J Opt
https://doi.org/10.1007/s12596-023-01639-3

RESEARCH ARTICLE

Image encryption using a 3D modular hybrid chaotic system and CRC key generator

A. Yousefian Darani[1] · Y. Khedmati Yengejeh[1] · R. Hosseinzadeh[1] · H. Pakmanesh[1]

Received: 22 October 2023 / Accepted: 23 December 2023

Abstract With the development of social networks, the demand for information security is increasing day by day, and as images are one of the most commonly used formats, this paper introduces a novel method for encrypting images using a new 3D modular hybrid chaotic map with two control parameters. The significance of chaotic maps in secure information transformation algorithms lies in the sensitivity of digital images to minor pixel alterations and the high degree of similarity between neighboring pixels. Another aspect of cryptographic algorithms is the creation of a key space that is sensitive to input images. As such, a primary objective in this procedure is to incorporate a cyclic redundancy check and the discrete framelet transform to generate an optimal key space. Finally, aforementioned tools lead to the implementation of a secure encryption algorithm based on cellular automata and simulation results severely demonstrate robustness and resistance of the proposed scheme against data loss attacks.

Keywords Cryptography · Image · Chaotic system · Cellular automata · CRC

✉ A. Yousefian Darani
youseffian@gmail.com; yousefian@uma.ac.ir

Y. Khedmati Yengejeh
khedmati.y@uma.ac.ir

R. Hosseinzadeh
r.hosseinzadeh@uma.ac.ir

H. Pakmanesh
pakmanesh_hosein@yahoo.com

[1] Department of Mathematics, University of Mohaghegh Ardabili, 56199-11367 Ardabil, Iran

Mathematics Subject Classification 94A60 · 37D45 · 37B15 · 94Bxx

Introduction

Image security is a crucial research area in information security, and cryptography is one of the most effective methods available in this field [1]. In innovative encryption techniques, secret images are transformed to meaningless noise-like images. Chaos-based image encryption methods have become one of the most ideal methods [2–4]. Chaos systems need to be evaluated by some severe tests, such as Lyapunov exponent, histogram and 0–1 test. The 0–1 test for chaos, which is a fundamental issue in the theory of chaotic dynamics to determine between regular and chaotic dynamics, was proposed by Gottwald and Melbourne [5–7]. Therefore, along with other methods of evaluating chaotic maps, we use these tests to confirm the chaotic behavior of the introduced 3D modular hybrid chaotic map with two control parameters.

A cyclic redundancy check (CRC) is one of the error detection techniques which generates some redundant bits called CRC to detect accidental changes to raw data in digital data and storage devices [8]. In order to produce the appropriate key space for the proposed algorithm, we find CRCs for different data blocks with different divisors.

Von Neumann was the first to introduce cellular automata (CA) as a tool for studying biological processes [9]. These automata belong to a class of discrete dynamical systems and possess unique characteristics such as linearity, reversibility and boundary conditions. The vast array of CA evolution rules allows for an extensive range of possibilities in generating sequences of CA data security. Additionally, one-dimensional CA substitutions solely comprise straightforward

Published online: 27 March 2024

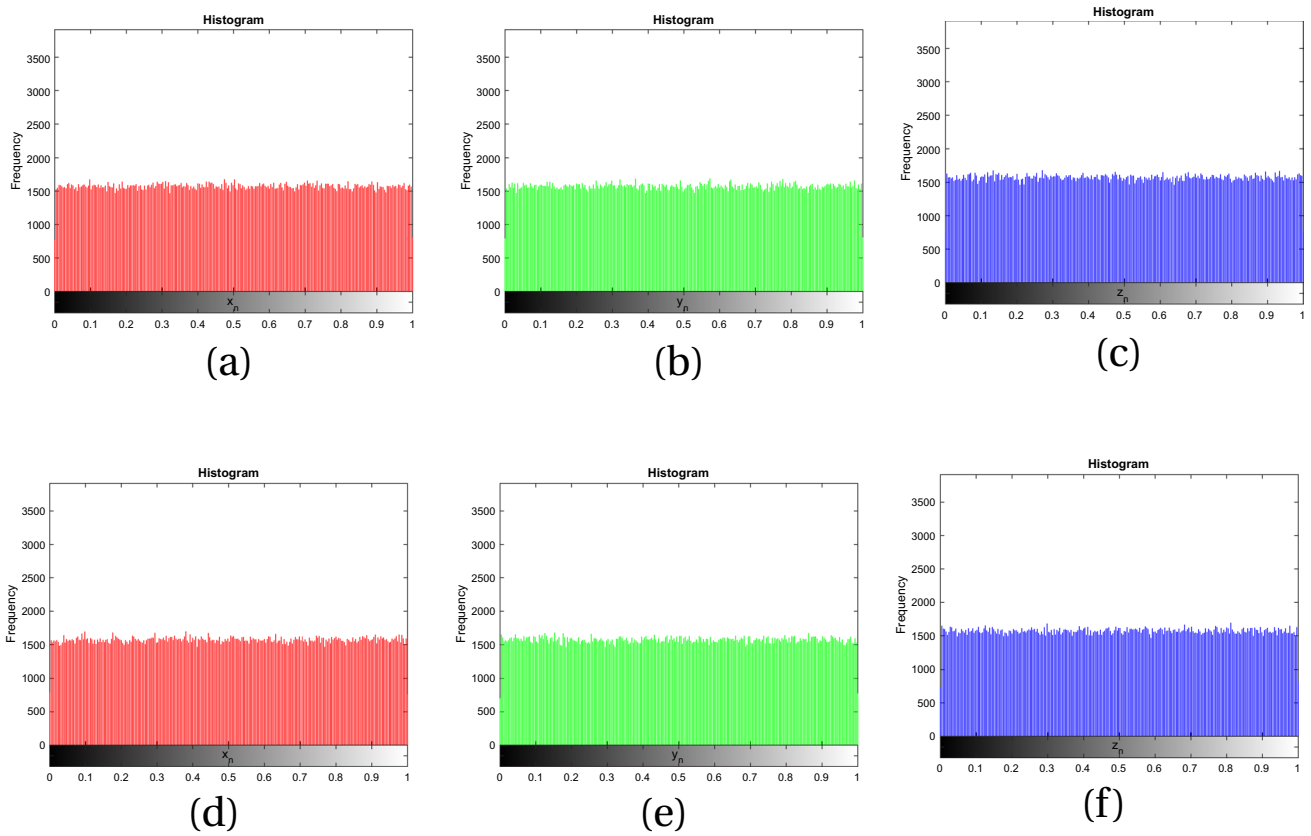

Fig. 1 Histogram for proposed 3D hybrid chaos map. **a–c** Histograms of C3D with $r = 1.85$ for x_n, y_n and z_n, respectively, **d–f** histograms of GC3D with $(r_1, r_2) = (1.05, 0.65)$ for x_n, y_n and z_n, respectively

and uncomplicated logic operations, making them easy to compute. Cryptographic techniques are typically more efficient when using CA because of its parallelism, regular and straightforward structure and uncomplicated hardware design. Therefore, they can be utilized in the development of encryption algorithms for image encryption. In recent years, many researchers have used cellular automata in their presented algorithm for image encryption [10–13]. In [10], a technique for reconstructing the produced sequence is introduced, which exploits the linearity of the CA-based model. In [11], a unique type of cellular automata with periodic boundaries and unity attractors is employed. To ensure security, the number of attractor states in the cellular automata is altered based on the encrypted image, and distinct key streams are utilized to encrypt various plain images. Discrete transforms provide a highly efficient method for compressing and decompressing images while preserving their quality and offer a high degree of security. Discrete framelet transform provides a more flexible and adaptable approach to image processing and allows for greater control over the compression and encryption process. Additionally, discrete frame transforms can be used to create more complex encryption schemes and can provide even greater security for sensitive data. Khedmati et al. [12] presented an image security transform algorithm based on cellular automata, discrete framelet transform and chaotic system. In [13], an image encryption technique is introduced, which utilizes 2D Moore Cellular Automata to ensure both security and efficiency. This method is lossless and can be easily implemented in hardware. To enhance the security and resilience of our proposed method, we incorporate cellular automata. This involves considering the pixels of the original image as states at different times sequentially. Additionally, we utilize the outputs of the chaos system at the outset of this process.

In this article, a robust and resistive image encryption algorithm against noise and data loss attacks is presented. This algorithm due to utilizing the proposed chaos map and designed cellular automata has high sensitivity to initial conditions, uncertainty, high complex behavior and so on. The main advantages of this work can be described as follows:

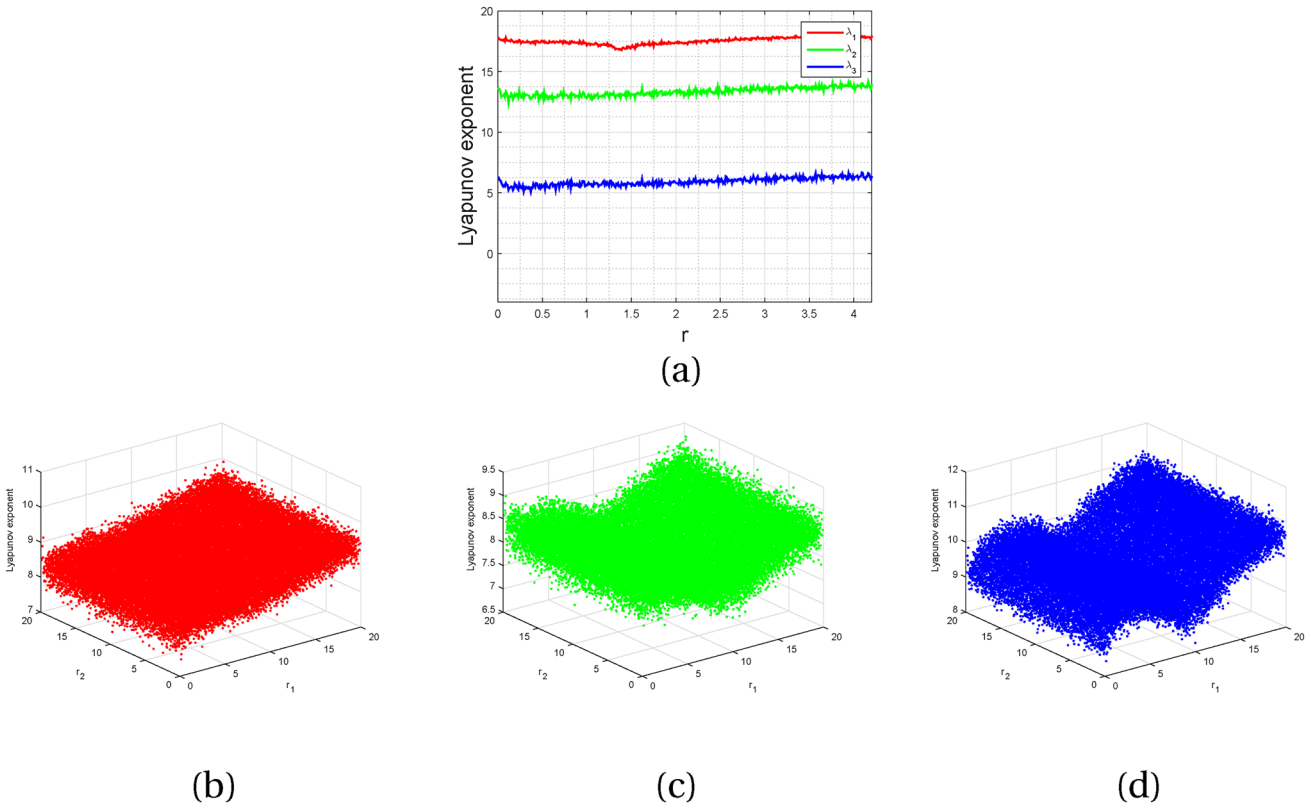

Fig. 2 Lyapunov exponent plot for proposed 3D hybrid chaos maps. **a** For C3D, **b–d** for GC3D

- We derive high-dimensional chaotic maps to have larger chaotic space.
- We create a strong key space based on the proposed chaos map and CRC error detection method.
- We designed a special shift operator to shuffle pixels of images, which results in the correlation reduction between pixels.
- We utilize cellular automata to make the image encryption method more efficient.

The layout of this paper is as follows. In Sect. 2, we introduce a three-dimensional hybrid chaos map and then this map is generalized to a map with two control parameters. Section 3 delves into the detail of the designed shift function. Generation of CRC key and its all aspects are described in Sect. 4. Section 5 gives a brief description of new cellular automata used in the new encryption algorithm whose construction is presented in Sect. 6. Finally, numerical performance can be found in Sect. 7 to demonstrate efficiency of the proposed method.

3D Modular hybrid chaotic maps

Sensitive dependence on initial conditions is a key characteristic of chaos, making chaotic systems valuable for use in the development of cryptographic schemes. High-dimensional chaotic maps offer a larger chaotic space, further enhancing their utility in this regard. In this section, we first introduce the three-dimensional hybrid chaos map C3D, then we generalize this map to a map with two control parameters GC3D. Sequences of this chaotic map are extracted and used to construct the proposed key space and to present a new encryption algorithm. C3D system with inputs x_0, y_0 and z_0 is defined as follows:

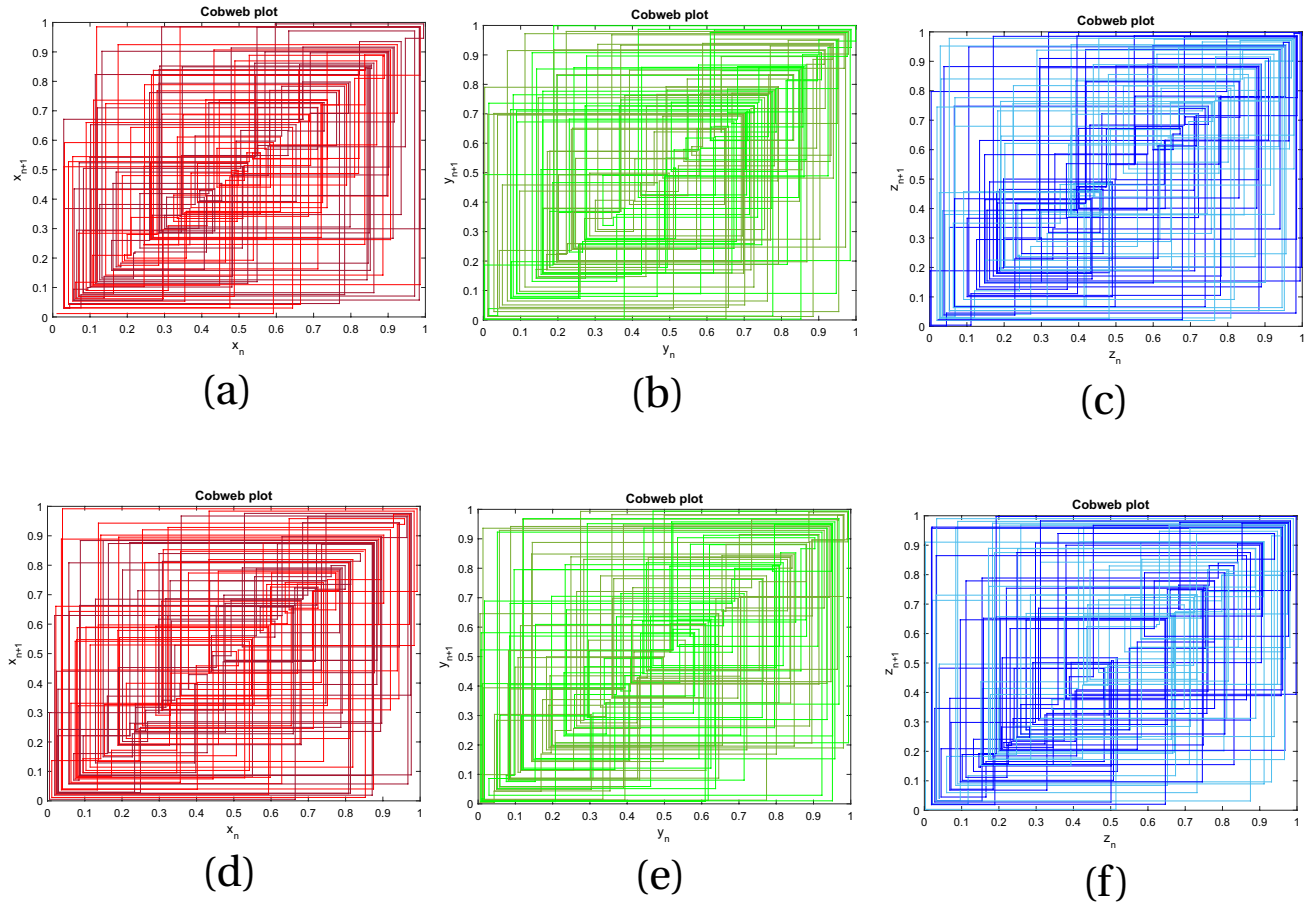

Fig. 3 Cobweb plot for proposed 3D hybrid chaos maps. **a–c** For C3D, **d–f** for GC3D

$$
\begin{aligned}
x_{n+1} &= E(x_n, y_n, z_n) \\
&\times \begin{cases} rx_n(1-x_n) + 2\sin(rx_n y_n) \\ \quad +2\cos(20rz_n) + (4-r)z_n, & \mod 1;\ z_n < x_n, \\ rx_n(1-x_n) + 2\sin(\pi r x_n) \\ \quad +\sin(x_n z_n) + 2(9-r)(1-z_n), & \mod 1;\ z_n \geq x_n, \end{cases} \\
y_{n+1} &= E(x_n, y_n, z_n) \\
&\times \begin{cases} ry_n(1-y_n) + 2\cos(rx_{n+1} + y_n) \\ \quad +\exp(y_n + z_n^2) + (4-r)x_{n+1}, & \mod 1;\ x_{n+1} < y_n, \\ \coth\left(ry_n(1-y_n)\right) + z_n^2 + 2\coth(\pi r y_n) \\ \quad +(20-r)^2(1-x_{n+1})^2, & \mod 1;\ x_{n+1} \geq y_n, \end{cases} \\
z_{n+1} &= E(x_n, y_n, z_n) \\
&\times \begin{cases} \sinh\left(rz_n(1-z_n)\right) + 2\cos(rz_n + y_{n+1} + z_n) \\ \quad +(4-r)y_{n+1}, & \mod 1;\ y_{n+1} < z_n, \\ rz_n(1-z_n) + 2\tan(ry_{n+1}) + \exp(2rx_n y_{n+1}) \\ \quad +\cos\left(2(7-r)(1-y_{n+1})\right), & \mod 1;\ y_{n+1} \geq z_n, \end{cases}
\end{aligned} \tag{1}
$$

in which $E(x, y, z) = \exp\left((0.1 + xyz)^{-1}\right)$. To generalize this map to GC3D with two control parameters r_1, rR_2, we replace r with r_1, $\frac{r_1+r_2}{2}$ and r_2 in the sub-functions related to x_{n+1}, y_{n+1} and z_{n+1}, respectively.

If the dynamical system is chaotic, then two trajectories starting very close together will rapidly diverge from each other. There are some ways of measuring chaos of a dynamical system which we will discuss in the following. A histogram is a graphical representation of the distribution of numerical data. It is a bar graph which height of each bar shows how many are in each value. Smooth histograms are one of the most important features of ideal chaotic maps. Figure 1 shows the uniform distribution of proposed 3D hybrid chaotic maps. Lyapunov exponent (LE) is a fundamental instrument of the chaos theory which measures rate of separation or convergence of infinitesimally close trajectories. LE of discrete-time system $x_{i+1} = f(x_i)$ can be calculated as:

$$
\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|. \tag{2}
$$

In Fig. 2, 501 and 4000 points are generated by C3D and GC3D, respectively, are used. Positive λ means the dynamical system has chaotic behavior and larger λ proves greater sensitivity, i.e., the faster two nearby trajectories pulled apart. On the other hand, negative λ means the dynamical system is either periodic or stable.

A cobweb plot is a particularly effective way to visualize the dynamics. Figure 3 depicts 100 points obtained by C3D and GC3D.

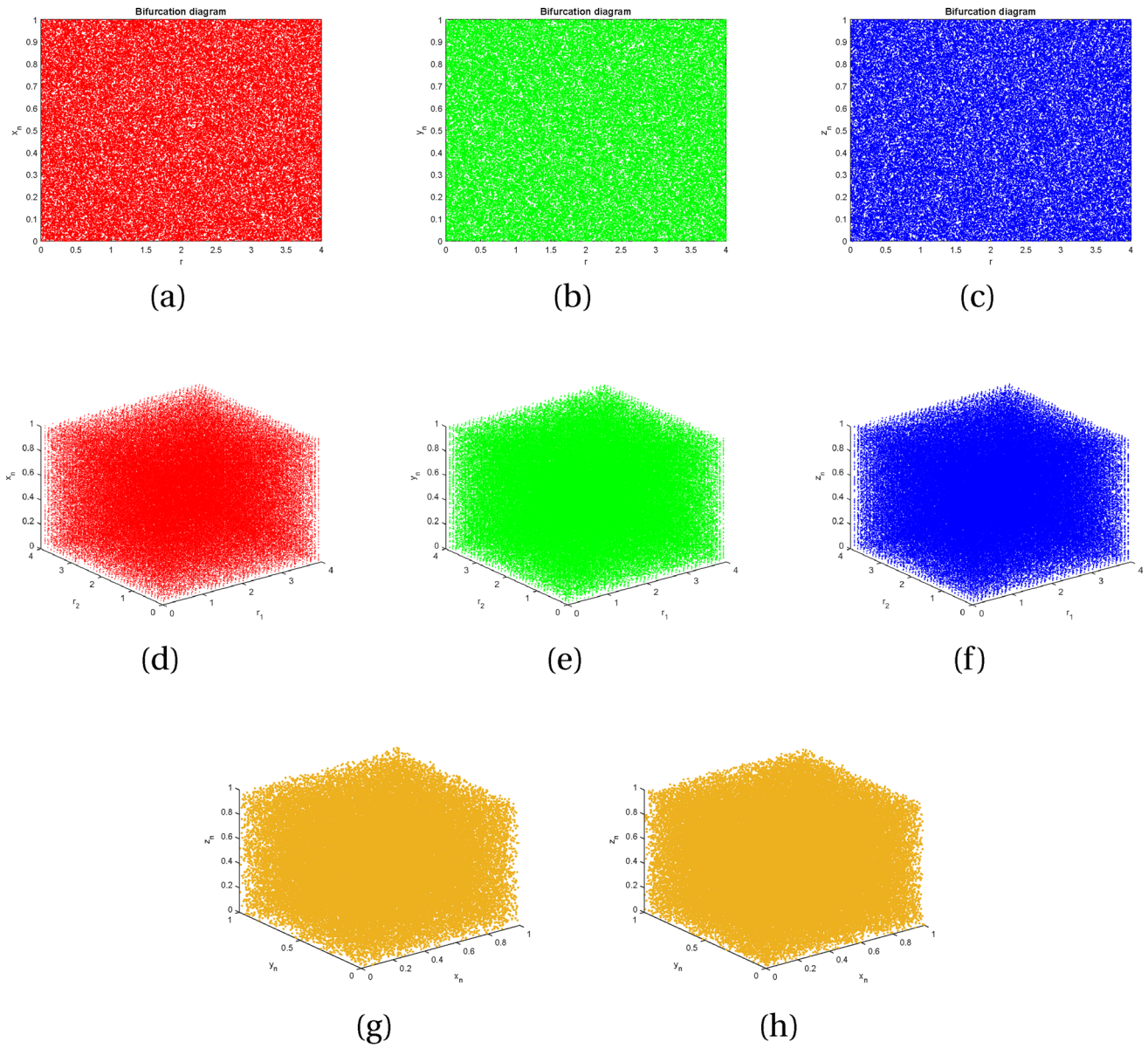

Fig. 4 Bifurcation diagram for proposed 3D hybrid chaos maps s and their trajectory. **a–c** Bifurcation diagram for C3D, **d–f** bifurcation diagram for GC3D, **g** trajectory of C3D under $r = 1.85$, **h** trajectory of GC3D under $(r_1, r_2) = (1.05, 0.65)$

The bifurcation diagram in Fig. 4 illustrates the values obtained by the map for various control parameters of C3D and GC3D. This diagram also analyzes the chaotic behavior of a nonlinear system. Also, the trajectories (x_i, y_i, z_i) are plotted by setting the parameters at fixed values in Fig. 4g, h.

Recently, the 0–1 test is also used to determine the regular from chaotic dynamics in dynamics systems, in which the input of the test is a one-dimensional time series $\phi(n)$ for $n \in \mathbb{N}$. In contrast to *LE*, this test is applied directly to the time series data and does not require phase space reconstruction. This test equips a 2D system as follow.

$$\begin{aligned} p(n+1) &= p(n) + \phi(n+1)\cos(cn), \\ q(n+1) &= q(n) + \phi(n+1)\sin(cn), \end{aligned} \tag{3.1}$$

where $c \in (0, 2\pi)$ is fixed. The mean square displacement of this 2D system is given as follow.

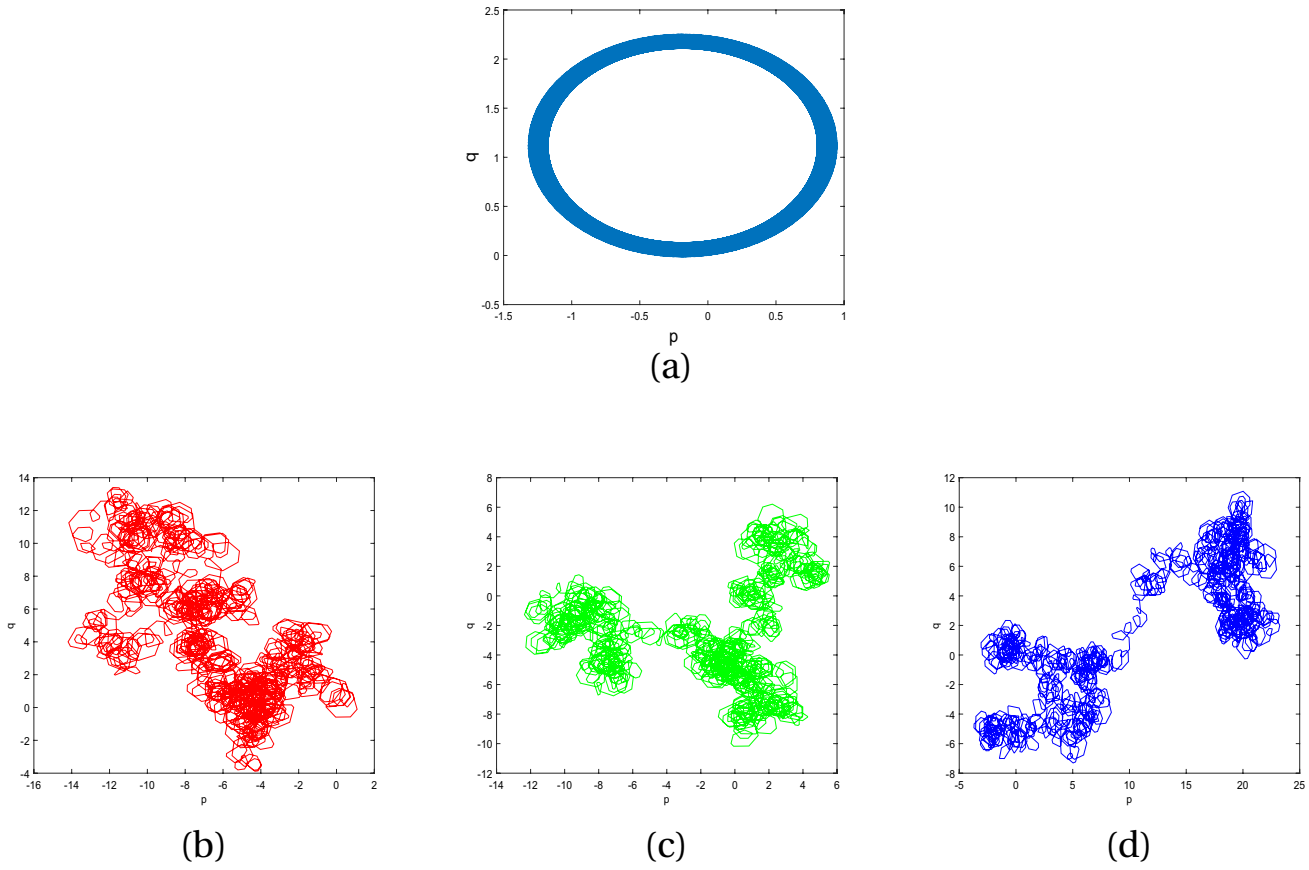

Fig. 5 Plot of p versus q for **a** regular dynamics logistic map at $r = 3.55$, **b–d** chaotic dynamics for GC3D at $(r_1, r_2) = (3.97, 1.56)$. We use 3000 data points

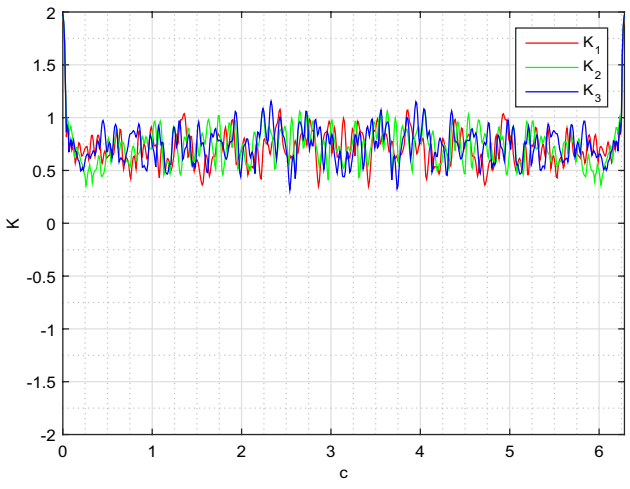

Fig. 6 Plot of c versus K at $(r_1, r_2) = (3.97, 1.56)$, corresponding to GC3D. We use 629 data points

$$M(n) = \lim_{n\to\infty} \frac{1}{N} \sum_{j=1}^{N} \Big([p(j+n) - p(j)]^2 + [q(j+n) - q(j)]^2 \Big), \quad n = 1, 2, \ldots, \tag{3.2}$$

and the growth of this 2D system is given by

$$K = \lim_{n\to\infty} \frac{\log\big(M(n)\big)}{\log(n)}. \tag{3.3}$$

The dynamics of $p(n)$ and $q(n)$ are bounded when the time series $\phi(n)$, represent regular dynamics. In this case, $M(n)$ is bounded as well and return $K = 0$. For the chaotic time series, we have $K = 1$. As we know, the logistic map $x_{n+1} = rx_n(1 - x_n)$ is not chaos for $r = 3.55$, so to better understand, the diagram related to this mapping is also given in Fig. 5d. Figure 5 shows the difference between regular and chaotic dynamics.

Plot of c versus K is given in Fig. 6.

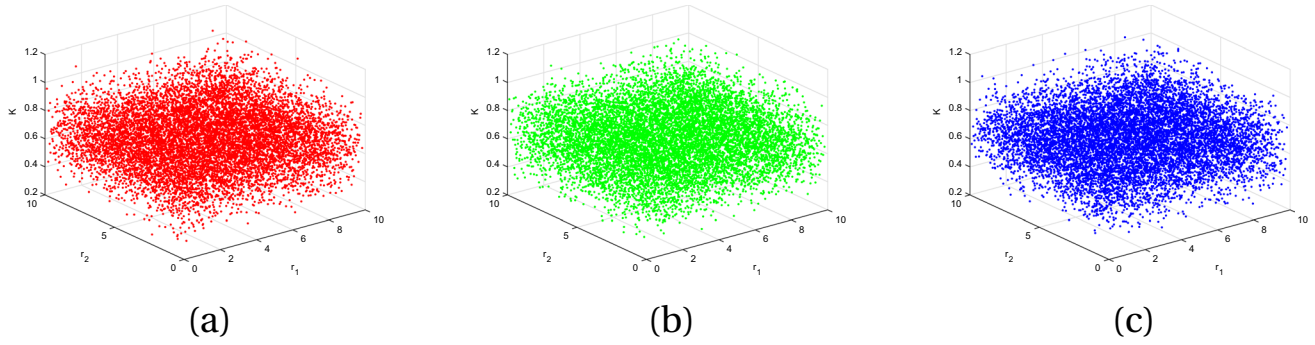

Fig. 7 Plot of r_1, r_2 versus K for GC3D for $0 \leq r_1, r_2 \leq 10$. We use 10,000 data points

In Fig. 7, we see the value of K for different values of r_1 and r_2.

Implementation of the 0–1 test for chaos is described in [7, 14]. For more on chaos test, we refer to [14].

Outputs of GC3DV $(x_0, y_0, z_0, r_1, r_2, n, k)$ are vectors X, Y, Z of length n which are obtained by repeating GC3D in $n-1$ steps and considering the first k decimal digits of outputs.

Shift functions

In this section, we introduce a shift function that operates based on vectors to shuffle pixels of images. First of all, we define some basic shift functions, and then we compose these functions to obtain the final one. In the continuation of this section $A, B, C \in \mathbb{R}^{n\times n}$, $V_1, V_2, V_3, V_5, V_6, V_7 \in \mathbb{Z}^{1\times(2n-3)}$ and $V_0, V_4 \in \mathbb{Z}^{1\times n}$. To obtain the output matrix DiagSh1D (A, V_1) diameters diag (A, i) are shifted $V_1(n-i-1)$ units, independently, where $i = -n+2 : n-2$.

for $i = -n+2 : n+2$ **do**
$O =$ circshift $\big($ diag $(A, i), V_1(n-i-1)\big)$;

SubDiagSh1D shift function is defined similarly, this time the shift operation is performed on sub-diameters. Intending to better understand these functions, we have applied them to a four-dimensional matrix.

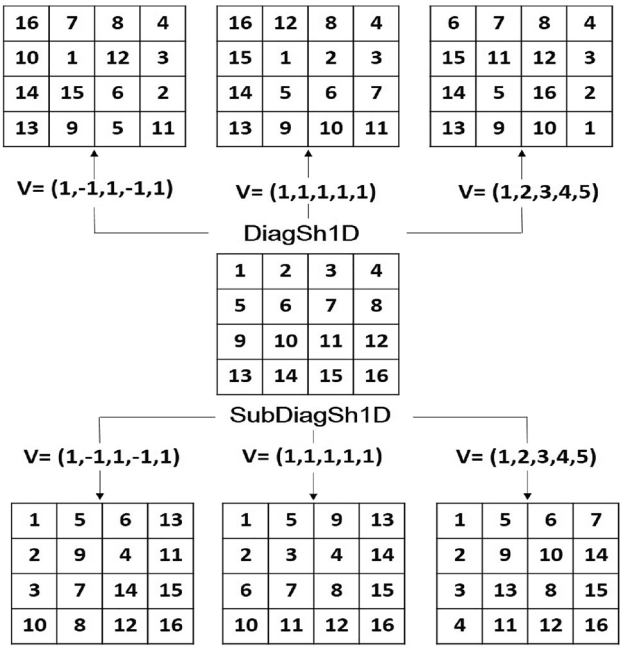

Fig. 8 Output image after applying the ColDiagRowSh3D shift function on the input image: **a** input image, **b** output image

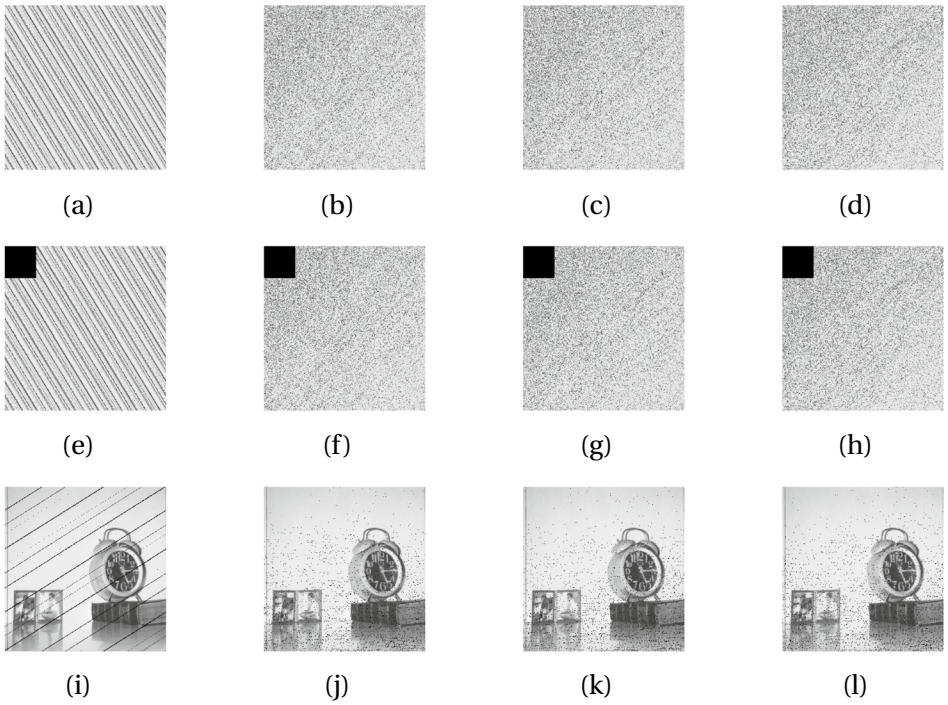

Fig. 9 Results for data loss attacks: **a** output image after applying ÙžArnolad transformation; **b–d** output images after applying the ColDiagRowSh3D; **e–h** cropping 50×50 pixels of images; **i–l** decrypted images corresponding the loss attacks

Now, we introduce the simple shift functions RowSh3D and ColSh3D, which are suitable for shuffling the pixels of color images. To obtain the output

$$[A_1, B_1, C_1] = \text{RowSh3D}\,(A, B, C, V_0), \tag{4}$$

the i-th row of the matrix $[A, B, C] \in \mathbb{R}^{n\times(3n)}$ is shifted $V_0(i)$ units, separately, and finally, the resulting matrix is divided into three equal parts to obtain $A_1, B_1, C_1 \in \mathbb{R}^{n\times n}$. ColSh3D shift function is defined similarly, this time the shift operation is implemented on columns of the matrix $[A;B;C] \in \mathbb{R}^{3n\times(n)}$. It is time to open up the main shift function ColDiagRowSh3D by using all the shift functions defined above, as follows.

$[A_1, B_1, C_1] = \text{RowSh3D}\,(A, B, C, V_0)$;
$A_2 = \text{DiagSh1D}\,(A_1, V_1)$;
$B_2 = \text{DiagSh1D}\,(B_1, V_2)$;
$C_2 = \text{DiagSh1D}\,(C_1, V_3)$;
$[A_3, B_3, C_3] = \text{ColSh3D}\,(A_2, B_2, C_2, V_4)$;
$A_4 = \text{SubDiagSh1D}\,(A_3, V_5)$;
$B_4 = \text{SubDiagSh1D}\,(B_3, V_6)$;
$C_4 = \text{SubDiagSh1D}\,(C_3, V_7)$;

Figure 8 shows the effect of this function on a color image, in which the input vectors components are between -200 and 200.

In order to investigate the performance of the proposed shift function and Arnold transformation, Fig. 8 is presented for the cropping attack. The Arnold transformation is defined as Function 1, and it is worth noting that the Fig. 9a is obtained after applying this transfer four times consecutively. Considering that our introduced shift function is three-dimensional, the corresponding image is considered three times as inputs of the function.

Input: A matrix $I \in \mathbb{R}^{n\times n}$
Output: A matrix $S \in \mathbb{R}^{n\times n}$
1: **for** $i = 1 : n$ **do**
2: **for** $i = n$ **do**
3: $x = \text{mod}(2i + j, n) + 1$;
4: $y = \text{mod}(i + j, n) + 1$;
5: $S(x, y) = I(i, j)$;
6: **return** the matrix S

Function 1 ArnoldTransformation

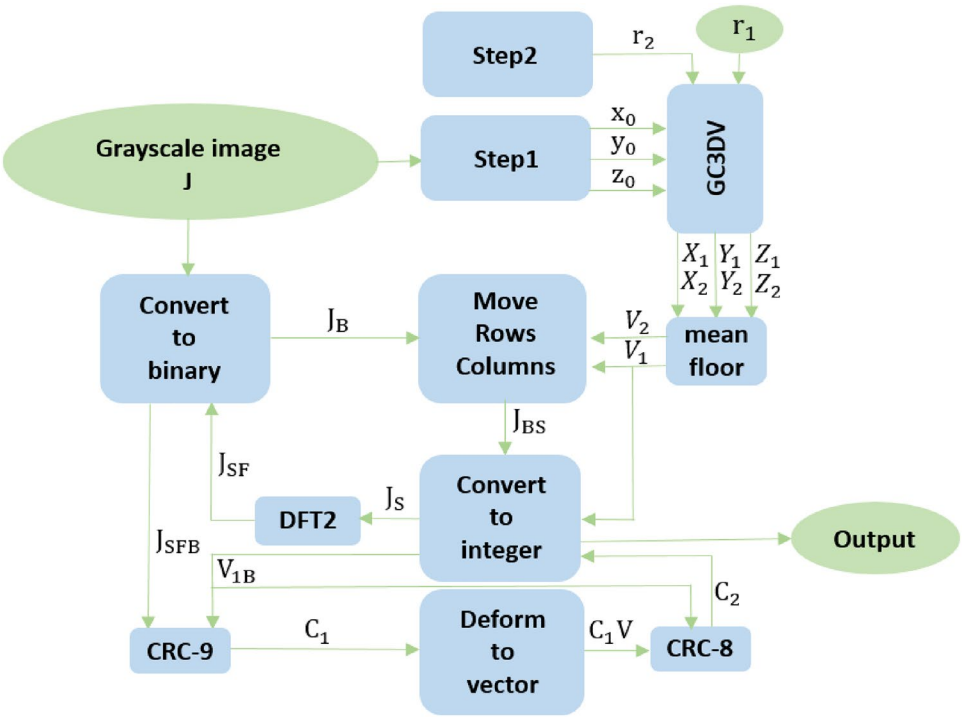

Fig. 10 Key generation for grayscale image

CRC as a key generator

The key space is a critical component of any encryption system, as it directly influences the level of security and ability to withstand attacks. In this section, we introduce functions named KS and KS3D to create a strong key space generator for the proposed algorithm. These functions are defined based on GC3D hybrid chaos map and CRC error detection method. In CRC error-detecting code, some bits are appended to blocks of data based on the remainder of a polynomial division of their contents. The algorithm is based on cyclic codes, and CRCs are easy to implement and to analyze mathematically. In practice, all commonly used CRCs employ the finite field of two elements, simply. A CRC is called an n-bit CRC (CRC-n) when its check value is n bits long.

The values of our key space, in addition to the initial input, also depend on secret images and the execution of the program. In a way, different outputs are obtained either by changing the secret image or by running algorithms for intended image. First, we introduce KS function, whose inputs are grayscale image $J \in \mathbb{R}^{n\times m}$ and control parameter $r_1 \in \mathbb{R}$, and its output is a number that is made during the following steps. As it can be seen, in the process of defining this function, CRC has been used $n+1$ times with different divisors, in which data blocks are initially binary formats of secret image rows and divisors are rows of binary matrix resulting from GC3D (see Fig. 10). The first two steps are related to the construction of vector $V \in \mathbb{R}^{n\times 1}$ based on generalized 3D modular hybrid chaos map GC3D.

Table 1 Key space generated by the KS for various images and running

Inputs		Output
J	r_1	
		146
Lena (256×256)	2.55	128
		57
		78
5.1.12 (256×256)	4.00	158
		20
		220
Boat (512×512)	0.50	172
		198
		28
5.2.08 (512×512)	1.37	83
		127

Step 1. Random selection of three numbers named x_0, y_0, z_0 from pixels double (J).

Step 2. Random production of the control parameter r_2.

Step 3. Constructing vectors V_1 and V_2 by applying GC3D.

$$[X_1, Y_1, Z_1] = \text{GC3DV}\,(x_0, y_0, z_0, r_1, r_2, n, 2);$$
$$[X_2, Y_2, Z_2] = \text{GC3DV}\,(2x_0, 2y_0, 2z_0, 2r_1, 2r_2, 8m, 2);$$
$$V_1 = \text{floor}\,\big(\text{mean}\,(X1, Y1, Z1)\big);$$
$$V_2 = \text{floor}\,\big(\text{mean}\,(X2, Y2, Z2)\big);$$

Step 4. Convert J to the binary format $J_B \in \mathbb{R}^{n\times 8m}$.

Step 5. Shuffling the rows and columns of J_B based on V_1 and V_2, respectively, to get J_{BS}.

Step 6. Converting J_{SB} to matrix $J_S \in \mathbb{R}^{n\times m}$ with integer array.

Step 7. Reducing the size of J_S to one-ninth of its size by using the DFT2 function.

$$J_{SF} = \big(\text{DFT2}\,(J_S)\big)\Big(1 : \tfrac{n}{3}, 1 : \tfrac{n}{3}\Big);$$

Step 8. Converting J_{SF} and V_1 to binary matrices and showing by J_{SFB} and V_{1B}, respectively.

Step 9. Finding the CRC for the data blocks $J_{SFB}(i, :)$ with the divisors

$$x^9 + \sum_{j=1}^{8} V_{1B}(i,j)x^{9-j} + 1,$$

to make a cell matrix with binary cells $C_1 \in (\mathbb{R}^{1\times 9})^{n\times 1}$ for $i = 1, 2, \ldots, n$.

for $i = 1 : n$ **do**

$$C_1(i, :) = \text{CRC}\,\Big(J_{SFB}(i, :), x^9 + \textstyle\sum_{j=1}^{8} V_{1B}(i,j)x^{9-j} + 1\Big);$$

Step 10. Deforming C_1 to binary vector $C_1 V \in \mathbb{R}^{1\times 9n}$.

Step 11. Random selection of one row of V_{1B} named l and finding $C_2 \in \mathbb{R}^{1\times 8}$ as follows.

$$C_2 = \text{CRC}\,\Big(C_1 V, x^8 + \textstyle\sum_{j=1}^{7} V_{1B}(l, j+1)x^{9-j} + 1\Big);$$

Step 12. Converting C_2 to integer array.

For secret color image $I \in \mathbb{R}^{n\times m\times 3}$ with channels R,G and B, we define KS3D function with four inputs I, r_1, r_2, r_3 and three outputs as follow.

$$out1 = \text{KS}\,(R, r1);$$
$$out2 = \text{KS}\,(G, r2);$$
$$out3 = \text{KS}\,(B, r3);$$

Results of KS for various inputs are given in Table 1. To show the output sensitivity of KS to small change in secret

Table 2 Key space sensitivity for various images

Inputs			Output
J	r_1		
		One bit is changed	177
Lena (256 × 256)	0.70	Original	202
		One bit is changed	32
		One bit is changed	242
Cameraman (256 × 256)	2.50	Original	145
		One bit is changed	178
		One bit is changed	19
Lena (512 × 512)	1.30	Original	46
		One bit is changed	73
		One bit is changed	66
5.2.08 (512 × 512)	5.50	Original	32
		One bit is changed	167

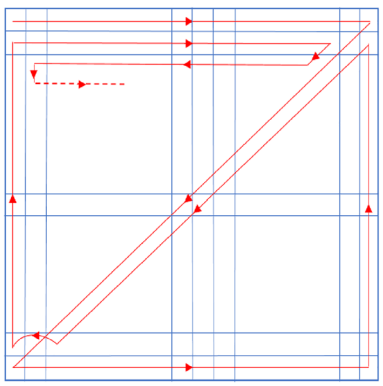

Fig. 12 Matrix to vector conversion path

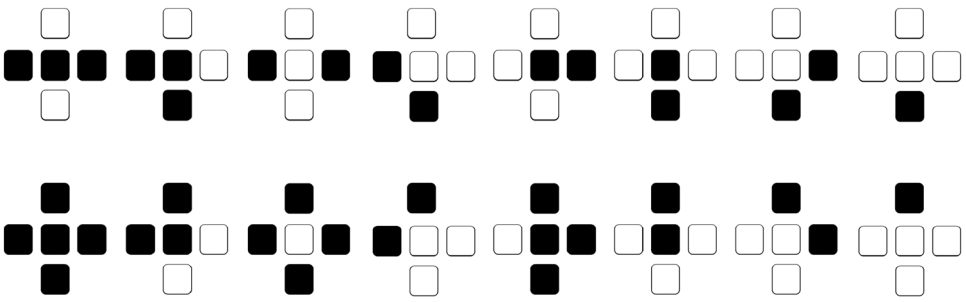

Fig. 11 Reversible cellular automata under Rule 87

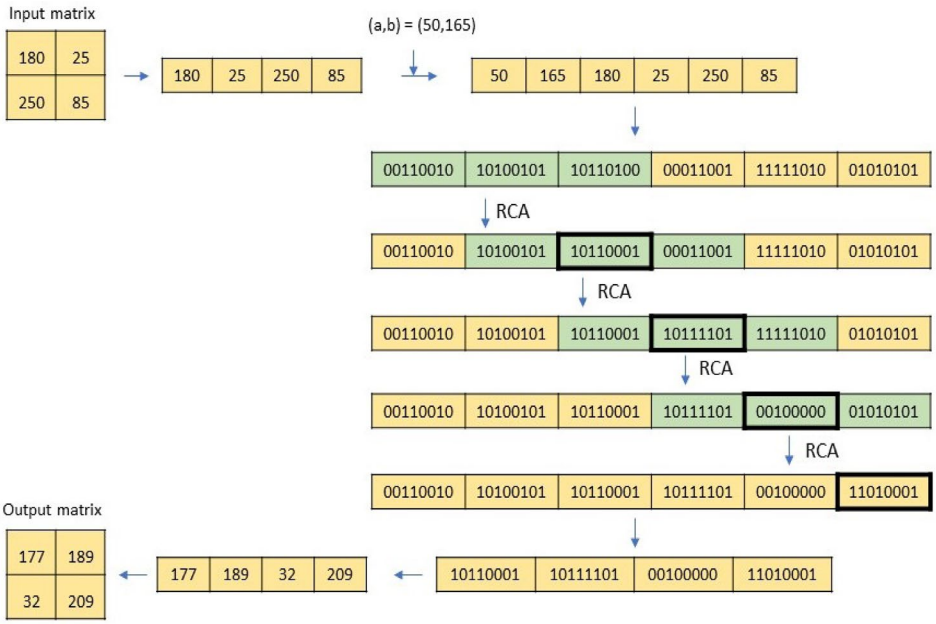

Fig. 13 Illustration of RCA step by step

image J, we keep all the parameters that are generated randomly in the key generation process as

$$(x_0, y_0, z_0, r_2, l) = \Big(\text{double}\,(J)(2,2),\ \text{double}\,(J)(3,3),\ \text{double}\,(J)(4,4), 4, 50 \Big),$$

and then add one unit to one of the pixels of J. The results are shown in Table 2.

Cellular automata

Cellular automata (CA) is a discrete dynamical systems consisting of multiple cells, each with a finite number of states, such as on/off or white/black. These cells are arranged in networks by placing them adjacent to one another. Reversible and irreversible are two main categories of cellular automata. In reversible cellular automata (RCA), every configuration of it has only one preceding configuration, and therefore, its evolutionary process can be traced back uniquely. The second-order cellular automata method is also a type of RCA, where the new state is obtained by combining states from two previous steps of the automata. In simpler terms, to determine the state at time $t + 2$ in a one-dimensional reversible cellular automaton, we combine the current state at time t with the next moment at time $t + 1$. This process involves eight positions in a one-dimensional CA and duplex positions in a one-dimensional RCA, as illustrated in Fig. 11.

In this figure, the top row shows the current state at t, the middle row represents neighborhoods of state $t + 1$, and the bottom row displays the rule used to adjust the state at time $t + 2$ based on the previous two states. In the presented algorithm, we define the RCA function with inputs of a matrix $M \in \mathbb{Z}_{256}^{m\times n}$ and two numbers (a, b) in the range [0, 255]. We use RCA to cipher the matrix M, which is detailed in Function 2. The resulting ciphered matrix is returned as $M_{en} = \text{RCA}\left(M, a, b\right) \in \mathbb{Z}_{256}^{m\times n}$. In RCA, transforming the matrix into a vector is based on the approach outlined in article [15], as illustrated in Fig. 12. The two numbers are added to the first of the vector. The first number serving as the neighborhood and the second number serving as the rule for encrypting the next number. Then, with the second number now serving as the neighborhood and the previously encrypted number serving as the rule for encrypting the next number. The process continues in the same manner until the entire vector has been encrypted. Figure 13 illustrates an example of a two-by-two matrix that has been encrypted using RCA. Additionally, Fig. 14 shows the output obtained by applying the Function 2 with inputs a gray square image of size 512 and numbers (50, 165).

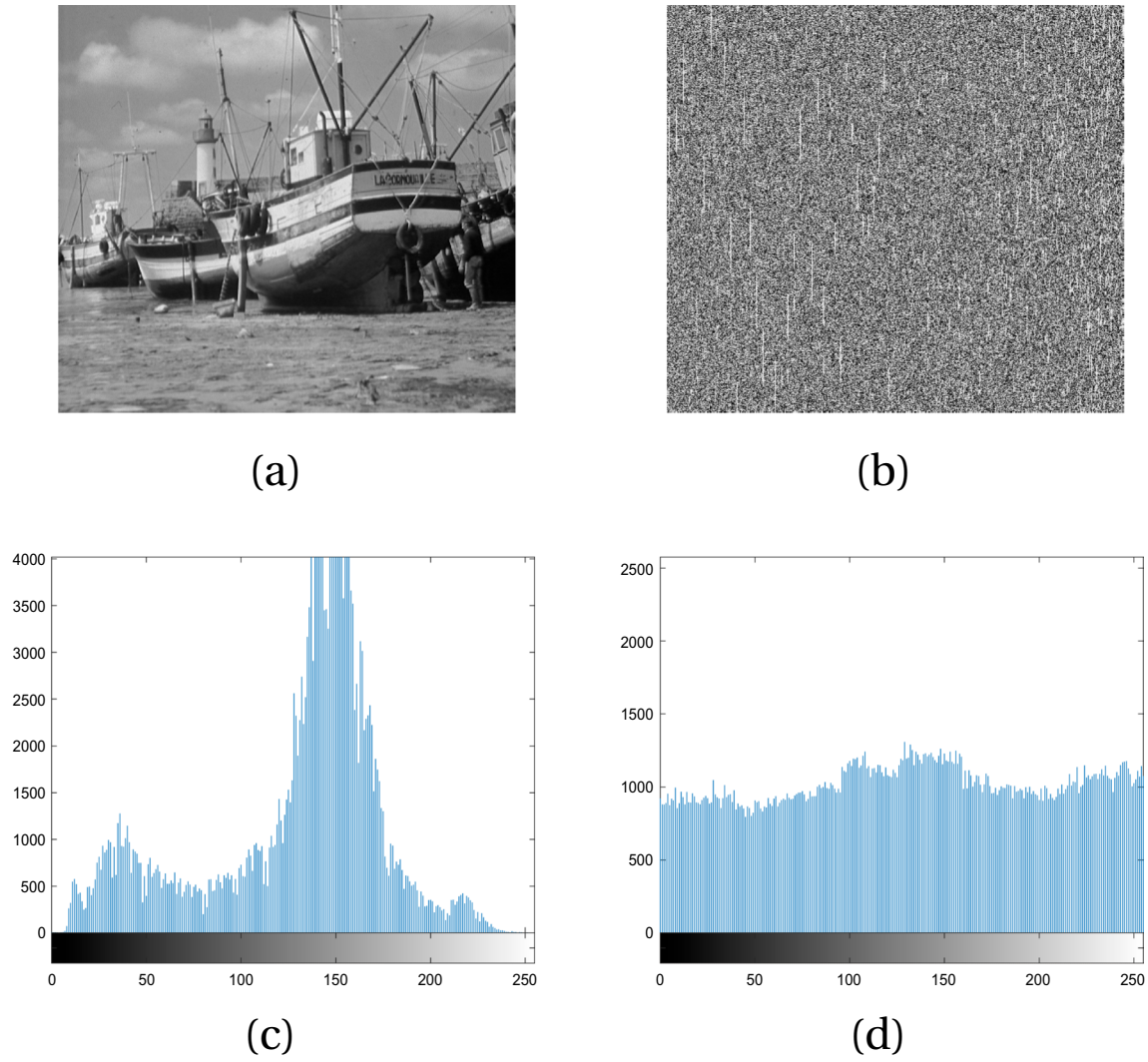

Fig. 14 RCA results: **a** plain image, **b** ciphered image (**a**), **c** histogram of plain image, **d** histogram of ciphered image

Input: A twin $(a, b) \in \mathbb{Z}_{256}^2$ and a matrix $M \in \mathbb{Z}_{256}^{m\times n}$.
Output: A matrix $M_{\text{en}} \in \mathbb{Z}_{256}^{m\times n}$.
1: $M_{vec} \leftarrow \text{MatrixToVector(M, [1, mn])}$
2: $M_v \leftarrow \text{PutFirst(a, b, M}_{\text{vec}})$
3: **for** $i = 1, \ldots, mn+2$ **do**
4: $\quad M_{\text{ca}}(i) \leftarrow \text{ReversibleAutomaton}(M_v(i), M_v(i+1), M_v(i+2))$
5: $M_{\text{en}} \leftarrow \text{VectorToMatrix(M}_{\text{ca}}, \text{[m, n])}$
6: **return** the matrix M_{en}.

Function 2

Fig. 15 Encrypted and decrypted images using the proposed algorithm

Proposed image encryption model

In this section, we leave the detailed steps of the proposed encryption algorithm.

Step 1 . A color image I of size $n \times n$ is divided into three color matrices as

$$[I^r, I^g, I^b] = \text{DivideColors}\,(I)$$

Step 2 . We define X^r, X^g and X^b vectors based on the proposed chaotic map as:

$$[X^r, X^g, X^b] = \text{GC3D}\,(out1, out2, out3, r_1, r_2, N^2 + N), \tag{5}$$

where $out1, out2$ and $out3$ were obtained from KS function (Sect. 4). Since all the ranges of X^r, X^g and X^b are [0, 1], set

$$X^r = \text{floor}\,(10^3.X^r), \quad X^g = \text{floor}\,(10^3.X^g), \quad X^b = \text{floor}\,(10^3.X^b).$$

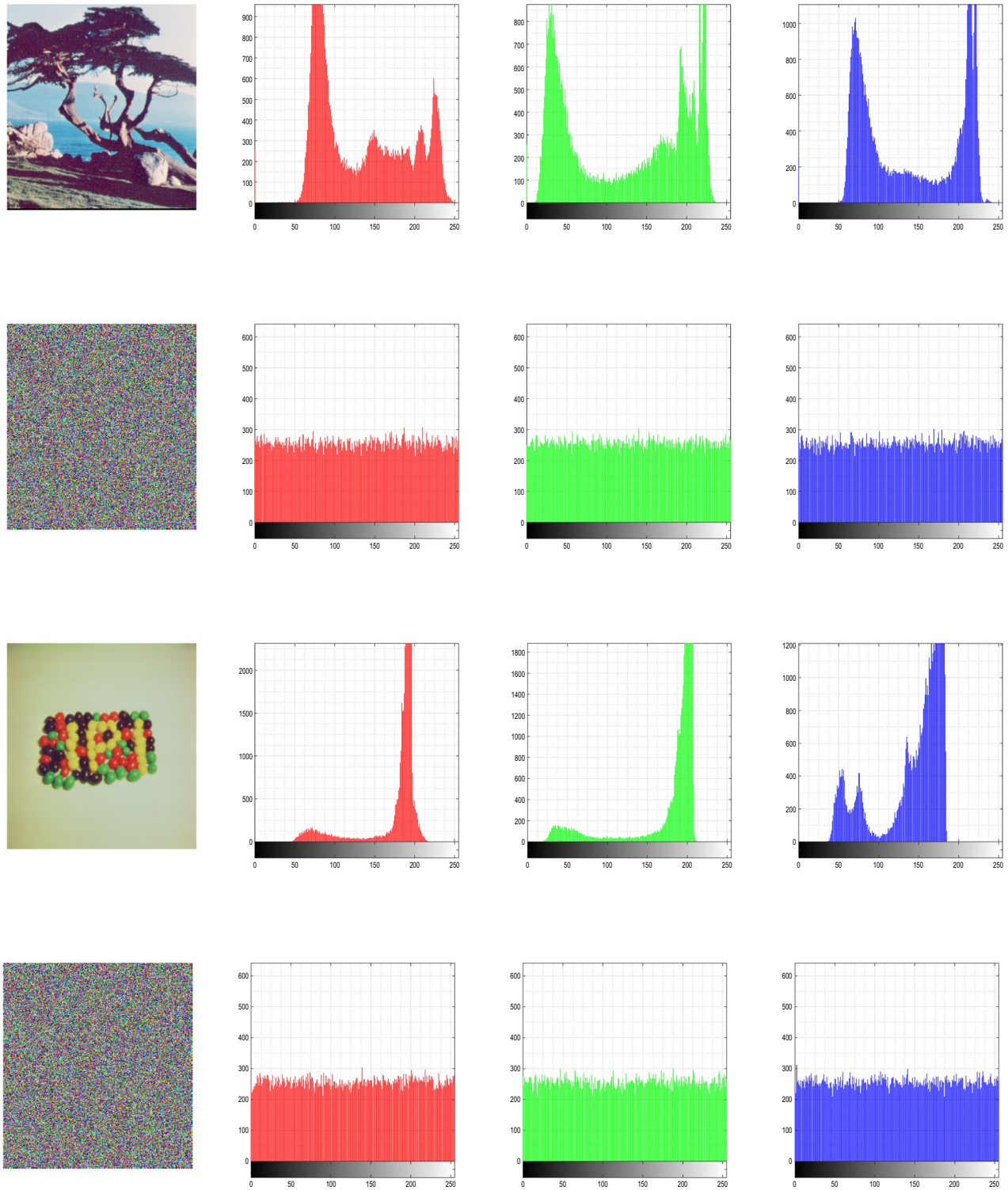

Fig. 16 Original images and encryption results and their histograms in the red, green and blue components

Table 3 Information entropy values using the proposed and different encryption methods

Image		Proposed	Ref. [16]	Ref. [17]	Ref. [18]
	R	7.9969	7.9973	7.9892	7.9966
Lena	G	7.9976	7.9969	7.9898	7.9972
256×256	B	7.9974	7.9971	7.9899	7.9967
	R	7.9971	7.9969	–	–
Couple	G	7.9968	7.9973	–	–
256×256	B	7.9974	7.9971	–	–
	R	7.997	7.9968	–	–
Tree	G	7.9971	7.9972	–	–
256×256	B	7.9972	7.9975	–	–

Step 3 . In this step, the shift function ColDiagRowSh3D (see Sect. 3) is called by quantifying eight vectors $V_1, V_2, V_3, V_5, V_6, V_7 \in \mathbb{Z}^{1\times(2n-3)}$ and $V_0, V_4 \in \mathbb{Z}^{1\times n}$ from the set vectors (6.1). And then, pixels of the image are scrambled as follows:

$$[I_1^r, I_1^g, I_1^b] = \text{ColDiagRowSh3D}\,(I^r, I^g, I^b, V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7).$$

Step 4 . Here, we use reversible cellular automata (RCA) introduced in Sect. 5 to have more confuse and diffuse. The function RCA is called as

$$I_2^k = \text{RCA}\,(I_1^k, a_k, b_k), \quad k = r, g, b,$$

where a_k, b_k are in $\mathbb{Z}_{256}^2$ and obtained from the new chaotic map.

Step 5 . The sequences X^k, $k = r, g, b$ is used as the original row of the circulant matrix M^k, $k = r, g, b$ as follows:

$$M^k(1, 1 : N) = X^k(1 : N), \quad k = r, g, b.$$

To have low relevance among the column vectors, we construct the rest of matrix in this way:

$$M^k(i, 1{:}N) = \text{circshift}\left(M^k(i-1, :), [0, (-1)^i.X^{k+}(i)]\right),$$
$$M^k(i, 1) = X^{k-}(i) \cdot M^k(i, 1), \quad 2 \le i \le N, \quad k = r, g, b,$$

where $k+$ stands for the next value of K, depending on the value of k and $k-$ is the previous one, for example if $k = g$, b is allocated for $k+$ and $k- = r$.

Table 5 Comparison of correlation coefficients for color images using the proposed and [16] algorithms

		Algorithm	Direction		
			Horizontal	Vertical	Diagonal
Lena	Proposed	R	0.0029	– 0.0009	0.0039
		G	0.0019	0.0054	0.0016
		B	– 0.0041	– 0.0058	– 0.0027
	Ref. [16]	R	– 0.0029	0.0013	– 0.0026
		G	– 0.0032	– 0.0032	– 0.0029
		B	0.0040	– 0.0018	0.0012
Tree	Proposed	R	– 0.0039	– 0.0027	0.0062
		G	– 0.0018	0.0060	– 0.0028
		B	0.0013	– 0.0029	– 0.0028
	Ref. [16]	R	– 0.0119	– 0.0034	0.0097
		G	0.0236	0.0004	0.0156
		B	0.0349	0.0218	– 0.0265

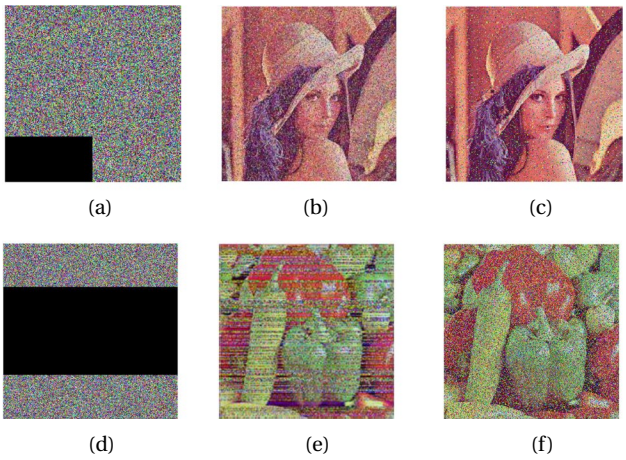

Fig. 17 Crop attack results: **a** cipher image with 1/8 data loss, **d** cipher image with 1/2 data loss; **b** decrypted by Ref. [23]; **e** decrypted by Ref. [16]; **c, f** decrypted by the proposed algorithm

Step 6 . Then, XOR operation is carried out to get the final encrypted image:

$$I_{enc}^k = A_2^k \oplus M^k, \quad k = r, g, b.$$

Table 4 Average NPCR and UACI values for Lena ($256 \times 256 \times 3$)

Measure		Proposed	Ref. [16]	Ref. [19]	Ref. [20]	Ref. [21]	Ref. [22]
	R	99.646	99.60	99.6185	99.6103	99.65	99.6210
NPCR	G	99.593	99.61	99.6090	99.6098	99.52	99.6314
	B	99.606	99.61	99.6254	99.6089	99.70	99.6234
	R	33.342	33.4655	33.4810	33.4655	33.43	33.4750
UACI	G	33.627	33.4781	33.4701	33.4652	33.49	33.4862
	B	33.424	33.4746	33.4526	33.4591	33.51	33.6851

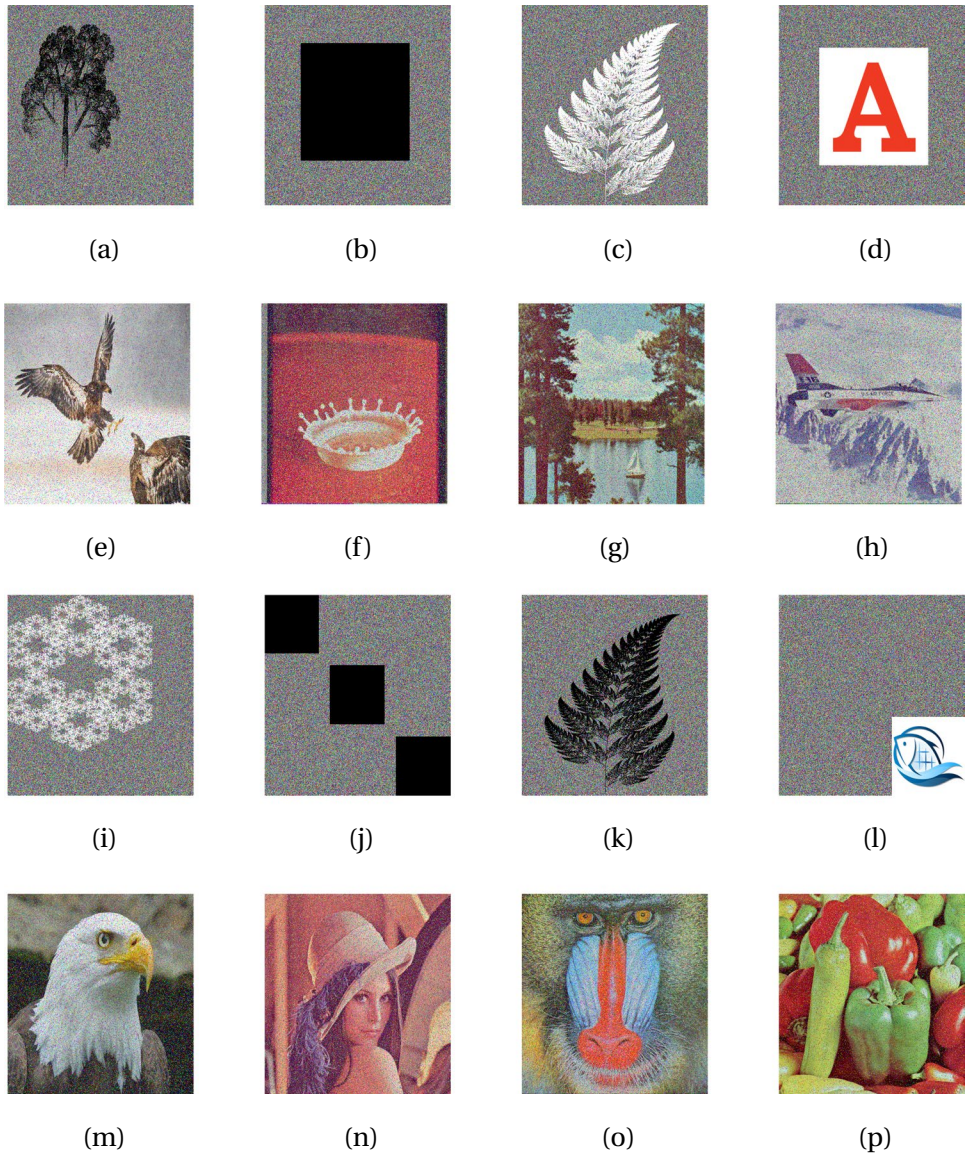

Fig. 18 Results for data loss attacks: **a** encrypted image splash; **b** cropping $300 \times 300 \times 3$ pixels of image; **c** data loss attacks by fractal graphs; **d** data loss attacks by color logo images($300 \times 300 \times 3$); **e–h** decrypted images corresponding the loss attacks; **i** data loss attacks by fractal graphs; **j** cropping $3 \times (150 \times 150 \times 3)$ pixels of image; **k** data loss attacks by fractal graphs; **l** data loss attacks by color logo images($150 \times 150 \times 3$); **m–p** decrypted images corresponding the loss attacks

Performance and security analysis

Experiments have been conducted in this section to evaluate the performance and security of the proposed algorithm. The total time of the encryption algorithm for RGB images 256×256 is about 48 s. The results pertaining to different properties of this algorithm have been compared with those of other studies, and the corresponding figures and tables are presented below. The new model is applied to some color images and simulation results are shown in Fig. 15. In this figure, the first columns are the plain images, the second columns are their cipher images and the third columns are their decrypted images. As it can be seen from this figure, the decrypted images are the same with the plain images which display security and effectiveness of the algorithm.

Histogram analysis

To show the proposed algorithm's resistance against statistical attacks, we use the histogram plot of the encrypted images, which shows frequency distributions of encryption systems. Figure 16 shows that the histogram plot is flat and uniform, thus the proposed scheme is significantly secure.

Information entropy analysis

Information entropy is a measure of the amount of uncertainty or randomness in a system. The entropy value for an ideally encrypted gray level image is very close to 8. In other words, any encrypted image having this value has been more randomly distributed. We conducted experiments to compare the entropy values of the proposed algorithm with those of

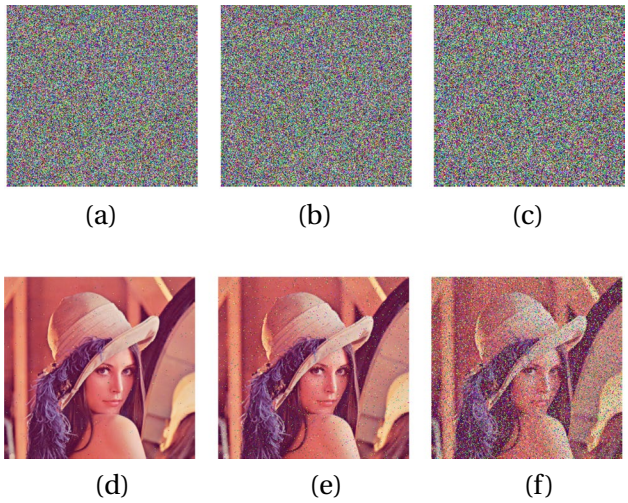

Fig. 19 Results for noise attacks: **a** noisy image by salt and pepper with density = 0.001; **b** noisy image by salt and pepper with density = 0.01; **c** noisy image by salt and pepper with density = 0.1; **d** decrypted image of (**a**); **e** decrypted image of (**b**); **f** decrypted image of (**c**)

recent encryption algorithms in Table 3. This table demonstrates that the information entropy value of encrypted images is almost 8, indicating that the proposed algorithm has minimal information loss and is highly secure against entropy attacks.

Differential attacks

The Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are using to measure the resistance of an encrypted image against differential attacks. Ideally these values are close to 100 and 33.33%, respectively, that show the highly sensitivity of an algorithm to ever minor changes in plain images. To test the proposed algorithm, the obtained results for NPCR and UACI have been compared with different algorithms in Table 4.

Statistical analysis

In order to gain an efficient encryption algorithm, the pixel correlations should be eliminated in the plain image and the encrypted image should have sufficiently low correlations close to 0. We have compared and tabulated the results of the proposed algorithm with those of [16] in Table 5.

Noise and crop attack analysis

A robust and effective encryption algorithm should be resist against data loss attacks in various types, namely noise, crop attacks and so on. Figures 18 and 19 demonstrate the ability of the new scheme. Figure 18 displays results of the decrypted images after cropping attacks, and Fig. 19 shows the decrypted images after contaminating cipher images by salt-and-pepper noise with different noise densities. We have compared and presented results of the proposed algorithm alongside existing methods in Fig. 17 which shows that the proposed algorithm possesses good performance against the noise and crop attacks.

Conclusion

This paper presents a pioneering approach to image encryption, applying a novel 3D modular hybrid chaotic map governed by two control parameters. The system comprises four components:

- A novel chaotic system: We introduce high-dimensional chaotic maps with dual control parameters to expand the chaotic space, enhancing security.
- A shift operator: We present a shift operator to rearrange image pixels, effectively reducing correlations between adjacent pixels.
- A key space: Our method establishes a robust key space derived from the proposed chaos map and CRC error detection technique, ensuring enhanced security.
- Cellular automata integration: We incorporate cellular automata to optimize the efficiency of the image encryption process, further enhancing its effectiveness.

Data availability The data that supports the findings of this study are available upon request.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper and they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. N.K. Nishchal, *Optical Cryptosystems* IOP Publishing, (2019)
2. Z. Hua, Y. Zhou, Image encryption using 2D Logistic-adjusted-Sine map. Inf. Sci. **339**, 237–253 (2016)
3. Y. Zhang, The unified image encryption algorithm based on chaos and cubic S-Box. Inf. Sci. **450**, 361–377 (2018)
4. F. Benkhedir, N. Hadj Said, A. Ali Pacha, A. Hadj Brahim, Image encryption based on 5-D hyper-chaotic and a novel chess game permutation. J. Opt. 1–34 (2023)
5. G.A. Gottwald, I. Melbourne, A new test for chaos in deterministic systems. Proc. R. Soc. A: Math. Phys. Eng. Sci. **460**(2042), 603–611 (2004)

6. G.A. Gottwald, I. Melbourne, Testing for chaos in deterministic systems with noise. Phys. D: Nonlinear Phenom. **212**(1–2), 100–110 (2005)
7. G.A. Gottwald, I. Melbourne, On the implementation of the 0–1 test for chaos. SIAM J. Appl. Dyn. Syst. **8**(1), 129–145 (2009)
8. W.W. Peterson, D.T. Brown, Cyclic codes for error detection. Proc. IRE **49**(1), 228–235 (1961)
9. J.V. Neumann, *Theory of Self-Reproducing Automata*, ed. by Arthur W. Burks (1966)
10. A. Fuster-Sabater, P. Caballero-Gil, On the use of cellular automata in symmetric cryptography. Acta Appl. Math. **93**(1–3), 215–236 (2006)
11. A.A. Abdo, S. Lian, I.A. Ismail, M. Amin, H. Diab, A cryptosystem based on elementary cellular automata. Commun. Nonlinear Sci. Numer. Simul. **18**(1), 136–147 (2013)
12. Y. Khedmati, R. Parvaz, Y. Behroo, 2D Hybrid chaos map for image security transform based on framelet and cellular automata. Inf. Sci. **512**, 855–879 (2020)
13. S. Roy, M. Shrivastava, U. Rawat, C.V. Pandey, S.K. Nayak, IESCA: An efficient image encryption scheme using 2-D cellular automata. J. Inf. Secur. Appl. **61**, 102919 (2021)
14. K. Lok, The Lyapunov Exponent Test and the 0–1 Test for Chaos compared, Doctoral dissertation, Faculty of Science and Engineering, (2016)
15. R. Parvaz, Y. Khedmati Yengejeh, Y. Behroo, A new 4D chaos system with an encryption algorithm for color and grayscale images. Int. J. Bifurc. Chaos Appl. Sci. Eng. **32**(14), 2250214 (2022)
16. X. Chai, X. Fu, Z. Gan, Y. Lu, Y. Chen, A color image cryptosystem based on dynamic DNA encryption and chaos. Signal Process. **155**, 44–62 (2019)
17. X. Wu, K. Wang, X. Wang, H. Kan, J. Kurths, Color image DNA encryption using NCA map-based CML and one-time keys. Signal Process. **148**, 272–287 (2018)
18. A. ur Rehman, X. Liao, R. Ashraf, S. Ullah, H. Wang, A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. Optik **159**, 348–367 (2018)
19. X. Wang, N. Guan, Chaotic image encryption algorithm based on block theory and reversible mixed cellular automata. Opt. Laser Technol. **132**, 106501 (2020)
20. L. Huang, S. Cai, X. Xiong, M. Xiao, On symmetric color image encryption system with permutation-diffusion simultaneous operation. Opt. Lasers Eng. **115**, 7–20 (2019)
21. X.Y. Wang, Z.M. Li, A color image encryption algorithm based on Hopfield chaotic neural network. Opt. Lasers Eng. **115**, 107–118 (2019)
22. R. Vidhya, M. Brindha, A chaos based image encryption algorithm using Rubik's cube and prime factorization process (CIERPF). J. King Saud Univ. Comput. Inf. Sci. **34**(5), 2000–2016 (2020)
23. A.Y. Niyat, M.H. Moattar, M.N. Torshiz, Color image encryption based on hybrid hyper-chaotic system and cellular automata. Opt. Lasers Eng. **90**, 225–237 (2017)