H. PSNR ANALYSIS

PSNR stands for Peak signal-to-noise ratio. Notion behind the paradigm of image encryption is to cause an optimal difference between the two images. These images are input plain image and the output encrypted image. A validation metric called PSNR is normally employed to guage this difference. The mathematics corresponding to this concept is [5]

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

Variables being used in the above equation are described below:

(M, N): dimension of the image, $P_0(i,j)$: pixel value of the original image at address(i,j), $P_1(i,j)$: pixel value of the encrypted image at address (i,j). Further, *MSE* corresponds to the mean squared error value. Relatively bigger values of *MSE* and smaller values of *PSNR* imply the better encryption and security effects.

PSNR values by various algorithms have been written in the Table 12. 'O-C' refers to the *PSNR* value between the original and cipher images, and 'O-D' to the original and decrypted images. Table depicts that the *PSNR* values are always infinite (∞) for the original and decrypted images. A fact that can be readily inferred from this phenomenon is $MSE = 0$. We will further infer that the proposed encryption scheme is lossless. Moreover, the proposed algorithm renders the better results of *PSNR* metric for the Lena image when compared with [54], [55].

I. MEAN ABSOLUTE ERROR (MAE)

This metric finds the difference/discrepancy in the plain input image and the encrypted output image. For the better security effects, its value should be large. Mathematically, this can be written as:

$$MAE = \frac{1}{c \times d}\sum_{r=1}^{c}\sum_{s=1}^{d}|C(r,s) - P(r,s)| \tag{19}$$

In the equation for *MAE*, *P* corresponds to the plain image and *C* to the cipher image. *c* and *d* are the width and height of the image. Table 13 gives the results of this important metric. Besides, this table also compares our results with those of some other schemes [41], [42], [54], [56]. Our scheme performs better than the ones given in [41], [42] for the Lena image.

J. NOISE AND DATA CROP ATTACKS

Real world is very precarious and fuzzy. Things do not always proceed as they are expected. Sometimes, the encrypted images get contaminated due to some noise either during the

TABLE 12. The PSNR results.

		Lena	Baboon	Camera man	Airplane	Moon	Clock	Aerial	Chemical plant
Proposed	(O-D)	∞	∞	∞	∞	∞	∞	∞	∞
	(O-C)	8.5674	10.0322	8.3444	7.7728	10.2202	7.2448	9.2693	9.2036
Ref. [54]	(O-C)	8.6878							
Ref. [55]	(O-C)	9.0486							
Ref. [41]	(O-C)	8.1300							
Ref. [43]	(O-C)	8.5581							

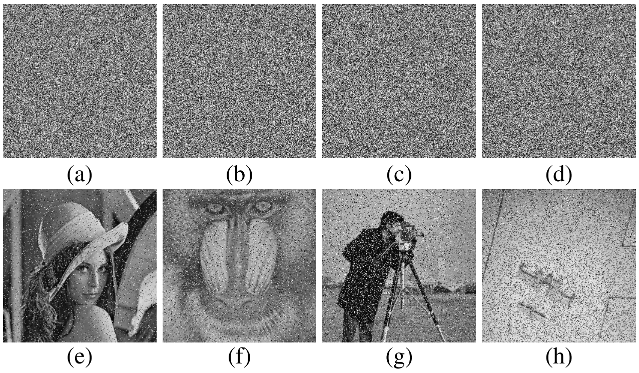


FIGURE 12. Pepper & Salt noise attack with the format of (image, noise density):(a) (Encrypted Lena image, 0.1); (b) (Encrypted Baboon image, 0.2); (c) (Encrypted Camera man image, 0.3); (d) (Encrypted Airplane image, 0.4); (e) Decrypted Lena image from (f); (g) Decrypted Baboon image from (h); (i) Decrypted Camera man image from (j); (k) Decrypted Airplane image from (l).

storage or during their transmission from one point to the other. Apart from that, some portion of an image may get lost, often called data crop attack. A nice encryption scheme has the inbuilt quality to bear the noise and data loss attacks. With different noise densities, Figures 12a to 12d depict the cipher images which have been adulterated by the noise of Pepper & Salt. These noise densities are respectively 0.1, 0.2, 0.3 and 0.4. Now to restore the original plain images, the decryption algorithm has been applied on these adulterated figures and the results obtained have been drawn in the Figures 12e to 12h. One can easily appreciate that a considerable original visual information is still intact.

Besides, using different data loss attacks, Figures 13a to 13c draw the encrypted images of Lena, Baboon and Camera man. Later on, these images were sent to the decryption machinery. Figures 13d to 13f plot the decrypted images so obtained. Again one can easily appreciate the original plain images. Hence we are justified in saying that the proposed image encryption and decryption schemes have the inbuilt ability to avert any threat of data loss and noise attacks.

K. ENCRYPTION TIME AND COMPLEXITY ANALYSIS

The proposed image cipher has been programmed under the Intel(R) Core(TM) i7-3740QM CPU @ 2.70GHz, RAM = 8.00 GB, System Type: 64-bit Operating System, x64-based processor. Moreover, Windows 10 and MATLAB R2016a are respectively the operating system and the programming language, which have been used in this study.

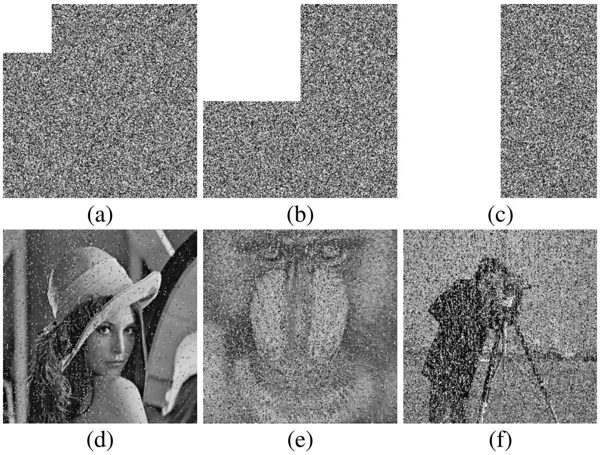


FIGURE 13. Data crop attack with format (type of image, name of image, data crop area): (a) (Encrypted, Lena, 64 × 64); (b) (Encrypted, Baboon, 128 × 128); (c) (Encrypted, Camera man, 256 × 128); (d) Decrypted Lena image from (e); (f) Decrypted Lena image from (g); (h) Decrypted Lena image from (i).

TABLE 13. The results of *MAE*.

Name of image	MAE
Lena	77.9631
Baboon	67.8594
Camera man	79.9358
Airplane	84.9533
Moon	66.7956
Clock	90.7784
Aerial	72.6714
Chemical plant	73.2251
Average	**76.7728**
Ref. [41]	77.5379
Ref. [42]	77.8772
Ref. [54]	82.8419
Ref. [56]	80.2

Certainly, the security is the topmost objective for any endeavor of cryptography. Encryption time, although less important than security, is also a very important consideration for the cryptographers. The ciphers having relatively less response time witness rosier prospects for their application in the industry. Normally the security and time are reciprocally interrelated. So, this is the job of the cryptographer to come up with a proper equilibrium between these two competing requirements while writing any cipher. In academia, two methods exist for the speed/performance analysis of the algorithms. These two methods are often called as empirical and theoretical methods. In the former method, the response time of the algorithm is directly figured out by using the timing devices like stopwatch. Table 14 shows the consumption of time by the cipher in encrypting the chosen images. The average time taken for all the images is 2.7236 seconds which is better than [43]. Besides, a relevant idea dubbed as encryption throughput (*ET*) is also discussed by researchers in this context. *ET* provides a sort of rate since it corresponds to the quantity of image being encrypted in the given time. The average value of *ET* for all the chosen images is 0.1941 which is better than 0.1707 [43].

Although empirical analysis is very simple, trivial and straightforward approach to measure the speed performance of the ciphers, but it is plagued with a number of loopholes. For instance, the time taken by the ciphers do not directly emanate from the body of algorithms, rather, it is 'polluted' due to the number of other extrinsic factors like particular input, underlying software and hardware, compiler *etc*. To transcend these particularities and limitations, a new setting is required which renders us a pure and innate performance of the algorithm under analysis. The latter or theoretical analysis provides us this setting. In it, Asymptotics [57] —a theory of mathematics, is usually employed.

TABLE 14. Algorithm's encryption speed and comparison with other schemes.

Algorithm	Image	Speed (sec)	Mbit/sec
Proposed	Lena	2.6624	0.2
	Baboon	2.6629	0.1907
	Camera man	2.6547	0.1975
	Airplane	2.6912	0.1979
	Moon	2.7923	0.1907
	Clock	2.8976	0.1838
	Aerial	2.7633	0.1927
	Chemical plant	2.6647	0.1998
	Average	**2.7236**	**0.1941**
Ref. [41]	Lena	-	4.87
Ref. [42]	Lena	-	1.28
Ref. [43]	Lena	3.1143	0.1707

Here the computational complexity for the generation of the chaotic data and the image cipher will be analyzed. The time consuming operations for converting the raw data given by the chaotic map into the five key streams, *i.e.*, $dist1$, $dist2$, $direction$, $mask - one$ and $mask - two$ take $\Theta(5mn)$ in Step 3 of the Section III-A. The time consuming parts of the algorithm ISUC contributes $\Theta(5mn)$ cost in Step 1 of the Section III-B. The cost for DNA encoding both the image and mask images is $\Theta(3mn)$ in Step 2. Further, the operations of DNA XOR and DNA Addition cost $\Theta(2mn)$ in Step 3. Lastly, the cost for converting the DNA encoded image to the decimal form is $\Theta(mn)$ in Step 4. By adding all these costs, the total cost is $\Theta(16mn)$, which is better than $\Theta(24mn)$ [41] and $\Theta(24mn)$ [42].

VI. CONCLUSION

The proposed image cipher has harnessed the potential of theory of chaos, DNA computing and chess piece Castle in one setting. Four dimensional chaotic map provided the streams of random numbers to perform the confusion and diffusion operations. The random movement of Castle on the hypothetical large chessboard scrambled the pixels of the given input gray scale image. To realize the diffusion effects, the scrambled image and the two streams of random numbers were converted into DNA strands followed by the DNA XOR and DNA Addition operations. These DNA level diffusion operations gave very promising results. SHA-256 hash codes have been introduced in the encryption algorithm in order to incorporate the plaintext sensitivity. Comprehensive security analyses and the computer simulation depict the robustness, impregnability and immunity to the diverse threats and the potential for some real world application of the encryption algorithm.

ACKNOWLEDGMENT

(Nadeem Iqbal and Rizwan Ali Naqvi are co-first authors.)

REFERENCES

[1] 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.

[2] 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.

[3] T. Sivakumar and R. Venkatesan, "A new image encryption method based on knight's travel path and true random number," *J. Inf. Sci. Eng.*, vol. 32, no. 1, pp. 133–152, 2016.

[4] N. Iqbal, S. Abbas, M. A. Khan, A. Fatima, A. Ahmed, and N. Anwer, "Efficient image cipher based on the movement of king on the chessboard and chaotic system," *J. Electron. Imag.*, vol. 29, no. 2, 2020, Art. no. 023025.

[5] 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.

[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, 2020.

[7] 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.

[8] 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.

[9] 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.

[10] M. Hanif, S. Abbas, M. A. Khan, N. Iqbal, Z. U. Rehman, M. A. Saeed, and E. M. Mohamed, "A novel and efficient multiple RGB images cipher based on chaotic system and circular shift operations," *IEEE Access*, vol. 8, pp. 146408–146427, 2020.

[11] J.-P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," in *The Theory of Chaotic Attractors*. Springer, 1985, pp. 273–312.

[12] G. Ambika, "Ed Lorenz: Father of the 'butterfly effect,'" *Resonance*, vol. 20, no. 3, pp. 198–205, Mar. 2015.

[13] Z. Hua, Y. Zhang, and Y. Zhou, "Two-dimensional modular chaotification system for improving chaos complexity," *IEEE Trans. Signal Process.*, vol. 68, pp. 1937–1949, 2020.

[14] Z. Hua, Z. Zhu, Y. Chen, and Y. Li, "Color image encryption using orthogonal Latin squares and a new 2D chaotic system," *Nonlinear Dyn.*, vol. 104, pp. 1–18, May 2021.

[15] L. Pray, "Discovery of DNA structure and function: Watson and Crick," *Nature Educ.*, vol. 1, no. 1, p. 100, 2008.

[16] J. D. Watson and F. H. C. Crick, "Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid," *Nature*, vol. 171, pp. 737–738, Apr. 1953.

[17] J. Wu, X. Liao, and B. Yang, "Image encryption using 2D Hénon-Sine map and dna approach," *Signal Process.*, vol. 153, pp. 11–23, Dec. 2018.

[18] M. A. B. Farah, R. Guesmi, A. Kachouri, and M. Samet, "A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation," *Opt. Laser Technol.*, vol. 121, Jan. 2020, Art. no. 105777.

[19] R. Enayatifar, F. G. Guimarães, and P. Siarry, "Index-based permutation-diffusion in multiple-image encryption using DNA sequence," *Opt. Lasers Eng.*, vol. 115, no. 3, pp. 131–140, 2019.

[20] H. Nematzadeh, R. Enayatifar, M. Yadollahi, M. Lee, and G. Jeong, "Binary search tree image encryption with DNA," *Optik*, vol. 202, Feb. 2020, Art. no. 163505.

[21] T. Li, J. Shi, X. Li, J. Wu, and F. Pan, "Image encryption based on pixel-level diffusion with dynamic filtering and DNA-level permutation with 3D Latin cubes," *Entropy*, vol. 21, no. 3, p. 319, Mar. 2019.

[22] J. Wu, J. Shi, and T. Li, "A novel image encryption approach based on a hyperchaotic system, pixel-level filtering with variable kernels, and DNA-level diffusion," *Entropy*, vol. 22, no. 1, p. 5, Dec. 2019.

[23] J. S. Khan, W. Boulila, J. Ahmad, S. Rubaiee, A. U. Rehman, R. Alroobaea, and W. J. Buchanan, "DNA and plaintext dependent chaotic visual selective image encryption," *IEEE Access*, vol. 8, pp. 159732–159744, 2020.

[24] Y. Sha, Y. Cao, H. Yan, X. Gao, and J. Mou, "An image encryption scheme based on IAVL permutation scheme and DNA operations," *IEEE Access*, vol. 9, pp. 96321–96336, 2021.

[25] X. Wang and Y. Li, "Chaotic image encryption algorithm based on hybrid multi-objective particle swarm optimization and DNA sequence," *Opt. Lasers Eng.*, vol. 137, Feb. 2021, Art. no. 106393.

[26] G. Ye, C. Pan, Y. Dong, Y. Shi, and X. Huang, "Image encryption and hiding algorithm based on compressive sensing and random numbers insertion," *Signal Process.*, vol. 172, Jul. 2020, Art. no. 107563.

[27] N.-R. Zhou, L.-X. Huang, L.-H. Gong, and Q.-W. Zeng, "Novel quantum image compression and encryption algorithm based on DQWT and 3D hyper-chaotic Henon map," *Quantum Inf. Process.*, vol. 19, no. 9, pp. 1–21, Sep. 2020.

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

[29] 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, 2015.

[30] W. Feng, Y. He, H. Li, and C. L. Li, "Cryptanalysis and improvement of the image encryption scheme based on 2D logistic-adjusted-sine map," *IEEE Access*, vol. 7, pp. 12584–12597, 2019.

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

[32] W. Feng and Y.-G. He, "Cryptanalysis and improvement of the hyper-chaotic image encryption scheme based on DNA encoding and scrambling," *IEEE Photon. J.*, vol. 10, no. 6, pp. 1–15, Dec. 2018.

[33] 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.

[34] W. Feng, Y.-G. He, H.-M. Li, and C.-L. Li, "Cryptanalysis of the integrated chaotic systems based image encryption algorithm," *Optik*, vol. 186, pp. 449–457, Jun. 2019.

[35] R. Lan, J. He, S. Wang, T. Gu, and X. Luo, "Integrated chaotic systems for image encryption," *Signal Process.*, vol. 147, pp. 133–145, Jun. 2018.

[36] W. Feng and J. Zhang, "Cryptanalzing a novel hyper-chaotic image encryption scheme based on pixel-level filtering and DNA-level diffusion," *IEEE Access*, vol. 8, pp. 209471–209482, 2020.

[37] Y. Chen and Y.-Q. Yang, "A new four-dimensional chaotic system," *Chin. Phys. B*, vol. 19, no. 12, Dec. 2010, Art. no. 120510.

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

[39] *Chess—Wikipedia*. Accessed: Aug. 14, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Chess

[40] *IEEE Standard for Binary Floating-Point Arithmetic*, Standard IEEE 754-2019, C/MSC-Microprocessor Standards Committee, American National Standards Institute, 1985.

[41] 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.

[42] 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.

[43] Z. Bashir, N. Iqbal, and M. Hanif, "A novel gray scale image encryption scheme based on pixels' swapping operations," *Multimedia Tools Appl.*, vol. 80, pp. 1–26, Jan. 2020.

[44] 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.

[45] M. A. Murillo-Escobar, M. O. Meranza-Castillón, R. M. López-Gutiérrez, and C. Cruz-Hernández, "Suggested integral analysis for chaos-based image cryptosystems," *Entropy*, vol. 21, no. 8, p. 815, Aug. 2019.

[46] X. Wei, L. Guo, Q. Zhang, J. Zhang, and S. Lian, "A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system," *J. Syst. Softw.*, vol. 85, no. 2, pp. 290–299, 2012.

[47] 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.

[48] 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*, vol. 153, pp. 117–134, Jan. 2018.

[49] 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, 2011.

[50] Y.-Q. Zhang and X.-Y. Wang, "A new image encryption algorithm based on non-adjacent coupled map lattices," *Appl. Soft. Comput.*, vol. 26, pp. 10–20, Jan. 2015.

[51] 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.

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

[53] 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.

[54] 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.

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

[56] 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, 2014.

[57] V. Ramesh and R. Gowtham, "Asymptotic notations and its applications," *Ramanujan Math. Soc., Math. Newsl.*, vol. 28, no. 4, pp. 10–16, 2017.

NADEEM IQBAL received the M.Phil. degree in computational science and engineering from the NUST, Islamabad, and the Ph.D. degree in computer science from NCBA&E, Lahore, Pakistan. He is currently working as an Assistant Professor with the Department of Computer Science and Information Technology, The University of Lahore (UOL), Lahore. Prior to joining the UOL, he worked in various academic institutions and has guided numerous bachelor's and master's students. His current research interests include multimedia encryption, computer graphics, and the philosophy of mathematics.

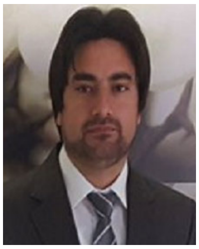
RIZWAN ALI NAQVI received the B.S. degree in computer engineering from COMSATS University, Pakistan, in 2008, the M.S. degree in electrical engineering from Karlstad University, Sweden, in 2011, and the Ph.D. degree in electronics and electrical engineering from Dongguk University, South Korea, in 2018. From 2011 to 2012, he was a Lecturer with the Computer Science Department, Sharif College of Engineering and Technology, Pakistan. He joined the Faculty of Engineering and Technology, The Superior College, Pakistan, as a Senior Lecturer, in 2012. From 2018 to 2019, he worked as a Postdoctoral Researcher with Gachon University, South Korea. He is currently working as an Assistant Professor with Sejong University, South Korea. His research interests include gaze tracking, biometrics, computer vision, artificial intelligence, machine learning, deep learning, and medical imaging analysis.

MUHAMMAD ATIF received the Ph.D. degree from Eindhoven University of Technology, The Netherlands, in 2011. He is currently an Associate Professor and the Head of the Computer Science and Information Technology Department, The University of Lahore. He is also working as an Assistant Director of the Office of Research, Innovation, and Commercialization (ORIC) and supervising various research and development projects. His research interests include the formal analysis of distributed algorithms and machine learning.

MUHAMMAD ADNAN KHAN received the B.S. and M.Phil. degrees from International Islamic University, Islamabad, Pakistan, by obtaining the Scholarship Award from the Punjab Information and Technology Board, Government of Punjab, Pakistan, and the Ph.D. degree from ISRA University, Pakistan, by obtaining the Scholarship Award from the Higher Education Commission, Islamabad, in 2016. He is currently working as an Assistant Professor with the Pattern Recognition and Machine Learning Laboratory, Department of Software, Gachon University, South Korea. Before joining Gachon University, he 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 also guiding five Ph.D. scholars and six M.Phil. scholars. He has published more than 190 research articles with Cumulative JCR-IF 290+ in international journals as well as reputed international conferences. His research interests include machine learning, MUD, image processing and medical diagnosis, and channel estimation in multi-carrier communication systems using soft computing.

MUHAMMAD HANIF received the B.S. degree in information technology from the University of Malakand, Pakistan, the M.S. degree in information technology from SEECS, NUST, Islamabad, Pakistan, and the Ph.D. degree in computer science from 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 for the past eight years. His research interests include images cryptography, computer graphics, networking, cloud computing, and the Internet of Things.

SAGHEER ABBAS received the Ph.D. degree from the School of Computer Science, NCBA&E, Lahore, Pakistan, in 2016. He is currently working as an Associate Professor with the School of Computer Science, NCBA&E. He has been teaching graduate and undergraduate students in computer science and engineering for the past eight years. He has published about 80 research articles with Cumulative JCR-IF 150+ in international journals as well as reputed international conferences. His research interests include cloud computing, the IoT, intelligent agents, image processing, and cognitive machines with various publications in international journals and conferences.



DILDAR HUSSAIN received the B.S. degree in computer science from Kohat University of Science and Technology, Pakistan, in 2010, and the Ph.D. degree in biomedical engineering from Kyung Hee University, South Korea, in 2019. From 2013 to 2019, he worked as a Research and Development Engineer with YOZMA BMTech Company Ltd., South Korea, where he developed diagnostic imaging equipment instruments, such as DXA, Chats X-rays, and Ultrasonic. He is currently working as a Postdoctoral Research Fellow with the School of Computational Science, Korea Institute for Advanced Study (KIAS), which is a subordinate institute of KAIST, South Korea. His research interests include bioinformatics, medical imaging, medical image analysis, computer vision, biomedical natural image processing, artificial intelligence, machine learning, deep learning, and mineral and nutritional study.