Neural Computing and Applications (2020) 32:11859–11873
https://doi.org/10.1007/s00521-019-04668-x

ORIGINAL ARTICLE

A Pareto-optimal evolutionary approach of image encryption using coupled map lattice and DNA

Shelza Suri[1] · Ritu Vijay[1]

Received: 29 March 2018 / Accepted: 6 December 2019 / Published online: 13 December 2019

Abstract
Evolutionary algorithms are generally a suitable approach for optimization problems, having more than one conflicting objectives. For many complicated engineering optimization problems, multi-objective formulations are treated as realistic models. The paper presents and implements a Pareto-optimal image encryption algorithm that uses coupled map lattice (CML) chaos function and deoxyribonucleic acid (DNA) combination to encrypt an image. The discussed work uses multi-objective genetic algorithm (MOGA) to get the optimized results. The proposed two-step algorithm uses pseudo-random number generators, the chaotic method CML and DNA to create an initial population of DNA masks in its initial stage. The MOGA is applied in the second stage to obtain the best mask for encrypting the given plain image. The focus is on the generation of Pareto fronts by using the Pareto generation method of multi-objective optimization. The paper evaluates the performance of the implemented work using standard metrics like key sensitivity, secret key space, number of pixel change rate, unified average changed intensity, entropy, histogram and correlation coefficient. It also discusses the impact of using a genetic algorithm that uses more than one fitness function as the objective for encrypting images.

Keywords Image encryption · Chaos · CML · DNA · MOGA · Pareto optimization

1 Introduction

Due to fast and new emerging technologies, it is very hard to keep information secure. The encryption algorithms are used to protect the information like digital images from the unauthorized persons or attackers [1, 2]. In the last few decades, a lot of work has been done and is being carried out to develop good image encryption methods [3–6]. However, an efficient and secure image encryption method is still a challenging task.

In recent years, many algorithms have been designed using chaos theory as scientists established a perfect match between chaos theory and cryptography. Due to bulk data capacity and high correlations among pixels in an image, the conventional techniques like Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA) are not suitable [2, 7, 8]. Hraoui et al. [9] argue that chaos theory supersedes AES encryption in terms of computational time. Also, encryption using AES requires large computing power as well as high running speed, limiting its use for good encryption [10]. Considering the implementation aspects of AES, modern computers show that the table-lookup S-boxes in AES is vulnerable to the timing attacks that are based on cache-miss lowdown behavior. In contrast, chaotic sequences produce a high degree of confusion and diffusion, thereby improving the entropy of the encrypted image. To develop a larger key-space and higher security using chaotic algorithms, vivid approaches have been proposed in the literature [11–15].

The chaotic system is a nonlinear system having properties like sensitivity to initial conditions, ergodicity, randomness that makes it more suitable for secure image encryption [1, 8]. A chaos-based image encryption method has mainly two stages permutation and diffusion. In permutation, shuffling of the image pixels is done without changing their values and in diffusion, the value of the pixels is changed by using chaotic sequences. The

✉ Ritu Vijay
vritu@banasthali.ac.in

Shelza Suri
shelza_ecn@yahoo.com

[1] Department of Electronics, Banasthali Vidyapith, Banasthali, Rajasthan, India

permutation leads to better encryption, and diffusion improves the security. Therefore, to have an algorithm with good encryption and higher security, both permutation and diffusion are applied together [11, 12, 16, 17]. The literature also reveals a very important fact that use of only one chaotic map for encrypting image is less secure and has very small key space. Hence, many approaches have been proposed to develop algorithms with larger keyspace and higher security [11–15].

Aldeman in 1994 performed the first analysis of DNA computing. It has many features like large storage, very low power consumption and huge parallelism [18, 19]. Many image encryption schemes using the combination of DNA encoding and chaotic mapping have been proposed [1, 5, 12, 19–21]. These algorithms use the few biological and arithmetic operations of DNA sequences like DNA addition, subtraction and DNA XOR and all bases rules of DNA. Recently, Guesmi et al. used secure hash algorithm (SHA-2) with DNA to develop a chaos-DNA-based hybrid approach to encrypt images [1].

Evolutionary algorithms (EAs) are another recent approach that is being used by the researchers to obtain image encryption algorithms with higher security and better encryption [2, 22–24]. In [2, 22], two different fitness functions are used as primary fitness functions. However, in [23], it is clearly shown that using a single function as the prime objective affects the value of another objective. The algorithm uses chaos-DNA combination to generate an initial population of DNA masks and bi-objective genetic optimization to generate the final cipher image.

Many real engineering problems are not satisfactorily characterized by single objective measures and require the simultaneous optimization of more than one conflicting objectives at the same time. The systems which have multiple conflicting objectives are not easily optimized with the simple optimization process [25]. Image encryption is also a multi-objective problem. The proposed image encryption algorithm is an extension to the work done in [23]. The implemented work uses two-dimensional chaotic function coupled map lattice (CML) and the GA-based multi-objective optimization to generate the Pareto-optimal solution. The work of [23] uses two chaos functions logistic map (LM) and transformed logistic map (TLM) with combination of DNA and weighted bi-objective GA. The proposed work of this paper combines CML with DNA–GA combination. It evaluates information entropy and CC with respect to a number of iterations with all three chaotic functions LM, TLM and CML. Two hundred iterations are taken into consideration to compute the information entropy. The results show that the CML–DNA–GA approach gives the highest entropy and the lowest CC values. Also, instead of using weighted bi-objective optimization that requires fixed weights assigned by the user, it uses Pareto-optimal optimization approach to have better results. Section 2 discusses the elementary concepts of CML, DNA and MOGA approach. Section 3 presents the proposed approach; the experimental results are given in Sects. 4 and 5 concludes the proposal.

2 Elementary concepts

This section of paper discusses fundamentals of chaotic function CML, DNA computing and MOGA approach of optimization.

2.1 Coupled map lattice (CML)

The simplest and the most commonly used chaos map employed by the researchers to perform image encryption is logistic map (LM) [2]. It is mathematically expressed as:

$$X_{j+1} = \mu * X_j\left(1 - X_j\right) \tag{1}$$

where logistic map parameter $\mu \in [0, 4]$ and $X \in [0, 1]$. However, recent researchers have proven that the LM suffers from shortcomings such as uneven distribution of sequences, stable and blank windows, and a weak key as shown by Fig. 1.

Coupled map lattice (CML) was introduced and investigated by Kaneko as a two-dimensional model for spatiotemporal chaos [26, 27]. The CML function has discrete time and discrete space with continuous state. It is being used widely for implementing chaotic cryptosystem in the current research scenario [21, 28–30]. It performs well in encryption due to its intrinsic properties like large key space, longer periodicity, more initial parameters and parallel implementation [21]. There are two profound advantages of CML. The first is that CML can be efficiently managed numerically as well as analytically [31]. The other and most important merit is that CML incorporates the crucial features of spatiotemporal chaos. Spatiotemporal chaotic systems produce self-synchronizing stream cipher, consequently providing a more secure encryption as compared to both low-dimensional chaotic systems [32–35] and hyper-chaotic systems [4, 36, 37]. The spatiotemporal features allow CML to overcome the weakness of encryption efficiency, security and inability to resist chosen-plaintext and known-plaintext attacks. Moreover, the spatiotemporal features also instill high and positive Lyapunov exponents, superior key space, greater ergodicity and inaccurate prediction of chaotic sequences, thereby proving that CML is a better solution for data protection [38]. The two-dimensional dynamical map of CML is mathematically expressed as [39, 40]:

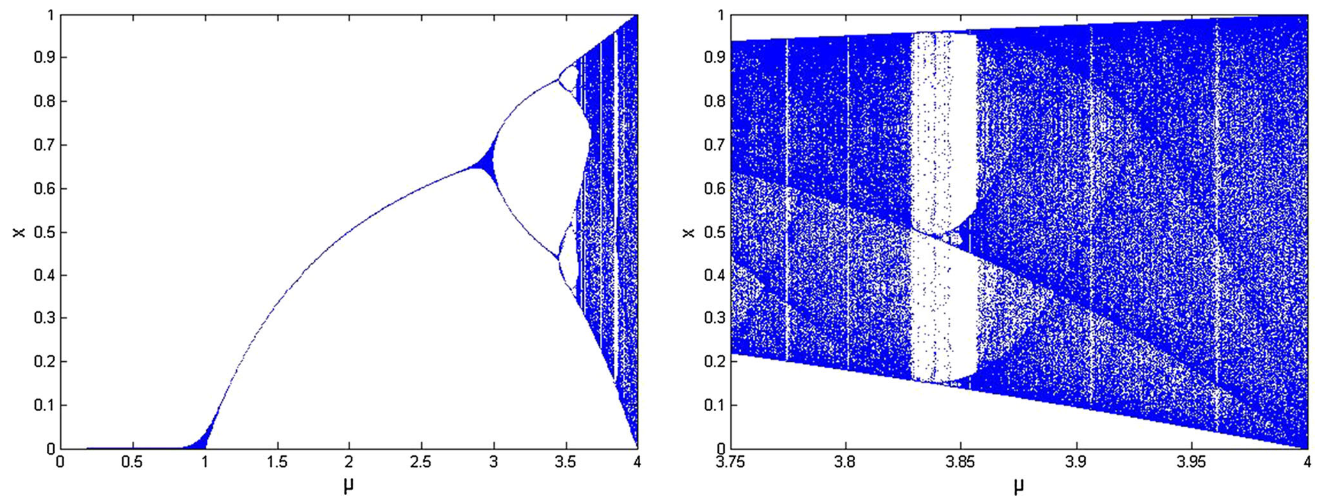


Fig. 1 Logistic map bifurcation and blank window

$$x_{i+1}(n) = (1-\delta)f(x_i(n)) + \delta f(x_i(n-1)) \tag{2}$$

where n is lattice site index and can have maximum value equal to lattice length L, i.e., $n = 1, 2, \ldots\ldots L.$, $x_i(n)$ defines the state variable of *nth* site at time *i*., δ represents coupling strength and lies in the unit interval (0,1) and function f is chaotic logistic map function.

Although a chaos system possesses well-defined properties for encryption, the computer realization of low-dimensional chaos system is a fundamental problem that persists. Due to the finite precision in computer simulation, it is difficult to produce a complete chaotic orbit by the trajectories of that chaotic system [41]. Infinite precision is required for determinism of long-term chaotic behavior. The chaotic system produces different initial states due to its periodicity, and hence, observations cannot be determined [42]. Precisely, there are mainly two problems that occur due to this finite precision of computer systems: transitory time to periodic orbits and average period. However, CML falls in the category of spatiotemporal chaotic system which shows chaotic properties in both time and space [14, 32, 43–45]. Due to multiple chaotic coupled oscillators present in spatiotemporal systems, CML exhibits a periodicity larger than that of temporal chaotic systems [46]. Hence, the sufficiently large period of the CML chaotic orbit is hardly ever reached in practical conditions, thereby avoiding the concern of periodicity [31].

2.2 Deoxyribonucleic acid (DNA)

Nowadays, DNA computing is being used as a popular tool for developing higher security cryptosystems. DNA-based image cryptosystems use DNA as a source to carry information and use DNA computing methods for achieving encryption [19]. DNA has four nucleic acid bases that are adenine (A), cytosine (C), guanine (G) and thymine (T) where A is reciprocal of T and G is reciprocal of C and vice versa. These relationships are termed as Watson–Crick complement rule given by the two scientists. Similar to the conventional binary operations, the DNA sequences also have addition, subtraction and XOR operations. Table 1 gives the DNA encoding rules, and Table 2 shows DNA XOR operation that is used widely for performing encryption.

Table 1 DNA encoding rules

Rule	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
00	A	A	T	T	C	C	G	G
01	G	C	G	C	T	A	T	A
10	C	G	C	G	A	T	A	T
11	T	T	A	A	G	G	C	C

Table 2 DNA XOR operation

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

2.3 Multi-objective optimization

The objectives in many real-time problems are conflicting with each other, so optimization using one objective may provide an unacceptable result with respect to another objective. Therefore, optimization using more than one objective, which satisfies the objectives at an acceptable level, is preferable [25]. A multi-objective optimization is formulated as [25, 47, 48]:

$$\text{Min } or \text{ Max } q = f(p) = \left((f_1(p), f_2(p) \ldots \ldots \ldots f_n(p)) \middle| \begin{matrix} p = (p_1, p_2 \ldots \ldots p_n) \in P \\ q = (q_1, q_2 \ldots q_m) \in Q \end{matrix} \right) \tag{3}$$

where P is the parameter space corresponding to decision vector p and Q is the objective space corresponding to objective vector q.

The simultaneous optimization of multiple objectives has a set of alternate solutions that are known as the non-dominated or Pareto-optimal solutions. For these solutions, one objective cannot be improved without degrading other objective. The Pareto sets or non-dominated solutions show different trade-off or compromises between the objectives [25].

Genetic algorithms can explore for many non-inferior solutions equally by maintaining a population of solutions, which makes GAs very interesting for dealing with multi-objective problems. Genetic algorithm has been successfully applied in image encryption algorithms that are proposed in [2, 23, 24]. The algorithms proposed in [2, 24] use single objective, and algorithm in [23] converts the bi-objective problem to single objective by using weighted sum approach [49, 50]. However, weighted sum approach also requires prior information about the weights to be assigned to each objective. The proposed approach uses multi-objective GA that adds the obtained non-dominated solutions to the population.

3 Proposed method

This section discusses the proposed approach to encrypt the image. The algorithm mainly consists of six steps.

3.1 Image input and shuffling using CML

The first step converts the two-dimensional gray scale image into a one-dimensional array. Then, this one-dimensional array is shuffled using location map generated using CML function. The pseudo-code shows how a shuffled image is obtained using the CML function.

```
==================================================================
 Image Input and Shuffling
==================================================================
declare
     input_image[1...M, 1...N] /* input image */
     n = M*N /* total number of pixels */
     odim[1...n] /* 1-dimentional image */
begin
     /* converting 2D image to 1D */
     for i = 1 to M
     do
          vert_index = (i * N) - N
          for j = 1 to N
          do
               odim[vert_index + j] = input_image[i, j]
          end
     end

     x0 = [0.31456, 0.65321] /* initial seed for cml */
     x = CML(n, x0) /* Generating Coupled Map Lattice */

     shuffled_image[1...n] /* shuffled image */
     /* shuffing image */
     for i = 1 to n
     do
          shuffled_image[i] = odim[i] XOR ((x[i] * 10^16) % 256)
     end
end
==================================================================
```

3.2 DNA encoding

The shuffled image of step one is converted into DNA sequence using DNA encoding rules. The pseudo-code describes how second step is performed using the input shuffled image from the first step.

```
=====================================================================
DNA Encoding
=====================================================================
declare
      rules_on_shuffle[1...n] /* dna encoding rule */
begin
      /* generating dna encoding rules */
      for i = 1 to n
      do
            rules_on_shuffle[i] = ((x[i] * shuffled_image[i]) % 8) + 1
      end

      dna = '' /* empty DNA sequance */
      bin[1...8] /* array to store 8bit binary representation */
      /* converting shuffled image into DNA sequance using generated
rules */
      for i = 1 to n
      do
            bin = 8-bits binary representation of shuffled_image[i]
            for j = 1 to length(bin) (step 2)
            do
                  dna = dna | dna_symbol(rules_on_shuffle[i], bin[j],
bin[j+1]) /* | - is concatation operator */
            end
      end
end
=====================================================================
```

3.3 Secret key generation using PRNG function

The third step of the proposed algorithm generates the secret key using the PRNG function [51]. The function has been used to increase the security. The initial condition x_0 and control parameter a are calculated using 32 digits of hexadecimal that are of 128 bits. This secret key is divided into four sections for the calculations of parameter and initial condition as described in [52–54]. The pseudo-code shows how this step of proposed approach is performed.

```
=====================================================================
Secret Key Generation using PRNG function
=====================================================================
declare
      hex[] = an array for 32 hexadecimal digits /* a 32 length integer
array holding decimal equivalents for each hexadecimal digit */
      A = B = C = D = 0 /* variables to hold values for 8 digits of
hexadecimal taken together */
      a = 0 /* control parameter */
      x0 = 0 /* initial condition */
begin
      /* calculating decimal equivalent for hexadecimals each */
      for i = 0 to 7
            A = A*16 + h[i]
            B = B*16 + h[8+i]
            C = C*16 + h[16+i]
            D = D*16 + h[24+i]

      A = A mod 2^32+1
      B = B mod 2^32+1
      C = C mod 2^32+1
      D = D mod 2^32+1

      /* generating parameter and initial conditions */
      a = 3.999 + ((A+B) mod 1)*0.001
      x0 = (C+D) mod 1
end
=====================================================================
```

3.4 GA initialization and fitness function calculation

This step of proposed method initializes the parameters of GA, and random initial population is generated. The parameters of evolutionary algorithms, including GA, depend on the specific problem. Different combinations of the GA parameters such as population size (e.g., 10, 20, 30), mutation rate and crossover rate have been tested using multiple runs of the proposed algorithm to evaluate entropy and CC parameters. The combination with population size of 20, mutation rate of 0.06 and crossover rate of 0.6 produced better results. Hence, the said GA parametric values have been used to obtain the results. The pseudo-code shows the parameter initialization and population generation for the GA.

3.5 Pareto-front computation and multi-objective optimization

The step five of the proposed algorithm computes the Pareto-front by applying multi-objective optimization. The two objectives used are entropy and correlation coefficient (CC). The entropy has been maximized, and the CC has been minimized simultaneously to obtain the non-dominated set of solutions. Each time the current generated Pareto solutions are concatenated with the population of the previous iteration to have the input population for the next iteration. The pseudo-code gives the detailed description of the optimization and computation performed.

```
======================================================================
GA Parameter Initialization & Initial Population Generation
======================================================================
declare
      max_iteration = 1000 /* maximum GA iteration */
      population_size = 20 /* population size */
      mutation_rate = 0.06 /* mutation rate */
      crossover_rate = 0.6 /* crossover rate */
      population[1...population_size] /* array to hold current
population */
      population_keys[1...population_size, 1...2] /* array to hold keys
related to population[i] */
      current_seed = seed /* initial seed */
begin
      /* generating random population */
      for i = 1 to population_size
      do
            random_mask = generate_random_mask(current_seed, n)
            population[i] = random_mask.dna
            population_keys[i, 1] = random_mask.keys[1]
            population_keys[i, 2] = random_mask.keys[2]
            current_seed = random_mask.next_seed
      end
end
======================================================================
=======================================================================
Multi-objective algorithm and Pareto front calculation
=======================================================================
declare
      objective_count = 2 /* number of objectives */
      fitnesses[1...population_size, 1...2] /* 2-dimentional array for
fitnesses */
      decoded[1...n] /* array to store decoded DNA sequance */
      previous_pareto[] /* array to store previously generated pareto
solutions */
      previous_fitnesses[] /* array to store fitnesses of previous
pareto */
begin
      /* evolution algorithm */
      for iteration = 1 to max_iteration
      do
            //////* fitness calculation *//////
            for i = 1 to population_size
            do
                  Qdash = dnaxor(dna, population[i])
                  current_rule = 1
                  for ii = 1 to 4n (step 4)
                  do
                        s = ''
                        for j = ii to ii+3
                        do
                              s = s |
dna_to_binary(rules_on_shuffle[current_rule], Qdash[j])
                        end
                        decoded[c] = binary_to_dec(s)
                        c = c + 1
                  end
                  tmp_img = to_two_dime(decoded, M, N)
                  fitnesses[i, 1] = entropy(tmp_img)
                  fitnesses[i, 2] = coorelation(input_image, tmp_img)
            end

            //////* pareto front generation *//////
            input = [] /* empty input array */
            for i = 1 to length(previous_pareto) /* append content of
previous_pareto to input */
            do
                  input[next_index] = previous_pareto[i]
            end
            for i = 1 to length(population) /* append content of
current population */
            do
                  input[next_index] = population[i]
            end
            previous_pareto = [] /* clear previous pareto sols */
            input_fitnesses = vertical_concat(fitnesses,
previous_fitnesses)
            fronts = [] /* current pareto solutions */
            for k = 1 to length(input_fitnesses)
            do
                  j = 0 /* meaningful pareto count */
                  fitness_k = input_fitnesses[k]
                  for kk = 1 to length(input_fitnesses)
                  do
                        if k != kk
                        do
                              j = j + 1
                              tmp_fitness[j, 1] = fitness_k[1] -
input_fitnesses[kk, 1]
                              tmp_fitness[j, 2] = fitness_k[2] -
input_fitnesses[kk, 2]
                        end
                  end
                  if fitness_k contains non-zero element /* if it is
true means kth solution is in pareto front */
=======================================================================
======================================================================
                do
                     fronts[next_index] = k
                end
           end
           previous_fitnesses = [] /* clear it */
           for f = 1 to length(fronts)
           do
                previous_fronts[f] = input[fronts[f]] /* add current
fronts to previous_fronts */
                previous_fitnesses[next_index] =
input_fitnesses[fronts[f], 1:2] /* add respective front fitness to
previous_fitnesses */
           end

           /* flush current population and generate next generation by
crossover and mutation */
           new_data = crossover_and_mutation(population,
population_keys, mutation_rate, crossover_rate)
           population = new_data.population
           population_keys = new_data.population_keys
      end
end
======================================================================
```

3.6 Image fusion and mask

The final step of the proposed approach chooses the solution with the maximum entropy value. The DNA XOR operation is used to obtain the encrypted image. The pseudo-code describes how cipher image is created using the XOR operation.

```
======================================================================
Image Fusion and Mask
======================================================================
global_best = maximum entropy from previous_fronts
Qdash = dnaxor(dna, global_best)
Generate encrypted image using Qdash
======================================================================
```

4 Simulation and result analysis

The current section discusses the simulation details, input details and performs analysis of the proposed encryption algorithm in terms of different parameters. A system with an i-7 processor, Windows 10 operating system, and the MATLAB version 2013 has been used for the implementation. As described in the previous section, an initial population of size 20 with mutation rate of 0.06 and cross over rate of 0.6 has been used. A total of 1000 iterations are used to obtain the simulation results.

4.1 Test set

The proposed algorithm has been tested using three different grayscale images "Pepper," "Lena" and "Onion" as shown in Fig. 3. Each testing image is of size 256*256. As described by the results in the figure, the encrypted images have no relationship with the source images. Hence, a good encryption results are shown by the proposed algorithm.

4.2 Key sensitivity analysis

A good encryption is the one that is sensitive to changes in its keys and makes brute-force attack difficult. There are two ways to determine key sensitivity as given in [55] that are: (i) when a same image is encrypted using secret keys having nominal difference, the resulting encrypted images must be completely different, and (ii) when an encrypted image is decrypted using a decryption key slightly different from the encryption keys, the correct plain image cannot be obtained.

First, the plain image shown in Fig. 2a is encrypted using two secret keys having a slight difference, which are $x_0 = 0.345687650000123$ and $x_0 = 0.345687650000124$. The resulting two encrypted images are shown in Fig. 2b and in Fig. 2c, which have a difference of 99.64%. Secondly, when the plain image encrypted with $x_0 = 0.345687650000123$ is decrypted using the decryption key with $x_0 = 0.345687650000122$, the decrypted image is different from the original image. The resulting decrypted image is shown in Fig. 2d. Thus, the algorithm shows high key sensitivity.

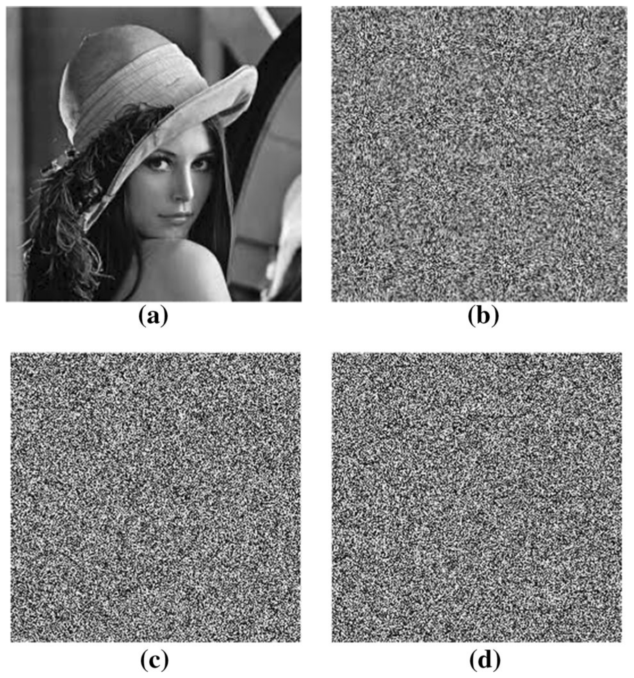


Fig. 2 **a** Plain image. **b** Encrypted image with first secret key. **c** Encrypted image with second secret key. **d** Decrypted image using decryption key different from encryption key

4.3 Key space analysis

This analysis examines the ability to resist the brute-force attack which depends on the number of keys possible in the encryption algorithm. This paper uses PRNG to generate128 bit secret key, which thus makes the keyspace as large as 2^{128}. This large keyspace satisfies the basic requirement to resist brute-force attack which says that key space must have more than 2^{100} possible keys to resist an exhaustive attack [54].

4.4 Histogram analysis

A uniform histogram makes it difficult to attack image data statistically. As shown in Fig. 2, the unencrypted source image histogram has a number of pixel values concentrated in some area, while the histogram of the encrypted image is uniform. The magnitude of uniformity for all three image data set specifies the efficiency of the proposed image encryption algorithm.

4.5 Entropy analysis

Entropy analysis defines the magnitude of uncertainty and randomness. An algorithm with the large information entropy value is considered to be more secure than the algorithm with the low entropy value. The entropy, $H(s)$ given in Eq. (4), of any image is defined as a function of number of gray levels used in image N and probability of the occurrence of symbols $P(s_i)$ [2].

$$H(s) = \sum_{i=0}^{2N-1} P(s_i) \log_2\left(\frac{1}{p(s_i)}\right) \quad (4)$$

The proposed work has evaluated information entropy using entropy only fitness function of the Pepper image with respect to a number of iterations with all three chaotic functions LM, TLM and CML using entropy only fitness function. Two hundred iterations are taken into consideration to compute the information entropy. Figure 4 shows that the CML–DNA–GA approach gives the highest entropy. Also, it converges faster than the LM–DNA–GA and TLM–DNA–GA combinations.

Table 3 and Fig. 6 show that the entropy value obtained by the proposed algorithm is closer to eight that is considered as an ideal value for a good encryption algorithm. It can be observed from the results of Table 3 that using Entropy and CC simultaneously as a fitness function gives balance results with respect to both the fitness functions, whereas using a single fitness function degrades the performance of the algorithm with respect to other fitness function.

4.6 Correlation coefficient (CC) analysis

Correlation coefficient (CC) for an efficient and secure encryption algorithm should have value close to zero. The correlation coefficient between two adjacent pixels is computed by using Eqs. (5)–(8).

$$COV(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_{i\_}E(x))(y_i - E(y)) \quad (5)$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (6)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_{i\_}E(x))^2 \quad (7)$$

$$r_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x) \times \sqrt{D(y)}}} \quad (8)$$

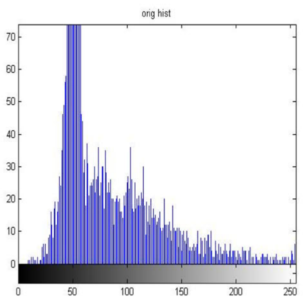
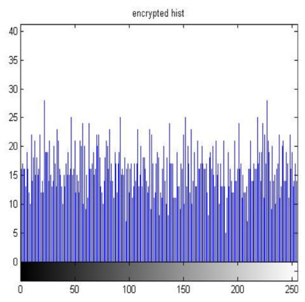
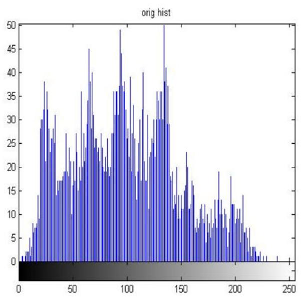
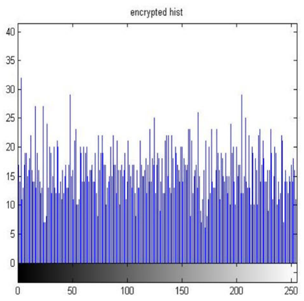
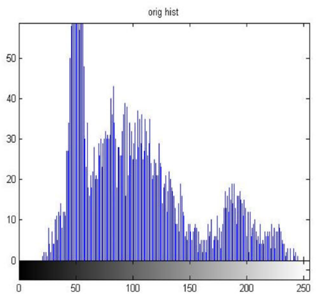
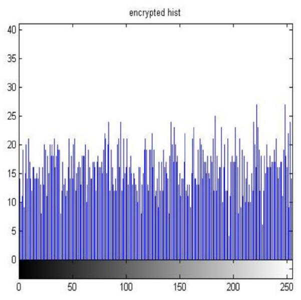
Source Image	Encrypted Image	Histogram of Source Image	Histogram of Encrypted Image
Pepper			
			
Onion			

Fig. 3 Histogram analysis and encrypted image

where $E(x)$ is expectation or mean, $D(x)$ denotes variance and $cov(x, y)$ computes the covariance. Also, x and y are the gray values of two adjacent image pixels.

The proposed work has evaluated CC using CC only fitness function of the Peppers image with respect to the number of iterations with all three chaotic functions LM, TLM and CML. Two hundred iterations are taken into consideration to compute the CC. Figure 5 shows that the CML–DNA–GA approaches to zero value faster than the other two methods and also, has better CC value than the LM–DNA–GA and TLM–DNA–GA combinations.

Table 3 and Fig. 6 show the CC obtained by the proposed algorithm for the all three image data set. It can be observed from the results of Table 3 that using Entropy and CC simultaneously as a fitness function gives balance results with respect to both the fitness functions, whereas using a single fitness function degrades the performance of the algorithm with respect to other fitness function.

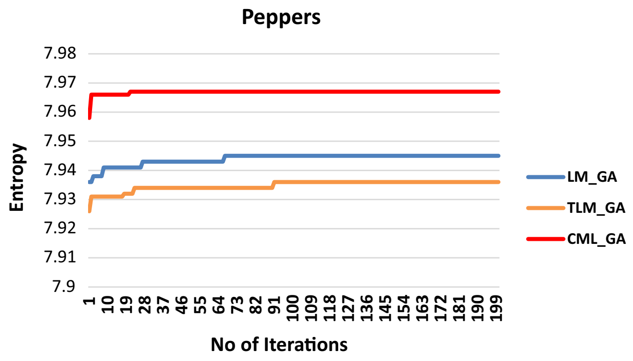


Fig. 4 Peppers_entropy

4.7 Analysis of differential parameters

Intruders try to attack the encrypted images by analyzing how making slight changes in the input can affect the output. To analyze the effect of one-pixel change, two parameters number of pixels change rate (NPCR) and unified average changed intensity (UACI) are commonly used. NPCR given by Eq. (9) is the pixel change rate with respect to one-pixel change in the plain image, and UACI given by Eq. (10) computes the average intensity difference between the plain and the encrypted image.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{9}$$

where a $D(i, j)$ is a two-dimensional array having the same size as the encrypted images

$$UACI = \frac{1}{W \times H}\left[\sum_{ij} \frac{C_1(i,j) - C_2(i,j)}{255}\right] \times 100\% \tag{10}$$

$W =$ width of C_1 and C_2, H=Height of C_1 and C_2, where $C_1(i, j)$ and $C_2(i,j)$ are corresponding images to the original images that have only one-pixel difference. Table 3 shows the NPCR and UACI values evaluated by the proposed image encryption algorithm.

4.8 Comparison with earlier proposed techniques

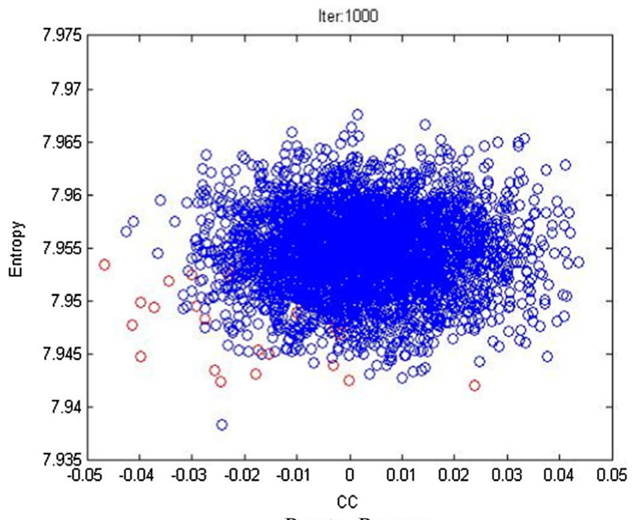
This section gives comparison of the proposed approach with the earlier proposed approaches in [1, 2, 23, 24]. The proposed algorithm has the better key space as is proposed in [1]. Undoubtedly, the LM map used in [2] and [23, 24] is the most commonly used and mathematically simplest to implement. However, as described in Fig. 1 earlier it suffers from various limitations. Hence, the proposed work uses CML location map that outperforms the LM location map used in [2, 23, 24]. It uses multi-objective optimization that offers better results than the weighted sum approach used by the authors of [23]. Also, the proposed approach opens a new area of multi-objective optimization of image encryption techniques for future work in the field of image encryption (Table 4).

5 Conclusion and future work

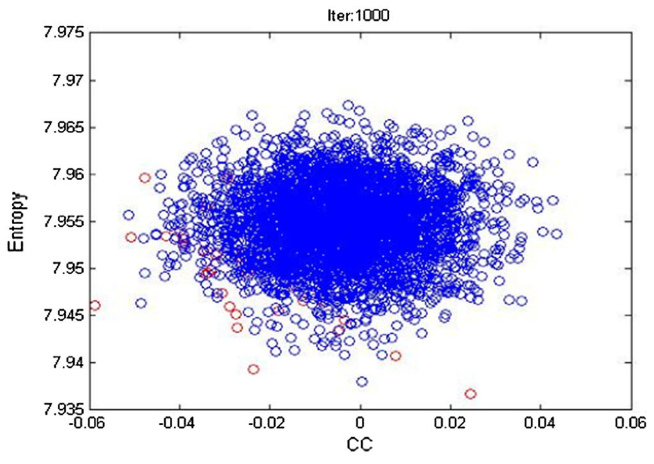
The authors in [2] proved that CC is a better optimizer than entropy. However, in their extended work in [24], entropy is used as fitness function. In [23], weighted sum GA is

Table 3 Comparison using different fitness functions

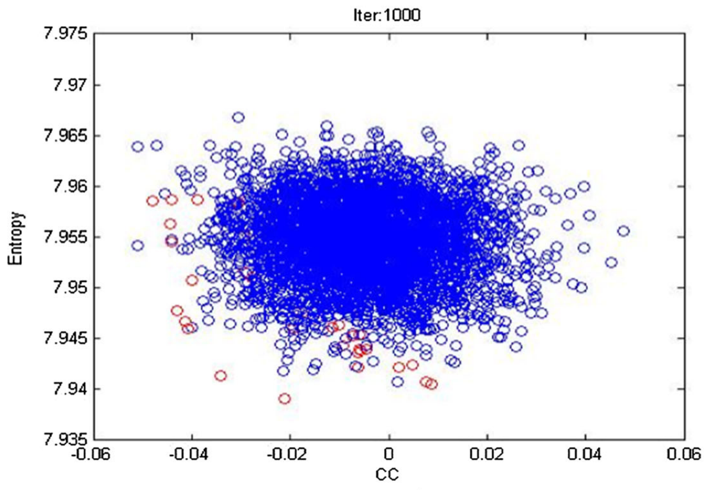
Fitness functions	Image	Entropy	CC	NPCR	UACI
Entropy only	Lena	7.9686	− 0.0682	99.4054	32.0813
	Onion	7.9679	− 0.0625	99.5091	30.7565
	Pepper	7.9667	− 0.0686	99.4609	31.3851
	Average	7.9677	− 0.0664	99.4584	31.4076
CC only	Lena	7.9433	− 0.0324	99.4338	31.4629
	Onion	7.9451	− 0.0385	99.4494	31.4027
	Pepper	7.9513	− 0.0305	99.4559	31.3554
	Average	7.9465	− 0.0338	99.4463	31.4070
Entropy and CC	Lena	7.9535	− 0.0467	99.4629	31.9421
	Onion	7.9461	− 0.0586	99.3652	30.9232
	Pepper	7.9563	− 0.0444	99.5361	31.1996
	Average	7.9519	− 0.0499	99.4547	31.3549



Pareto_Peppers



Pareto_Lena



Pareto_Onion

Fig. 6 Pareto front computation

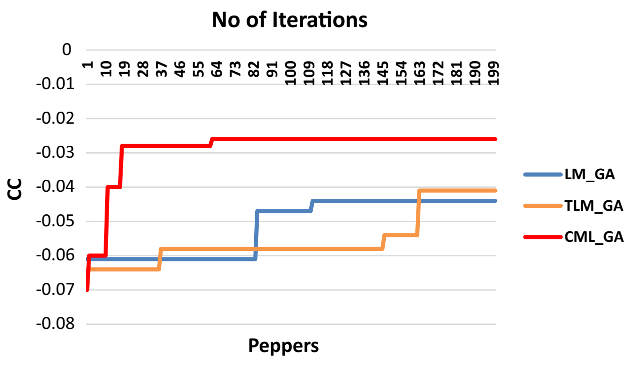


Fig. 5 Pepper_CC

used for having balanced results with respect to both entropy and CC. The prime objective of the implemented work is image encryption with the aid of DNA and multi-objective GA optimization. The discussed work computes the Pareto fronts by using two fitness functions Entropy and CC simultaneously. The performance of the proposed algorithm has been tested using standard parameters, and comparison has been done with some recently proposed techniques. The work can further be extended by combining more encryption methods with more advanced multi-objective optimization techniques.

Table 4 Comparison with earlier proposed techniques

Algorithms	Approach	Encryption parameters					Observations	Limitations
		Key space	Entropy	CC (Avg.)	NPCR	UACI		
Abdullah et al. [2]	LM + GA	120-bits	7.9978	0.003	97.1394	33.1084	Optimization method GA enhances the performance of LM-based approach	LM suffers from the problems of uneven distribution of sequences, stable and blank windows, and a weak key
Enayatifar et al. [24]	LM + DNA + GA	120-bits	7.9997	0.0025	99.7103	33.6297	DNA encoding increases the encryption efficiency of the results proposed by the authors in [2]	LM suffers from the problems of uneven distribution of sequences, stable and blank windows, and a weak key
Guesmi et al. [1]	Lorenz system + DNA	SHA-2	–	0.0560	–	–	SHA-2 increases key space, and DNA increases the encryption efficiency	Only CC with higher values was evaluated
Suri et al. [23]	LM + DNA + Weighted GA and TLM + DNA + Weighted GA	120-bits	7.8091 (LM) 7.8035 (TLM)	− 0.0548 (LM) − 0.0433 (TLM)	99.707 (LM) 99.5117 (TLM)	30.5706 (LM) 30.617 (TLM)	Using weighted GA gives balanced results with respect to applied fitness functions than using a single fitness function	TLM removes problems of LM; however, CML is more powerful and dynamical chaos than LM and TLM
Proposed approach	CML + DNA + MOGA	PRNG 128-bits	7.9519	− 0.0499	99.4547	31.3549	Use of CML–DNA with Pareto-optimal optimization has provided good value of encryption parameters	There is still a scope of improvement in encryption parameters by using improved methods of chaos and optimization methods

Compliance with ethical standards

Conflict of interest The authors declare that they have any conflict of interest.

References

1. Guesmi R et al (2016) A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2. Nonlinear Dyn 83(3):1123–1136
2. Abdullah AH, Enayatifar R, Lee M (2012) A hybrid genetic algorithm and chaotic function model for image encryption. AEU-Int J Electron Commun 66(10):806–816
3. Guan Z-H, Huang F, Guan W (2005) Chaos-based image encryption algorithm. Phys Lett A 346(1):153–157
4. Gao T, Chen Z (2008) A new image encryption algorithm based on hyper-chaos. Phys Lett A 372(4):394–400
5. Chai X, Chen Y, Broyde L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88:197–213
6. Ye G, Huang X (2017) An efficient symmetric image encryption algorithm based on an intertwining logistic map. Neurocomputing 251:45–53

7. Cheddad A et al (2010) A hash-based image encryption algorithm. Opt Commun 283(6):879–893
8. Hao Z et al (2017) Application of coupled map lattice with parameter q in image encryption. Opt Lasers Eng 88:65–74
9. Hraoui S et al. (2013) Benchmarking aes and chaos based logistic map for image encryption. In: Computer systems and applications (AICCSA), 2013 ACS international conference on. IEEE
10. He J, Li Z, Qian H (2010) Cryptography based on spatiotemporal chaos system and multiple maps. JSW 5(4):421–428
11. Zhang Q, Xue X, Wei X (2012) A novel image encryption algorithm based on DNA subsequence operation. Sci World J 2012:286741. https://doi.org/10.1100/2012/286741
12. Wei X et al (2012) A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. J Syst Softw 85(2):290–299
13. Liu J, et al (2008) A cryptosystem based on multiple chaotic maps. In: Information processing (ISIP), 2008 international symposiums on. IEEE
14. Lian S (2009) Efficient image or video encryption based on spatiotemporal chaos system. Chaos Solitons Fract 40(5):2509–2519
15. Hermassi H, Rhouma R, Belghith S (2009) A modified hyperchaos based image cryptosystem. In: Systems, signals and devices, 2009. SSD'09. 6th international multi-conference on. IEEE
16. Honge R et al (2007) A chaotic algorithm of image encryption based on dispersion sampling. In: Electronic measurement and instruments. ICEMI'07. 8th International conference on. IEEE
17. Hong-e R et al (2007) Block sampling algorithm of image encryption based on chaotic scrambling. In: Computational intelligence and security workshops, 2007. CISW 2007. International conference on. IEEE
18. Adleman LM (1994) Molecular computation of solutions to combinatorial problems. Nature 369:40
19. Xie T, Liu Y, Tang J (2014) Breaking a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Opt-Int J Light Electron Opt 125(24):7166–7169
20. Zhang Q, Guo L, Wei X (2010) Image encryption using DNA addition combining with chaotic maps. Math Comput Model 52(11):2028–2035
21. Wang X, Zhang H, Bao X (2016) Color image encryption scheme using CML and DNA sequence operations. Biosystems 144:18–26
22. Enayatifar R, Abdullah AH, Lee M (2013) A weighted discrete imperialist competitive algorithm (WDICA) combined with chaotic map for image encryption. Opt Lasers Eng 51(9):1066–1077
23. Suri S, Vijay R (2017) A Bi-objective genetic algorithm optimization of chaos-DNA based hybrid approach. J Intell Syst. Retrieved 23 Aug 2017, from https://doi.org/10.1515/jisys-2017-0069
24. Enayatifar R, Abdullah AH, Isnin IF (2014) Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Opt Lasers Eng 56:83–93
25. Ngatchou P, Zarei A, El-Sharkawi A (2005) Pareto multi objective optimization. In: Intelligent systems application to power systems, 2005. Proceedings of the 13th international conference on. IEEE
26. Kaneko K (1992) Overview of coupled map lattices. Chaos: Interdiscip J Nonlinear Sci 2(3):279–282
27. Kaneko Kunihiko (1989) Spatiotemporal chaos in one-and two-dimensional coupled map lattices. Physica D 37(1-3):60–82
28. Ge X et al (2011) Cryptanalysis of a spatiotemporal chaotic image/video cryptosystem and its improved version. Phys Lett A 375(5):908–913
29. Xing-Yuan W, Xue-Mei B (2013) A novel image block cryptosystem based on a spatiotemporal chaotic system and a chaotic neural network. Chin Phys B 22(5):050508
30. Xing-Yuan W, Lin-Tao L (2013) Cryptanalysis and improvement of a digital image encryption method with chaotic map lattices. Chin Phys B 22(5):050503
31. Li P et al (2006) A multiple pseudorandom-bit generator based on a spatiotemporal chaotic map. Phys Lett A 349(6):467–473
32. Pareek NK, Patidar V, Sud KK (2006) Image encryption using chaotic logistic map. Image Vis Comput 24(9):926–934
33. Gao H et al (2006) A new chaotic algorithm for image encryption. Chaos Solitons Fract 29(2):393–399
34. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurc Chaos 8(06):1259–1284
35. Akhshani A et al (2010) A novel scheme for image encryption based on 2D piecewise chaotic maps. Opt Commun 283(17):3259–3266
36. Liu H, Wang X (2011) Color image encryption using spatial bit-level permutation and high-dimension chaotic system. Opt Commun 284(16):3895–3903
37. Chen G, Mao Y, Chui CK (2004) A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fract 21(3):749–761
38. Song C-Y, Qiao Y-L, Zhang X-Z (2013) An image encryption scheme based on new spatiotemporal chaos. Opt-Int J Light Electron Opt 124(18):3329–3334
39. Ahadpour S, Sadra Y (2012) A chaos-based image encryption scheme using chaotic coupled map lattices. arXiv preprint arXiv:1211.0090
40. Wu X (2013) A novel chaos-based image encryption scheme using coupled map lattices. In: Fuzzy systems and knowledge discovery (FSKD), 2013 10th international conference on. IEEE
41. Wang S et al (2004) Periodicity of chaotic trajectories in realizations of finite computer precisions and its implication in chaos communications. Int J Mod Phys B 18(17–19):2617–2622
42. Kocarev L (2001) Chaos-based cryptography: a brief overview. IEEE Circuits Syst Mag 1(3):6–21
43. Li C et al (2009) Cryptanalysis of an image encryption scheme based on a compound chaotic sequence. Image Vis Comput 27(8):1035–1039
44. Rhouma R, Solak E, Belghith S (2010) Cryptanalysis of a new substitution–diffusion based image cipher. Commun Nonlinear Sci Numer Simul 15(7):1887–1892
45. Yoon E-J et al (2011) Cryptanalysis of an enhanced spatiotemporal chaotic image/video cryptosystem. EURASIP J Adv Signal Process 1:461563
46. Wang Y et al (2011) A new chaos-based fast image encryption algorithm. Appl Soft Comput 11(1):514–522
47. Ming Z, Shudong S (1999) Theory of genetic algorithm and its application. National Defense Industry Publishing Company, Beijing, pp 125–127
48. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3(4):257–271
49. Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: a tutorial. Reliab Eng Syst Saf 91(9):992–1007
50. Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulationdiscussion and generalization. Icga 93(July):416–423
51. Murillo-Escobar MA et al (2017) A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. Nonlinear Dyn 87(1):407–425
52. Murillo-Escobar MA et al (2015) A RGB image encryption algorithm based on total plain image characteristics and chaos. Sig Process 109:119–131

53. Murillo-Escobar MA, et al (2014) A novel symmetric text encryption algorithm based on logistic map. In: Proceedings of the international conference on communications, signal processing and computers (ICNC'14)
54. Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurc Chaos 16(08):2129–2151
55. Fu C et al (2011) A novel chaos-based bit-level permutation scheme for digital image encryption. Opt Commun 284(23):5415–5423