AIMS *Mathematics*

AIMS Mathematics, 11(6): 17166–17190.
DOI: 10.3934/math.2026704
Received: 19 March 2026
Revised: 25 April 2026
Accepted: 01 June 2026
Published: 15 June 2026

https://www.aimspress.com/journal/Math

Research article

A new color image encryption algorithm based on orthogonal Lagrange-Fourier moments and a 6D chaotic system of fractional-order

Faisal S. Alsubaei[1], Mohamed Meselhy Eltoukhy[2], Mohamed M. Darwish[3] and Khalid M. Hosny[4,*]

[1] Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia
[2] Department of Information Technology, College of Computing and Information Technology at Khulais, University of Jeddah, Jeddah, Saudi Arabia
[3] Department of Computer Science, University College of Al Wajh, University of Tabuk, Tabuk, Saudi Arabia
[4] Department of Information Technology, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt

* **Correspondence:** Email: k_hosny@yahoo.com.

Abstract: To enhance image security in transmission and storage, we proposed a new color image encryption approach based on orthogonal Lagrange-Fourier moments (OLFMs) and a 6D fractional-order chaotic system (6D-FCS), in which the complex dynamic behavior of 6D-FCS is leveraged to generate highly unpredictable key streams, thereby ensuring strong confusion and diffusion properties. The proposed encryption algorithm comprised two stages: First, the 6D-FCS was employed to generate chaotic sequences that shuffle pixel positions, establishing the confusion stage and ensuring a high level of pixel scrambling. Second, OLFMs were used to modify pixel values, thereby achieving the diffusion stage. The numerical simulations and security analysis showed that the proposed encryption algorithm provides a key space greater than 2^{647}, indicating strong security and outstanding encryption performance, as evidenced by key space, correlation coefficient, information entropy, histogram, UACI, and NPCR, thereby resisting statistical, differential, noise, and cropping attacks, highlighting its efficacy for image encryption. Moreover, the proposed encryption algorithm exhibits significant security and superiority to the related algorithms.

Keywords: 6D fractional-order chaotic system; confusion and diffusion stages; orthogonal Lagrange-

Fourier moments; color image encryption
Mathematics Subject Classification: 94A08

1. Introduction

Securing digital images has received considerable attention due to the rapid growth of multimedia sharing and digital communication, as well as the sensitivity of certain images. Image encryption is a powerful technology for safeguarding digital images, ensuring the integrity and privacy of visual content, particularly in healthcare, military, and surveillance [1–3]. Accordingly, various technologies have been employed to propose image encryption methods, including quantum computing [4,5], optical methods [6], Toeplitz linear systems [7], orthogonal moments [8,9], deep learning [10], and chaotic systems [11,12].

Chaotic systems, among many encryption techniques, exhibit unpredictable, irregular trajectories and are highly sensitive to initial conditions. Moreover, chaotic systems produce sequences, exhibiting nonperiodic, noise-like, and pseudorandom characteristics. Thus, chaotic-system-based cryptosystems are more secure and better suited for encrypting images [13–15]. However, the algorithms presented are based on integer-order chaotic systems, which inherently constrain their chaotic richness and complexity. Because FCS exhibit richer, more complex dynamics, memory, an extensive key space, and nonlocality, which increase sensitivity to initial conditions and parameter variations and lead to highly unpredictable behavior compared with integer-order chaotic systems, they have attracted considerable attention, especially in image encryption [16–18]. Additionally, fractional order can be treated as an extra parameter within the total key space, and chaotic systems with fractional-order dynamics exhibit a broad spectrum of attractors and more complex sequences as the fractional order varies, resulting in significantly enhanced resistance to brute-force, differential, and statistical attacks [19]. Moreover, orthogonal moments (OMs) have been applied to image encryption [20–23] due to their ability to extract important visual features, describe and represent images, and exhibit high noise resistance and invariance to geometric transformations.

Motivated by the intrinsic characteristics of orthogonal moments and fractional-order chaotic systems, we propose a novel color image encryption algorithm based on a 6D fractional-order chaotic system and orthogonal Lagrange–Fourier moments. The high-dimensional fractional dynamics of the 6D-FCS generate complex, highly unpredictable chaotic sequences that are effectively employed in the confusion process to achieve efficient and secure pixel permutation.

In the diffusion process, OLFMs are combined with the generated chaotic sequences to perform effective pixel transformations, ensuring uniform propagation of pixel variations across the image. This integration significantly strengthens diffusion capability while enhancing key sensitivity and randomness. Additionally, the proposed algorithm introduces a hybrid design that combines high-dimensional fractional chaos with moment-based diffusion, resulting in improved statistical properties and greater resistance to differential, statistical, and brute-force attacks. Consequently, the suggested algorithm achieves superior security and provides a robust framework for color image encryption. To the best of our knowledge, the 6D-FCS has not been explored for image encryption, and its integration with OLFM-based diffusion in a unified approach has not been reported. The major contributions of this paper are:

- We propose an image encryption algorithm that combines the enhanced dynamic behavior and chaotic performance of a 6D fractional-order chaotic system with orthogonal moments.

- We employ a 6D fractional-order chaotic system to generate complex chaotic sequences for the confusion stage, increasing key sensitivity, enhancing permutation unpredictability, and strengthening overall security.
- We integrate OLFMs with chaotic sequences for the diffusion stage, thereby enhancing the resistance of the suggested algorithm against statistical and differential attacks.

The remainder of this paper is as follows: In Section 2, we summarize the related works. In Section 3, we describe the proposed color image encryption algorithm. In Section 4, we present simulation results and performance evaluation. In Section 5, we summarize conclusions.

2. Related works

Fractional-order chaotic systems have been widely used in the design of image encryption algorithms. Using the fractional-order Chen chaotic system to generate a substitution box [24], Ullah et al. proposed image encryption approaches that integrate Arnold's cat map, S-box, and RSA algorithm with the 2D Henon chaotic map [25]. By combining a FO4D hyperchaotic Chen map and a sine chaotic map, Alexan et al. [26] proposed a hybrid DNA-coding algorithm for encrypting color images. Using eight-base DNA cubes and a 5D fractional-order complex chaotic system, Wu et al. [27] suggested a method for encrypting a color image. High-dimensional chaos generates complex keystreams, while DNA cube operations introduce multi-layered confusion and diffusion at the nucleotide level. Hosny et al. [28] integrated the Fibonacci Q-matrix with a fractional-order 4D hyperchaotic Chen system to introduce an approach for color image encryption. Using a combination of an extended DNA encoding approach and a fractional-order 5D hyperchaotic system, Meng and Wu [29] proposed an algorithm for encrypting color images. Yan et al. [30] incorporated a memristor into a 5D hyperchaotic system with fractional-order dynamics and introduced a Rubik's Cube-based scrambling algorithm to encrypt color images. In this system, an enhanced Arnold transform and two Rubik's Cube scrambling algorithms are combined with a double-diffusion process to encrypt an image. Salman et al. [31] developed a fractional-order chaotic circuit system based on the hyperbolic sine function and successfully applied it to image encryption. Integrating the extended Zig-Zag transform with a fractional-order conservative memristive hyperchaotic system (FCMHS), F. Meng et al. [32] designed an algorithm for encrypting color images. Nabil and Tayeb [33] proposed a conformable fractional-order chaotic system with a wide chaotic range and rich dynamics, and applied it to encrypt a color image. In a subsequent study, using the Caputo operator, the researchers in [34] introduced a discrete-time fractional-order map that exhibits high complexity and sensitive chaotic behavior for secure color image encryption. Based on Galois field operations and a fractional-order hyperchaotic complex system, Yang et al. [35] introduced an image encryption algorithm. Rahman et al. [36] proposed a color image encryption scheme based on a hyperchaotic memcapacitive model. Using a fractional-order hyperchaotic system, Gao et al. [37] proposed an algorithm for encrypting multiple images. The researchers in [38] proposed a medical image encryption algorithm in which the confusion process is governed by a 4D fractional-order chaotic system, while the diffusion process combines a one-dimensional chaotic map with a symmetric matrix. A method for encrypting color images is proposed in [39], which incorporates a fractional-order laser system, extended DNA coding, and the Zig-Zag transform.

3. The proposed encryption algorithm

In this section, we introduce an algorithm for color image encryption that integrates an F6DCS for confusion with an OLFM for diffusion. In the first subsection, we present the mathematical equations and briefly discussed how to compute the OLFMs. In the second subsection, we introduce the F6DCS fractional-order chaotic system and then presented the required tests to demonstrate its dynamical behavior.

3.1. Orthogonal Lagrange-Fourier moments

The OLFMs using the orthogonal Lagrange polynomials are summarized in this section.

Let $\{r_i\}_{i=0}^{N-1}$be a set of distinct real numbers such that $r_i \neq r_j$ for all $i \neq j$. The corresponding Lagrange basis polynomials $l_i(r)$, for $i = 0, \dots, N-1$, associated with this set are defined as follows [40]:

$$l_i(r) = \prod_{j=0\ j\neq i}^{N-1} \frac{r-r_j}{r_i-r_j}, \qquad i = 0,1, \dots, N-1. \tag{1}$$

The Lagrange polynomials l_i obey the following relation:

$$l_i(r_j) = \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \tag{2}$$

where δ_{ij} is the Kronecker delta.

Using Eq (2), the Lagrange polynomials l_i are shown to be orthogonal under the discrete inner product, satisfying the orthogonality relation as follows [40]:

$$\langle l_m\ , l_n \rangle = \sum_{i=0}^{N-1} l_m(r_i) l_n(r_i) = \delta_{mn}. \tag{3}$$

Using the Lagrange polynomials described earlier, the OLFMs are obtained by converting the image from the Cartesian to the polar domain via the rectangle-to-circle mapping [41]. This transformation enables computing orthogonal moments by representing the $2N \times 2N$ image as concentric circular regions. The resulting disk is reorganized into N radial rings, each separated by circular rays.

$$\{r_j = j,\ j = 0, 1,2, \dots, N-1\}. \tag{4}$$

The disc is partitioned into unit radial tracks, and the number of pixels contained in each track is determined by:

$$N_j = \pi(2j+1),\ j = 0, 1, 2, \dots, N-1. \tag{5}$$

Let N_j denote the number of pixels in the j^{th} track. The associated angular division is therefore $\Delta\theta_j$, defined by:

$$\Delta\theta_j = \frac{2\pi}{N_j},\ j = 0,1,2, \dots, N-1. \tag{6}$$

$$\theta_{jt} = \frac{2\pi t}{N_j},\ j = 0,1,2, \dots, N-1\ ,\ t\ =\ 0, 1, .., N_j\ -\ 1. \tag{7}$$

Hence, we define the basic functions as the following set:

$$D_{nm}\left(\overline{r_j}, \overline{\theta_{jt}}\right) = l_n\left(\overline{r_j}\right) e^{im\overline{\theta_{jt}}}, \tag{8}$$

where:

$$\overline{r_j} = \frac{r_j + r_{j+1}}{2} = \frac{2j+1}{2}. \tag{9}$$

$\{l_j\}_{j=0}^{N-1}$ denotes the orthogonal Lagrange polynomials related to the set $\{\bar{r}_j\}_{j=0}^{N-1}$, and

$$\overline{\theta_{jt}} = \frac{\theta_{j,t} + \theta_{j,(t+1)}}{2} = \frac{\pi(2t+1)}{N_j} \quad j = 0,1,2,\ldots,N-1 \ , t \ = \ 0,1,..,N_j \ - \ 1. \tag{10}$$

The complex function $e^{im\overline{\theta_{jt}}}$, $m \in Z$ satisfy orthogonality as:

$$\sum_{t=0}^{N_j - 1} e^{im\overline{\theta_{jt}}} e^{in\overline{\theta_{jt}}} = \sum_{t=0}^{N_j - 1} e^{im\frac{\pi(2t+1)}{N_j}} \ e^{-in\frac{\pi(2t+1)}{N_j}} = N_j \, \delta_{nm}. \tag{11}$$

Using Eqs (3) and (11), the orthogonality of $D_{nm}$is obtained as:

$$\langle D_{nm}, D_{uv} \rangle = \sum_{i=0}^{N-1} \sum_{t=0}^{N_j - 1} D_{nm}\left(\overline{r_j}, \overline{\theta_{jt}}\right) \overline{D_{uv}\left(\overline{r_j}, \overline{\theta_{jt}}\right)} = \pi(2n+1)\delta_{nu} \ \delta_{mv}. \tag{12}$$

The OLFMs for an image f, are defined as follows for all $n = 0,1,\ldots,N-1$and $m \in \mathbb{Z}$:

$$LF_{nm}(f) = \frac{1}{N_n} \langle f\left(\overline{r_j}, \overline{\theta_{jt}}\right), D_{nm}\left(\overline{r_j}, \overline{\theta_{jt}}\right) \rangle = \frac{1}{N_n} \sum_{i=0}^{N-1} \sum_{t=0}^{N_j - 1} f\left(\overline{r_j}, \overline{\theta_{jt}}\right) \overline{D_{uv}\left(\overline{r_j}, \overline{\theta_{jt}}\right)}$$

$$= \frac{1}{N_n} \sum_{i=0}^{N-1} \sum_{t=0}^{N_j - 1} l_n\left(\overline{r_j}\right) e^{-im\overline{\theta_{jt}}} f\left(\overline{r_j}, \overline{\theta_{jt}}\right). \tag{13}$$

Applying Eq (2) with $l_n\left(\bar{r}_j\right) = \delta_{nj}$, the $LF_{nm}(f)$ simplifies to:

$$LF_{nm}(f) = \frac{1}{N_n} \sum_{t=0}^{N_n - 1} e^{-im\overline{\theta_{nt}}} f(\overline{r_n}, \overline{\theta_{nt}}). \tag{14}$$

Equation (14) enables direct computation of orthogonal moments without the use of Lagrange polynomials, thereby preventing geometric and numerical integration errors. Additionally, $LF_{nm}(f)$ relies solely on the pixels of the n-th ring. Consider a color image $g(r,\theta)$ composed of the red (R), green (G), and blue (B) channels, i.e.,

$$g(r,\theta) = \left(g_R(r,\theta), g_G(r,\theta), g_B(r,\theta)\right). \tag{15}$$

For the color image $g(r,\theta)$, the multi-channel OLFMs of order $(n+m)$ is defined by:

$$MLF_{nm}(g) = \left(MLF_{nm}(g_R) \ , MLF_{nm}(g_G), MLF_{nm}(g_B)\right), \tag{16}$$

where $MLF_{nm}(g_C)$, $C \in R, G, B$ is defined by.

$$MLF_{nm}(g_C) = \frac{1}{N_n} \sum_{j=0}^{N-1} \sum_{t=0}^{N_j - 1} l_n\left(\overline{r_j}\right) e^{-im\overline{\theta_{jt}}} g_C\left(\overline{r_j}, \overline{\theta_{jt}}\right). \tag{17}$$

Using Eqs (2) and (17) can be rewritten as:

$$MLF_{nm}(g_C) = \frac{1}{N_n} \sum_{t=0}^{N_n - 1} e^{-im\overline{\theta_{nt}}} g_C(\overline{r_n}, \overline{\theta_{nt}}). \tag{18}$$

3.2. The 6D fractional chaotic system

Throughout this section, ${}_0^C D_t^\nu$ is denoted by D^ν for simplicity. The corresponding 6D-nonlinear fractional autonomous system is introduced as follows in [42]:

$$\begin{aligned} D^\nu y_1 &= \alpha_1(y_2 - y_1) + y_2 y_3 y_4, \\ D^\nu y_2 &= \beta_1(y_2 + y_1) - y_1 y_3 y_4, \\ D^\nu y_3 &= -\gamma y_3 + y_1 y_2 y_4 + y_5 y_6 y_4, \\ D^\nu y_4 &= -\delta y_4 + y_1 y_2 y_3 + y_5 y_6 y_3, \\ D^\nu y_5 &= \alpha_2(y_6 - y_5) + y_6 y_3 y_4, \\ D^\nu y_6 &= \beta_2(y_6 + y_5) - y_5 y_3 y_4, \end{aligned} \tag{19}$$

where D^ν stands for the Caputo fractional derivative $(0 < \nu \leq 1)$, $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \delta$ are the parameters of the system and all of them are positive fundamental constants, and y_i $(1 = 1,2,3, \ldots .,6)$. These are the system's state variables. Taking $\alpha_1 = \alpha_2 = a$, $\beta_1 = \beta_2 = b$, $\gamma = c$, and $\delta = d$, fractional order $\nu = 0.8515,\ 0.9010,\ 0.9505,\ 1$, and initial conditions $Y(0) = (1.77,\ 1.66,\ 7.2,\ 1.5,\ 1.77,\ 1.66)$.

3.3. Dynamical analysis

In this section, the dynamic behavior and chaotic performance of 6D-FCS are analyzed and evaluated using Lyapunov Exponents (LEs), and the NIST test. For simplicity and to avoid redundancy, only the LEs characteristics are summarized, while the full dynamic behavior analysis can be found in the original study [42].

3.3.1. Lyapunov exponents

As an effective quantitative measure of complexity in nonlinear systems, the LE is a crucial index for analyzing the dynamic behavior of chaotic systems. A system is identified as chaotic when its largest LE is positive. If more than one LE is positive, the system is identified as hyperchaotic, highlighting more random and complex behavior than standard chaotic systems. In general, higher LE values imply increased dynamical complexity. Table 1 reports LE values for various fractional orders ν. It can be observed that LE_1 and LE_2 are positive for the fractional-order, range $\nu \in [0.8515, 1]$. Setting the parameters as $a = 30$, $b = 10$, $c = 1$, and $d = 10$, the findings demonstrate that the system in Eq (19) exhibits hyperchaotic behavior for the v element in the range $0.8515 < \nu < 1$.

The fractional order is set to 0.9, at which the system exhibits the two largest positive LEs; this setting is used to design the proposed color image encryption method.

Table 1. LEs for different fractional orders v [42].

LE	Fractional order			
	0.8515	0.9010	0.9505	1.0
LE1	5.9389	4.9808	3.6936	0.3514
LE2	0.0031	0.0362	0.0006	0.0007
LE3	-0.1850	-0.0060	-0.0006	-0.0068
LE4	-8.7262	-6.9942	-5.3108	-3.8473
LE5	-40.1658	-32.1153	-25.4198	-19.9668
LE6	-57.4287	-46.8727	-37.3238	-27.2099

3.3.2. NIST SP800-22 test

The NIST is a widely used test suite comprising 15 subtests to assess the randomness of chaotic sequences. The system-generated chaotic sequence is converted into a binary sequence and subjected to the NIST test. The evaluation is performed for 100 independent sequences, where each binary sequence is generated with a length of at least 10^6 bits. Table 2 shows that the chaotic sequence passed all NIST tests with p-values > 0.01, indicating strong randomness. These results demonstrate the effectiveness of the suggested algorithm for key generation and secure encryption.

Table 2. NIST test.

No.	Test	P-Value	Result
1	Frequency	0.3646	✓
2	Block Frequency	0.7017	✓
3	Cumulative Sums	0.5553	✓
4	Runs	0.5038	✓
5	Longest Run	0.4668	✓
6	Binary Matrix Rank	0.3721	✓
7	FFT	0.3433	✓
8	Non-Overlapping Template	0.1819	✓
9	Overlapping Template	0.3738	✓
10	Approximate Entropy	0.6243	✓
11	Random Excursions	0.4538	✓
12	Random Excursions Variant	0.5754	✓
13	Linear Complexity	0.5417	✓
14	Serial	0.4016	✓
15	Universal	0.6206	✓

3.4. Proposed algorithm of encryption

In this section, we introduce an algorithm for color image encryption that integrates an F6DCS for confusion with an OLFM for diffusion. The fractional 6D chaotic system generates strong pseudo-random sequences for secure pixel scrambling. Next, OLFM coefficients are used to generate diffusion masks that modify pixel values and enhance resilience against statistical and differential attacks. Figure 1 presents the encryption process, and Algorithm 1 describes it.

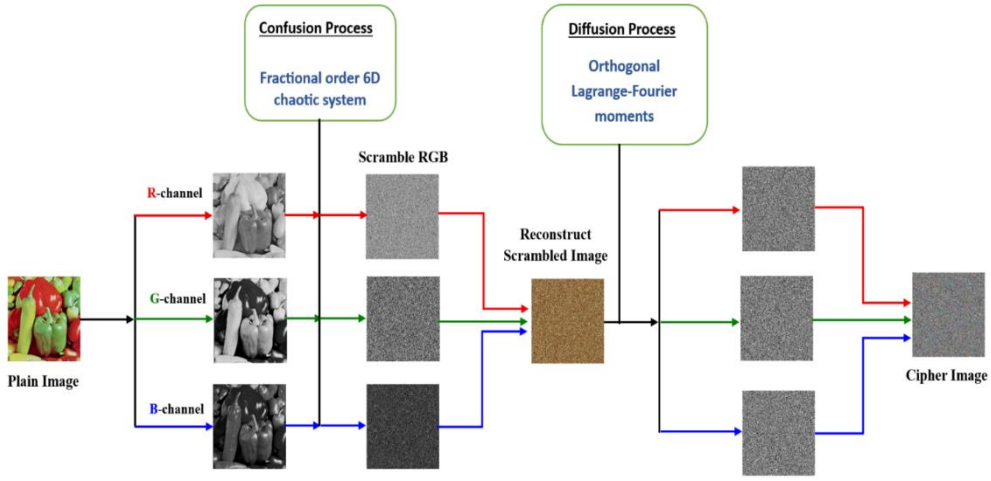

Figure 1. Encryption process.

Algorithm 1. Encryption process.

Input: Plain image $\boldsymbol{I}$ with size $M \times N$, initial conditions $y_1, y_2, y_3, y_4, y_5, y_6$, parameters $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \delta$, and the fractional order $\upsilon \in (0, 1]$.

Output: Encrypted image E.

1. The system in Eq (1) is iterated for $N \times M + T$ iterations and discard T.
2. Generate the six chaotic sequences $Y_1, Y_2, \ldots, Y_6$, of length $N \times M$ from the final iterations.
3. Normalize the sequences to the range [0, 255] to generate key streams using Eq (20).
4. Flatten the color image I into a vector, V_R, V_G, and V_B.
5. Use chaotic sequences Y_2, Y_4, Y_5 to generate a permutation index, P, by sorting the values in ascending order.
6. Permute and rearrange the pixel positions of each channel according to the permutation index to generate shuffled vectors, $S_R, S_G, and\ S_B$ using Eq (21).
7. Use three chaotic sequences Y_1, Y_3, Y_6 from the six chaotic sequences $Y_1, Y_2, \ldots, Y_6$ to generate inter-channel permutation indices.
8. Swap or shuffle the R, G, and B components' values at corresponding pixel positions according to these indices.
9. Reshape the shuffled vectors S_R, S_G, and S_B to reconstruct the confused channels C_R, C_G, and C_B with dimensions $M \times N$.
10. For the scrambled image C, the OLFM coefficients are computed using Eq (18).
11. The OLFM moments are utilized to construct a key, K_{OLFM}, using Eq (22).
12. The $M \times N$ encrypted image E is construed using Eq (23).

3.4.1. Encryption process

The encryption process is described through the following steps:

Step 1. F6DCS based Key generation.

1. Initialize the F6DCS with initial values $y_1, y_2, y_3, y_4, y_5, y_6$, parameters $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \delta$, and the fractional order $\upsilon \in (0, 1]$ to generate a secret key.
2. The system in Eq (1) is iterated for $N \times M + T$ iterations, where $N \times M$ is the number of pixels, and T is the transient iteration count used to eliminate transient behavior. Performing sufficient

iterations ensures highly unpredictable chaotic sequences, increases sensitivity to initial conditions, and strengthens the encryption's security.

3. Generate the six chaotic sequences $Y_1, Y_2, \ldots, Y_6$, of length $N \times M$ from the final iterations.
4. Normalize the sequences to the range [0, 255] to generate key streams using the floor function $\lfloor . \rfloor$ in Eq (20) as follows:

$$K = mod\left(\left\lfloor \sum_{i=1}^{6} Y_i \times 10^{15} \right\rfloor, 256\right). \tag{20}$$

Step 2. Pixel-level confusion.

1. Flatten the color image $\boldsymbol{I}$ into a one-dimensional vector, V_R, V_G, and V_B for each channel (R, G, B).
2. Use chaotic sequences Y_2, Y_4, Y_5 to generate a permutation index, P_R, P_G, and P_B for the R, G, and B channels, respectively, each obtained by sorting the values in ascending order.
3. Permute and rearrange the pixel positions of each channel according to the permutation index to generate shuffled or scrambled vectors, such as:

$$\begin{aligned} S_R &= V_R(P_R(i)), \\ S_G &= V_G(P_G(i)), \\ S_B &= V_B(P_B(i)), \qquad i = 1,2, \ldots, MN. \end{aligned} \tag{21}$$

This ensures that pixel positions are completely shuffled by chaotic sequences.

Step 3. Inter-channel scrambling. This step removes the correlation among color channels, thereby further increasing confusion.

1. Use three chaotic sequences Y_1, Y_3, Y_6 from the six chaotic sequences $Y_1, Y_2, \ldots, Y_6$.
2. Normalize the sequences to the range [0, 255] as $H = mod(\ \lfloor (Y_1 + Y_3 + Y_6) \times 10^{15} \rfloor, 256)$ and reshaped to $M \times N$.
3. Compute $D = mod(\mathrm{H}, 3)$, if
   - $D = 0$: (R, G, B) → (G, R, B);
   - $D = 1$: (R, G, B) → (B, G, R);
   - $D = 2$: (R, G, B) → (R, B, G).

Step 4. Shuffled image.

1. Reshape the shuffled vectors S_R, S_G, and S_B to reconstruct the confused channels C_R, C_G, and C_B with dimension $M \times N$.
2. Concatenate the confused channels C_R, C_G, and C_B to form the scrambled color image C.

Step 5. Diffusion process. In the diffusion stage, OLFM moments are used to modify the pixel values of the shuffled color image, ensuring that minor changes propagate throughout the image, thereby strengthening robustness to differential and statistical attacks.

1. For the scrambled image C, the OLFM coefficients are computed using Eq (18).
2. The OLFM moments are utilized to construct a key as:

$$K_{\mathrm{OLFM}} = mod(\lfloor |MLF_{nm}| \times 10^{15} \rfloor, 256). \tag{22}$$

3. Flatten the scrambled image C into a one-dimensional vector, CV_R, CV_G, and CV_B, for each channel (R, G, B).
4. The diffused vectors EV_R, EV_G, and EV_B are construed as follows:

$$\begin{aligned} EV_R &= K_{OLFM} \oplus K \oplus CV_R, \\ EV_G &= K_{OLFM} \oplus K \oplus CV_G, \end{aligned} \tag{23}$$

$$EV_B = K_{OLFM} \oplus K \oplus CV_B.$$

5. Reshape the diffused vectors EV_R, EV_G, and EV_B to reconstruct the diffused channels E_R, E_G, and E_B with dimension $M \times N$.
6. The $M \times N$ encrypted image E is construed by concatenating the diffused channels E_R, E_G, and E_B.

3.4.2. Decryption process

To ensure exact reconstruction of the plain image, the decryption process used the same key as the encryption process, thereby ensuring reversibility. Algorithm 2 depicts the decryption process.

Algorithm 2. Decryption process.

Input: Encrypted image E, initial conditions $y_1, y_2, y_3, y_4, y_5, y_6$, parameters $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \delta$, and the fractional order $\upsilon \in (0, 1]$.

Output: Decrypted image I.

1. Use the same initial variables and parameters as in the encryption.
2. Generate the six chaotic sequences $Y_1, Y_2, \ldots, Y_6$, of length $N \times M$, as the same in the encryption.
3. Flatten the encrypted image E into a vector, EV_R, EV_G, and EV_B.
4. Use chaotic sequences Y_2, Y_4, Y_5 to generate a permutation index, *P*, by sorting the values in ascending order.
5. Permute and rearrange the pixel positions of each channel according to the permutation index to restore shuffled vectors, DS_R, DS_G, and DS_B.
6. Reshape the restored shuffled vectors DS_R, DS_G, and DS_B to reconstruct the confused channels DC_R, DC_G, and DC_B with dimensions $M \times N$.
7. Merge the confused DC_R, $\mathrm{DC_G}$, and $\mathrm{DC_B}$ channels to construct the 2D restored scrambled image DC.
8. Using the key K_{OLFM} as the same in the encryption, the $M \times N$ decrypted image I is constructed by $I = K_{OLFM} \oplus K \oplus DC$.

4. Experimental results and security analysis

Simulation experiments are conducted to verify the performance and security analysis of the suggested algorithm. We select eight standard color images, sized 256 × 256 and 512 × 512, that serve as plain images from the USC-SIPI [43]. A standard environment is used across all simulations: MATLAB 2015a on Windows 10, with 8.0 GB of RAM and an Intel(R) Core (TM) i7-2.9 GHz CPU.

4.1. The visual analysis

The visual analysis of the proposed algorithm is performed using the plain images shown in Figures 2(a)–(h). Figure 2 presents the encryption and decryption results. The encrypted images are shown in Figures 2(i)–(p), which reveals that the proposed encryption algorithm effectively conceals the information in the original images, thereby ensuring security during transmission. Moreover, Figures 2(q)–(x) show the decrypted images, which are identical to the original images, confirming successful decryption.

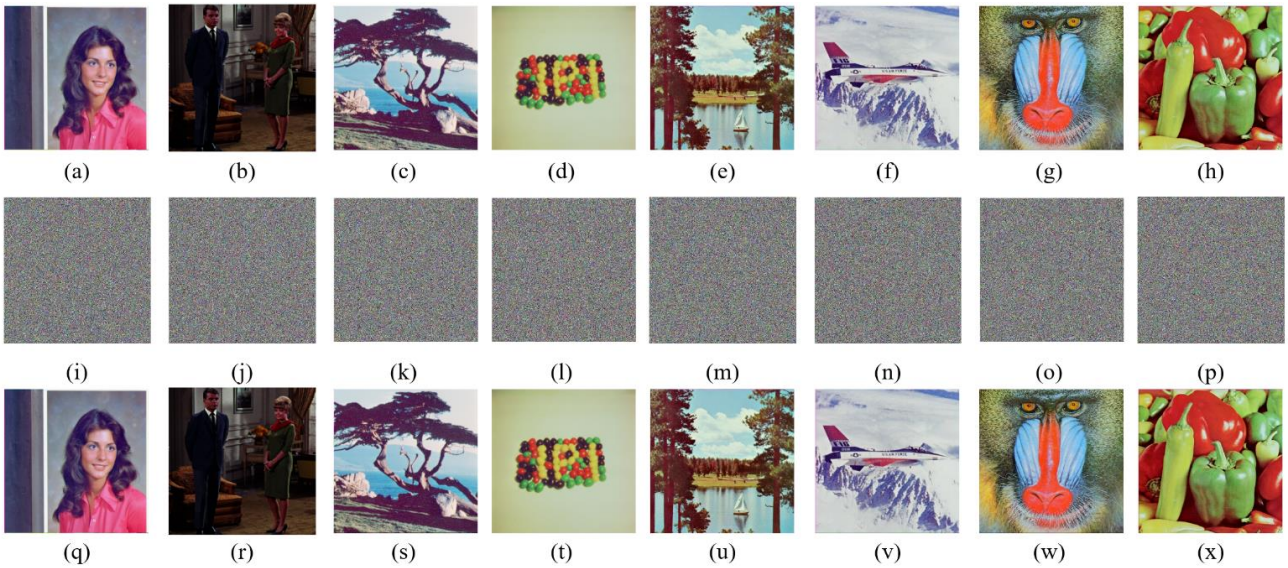

Figure 2. The encryption and decryption results.

4.2. Keyspace analysis

The key space comprises all possible keys in the cryptosystem. It assesses the encryption algorithm's resilience to brute-force attacks. The minimum key space required for image encryption algorithms to resist brute-force attacks is 2^{100}. The keys of the proposed algorithm include the fractional order υ, initial conditions, $y_1, y_2, y_3, y_4, y_5, y_6$, and $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \delta$, parameters for the system in Eq (1). If the computational precision is 10^{-15}, then the suggested algorithm key space is $10^{195} \approx 2^{647} > 2^{100}$. Table 3 presents the key space of the proposed algorithm compared with existing algorithms. As shown in Table 3, the proposed algorithm has a larger key space, demonstrating strong resilience against brute-force attacks.

Table 3. Key space analysis.

Algorithm	Proposed	[8]	[9]	[12]	[13]	[15]	[28]	[29]	[39]
Key space	2^{647}	2^{293}	2^{498}	2^{598}	2^{261}	2^{512}	2^{498}	2^{360}	2^{239}

4.3. Key sensitivity analysis

Cryptosystems should be highly sensitive to minor changes in keys. Moreover, verifying a powerful encryption algorithm provides minor key alterations that produce different encrypted or decrypted images. This experiment evaluates the sensitivity of the key by modifying only the y_1 value by a slight perturbation 10^{-8}, while the remaining components of the key are held constant. The correct key $K_1 = (y_1, y_2, y_3, y_4, y_5, y_6, \alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \delta)$, and the modified key $K_1' = (y_1', y_2, y_3, y_4, y_5, y_6, \alpha_1, \alpha_2, \beta_1, \beta_2, \gamma, \delta)$.

Both are used to encrypt the standard "Airplane". The results for key sensitivities are shown in Figure 3. Figure 3(a) shows the plain image "Airplane", while the ciphered images produced using the correct key K_1 and a slightly modified key K_1' are presented in Figures 3(b),(c), respectively. Subtracting the ciphertexts in Figures 3(b),(c) yields a different image shown in Figure 3(d),

confirming the high sensitivity of the proposed encryption approach to minor key alterations. Using the ciphertext of the "Airplane" image from the above experiment as an example, decryption is conducted using the original key K_1 and a slightly modified key K_1', as illustrated in Figure 4.

The image is decrypted with K_1 (Figure 4(b)), which perfectly matches the original, while decryption using K_1' (Figure 4(c)) fails, producing an unrecognizable result. These findings confirm that the proposed algorithm is highly sensitive to the secret key, and even slight variations in key parameters can result in decryption failure or incorrect decryption.

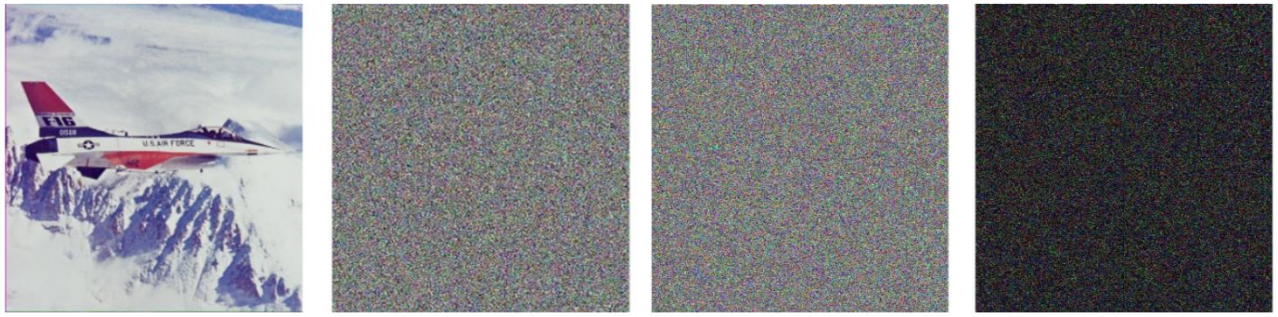

Figure 3. (a) Plain image, (b) correct encryption with K_1, (c) encryption with the wrong key K_1', and (d) the corresponding difference image between the ciphertexts in (b) and (c).

Figure 4. (a) Plain image, (b) correct decryption with K_1, (c) failed decryption with the wrong key, K_1'.

4.4. Analysis of information entropy

Information entropy (IE) is commonly utilized to evaluate the randomness of encrypted images. For an 8-bit image, the entropy is approximately 8; higher entropy indicates greater randomness, stronger resilience to statistical attacks, and more efficient protection of the information in plain images, making decryption without the correct key extremely difficult. IE is computed using the following formula:

$$H = -\sum_{i=0}^{L-1} P(i) \log_2 P(i), \tag{24}$$

where $P(i)$ refers to the probability of the occurrence of i, and L represents the number of gray

levels. Entropy is measured in bits, and for an 8-bit image, the theoretical maximum is 8 bits. Since each pixel can take 256 possible intensity levels, the ideal value of entropy is equal to $-\sum_{i=0}^{255} {}^{1}/_{256} \log_2 {}^{1}/_{256} = 8$.

The IE values are computed for plain and encrypted images to assess the proposed algorithm's efficacy. Table 4 presents the IE values for the R, G, and B channels of 256 × 256- and 512 × 512-ciphered images. These findings demonstrate that the values of IE are close to 8 for all ciphered images, indicating that the ciphered images have better statistical properties and that the suggested algorithm exhibits significant randomness. Table 5 presents the IE results for 256 × 256 and 512 × 512 color images, evaluating the proposed algorithm. These results show that the IE values of all ciphered images are close to 8, indicating that the proposed algorithm successfully produces strongly random ciphered images and disrupts statistical patterns. Using the 512 × 512 encrypted "Peppers" image, the average IE values for the proposed algorithm and existing algorithms are compared in Table 6. It is observed from Table 6 that the proposed algorithm achieves a higher average entropy than [27], the same average entropy as [39], is slightly better than [28,29], and slightly lower than [8,13]. Overall, the average entropy value of the proposed algorithm is 7.99923, which is close to 8. This indicates greater randomness in the encrypted data, thereby demonstrating robust encryption performance and effective resilience to statistical attacks.

Table 4. Entropy of RGB channels in encrypted images.

Image	Encrypted image							
	256×256				512×512			
	R	G	B	Average	R	G	B	Average
Lake	7.9975	7.9973	7.9972	7.9973	7.9993	7.9993	7.9994	7.9993
Baboon	7.9971	7.9971	7.9974	7.9972	7.9993	7.9994	7.9993	7.9993
Peppers	7.9963	7.9972	7.9974	7.9970	7.9993	7.9992	7.9992	7.9992
Airplane	7.9976	7.9974	7.9974	7.9975	7.9994	7.9993	7.9993	7.9993
Jelly	7.9972	7.9974	7.9973	7.9973	7.9992	7.9994	7.9992	7.9993
Tree	7.9974	7.9971	7.9972	7.9972	7.9993	7.9994	7.9993	7.9993
Couple	7.9972	7.9975	7.9972	7.9973	7.9994	7.9993	7.9994	7.9994
Female	7.9972	7.9972	7.9968	7.9971	7.9992	7.9994	7.9992	7.9993

Table 5. Entropy of entire color images.

Image	Plain image	Encrypted image	
		256 × 256	512 × 512
Lake	7.7622	7.9991	7.9998
Baboon	7.7626	7.9990	7.9998
Peppers	7.6781	7.9989	7.9997
Airplane	6.6660	7.9992	7.9998
Jelly	6.5869	7.9992	7.9998
Tree	7.5247	7.9990	7.9997
Couple	6.4567	7.9990	7.9998
Female	7.6005	7.9991	7.9997

Table 6. Information entropy comparison.

Algorithm	Entropy			
	R	G	B	Average
Proposed	7.9993	7.9992	7.9992	7.99923
Hosny et al. [8]	7.9992	7.9993	7.9993	7.99926
Tahiri et al. [9]	N/A	N/A	N/A	N/A
Girdhar & Kumar [12]	7.9970	7.9968	7.9970	7.9969
Muhammed et al. [13]	7.9993	7.9993	7.9993	7.9993
Liu et al. [15]	N/A	N/A	N/A	N/A
S. Ullah, et al. [25]	N/A	N/A	N/A	N/A
L. Wu et al. [27]	7.9983	7.9983	7.9971	7.9979
Hosny et al. [28]	7.9992	7.9991	7.9993	7.9992
Meng & Wu [29]	7.9993	7.9991	7.9992	7.9992
Meng et al. [39]	7.9993	7.9992	7.9992	7.99923

4.5. Histogram analysis

Histogram analysis is a crucial metric for assessing the uniformity and randomness of a ciphered image. A good encryption technique should produce an encrypted image with a flat histogram, indicating that pixel values are distributed uniformly across the full range. The histograms of the R, G, and B channels of the plain and encrypted images "Lake" and "Peppers" are shown in Figures 5 and 6, respectively. The histograms of the ciphered images are highly uniform and differ considerably from those of the plain images, confirming improved security. A flatter histogram makes it more challenging to extract statistical information of the plaintext pixels, demonstrating that the proposed algorithm effectively disrupts the structure of the plain images and increases robustness against statistical attacks.

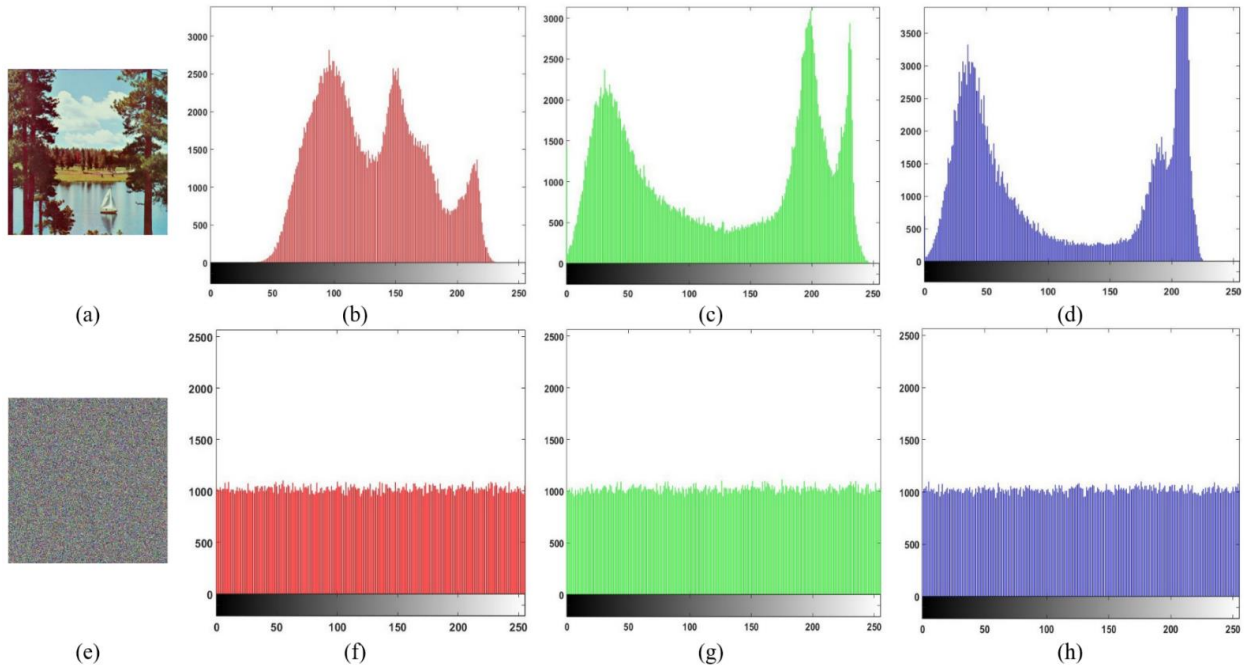

Figure 5. Plain and encrypted image "lake" histograms.

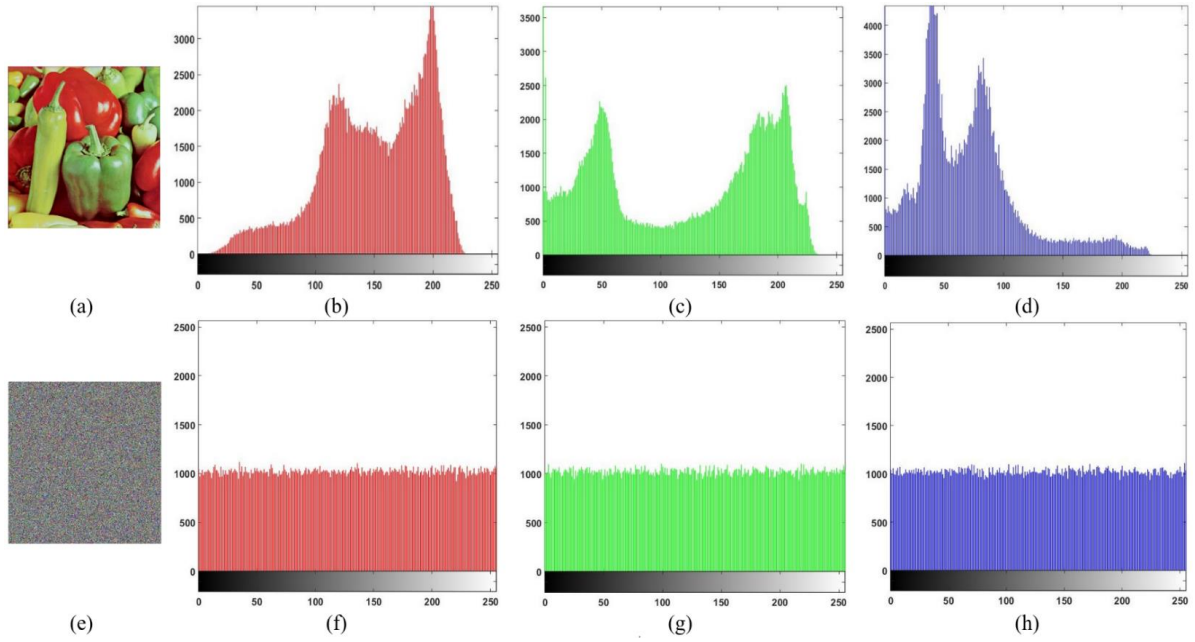

Figure 6. Plain and encrypted image "peppers" histograms.

4.6. Correlation analysis

The efficacy of the suggested encryption algorithm is verified using the correlation coefficient between neighboring pixels in the horizontal (H), vertical (V), and diagonal (D) directions. A robust encryption algorithm must effectively break the inherent statistical dependencies in plain images to reduce correlation and enhance security, because the high correlation frequently found among neighboring pixels in plain images is close to 1. The correlation of the ciphered images is approximately zero, indicating minimal association between adjacent pixels. Equation (25) is utilized to calculate the correlation coefficient R_{xy}, where $E(x)$ represents the mean, $D(x)$ refers to the variance, and $Cov(x, y)$ represents the covariance between x and y.

$$R_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{25}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{26}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} \big(x_i - E(x)\big)^2, \tag{27}$$

$$cov(x, y) = \frac{1}{N}\sum_{i=1}^{N} \big(x_i - E(x)\big)\big(y_i - E(y)\big). \tag{28}$$

We compute the correlation coefficient by randomly choosing 10,000 pairs of neighboring pixels from plain and encrypted images in all directions. Table 7 presents the correlation coefficients for each component of the test images, demonstrating that the coefficients for the cipher and original images are close to 0 and 1, respectively. Thus, the encrypted images exhibit no correlation with the plain images. Figure 7 depicts the correlation distribution for the selected plain images "Female", "Tree", and "Baboon" and their cipher images, demonstrating that the neighboring pixel distribution in the encrypted images is random, indicating a weak correlation. However, the adjacent pixel distributions in the plain images are highly correlated. These findings demonstrate that the encryption algorithm effectively breaks the relationships among neighboring pixels, highlighting its high resistance to

statistical attacks. Using the 256 × 256 "Pepper" image, the correlation coefficients for the proposed algorithm and existing algorithms are compared in Table 8. These results show that the proposed algorithm achieves better performance in reducing correlation.

Table 7. The results of the correlation coefficient.

Image		Plain			Encrypted		
		R	G	B	R	G	B
Lake	H	0.9560	0.9711	0.9707	-0.0037	0.0067	0.0030
	V	0.9538	0.9669	0.9694	-0.0063	-0.0036	0.0037
	D	0.9401	0.9524	0.9533	0.0038	-0.0015	-0.0016
Baboon	H	0.9214	0.8649	0.9071	-0.0039	0.0067	-0.0010
	V	0.8666	0.7615	0.8814	-0.0025	0.0060	0.0039
	D	0.8540	0.7330	0.8406	0.0020	0.0089	0.0008
Peppers	H	0.9628	0.9817	0.9660	-0.0075	0.0062	-0.0040
	V	0.9659	0.9818	0.9688	-0.0077	-0.0035	-0.0031
	D	0.9573	0.9681	0.9475	0.0027	0.0015	0.0088
Airplane	H	0.9708	0.9577	0.9636	0.0039	0.0036	0.0008
	V	0.9553	0.9680	0.9329	0.0045	-0.0047	0.0006
	D	0.9356	0.9338	0.9109	-0.0024	-0.0028	-0.0027
Jelly	H	0.9934	0.9942	0.9974	0.0095	0.0059	0.0102
	V	0.9940	0.9948	0.9971	0.00179	-0.0019	0.0010
	D	0.9874	0.9891	0.9939	0.0060	0.0064	-0.0104
Tree	H	0.9901	0.9925	0.9909	-0.0025	-0.0070	0.0013
	V	0.9841	0.9866	0.9854	0.0013	-0.0019	-0.0028
	D	0.9766	0.9808	0.9790	0.0043	0.0023	-0.0035
Couple	H	0.9491	0.9292	0.9180	0.0015	-0.0056	-0.0002
	V	0.9564	0.9541	0.9416	0.0020	-0.0063	-0.0042
	D	0.9165	0.9012	0.8909	0.0010	-0.0022	0.0009
Female	H	0.9946	0.9912	0.9880	0.0039	-0.0056	-0.0002
	V	0.9969	0.9956	0.9934	0.0028	-0.0063	-0.0042
	D	0.9917	0.9873	0.9813	0.0012	-0.0022	0.0069

Table 8. Correlation coefficient comparison.

Algorithm	Direction	Encrypted		
		R	G	B
Proposed	H	-0.0075	0.0062	-0.0040
	V	-0.0077	-0.0035	-0.0031
	D	0.0027	0.0015	0.0088
Hosny et al. [8]	H	0.0018	0.0033	-0.0071
	V	-0.0053	0.0026	-0.0119
	D	0.0065	0.0220	0.0165 0.0186
Tahiri et al. [9]	H	N/A	N/A	N/A
	V	N/A	N/A	N/A
	D	N/A	N/A	N/A

Continued on next page

Algorithm	Direction	Encrypted R	Algorithm G	Direction B
Muhammed et al. [13]	H	-0.001551	0.001826	-0.000948
	V	0.000685	-0.002628	-0.001039
	D	0.002326	−0.000020	0.002633
Liu et al. [15]	H	N/A	N/A	N/A
	V	N/A	N/A	N/A
	D	N/A	N/A	N/A
S. Ullah, et al. [25]	H	0.0002	0.0038	-0.0014
	V	-0.0026	-0.0024	-0.0022
	D	-0.0044	-0.0004	-0.0024
Hosny et al. [28]	H	N/A	N/A	N/A
	V	N/A	N/A	N/A
	D	N/A	N/A	N/A
Meng & Wu [29]	H	-0.0031765	0.014192	0.011638
	V	0.0077593	0.014512	-0.003386
	D	-0.0038511	0.0029724	-0.0015971
Meng et al. [39]	H	0.0081944	−0.0083731	−0.014572
	V	0.0017356	−0.0084463	0.0039651
	D	0.0061784	0.014309	0.019823

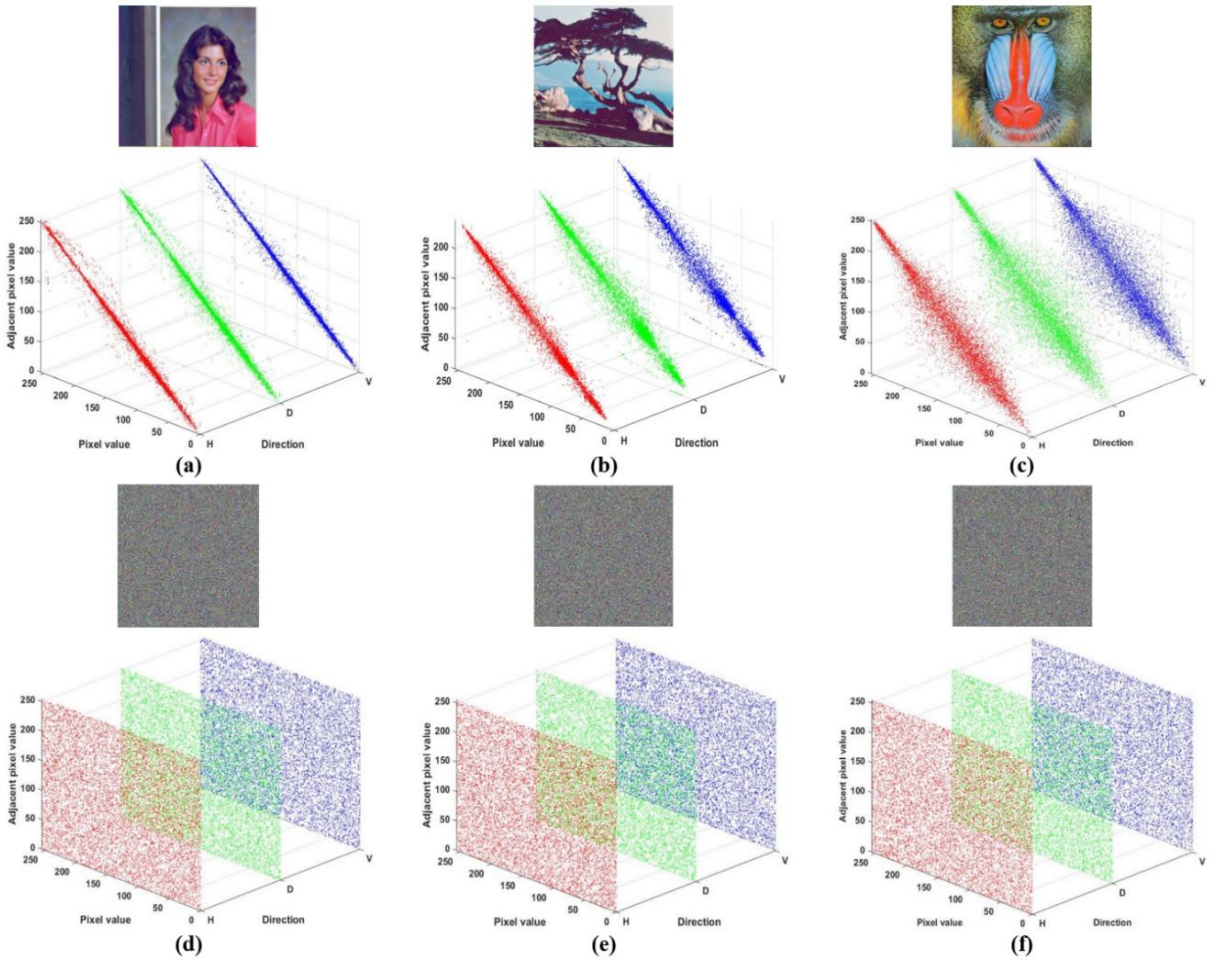

Figure 7. Correlation of adjacent pixels in (H, D, V): (a)–(c) Plain images and (d)–(f) ciphered images.

4.7. Differential attack analysis

A differential attack assesses the resistance of an encryption algorithm by encrypting two nearly identical plain images and examining the resulting differences in their ciphertexts. A secure encryption scheme should exhibit high sensitivity to minor variations in the input, such that even a negligible alteration in the plaintext produces a significantly different encrypted output. Accordingly, the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are used as evaluation metrics of sensitivity, computed for the two cipher images, C_1 and C_2, using Eqs (29) and (31), respectively.

$$NPCR = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j) \times 100\%, \tag{29}$$

$$D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j), \\ 1, & \text{if } C_1(i,j) \neq C_2(i,j), \end{cases} \tag{30}$$

$$UACI = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|C_1(i,j)-C_2(i,j)|}{255} \times 100\%. \tag{31}$$

The NPCR and UACI theoretical values are 99.6094% and 33.4635%, respectively. The NPCR and UACI results for each component across all images are summarized in Table 9. These findings indicate that UACI and NPCR are close to their theoretical values, demonstrating that the proposed encryption algorithm is highly robust against differential attacks. Using a 512×512 "Baboon" image, Table 10 compares the suggested algorithm with the methods in terms of NPCR and UACI. The suggested algorithm exhibits superior performance, indicating strong robustness against differential attacks.

Table 9. UACI and NPCR values.

Image	UACI (%)			NPCR (%)		
	R	G	B	R	G	B
Lake	33.47	33.45	33.48	99.6162	99.6185	99.6037
Baboon	33.46	33.47	33.46	99.6086	99.6105	99.6132
Peppers	33.45	33.46	33.46	99.6155	99.6231	99.6269
Airplane	33.46	33.45	33.48	99.6033	99.6040	99.6037
Jelly	33.48	33.46	33.45	99.6208	99.6201	99.5834
Tree	33.46	33.48	33.46	99.6258	99.6215	99.6265
Couple	33.55	33.52	33.50	99.6206	99.6053	99.6310
Female	33.54	33.52	33.55	99.6197	99.6082	99.5983

Table 10. UACI and NPCR for Baboon image.

Image	UACI (%)			NPCR (%)		
	R	G	B	R	G	B
Proposed	33.4641	33.4705	33.4623	99.6086	99.6105	99.6132
Hosny et al. [8]	33.4463	33.4510	33.4502	99.5943	99.6024	99.6057
Tahiri et al. [9]	N/A	N/A	N/A	N/A	N/A	N/A
Muhammed et al. [13]	33.5140	33.4495	33.4674	99.6002	99.6147	99.6052
Liu et al. [15]	N/A	N/A	N/A	N/A	N/A	N/A
Ullah, et al. [25]	N/A	N/A	N/A	N/A	N/A	N/A
L. Wu et al. [27]	33.3231	33.3644	33.2469	99.5888	99.6088	99.6033
Hosny et al. [28]	33.4024	33.4443	33.4964	99.6048	99.6162	99.5937
Meng & Wu [29]	33.5176	33.5377	33.4980	99.6201	99.6014	99.6132
Meng et al. [39]	33.4551	33.4352	33.4769	99.6063	99.6086	99.6086

4.8. Robustness analysis

The resistance of the suggested encryption algorithm is examined against Salt-and-pepper noise (SPN) and cropping attacks.

Noise attack test: This experiment is conducted to assess the proposed algorithm's resistance to SPN. First, the proposed algorithm encrypts the plain image "Jelly". Then, the ciphered image of "Jelly" is subjected to SPNs at 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, and 0.2. Finally, the decrypted images are recovered from the noisy ciphered images and are depicted in Figure 8, while Table 11 presents the corresponding PSNR values under SPN attacks.

The results indicate that even when significant noise is added to the ciphered images, the decrypted images visually preserve the essential content of the plain images and remain recognizable. Thus, the proposed algorithm is more resilient to noise attacks.

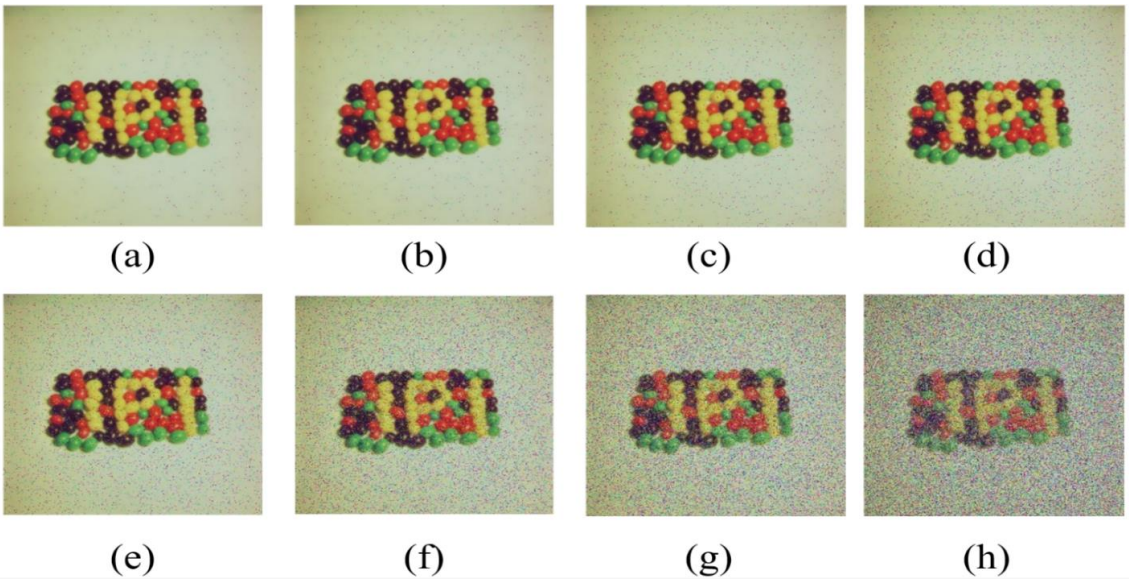

Figure 8. Noise attack analysis: (a) 0.001 SPN decrypted Jelly, (b) 0.002 SPN decrypted Jelly, (c) 0.005 SPN decrypted Jelly, (d) 0.01 SPN decrypted Jelly, (e) 0.02 SPN decrypted Jelly, (f) 0.05 SPN decrypted Jelly, (g) 0.1 SPN decrypted Jelly, and (h) 0.2 SPN decrypted Jelly.

Crop attack test: Partial loss of image data in addition to noise interference can occur during the storage or transmission of an image over networks. Thus, we assess the proposed algorithm's resilience to cropping attacks. Random cropping is applied in this experiment, where the encrypted "Jelly" image is cropped at various rates (1/6, 1/8, 1/4, and 1/2), as shown in Figure 9(a)–(d). Then, the cropped images are decrypted, as shown in Figure 9(e)–(h), and the corresponding PSNR values under cropping attacks are reported in Table 11. The results of the experiment show that the decrypted image preserves the main content of the original image and remains recognizable, even when the encrypted image has suffered significant loss of resolution. Thus, the proposed algorithm possesses significant robustness to cropping attacks.

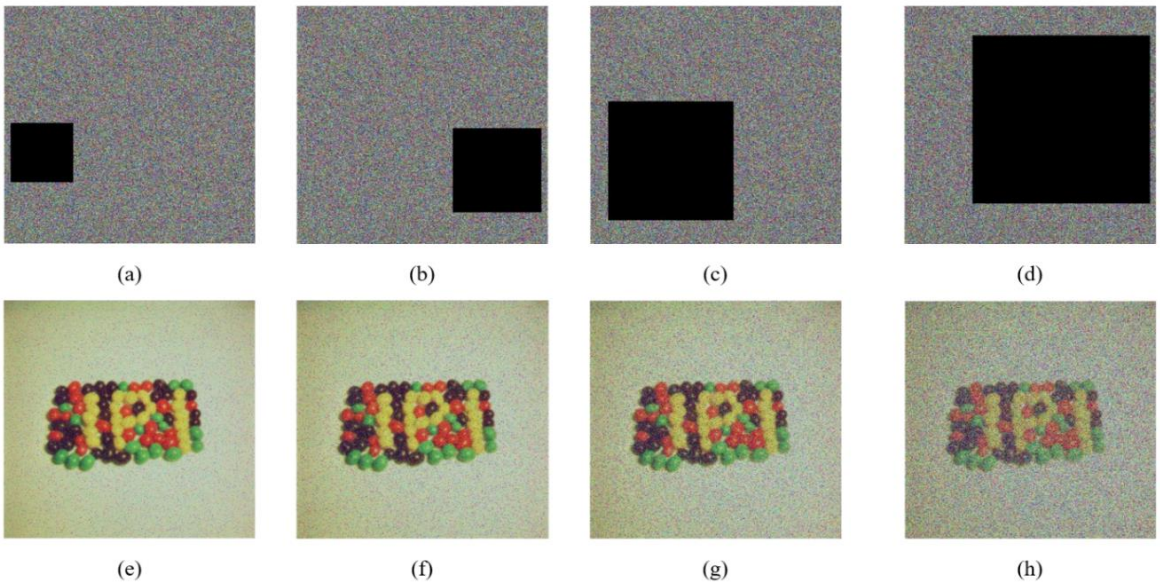

Figure 9. Cropping attack analysis: (a)–(d) Encrypted images with 1/16, 1/8, 1/4, and 1/2 data loss; and (e)–(h) decrypted images.

Table 11. PSNR with different Salt & Pepper noise and Cropping attacks.

Attacks Salt & Pepper	PSNR (dB)	Cropping attack	PSNR (dB)
0.001	35.7192	1/16	20.5786
0.002	32.3598		
0.005	28.3012	1/8	17.6031
0.01	25.5372		
0.02	22.6040	1/4	14.6067
0.05	18.6519		
0.1	15.8207	1/2	11.5992
0.2	13.0235		

4.9. Complexity and time efficiency analysis

The computational time of the encryption algorithm designed in this paper mainly consists of the time required to generate chaotic sequences using F6CS, the confusion process, and the diffusion

process. Consider a plain image, which has a size of $M \times N$. The complexity of generating a chaotic sequence using the 6D fractional-order chaotic system (6DFCS), with length $L = M \times N$, is $O(6MN)$. The confusion process, which involves pixel position permutation, has a computational complexity of $O(3MN)$. In the diffusion process, several operations are performed, including computing orthogonal Lagrange–Fourier moments (OLFMs), performing modular arithmetic operations, and applying XOR-based pixel-value diffusion. With the maximum orders, $nmax$ and $mmax$ of the OLFMs, the time complexity is $O(3\mathrm{n}max \times \mathrm{m}max \times M \times N)$, and modular operation and XOR-based pixel value diffusion have a complexity of $O(3M \times N)$. Overall, the algorithm achieves a complexity of $O(6MN) + O(3MN) + O(3\mathrm{n}max \times \mathrm{m}max \times MN) + O(3MN) \approx O(MN)$, indicating scalability with respect to image size.

To evaluate computational efficiency, the proposed algorithm is executed 30 times for plain images of varying sizes, and the encryption and decryption times were computed and averaged, as shown in Table 12. The results demonstrate the efficiency and scalability of the suggested algorithm across image sizes, indicating a good balance between security and computational cost. They also show that the execution time is acceptable, making the algorithm suitable for real-time applications.

Table 12. Encryption and decryption time.

Image size	Encryption time	Decryption time
256 × 256	0.064896	0.040391
512 × 512	0.405281	0.264830
1024 × 1024	0.960485	0.738521

5. Conclusions

In this paper, we propose a new encryption algorithm for securing color images using orthogonal Lagrange–Fourier moments and a 6D fractional-order chaotic system. In this algorithm, a 6D fractional-order chaotic system is employed to permute pixel positions during the confusion process. Subsequently, orthogonal Lagrange–Fourier moments are used to modify pixel values during the diffusion process.

Numerical experiments and security analysis show that the proposed algorithm exhibits greater key sensitivity, a larger key space, and higher security. Extensive experiments are conducted to examine statistical characteristics, including histogram, correlation, and information entropy analyses, demonstrating the proposed algorithm's resistance to statistical attacks. Furthermore, the algorithm achieves strong robustness against differential, cropping, and noise attacks. Comparing the proposed algorithm with other algorithms reveals superior encryption performance and improved security.

This indicates that the proposed algorithm is applicable to securely transmitting color images. In future work, we will extend the proposed algorithm to safeguard medical images by combining watermarking and encryption, thereby providing a higher level of security.

Author contributions

Faisal S. Alsubaei: Methodology, resources, writing–review and editing, funding acquisition; Mohamed Meselhy Eltoukhy: Resources, writing–review and editing, funding acquisition; Mohamed M. Darwish: Software, validation, formal analysis, investigation, data curation, writing–original draft preparation, writing–review and editing, visualization; Khalid M. Hosny: Conceptualization,

methodology, software, formal analysis, data curation, writing–original draft preparation, writing–review and editing, project administration. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Funding

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-25-DR-708). Therefore, the authors thank the University of Jeddah for its technical and financial support.

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. Z. Li, X. Meng, C. Bi, J. Cao, Quasi-projective synchronization of fractional-order quaternion fuzzy cellular neural networks with applications to image encryption, *J. Appl. Math. Comput.*, **72** (2026). https://doi.org/10.1007/s12190-025-02675-x
2. Y. Guo, L. Zhou, Synchronization of uncertain complex-valued coupled memristive neural networks with proportional delays via event-triggered sliding mode control for image encryption, *J. Appl. Math. Comput.*, **72** (2026). https://doi.org/10.1007/s12190-026-02789-w
3. A. Toktas, U. Erkan, S. Gao, C. Pak, A robust bit-level image encryption based on Bessel map, *Appl. Math. Comput.*, **462** (2024), 128340. https://doi.org/10.1016/j. amc.2023.128340
4. V. Verma, S. Kumar, Quantum image encryption algorithm based on 3D-BNM chaotic map, *Nonlinear Dynam.*, **113** (2025), 3829–3855. https://doi.org/10.1007/s11071-024-10403-6
5. X. B. Liu, C. Liu, Quantum image encryption scheme using independent bit-plane permutation and baker map, *Quantum Inf. Process*, **22** (2023), 262. https://doi.org/10.1007/s11128-023-04026-w
6. Z. Guo, S. H. Chen, L. Zhou, L. H. Gong, Optical image encryption and authentication scheme with computational ghost imaging, *Appl. Math. Model.*, **131** (2024), 49–66. https://doi.org/10.1016/j.apm.2024.04.012
7. X. Zhang, Y. P. Zheng, Z. L. Jiang, H. Byun, Numerical algorithms for corner-modified symmetric Toeplitz linear system with applications to image encryption and decryption, *J. Appl. Math. Comput.*, **69** (2023), 1967–1987. http://doi.org/10.1007/s12190-022-01819-7
8. K. M. Hosny, S. T. Kamal, M. M. Darwish, A novel color image encryption based on fractional shifted Gegenbauer moments and 2D logistic-sine map, *Vis. Comput.*, **39** (2023), 1027–1044. https://doi.org/10.1007/s00371-021-02382-1
9. M. A. Tahiri, H. Karmouni, A. Bencherqui, A. Daoui, M. Sayyouri, H. Qjidaa, et al., New color image encryption using hybrid optimization algorithm and Krawtchouk fractional transformations, *Vis. Comput.*, **39** (2023), 6395–6420. https://doi.org/10.1007/s00371-022-02736-3
10. O. P. Singh, K. N. Singh, A. K. Singh, A. K. Agrawal, Deep learning-based image encryption techniques: Fundamentals, current trends, challenges and future directions, *Neurocomputing*, **612** (2025). https://doi.org/10.1016/j.neucom.2024.128714

11. Q. Lai, G. Hu, U. Erkan, A. Toktas, High-efficiency medical image encryption method based on 2D Logistic-Gaussian hyperchaotic map, *Appl. Math. Comput.*, **442** (2023), 127738. https://doi.org/10.1016/j.amc.2022.127738
12. A. Girdhar, V. Kumar, Color image encryption based on planetary encoding paradigm and 5D hyper chaotic Lorenz system, *Comput. Electr. Eng.*, **118** (2024), 109352. https://doi.org/10.1016/j.compeleceng.2024.109352
13. M. S. KAYA, İ. Kenan, Knit scrambling: A novel image scrambling framework and its demonstration in image encryption, J. *Inf. Secur. Appl.*, **97** (2026), 104326. https://doi.org/10.1016/j.jisa.2025.104326
14. K. M. Hosny, S. T. Kamal, M. M. Darwish, A color image encryption technique using block scrambling and chaos, *Multimed. Tools Appl.*, **81** (2022), 505–525. https://doi.org/10.1007/s11042-021-11384-z
15. X. Liu, S. Zheng, J. Yang, Color image encryption scheme based on a novel 2D-CLCM chaotic system and RNA encoding, *Math. Comput. Simulation*, **239** (2025), 340–360. https://doi.org/10.1016/j.matcom.2025.06.009
16. H. Çelik, N. Doğan, Enhancing image encryption through fractional-order chaotic systems: A comparative analysis of six diverse models, *Inf. Sci.*, **730** (2025), 122907. https://doi.org/10.1016/j.ins.2025.122907
17. J. L. Hou, R. Xi, P. Liu, T. L. Liu, The switching fractional order chaotic system and its application to image encryption, *IEEE-CAA J. Automatic.*, **4** (2017), 381–388. https://doi.org/10.1109/JAS.2016.7510127
18. S. Iqbal, J. Wang, H. Calgan, Fractional chaotic dynamics in the Rucklidge system and its application to image encryption, *Nonlinear Dynam.*, **113** (2025), 26815–26839. http://dx.doi.org/10.1007/s11071-025-11477-6
19. A. Gokyildirim, S. Çiçek, H. Calgan, A. Akgul, Fractional-order Sprott K chaotic system and its application to biometric iris image encryption, *Comput. Biol. Med.*, **179** (2024), 108864. https://doi.org/10.1016/j.compbiomed.2024.108864
20. O. El Ogri, J. El-Mekkaoui, M. Benslimane, A novel medical image security and compression based on multiple-order fractional quaternion Hahn moments and 2D-chaotic map, *Neural Comput. Appl*., **37** (2025), 20663–20690. https://doi.org/10.1007/s00521-025-11523-9
21. M. S. Mahdi, H. Najm, A. H. Alsaeedi, R. R. N. Alogaili, Z. A. A. Alyasseri, Z. A. Abas, Hybrid image encryption model based on Zernike moments keys generation and Rubik's cube scrambling, *IEEE Access*, **14** (2026). https://doi.org/10.1109/ACCESS.2026.3682475
22. K. El-Khanchouli, H. Mansouri, A. Bencherqui, H. Karmouni, N. E. Joudar, M. Sayyouri, Proposed quaternion fractional dual-Hahn moments for color image reconstruction and encryption, *Knowl.-Based Syst*., **4** (2025). https://doi.org/10.1016/j.knosys.2025.114467
23. N. R. Zhou, L. J. Tong, W. P. Zou, Multi-image encryption scheme with quaternion discrete fractional Tchebyshev moment transform and cross-coupling operation, *Signal Process.*, **211** (2023), 109107. https://doi.org/10.1016/j.sigpro.2023.109107
24. S. Ullah, X. Liu, A. Waheed, S. Zhang, S-boxes on the combined dynamics of fractional-order chaotic systems and their application to provably secure image encryption, *Multimed. Tools Appl.*, **84** (2025), 44145–44181. https://doi.org/10.1007/s11042-025-20882-3
25. S. Ullah, X. Liu, A. Waheed, S. Zhang, S-box using fractional-order 4D hyperchaotic system and its application to RSA cryptosystem-based color image encryption, *Comput. Stand. Interfaces*, **93** (2025), 103980. https://doi.org/10.1016/j.csi.2025.103980

26. W. Alexan, M. Gabr, E. Mamdouh, R. Elias, A. Aboshousha, Color image cryptosystem based on sine chaotic map, 4D Chen hyperchaotic map of fractional order and hybrid DNA coding, *IEEE Access*, **11** (2023), 54928–54956. https://doi.org/10.1109/ACCESS.2023.3282160
27. L. M. Wu, Z. J. Tian, W. Chen, Color image encryption scheme based on 5D fractional-order complex chaotic system and eight-base DNA cubes, *Expert Syst. Appl.*, **299** (2026), 129950. https://doi.org/10.1016/j.eswa.2025.129950
28. K. M. Hosny, S. T. Kamal, M. M. Darwish, Novel encryption for color images using fractional-order hyperchaotic system, *J. Amb. Intel. Hum. Comp.*, **13** (2022), 973–988. https://doi.org/10.1007/s12652-021-03675-y
29. F. Q. Meng, G. Wu, A color image encryption and decryption scheme based on extended DNA coding and fractional-order 5D hyper-chaotic system, *Expert Syst. Appl.*, **254** (2024), 124413. https://doi.org/10.1016/j.eswa.2024.124413
30. S. H. Yan, D. F. Jiang, Y. Cui, H. B. Zhang, L. Li, J. W. Jiang, A fractional-order hyperchaotic system that is period in integer-order case and its application in a novel high-quality color image encryption algorithm, *Chaos Soliton. Fract.*, **182** (2024). https://doi.org/10.1016/j.chaos.2024.114793
31. H. A. Salman, F. M. Kamal, A. H. Kader, A novel chaotic fractional-order hyperbolic-sine-based electrical circuit: Stability analysis, simulation, and image encryption application, *Int. J. Dyn. Control.*, **13** (2025), 1–25. https://doi.org/10.1007/s40435-025-01699-2
32. F. Meng, G. Wu, J. Zhang, A novel color image encryption algorithm based on fractional-order conservative memristive hyperchaotic system and extended zig-zag transform, *J. King Saud Univ.-Com.*, **37** (2025), 151. https://doi.org/10.1007/s44443-025-00163-7
33. H. Nabil, H. Tayeb, Fractional-order supply chain finance system with conformable derivative: Chaotic dynamics, complexity analysis, and RGB color image encryption, *J. Comput. Appl. Math*., **476** (2026), 117105. https://doi.org/10.1016/j.cam.2025.117105
34. H. Nabil, H. Tayeb, Dynamical analysis of a novel fractional-order hyperchaotic map and its application for fast color image encryption, *J. Comput. Appl. Math.*, **481** (2026) 117263. https://doi.org/10.1016/j.cam.2025.117263
35. F. Yang, J. Mou, J. Liu, C. Ma, H. Yan, Characteristic analysis of the fractional-order hyperchaotic complex system and its image encryption application, *Signal Process.*, **169** (2020), 107373. https://doi.org/10.1016/j.sigpro.2019.107373
36. Z. A. S. Rahman, B. H. Jasim, Y. I. Al-Yasir, R. A. Abd-Alhameed, Efficient colour image encryption algorithm using a new fractional-order memcapacitive hyperchaotic system, *Electronics*, **11** (2022), 1505. https://doi.org/10.3390/electronics11091505
37. X. Gao, J. Yu, S. Banerjee, H. Yan, J. Mou, A new image encryption scheme based on fractional-order hyperchaotic system and multiple image fusion, *Sci. Rep*., **11** (2021), 15737. https://doi.org/10.1038/s41598-021-94748-7
38. A. S. A. Alghawli, M. M. Darwish, Secure medical image encryption for healthcare applications: A fractional 4D chaotic system and symmetry-matrix-based approach, *AIMS Math.*, **11** (2026), 6744–6776. https://doi.org/10.3934/math.2026279
39. F. Q. Meng, Z. G. Gu, A color image-encryption algorithm using extended DNA coding and zig-zag transform based on a fractional-order laser system, *Fractal Fract.*, **7** (2023), 795. https://doi.org/10.3390/fractalfract7110795
40. A. Hjouji, Orthogonal invariant Lagrange-Fourier moments for image recognition, *Expert Syst. Appl.*, **199** (2022), 117126. https://doi.org/10.1016/j.eswa.2022.117126
41. C. Singh, E. Walia, Computation of Zernike moments in improved polar configuration, *IET Image Process*., **3** (2009), 217–227. https://doi.org/10.1049/iet-ipr.2008.0142

42. M. Higazy, N. Almalki, S. Muhammad, A. Al-Ghamdi, A. A new fractional order 6D chaotic model: Study of model dynamics, system structure graph, electronic circuit realization, and fractional control, *J. Ocean Eng. Sci.*, **9** (2024), 112–125. https://doi.org/10.1016/j.joes.2022.04.002
43. USC-SIPI Database, Available from: https://sipi.usc.edu/database/.

AIMS Press