Hybrid chaos-based image encryption algorithm using Chebyshev chaotic map with deoxyribonucleic acid sequence and its performance evaluation

Jai Ganesh Sekar[1], Ezhumalai Periyathambi[2], Arun Chokkalingam[1]
[1]Department of Electronics and Communication Engineering, R.M.K. College of Engineering and Technology, Tamil Nadu, India
[2]Department of Computer Science and Engineering, R.M.D Engineering College, Tamil Nadu, India

Article Info

Article history:

Received Jun 1, 2022
Revised Aug 25, 2022
Accepted Aug 31, 2022

Keywords:

Deoxyribonucleic acid coding
Image encryption
Logistic Chebyshev map
Permutation and diffusion logistic map
Ribonucleic acid coding
SHA-256 hash function
Sine Chebyshev map

ABSTRACT

The media content shared on the internet has increased tremendously nowadays. The streaming service has major role in contributing to internet traffic all over the world. As the major content shared are in the form of images and rapid increase in computing power a better and complex encryption standard is needed to protect this data from being leaked to unauthorized person. Our proposed system makes use of chaotic maps, deoxyribonucleic acid (DNA) coding and ribonucleic acid (RNA) coding technique to encrypt the image. As videos are nothing but collection of images played at the rate of minimum 30 frames/images per second, this methodology can also be used to encrypt videos. The complexity and dynamic nature of chaotic systems makes decryption of content by unauthorized personal difficult. The hybrid usage of chaotic systems along with DNA and RNA sequencing improves the encryption efficiency of the algorithm and also makes it possible to decrypt the images at the same time without consuming too much of computation power.

Corresponding Author:

Jai Ganesh Sekar
Department of Electronics and Communication Engineering, R.M.K. College of Engineering and Technology
Thiruvallur, Tamil Nadu, India
Email: jaisekar71@gmail.com

1. INTRODUCTION

In this age of connected devices and vehicles, game streaming services, multimedia streaming services the internet traffic has gone up and moreover smart phones nowadays ships with high quality camera setups in them so the number of photos, videos that are taken by all has increased many folds then it was before some decades [1]. As these photos and videos will be stored anywhere like hard drives, flash drives or even in cloud they can be easily accessed if the security is compromised [2]. So, if any unauthorized user has access to their accounts, they cannot get useful information from that image as they will be in encrypted form [3]. Starting at now, numerous innovations are appropriate for guaranteeing a significant level of security of clinical images, for example, steganography [4], watermarking [5], and encryption [6]. Truth be told, to change a unique image into an unrecognizable one, encryption procedures can be utilized. Encryption of clinical images [7] is one of the most advantageous systems to ensure the security of patients' close to home data over open systems against pernicious assaults. Since clinical images are the private information of patients [8], guaranteeing their safe stockpiling and transmission has become a significant issue for clinical applications in true issues [9]. In the most recent decades [10], chaotic cryptosystems have been tremendously concentrated because of their arbitrary conduct, ergodicity, and affectability to mystery keys [11], an after effect of the underlying conditions and control parameters [12]. Accordingly, chaotic maps satisfy the great Shannon necessities concerning confusion and diffusion [13] and they are suited for issues in cryptography like image encryption [14].

Zhou *et al.* [15] proposed straightforward and powerful disorder-based encryption plot utilizing a mix of strategic and tent maps, in particular the logistic-tent system (LTS). Right off the bat, an arbitrary pixel is embedded toward the start of each line in the first image viable. Along these lines, each line is isolated into a one-dimensional (1D) framework, and a replacement procedure, in light of the LTS framework, is applied to each line. From there on, all 1D grids are joined into a 2D lattice as indicated by their line positions in the first image. At that point, the acquired 2D framework is pivoted 90 degrees counter-clockwise. The encoded image is at last acquired by rehashing these tasks multiple times. Another chaotic map-based encryption conspire is proposed. It is made out of two layers, i.e., a dispersion layer followed by a piece change layer the dissemination procedure, is iterated for specific number of times, depends on a paired lattice of size 32×32 [16]. The change is finished utilizing another adjusted 2-D chaotic cat map [17], running at the information bit level, and it recovers its dynamic key (i.e., beginning condition and control parameter) from a disordered generator. This is iterated for n number of times where n is defined by the algorithm. The whole procedure is rehashed until it arrives at the necessary security level [18].

Another image encryption plot named the tent-logistic guide-based information encryption calculation (TL-DEA) is proposed. The plan accomplishes two rounds of a scale independent pixel (SP)-organize on every pixel. A unique image is separated into information squares of a fixed length. At that point, replacement and change forms are performed on each square, utilizing the tent-logistic guide. All scrambled squares are joined to get the encoded image [19]. As of late, another clinical image encryption plot dependent on the edge maps of a source image is structured. It is named as medical image encryption (MIE)–using Bitwise XOR(BX) MIE-BX and comprises of three stages, in particular piece plane disintegration, age of an arbitrary succession, and change. A reversible decay technique is first applied to the info clinical image so as to create some piece of planes [20]. At that point, the edge maps (i.e., double frameworks with a similar size as the first piece planes) acquired from the source images are XORed with the first piece planes. At last, the bit places of all the got bit-planes are rearranged and afterward gathered together with a pixel dissemination activity, which creates an encoded image [21].

The chaos-based tent maps (CTM) with the rectangular change (RT) for an image encryption plot structure. The plan is made out of t pixel-stage adjusts, trailed by a pixel-dispersion layer. The pixel stages are performed utilizing the improved two-dimensional rectangular change, while the pixel-dispersion layer is constrained by chaos-based tent maps [22]. The use of deoxyribonucleic acid (DNA) computing has evolved and put into use in various fields, the more obvious field is cryptography. It has impressive features like exploitation of parallelism, whopping enormous storage, and ultra-efficient power consumption [23], [24], which makes it useful in encryption domain. Hence different encryption methodologies combining chaotic systems in combination with DNA encoding was proposed in recent times [25]. Image encryption plot dependent on DNA succession activities and disordered frameworks. The plan follows dispersion change engineering. The dissemination deals with the pixels and is constrained by pseudorandom groupings got from a spatiotemporal mayhem framework, in particular [26]. Subsequent to being DNA encoded, the befuddling image is utilized to refresh the underlying states of the coupled map lattice (CML), making the cryptosystem strong against known-plaintext and picked plaintext assaults. By utilizing the new starting conditions, the DNA-level stage working both on the lines and sections is performed. The permuted DNA network is then confounded indeed and DNA decoded to get the scrambled image [27].

What is more, a safe and proficient image encryption conspire dependent on self-versatile stage dissemination and DNA arbitrary encoding is proposed [28]. This plan is made out of n adjusts, every one of them comprising of four stages, to be specific DNA irregular encoding, self-versatile change, self-versatile dissemination, and DNA arbitrary interpreting [29]. The stage and dispersion strategies both work on the pixels, administered by the hyper-chaotic Lorenz framework, and upheld by a quantization procedure. The cryptosystem has great insusceptibility against plaintext assaults, as the quantization forms are upset by the inborn highlights of the first image [30]. In contrast to standard images, clinical images have an unmistakable element, i.e., they contain over 70% of 0's bits [31]. Consequently, in the first image, the higher piece planes are very like the lower ones, which makes defenseless the image cryptosystems with a high fixation to higher piece planes as the lower bit-planes contain critical data [32]. From the premise, the pixel adjustment turns out to be low productivity in the encryption procedure. In like manner, there is a desperation of a fitting and proficient image cryptosystem to oblige the test of 0's bits in clinical images. Moreover, it ought to react to the low multifaceted nature and high proficiency required by telemedicine applications [33], [34]. Babu *et al.* [35] have proposed pipelined SMS4 plain text encryption architecture. The paper focusing on how to encrypt the plain text faster than other existing algorithms. Babu *et al.* [36] have surveyed various SMS4 plain text encryption architecture and analyzed differential and side channel attacks on SMS4 architecture.

As of today, the advanced encryption standard (AES) is considered to be the safest encryption technique [37]. In AES, the data can be encrypted and decrypted using the same key. Hence this algorithm is called symmetric-key algorithm. These systems are not dynamic in nature and one key is used for both encryption and decryption [38]. It is difficult for traditional binary based computers to decrypt the image if key is not known but some companies have started using quantum computers which are many times powerful than traditional binary computers [39]. So, it would not take much time to decrypt the image encrypted by AES even if the key is not known. So, a complex and dynamic system like chaotic systems is used for encrypting the image in-order to protect the image from being decrypted by unauthorized person [40]. These systems are very sensitive to initial conditions, even a small value change say a number change decimal part can also result in completely different output by the system making it more difficult to decrypt the image if the input condition accuracy is not maintained up to decimal part.

The proposed work outperforms the existing chaotic image encryption algorithms and it has a high key space capacity up to 2^{761} which is a great characteristic to resist the brute force attacks. The proposed work has also achieved very low correlation among horizontal, vertical and diagonal adjacent pixels, which reflects the good diffusion property of an encryption algorithm. Few other analyses such as histogram analysis, number of pixel change ratio (NPCR) and unified average change intensity (UACI) tests also proves that the proposed algorithm outperforms the other encryption systems operates on the same grounds.

2. PRELIMINARIES

2.1. Logistic map

A logistic map can be represented by (1).

$$X_{i+1} = bX_i(1 - X_i) \tag{1}$$

where $X_n \in (0,1)$ *and* $b \in [0,4]$. Here, X_n and b are the independent and control parameter of the logistic map respectively and *i* ranges from 0, 1, 2, 3, 4, ..., *n*, where *n* is the number of pixels [41]. The chaotic nature of the logistic map is shown when the value of *b* is between 3.5699456 and 4 (including 4). But unfortunately, if value is 3.82 or 3.85 the number of values produced in the sequence is very less, so those two values cannot be used as control parameter as values generated using them does not have randomness [42]. So, a value closer to 4 is selected and used as control parameter for logistic map.

2.2. Logistic Chebyshev map

The logistic Chebyshev map is a one-dimensional which is the combination of logistic and Chebyshev seed maps. A logistic Chebyshev map can be represented by (2),

$$X_{n+1} = \left(aX_n(1 - X_n) + \frac{(4-a)}{4}\cos(c.\arccos(X_n))\right) mod\ 1 \tag{2}$$

here $X_n \in (0,1)$ *and* $a \in$) and where X_n and *a* are initial values and control parameters respectively. Here c ∈ ℕ which is the degree of Chebyshev map.

2.3. Sine Chebyshev map

The sine Chebyshev map is a one-dimensional which is the combination of sine and Chebyshev seed maps. A sine Chebyshev map can be represented by (3).

$$X_{n+1} = \left(aSin(\pi X_n) + \frac{(4-a)}{4}\cos(c.\arccos(X_n))\right) mod\ 1 \tag{3}$$

where $a \in (0,4]$,$X_n \in (0,1)$ *and* c ∈ ℕ here X_n and a are initial and control parameter respectively.

2.4. Deoxyribonucleic acid (DNA) coding

All the cells in the living organisms have DNA. The DNA has two strands that are referred to as polynucleotides. It has double helical structure and made up of five atoms namely hydrogen, nitrogen, phosphorus, carbon, oxygen [43]. The DNA strands are made of simple units called nucleotides which consist of nitrogenous base, sugar and phosphate group. The nitrogenous bases include adenine (A), guanine (G), thymine (T) and cytosine (C). The bases are of two types: purines and pyrimidines [44]. Purines includes adenine and guanine and pyrimidines includes thymine and cytosine. According to the Erwin Chargaff base pair rules, adenine of one strand binds with thymine of the other. Similarly, guanine of one strand binds with cytosine of the other. Based on the nitrogenous bases, a coding table is created which is used for encoding the pixels in the image [45]. Table 1 shows the DNA coding.

2.5. Ribonucleic acid (RNA) coding

RNA stands for ribonucleic acid. Similar to DNA, it is made up of nucleotides but it consists of a single strand. The nitrogenous bases include adenine (A), guanine (G), uracil (U) and cytosine (C) where adenine and guanine are purines and uracil and cytosine are pyrimidines. The complementary base to adenine is guanine and uracil is cytosine and vice versa. Similar to the DNA coding table, based on the nitrogenous bases coding table is created for encoding of pixels in the image. Table 2 shows the RNA coding.

Table 1. DNA encoding rule

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

Table 2. RNA encoding rule

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

3. PROPOSED ENCRYPTION METHOD

3.1. Generation of initial values and control parameters for the chaotic maps

The secure hashing algorithm (SHA) is a block cipher used to generate hash which is unique for given file. That is no two different files can have same hash as even a bit difference produces completely different hash. Here SHA-2(SHA-256) algorithm is used to generate hash. Let *xd0*, *ad0* be the initial condition and control parameter for Logistic Chebyshev map. Similarly, *xdd0*, *add0* be the initial and control parameter for sine Chebyshev map respectively. Let set of keys used for generating initial condition and control parameter be denoted by $k1$ and $k2$. The $k1$ key set is used to generate initial condition and control parameter for logistic Chebyshev map and similarly $k2$ key set is used for sine Chebyshev map. Let A and B be the hash generated by applying hash function to the input image and sum of the pixels respectively. Both the hashes will be in the form of hexadecimal values which is converted into binary form to obtain 256-bits for A as well as B. The 256-bits of A are split into 8-bits to form one key (since it is 256-bits we get 32 keys). The keys obtained are denoted by $k1 = k1\_1, k1\_2, k1\_3, \ldots\ldots k1\_32$. Similarly, the key set $k2$ is obtained and denoted by $k2 = k2\_1, k2\_2, k2\_3 \ldots, k2\_32$.

$$xd0 = \left(\frac{1}{3}\right)\left(x0 + 2 * \left(\frac{1}{256}\right) * bin2dec(k1_9 \oplus ,\ldots. \oplus k1_{16})\right) \tag{4}$$

$$ad0 = \frac{1}{3}\left(a0 + \frac{2}{64} * bin2dec(K1_1 \oplus \ldots.. \oplus K1_8)\right) \tag{5}$$

$$xd1 = \frac{1}{3}\left(x1 + \frac{2}{256} * bin2dec(k2_{17} \oplus ,\ldots., \oplus k2_{24})\right) \tag{6}$$

$$ad1 = \frac{1}{3}\left(a1 + \frac{2}{64} * bin2dec(k2_{25} \oplus ,\ldots, \oplus k2_{32})\right) \tag{7}$$

The key set 1 and 2 is applied to the (4)-(7) to get initial condition and control parameter for logistic Chebyshev map and sine Chebyshev map.

3.2. Generation of coordinates for shuffling of pixels

The new position of a pixel denoted by (*xx*, *yy*). The sum denotes the sum of all the pixels in the given image or given plane of an image. The new coordinates for pixels are created using the following steps.

Step 1: First a sequence s_1 is generated using (1). The initial and control parameter is given to the map based on the condition of the map. The *xx* coordinate is generated by applying the sequence to the (8).

$$xx_i = floor\left(mod\left(\left(s_1 * 10^8 + \frac{sumn}{(m*n*256)}\right), m\right)\right) \tag{8}$$

Step 2: Next a sequence s_2 is generated using (1). Different or same initial and control parameter is given to the map based on the specified parameter constraints of the map. The *yy* coordinate is generated by applying the equation to the (9).

$$yy_j = floor\left(mod\left(\left(s_2 * 10^8 + \frac{sumn}{(m*n*256)}\right), n\right)\right) \qquad (9)$$

All the pixels in the image are the moved to their new positions (i.e. (xx, yy)) to formed a shuffled image.

3.3. Encryption process

The proposed encryption algorithm involves two vital stages: generating initial values and control parameters for chaotic maps and generating coordinates for shuffling pixels. Chaos theory ensures unpredictability, as chaotic maps serve as the foundation for subsequent operations. Shuffling pixels introduces confusion and diffusion, making the encrypted data unrecognizable compared to the original. Multiple iterations further strengthen security. This algorithm harnesses chaos theory and spatial transformations to create a highly robust cryptographic scheme, protecting sensitive data from unauthorized access and attacks. As technology advances, the algorithm remains a cutting-edge solution for data security, ensuring confidentiality in today's data-driven world. It stands as a stalwart guardian, safeguarding valuable information from potential threats and adversaries. The below block diagram shows the encryption process taking place in different levels are shown in Figure 1. The encryption process is carried in serious of steps:

Step 1 : An image of dimension $M \times N \times P$, where M represents number of rows, N represents number of columns and P represents number of planes is taken as input. As we considered here is a RBG image the P value is 3. Let us assume initial value and control parameter be some value for $x0, a0, x1, a1$. Here *x* denotes initial value and a denotes control parameter.

Step 2 : The assumed initial values and control parameters are updated using (4) to (7).

Step 3 : Apply the updated initial conditions and control parameters to the respective maps (i.e., *xd0, ad0* to logistic Chebyshev map (2) and $xd1$, $ad1$ to sine Chebyshev map (3) to get required number of values ($M \times N$ number of values) using those maps. Let $M \times N$ values obtained by logistic Chebyshev be denoted by X and those $M \times N$ values obtained by sine Chebyshev map be denoted by Y.

Step 4 : Equate the values above 0.5 to 1 and values equal to 0.5 or lesser than 0.5 to 0 in both the maps (X and Y). Reshape X and Y to $M \times N$.

Step 5 : The input RBG image is split into three bands R, G and B bands (into three different images). The three planes are applied separately to (8) and (9) to shuffle the pixels in each plane (R plane, G plane and B plane). The planes obtained after shuffling are denoted SR, SG and SB matrix

Step 6 : The SR, SG and SB matrix are converted into 8-bit binary representation to get binary form pixel values of the matrix represented by BSR, BSG and BSB matrix.

Step 7 : The bits in BSR, BSG, BSB are encoded using randomly selected DNA coding rule to get matrix DER, DEG, DEB matrix. The X matrix obtained in step 4 is used to invert the DNA codes. The code in the DER, DEG, DEB matrix will be inverted if the value corresponding to the location in X is 1 else the code remains same (i.e., Let $DER\ (1,1) = A, X(1,1) = 1,\ then\ DER(1,1) = T, if\ X(1,1) = 0$ then there will be no chance in the code). From the DNA table it can be observed that A is always complementary to T and vice versa and similarly G is complementary to C and vice versa. The DNA inverted matrix is then decoded using same DNA decoding rule or different rule to get matrix DDR, DDG and DDR (binary representation)

Step 8 : For RNA encoding, the procedure is repeated as shown in Step 7, except instead of X matrix Y matrix is used and RNA coding table is used instead of DNA coding table. The resultant image matrix will be RDR, RDG and RDB matrix (binary representation)

Step 9 : The RDR, RDG and RDB matrix which is in the form of binary are converted into decimal form. The resultant matrix obtained is represented by R, G and B. The R, G and B matrix are combined to get the encrypted image.

3.4. Decryption process

The decryption process takes place in a reverse order of the encryption process. The general block diagram of decryption process is shown in Figure 2. The decryption process is carried in series of different steps.

Step 1: First the received encrypted image is read. The input encrypted image is then separated into different bands namely the R, G and B band. The R, G and B band images are converted into 8-bit binary.

Step 2: The RNA encoding (applying same encoding rule used in encryption part) is applied to the 8-bit binary images (R, G and B images). Then the initial conditions, control parameters used in encryption part is applied to sine Chebyshev map to generate the values. The obtained values are the equated to 0 or 1 based on the condition if the value at the position is greater than 0.5 it is equated to 1 and if it is lesser than or equal to 0.5 it is equated to 0. The obtained value set is used to invert the RNA code based on the control bit used on the encryption part. Then the matrix (R, G and B matrix) containing the RNA codes are decoded into binary form based on.

Step 3: Similarly, the initial conditions and parameters used for logistic Chebyshev map is used to generate the values and other steps are similar to step 2. After that the 8-bit binary values are converted into uint8 type.

Step 4: The initial and control parameters used in encryption process is used to generate the coordinates using logistic map equation to map the coordinates to their original positions. The output matrices (R, G and B matrix) are obtained.

Step 5: The obtained R, G and B matrices are combined to form the original image (the image before encryption was performed).

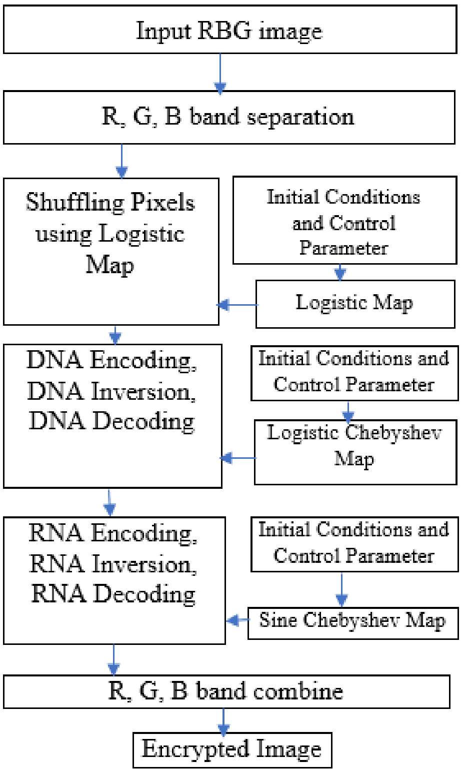

Input RBG image
R, G, B band separation
Shuffling Pixels using Logistic Map
Initial Conditions and Control Parameter
Logistic Map
DNA Encoding, DNA Inversion, DNA Decoding
Initial Conditions and Control Parameter
Logistic Chebyshev Map
RNA Encoding, RNA Inversion, RNA Decoding
Initial Conditions and Control Parameter
Sine Chebyshev Map
R, G, B band combine
Encrypted Image

Figure 1. Encryption process

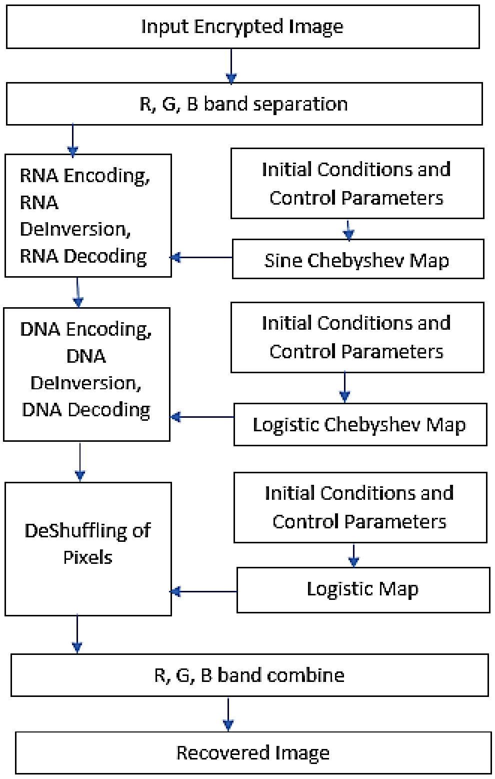

Input Encrypted Image
R, G, B band separation
RNA Encoding, RNA DeInversion, RNA Decoding
Initial Conditions and Control Parameters
Sine Chebyshev Map
DNA Encoding, DNA DeInversion, DNA Decoding
Initial Conditions and Control Parameters
Logistic Chebyshev Map
DeShuffling of Pixels
Initial Conditions and Control Parameters
Logistic Map
R, G, B band combine
Recovered Image

Figure 2. Decryption process

4. EXPERIMENTAL ANALYSIS

Some experiments are conducted and their results are used to demonstrate the encryption algorithm proposed here. All works were done in MATLAB 2018a in a desktop computer. The desktop computer has AMD Ryzen 5 2400G APU base clock 3.6 GHz overclockable up to 3.9 GHz, 8 GB DDR4 RAM and 1 TB HDD. All images are resized to 225×225 to make analysis easier.

Figure 3 depicts the various stages of encryption of a plain input image [46]. The plain Lena image is received at input and the same is depicted in Figure 3(a). Then the input RGB image is resized and then separated in to 3 band namely R band, G band and B band respective as shown in Figure 3(b). Further the band separated images are encrypted with DNA encoding sequence and then followed by RNA encoding the resulting outputs are shown in the Figures 3(c) and 3(d) respectively. Now to get the final Cipher image all the band separated images are combined together and the final encrypted image is shown in the Figure 3(e).

These cipher images are transmitted over the wireless channel and then the decryption stages are performed at the receiver side. When the cipher image as depicted in Figure 4(a) is passed through the decryption algorithm, they are band separated as RGB bands and then passed on to the RNA decoding sequence further followed by DNA decoding sequence that results in the output as depicted in the Figures 4(b) and 4(c) respectively. While deshuffling the pixels to the original positions the RGB bands of the original plain images are retrieved as depicted in the Figure 4(d). To get the original color image or RGB image the RGB band images are combined together to get the input color image as depicted in the Figure 4(e).

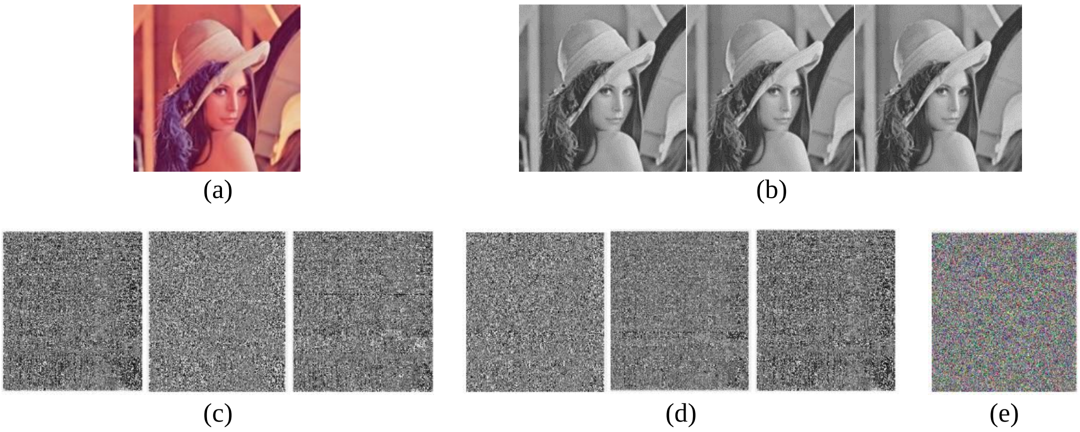


Figure 3. Encryption stages of an input image, (a) plain image input, (b) RGB band separation of plain image, (c) DNA encrypted bands, (d) RNA encrypted bands, and (e) combined encrypted image

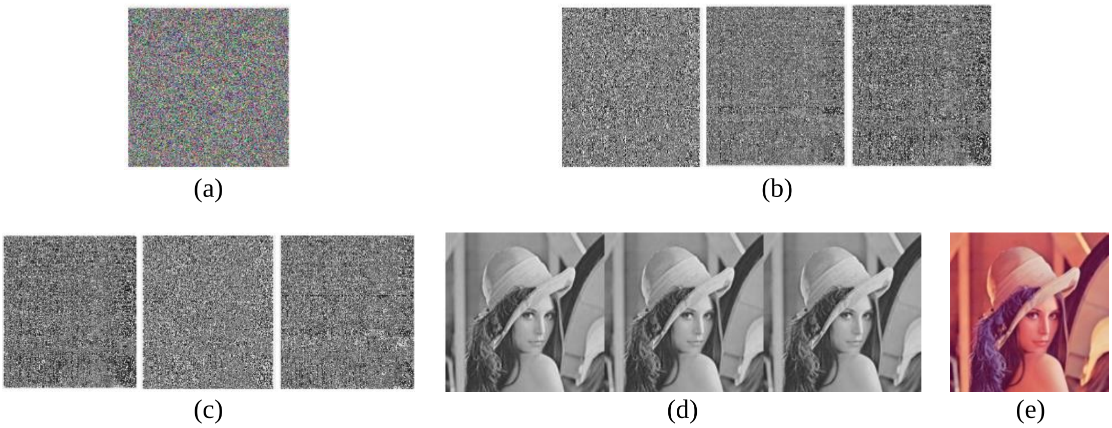


Figure 4. Decryption stages of input cipher image. (a) cipher image, (b) band separated RNA decoded images, (c) band separated DNA decoded images, (d) reshuffled RGB planes of plain image, and (e) estimated combined plain image

4.1. Key space analysis

The term “key space” is nothing but collection of all the keys that the encryption process could use. An encryption is said to be good only if it is sensitive to the cipher key used. In order to make brute force infeasible the key space must be large. The number of bits used for DNA and RNA encryption itself is 16 bits for single band, for 3 bands it triples upto48 bits. The key space of keys used in maps is 457 that is illustrated in Table 3. The total key space for the proposed solution is 2^{761} which is sufficient enough for most applications in order to resist brute force attack.

Table 3. Key space

	[37]	[39]	Proposed
Key Space	2^{295}	2^{716}	2^{761}

4.2. Correlation between adjacent pixels

Correlation is defined as the degree of closeness between two neighboring pixels. The original image has high correlation between adjacent pixels. Usually horizontal correlation, vertical correlation and diagonal correlation are measured for the original image and the encrypted image. An image encryption is said to be a good one if the cipher image obtained by the process has very low correlation between horizontal, vertical and diagonal pixels. The correlation coefficient of the plain image and cipher image are depicted in Table 4. The Table 5 compares various algorithms and proposed algorithm horizontal, vertical and diagonal pixels.

Table 4. Illustration of vertical, horizontal, and diagonal correlation of pixels between plain image and cipher image

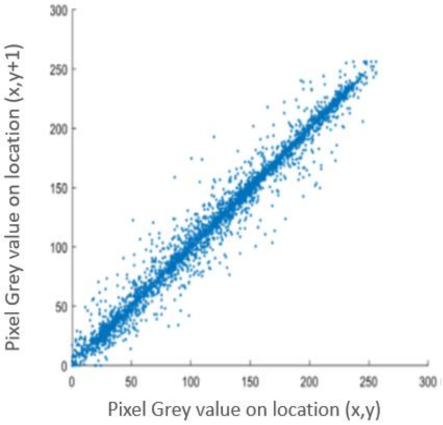
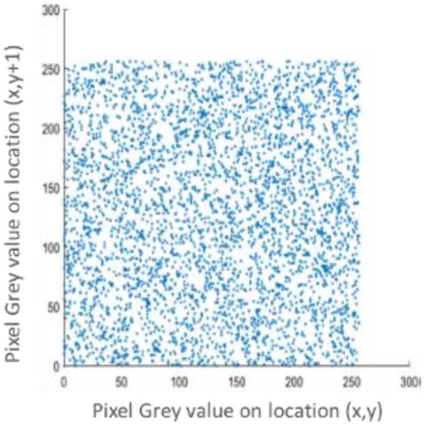
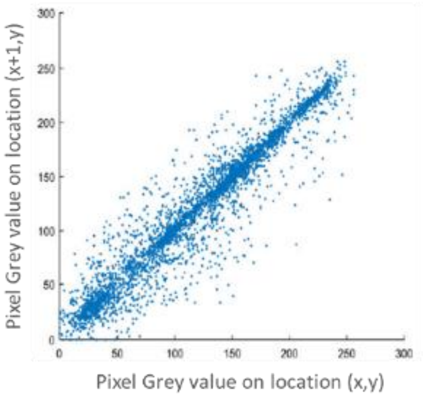
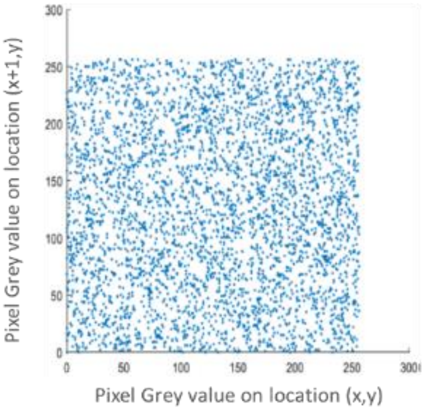
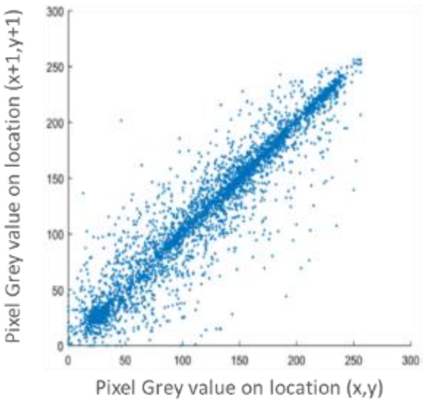
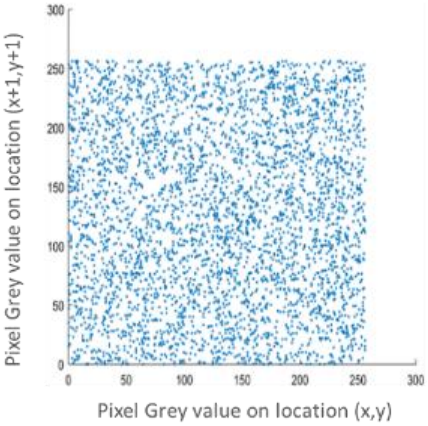
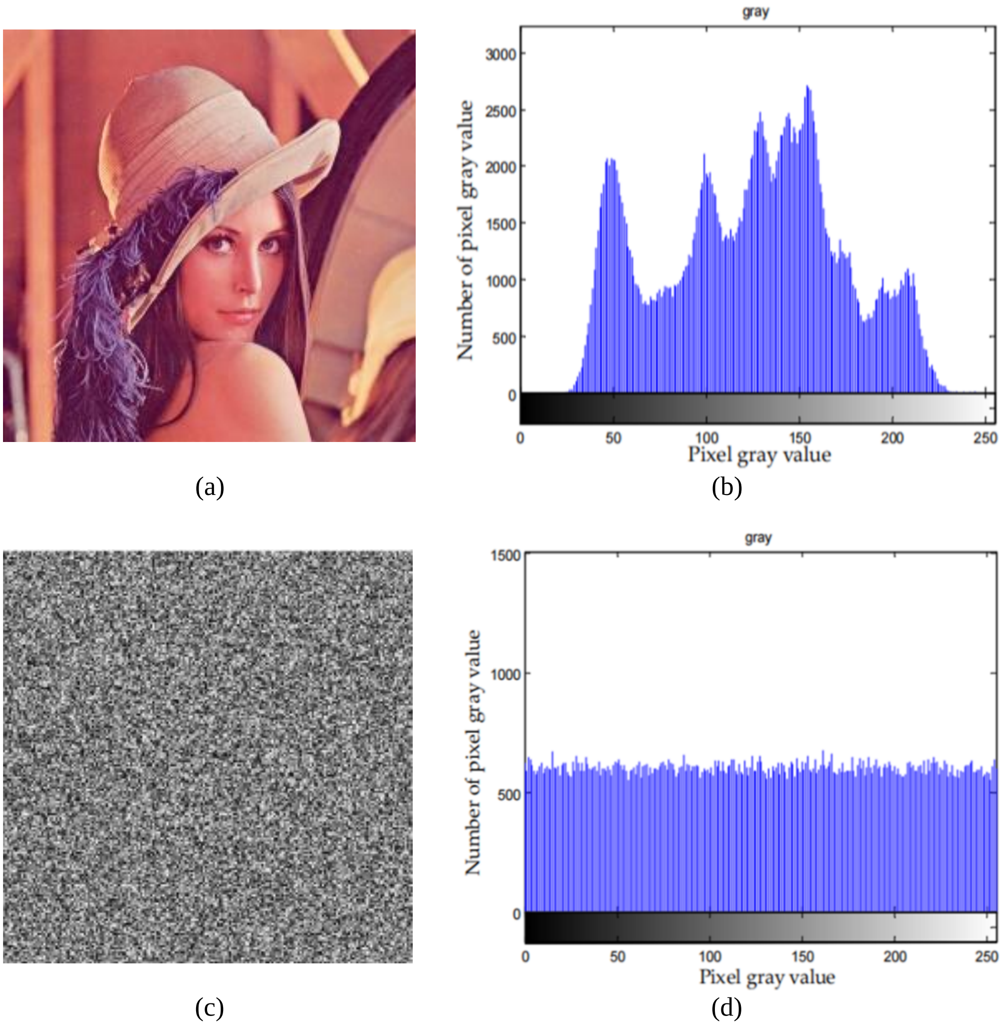
Description	Input Lena image	Cipher image of Lena
Input images		
Vertical correlation		
Horizontal correlation		
Diagonal correlation		

Table 5. Compares various algorithms horizontal, vertical and diagonal pixels

Direction	Horizontally	Vertically	Diagonally
Hua and Zhou [40]	0.0013	0.0007	0.0019
Wang *et al.* [27]	0.0007	0.0021	0.0149
Zhang *et al.* [34]	0.0035	0.0028	0.0010
Zhu *et al.* [33]	0.0058	0.0061	0.0059
Proposed	0.000059	0.0041	0.00013

4.3. Histogram test

A histogram of the image shows the distribution of the pixel values of an original image is shown in images Figure 5 represents the histogram analysis for Lena image where Figures 5(a) and 5(b) shows the plain image and its histogram of Lena image respectively. When the image is encrypted using the proposed algorithm the resulting cipher image and its histogram is depicted in the Figures 5(c) and 5(d), respectively. From the results it is evident that the proposed algorithm performs a better encryption, and the histogram of the cipher image is evenly distributed over entire values of the grey levels. Similarly, Figure 6 depicts the histogram analysis for the Baboon image. Figures 6(a) and 6(b) shows the plain image of baboon and its histogram, respectively. Irrespective of the input image the encryption algorithm performs well in all conditions and the cipher image and histogram of the cipher image for baboon is illustrated in the Figures 6(c) and 6(d), respectively. It is apparent from the results that histogram of the encrypted image is distributed evenly at all pixel levels, thereby providing better confusion and diffusion rate for the encryption scheme.



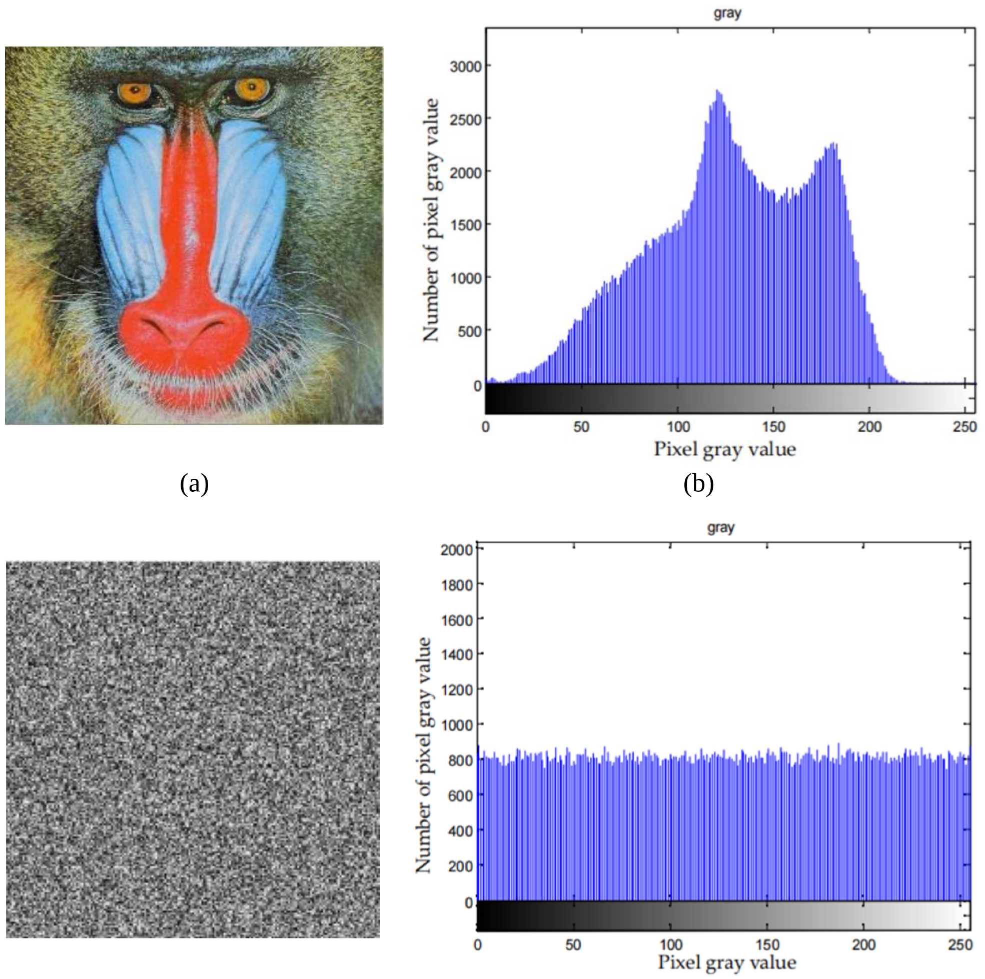
(a)

(b)

(c)

(d)

Figure 5. Histogram analysis of Lena image, (a) plain image of Lena, (b) histogram of plain Lena image, (c) cipher image of Lena and (d) histogram of Lena cipher image



(a) (b)

(c) (d)

Figure 6, Histogram analysis of baboon image, (a) plain image of baboon, (b) histogram of plain baboon image, (c) cipher image of baboon, and (d) histogram of baboon cipher image

5. CONCLUSION

This paper proposes a hybrid chaotic encryption system for color images while incorporating Chebyshev chaotic system along with DNA and RNA encoding to improve the confusion and diffusion parameters of encryption. The proposed algorithm is simulated on a laptop powered by 2.5 GHz AMD Ryzen 7 Processor with MATLAB simulation tool. The parameters such as Key sensitivity analysis, key space analysis, histogram analysis, NPCR and UACI analysis were also performed to validate the proposed algorithm. The proposed algorithm yields better security on color image encryption as compared to few other algorithms operated on the same grounds. The proposed algorithm achieved a better key space of 2^{761} which is far enough to resist the brute force attacks. Further the proposed method outperforms the superior algorithms in terms of Histogram analysis, NPCR and UACI respectively.

REFERENCES

[1] B. Feng, W. Lu, and W. Sun, “Secure binary image steganogra-phy based on minimizing the distortion on the texture,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 243–255, Feb. 2015, doi: 10.1109/TIFS.2014.2368364.

[2] S. Ahani and S. Ghaemmaghami, “Colour image steganography method based on sparse representation,” *IET Image Processing*, vol. 9, no. 6, pp. 496–505, Jun. 2015, doi: 10.1049/iet-ipr.2014.0351.

[3] N. M. Makbol, B. E. Khoo, T. H. Rassem, and K. Loukhaoukha, “A new reliable optimized image watermarking scheme based on the integer wavelet transform and singular value decomposition for copyright protection,” *Information Sciences*, vol. 417, pp. 381–400, Nov. 2017, doi: 10.1016/j.ins.2017.07.026.

[4] T. Zong, Y. Xiang, S. Guo, and Y. Rong, “Rank-based image water-marking method with high embedding capacity and robustness,” *IEEE Access*, vol. 4, pp. 1689–1699, 2016, doi: 10.1109/ACCESS.2016.2556723.

[5] G. Ye and X. Huang, “An image encryption algorithm based on autoblocking and electrocardiography,” *IEEE MultiMedia*, vol. 23, no. 2, pp. 64–71, Apr. 2016, doi: 10.1109/MMUL.2015.72.

[6] M. Dridi, M. A. Hajjaji, B. Bouallegue, and A. Mtibaa, “Cryptography of medical images based on a combination between chaotic and neural network,” *IET Image Processing*, vol. 10, no. 11, pp. 830–839, Nov. 2016, doi: 10.1049/iet-ipr.2015.0868.

[7] A. Al-Haj, G. Abandah, and N. Hussein, “Crypto-based algorithms for secured medical image transmission,” *IET Information Security*, vol. 9, no. 6, pp. 365–373, Nov. 2015, doi: 10.1049/iet-ifs.2014.0245.

[8] J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps,” *International Journal of Bifurcation and Chaos*, vol. 08, no. 06, pp. 1259–1284, Jun. 1998, doi: 10.1142/S021812749800098X.

[9] W. Cao, Y. Zhou, C. L. P. Chen, and L. Xia, “Medical image encryption using edge maps,” *Signal Processing*, vol. 132, pp. 96–109, Mar. 2017, doi: 10.1016/j.sigpro.2016.10.003.

[10] J. B. Lima, F. Madeiro, and F. J. R. Sales, “Encryption of medical images based on the cosine number transform,” *Signal Processing: Image Communication*, vol. 35, pp. 1–8, Jul. 2015, doi: 10.1016/j.image.2015.03.005.

[11] D. Levy, “Chaos theory and strategy: Theory, application, and managerial implications,” *Strategic Management Journal*, vol. 15, no. S2, pp. 167–178, Jun. 2007, doi: 10.1002/smj.4250151011.

[12] A. Belazi, R. Rhouma, and S. Belghith, “A novel approach to construct S-box based on Rossler system,” in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug. 2015, pp. 611–615, doi: 10.1109/IWCMC.2015.7289153.

[13] C. E. Shannon, “Communication theory of secrecy systems,” *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, Oct. 1949, doi: 10.1002/j.1538-7305.1949.tb00928.x.

[14] Z. Hua and Y. Zhou, “Design of image cipher using block-based scrambling and image filtering,” *Information Sciences*, vol. 396, pp. 97–113, Aug. 2017, doi: 10.1016/j.ins.2017.02.036.

[15] Y. Zhou, L. Bao, and C. L. P. Chen, “A new 1D chaotic system for image encryption,” *Signal Processing*, vol. 97, pp. 172–182, Apr. 2014, doi: 10.1016/j.sigpro.2013.10.034.

[16] Y. Zhou, Z. Hua, C.-M. Pun, and C. L. P. Chen, “Cascade chaotic system with applications,” *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 2001–2012, Sep. 2015, doi: 10.1109/TCYB.2014.2363168.

[17] A. Kanso and M. Ghebleh, “An efficient and robust image encryption scheme for medical applications,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 24, no. 1–3, pp. 98–116, Jul. 2015, doi: 10.1016/j.cnsns.2014.12.005.

[18] S. El Assad and M. Farajallah, “A new chaos-based image encryption system,” *Signal Processing: Image Communication*, vol. 41, pp. 144–157, Feb. 2016, doi: 10.1016/j.image.2015.10.004.

[19] X. Wu, B. Zhu, Y. Hu, and Y. Ran, “A novel colour image encryption scheme using rectangular transform-enhanced chaotic tent maps,” *IEEE Access*, p. 1, 2017, doi: 10.1109/ACCESS.2017.2692043.

[20] Z. Hua, S. Yi, and Y. Zhou, “Medical image encryption using high-speed scrambling and pixel adaptive diffusion,” *Signal Processing*, vol. 144, pp. 134–144, Mar. 2018, doi: 10.1016/j.sigpro.2017.10.004.

[21] L. M. Adleman, “Molecular computation of solutions to combinatorial problems,” *Science*, vol. 266, no. 5187, pp. 1021–1024, Nov. 1994, doi: 10.1126/science.7973651.

[22] G. Xiao, M. Lu, L. Qin, and X. Lai, “New field of cryptography: DNA cryptography,” *Science Bulletin*, vol. 51, no. 12, pp. 1413–1420, Jun. 2006, doi: 10.1007/s11434-006-2012-5.

[23] J. Chen, Z. Zhu, L. Zhang, Y. Zhang, and B. Yang, “Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption,” *Signal Processing*, vol. 142, pp. 340–353, Jan. 2018, doi: 10.1016/j.sigpro.2017.07.034.

[24] S. Sun, “A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling,” *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–14, Apr. 2018, doi: 10.1109/JPHOT.2018.2817550.

[25] Q. Zhang, L. Guo, and X. Wei, “A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system,” *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 18, pp. 3596–3600, Sep. 2013, doi: 10.1016/j.ijleo.2012.11.018.

[26] P. Zhen, G. Zhao, L. Min, and X. Jin, “Chaos-based image encryption scheme combining DNA coding and entropy,” *Multimedia Tools and Applications*, vol. 75, no. 11, pp. 6303–6319, Jun. 2016, doi: 10.1007/s11042-015-2573-x.

[27] X.-Y. Wang, Y.-Q. Zhang, and X.-M. Bao, “A novel chaotic image encryption scheme using DNA sequence operations,” *Optics and Lasers in Engineering*, vol. 73, pp. 53–61, Oct. 2015, doi: 10.1016/j.optlaseng.2015.03.022.

[28] X. Wu, H. Kan, and J. Kurths, “A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps,” *Applied Soft Computing*, vol. 37, pp. 24–39, Dec. 2015, doi: 10.1016/j.asoc.2015.08.008.

[29] X. Huang and G. Ye, “An image encryption algorithm based on hyper-chaos and DNA sequence,” *Multimedia Tools and Applications*, vol. 72, no. 1, pp. 57–70, Sep. 2014, doi: 10.1007/s11042-012-1331-6.

[30] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, “An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations,” *Signal Processing: Image Communication*, vol. 52, pp. 6–19, Mar. 2017, doi: 10.1016/j.image.2016.12.007.

[31] J. Chen, Z. Zhu, C. Fu, L. Zhang, and Y. Zhang, “An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 23, no. 1–3, pp. 294–310, Jun. 2015, doi: 10.1016/j.cnsns.2014.11.021.

[32] Y.-Q. Zhang and X.-Y. Wang, “Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation,” *Nonlinear Dynamics*, vol. 77, no. 3, pp. 687–698, Aug. 2014, doi: 10.1007/s11071-014-1331-3.

[33] Z. Zhu, W. Zhang, K. Wong, and H. Yu, “A chaos-based symmetric image encryption scheme using a bit-level permutation,” *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, Mar. 2011, doi: 10.1016/j.ins.2010.11.009.

[34] W. Zhang, K. Wong, H. Yu, and Z. Zhu, “A symmetric color image encryption algorithm using the intrinsic features of bit distributions,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 3, pp. 584–600, Mar. 2013, doi: 10.1016/j.cnsns.2012.08.010.

[35] M. Babu, C. Mukuntharaj, and S. Saranya, “Pipelined Sms4 cipher design for fast encryption using Twisted BDD S-box architecture,” *International Journal of Computer Applications & Information Technology*, vol. 1, no. 3, pp. 26–30, 2012.

[36] M. Babu and G. A. Sathish Kumar, “In depth survey on SMS4 architecture,” in *2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW)*, Dec. 2018, pp. 33–36, doi: 10.1109/I2C2SW45816.2018.8997162.

[37] X. Wang, S. Wang, Y. Zhang, and K. Guo, “A novel image encryption algorithm based on chaotic shuffling method,” *Information Security Journal: A Global Perspective*, vol. 26, no. 1, pp. 7–16, Jan. 2017, doi: 10.1080/19393555.2016.1272725.

[38] A. Belazi, M. Khan, A. A. A. El-Latif, and S. Belghith, “Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption,” *Nonlinear Dynamics*, vol. 87, no. 1, pp. 337–361, Jan. 2017, doi: 10.1007/s11071-016-3046-0.

[39] A. Belazi, M. Talha, S. Kharbech, and W. Xiang, “Novel medical image encryption scheme based on chaos and DNA encoding,” *IEEE Access*, vol. 7, pp. 36667–36681, 2019, doi: 10.1109/ACCESS.2019.2906292.

[40] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Information Sciences*, vol. 339, pp. 237–253, Apr. 2016, doi: 10.1016/j.ins.2016.01.017.

[41] S. S. Oleiwi, G. N. Mohammed, and I. Al_Barazanchi, "Mitigation of packet loss with end-to-end delay in wireless body area network applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 460–470, Feb. 2022, doi: 10.11591/ijece.v12i1.pp460-470.

[42] R. Majdoul, A. Touati, A. Ouchatti, A. Taouni, and E. Abdelmounim, "Comparison of backstepping, sliding mode and PID regulators for a voltage inverter," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 166–178, Feb. 2022, doi: 10.11591/ijece.v12i1.pp166-178.

[43] S. Murugan, G. Babu T R, and S. C, "Underwater object recognition using KNN classifier," *International Journal of MC Square Scientific Research*, vol. 9, no. 3, Dec. 2017, doi: 10.20894/IJMSR.117.009.003.007.

[44] E. P. Kannan and T. V Chithra, "Lagrange interpolation for natural colour image demosaicing," *International Journal o Advances in Signal and Image Sciences*, vol. 7, no. 2, pp. 21–30, Dec. 2021, doi: 10.29284/IJASIS.7.2.2021.21-30.

[45] A. P, M. Meenakumari, S. L, R. N, S. Jayaprakash, and S. Murugan, "Intelligent power control models for the IoT wearable devices in BAN networks," in *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, Jan. 2023, pp. 820–824, doi: 10.1109/IITCEE57236.2023.10090918.

[46] S. J. J. Thangaraj, N. Ramshankar, E. Srividhya, S. Jayanthi, R. Kumudham, and C. Srinivasan, "Sensor node communication based selfish node detection in mobile wireless sensor networks," in *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, Jan. 2023, pp. 1221–1226, doi: 10.1109/IITCEE57236.2023.10091048.

BIOGRAPHIES OF AUTHORS

Jai Ganesh Sekar is currently working as an Assistant Professor in Department of Electronics and Communication Engineering at RMK College of Engineering and Technology, Thiruvallur. He obtained his master's degree in Communication Systems from Anna University, Tamil Nadu. His research interests are wireless communication, cryptology, chaotic communications, and cyber security. He can be contacted at welcometojaimail@gmail.com.

Ezhumalai Periyathambi working as the Professor and Head of the Department of Computer Science and Engineering at R.M.D Engineering College, Chennai. He completed his B.E (CSE) in the year 1992 from University of Madras and M.Tech (CSE) in the year 2006 from J.N.T. University Hyderabad and Ph.D in Design and Implementation of High Performance Architecture for NoC System from Anna University, Chennai in the year 2012. He can be contacted at email ezhumalai.es@gmail.com.

Arun Chokkalingam completed hi PhD from Anna University in the year 2009. He is working as Professor in Electronics and Communication Engineering at RMK College of Engineering and Technology, Thiruvallur. His research areas include VLSI, digital signal processing, wireless communication and many more. He is a life member of many professional bodies such as IET, ISTE. He can be contacted at email arunece@gmail.com.