Multimedia Tools and Applications (2020) 79:17497–17520
https://doi.org/10.1007/s11042-019-08352-z

A secure image encryption scheme based on genetic operations and a new hybrid pseudo random number generator

Bhaskar Mondal[1] · Tarni Mandal[2]

Received: 18 January 2019 / Revised: 14 August 2019 / Accepted: 9 October 2019 /
Published online: 20 February 2020

Abstract
This paper presents an encryption scheme based on genetic operations and a new hybrid pseudo random number generator (HPRNG). The new HPRNG is designed based on linear feedback shift register (LFSR), chaotic asymmetric tent map and chaotic logistic map. The scheme uses XOR and genetic operations (mutation, and multipoint crossover) to encrypt the image blocks. The first block of the plain image is encrypted with the help of a pseudo-random bit sequence generated by the HPRNG. The subsequent blocks are based on the previous cipher block and the XOR operator. The scheme can be extended to encrypt color images and text as well. The cipher images produced have very low correlation with their corresponding plain images and have high values of entropy, making it unpredictable and difficult to detect redundancies in the image pixel values. More over the scheme is compared with some existing schemes and found that the proposed scheme is comparatively secure and efficient.

Keywords Image encryption · Chaotic map · LFSR · Mutation · Multipoint crossover

1 Introduction

Information security has become the part and parcel of today's world. Images are the most common form of multimedia on the internet. Security has become one of the most important issues during the transmission of information over the network. The main aim of cryptography is to convert the information to be kept confidential into an unrecognizable content by encryption so that only the authorized persons can have the ability to recover the original

✉ Bhaskar Mondal
bhaskarmondal.cs@gmail.com

Tarni Mandal
tmandal.math@nitjsr.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Patna, Patna, India

[2] Department of Mathematics, National Institute of Technology Jamshedpur, Jamshedpur, India

data without losing anything important. An eavesdropper finds it impossible to decipher the encrypted message or image. The cipher image obtained after encryption has no relevance to the original image without decryption. Images have some special properties that makes them different from texts. These properties like high content, redundancy and high correlation among the adjacent pixels make the encryption process worth of a concern. An efficient image encryption algorithm should generate a random cipher image at a good speed [11].

In [13], the authors proposed a scheme based on two chaotic maps and XOR operation. The encryption technique includes two phases; one permutation and one substitution stage. In permutation circular shifts are used, each row of plaintext image is shifted circularly based on a number in a random matrix K_1. So, the number of rotations are different for each rows. Then, the resulted image is shifted circularly column wise. For vertical rotation, every column is shifted circularly using numbers in a random matrix K_2. The resultant matrix is XORed with a PRNS to obtain the encrypted image.

In [7], the authors has demonstrated the weakness of an chaotic map based color image encryption algorithm by successfully chosen-plaintext attack and chosen-ciphertext attack. In [16] a dynamic s-box based image encryption scheme is presented but performance and security of s-boxes for stream cipher should be compared with other schemes. In [5] the authors uses a Chebyshev chaotic map for generating a PRNS, based on the PRNS the bits are permuted first. It stores the 8 bit plains separately (assuming 8 bit gray level images), which results to handle a matrix of size 8times than the original image. For permutation, this scheme needs to sort the PRNS, which adds additional computational overhead of sorting the long chaotic sequence. In the second part the scheme permutes the bit plain again based on an Arnold's cat map In the results the authors did not mention the number of rounds needed to achieve the results. As the scheme permutes the bit plains twice, it results a diffusion effect on encrypted image. But as permutation only ciphers are weaker to the cryptanalysis, the overall scheme becomes a high complexity and low in security scheme [10].

In [6], the authors have proved that any permutation-only encryption , of any structure, the plain image can be recovered completely by chosen-plaintext attack. They have success fully brocken the Fu's scheme and Rahman's Scheme. They need $n = \lceil(\log_L(MN))\rceil \times (O(n.MN))$ number of plain images for a successful attack. Where M is number of rows and M is number of columns of the image.

In [3], the authors proposed an encryption scheme using chaotic map and genetic operations for Wireless Sensor Networks (WSN). The initial parameters are communicated using a secure channel or a key exchange protocol, which uses ECC over prime field for key establishment. For generating pseudo-random sequences, it uses N-logistic tent map. Genetic operations are used for encrypting the pixel values. However, it is found that genetic operations do not generate a quality encryption effect. Since this algorithm uses blocks of plaintext, it needs padding if the size of plaintext is less than the size of the predefined blocks. The whole encryption process is repeated for each block of the plaintext. Each of these blocks require 256 bit pseudo random bit sequence, which is independent of the sequence generated for the previous block. So, for a sufficiently large image as 512 × 512, the generation of sequence takes considerable amount of time. To achieve a fair encryption effect, the scheme takes n numbers of iterations, where n is the numbers of bytes in the block to be encrypted. The crossover operation is done repeatedly so that each byte in mutated plaintext performs the operation at least twice. Hence a higher value of n leads to exponentially high computational overhead.

In [2] the authors have presented an encryption scheme, which has four phases. This scheme starts with a diffusion phase, based on bitwise XOR operation and a new chaotic map followed by a substitution phase based on S-boxes. Further, a diffusion phase is added based on chaotic logistic map followed by a block permutation phase accomplished by a permutation MAP function. This algorithm is like using two chaos based algorithms in serial, which results a high computational overhead.

This paper presents a novel encryption scheme based on a HPRNG and genetic operations. The new HPRNG id designed based on Linear Feedback Shift Register(LFSR) and chaotic asymmetric tent map is used to generate the initial vector for the LFSR. The generated sequence is used to encrypt the image blocks by operations of XOR and genetic operations (mutation, and multipoint crossover). The first block of the plain image is encrypted with help of a pseudorandom bit sequence generated by the HPRNG. Therefore, the proposed scheme generates pseudorandom bit sequences of length 256 for one time, which reduces the computational overhead of generating a huge amount of PRNS generation. The subsequent blocks are based on the previous cipher block and the XOR operator followed by mutation and multipoint crossover operation, which need a low amount of computational cost. Hence the overall process remains a lightweight scheme and useable for WSNs. The scheme can be extended to color images and text as well. The cipher image produced has a very low correlation with the plain image and has high values of entropy, making it unpredictable and difficult to detect redundancies in the image pixel values. Further, the test results are compared with AES [12], Fu et al.'s Scheme [5], Li et al.'s Scheme [8] and Zhou et al.'s Scheme [17]. In most of the cases the proposed scheme demonstrate better performance than the existing.

In the next section (Section 2) the Preliminaries are discussed. In Section 3 the image encryption algorithm is presented followed by the experimental results and security analysis in Section 4. Finally the summary is given in Section 5.

2 Preliminaries

2.1 Chaotic logistic map

Logistic map is one of the simplest chaotic maps, described by (1) [4].

$$\begin{aligned} x_{k+1} &= f(x) = \mu x_k(1-x_k) \\ &\mu \in (0,4), x_k \in (0,1) \end{aligned} \tag{1}$$

when μ belongs in range $(3.5699456, 4)$, the map demonstrate chaotic behavior. The logistical map provides valuable characteristics like simple structure, extreme sensitivity to initial conditions.

For the logistic map, we have an equilibrium point at $x_0 = 1 - 1/\mu$. The derivative of f at that point is:

$$\begin{aligned} & f'(x_0) = \mu(1-2x_0) \\ & = \mu(1-2(1-\tfrac{1}{\mu})) \\ & = -\mu + 2 \end{aligned} \tag{2}$$

Hence, $|f'(x_0)| < 1 \; if \; |2-\mu| < 1$. Thus this equilibrium point is stable if $1 < \mu < 3$. It is unstable for $\mu > 3$, and for $\mu < 1$. So the "behavior" of the function changes at $\mu = 1$ and $\mu = 3$ which are called bifurcation points. The phase diagram of logistic map with initial condition $\mu = 2.0$ and $x_0 = 0.5$ is shown in Fig. 1.

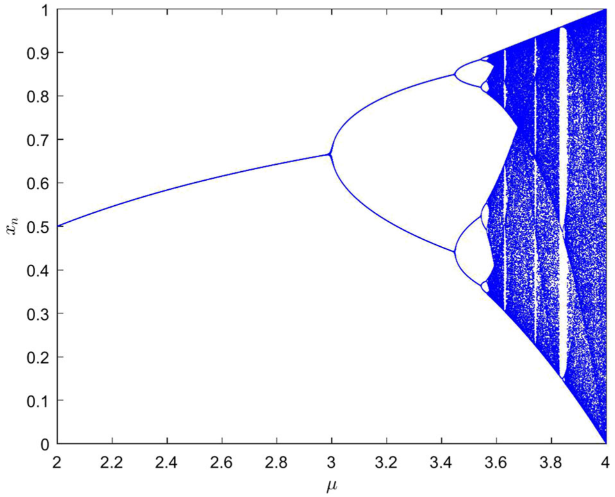


Fig. 1 The phase diagram of Logistic map with initial condition $\mu = 2.0$ and $x_0 = 0.5$

The Lyapunov exponent of Logistic map can be described as (3). It is plotted in Fig. 2 with initial condition $x_0 = 0.5$ and control parameter $\mu = 3.99997$. The calculated value of Lyapunov exponent is 0.690193 [9].

$$\lambda(r; x_0) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \ln |r(1 - 2x_i)| \tag{3}$$

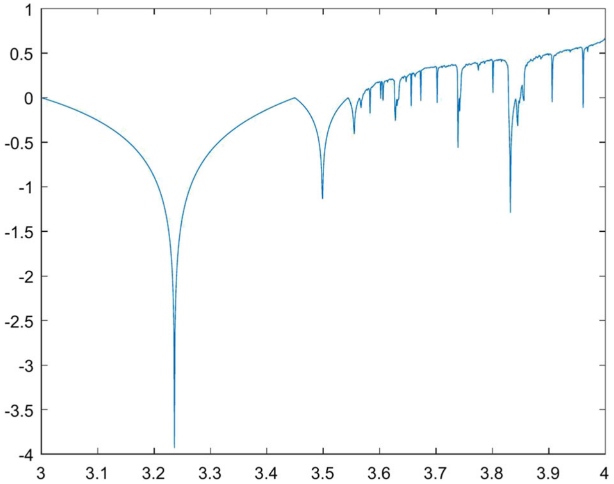


Fig. 2 The Lyapunov exponent of Logistic map

2.2 Linear feedback shift register

A linear feedback shift register (LFSR) is a shift register whose output acts as a feedback and as input for the next output. The inputs to the register depend linearly on the previous states as well. Exclusive-or is the most frequent linear function employed in LFSR. The initial state of the LFSR is termed as the seed. The output of the register is deterministic and completely depends on the previous states. A LFSR can generate a very long cycle with a random appearing sequence of bits, provided the feedback function and the tap positions are choosen well. In Fig. 3 A 32-bit LFSR is given.

The output obtained from the shift register occurs in runs of zeroes and ones. As an example, the output bits 00111001 has four runs of lengths 2, 3, 2, 1, specified in the order of occurence. There occurs 2^{n-1} runs in a complete period of a maximal LFSR, like a LFSR of 8-bit seed has 128 runs. Out of these runs, 50% are of length 1 bit, 25% are of length 2 bits, and up to a single run of length $n - 1$ bits of all 0's, and a single run of length $n - 1$ bits of all 1's [15].

The initial states of LFSR and the tap positions are choosen such that they generate approximately the equal numbers of ones and zeros with a shorter sequence of consecutive same bits. For the LFSR to be of maximal length, the number of taps should be even and the set of taps taken altogether must be relatively prime.

2.3 Genetic operations

2.3.1 Mutation

Genetic Mutation is the operation that introduces diversity from one generation to the next generation. The basis of genetic mutation lies in the biological mutation. It changes one or more genes from the current generation to obtain the next generation genes. The genes may even completely change as a result of mutation operation. This makes genetic algorithms (GA) achieve better genes by going through mutation process. Example of two random cromosom A and B are given below:

- Cromosom A: 1 0 0 1 0 1 1 0
- Cromosom A: 1 0 1 0 1 1 0 0

An cromosom may be encoded useing binary encoding, permutation encoding, value encoding, or tree encoding; Binary encoding is the mosr populer among them in which cromosomes are encoded in 0 and 1 binary string. In this data is represented in binary strings and then represented as cromosoms.

The main purpose of mutation in GAs is to preserve and introduce diversity. Mutation can result into different transformations in the bit sequence depending on its various types. It can be bit flip mutation, random setting, scramble or inversion as summerised in Table 1. The author has proposed to use inversion mutation to invert the bit sequence. The bits in

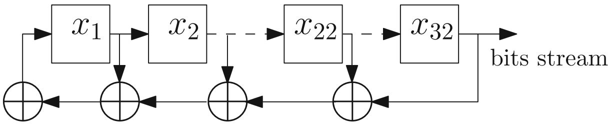


Fig. 3 A 32-bit LFSR

Table 1 Various types of mutations

Mutation type	Before mutation	After mutation
Inversion	10010110	10101010
Swap	10010110	10110100
Scramble	10010110	10011001
Flip	10010110	10110110
Insert	10110110	10110111

each of the blocks are flipped left to right thereby significantly increasing the robustness and security of encryption as shown below example [3].

$$10010110 \rightarrow 01101001$$

2.3.2 Crossover

Crossover operations are generally performed over a single point, two points or uniform points as presented in Table 2. A single crossover point on both input parents bit sequences is selected in the case of single point crossover. All the bits following that crossover point in both the sequences is swapped between the two parent sequences to obtain the children sequences [3].

Multi-point crossover is performed by selecting two points on the parent sequences and swapping the bits between the two points in both the parent sequences to obtain two child sequences. In the proposed scheme, the two points are choosen on the basis of the number of zeros and ones in the bytes of generated random number. An example is shown in Fig. 4.

Mutation and crossover are used in genetic algorithms to generate diversity and produce newer genes from the existing genes.

3 Proposed scheme

3.1 Design of HPRNG

The hybrid PRNG is designed based on a 128 bit LFSR and chaotic maps. LFSRs are good option for PRNG, but there are several successful attacks on LFSR based PRNG, which reviles the secret key. To overcome these problems and generate high-quality random sequence an extra bit is XORed with the feedback of LSFR in every clock cycle. The extra bit is generated by an chaotic logistic map as presented in Fig. 5. Initial 128 bits for the LFSR circuit is called as initial vector (IV). To generate PRN using this hybrid PRNG, it needs two sets of keys, the initial parameters for logistic map and IV for for LFSR. The two

Table 2 Various ways of cross over

Crossover	Parent A	Parent B	Offspring
Single Point	00110100	00100000	00110000
Two Points	00110100	00100000	00100000
Uniform Points	00110100	00100000	01000000

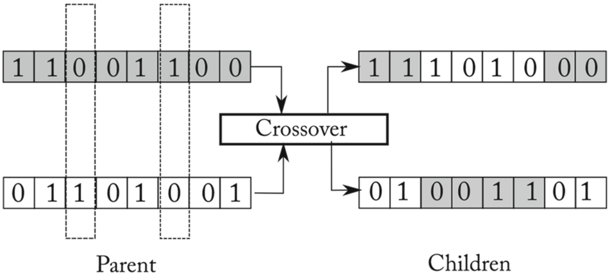


Fig. 4 Two points for multipoint crossover 3^{rd} bit and 6^{th} bit

sets of key together represented as $K = \{x_0, \mu, IV\}$. The tap positions are decided based on the polynomial $b^{32} + b^{22} + b^2 + b^1 + 1$. For the convenience of representation the above 32 bit implementation is presented in Fig. 5. However, for the encryption scheme presented here uses a 128 bit LFSR with feedback polynomial $b^{128} + b^{127} + b^{126} + b^{121} + 1$ for the HPRNG. The 128 bit LFSR has a key-space of $2^{127} - 1$. The skew tent map is an optional to the HPRNG and it has no effect on the efficiency of the HPRNG. It is used as initial vector generator for the LFSR.

3.2 Design of the encryption scheme

The pixel values in the plain image is converted into binary bits and grouped into blocks of 256 bits each. These blocks are encrypted in multiple rounds. In each round, different operations are performed on each block of the image, consisting of 32 pixels. Each block requires a 512 bit sequence K_i for determining on which parts of the block, the specific operations will be performed. This K_i also determines K_{i+1}, which is used for the encryption of the next block in the image. The whole process is represented in Fig. 6

For each round, the encryption process occurs in following phases:

Initially an input image I is taken and divide I in $n : n = 0, 1, 2 \ldots, n$ blocks of 256 bit each. Then generate 512 bit random sequence S_r using the HPRNG. The S_r is divided in in two parts, namely L_0 and R_0.

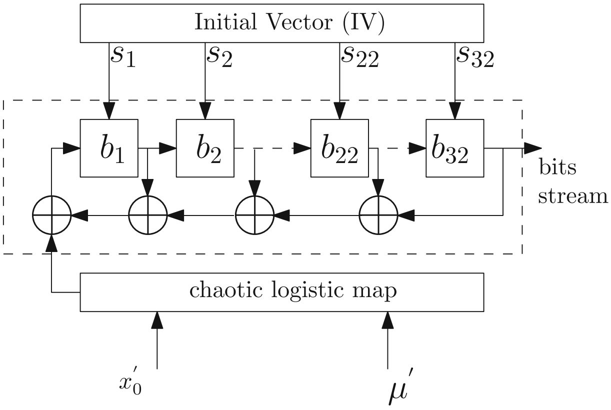


Fig. 5 A 32 bit HPRNG

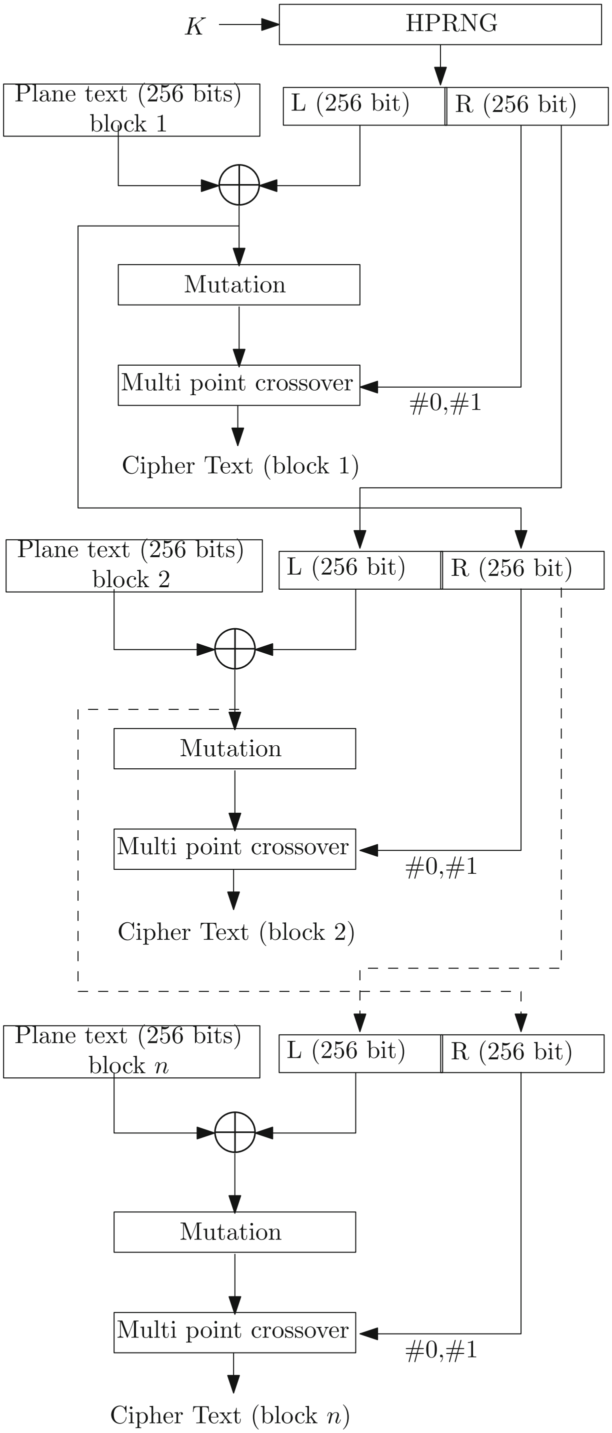


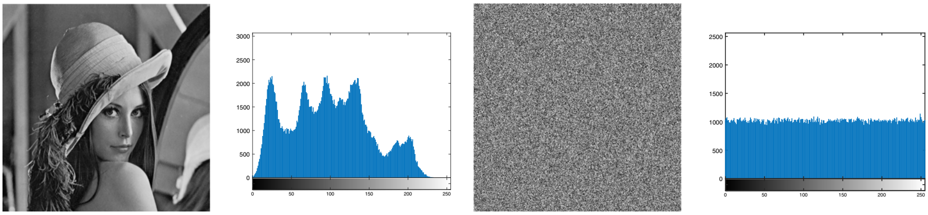
Fig. 6 The proposed encryption algorithm

The n_i block (256 bits) of the plain image are XORed with the 256 bits of left half (L_i) of the generated sequence. The output obtained (R_{i+1}) serves the dual purpose in the encryption process.

Algorithm 1 The encryption process.

```
procedure ENCRYPTION
    $I = readplainImage('plain\_image')$
    Divide $I$ in $n_i : i = 0, 1, 2 \ldots, n$ blocks of 256 bit each.
    $S_r = hprng(512)$
    $L_0 = S_0, S_1, \ldots S_{255}$
    $R_0 = S_{256}, S_{257}, \ldots S_{512}$
    for $i = 1$ to $n$ do
        $R_{i+1} = n_0 \oplus L_i$
        $R_{mi+1} = mutation(R_{i+1})$
        $C_i = multiCrossOver(R_{mi+1}, R_i)$
        $L_{i+1} = R_i$
    end for
end procedure
procedure MUTATION($R_{i+1}$)
    $i$ = 1s in $R_{i+1}$)
    $j$ = 0s in $R_{i+1}$)
    if $i\%2 == 0$ then
        $R_{mi+1}$ = Left Circular shift $R_{i+1}$ for $j$ times
    else
        $R_{mi+1}$ = Right Circular shift $R_{i+1}$ for $j$ times
    end if
end procedure
procedure MULTICROSSOVER($R_{mi+1}, R_i$)
    $R_{mi+1} = B_1 B_2 \ldots B_{32}$
    for $k = 1$ to 32 do
        $i$ = 1s in $B_k$
        $j$ = 0s in $B_k$
        $p_1 = B_{k0} \ldots B_{ki}$
        $p_2 = B_{k(i+1)} \ldots B_{kj}$
        $p_3 = B_{k(j+1)} \ldots B_{k7}$
        Devide $R_i$ in bytes $B_k$ where $k = 1, 2, \ldots 256/8$
        $i$ = 1s in $R_i$
        $j$ = 0s in $R_i$
        $q_1 = R_{k0} \ldots R_{ki}$
        $q_2 = R_{k(i+1)} \ldots R_{kj}$
        $q_3 = R_{k(j+1)} \ldots R_{k7}$
      $m_k = p_1 q_2 p_3$
    end for
    $C_i = m_1 m_2 \ldots m_3 2$
end procedure
```

Then it acts as the right half (R_{i+1}) of the random sequence for the next block. The generation of R_{i+1} using n_i and R_i makes the encryption process more secure due to the use of different random sequences for each block in each round of the encryption process.



(a) (b) (c) (d)

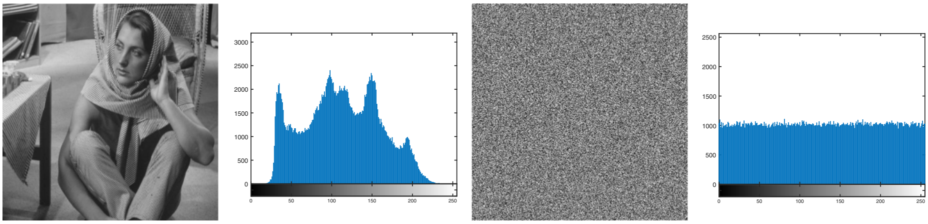
Fig. 7 Test the scheme on lena image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image

As the pixels in a plain image are highly correlated, introducing mutation by flipping the bits left to right brings a drastic change in the pixel values, making the adjacent pixel bit representation completely different. This brings the required change in the pixel distribution in order to maximize the effects of multipoint crossover in the next step.

The genetic crossover operation brings about a change in the order of the bit sequence. This change is in relation to the adjacent bit sequences. In the multipoint crossover operation, each byte in the block is divided into three parts based on the count of $0s$ and $1s$ in R_i of the random sequence. The adjacent bits in the block are crossed over by swapping the first and the last parts of each byte. Genetic operations serve the major purpose of introducing and maintaining the reasonable amount of diversity in the bit sequence of the cipher image $C = C_0C_1 \ldots C_n$. In the encryption process all the XOR operations are followed by genetic mutation and multipoint crossover process which adds benefits of confusion and diffusion. Therefore, the cipher will not be vulnerable to the chosen-plain-text attack and the encryption process remains strong.

4 Security analysis and experimental results

This part analyses the experimental results and the effectiveness of the proposed encryption. For the purpose of analysis images of different nature and sizes are taken namely lena image (512 × 517) in Fig. 7, barbara image (512 × 512) in Fig. 8, baboon image (512 × 512)



(a) (b) (c) (d)

Fig. 8 Test the scheme on barbara image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image

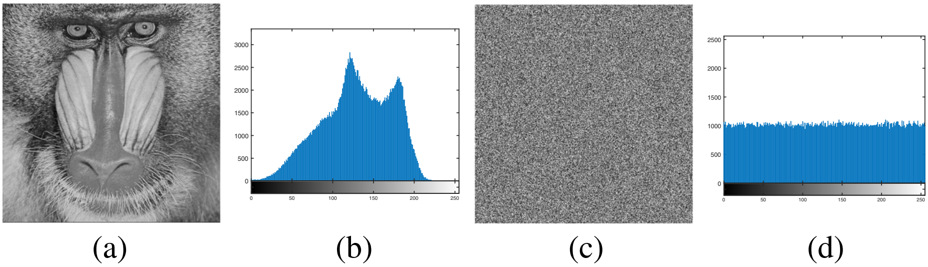


Fig. 9 Test the scheme on baboon image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image

in Fig. 9, Airplane image (512 × 512) in Fig. 10, pepper image (512 × 517) in Fig. 11, goldhill image (512 × 512) in Fig. 12, Finger print image (509 × 612) in Fig. 15, Text image (960 × 493) in Fig. 14, and Squares (600 × 517) in Fig. 13. Further, a color image is encrypted in Fig. 16 to test the capability of the scheme to encrypt color images. The results proves that the propose scheme is robust and applicable to any kind of image with capability of giving high security (Fig. 15).

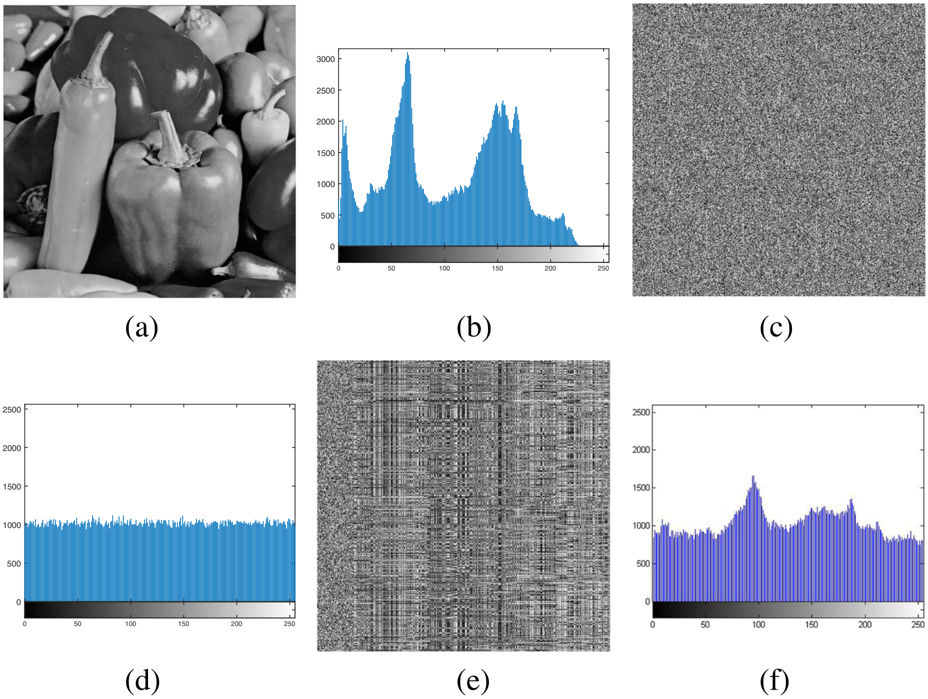
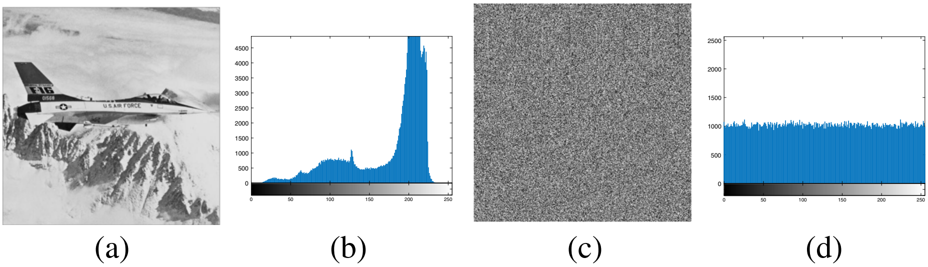
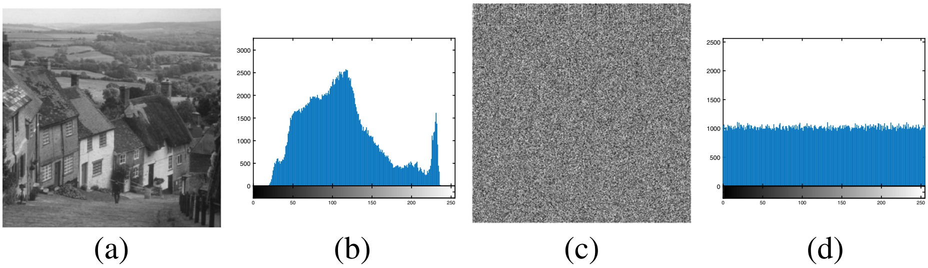


Fig. 10 Test the scheme on peppers image to compare histogram of plain image and cipher image. **a** Test plain (pepper) image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image **e** cipher generated using method in Ref. [17] **f** histogram of cipher generated using method in Ref. [17]



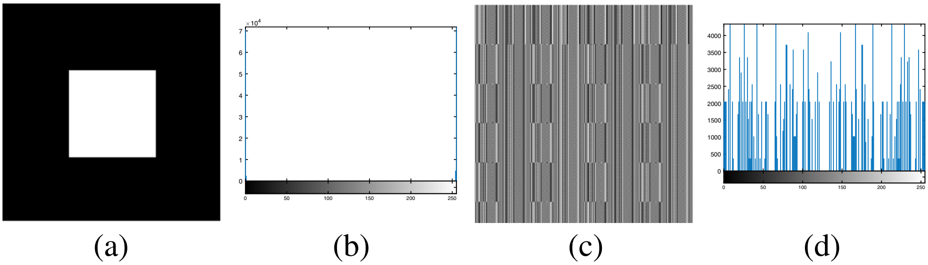
(a) (b) (c) (d)

Fig. 11 Test the scheme on airplane image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image



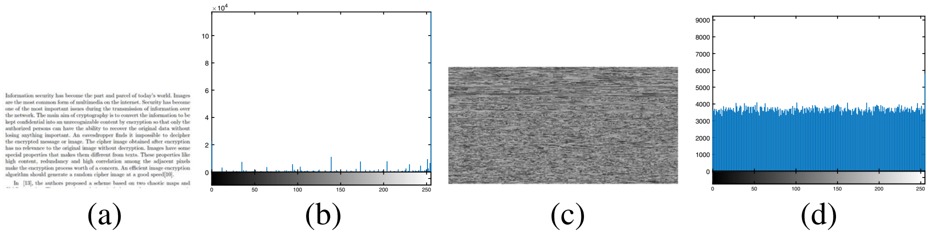
(a) (b) (c) (d)

Fig. 12 Test the scheme on goldhill image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image



(a) (b) (c) (d)

Fig. 13 Test the scheme on Black and white square image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image



Information security has become the part and parcel of today's world. Images are the most common form of multimedia on the internet. Security has become one of the most important issues during the transmission of information over the network. The main aim of cryptography is to convert the information to be kept confidential into an unrecognizable content by encryption so that only the authorized persons can have the ability to recover the original data without losing anything important. An eavesdropper finds it impossible to decipher the encrypted message or image. The cipher image obtained after encryption has no relevance to the original image without decryption. Images have some special properties that makes them different from texts. These properties like high content, redundancy and high correlation among the adjacent pixels make the encryption process worth of a concern. An efficient image encryption algorithm should generate a random cipher image at a good speed[10].

In [13], the authors proposed a scheme based on two chaotic maps and

(a) (b) (c) (d)

Fig. 14 Test the scheme on text image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image

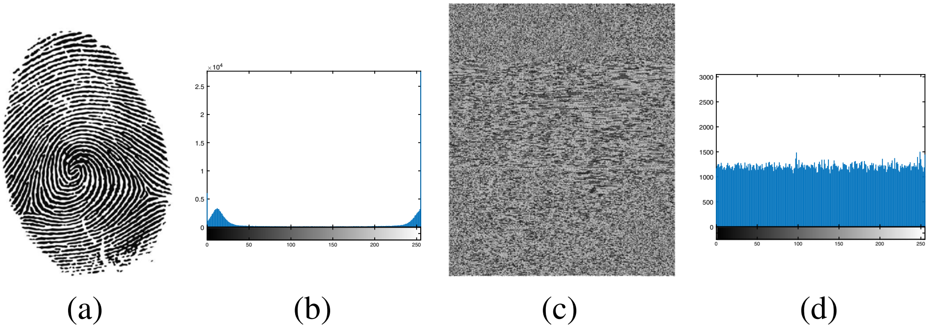


Fig. 15 Test the scheme on a biometric Fingerprint image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of plain image **c** cipher image, **d** Histogram of cipher image

4.1 Die hard statistical test of the HPRNG

The HPRNG is being tested with NIST cryptographic random number test suite, DIEHARD [1]. The HPRNG passes all the tests and shows hing randomness. The test results are given in Table 3.

4.2 Histogram analysis

Histogram reflects the frequency of different pixel values present in the image. It is a must to have a uniform and flat histogram of a cipher image in order to forbid the evesdropper

Table 3 Die hard statistical test results

Test name	P value of HPRNG	Result
Birthday spacings	0.567561	Pass
Frequency	0.645751	Pass
Cumulative sums	0.636215	Pass
Runs	0483624	Pass
Bitstream	0.498281	Pass
Stream count-the-ones	0.512583	Pass
Overlapping sums	0.008502	Pass
Parking lot	0.816929	Pass
Minimum distance	0.755684	Pass
Byte count-the-ones	0.569625	Pass
Craps	0.456867	Pass
DNA	0.556464	Pass
Rank	0.743526	Pass
Linear complexity	0.856325	Pass
FFT	0.109235	Pass
Block frequency	0.848526	Pass
Longest run	0.232654	Pass
Serial	0.344285	Pass
3d spheres	0.383131	Pass
Square	0.496854	Pass

from gaining any beneficial statistical knowledge. The histogram of the encrypted images are plotted and analysed here. It shows that the histogram of the encrypted image is uniform which makes statistical attacks difficult. The original test images are shown in subfigure (a) in all Figs. 7 to 12, their corresponding histogram shown in subfigure (b), in each sub Fig. (c) the encrypted images are presented and corresponding histogram of encrypted images are shown in subfigure (c) in all Figs. 7 to 16. The histograms corresponding to cipher images are almost flat and uniform and contain nearly the same frequency of all intensity values. This shows that the encryption algorithm is efficient enough for encrypting images.

4.3 Key space analysis

The method uses chaotic logistic map, which involves two real numbers x_0, μ as its initial conditions The key $K = \{x_0, \mu, IV\}$. Because the precision of the parameters x_0, μ are taken as 10^{-10} which is roughly equal to 2^{133} for the logistic map. The LFSR uses IV which is of 128 bits which creates key space of 2^{128}. Therefor the final key space becomes $2^{133} \times 2^{128} = 2^{261}$. This large key space eliminates all brute force and exhaustive attacks.

4.4 Correlation coefficient

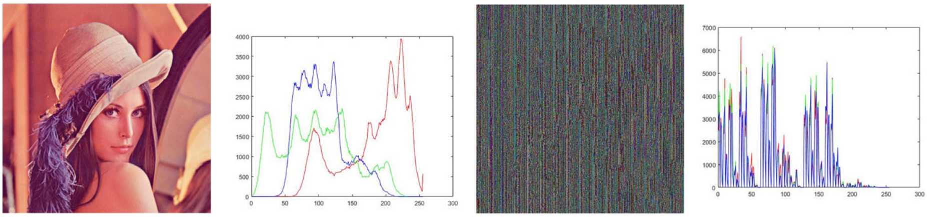
It tells us how much relation exists between the same pixels of the original and the encrypted image. It is calculated from the formula below (4):

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\left((A_{mn} - \overline{A})^2\right)\left((B_{mn} - \overline{B})^2\right)}} \tag{4}$$

Where A and B are the original and the encrypted image respectively and their means. The lower the value of the correlation coefficient indicated a better encryption quality. The test results are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 4.

4.5 Plain-text avalanche effect

Avalanche effect is the study of how differences in an input can affect the resultant difference at the output. Attackers take a pair of images which differ in small magnitude and then generate their cipher images from the same algorithm. Then they compare the two encrypted images, hoping to detect statistical patterns in their distribution. There are two methods used to find performance against differential attacks.



(a) (b) (c) (d)

Fig. 16 Test the scheme on lena color image to compare histogram of plain image and cipher image. **a** Test plain image, **b** Histogram of color image **c** cipher color image, **d** Histogram of cipher color image

Table 4 Correlation coefficient test results and comparison with AES [12], Fu et al.’s method [5], Li et al.’s method [8] and Zhou et al.’s method [17]

Image	Test	AES [12]	Fu et al.’s method [5]	Li et al.’s method [8]	Zhou et al.’s method [17]	Proposed method
Lena	Horizontal	0.0385	0.0230	–0.2881	0.2053	–0.0042
	Vertical	0.0140	0.0485	0.0504	0.2947	–0.0371
	Diagonal	0.0487	–0.0110	–0.0690	0.1699	–0.0304
	Correlation	0.0015	–0.0092	1.2899e–05	0.1114	0.0012
Baboon	Horizontal	0.0107	–0.0027	–0.4484	0.2287	0.0312
	Vertical	0.0008	0.0434	0.0243	0.1719	0.0016
	Diagonal	0.0246	0.0255	–0.0194	0.0284	0.0422
	Correlation	–0.0013	–5.0646e–05	–4.1219e–04	0.0141	2.6075e-04
Peppers	Horizontal	0.0309	0.0329	–0.3148	0.1076	–0.0327
	Vertical	–0.0188	0.0014	–0.0368	0.2099	0.0134
	Diagonal	–0.0223	0.0138	0.0382	–0.0352	0.0119
	Correlation	0.0021	2.8203e-04	–6.0429e-04	0.0353	–3.8166e-04
Goldhill	Horizontal	0.0028	–0.0474	–0.3353	0.6334	0.0094
	Vertical	–0.0114	0.0071	0.0129	0.2541	0.0392
	Diagonal	0.0742	0.0635	0.0199	0.1867	–0.0419
	Correlation	3.2630e-04	–6.4514e-04	–9.6777e-04	0.0866	–0.0019
Airplain	Horizontal	0.0721	0.0074	–0.4578	0.2069	0.0261
	Vertical	–0.0402	–0.0417	–0.0235	0.1214	–0.0115
	Diagonal	–0.0080	–0.0034	0.0121	0.0299	0.0381
	Correlation	2.4233e-04	1.8963e-05	–0.0060	0.0250	0.0012
Barbara	Horizontal	–0.0224	–0.0423	–0.4013	0.2215	0.0203
	Vertical	0.0060	–0.0039	0.0808	0.2941	0.0156
	Diagonal	–0.0087	0.0746	–0.0544	0.1395	–0.0388
	Correlation	0.0020	0.0025	–2.7831e-04	0.1398	0.0044
Squares	Horizontal	0.9643	–0.3698	0.0276	0.3386	0.0266
	Vertical	0.1656	–0.2302	0.0597	0.3568	0.7834
	Diagonal	0.1627	0.3338	0.0139	0.0338	0.0699
	Correlation	–0.2382	–0.0084	–0.0066	–0.0648	0.0021

Number of pixel change rate (NPCR) it measures the percentage of different pixels between two cipher images whose plain images have only one pixel difference. Larger value is better. NPCR is calculated using (5). The test values of NPCR are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 5.

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

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100 \tag{5}$$

Unified average changing intensity (UACI) it measures the average intensity of differences between two cipher images by selecting 5 pixels randomly and the pixel value is changed by 1 bit. The test values of UACI are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 5.

$$UACI : U(C_1, C_2) = \sum_{i,j} \frac{D(i,j)}{T} \times 100\% \tag{6}$$

4.6 Peak signal-to-noise ratio (PSNR) and perceptual security analysis

Peak signal-to-noise ratio is one of the most important indexes for encrypted image quality. The test values of PSNR are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 6.

4.7 Information entropy analysis

The information entropy is defined as the degree of uncertainties in the system. The greater the entropy, the more is the randomness in the image, or the image is more uniform. Thus statistical attacks become difficult. Entropy is defined as in (7)

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \times log_2 \left[\frac{1}{p(m_i)} \right] \tag{7}$$

where p is the histogram counts returned from histogram. For an ideal random image, the entropy is calculated to be 8. So an entropy closer to 8 demonstrate better randomness in the image. The test values of entropy are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 6.

4.8 Avalanche effect

A small change is key or plain text should cause significant change in cipher image. Strict Avalanche 50 bit change in ciphered for 1 bit change in plain. To calculate Avalanche effect Mean Squared Error (MSE) is used in (8). Let C_1 and C_2 are two ciphered image with key differing in a single bit, expressed as (8). The test values of MSE are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 7.

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [C_1(i,j) - C_2(i,j)]^2 \tag{8}$$

Table 5 NPCR and UACI test results and comparison with AES [12], Fu et al.’s method [5], Li et al.’s method [8] and Zhou et al.’s method [17]

Image	Test	AES [12]	Fu et al.’s method [5]	Li et al.’s method [8]	Zhou et al.’s method [17]	Proposed method
Lena	NPCR	0.0061	0.0015	97.7646	99.1760	99.6342
	UACI	0.0023	3.8895e-05	32.3764	21.5003	34.1632
Baboon	NPCR	0.0061	0.0019	96.9589	99.3019	99.6361
	UACI	0.0019	2.2290e-04	31.2434	18.7226	34.1363
Peppers	NPCR	0.0061	0.0019	97.5632	99.2931	99.6391
	UACI	0.0020	1.6904e-04	32.3835	25.3434	34.2135
Goldhill	NPCR	0.0061	0.0019	97.3236	99.1867	99.6506
	UACI	0.0014	1.2117e-04	31.9542	19.4052	34.3525
Airplain	NPCR	0.0061	0.0015	95.9206	97.9420	99.6655
	UACI	0.0020	2.0794e-04	28.9226	17.9106	33.8664
Barbara	NPCR	0.0061	0.0019	97.3125	99.2607	99.6227
	UACI	0.0016	3.9493e-04	31.8001	19.4765	34.1073
Squares	NPCR	0.0061	0.0011	95.8599	43.3239	99.9664
	UACI	0.0021	5.5351e-05	37.3938	39.1000	35.9026

Table 6 PSNR and entropy test results and comparison with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17]

Image	Test	AES [12]	Fu et al.'s method [5]	Li et al.'s method [8]	Zhou et al.'s method [17]	Proposed method
Lena	PSNR	–45.5891	–45.4115	–45.1505	–42.9793	–45.5964
	Entropy	7.9994	7.9694	7.3866	7.5983	7.9994
Baboon	PSNR	–44.6441	–44.6426	–45.1152	–41.5511	–44.6335
	Entropy	7.9993	7.9983	7.2101	7.3701	7.9992
Peppers	PSNR	–45.6949	–45.4936	–45.4968	–44.0536	–45.7079
	Entropy	7.9994	7.9549	7.3597	7.5715	7.9993
Goldhill	PSNR	–45.1146	–45.1102	–45.2077	–42.4816	–45.1275
	Entropy	7.9993	7.9987	7.2991	7.4778	7.9992
Airplain	PSNR	–46.1748	–45.6590	–48.5942	–42.5257	–46.1722
	Entropy	7.9981	7.9357	7.0227	6.7631	7.9991
Barbara	PSNR	–44.9917	–44.9356	–45.1057	–41.8555	–44.9860
	Entropy	7.9992	7.9929	7.2990	7.4664	7.9993
Squares	PSNR	–49.4235	–48.4417	–49.1325	–50.1586	–49.5584
	Entropy	5.1085	6.1072	7.0880	1.1095	6.8240

Table 7 MSE, irregular deviation, and maximum deviation test results and comparison with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17]

Image	Test	AES [12]	Fu et al.'s method [5]	Li et al.'s method [8]	Zhou et al.'s method [17]	Proposed method
Lena	MSE	2.3735e+09	2.2784e+09	2.1455e+09	1.3014e+09	2.3775e+09
	irrDev	319404	301528	238494	294992	320054
	maxDev	262144	262144	262144	308922	262144
Baboon	MSE	1.9094e+09	1.9087e+09	2.1281e+09	936681460	1.9047e+09
	irrDev	258494	262694	225384	315888	257364
	maxDev	262144	262144	262144	338114	262144
Peppers	MSE	2.4320e+09	2.3219e+09	2.3236e+09	1.6667e+09	2.4393e+09
	irrDev	308750	283674	225616	281908	309092
	maxDev	262144	262144	262144	298970	262144
Goldhill	MSE	2.1278e+09	2.1257e+09	2.1740e+09	1.1605e+09	2.1342e+09
	irrDev	292954	292434	213268	311800	292544
	maxDev	262144	262144	262144	307010	262144
Airplain	MSE	2.7162e+09	2.4120e+09	4.7413e+09	1.1723e+09	2.7146e+09
	irrDev	159490	188260	112020	344226	157012
	maxDev	262144	262144	271956	337240	262144
Barbara	MSE	2.0685e+09	2.0419e+09	2.1235e+09	1.0047e+09	2.0658e+09
	irrDev	293252	283768	207786	311462	293062
	maxDev	262144	262144	262144	331816	262144
Squares	MSE	5.7390e+09	4.5777e+09	5.3670e+09	6.7973e+09	5.9200e+09
	irrDev	476836	431210	412552	509062	398014
	maxDev	453266	362580	262144	514302	371684

4.9 Maximum deviation

The high amount of irregularity and deviation of pixels among the original plaintext and cipher image provides good quality of encryption. First the histogram for plaintext image and the encrypted image is taken and their differences are calculated. Let d_i be the absolute difference between the 2 histograms for intensity I, then maximum deviation, D is calculated as shown in (9)

$$D = \frac{d_0 + d_{255}}{2} + \sum_{i=1}^{256} d_i \tag{9}$$

The test values of maximum deviation are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 7.

4.10 Irregular deviation

An important property of encryption is that the plain pixels are having uniform probability distribution to effect all pixels uniformly during encryption process . If the encryption technique considers the plain image as source of random values, the deviation will have uniform distribution. The irregular deviation is measure of uniformity of distribution in the histogram. The uniform distribution of irregular distribution demonstrates the good quality of the encryption algorithm.

First the absolute difference of the plaintext image and the encrypted image is taken and its histogram, H is calculated. The average value for H is calculated as in (10) and (11).

$$M_H = \frac{1}{256} \sum_{i=0}^{255} h_i \tag{10}$$

Where h_i is the amplitude of the histogram at index i. Thereafter irregular deviation is calculated by (11)

$$I_d = \sum_{i=0}^{255} |h_i - M_H| \tag{11}$$

The test values of irregular deviation are presented and compared with AES [12], Fu et al.'s method [5], Li et al.'s method [8] and Zhou et al.'s method [17] in Table 7.

4.11 Performance analysis

The performance efficiency of the proposed scheme was evaluated as time required to encrypt an image in real time. The time consumption was calculated on a system configured with Intel processor Core i5-8250U; 4 GB RAM, 500 GB SSD and Windows 10 operating system. The performance are compared with some existing schemes in Table 8. The encryption process was run with 256 bit kye (K) for all the cases. The encryption time and decryption time are also equal for all the cases as the number of operations are equal for both the processes. It can be notice that the proposed scheme is producing cipher in comparatively faster and very small time.

Table 8 Time needed for encryption or decryption in second with 128 bit key

Image (Size)	Time taken by (in Sec.) Proposed Scheme	Li et al.'s method [8]	Tong et al.'s method [14]
Airplane(512 × 512)	28.850	30.925	98.760
Baboon (512 × 512)	29.659	32.376	106.863
Barbara (512 × 512)	28.965	32.349	104.935
Finger print (509 × 612)	41.975	50.768	162.407
Goldhill (512 × 512)	30.623	30.082	104.762
Lena (512 × 517)	35.810	34.855	109.653
Peppers (512 × 512)	29.726	33.672	99.762
Text image (960 × 493)	43.856	49.051	#
Squares (600 × 517)	27.947	33.957	112.762

5 Conclusion

The proposed encryption algorithm XOR operation and genetic operations (mutation and crossover) are used. The proposed method generates pseudorandom bit sequences of length 256 for one time, which reduces the computational overhead drastically. The large key space makes the encryption more secure and difficult for eavesdropper to compromise the confidentiality of the encrypted information. Histograms of cipher images are flat and uniform and have almost same frequency for each intensity level. The correlation between neighboring pixels in the cipher images turns out to be negligible and nearly equal to zero. Another advantage of the proposed method is that it can be applied to text data as well, despite that the analysis is performed only on images. In the method, the random sequence for current block is generated by previously encrypted block. So, one cannot decrypt the current cipher block without the knowledge of the previous image block. The proposed algorithm presents several interesting features, such as a higher level of security, large key usage, and uniform distribution of pixel values.

Compliance with Ethical Standards This study was carried out at National Institute of Technology Jamshedpur. The research work/ study was not funded by any body or organization.

Conflict of interests Author Dr. Bhaskar Mondal declares that he has no conflict of interest. Author Dr. Tarni Mandal declares that he/she has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

1. Bassham LE III, Rukhin AL, Soto J, Nechvatal JR, Smid ME, Barker EB, Leigh SD, Levenson M, Vangel M, Banks DL, Heckert NA, Dray JF, Vo S (2010) Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep., Gaithersburg, MD, United States
2. Belazi A, El-Latif AAA, Belghith S (2016) A novel image encryption scheme based on substitution-permutation network and chaos. Signal Process 128:155–170. https://doi.org/10.1016/j.sigpro.2016.03.021. http://www.sciencedirect.com/science/article/pii/S0165168416300147
3. Biswas K, Muthukkumarasamy V, Singh K (2015) An encryption scheme using chaotic map and genetic operations for wireless sensor networks. Sensors J IEEE 15(5):2801–2809. https://doi.org/10.1109/JSEN.2014.2380816
4. Boeing G (2016) Visual analysis of nonlinear dynamical systems: chaos, fractals, self-similarity and the limits of prediction. Systems, 4(4). https://doi.org/10.3390/systems4040037, http://www.mdpi.com/2079-8954/4/4/37
5. Fu C, bin Lin B, sheng Miao Y, Liu X, jie Chen J (2011) A novel chaos-based bit-level permutation scheme for digital image encryption. Opt Commun 284(23):5415–5423. https://doi.org/10.1016/j.optcom.2011.08.013. http://www.sciencedirect.com/science/article/pii/S0030401811008431
6. Jolfaei A, Wu X, Muthukkumarasamy V (2016) On the security of permutation-only image encryption schemes. IEEE Trans Inf Forens Secur 11(2):235–246. https://doi.org/10.1109/TIFS.2015.2489178
7. Li C, Zhang LY, Ou R, Wong KW, Shu S (2012) Breaking a novel colour image encryption algorithm based on chaos. Nonlin Dyn 70(4):2383–2388. https://doi.org/10.1007/s11071-012-0626-5
8. Li Y, Wang C, Chen H (2017) A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Opt Lasers Eng 90:238–246. https://doi.org/10.1016/j.optlaseng.2016.10.020. http://www.sciencedirect.com/science/article/pii/S0143816616303517

9. Mondal B, Mandal T (2017) A light weight secure image encryption scheme based on chaos & dna computing. J King Saud Univ- Comput Inf Sci 29(4):499–504. https://doi.org/10.1016/j.jksuci.2016.02.003. http://www.sciencedirect.com/science/article/pii/S1319157816300027
10. Mondal B, Biswas N, Mandal T (2017) A comparative study on cryptographic image scrambling. In: Proceedings of the second international conference on research in intelligent and computing in engineering, annals of computer science and information systems, vol 10. PTI, pp 261–268, https://doi.org/10.15439/2017R47
11. Mondal B, Kumar P, Singh S (2018) A chaotic permutation and diffusion based image encryption algorithm for secure communications. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-018-6214-z
12. Nechvatal J, Barker E, Bassham L, Burr W, Dworkin M, Foti J, Roback E (2001) Report on the development of the advanced encryption standard (aes). J Res Nat Instit Standards Technol 106(3):511
13. Parvin Z, Seyedarabi H, Shamsi M (2014) A new secure and sensitive image encryption scheme based on new substitution with chaotic function. Multimed Tools Appl 75(17):10631–10648. https://doi.org/10.1007/s11042-014-2115-y
14. Tong X-J, Wang Z, Zhang M, Liu Y (2013) A new algorithm of the combination of image compression and encryption technology based on cross chaotic map. Nonlin Dyn 72(1-2):229–241. https://doi.org/10.1007/s11071-012-0707-5
15. Wang LT, McCluskey EJ (1988) Linear feedback shift register design using cyclic codes. IEEE Trans Comput 37(10):1302–1306. https://doi.org/10.1109/12.5994
16. Wang X, Wang Q (2014) A novel image encryption algorithm based on dynamic s-boxes constructed by chaos. Nonlin Dyn 75(3):567–576. https://doi.org/10.1007/s11071-013-1086-2
17. Zhou N, Zhang A, Zheng F, Gong L (2014) Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing. Opt Laser Technol 62(2):152–160. https://doi.org/10.1007/s11433-009-0032-2

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

Bhaskar Mondal (Ph. D.) received his B.Tech. in Computer Science and Engineering (CSE) from WBUT in 2008 and MTech in CSE from Kalyani Government Engineering College, West Bengal, India in 2010. He was awarded with PhD from NIT Jamshedpur, India in 2018. He is saving as an Assistant Professor at National Institute of Technology Patna, India. His research interest includes image encryption, cryptography, machine learning and data analytics.

Tarni Mandal (Ph. D.) is Professor in the Department of Mathematics, National Institute of Technology, Jamshedpur, India. He has published more than 40 research papers in renowned international conference and journals. His area of interest includes cryptography, optimization, fuzzy theory, etc.