



# A novel DNA-based key scrambling technique for image encryption

Machbah Uddin<sup>1,2</sup> · Farah Jahan<sup>1</sup> · Mohammad Khairul Islam<sup>1</sup> · Md. Rakib Hassan<sup>2</sup>

Received: 1 February 2021 / Accepted: 26 August 2021 / Published online: 13 September 2021  
© The Author(s) 2021

## Abstract

Nowadays, data are the most valuable content in the world. In the age of big data, we are generating quintillions of data daily in the form of text, image, video, etc. Among them, images are highly used in daily communications. Various types of images, e.g., medical images, military images, etc. are highly confidential. But, due to data vulnerabilities, transmitting such images in a secured way is a great challenge. For this reason, researchers proposed different image cryptography algorithms. Recently, biological deoxyribonucleic acid (DNA)-based concepts are getting popular for ensuring image security as well as encryption as they show good performance. However, these DNA-based methods have some limitations, e.g., these are not dynamic and their performance results are far from ideal values. Further, these encryption methods usually involve two steps, confusion and diffusion. Confusion increases huge time complexity and needs to send one or more additional map tables with a cipher to decrypt the message. In this research, we propose a novel and efficient DNA-based key scrambling technique for image encryption that addresses the above limitations. We evaluate our proposed method using 15 different datasets and achieved superior performance scores of entropy, keyspace, cipher pixel correlations, variance of histogram, time complexity and PSNR. The experimental results show that our method can be used for image encryption with a high level of confidentiality.

**Keywords** DNA encryption · Image encryption · Novel key scrambling · Large key · DNA operation

## Introduction

This is the era of big data which indicates that we are generating data in every aspect of our digital life. Data may come in various formats including image, video, audio, number, text, etc. Among them, the world is accumulating image data enormously using different social applications. Due to the popularity of using image data, it is a major concern for researchers to keep data secured during storage and transmis-

sion. Such security is ensured by incorporating cryptography techniques where cryptography deals with changing a given data to a format that has no visual or textual meaning at a glance and sometimes it is in an unreadable format too.

Currently, a popular trend is to incorporate the concept of biological DNA operations for ensuring image security. Here, binary [37,39] or hexadecimal [21] number based different rules are used by the researchers for DNA mapping. Several other operations on DNA (e.g., XOR, addition, subtraction, etc.) are also applied. Such techniques are used for encryption and decryption of the entire image. As reported by many researchers, DNA-based techniques proved to be the most successful method for image encryption because of less time complexity and massive parallelism [7,11].

But in most cases, the DNA-based methods used map-based encryption techniques. Therefore, an additional table had to be sent to the receiver for decoding the encrypted image. On the other hand, in the case of pixel substitution, it is not required to send any table, but it is not highly resistive to different attacks. Several methods also used confusion and diffusion techniques for encryption, but the confusion process takes a huge time. Different studies used the scrambling techniques on the block, pixel, bit, row, or columns which

✉ Machbah Uddin  
machbah.csm@bau.edu.bd

Farah Jahan  
farah\_csc@cu.ac.bd

Mohammad Khairul Islam  
mkislam@cu.ac.bd

Md. Rakib Hassan  
rakib@bau.edu.bd

<sup>1</sup> Department of Computer Science and Engineering,  
University of Chittagong, Chittagong 4331, Bangladesh

<sup>2</sup> Department of Computer Science and Mathematics,  
Bangladesh Agricultural University, Mymensingh 2202,  
Bangladesh

needed huge time. Only one study used the key scrambling technique [24] by repeating its sequence with increasing numbers. So, it was not very difficult for attackers to assume the original message. Again, multiple DNA-based operations increase time complexity to a significant scale which is difficult to use in practical aspects. However, the performance results achieved by the existing studies are far from ideal values.

In this research, we propose a method for image encryption using a novel DNA-based key scrambling technique where we do not involve any confusion stage, and there is no need to send an additional table with cipher image to decrypt the message by the receiver. In our method, we receive an encryption key of any length for encrypting a plain image. Our proposed method generates a key of a size similar to the size of the image in bits from a given key using Section 3.1. Hence, every pixel of the input image can be encoded with a different key to generate a cipher.

The following list includes the significant contributions of our work:

- We propose a novel DNA operation-based image encryption technique.
- We introduce a novel key scrambling method for generating a large key from a given short key.
- Our method works dynamically by processing both the gray and color images.
- It can work with variable input key length.
- It does not involve any confusion stage, thus it reduces the encryption time.
- It can decrypt a cipher using the short input key only. Hence, there is no need to send one or more additional tables with a cipher image that requires high transmission bandwidth.
- We evaluate our method using 15 different datasets for justifying the robustness of our method. Among the 15 datasets, 12 are used in 12 existing studies and the remaining 3 are publicly available standard datasets.
- Our approach achieves competitive performance with the state-of-the-art techniques in terms of entropy, key space, variance of histogram, pixel correlations (e.g., horizontal, vertical and diagonal), PSNR, and time complexity.

Our proposed work is organized as follows. In the next section, we provide a background of existing studies on image encryption. In the subsequent section, we illustrate our proposed method in detail following which we discuss the datasets and experimental configurations. Performance evaluation metrics, experimental results in comparison with few state-of-the-art methods are presented in the penultimate section. Finally, we summarize our work, achievements, limitations and provide future directions.

## Background study

Recent trends show that image encryption techniques have successfully attracted researchers in this domain [7]. Every researcher is trying to overcome the performance of existing methods [7,11]. To understand the contributions of our proposed model, here we discuss the existing DNA-based encryption techniques with their strength and limitations.

Jithin et al. [11] instantiated a color image encryption technique based on the Arnold map, DNA encoding operation and a Mandelbrot set. They applied a chaotic map selection technique and chose Arnold's map among several available maps. Then they used multiple DNA encoding and decoding operations for a key generation which increased the time complexity remarkably. To perform the decryption operation, it required to send multiple tables like keymap, confusion and diffusion tables which were clearly burdensome. In addition, they used only one rule among eight rules for DNA encoding which was a limitation of their method. Because it always had an encoded value for an intensity value and if the hacker could know the encode value of any intensity, then he would be able to get an idea of a large part of the plain image.

Farah et al. [6] ushered chaotic map, fractional Fourier transformation (FRFT) and DNA XOR operation based image encryption technique. They performed DNA XOR operations on plain image four times and used a fixed 256-bit key which limited the key space of the overall system. The repeated XOR operation increased the system complexity. Dongming et al. [10] proposed a DNA operation based image encryption model where they generated a key image using a chaotic map. The key image and the plain image were encoded using DNA XOR operation and generated a disordered image. Later this image was sorted in ascending order of intensities and a cipher image was obtained with a table containing the actual order of pixel intensities. However, the system works only for gray images and their reconstruction rate was only 95%. In addition, they were required to send an additional table to decrypt the cipher image which was obviously a freight.

Wu et al. [39] proposed an improved chaotic map for the initial key generation which was the combination of Hanon and sine maps. Then they applied DNA XOR operation on the plain image and the map, and thus generated an initial cipher image. Further, they applied bit permutation techniques between plain image pixels and initial cipher images to get the final cipher image. They obtained satisfactory performance scores. However, the permutation of bits was used to take large computational time. Also, this technique was applied only for gray images. Wu et al. [41] introduced a new method for image encryption where they used NCA map based CML key and pseudo-random number generation. At first, they generated an SHA-256 hash code from a plain RGB image. Then they partitioned the RGB image into

three different channels, and each channel was encoded with different DNA rules. Subsequently, those three DNA matrices were merged into a new matrix and a row column based index diffusion technique was used to confuse the attackers. After that, the matrix was partitioned into  $N$  blocks and then DNA XOR, addition and subtraction operations were performed on each block. The sequence of operations on blocks was determined by a previously generated random number. Finally, a pixel-level permutation was applied to generate the cipher image. The system showed good results in different evaluation metrics. But the use of pixel permutation and multilevel DNA operations increased the time complexity of the approach.

Wang et al. [37] proposed an algorithm for image encryption based on the chaotic map. They converted the plain image pixels into DNA for diffusing the image. Later, they used DNA addition and subtraction rules for DNA permutation. Also, they used a bit shifting algorithm for creating confusion on pixels as well as on the entire image. After applying all these steps, they got the final cipher image. They achieved good performance scores. However, the confusion step of their method is time consuming. Wu et al. [40] used DNA operation based security for cloud CCTV systems. They used a hyperchaotic system for keymap generation which was used to diffuse the plain image. Besides, the DNA encoding technique was applied on plain images, and both the plain and encoded images were diffused together to prepare a cipher image. They achieved a good performance, but their system is limited to gray images and they need to send additional tables to decrypt the image.

Nabarun et al. [21] proposed an image encryption method based on DNA mapping where they converted the image pixels into three different text files e.g., encoded text (CODE\_FILE), code combinations (CODE), most frequent pixels (MOST\_OCCUR\_FILE) files where CODE and MOST\_OCCUR\_FILE files worked as secret key files. This method decrypted the image without any loss. But the method prepared three different files from a single image file and the CODE\_FILE size is very large in comparison with the original image size.

Nematzadeh et al. [22] conducted an experiment to secure image transmission over an unsecured network using a binary search tree (BST) and DNA encoding. They generated a secret key and then a candidate BST in a deterministic way by exploiting a logistic chaotic map. The BST was designed based on a random value with the basis of the chaotic map. Later, this BST is converted to DNA which is XORed with the plain image DNA. For the conversion of binary to DNA of both the BST and plain image, the encoding rule (among 8 rules) is determined by a chaotic map. They achieved high performance scores in different parameters. But their method needed to send the BST information to the receiver to decrypt

the image. Besides, an increase in the size of the key was used to increase the time complexity.

Wang et al. [34] introduced a coupled map lattices (CML) and DNA sequence operation based image encryption technique where they utilized bit shifting on even rows and columns of a DNA matrix. In the case of odd rows and columns, they used an index scrambling method for pixel diffusion. But bit scrambling operations took huge CPU time. Xingyuan Wang & Chuanming Liu [35] presented an image encryption technique using DNA-based operations. They applied double DNA encoding and decoding technique on the entire image and rotated the cipher image by 90 degrees. Due to double encryption and pixel confusion through 90 degree rotation, the time complexity increased largely for both the encryption and decryption process.

Belazi et al. [3] presented a system for medical image encryption. They divided the images into different blocks and then these blocks were permuted using Fermat's little theorem. They used a sine-Chebyshev map for key generation and utilized this map in DNA XOR, decode and complement operations. Finally, a cipher image was generated. The confusion step increased the time complexity and they conducted this experiment only for gray images. Yadollahi et al. [43] proposed nucleic acid operation based image security ensuring model only for gray images. They converted the DNA cipher image to an RNA image; for that, they converted a 2D DNA image to a 1D RNA image, then every six bits were mapped to an RNA codon. Finally, they used RNA complement operation and obtained the cipher image. However, multilevel encryption increases the time complexity.

Different researchers worked on scrambling techniques and they used pixel, bit, block, row and columns for scrambling. Zhongyun et al. [9] worked for encrypting medical images where they used randomized block-wise scrambling technique for confusion. Ramasamy et al. [25] proposed a new enhanced logistic map for the chaotic solution and performed a block scrambling and zigzag transformation for pixel confusion. Shuliang Sun [30] proposed an image encryption schema where they used pixel and bit-level scrambling technique based on a chaotic map. For all scrambling techniques, they used the confusion stage. Wang et al. [36] introduced an image encryption technique where they exploited Hash table structure scramble and DNA sequence operations. However, confusion techniques using pixel, bit, or block scrambling are time consuming. To the best of our knowledge, the key scrambling technique was used only by Prashan Premaratne and Malin Premaratne in [24] where they generated the required key from a given short key to encrypt the whole image by key repetition. Finally, they performed row–column, column–row and circular shifting. This technique was computationally effective as it did not involve any confusion stage. However, they applied their technique only on a single image, and they did not mea-

sure their performance concerning any standard encryption quality evaluation metric.

In the above paragraphs, we have mentioned the pros and cons of several existing studies. Now, we summarize the overall limitations of state-of-the-art techniques below:

- The number of images in datasets used by most of the studies was very small in numbers (usually, 4 to 10).
- The chaos based methods mostly depended on map selection and there might be enough room for a security breach.
- In general, chaos based solutions consisted of two steps e.g., confusion and diffusion where the confusion step increased significant time complexity.
- It was necessary to send an additional map table with cipher image to a receiver to decrypt the message which caused overhead and increased the transmission time.
- Many researchers used different types of scrambling techniques at the pixel, bit, block, row, or column level in their works. Only a single work used the key scrambling technique. To the best of our knowledge, no one used the scrambling technique with DNA-based image encryption technique.
- The performance scores of existing studies were far from the ideal scores. So, there is still enough space to improve these scores towards the ideal ones.

Being inspired by the existing studies, we propose a novel method for DNA-based image encryption. We introduce a key scrambling technique for encrypting each pixel in a dynamic way which proves to be capable of overcoming the above-mentioned limitations.

## Proposed methodology

In this section, we present our proposed method which contains insight into the key scrambling, image encryption and decryption processes.

### Key scrambling

Given a plain image  $I$  of  $l_I$  bits and an initial key  $K$  of length  $l_K$  bits, we aim to generate a large key  $G$  from  $K$  which is as large as the number of bits in the image  $I$ . The key scrambling process generates such a unique and large key  $G$  with a series of mutation, replication, append, reverse and rotation operations. Algorithm 1 describes the whole process of generating the key  $G$  of size  $l_I$ .

#### Key mutation:

It is a technique by which an existing key interchanges its bit positions to generate a new key. Here, two consecutive bits of  $K$  swap if they are not equal. This process is mathematically represented by Eq. 1. In this way, we can generate a large

### Algorithm 1: Key scrambling algorithm

---

**Input:** Key  $K$ , Required bits  $l_I$   
**Output:** Key  $G$  of  $l_I$  length

```

1  $R = \text{Replicate}(K)$ ;
2  $G = \text{Append}(G, R)$ ;
3  $R = \text{Reverse}(R)$ ;
4  $G = \text{Append}(G, R)$ ;
5 while  $\text{length}(G) < l_I$  do
6   for ( $i = 1 : (\text{length}(K) - 1)$ ) {
7     if ( $K_i \neq K_{i+1}$ ) AND ( $\text{length}(G) < l_I$ ) then
8        $\text{Mutation}(K_i \text{ and } K_{i+1})$ 
9        $S = \text{Replicate}(K)$ ;
10      for ( $j = 1 : \text{length}(S) \text{ AND } (\text{length}(G) < l_I)$ ) {
11         $R = \text{Replicate}(S)$ ;
12         $G = \text{Append}(G, R)$ ;
13         $R = \text{Reverse}(R)$ ;
14         $G = \text{Append}(G, R)$ ;
15         $\text{LeftRotation}(S)$ ;
16      }
17    }
18  }
19   $K = \text{Replicate}(G)$ ;
20 Return  $G$ ;

```

---

number of unique keys of size  $l_K$  from a single key.

$$K = K_i \Leftrightarrow K_{i+1} \text{ if } (K_i \neq K_{i+1}) \text{ and } i \in \{1, 2, \dots, (l_K - 1)\}, \quad (1)$$

where  $K_i$  represent the  $i^{\text{th}}$  bit in  $K$  and  $\Leftrightarrow$  operator represents a swap operation between the  $i^{\text{th}}$  and  $(i + 1)^{\text{th}}$  bits.

#### Key replication:

Replication is a process which creates a copy of a key. The replication operation  $R = K$  is performed using Eq. 2.

$$R_i = K_i \text{ where } i \in \{1, 2, \dots, l_K\}. \quad (2)$$

#### Key append:

This process appends a new key at the end of an existing sequence. Let, the existing sequence be  $G$  of length  $l_G$  and the candidate sequence to be appended be  $R$  of  $l_R$  length. The append operation is performed as per Eq. 3.

$$G_{l_G+i} = R_i \text{ where } i \in \{1, 2, \dots, l_R\}. \quad (3)$$

#### Key reverse:

This process generates a new key from a given key. In this process, we organize the bits in the given key in reverse order. Let, the target key be  $R$ . The key reverse is accomplished using Eq. 4.

$$R_i = K_{l_K-i+1} \text{ where } i \in \{1, 2, \dots, l_K\}. \quad (4)$$

#### Key left rotation:

The left rotation indicates an event by which all bits change their positions in a key where the MSB bit moves to LSB

**Fig. 1** Pictorial view of a key scrambling technique for large key generation from the given input key “101” using Algorithm 1 where the target length is 50

Key Processing	Steps	Large Key (G) Generation
Input Key, K	Phase 1	
Replicate	Phase 2	Append
	Phase 3	Reverse Append
Mutation of 1st & 2nd bits	Phase 4	
Replication of Key	Phase 5	Append
	Phase 6	Reverse Append
Key Left Rotation	Phase 7	Append
	Phase 8	Reverse Append
Key Left Rotation	Phase 9	Append
	Phase 10	Reverse Append
Key Left Rotation	Phase 11	Original state, no append, go for mutation
Mutation of 2nd & 3rd bits	Phase 12	Not Possible, two bits are same
New Key K is G		
Replicate the 24-bit G to K and repeat all the above steps until reaching the target length		

and every other bit performs a 1-bit left shift operation. Let, the given key be  $S$  of length  $l_S$ . The left rotation process is performed using Eq. 5. In this equation, we temporarily store the MSB,  $S_1$  in a buffer  $t$ , then shift the rest of the bits one bit left. Finally, we replace the LSB with the buffer  $t$ .

$$\begin{aligned}
 t &= S_1 \\
 S_i &= S_{i+1} \text{ where } i \in \{1, 2, \dots, (l_S - 1)\} \\
 S_{l_S} &= t.
 \end{aligned}
 \tag{5}$$

Let  $K = 101$  and  $l_I = 50$ . Figure 1 depicts the key processing steps of Algorithm 1 as an example.

**Image encryption**

Given a plain image  $P$  with height, width and channel of  $h, w$  and  $d$ , respectively and a key  $K$  with length of  $l_K$  bits and a pixel at location  $(i, j)$  and its intensity at channel  $k$  be denoted respectively by  $P(i, j)$  and  $P(i, j, k)$  where  $i, j$  and  $k$  represent the row, column and channel numbers of the pixel respectively. At first, we generate a large key  $G$  of the length of  $l_P$  bits as in Eq. 6 using the key scrambling technique as described in Section 3.1.

$$l_P = h \times w \times d \times 8.
 \tag{6}$$

To perform a DNA operation on the intensity  $P(i, j, k)$  using a key, we select a DNA Rule  $r$  using Eq. 7.

$$r = \text{rem}(((i - 1) \times w) + (j - 1), 8) + 1.
 \tag{7}$$

To encrypt the 8-bit intensity  $P(i, j, k)$ , we select an 8-bit key segment  $Y$  from  $G$  using Eq. 8.

$$\begin{aligned}
 \text{keyPos} &= ((i - 1) \times w + (j - 1)) \times 24 + ((k - 1) \times 8) + 1 \\
 Y &= G(\text{keyPos} : \text{keyPos} + 7).
 \end{aligned}
 \tag{8}$$

The detail of our proposed encryption algorithm is presented in Algorithm 2 and the overall encryption system is depicted in Fig. 2.

**Algorithm 2:** DNA-based image encryption

```

Input: Plain Image  $P$ , Given Key  $K$ 
Output: Cipher Image  $C$ 
1  $[h, w, d] = \text{size}(P)$ 
    $h, w$  and  $d$  are height, width and channel of  $P$ 
2  $l_I = h \times w \times d \times 8$ ;
3  $G = \text{KeyScrambling}(K, l_I)$ ;
4 for ( $i = 1 : h$ ) {
5   for ( $j = 1 : w$ ) {
6      $r = \text{rem}(((i - 1) \times w) + (j - 1), 8) + 1$ ;
7     for ( $k = 1 : d$ ) {
8        $\text{keyPos} = ((i - 1) \times w + (j - 1)) \times 24 + ((k - 1) \times 8) + 1$ ;
9        $Y = G(\text{keyPos} : \text{keyPos} + 7)$ ;
10       $\text{pixDNA} = \text{DNAEncode}(\text{pixBit}, r)$ ;
11       $\text{keyDNA} = \text{DNAEncode}(Y, r)$ ;
12       $\text{xoredDNA} = \text{DNAXOR}(\text{pixDNA}, \text{keyDNA})$ ;
13       $\text{valDecode} = \text{DNADecode}(\text{xoredDNA}, r)$ ;
14       $\text{valueDec} = \text{Bin2Dec}(\text{valDecode})$ ;
15       $C(i, j, k) = \text{valueDec}$ ;
16    }
17  }
18 }
19 Return  $C$ ;
    
```

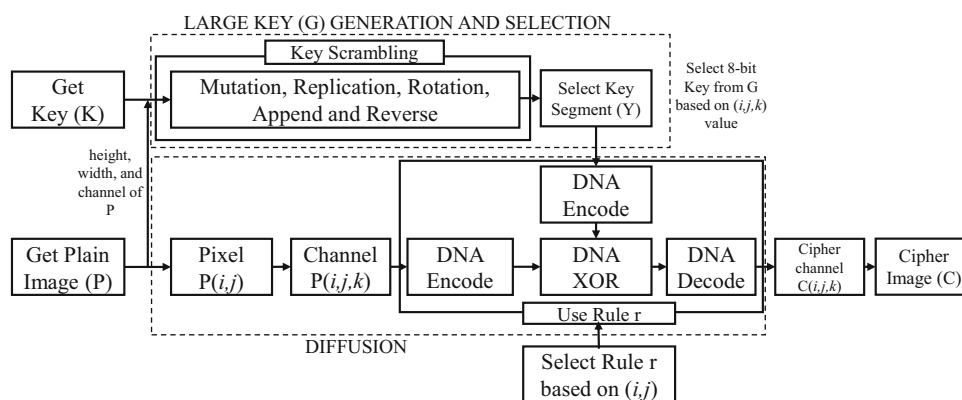
**DNA encode:**

Generally, DNA consists of four different chemical bases e.g. Adenine (“A”), Guanine (“G”), Cytosine (“C”) and Thymine (“T”) [22]. Again, each base has two strains. Some researchers [3,10] proposed mapping of DNA base to two binary bits. In this way, a total of 24 different mappings are possible. But due to the complementary property of base pairs, only 8 rules are possible [3,10,28] which are shown in Table 1.

In our proposed model, given a string of bits and a rule, our system returns a corresponding DNA sequence. For example, if a pixel channel intensity is  $163_{10} = 10100011_2$  and the Rule-2 is selected, the encoded DNA will contain the bit sequence 10, 10, 00 and 11 which represents “G”, “G”, “A” and “T” respectively. Thus, the final DNA sequence is “GGAT”.

**DNA XOR:**

**Fig. 2** Block diagram of the proposed image encryption system



**Table 1** DNA encoding rules

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

**Table 2** DNA XOR operations

XOR	A	T	C	G
A	A	T	C	G
T	T	A	G	C
C	C	G	A	T
G	G	C	T	A

Being inspired by binary XOR operations [10], XOR operations on DNA sequences are incorporated for image encryption. Given two DNA sequences, the DNA XOR operation between the sequences returns one DNA. Let us consider two sequences “TACC” and “CATG”. Then, DNA XOR operation takes place between the 1<sup>st</sup> bases of the 1<sup>st</sup> and 2<sup>nd</sup> sequences, i.e., “T” and “C”. As per Table 2, the XOR operation yields “G”. Similarly, the XOR operations for 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> bases yield “A”, “G” and “T”, respectively. Hence, the final DNA sequence is “GAGT”.

**DNA decode:**

It is the reverse process of DNA encode [10]. This time we receive a DNA sequence and a rule. Our system generates the corresponding binary bits as depicted in Table 1. Let us consider a DNA sequence “ATTC” and Rule-5. Our system, then generates the bit sequences 11 for “A”, 01 for “T”, 01 for “T” and 00 for “C”. Hence, the final bit string is 11010100<sub>2</sub>.

**Image decryption**

Given a cipher image, an image decryption process is supposed to generate the original plain image. An appropriate decryption algorithm reconstructs the image perfectly, i.e.,

100% similar to the plain image. Computationally, image decryption is an opposite sequence of operations of the encryption process 3.2. Algorithm 3 presents the working procedure of image decryption.

**Algorithm 3:** DNA-based image decryption.

```

Input: Cipher Image C, Received Key K
Output: Decrypted Image D
1 [h, w, d] = size(C)
   h, w and d are height, width and channel of the image C
   respectively
2 lI = h × w × d × 8 ;
3 G = KeyScrambling(K, lI);
4 for ( i = 1; i <= h; i = i + 1 ) {
5     for ( j = 1; j <= w; j = j + 1 ) {
6         r = rem(((i - 1) × w) + (j - 1), 8) + 1;
7         for ( k = 1; k <= d; k = k + 1 ) {
8             keyPos = ((i - 1) × w + (j - 1)) × 24 + ((k - 1) × 8) + 1;
9             Y = G(keyPos : keyPos + 7);
10            pixBit = Dec2Bin(C(i, j, k));
11            pixDNA = DNAEncode(pixBit, r);
12            keyDNA = DNAEncode(Y, r);
13            xorResult = DNAXOR(pixDNA, keyDNA);
14            valDecode = DNADecode(xorResult, r);
15            value = Bin2Dec(valDecode);
16            D(i, j, k) = value;
17 Return D;

```

Figure 4 presents the decryption process of the cipher of the red channel of the first pixel of the sample image used as an example of our encryption process in Fig. 3.

**Dataset and experimental configurations**

We validate our method using two categories of datasets. In the first category, we select some existing methods related to our work and choose datasets used by them. The second category consists of publicly available three standard image datasets. For the first category, we compare the perfor-

Location	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	
Key Bits	1	1	0	1	1	1	0	1	1	1	1	0	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0	1	0	1	1	0
Key DNA	T	G	T	G	T	C	G	G	C	C	T	C	A	C	C	T	C	T	C	T	G	C	C	G	C	A	C	C	T	C	T	G	C	C	G														
Pixel Bits	0	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0	0	0	1	1	1	
Pixel DNA	G	G	T	T	T	T	A	T	C	T	G	T	C	C	T	T	T	T	A	G	G	A	C	T	C	A	C	C	A	C	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A			
Pixel Num	Pixel 1											Pixel 2																																					
Rule Num	Rule 1											Rule 2																																					
XOR Res	C	A	A	C	A	G	G	C	A	G	C	G	C	A	G	A	G	A	C	A	G	A	G	A	C	C	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C		
Cipher Bits	1	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	

(a)

Location	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Key Bits	1	0	1	1	1	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1	1	0	0	0	0	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0	1
Key DNA	C	T	C	T	T	A	C	T	G	G	T	A	A	C	C	T	C	T	C	T	C	T	C	T	C	T	C	T	G	C	C	G																
Pixel Bits	0	1	0	1	1	1	1	1	0	1	1	0	0	1	1	0	1	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	0	0	1	1	0	1	1	0	0
Pixel DNA	G	G	T	T	T	G	C	G	C	C	G	A	G	C	C	T	T	T	T	A	G	G	A	C	T	C	A	C	A	C	T																	
Pixel Num	Pixel 9											Pixel 10																																				
Rule Num	Rule 1											Rule 2																																				
XOR Res	T	C	G	A	C	C	T	G	T	A	T	G	C	A	G	A	G	A	C	C	A	C	C	A	C	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C		
Cipher Bits	1	1	1	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	1

(b)

Fig. 3 An example of image encryption with a 192-bit key. Encryption steps and results of a 1st and 2nd pixels, and b 9th and 10th pixels. It can be noted that the red channel of the 1st and 9th pixels has the same

intensity of 95<sub>10</sub>, and Rule-1 is used on both of them. The resultant ciphers for the two pixels’ red channels are very different

Fig. 4 The decryption process of the cipher of the red channel of the first pixel of the sample image (Fig. 3) is used as an example of our encryption process. The red, green and blue columns represent the cipher of the red, green and blue channels of the pixel

		First Pixel																						
Location	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Key Bits	1	1	0	1	1	1	0	1	1	1	1	0	0	1	0	1	1	0	1	0	1	1	1	0
Key DNA	T	G	T	G																				
Cipher pixel Bits	1	0	0	0	0	0	1	0																
Cipher pixel DNA	C	A	A	C																				
Pixel Num	Pixel 1																							
Rule Num	Rule 1																							
XOR Res	G	G	T	T																				
Decrypted Bits	0	1	0	1	1	1	1	1																

mance of our approach with some existing approaches listed in Table 3. These approaches considered different image datasets with variations in the number of images, resolutions and image types (e.g., gray or color).

We use the second category of the datasets to prove the robustness of our method. Here, we take three more standard datasets in addition to the datasets of existing studies. These are (i) image enhancement dataset, (ii) USC-SIPI dataset and (iii) information hiding datasets where (i) and (iii) contain only color images and (ii) contains both the gray and color images. The specifications of these datasets are given in Table 4. We prepare 13 subsets from these datasets considering the different properties of the images (e.g., size, color, etc.) to conduct experiments on different setups listed in Table 5 and the detailed result is presented in 5.12.

### Experimental results and discussion

This section discusses different performance evaluation metrics, results obtained using our approach and performance comparison with the-state-of-art methods. We consider different evaluation metrics, some of which we choose from existing studies, e.g., key space [4,11,22,34,35,41,44,48], key sensitivity [4,11,22,34,35,41,44,48], histogram analysis [4,6,22,34,41,44,48], variance of histogram [11], entropy [4,6,11,22,34,35,41,48], cipher pixels’ correlation [4,6,11,22,34,35,41,44,48], resisting differential attacks [4,6,11,22,34,35,41,48], PSNR [11,34,48], noise attack [11,34,41,44], image reconstruction [5,18,48] and time complexity [4,16,34,41]. We also use variable length key and statistical parameters as evaluation metrics, and applied them in publicly available datasets. Although it is not necessary to use all these evaluation metrics, we analyze them to show the robustness of our proposed method from different angles.

**Table 3** Datasets collected from recent existing studies

Existing studies	Description of dataset images
Jithin <i>et al.</i> [11]	10 color images (Lena, Baboon, Barbara, Corn Field, Flower, Lake, Monarch, Peppers, Soccer and Yacht) of dimensions $256 \times 256$ .
Nematzadeh <i>et al.</i> [22]	8 gray images (Peppers, House, Airplane, Cameraman, Lena, Boat, Painter and Baboon) of 3 different dimensions e.g., $512 \times 512$ , $256 \times 256$ and $128 \times 128$ .
Wang <i>et al.</i> [34]	4 gray images e.g., Lena, Cameraman, Pepper and House of dimensions $256 \times 256$ .
Chen <i>et al.</i> [4]	Single Lena image of dimension $256 \times 256$ .
Wang <i>et al.</i> [35]	9 gray images (Bird, Boat, Brain, Finger, House, Lena, Moon, Peppers and Plain).
Nabarun <i>et al.</i> [21]	Color Lena image of dimension $256 \times 256$ .
Farah <i>et al.</i> [6]	3 color images e.g., Lena, Jasmin and Baboon of dimensions $256 \times 256$ .
Wu <i>et al.</i> [40]	3 gray images e.g., Boat, Fruit and Baboon of dimensions $760 \times 576$ , $444 \times 336$ and $512 \times 512$ respectively.
Wang <i>et al.</i> [36]	5 images (3 gray from USC-SIPI dataset, 1 black and 1 white) of dimensions $256 \times 256$ .
Yadollahi <i>et al.</i> [43]	8 gray images (Baboon, Boat, Cameraman, House, Lena, Barbara, Snow and Peppers) of 2 different dimensions $512 \times 512$ and $256 \times 256$ .
Ye <i>et al.</i> [45]	3 images e.g., Lena, Barb and Baboon of dimensions $256 \times 256$ , $512 \times 512$ and $512 \times 512$ , respectively.
Dongming <i>et al.</i> [10]	Gray Boat image of dimension $256 \times 256$ .

**Table 4** Specifications of the datasets used for robustness testing

Sl no	Dataset	No of images	Type	Dataset description
1	Image enhancement ( <a href="#">URL</a> )	24	Color	Busting image enhancement and tone-mapping algorithms: The most challenging cases
2	USC-SIPI ( <a href="#">URL</a> )	Total 30 (Gray 15, Color 15)	Gray Color	To support research in image processing, image analysis and machine vision
3	Information hiding ( <a href="#">URL</a> )	22	Color	To facilitate image hiding, encryption and watermarking research

**Table 5** Different setups for robustness testing with their acronyms

No	Setup	Number of images	Considered subset images
1	ImgEnhc	24	All images of Image Enhance dataset
2	USCSIPI	30	All images of USC-SIPI dataset
3	InfHide	22	All images of Information Hiding dataset
4	Col	61	All color images of three datasets
5	Gray	15	All gray images of USC-SIPI dataset
6	Col_256	14	Only $256 \times 256$ color images of three datasets
7	Col_512	21	Only $512 \times 512$ color images of three datasets
8	Col_1024	16	Only $1024 \times 1024$ color images of three datasets
9	Col_1728	12	Only $1728 \times 2304$ or $2304 \times 1728$ color images of all datasets
10	Gray_256	7	Only $256 \times 256$ gray images of three datasets
11	Gray_512	5	Only $512 \times 512$ gray images of three datasets
12	Gray_1024	3	Only $1024 \times 1024$ gray images of three datasets
13	Lena	18	Only Lena images with different resolutions for comparing with existing works



From Table 5, we see that the number of images varies from 1 to 61 for each experimental setup. We get one result for each image in each experiment. So, if there are 15 images in a dataset, we get 15 results. For brevity, we only present the best results as the existing studies did the same. Therefore, we compare our best results on each dataset with the best results found by the existing studies.

## Entropy

The entropy value of an image indicates the randomness of pixels that have the same intensity. In other words, entropy is the pixel crowdedness of an image. If the pixels are uniformly distributed, the entropy value becomes high which means it will be difficult for an intruder to perform any statistical attack [7,29,38]. The entropy  $E_2$  is measured by Eq. 9 based on the probability of intensity in any distribution [4,6,11,22,34,35,41,48].

$$E_2(Z) = - \sum_{i=0}^{2^n-1} P(z_i) \times \log_2 P(z_i), \quad (9)$$

where  $n$  is the number of bits used to represent the gray value of a pixel,  $z_i$  represents the height of  $i^{th}$  bin in the histogram  $z$ ,  $P(z_i)$  is the probability of having the  $i^{th}$  gray level in the image.

The ideal value of entropy in image encryption is 8 [7,23,42]. Our experimental results are given in Table 6 and we observe that our method achieves a good entropy value (best results in the 3rd column) which is closer to the ideal one than the existing studies (best results in  $2^{nd}$  column). We achieve the highest entropy score for Ye et al. [45]. We obtain each of the entropies 7.9998842 and 7.9998817 for a couple of datasets. For example, 7.9998842 is ordained for the gray color Peppers image with 512x512 resolution both in the datasets used in [22] and [43].

It is notable that for all datasets, we achieve better scores than the existing ones. Hence, for any kind of statistical attack, our method has better protection ability than existing methods.

## Key space

In image encryption, the strength of the secret key plays an important role [4,11,22,41,44,48]. Generally, the longer the key, the stronger the resistivity against brute force attacks [8]. But the longer the key length, the higher the time complexity. However, the standard size of key space is  $2^{100}$  [7]. In our approach, the minimum key size is 15 characters and the maximum is the size of the input plain image in bits. If we consider the minimum characters, then the key length becomes  $15 \times 8 = 120$  bits, i.e., the size of the key space

**Table 6** Entropy comparison between existing methods and our proposed method. The 1st column represents the list of existing methods. 2nd and 3rd columns represent entropy obtained by the existing methods and our proposed method, respectively. Each row uses the same dataset used in the study mentioned in the 1st column. For each dataset, the comparative best entropy between the existing method and our proposed method are shown with an asterisk (\*) sign

Dataset used in	Existing entropy	Our entropy
Jithin et al. [11]	7.9998	7.9998943*
Nematzadeh et al. [22]	7.9991	7.9998842*
Wang et al. [34]	7.9975	7.9998817*
Chen et al. [4]	7.9974	7.9998927*
Wang et al. [35]	7.9975	7.9998817*
Farah et al. [6]	7.9991	7.9998935*
Wu et al. [40]	7.99947	7.9998873*
Wang et al. [36]	7.9975	7.9998796*
Yadollahi et al. [43]	7.9994	7.9998842*
Ye et al. [45]	7.99924	7.9998949*

becomes  $2^{120}$ . In our method, we generate a key  $G$  large enough to encode each pixel with a different portion of the key generated by the key scrambling technique using Section 3.1 Algorithm 1. For example, if we use a color image of size  $512 \times 512$  as a plain image for encryption in our method, the length of  $G$  need to be  $512 \times 512 \times 3 \times 8$  or 6291456 bits, i.e., the size of the key space is  $2^{6291456}$  which is practically a large space.

## Key sensitivity

Key sensitivity tests the visual change of a cipher image for a slight change in the encryption key keeping its length fixed [11]. Different researchers used this evaluation metric to prove the strength of their methods [4,11,22,34,35,41,44,48]. Let, a plain image be encrypted with two different keys  $K_1 = \text{“BangladeshIsMyHomeLand”}$  and  $K_2 = \text{“BaNgLaDeShIs-MyHoMeLaNd”}$ . The only difference between the two keys is capitalizing the third character and then every second character. The effects of such changes in the cipher and histogram of the Lena image are depicted in Fig. 5. We can see that the two ciphers and corresponding histograms are visually very much different. This illustration strongly endorses that our method is highly sensitive to key variations.

## Variance of histogram

It is a metric to evaluate the uniformity of histograms of different cipher images. This metric determines the strength of confusion and diffusion ability of a method [11]. In this experiment, we generate two cipher images from a plain image using two different keys. Let, a plane image be  $P$ ,

**Table 7** Comparison of Variance of Histogram (VoH) between our proposed and existing methods. The 1st column represents the list of existing methods. 2nd and 3rd columns represent the best VoH score obtained by the existing method and our proposed method, respectively. Each row uses the same dataset used in the study mentioned in the 1st column. For each dataset, the comparative best results between the existing method and our proposed method are shown with an asterisk (\*) sign

Dataset used in	Existing VoH	Our VoH
Jithin et al. [11]	240.16	199.16*
Ye et al. [45]	248	186.42*

the two keys  $K_1$  and  $K_2$ , the generated cipher images  $C_1$ , and  $C_2$  and their corresponding histograms  $H_1 = hist(C_1)$ , and  $H_2 = hist(C_2)$  respectively. We calculate the variance of histogram  $V$  using Eq. 10. The smaller the value of  $V$ , the better the encryption quality [11]. Table 7 compares the variance of histogram obtained by two existing methods with our method. It is clearly seen that for each dataset, our method outperforms the existing methods. We achieve the best score for Ye et al. [45] dataset which depicts that our method has a significant improvement over the existing methods. Hence,

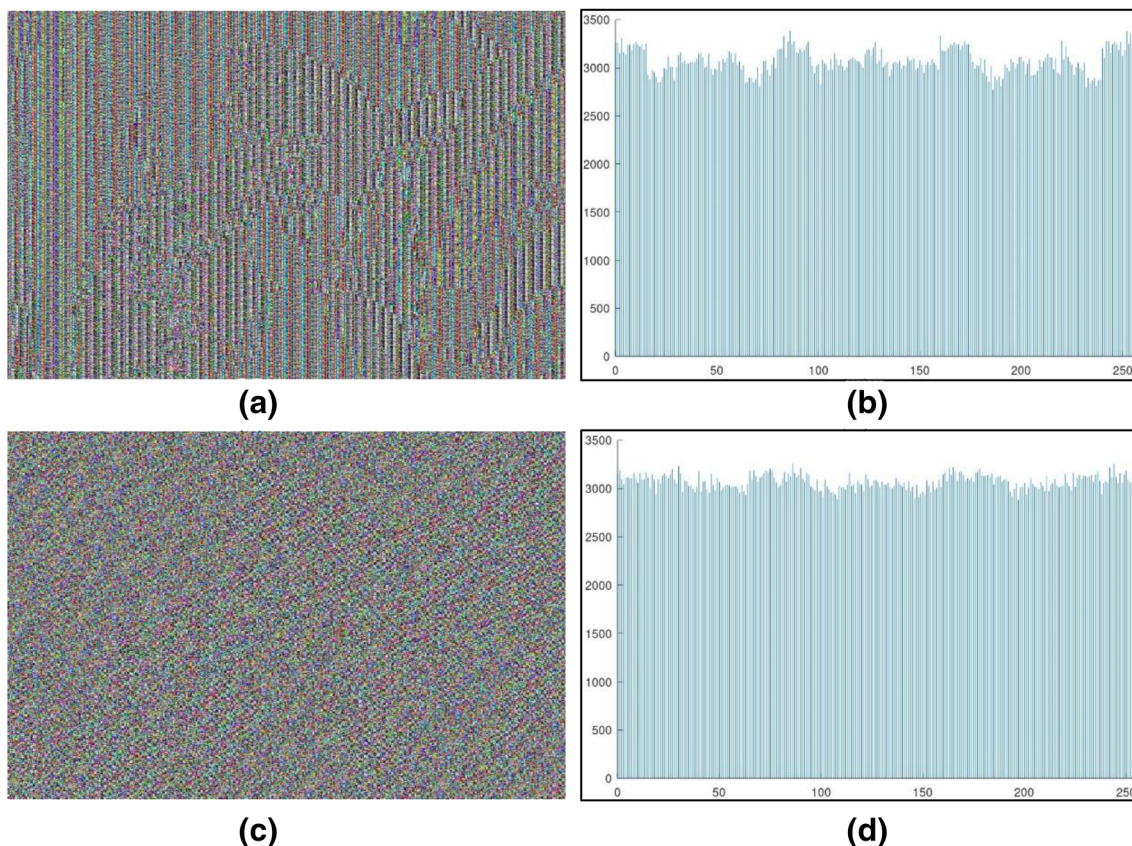
our method shows strong confusion and diffusion ability.

$$V = \left( \frac{1}{M^2} \right) \sum_{i=1}^M \sum_{j=1}^M (H_1(i) - H_2(j))^2, \quad (10)$$

where  $M$  is the number of intensity levels which is usually 256,  $H_1(i)$  is the  $i^{th}$  element of histogram  $H_1$ , and  $H_2(j)$  is the  $j^{th}$  element of histogram  $H_2$ .

### Cipher pixels correlation

In a normal distribution, neighbor pixels are correlated which leaves space for intruders to estimate neighboring pixels by knowing a specific segment. To secure image contents from intruders, image encryption research focuses on distorting such relationships in cipher images [12,26,31]. The correlation may occur with horizontal, vertical, or diagonal neighbors. The correlation value ranges from  $-1$  to  $1$ , where  $1$  indicates a strong positive correlation,  $-1$  means strong negative correlation and  $0$  means no correlation. The ideal value of correlation for a cipher image is expected to be  $0$ , i.e., no correlation among the pixels. Different studies use this evaluation metric to test the ability to protect any kind of



**Fig. 5** Cipher images and their corresponding histograms of Lena image with a resolution of  $512 \times 512$ : **a, b** depicts the cipher & its histogram for the key “BangladeshIsMyHomeLand” and **c, d** depicts the cipher & its histogram for the key “BaNgLaDeShIsMyHoMeLaNd”, respectively

**Table 8** Correlation comparison between existing methods and our proposed method. The 1st column represents the list of existing methods. Columns 2–7 represent the best correlations obtained by the existing method (columns 2–4) and our proposed method (columns 5–7), respec-

tively. Each row uses the same dataset used in the study mentioned in the 1st column. For each dataset, the comparative best results between the existing method and our proposed method are shown with an asterisk (\*) sign

Dataset used in	Existing Correlations			Our correlations		
	H	V	D	H	V	D
Jithin et al. [11]	-0.00116	0.00106	-0.0043	-0.00012*	-0.00012*	0.00012*
Nematzadeh et al. [22]	0.0007	0.0031	0.0007	-0.00012*	-0.00012*	0.00012*
Wang et al. [34]	0.0013	0.0009	0.0012	0.00022*	0.00026*	-0.00031*
Chen et al. [4]	0.0022	0.0013	0.0008	-0.00012*	-0.00012*	0.00012*
Wang et al. [35]	-0.0085	-0.0031	-0.0019	-0.00025*	-0.00019*	-0.00039*
Farah et al. [6]	0.0693	0.0610	-0.0242	0.00016*	0.00015*	-0.00019*
Wu et al. [40]	0.01576	0.00229	-0.0336	0.00012*	-0.00012*	0.00012*
Wang et al. [36]	0.0015	0.0021	-0.0005	-0.00026*	-0.00033*	-0.00019*
Yadollahi et al. [43]	0.0059	0.0047	0.0058	-0.00012*	-0.00012*	0.00012*
Ye et al. [45]	0.03989	0.03448	0.00195	-0.00025*	0.00029*	0.00022*

statistical attack [4,6,11,22,34,35,41,44,48]. The correlation is calculated using Eqs. 11 to 14 [11,22,34].

$$E(x) = \frac{1}{N} \sum_{i=1}^N x_i \tag{11}$$

$$S(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2 \tag{12}$$

$$cov(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))(y_i - E(y)) \tag{13}$$

$$r_{x,y} = \frac{cov(x, y)}{\sqrt{S(x)}\sqrt{S(y)}}, \tag{14}$$

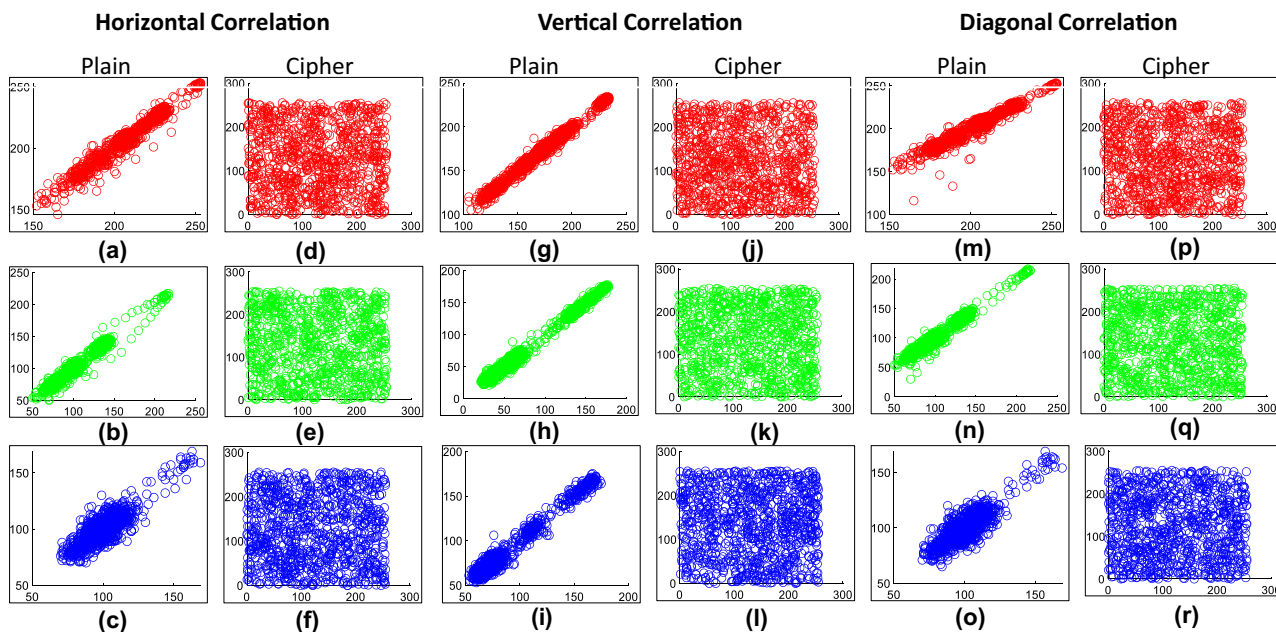
where  $x$  represents the set of pixel intensities in an image,  $x_i$  represents the  $i^{th}$  pixel in the set,  $y_i$  is the horizontal, vertical, or diagonal neighbor of  $x_i$ ,  $N$  is the number of pixels in the

image,  $E(x)$  is the average intensity of the image,  $S(x)$  is the standard deviation of intensities in the image,  $S(y)$  is the standard deviation of neighbors,  $cov(x, y)$  is the variance of  $x$  and  $y$ , and  $r$  is the correlation coefficient.

We present our experimental findings in Table 8. We apply our algorithm on 10 datasets that were used by existing studies listed in Table 3. We see that our method achieves better results than existing methods. We observe that in existing methods, the highest horizontal correlation is 0.0007, vertical 0.0009 and diagonal -0.0005 denoted by H, V and D respectively in the table. We achieve the highest correlations  $\pm 0.00012$ ,  $\pm 0.00012$  and  $\pm 0.00012$ , respectively, for H, V and D for five datasets. The result clearly indicates that our approach makes the system stronger in preventing all kinds of attacks than existing methods. Figure 6 shows the pictorial view of correlations of our approach.

**Table 9** Comparison between existing methods and our proposed method based on NPCR and UACI metrics. The 1st column represents the list of existing methods. Columns 2–5 represent the best NPCR and UACI values obtained by the existing method (Columns 2–3) and our proposed method (Columns 4–5), respectively. Each row uses the same dataset used in the study mentioned in the 1st column. For each dataset, the comparative best results between the existing method and our proposed method are shown with an asterisk (\*) sign

Dataset used in	Original Scores		Our scores	
	NPCR	UACI	NPCR	UACI
Jithin et al. [11]	99.64*	33.42	99.6339	33.4479*
Nematzadeh et al. [22]	99.6742*	33.6392*	99.6193	33.4471
Wang et al. [34]	99.64*	33.41	99.6082	33.4362*
Chen et al. [4]	99.61	33.44	99.6337*	33.4478*
Wang et al. [35]	99.60	33.47*	99.6066*	33.4397
Farah et al. [6]	99.5697	33.41	99.6337*	33.4483*
Wu et al. [40]	99.51	33.36	99.6199*	33.4458*
Wang et al. [36]	99.6459*	33.4633*	99.6045	33.4374
Yadollahi et al. [43]	99.6251*	33.1933	99.6177	33.4471*
Ye et al. [45]	99.6056	33.4173	99.6345*	33.4594*



**Fig. 6** Correlation analysis of Lena plain and cipher images. **a–f** horizontal, **g–l** vertical and **m–r** diagonal correlations. In each group, the first and second columns represent plain and cipher correlations respec-

tively. Again, the first, second and third rows indicate red, green, and blue channel correlations, respectively

### Resisting differential attacks

The intruders always try to decode a cipher and get its original message. One of the techniques they usually apply is to change small portions of the plain image pixels and observe what happens in the cipher image. They repeatedly apply the process to get any pitfall of encryption algorithms. We aim to develop a robust approach to pretend such attacks. Two metrics, e.g., the number of pixel change rates (NPCR) and unified average changing intensity (UACI) are used to check whether the system is strong enough to handle these differential attacks [1,13,33]. Different studies utilize these two metrics for evaluating their methods [4,6,11,22,34,35,41,48]. Let,  $P$  be a plain image. The system generates a cipher  $C_1$  from  $P$ . If a small change on  $P$  produces  $P_1$ , the encryption process generates a new cipher  $C_2$ . NPCR denoted by  $N_p$  and UACI denoted by  $U_a$  scores are obtained from  $C_1$  and  $C_2$  using Eqs. 15 and 17, respectively [1,13,33].

$$N_p = \frac{\sum_{i=1}^h \sum_{j=1}^w G_1(i, j)}{h \times w} \times 100, \tag{15}$$

where  $h$  and  $w$  are image height and width and  $G_1$  is obtained from Eq. 16.

$$G_1(i, j) = \begin{cases} 0, & \text{if } (C_1(i, j) == C_2(i, j)) \text{ for all } i, j \\ 1, & \text{if } (C_1(i, j) \neq C_2(i, j)) \text{ for all } i, j \end{cases}$$

$$\tag{16}$$

where  $C_1(i, j)$  the intensity value of cipher at location  $(i, j)$  and

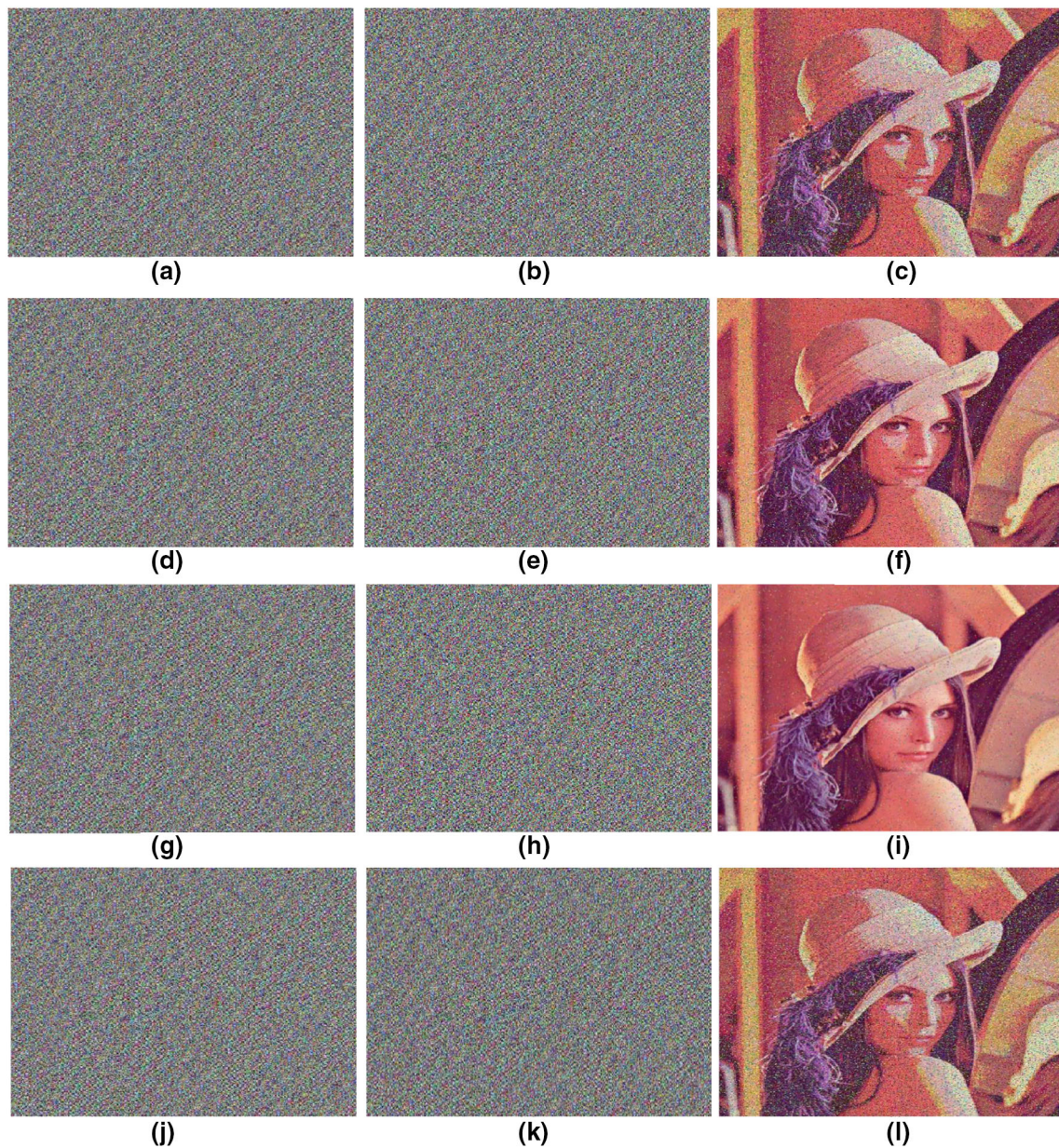
$$U_a = \frac{1}{255 \times h \times w} \left[ \sum_{i=1}^h \sum_{j=1}^w (C_1(i, j) - C_2(i, j)) \right] \times 100, \tag{17}$$

where  $h$  and  $w$  are image height and width, respectively.

The ideal values of NPCR and UACI are 99.61% and 33.46%, respectively [7,11]. Table 9 presents NPCR and UACI scores obtained by our proposed method on reference datasets used by existing studies. Our method has performed well on some datasets. We achieve the highest NPCR 99.6345% and UACI 33.4594% for the dataset used by Ye et al. [45]. We can see that our scores are very close to the ideal score though there are a few existing methods that have closer values than ours. Although our results are not always the best one in comparison to other existing works, it is still promising to defend against differential attacks.

### Peak signal-to-noise ratio (PSNR)

PSNR is an encryption quality measurement technique that measures the amount of wideness of the pixels in an



**Fig. 7** Different noise effects on the cipher image where the first, second and third columns contain Lena cipher image, Lena cipher image with noise, and the decrypted image from noisy cipher image, respectively.

In the second column **b** Gaussian noise with (“zero-mean” & variance 0.01), **e** Poisson noise with (mean 5.5), **h** Salt & Pepper noise with (density 0.05) and **k** Speckle noise with (mean 0 & variance 0.05)

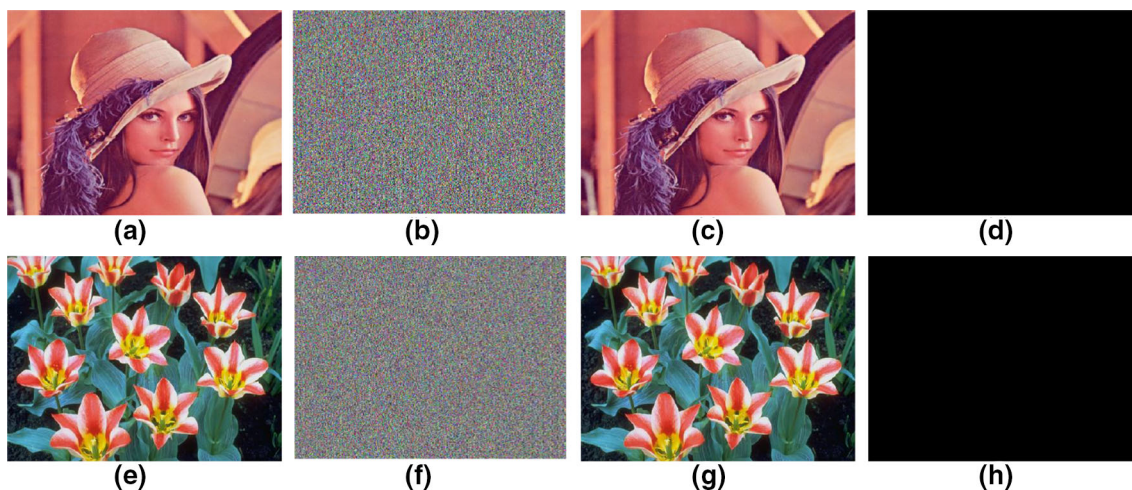
image [19]. It can be calculated from the mean square error value and it is a logarithmic modification of a plain image  $P$  and corresponding cipher image  $C$ . Generally, a large value of PSNR,  $P_s$  indicates a good quality of encryption [11] which is measured using Eqs. 18 to 20 [11,34,48]. Also, a large PSNR value makes it difficult for attackers to digest any message.

$$M_e = \frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w (P(i, j) - C(i, j))^2, \tag{18}$$

where  $h$  and  $w$  are the height and width of the image and  $P(i, j)$  and  $C(i, j)$  represent intensity at location  $(i, j)$  in images  $P$  and  $C$ , respectively, and

$$M_x = \max(\{P(i, j)\} | i = 1, 2, \dots, h \text{ and } j = 1, 2, \dots, w) \tag{19}$$

$$P_s = 10 \log_{10} \frac{M_x}{M_e}, \tag{20}$$



**Fig. 8** Two examples of image reconstruction. For Lena, **a–d** depict plain image, cipher image, reconstructed image, and plain and reconstructed image difference, respectively. Similarly, **e–h** depict the Tulip example

**Table 10** PSNR score comparison between existing methods and our proposed method. The 1st column represents the list of existing methods. 2nd and 3rd columns represent the best PSNR value obtained by the existing method and our proposed method respectively. Each row uses the same dataset used in the study mentioned in the 1<sup>st</sup> column. For each dataset, the comparative best results between the existing method and our proposed method are shown with an asterisk (\*) sign

Dataset used in	Existing PSNR	Our PSNR
Jithin et al. [11]	36.07	88.5642*
Nabarun et al. [21]	74.94	88.5642*
Wang et al. [34]	20.86	87.6532*
Dongming et al. [10]	45.08	88.6496*

where  $M_x$  is maximum value obtained in Eq. 19 and  $M_e$  mean square error obtained in Eq. 18. We present our results in Table 10 which compares our method with some existing works. It reveals that our PSNR score is better than existing works with a remarkable difference and we achieve the highest score for two datasets. We also observe a high PSNR value for the rest of the two datasets which are very close to the highest score. Hence, it alludes that our encryption technique generates a wider range of pixel distributions in cipher images which makes it difficult for the attackers to digest the message.

### Noise attack

During the transmission of an image, the cipher image may sometimes be altered or affected in a small amount due to various reasons. Hence, it becomes difficult to decrypt the noisy cipher because the cipher pixels are changed due to noise. Different researchers use this metric to express the strength of their encryption methods [11,34,41,44]. So, we

apply different noises in the cipher image and test the system whether is it capable to decrypt the cipher perfectly or not. Let, a cipher  $C$  is affected by Gaussian noise  $G_n$  (with mean 0 & variance 0.01) and produces a new cipher  $C_{gn}$ . The receiver receives the cipher  $C_{gn}$  and a key. Then the cipher is decrypted to  $D_{gn}$ . Similarly, we add Poison with mean 5.5, Speckle with mean 0 & variance 0.05, Salt & Peeper noises with default density 0.05 and reconstruct them in return. For all noises, we use the function *imnoise()* with default parameters in Matlab <sup>1</sup>. The results shown in Fig. 7 reveal that our approach can reconstruct the images properly and is capable to handle noise attacks.

### Image reconstruction

An image encryption system is considered to be efficient if its decryption process can reconstruct the encrypted image perfectly, i.e., the original image and reconstructed image are pixel-wise same. Different studies use this metric to evaluate the efficiency of the model [5,18,48]. In our system, we achieve this milestone. The visual result of Lena and Tulip images with resolution  $512 \times 512$  are shown in Fig. 8.

### Computational time complexity

Computational time complexity is a performance evaluation parameter that indicates how fast an algorithm gives results [46]. It means how many internal machine instructions have to be completed by the computer to complete an algorithmic task. The lower the time complexity of the image encryption algorithm, the better the model [2]. Different researchers use this metric to represent the time complex-

<sup>1</sup> <https://www.mathworks.com/help/images/ref/imnoise.html>

**Table 11** Comparison of algorithmic computational time complexity between our proposed method and some existing methods

Method	Time complexity
Wu et al. [41]	Step 1: $\theta(12 \times M \times N)$ , Step 2: $\theta(3 \times M \times N)$ , Step 3: $\theta(12 \times M \times N)$ , Step 4: $\theta(12 \times M \times N)$ , Step 5: $\theta(24 \times M \times N)$ , Step 6: $\theta(3 \times M \times N)$ , Step 7: $\theta(24 \times M \times N)$
Wang et al. [34]	Step 1: $O(9 \times M \times N)$ , Step 2: $O(8 \times M \times N)$ , Step 3: $O(4 \times M \times N)$ , Step 4: $O(4 \times M \times N)$
Jiahui et al. [39]	Step 1: $O(M \times N)$ , Step 2: $O(M \times N)$ , Step 3: $O(M \times N)$
Lidong et al. [16]	Step 1: $O(M + N)$ , Step 2: $O(8 \times M \times N)$ , Step 3: $O(4 \times M \times N)$
Our proposed method	Step 1 Key scrambling: $O(L \times \log(L))$ , Step 2 Diffusion: $O(M \times N)$

ity of their method [16,34,39,41]. Our proposed algorithm has two steps. In the first step, it generates a large key from a given short key using the key scrambling technique which has the time complexity of  $O(L \times \log(L))$ , where  $L$  is the length of the given short key. The second step is the encryption algorithm which takes  $O(M \times N)$  where  $M$  and  $N$  are the height and width of the plain image, respectively. The second step is more time consuming than the first step. Hence, our final time complexity is  $O(M \times N)$ . We present the time complexity of our method and compare it with some recent DNA-based image encryption techniques in Table 11. From the table, we see that our method is computationally very faster than other methods. As our method does not involve any confusion or pixel permutation steps, there exist only two steps that involve very fewer operations compared to others. Hence, our method is very well suited to perform faster image encryption.

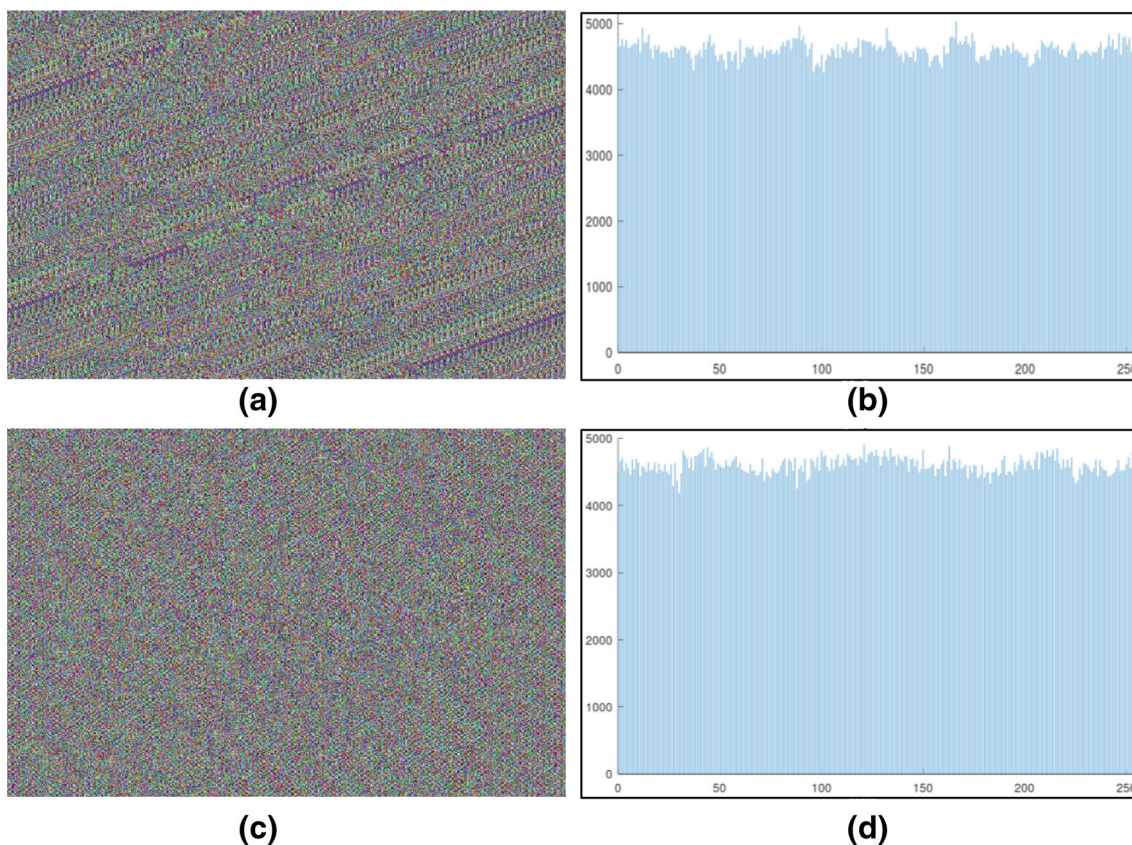
### Key of variable length

Different research groups focused on DNA-based image encryption techniques and used a fixed-length key using a chaotic map [2,14,15,17,20,27,47]. Some of them used SHA-256 [6,41,48], SHA-2 [32] and MD5 [33] for image encryption. On the contrary, our method can work with different key lengths. It can generate very different cipher images for two keys of different lengths. For instance, given two keys  $K_1 = \text{“BangladeshIsMyHomeLand”}$  and  $K_2 = \text{“BangladeshIsMyHomeLandILoveMyCountryVery-Much”}$ , our method generates the ciphers depicted in Fig. 9. The figure depicts huge textural variations on cipher images.

### Robustness in publicly available datasets

We use the datasets of existing studies for justifying the strength of our proposed method. We also justify the robustness of our system by applying our method on three standard and comparatively large datasets. We divide the datasets into 13 different subsets (Table 4) and apply our proposed method on each subset and then list the results of different parameters (key of variable length, variance of histogram, entropy, cipher pixels correlation, resisting differential attacks and PSNR) in Table 12. Here, we see that our method achieves satisfactory results on those subsets. In most cases, we find the best results for the image subset that contains only Lena images of different resolutions. The results are very much aligned with datasets from existing studies. Hence, we can conclude that our system is robust to challenging datasets.

Therefore, we conduct comprehensive experiments in terms of different performance evaluation metrics from different viewpoints where we use 12 datasets from existing studies for comparing the overall system performance and 3 standard datasets with 13 setups for robustness testing. We



**Fig. 9** Cipher images and their corresponding histograms of Tulip image. Top row: result of encryption with the key “*BangladeshIsMyHomeLand*”. Bottom row: result of encryption with the key “*BangladeshIsMyHomeLandILoveMyCountryVeryMuch*”

**Table 12** Different results from datasets of robustness testing: column 1 setup name according to column 2 Variance of Histogram, column 3 Entropy, column (4–6) Pixel Correlations, column 7 NPCR, column 8 UACI and the last column represent the PSNR scores. (\*) sign indicates the comparative best result

Setup Name	VoH	Entropy	Correlations			NPCR	UACI	PSNR
			H	V	D			
ImgEnhc	215.1	7.99988	0.00040	0.00022	0.00016	99.55	33.45	86.45
USCSIPI	207.16	7.99985	0.00033	0.00019	0.00026	99.56	33.45	84.02
InfHide	195.59	7.99986	0.00022	0.00026	-0.00031	99.60*	33.45*	87.65
Col	196.12	7.99987	-0.00040	-0.00038	0.00028	99.59	33.44	88.55
Gray	199.14	7.99986	-0.00025	-0.00019	-0.00039	99.55	33.43	84.12
Col_256	186.39*	7.99986	-0.00022	-0.00019	0.00013	99.58	33.41	84.90
Col_512	215.66	7.99987	0.00016	0.00013	-0.00019	99.56	33.42	86.12
Col_1024	202.79	7.99988	0.00012*	-0.00012*	0.00012*	99.58	33.41	87.12
Col_1728	215.62	7.99989*	-0.00026	-0.00035	-0.00019	99.59	33.41	88.55
Gray_256	213.53	7.99985	0.00019	0.00021	0.00033	99.41	33.40	84.25
Gray_512	206.19	7.99986	-0.00025	0.00039	0.00022	99.58	33.44	84.36
Gray_1024	188.15	7.99987	0.00017	-0.00033	-0.00019	99.59	33.43	85.54
Lena	187.12	7.99989*	-0.00012*	-0.00012*	0.00012*	99.59	33.43	88.56*



find that our novel key scrambling-based image encryption technique achieves the best entropy score in comparison with existing methods. We also achieve the best scores for key space, pixel correlations, variance of histogram, PSNR and time complexity. The scores of key sensitivity and differential attack show very promising results. In addition, we evaluate our system performance for image reconstruction and various noise attacks. Our reconstruction scores reveal that the proposed method is a novel method for image cryptography. Hence, we can conclude that our method is highly effective, suitable and has enough security strength for image encryption.

## Conclusion

For secured data transmission, image encryption is essential in the digital world. In this research work, a DNA operation using key scrambling based image encryption system is proposed. In our approach, we do not involve any confusion step. We conduct experiments on 15 datasets and evaluate the proposed approach using different performance metrics. We compare our results with cutting-edge techniques. We achieve the best scores which are very close to ideal milestones for six evaluation metrics. Our system is able to reconstruct the original image from the cipher image 100% accurately. We extend our experiments by applying 13 different data subsets from 3 publicly available datasets which clearly reveal that our system is equally performing. Our novel key scrambling technique is highly efficient for diffusing pixels. This efficiency is confirmed with achieved correlation and entropy scores. In addition, most of the recent works are supposed to send one or more tables for decryption, which is obviously burdensome for traffic. We reduce that burden in our approach without compromising security. Hence, it reduces transmission time and space. Moreover, our method is robust in terms of the number of channels and variable key length processing. Therefore, we can conclude that our method is highly efficient for image encryption and our key scrambling technique is new and innovative.

The advantages of our method are faster encryption time, less bandwidth to transmit the cipher image, less time to transmit and decrypt the image, a very high level of encryption quality on different evaluation metrics, and the system is robust and dynamic.

Although we achieve the best scores in different parameters, we aim to improve the competitive scores towards the ideal scores in future work. Moreover, we will test our system for hyperspectral image datasets.

**Acknowledgements** This work is supported by ICT Division, Ministry of Posts, Telecommunications and Information Technology, Government of Bangladesh through the Doctoral research fellowship funding

for Mr. Machbah Uddin in the 1st round of the fiscal year of 2020–2021. The authors are grateful to the funding authority.

## Declarations

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

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Anwar S, Meghana S (2019) A pixel permutation based image encryption technique using chaotic map. *Multimed Tools Appl* 78(19):27569–27590
- Batool SI, Waseem HM (2019) A novel image encryption scheme based on arnold scrambling and lucas series. *Multimed Tools Appl* 78(19):27611–27637
- Belazi A, Talha M, Kharbech S, Xiang W (2019) Novel medical image encryption scheme based on chaos and dna encoding. *IEEE Access* 7:36667–36681
- Chen C, Sun K, He S (2020) An improved image encryption algorithm with finite computing precision. *Signal Process* 168:107340
- Chen J, Chen L, Zhou Y (2020) Cryptanalysis of a dna-based image encryption scheme. *Inf Sci* 520:130–141
- Farah MB, Guesmi R, Kachouri A, Samet M (2020) A novel chaos based optical image encryption using fractional Fourier transform and dna sequence operation. *Opt Laser Technol* 121:105777
- Ghadirli HM, Nodehi A, Enayatifar R (2019) An overview of encryption algorithms in color images. *Signal Process* 15:18–31
- Han C (2019) An image encryption algorithm based on modified logistic chaotic map. *Optik* 181:779–785
- Hua Z, Yi S, Zhou Y (2018) Medical image encryption using high-speed scrambling and pixel adaptive diffusion. *Signal Process* 144:134–144
- Huo D, Zhu X, Dai G, Yang H, Zhou X, Feng M (2020) Novel image compression-encryption hybrid scheme based on dna encoding and compressive sensing. *Appl Phys B* 126(3):1–9
- Jithin K, Sankar S (2020) Colour image encryption algorithm combining, arnold map, dna sequence operation, and a mandelbrot set. *J Inf Secur Appl* 50:102428
- Khan JS, Ahmad J (2019) Chaos based efficient selective image encryption. *Multidimens Syst Signal Process* 30(2):943–961
- Kumar S, Pradhan C (2020) Color image encryption technique using 4d logistic map. In: *Progress in computing, analytics and networking*, Springer, New York, pp 75–82
- Li M, Xu M, Luo J, Fan H (2019) Cryptanalysis of an image encryption using 2d henon-sine map and dna approach. *IEEE Access* 7:63336–63345

15. Liu H, Wen F, Kadir A (2019) Construction of a new 2d chebyshev-sine map and its application to color image encryption. *Multimed Tools Appl* 78(12):15997–16010
16. Liu L, Wang D, Lei Y (2020) An image encryption scheme based on hyper chaotic system and dna with fixed secret keys. *IEEE Access* 8:46400–46416
17. Luo Y, Yu J, Lai W, Liu L (2019) A novel chaotic image encryption algorithm based on improved baker map and logistic map. *Multimed Tools Appl* 78(15):22023–22043
18. Mansouri A, Wang X (2020) A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. *Inf Sci* 520:46–62
19. Mathur A, Khunteta A, Verma AK (2020) An efficient approach for image encryption using zigzag, Arnold transformation and double random-phase encoding in fractional Fourier transform domain. In: *Microservices in big data analytics*, Springer, New York, pp 49–58
20. Mondal B, Singh S, Kumar P (2019) A secure image encryption scheme based on cellular automata and chaotic skew tent map. *J Inf Secur Appl* 45:117–130
21. Nandy N, Banerjee D, Pradhan C (2018) Color image encryption using dna based cryptography. *Int J Inf Technol* pp 1–8
22. Nematzadeh H, Enayatifar R, Yadollahi M, Lee M, Jeong G (2020) Binary search tree image encryption with dna. *Optik* 202:163505
23. Prajwalasimha S, Kavya S, Hema A, Anusha H, et al (2019) Modified gingerbreadman chaotic substitution and transformation based image encryption. In: *International conference on computational vision and bio inspired computing*, pp 606–614. Springer
24. Premaratne P, Premaratne M (2012) Key-based scrambling for secure image communication. In: *International conference on intelligent computing*, pp 259–263, Springer
25. Ramasamy P, Ranganathan V, Kadry S, Damaševičius R, Blažauskas T (2019) An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic-tent map. *Entropy* 21(7):656
26. Sinha RK, Agrawal I, Jain K, Gupta A, Sahu S (2020) Image encryption using modified rubiks cube algorithm. In: *Advances in computational intelligence*, pp 69–78. Springer
27. Sneha P, Sankar S, Kumar AS (2020) A chaotic colour image encryption scheme combining walsh-hadamard transform and arnold-tent maps. *J Ambient Intell Hum Comput* 11(3):1289–1308
28. Sokouti M, Sokouti B (2018) A prisma-compliant systematic review and analysis on color image encryption using dna properties. *Comput Sci Rev* 29:14–20
29. Sparavigna AC (2019) Entropy in image analysis
30. Sun S (2018) A novel hyperchaotic image encryption scheme based on dna encoding, pixel-level scrambling and bit-level scrambling. *IEEE Photon J* 10(2):1–14
31. Tewani R, Garg Y, Bagga JS, Singh A, Bhalsodia R (2020) Image encryption using permutation–diffusion approach. In: *Advances in data sciences, security and applications*, pp 363–373, Springer
32. Ur Rehman A, Liao X (2019) A novel robust dual diffusion/confusion encryption technique for color image based on chaos, dna and sha-2. *Multimed Tools Appl* 78(2):2105–2133
33. Ur Rehman A, Xiao D, Kulsoom A, Hashmi MA, Abbas SA (2019) Block mode image encryption technique using two-fold operations based on chaos, md5 and dna rules. *Multimed Tool Appl* 78(7):9355–9382
34. Wang X, Hou Y, Wang S, Li R (2018) A new image encryption algorithm based on cml and dna sequence. *IEEE Access* 6:62272–62285
35. Wang X, Liu C (2017) A novel and effective image encryption algorithm based on chaos and dna encoding. *Multimed Tool Appl* 76(5):6229–6245
36. Wang X, Liu L (2020) Image encryption based on hash table scrambling and dna substitution. *IEEE Access* 8:68533–68547
37. Wang XY, Li P, Zhang YQ, Liu LY, Zhang H, Wang X (2018) A novel color image encryption scheme using dna permutation based on the lorenz system. *Multimed Tool Appl* 77(5):6243–6265
38. Wen H, Yu S, Lü J (2019) Breaking an image encryption algorithm based on dna encoding and spatiotemporal chaos. *Entropy* 21(3):246
39. Wu J, Liao X, Yang B (2018) Image encryption using 2d henon-sine map and dna approach. *Signal Process* 153:11–23
40. Wu TY, Fan X, Wang KH, Lai CF, Xiong N, Wu JMT (2019) A dna computation-based image encryption scheme for cloud cctv systems. *IEEE Access* 7:181434–181443
41. Wu X, Wang K, Wang X, Kan H, Kurths J (2018) Color image dna encryption using nca map-based cml and one-time keys. *Signal Process* 148:272–287
42. Xie Y, Yu J, Guo S, Ding Q, Wang E (2019) Image encryption scheme with compressed sensing based on new three-dimensional chaotic system. *Entropy* 21(9):819
43. Yadollahi M, Enayatifar R, Nematzadeh H, Lee M, Choi JY (2020) A novel image security technique based on nucleic acid concepts. *J Inf Secur Appl* 53:102505
44. Ye G, Pan C, Dong Y, Shi Y, Huang X (2020) Image encryption and hiding algorithm based on compressive sensing and random numbers insertion. *Signal Process*, p 107563
45. Ye G, Pan C, Huang X, Zhao Z, He J (2018) A chaotic image encryption algorithm based on information entropy. *Int J Bifurcat Chaos* 28(01):1850010
46. Zefreh EZ (2020) An image encryption scheme based on a hybrid model of dna computing, chaotic systems and hash functions. *Multimed Tool Appl* 79(33):24993–25022
47. Zhang YQ, Hao JL, Wang XY (2020) An efficient image encryption scheme based on s-boxes and fractional-order differential logistic map. *IEEE Access* 8:54175–54188
48. Zhou M, Wang C (2020) A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. *Signal Process* 171:107484

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.