

Synchronization of a fractional-order chaotic memristive system and its application to secure image transmission

Lamia Chouchane, Hamid Hamiche, Karim Kemih,Ouerdia Megherbi, and Karim Labadi

Citation: Chin. Phys. B, 2025, 34 (**12**): 120509. DOI: 10.1088/1674-1056/ae0017
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Design of a memristive chaotic map and its adaptive regulation

Ya Wang(王亚), Xin-Lin Song(宋欣林), and Fei-Fei Yang(杨飞飞)
Chin. Phys. B, 2025, 34 (**9**): 090504. DOI: 10.1088/1674-1056/adea5c

Dynamical behaviors of a multifunctional neural circuit

Xiao-Hong Gao(高晓红), Kai-Long Zhu(朱凯龙), and Fei-Fei Yang(杨飞飞)
Chin. Phys. B, 2025, 34 (**9**): 090503. DOI: 10.1088/1674-1056/ae001b

A novel non-autonomous hyperchaotic map based on discrete memristor parallel connection

Weiping Wu(吴伟平), Mengjiao Wang(王梦蛟), and Qigui Yang(杨启贵)
Chin. Phys. B, 2025, 34 (**5**): 050503. DOI: 10.1088/1674-1056/adb8bc

Resonant tunneling diode cellular neural network with memristor coupling and its application in police forensic digital image protection

Fei Yu(余飞), Dan Su(苏丹), Shaoqi He(何邵祁), Yiya Wu(吴亦雅), Shankou Zhang(张善扣), and Huige Yin(尹挥戈)
Chin. Phys. B, 2025, 34 (**5**): 050502. DOI: 10.1088/1674-1056/adb8bb

Study and circuit design of stochastic resonance system based on memristor chaos induction

Qi Liang(梁琦), Wen-Xin Yu(于文新), and Qiu-Mei Xiao(肖求美)
Chin. Phys. B, 2025, 34 (**4**): 040502. DOI: 10.1088/1674-1056/adb268

Synchronization of a fractional-order chaotic memristive system and its application to secure image transmission

Lamia Chouchane[1,†], Hamid Hamiche[1,‡], Karim Kemih[2,§], Ouerdia Megherbi[1,¶], and Karim Labadi[3,#]

[1] *Laboratoire de Conception et Conduite des Systemes de Production (L2CSP), UMMTO, BP* 17 *RP*, 15000, *Tizi-Ouzou, Algeria*
[2] *Laboratoire dElectrotechnique et dElectronique Industrielle (L2EI), Universite de Jijel, BP* 98 *Ouled Aissa*, 18000 *Jijel, Algeria*
[3] *Laboratoire de Recherche en Eco-Innovation Industrielle et Energtique (LR2E), ECAM-EPMI/Quartz-Lab Cergy pontoise, Cergy-Pontoise Cedex, France*

(Received 30 May 2025; revised manuscript received 15 August 2025; accepted manuscript online 28 August 2025)

The dynamics of chaotic memristor-based systems offer promising potential for secure communication. However, existing solutions frequently suffer from drawbacks such as slow synchronization, low key diversity, and poor noise resistance. To overcome these issues, a novel fractional-order chaotic system incorporating a memristor emulator derived from the Shinriki oscillator is proposed. The main contribution lies in the enhanced dynamic complexity and flexibility of the proposed architecture, making it suitable for cryptographic applications. Furthermore, the feasibility of synchronization to ensure secure data transmission is demonstrated through the validation of two strategies: an active control method ensuring asymptotic convergence, and a finite-time control method enabling faster stabilization. The robustness of the scheme is confirmed by simulation results on a color image: $\chi^2 = 253/237/267$ (R/G/B); entropy ≈ 7.993; correlations between adjacent pixels in all directions are close to zero (e.g., -0.0318 vertically); and high number of pixel change rate and unified average changing intensity (e.g., 33.40% and 99.61%, respectively). Peak signal-to-noise ratio analysis shows that resilience to noise and external disturbances is maintained. It is shown that multiple fractional orders further enrich the chaotic behavior, increasing the systems suitability for secure communication in embedded environments. These findings highlight the relevance of fractional-order chaotic memristive systems for lightweight secure transmission applications.

Keywords: memristor, fractional-order chaotic system, synchronization, finite-time control, active control, cryptography, secure communications

PACS: 05.45.Pq, 05.45.Xt, 07.05.Dz, 05.45.Gg, 03.67.Dd

DOI: 10.1088/1674-1056/ae0017 **CSTR:** 32038.14.CPB.ae0017

1. Introduction

In 1971, Professor Chua introduced the concept of the memristor, recognized as the fourth element of a two-terminal circuit. He pointed out a serious shortcoming in the theoretical aspects: the absence of a relationship between electric charge and magnetic flux.[1] This transformative innovation paved the way for a device with immense potential: extreme miniaturization with reliable operation without leakage current, high integration density with processing speeds comparable to those of dynamic random-access memory. Until the prototype was built in Hewlett-Packard's labs in 2008, memristors had remained theoretical.[2,3] A key feature of the memristor is its ability to adapt its resistance to the current flowing through it and maintain that resistance even when the circuit is de-energized.

The memristor is a nanoscale device characterized by peculiar properties such as a pinched hysteresis loop in voltage-current characteristics, sensitivity to signal amplitude and frequency, and the ability to maintain states in the absence of power. With all of these characteristics, the memristor is poised to become a highly revolutionary device for improving circuit functionality and is used in a number of fields, including high-speed and secure image encryption using hyper-chaotic systems,[5] as well as the design of advanced encryption-oriented neural architectures, such as fractional-order memristive Hopfield networks implemented on field-programmable gate array platforms,[6] and in the development of efficient analog hardware accelerators for solving complex nonlinear equations, such as Riccati matrix equations, through neurodynamics-inspired architectures leveraging the parallel processing capabilities of memristive arrays.[7] These devices have also been used for accelerating computer vision tasks, notably through memristor-based parallel schemes for fast and rotation-invariant corner detection in the ORB (oriented FAST and rotated BRIEF) algorithm.[8] However, due to the difficulties of fabricating real memristors, researchers have turned to

†Corresponding author. E-mail: lamia.chouchane@ummto.dz
‡Corresponding author. E-mail: hamid.hamiche@ummto.dz
§Corresponding author. E-mail: k.kemih@gmail.com
¶Corresponding author. E-mail: ouerdia.megherbi@ummto.dz
#Corresponding author. E-mail: k.labadi@ecam-epmi.com

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

memristor emulators. Memristor emulators are real systems designed to mimic the electrical behavior of a memristor using standard electronic components so that they can be used for experimental investigations and to provide flexible means of evaluating new circuit designs.[9–14]

Chaotic systems, characterized by their sensitivity to initial conditions, ergodicity, and complex dynamical behavior, are increasingly being explored in diverse domains, including secure communications[15–19] and cryptography.[4–6,20–22] They also are important because they generate signals that are like noise, but good for encrypting images on safe social media platforms.[23] A recent study introduced a robust image encryption algorithm (IEASP) based on a 2D simplified quadratic polynomial map (2D-SQPM) and a pixel fusion strategy to achieve high encryption speed and strong security performance.[24] Another study presented a multi-image encryption algorithm that combines a fractional-order 3D Lorenz chaotic system with a 2D sinusoidally constrained polynomial hyperchaotic map (2D-SCPM). It can withstand statistical attacks quite effectively and is highly random.[25] Additionally, recent 1D chaotic systems like improved logistic maps, polyexponential hybrid maps, or spiral-transformed hybrid systems, have manifested structural simplicity while enhancing levels of security.[26–28] However, most of them are still unable to escape the confines of limited key space and are susceptible to statistical attacks. Higher dimensional chaotic systems have been proposed as solutions to this problem given that they provide a higher measure of dynamic complexity and increase their resistance to attacks.[29–31]

In recent years, the integration of memristors into chaotic systems has been increasingly examined due to their inherent nonlinearity, which is believed to enhance chaotic dynamics. The replacement of conventional nonlinear resistors has been achieved, paving the way for the use of memristors as prime elements in the design of chaotic circuits,[33–37] through which better performance and increased design freedom for parameterization have been enabled. The unique properties of memristors, such as memory and adaptability, together with the specific features of chaotic systems, are exploited in memristor-based chaotic systems (MCSs). This combination serves to increase not only the complexity but also the safety, making memristor-based chaotic systems a viable consideration for future applications. For example, it has been demonstrated in recent studies that MCSs can be applied to the design of robust image encryption algorithms.[29,38] In addition, the synchronization of these systems, which is considered essential for secure communication applications, has been explored using innovative approaches such as adaptive control[39] and fuzzy modeling,[40,41] as well as through more advanced methods including cluster synchronization in multiple nonlinearly coupled dynamical subnetworks of complex networks with nonidentical nodes,[42] cluster synchronization of two-layer nonlinearly coupled multiplex networks with multi-links and time-delays,[43] and fixed/predefined-time synchronization of coupled memristor-based neural networks subjected to stochastic disturbances.[44] Recent findings have also contributed to the research background, such as a chaotic secure communication scheme and networks based on synchronization of double-layered and multiple complex networks,[45] fixed-time control for interlayer synchronism under double-delay memristor-based neural networks and its application in secure communication,[46] and a robust synchronization-based chaotic secure communication scheme with double-layered and multiple hybrid networks.[47]

Two synchronization techniques were used in our study. The first, active control, refers to a synchronization method in which a control input is deliberately designed and applied to the slave system based on the error between master and slave states. The slave system is actively guided by this control signal to follow the master's dynamics, thus ensuring convergence and synchronization. The second technique is finite-time synchronization: a process in which the synchronization error between the states of the master and slave systems tends exactly towards zero in a finite time, rather than converging asymptotically towards zero. Unlike conventional methods, in which synchronization is only achieved asymptotically and in infinite time, this approach ensures that synchronization is attained within a bounded time interval, regardless of the initial conditions. This makes it particularly interesting for applications requiring fast, precise response times.

Fractional-order chaotic systems represent a significant advance in chaos theory. Unlike integer-order systems, fractional-order systems offer greater flexibility in the generation of pseudorandom sequences, with the ability to generate more diverse and less predictable trajectories.[48–54] This property makes them particularly suitable for applications in the field of encryption, where complexity and security are necessary conditions. However, fractional systems are still underutilized in color image encryption, where most research has focused on grayscale images.

Nevertheless, memristive and fractional-order chaotic systems offer many advantages in terms of dynamic complexity and security. Some challenges remain. Low-dimensional chaotic systems, while simple in construction, raise the issues of key space and resistance to cyber-exploitation. The memristive systems, meanwhile, will require emulators to be practical, which also imposes constraints on design and performance testing. Finally, fractional-order systems, while more useful, are fraught with parameter optimization and computational resources issues.

At the same time, the strength of encryption schemes should be evaluated using both theoretical metrics and mod-

ern cryptanalysis. Several recent studies have demonstrated that certain algorithms, though not trivial, are vulnerable to specific types of attacks. For example, in a recent study, the authors of Ref. [55] examined the authenticity of a medical privacy protection scheme (MPPS) based on chaos and deoxyribonucleic acid (DNA) coding. The MPPS was originally developed to encrypt color digital imaging and communications in medicine (DICOM) images. While the original authors claimed that the MPPS could withstand chosen-plaintext attacks, this study proves this claim to be false. The authors present an efficient chosen-plaintext attack capable of recovering the secret key using only two pairs of plain and cipher images. Finally, they question the scheme's other marketed attributes as they relate to current sound cryptographic principles. In their article, the authors of Ref. [56] perform a detailed analysis of an image encryption scheme that uses a Feistel network and dynamic DNA encoding. They identify several vulnerabilities, primarily in the design of the secret key and the encryption process itself. They also propose a viable chosen-plaintext attack and amendments to enhance the original scheme's security. In their paper,[57] Feng *et al*. thoroughly analyze an image encryption scheme based on the 2D logistic-adjusted sine map, identifying various security and design flaws. The authors propose enhancements to the scheme in conjunction with a chosen-plaintext attack algorithm that can completely recover the original image without knowledge of the secret key.

Additionally, in their study,[58] Zhao *et al*. examine a chaotic DNA-coding-based image encryption scheme that depends on the key for plaintext. They demonstrate a vulnerability in the scheme that can be exploited via a chosen-plaintext attack and propose a cryptanalysis method to recover the plaintext message and an improved chaotic map to enhance the security of the scheme. In Ref. [59], the authors cryptanalyze a chaotic image encryption method based on various discrete dynamical maps and conclude that security in the system is compromised with one round of encryption and a classic permutation-diffusion structure.

Our work attempts to address some of the aforementioned gaps by presenting a novel fractional-order chaotic memristive system applied to secure color image encryption. The method is implemented using multiple cryptographic keys while the properties of fractional systems are exploited to improve the security and efficiency of the overall encryption process.

Our contribution to this work can be summarized as follows:

(i) A novel fractional-order chaotic memristor system is designed and analyzed based on the Shinriki oscillator. The proposed chaotic oscillator incorporating a memristor is developed by substituting the nonlinear positive conductance element of the original Shinriki circuit with a memristor emulator. More dynamic complexity and nonlinear behavior are introduced into the system through the inclusion of a memristor emulator.

(ii) Two synchronization methods are used to the fractional-order chaotic system: (a) active control to achieve synchronization and (b) finite-time control synchronization. The second synchronization approach has specifically been applied to our newly developed fractional-order memristor-based chaotic system, designed to ensure secure transmission of a color image.

(iii) The application of the proposed system to secure communication is demonstrated by encrypting a color image. Multiple cryptographic keys are integrated into our approach, which significantly enhances the security and robustness of the system against attacks.

(iv) Enhancing chaotic behavior via the multiple fractional orders of the fractional order memristive chaotic system: It is shown that the use of multiple fractional orders in the system further enriches its chaotic behavior, making it very suitable for applications in secure communication.

The rest of this paper is organized as follows. Section 2 deals with the description of the fractional-order memristive chaotic system under study. Section 3 presents the studied synchronization approaches, including finite-time control and active control, along with their numerical results. The implementation of this proposed system into a secure communication scheme is explored in two subparts of Section 4, devoted to the encryption and decryption processes. In Section 5, we conduct a comprehensive statistical evaluation of the encryption security, examining it through perturbation analysis, key space assessment, and noise robustness analysis based on peak signal-to-noise ratio (PSNR). Finally, Section 6 concludes the paper and outlines further research perspectives.

2. Description of the studied fractional-order system

The present work starts from the theory of the Shinriki oscillator[60] and proposes a new chaotic system. The system is derived by replacing the nonlinear positive conductance of the Shinriki circuit with a memristor emulator, as described in Ref. [61], while varying some relevant parameters. The system exhibits very complex dynamics and a very high degree of nonlinearity, it also provides multiple parameters, increasing the secret key space and enhancing security in cryptographic uses. The system produces very high-quality chaotic signals and very rich strange attractors, making it particularly useful for secure communication and transmission applications.

This system is extended to a fractional-order model where the results show its extended potential with respect to more dynamic complexity and chaotic applications.

The fractional-order system can be expressed as follows:

$$\begin{cases} \dfrac{\mathrm{d}^{k_1} y_1}{\mathrm{d}t^{k_1}} = l_1 y_1 + (l_2 + l_3 y_4^2 + l_4 y_4^4)(y_2 - y_1), \\ \dfrac{\mathrm{d}^{k_2} y_2}{\mathrm{d}t^{k_2}} = -y_3 - l_5 y_2 - (l_6 + l_7 y_4^2 + l_8 y_4^4)(y_2 - y_1), \\ \dfrac{\mathrm{d}^{k_3} y_3}{\mathrm{d}t^{k_3}} = y_2, \\ \dfrac{\mathrm{d}^{k_4} y_4}{\mathrm{d}t^{k_4}} = l_9 + l_{10} y_4^2 + l_{11} y_4^4 + l_{12}(y_2 - y_1) + l_{13} y_4 (y_2 - y_1). \end{cases} \tag{1}$$

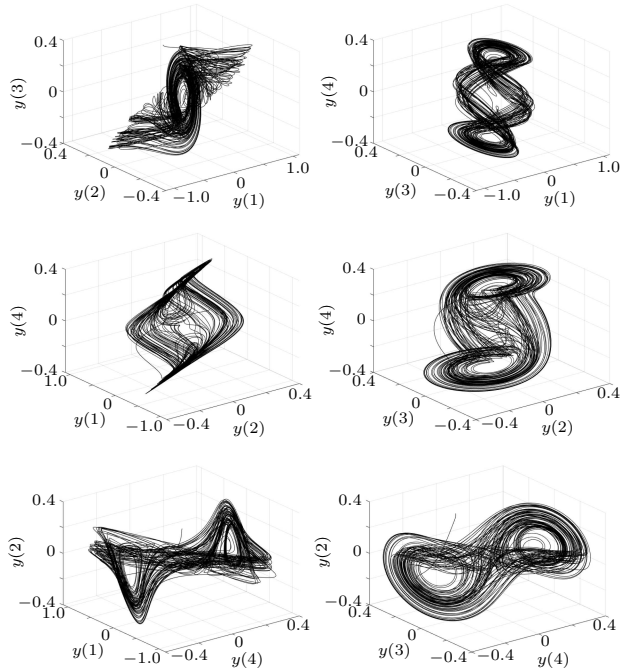


Fig. 1. The 3D representation shows the unpredictable nature of the behavior of the four states of the fractional-order chaotic memristive system which are described by the system (1) with the initial conditions $(y(0) = [0.1, 0.2, 0.3, 0.04]^{\mathrm{T}})$.

Table 1. The l_n parameters of system (1).

l_n	Values
l_1	1.5
l_2	0.3
l_3	9
l_4	330
l_5	0.07
l_6	0.3
l_7	2.1
l_8	2.6
l_9	10^{-7}
l_{10}	0.00003
l_{11}	0.008
l_{12}	0.5
l_{13}	0.3

The behavior analysis of this fractional-order system shows that the four states are unpredictable, as indicated in Fig. 1 with the initial conditions are $[y_1(0); y_2(0); y_3(0); y_4(0)] = [0.1, 0.2, 0.3, 0.04]^{\mathrm{T}}$. Given that the fractional orders k_n satisfy $(0 < k_n < 1)$, their respective values are as follows: $k_1 = 0.97$, $k_2 = 0.98$, $k_3 = 0.99$, $k_4 = 0.96$. Moreover, the values of l_n are given in Table 1.

3. Synchronization approaches: Finite-time control and active control

Synchronization between master and slave systems essentially consists in aligning their dynamic states, which is done through a specific coupling or control strategy. In other words,

$$\forall \quad S_i(0); M_i(0) : \lim_{t \to \infty} \|S_i - M_i\| = 0, \tag{2}$$

with M_i as the master system and S_i as the slave system.

In this paper, we will adopt two different methods using the previously defined fractional-order system for synchronization: finite-time control and active control. The following are the specifications of the master and slave systems involved in the synchronization effort: These systems are respectively described by

$$\begin{cases} D_t^{k_1} M_1 = l_1 M_1 + (l_2 + l_3 M_4^2 + l_4 M_4^4)(M_2 - M_1), \\ D_t^{k_2} M_2 = -M_3 - l_5 M_2 - (l_6 + l_7 M_4^2 + l_8 M_4^4)(M_2 - M_1), \\ D_t^{k_3} M_3 = M_2, \\ D_t^{k_4} M_4 = l_9 + l_{10} M_4^2 + l_{11} M_4^4 + l_{12}(M_2 - M_1) \\ \qquad + l_{13} M_4 (M_2 - M_1), \end{cases} \tag{3}$$

$$\begin{cases} D_t^{k_1} S_1 = l_1 S_1 + (l_2 + l_3 S_4^2 + l_4 S_4^4)(S_2 - S_1) + V_1, \\ D_t^{k_2} S_2 = -S_3 - l_5 S_2 - (l_6 + l_7 S_4^2 + l_8 S_4^4)(S_2 - S_1) + V_2, \\ D_t^{k_3} S_3 = S_2 + V_3, \\ D_t^{k_4} S_4 = l_9 + l_{10} S_4^2 + l_{11} S_4^4 + l_{12}(S_2 - S_1) \\ \qquad + l_{13} S_4 (S_2 - S_1) + V_4. \end{cases} \tag{4}$$

Here V_i $(i = 1, \ldots, 4)$ is the control input applied to achieve synchronization.

From the equations of the master (transmitter) and slave (receiver), we derive the following error system, based on the principle of Eq. (2). This system serves as the foundation for analyzing the synchronization process, capturing the differences between the states of the master and slave systems. Based on Eq. (2), the error system is formulated as follows:

$$\begin{cases} D_t^{k_1} e_1 = D_t^{k_1} S_1 - D_t^{k_1} M_1, \\ D_t^{k_2} e_2 = D_t^{k_2} S_2 - D_t^{k_2} M_2, \\ D_t^{k_3} e_3 = D_t^{k_3} S_3 - D_t^{k_3} M_3, \\ D_t^{k_4} e_4 = D_t^{k_4} S_4 - D_t^{k_4} M_4. \end{cases} \tag{5}$$

By replacing Eqs. (3) and (4) and simplifying, we obtain

the error system

$$\begin{cases} D_t^{k_1} e_1 = (l_1 - l_2)e_1 + l_2 e_2 \\ \qquad + l_3\left[S_4^2(S_2 - S_1) - M_4^2(M_2 - M_1)\right] \\ \qquad + l_4\left[S_4^4(S_2 - S_1) - M_4^4(M_2 - M_1)\right] + V_1, \\ D_t^{k_2} e_2 = -e_3 - (l_5 + l_6)e_2 + l_6 e_1 \\ \qquad - l_7\left[S_4^2(S_2 - S_1) - M_4^2(M_2 - M_1)\right] \\ \qquad - l_8\left[S_4^4(S_2 - S_1) - M_4^4(M_2 - M_1)\right] + V_2, \\ D_t^{k_3} e_3 = e_2 + V_3, \\ D_t^{k_4} e_4 = l_{10}(S_4^2 - M_4^2) + l_{11}(S_4^4 - M_4^4) \\ \qquad + l_{12}e_2 - l_{12}e_1 \\ \qquad + l_{13}\left[S_4(S_2 - S_1) - M_4(M_2 - M_1)\right] + V_4. \end{cases} \tag{6}$$

3.1. Finite-time synchronization

This refers to a synchronization process where the states of a slave system converge to those of the master system within a finite time,[62] after which the synchronization error becomes zero and remains stable.

Next, several important definitions and lemmas are given, which will be used for further theoretical development.

Definition 1[63,64] Systems (3) and (4) are said to achieve finite-time synchronization if there exists a constant $t_F > 0$ such that the limit of the error $e_i = S_i - M_i$ as t approaches t_F equals zero, that is, $\lim_{t \to t_F} |e_i| = 0$, and $|e_i| = 0$ for $t \geq t_F$, $i = 1, 2, 3, 4$.

This means that after a certain time $t_{sync} = t_F$, the errors e_i converge to zero and remain zero for all $t \geq t_F$.

Lemma 1[65] With $D_p^t = [D_t^k, D_t^k, \ldots, D_t^k]$, $0 < k \leq 1$, the trivial solution of the general chaotic system $D_t^\alpha X(t) = F(X(t))$, where $F : \mathbb{R}^n \to \mathbb{R}^n$, is asymptotically stable if there exists a positive definite Lyapunov function $V(x_1, x_2, \ldots, x_n)$ such that for all $t > 0$, $D_t^k V(X(t)) < 0$.

Lemma 2[66] For any $X(t) \in \mathbb{R}^n$ and $k \in]0, 1]$, for all $t > 0$,

$$\tfrac{1}{2} D_t^k \left(X^{\mathrm{T}}(t)X(t)\right) \leq X^{\mathrm{T}}(t) D_t^k (X(t)).$$

In this section, we will consider V_i to be defined as follows:

$$\begin{cases} V_1 = -\beta_1 e_1 - \beta_2 \mathrm{sgn}(e_1)|e_1|^\lambda \\ \qquad - l_3\left[S_4^2(S_2 - S_1) - M_4^2(M_2 - M_1)\right] \\ \qquad - l_4\left[S_4^4(S_2 - S_1) - M_4^4(M_2 - M_1)\right] + l_{12}e_4, \\ V_2 = -\beta_1 e_2 - \beta_2 \mathrm{sgn}(e_2)|e_2|^\lambda \\ \qquad + l_7\left[S_4^2(S_2 - S_1) - M_4^2(M_2 - M_1)\right] \\ \qquad + l_8\left[S_4^4(S_2 - S_1) - M_4^4(M_2 - M_1)\right] - (l_2 + l_6)e_1, \\ V_3 = -\beta_1 e_3 - \beta_2 \mathrm{sgn}(e_3)|e_3|^\lambda, \\ V_4 = -\beta_1 e_4 - \beta_2 \mathrm{sgn}(e_4)|e_4|^\lambda - l_{10}(S_4^2 - M_4^2) \\ \qquad - l_{11}(S_4^4 - M_4^4) - l_{12}e_2 - l_{13}S_2 e_4 - l_{13}M_4 e_2 \\ \qquad + l_{13}S_4 e_1 + l_{13}M_1 e_4, \end{cases} \tag{7}$$

where β_1 and β_2 are positive constants, $0 < \lambda < 1$. Their respective values are $\beta_1 = 1$, $\beta_2 = 4$, $\lambda = 0.9$. $\mathrm{sgn}(\cdot)$ represents the sign function. Their respective values are $\beta_1 = 1$, $\beta_2 = 4$, $\lambda = 0.9$.

Proof Let the Lyapunov function be defined as

$$V = \frac{1}{2}\sum_{i=1}^{4} e_i^2, \tag{8}$$

$$V = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2 + e_4^2). \tag{9}$$

As stated in Lemma 2, we obtain

$$DV = \frac{1}{2}(De_1^2 + De_2^2 + De_3^2 + De_4^2). \tag{10}$$

After substituting V_i into system (6), the simplified error system obtained is as follows:

$$\begin{cases} D_t^{k_1} e_1 = (l_1 - l_2)e_1 + l_2 e_2 + l_{12}e_4 \\ \qquad - \beta_1 e_1 - \beta_2 \mathrm{sgn}(e_1)|e_1|^\lambda, \\ D_t^{k_2} e_2 = -e_3 - (l_5 + l_6)e_2 - l_2 e_1 \\ \qquad - \beta_1 e_2 - \beta_2 \mathrm{sgn}(e_2)|e_2|^\lambda, \\ D_t^{k_3} e_3 = e_2 - \beta_1 e_3 - \beta_2 \mathrm{sgn}(e_3)|e_3|^\lambda, \\ D_t^{k_4} e_4 = -l_{12}e_1 - \beta_1 e_4 - \beta_2 \mathrm{sgn}(e_4)|e_4|^\lambda. \end{cases} \tag{11}$$

By evaluating this function along the trajectories of equations (11), we obtain the expression for its time derivative

$$\begin{aligned} \dot{V} &= e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 + e_4\dot{e}_4 \\ &= e_1\Big[(l_1 - l_2)e_1 + l_2 e_2 \\ &\quad + l_3\left[S_4^2(S_2 - S_1) - M_4^2(M_2 - M_1)\right] \\ &\quad + l_4\left[S_4^4(S_2 - S_1) - M_4^4(M_2 - M_1)\right] + V_1\Big] \\ &\quad + e_2\Big[-e_3 - (l_5 + l_6)e_2 + l_6 e_1 \\ &\quad - l_7\left[S_4^2(S_2 - S_1) - M_4^2(M_2 - M_1)\right] \\ &\quad - l_8\left[S_4^4(S_2 - S_1) - M_4^4(M_2 - M_1)\right] + V_2\Big] \\ &\quad + e_3\Big[e_2 + V_3\Big] \\ &\quad + e_4\Big[l_{10}(S_4^2 - M_4^2) + l_{11}(S_4^4 - M_4^4) \\ &\quad + l_{12}e_2 - l_{12}e_1 \\ &\quad + l_{13}\left[S_4(S_2 - S_1) - M_4(M_2 - M_1)\right] + V_4\Big]. \end{aligned} \tag{12}$$

After substituting V_i into system (12) and simplifying, the final expression is obtained as follows:

$$\begin{aligned} \dot{V} &= (l_1 - l_2)e_1^2 - (l_5 + l_6)e_2^2 \\ &\quad - \beta_1 e_1^2 - \beta_2 \mathrm{sgn}(e_1)|e_1|^\lambda \\ &\quad - \beta_1 e_2^2 - \beta_2 \mathrm{sgn}(e_2)|e_2|^\lambda \\ &\quad - \beta_1 e_3^2 - \beta_2 \mathrm{sgn}(e_3)|e_3|^\lambda \\ &\quad - \beta_1 e_4^2 - \beta_2 \mathrm{sgn}(e_4)|e_4|^\lambda, \end{aligned} \tag{13}$$

which according to Lemma 2 leads to

$$\dot{V} = (l_1 - l_2)e_1^2 - (l_5 + l_6)e_2^2$$

$$
\begin{aligned}
&-\beta_1 e_1^2-\beta_2|e_1|^{\lambda+1}-\beta_1 e_2^2-\beta_2|e_2|^{\lambda+1}\\
&-\beta_1 e_3^2-\beta_2|e_3|^{\lambda+1}-\beta_1 e_4^2-\beta_2|e_4|^{\lambda+1}\\
=\;&-(\beta_1-l_1+l_2)e_1^2-(\beta_1 l_5+l_6)e_2^2-\beta_1 e_3^2-\beta_1 e_4^2\\
&-\beta_2\left(|e_1|^{\lambda+1}+|e_2|^{\lambda+1}+|e_3|^{\lambda+1}+|e_4|^{\lambda+1}\right)\\
\leq\;&-\beta_2\left(|e_1|^{\lambda+1}+|e_2|^{\lambda+1}+|e_3|^{\lambda+1}+|e_4|^{\lambda+1}\right). \quad (14)
\end{aligned}
$$

Based on Lemma 1, our system is asymptotically stable.

3.2. Numerical results

In this section, numerical simulations are presented to validate the finite-time synchronization results. The l_n parameters for the master system (3) are specified in Table 1, with initial conditions set as follows: $M_1(0)=0.1$, $M_2(0)=0.2$, $M_3(0)=0.3$, and $M_4(0)=0.04$. For the response system (4), the parameters are identical to those of the master system, with the exception of the response system-specific parameters β_1, β_2, and λ, which are set to 1, 4, and 0.9, respectively. The initial conditions of the response system are specified as follows: $S_1(0)=0.01$, $S_2(0)=0.24$, $S_3(0)=0.02$, and $S_4(0)=0.4$. The two systems reach a state of synchronization at $t_{\text{sync}}=0.84$.

The synchronization of the transmitter and receiver systems is shown in Fig. 2, demonstrating the performance of the synchronization. Figure 3 shows the time evolution of the error system with a gradual decrease of the error over time, confirming the stability of the system and confirming the efficiency of the synchronization scheme.

For a better view of the finite-time synchronization process, the oblique line used as a reference for convergence is drawn in Fig. 4. The line illustrates a fast alignment of the system states, which validates the finite-time convergence and shows the strong nature of the adopted synchronization approach.

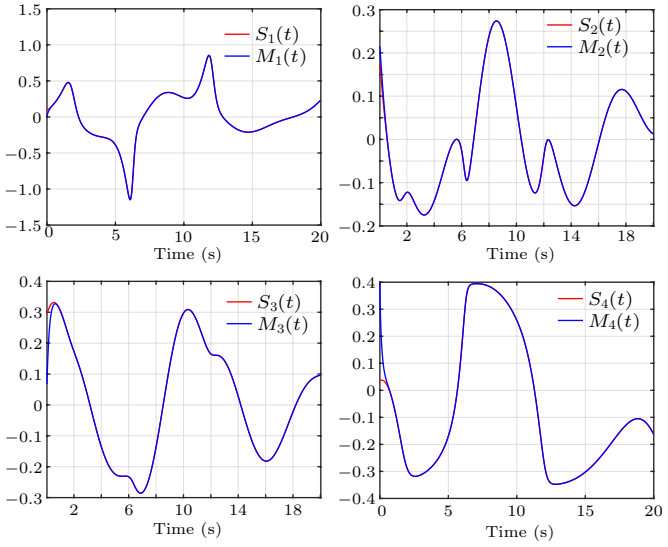


Fig. 2. Synchronized behavior between master and slave systems using the finite-time synchronization method.

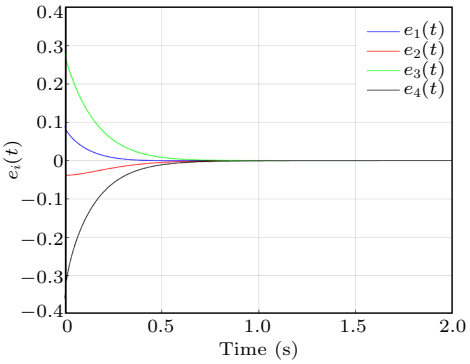


Fig. 3. Time evolution of error system.

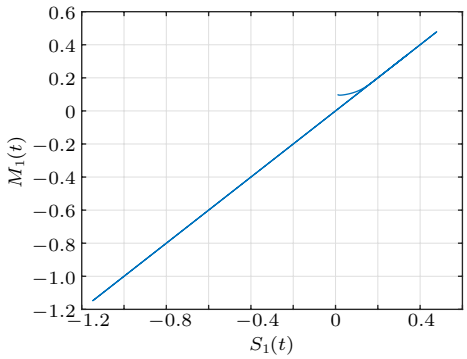


Fig. 4. Oblique line representing the evolution of synchronization: M_1 as a function of S_1.

3.3. Synchronization using active control

In this section, the error system remains identical to the one described in system (6). However, an adjustment is introduced in this method regarding the definition of V_i, which is now specified with system (15):

$$
\begin{cases}
V_1=-l_3\left[S_4^2(S_2-S_1)-M_4^2(M_2-M_1)\right]\\
\qquad -l_4\left[S_4^4(S_2-S_1)-M_4^4(M_2-M_1)\right]\\
\qquad +l_{12}e_4+(l_2-l_1-\alpha^2)e_1,\\
V_2=l_7\left[S_4^2(S_2-S_1)-M_4^2(M_2-M_1)\right]\\
\qquad +l_8\left[S_4^4(S_2-S_1)-M_4^4(M_2-M_1)\right]\\
\qquad -(l_2+l_6)e_1,\\
V_3=-e_3,\\
V_4=-l_{10}(S_4^2-M_4^2)-l_{11}(S_4^4-M_4^4)\\
\qquad -l_{12}e_2-l_{13}S_2e_4\\
\qquad -l_{13}M_4e_2+l_{13}S_4e_1+l_{13}M_1e_4.
\end{cases} \quad (15)
$$

Our error system (6) thus becomes

$$
\begin{cases}
D_t^{k1}e_1=-\alpha^2e_1+l_2e_2+l_{12}e_4,\\
D_t^{k2}e_2=-e_3-(l_5+l_6)e_2-l_2e_1,\\
D_t^{k3}e_3=e_2-e_3,\\
D_t^{k4}e_4=-l_{12}e_1-e_4.
\end{cases} \quad (16)
$$

3.4. Numerical results

In this section, numerical simulations are also presented to validate the synchronization results using the active control method. The parameters l_n of the drive system (3) and the initial conditions are the same as those used in the finite-time synchronization method. In addition, for the response system (4), the parameters are identical to those of the drive system with the α parameter set to 3. The initial conditions for the response system are set as follows: $S_1(0) = 0.25$, $S_2(0) = 0.18$, $S_3(0) = 0.22$, and $S_4(0) = 0.05$. The synchronized behavior of the drive and response systems is shown under the active control method, having highlighted their dynamic alignment, as shown in Fig. 5. Figure 6 shows the time evolution of the error system, which shows a gradual and consistent decrease, thus confirming the stability and effectiveness of the active control strategy.

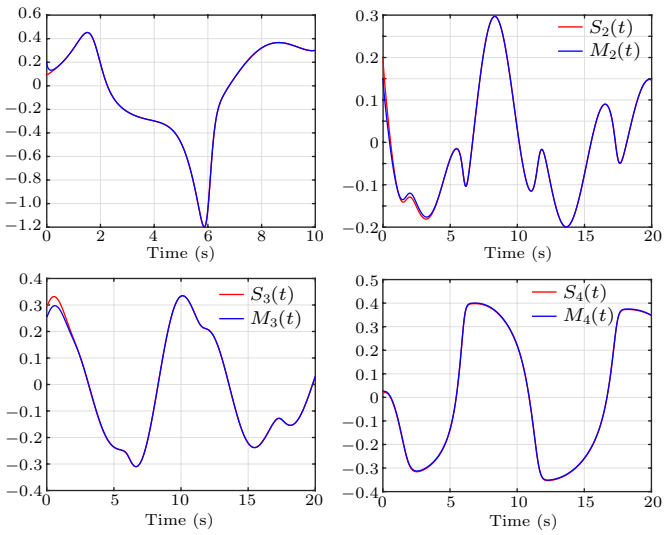


Fig. 5. Synchronized behavior between master and slave systems using the active control synchronization method.

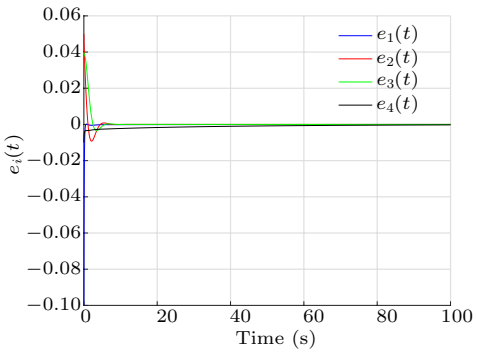


Fig. 6. Temporal evolution of system's error $e_i(t)$ using active control, showcasing error reduction and validating the stability and synchronization effectiveness.

In addition, in Fig. 7, the oblique line denotes the synchronization obtained by the active control approach. The efficient approximation of the system trajectories to the defined synchronization line provides a clear visual confirmation that underpins the reliability and effectiveness of the active control approach in maintaining synchronization.

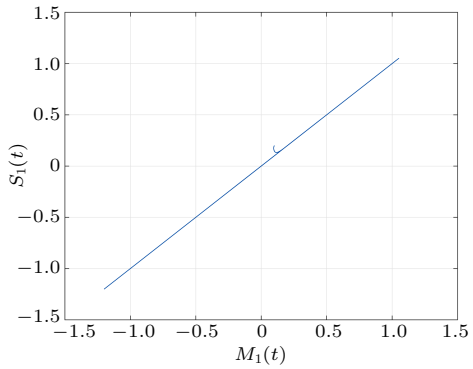


Fig. 7. Oblique line illustrating the dynamics of synchronization using active control: M_1 as a function of S_1.

The discussion in this paper covers two synchronization methods, called active control synchronization and finite time control synchronization. The results show that each of these synchronization techniques would serve to synchronize the uncontrolled chaotic system in question. However, of the two techniques, finite-time control synchronization is more advantageous because it allows shorter synchronization times, although it is obviously more complicated due to its multiple keys, and thus increases the security of the system. Therefore, for the purposes of the rest of this paper, this technique is chosen as the method to be used to secure communication.

4. Cryptographic system design: Encryption and decryption

As a specific cryptographic case study, we used the proposed fractional-order memristive chaotic system to demonstrate its potential application in secure communication (i.e., the encryption and decryption of a color image). The encryption and decryption procedures were performed entirely in MATLAB, taking advantage of the dynamic properties of the proposed system. In this way, increased security is achieved by using keys generated from the complex and unpredictable chaotic behavior of the system.

4.1. Cryptographic approach

Masterslave structures with finite-time control mechanisms have been used for the encryption process. This approach was chosen because of its ability to generate rich, complex keys, surpassing those obtained with the active control method. Due to this richness, the level of security in the encryption process is extremely increased, making the successful execution of cryptographic attacks more difficult. Such features are crucial to strengthening the security of the system against cryptanalytic attacks, including brute-force and differential attacks.

Moreover, reliably synchronizing the chaotic sequences to be used as encryption keys ensures that both the master and slave systems can faithfully reproduce the encryption keys on the receiver side. This improves decryption reliability and creates stronger properties that protect the encryption plan against cryptanalyst attacks.

To illustrate the design logic and justification of each encryption step, the complete process is described below.

4.1.1. Step 1: Image loading and channel separation

The first step is to load the original color image and separate it into red, green, and blue (RGB) channels. These channels can be independently manipulated, which can both enhance the diffusion and confusion properties of the algorithm. This step guarantees that the image data is properly structured before entering the encryption pipeline. Each channel will be subjected to a separate chaotic transformation. However, the RGB channels will be independently manipulated yet synchronously.

4.1.2. Step 2: Preprocessing and initial pixel permutation

This stage is designed to disrupt the spatial and spectral correlations commonly present in natural images. It includes: (i) A geometric tilting transformation that will help disrupt standard spatial regularity. (ii) Permutations of the RGB channels that promote confusion across channels. (iii) A global pixel permutation using a random index to erratically represent visual information. (iv) A transformation of the image into a 1D vector with zero-padding for size alignment. (v) A conversion into an 8-bit binary representation that allows for bitwise operations.

These operations will collectively increase the entropy and decrease the statistical predictability of the image. Both of these properties are helpful in resisting histogram and correlation attacks.

4.1.3. Step 3: Generation of chaotic sequences from the fractional-order memristive chaotic system at the transmitter level

The chaotic transmitter (master) generates two additional independent sequences based on the dynamics of the proposed memristor-based model. These sequences are highly sensitive to the initial conditions and system parameters and normalized within a set range and quantized into 8-bit values.

This step guarantees that the keystreams have excellent randomness and non-repeatability, both of which are necessary for cryptographic strength and key sensitivity.

4.1.4. Step 4: Dual chaotic XOR-based encryption

The binary image vector is encrypted through a two-stage XOR process. The first XOR step takes the image and mixes it with one chaotic sequence. The second XOR step mixesd

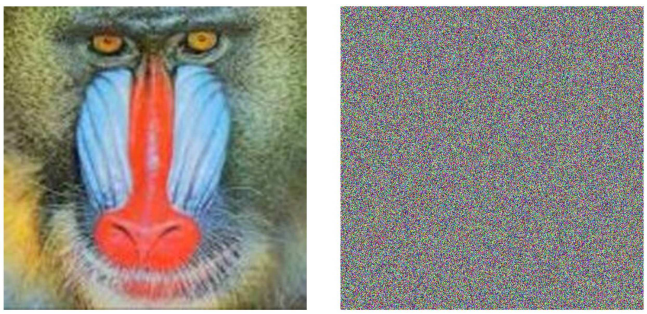


Fig. 8. (a) The image in its original form. (b) The image after encryption.

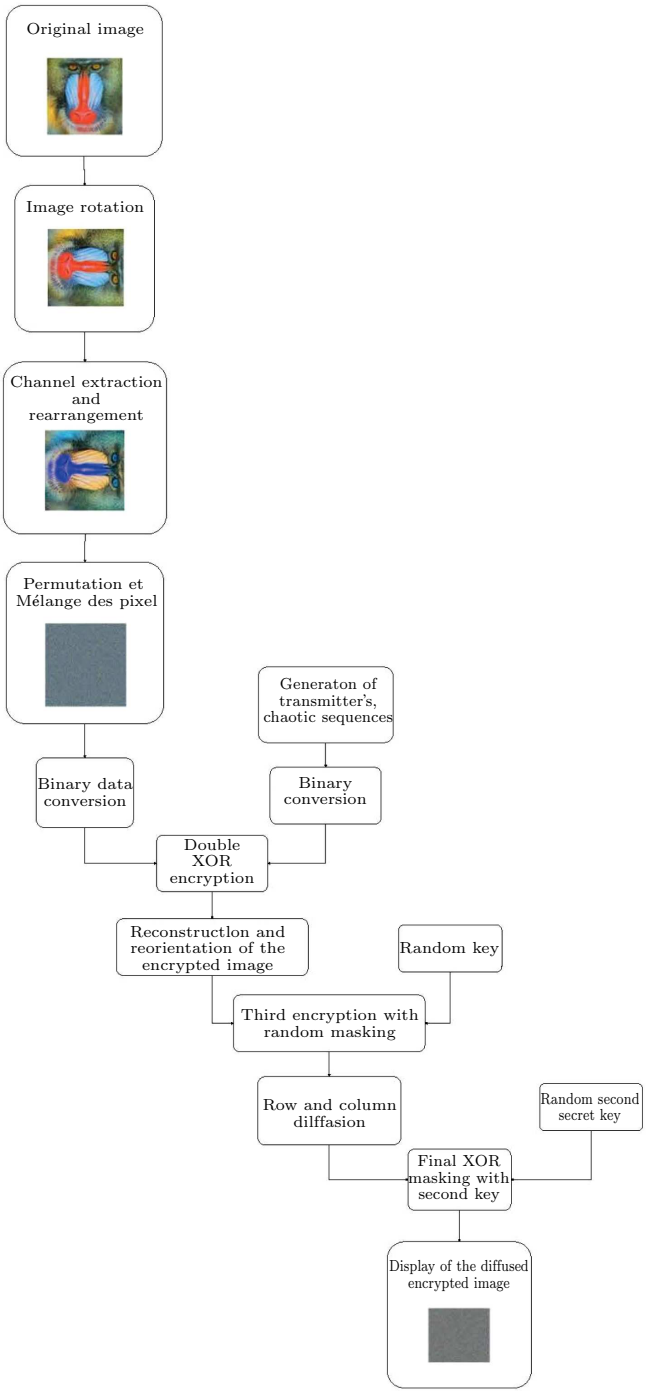


Fig. 9. Synoptic diagram outlining the encryption steps.

the resulting image with a second chaotic sequence, yielding deeper diffusion.

This two-layer structure of the encryption algorithm facilitates increased confusion and diffusion, resulting in an algorithm that is resilient to both differential and chosen-plaintext attacks.

4.1.5. Step 5: Image reconstruction and spatial transformation

After the encryption step, the binary data is converted back into decimal numbers. These values are then reshaped into a 3D matrix that restores the structure of the image. Finally, a spatial symmetry operation, such as a horizontal or vertical flip, is performed to further obscure residual spatial correlations, thereby enhancing security and completing the encryption procedure.

4.1.6. Step 6: Final masking with a random matrix

To augment the encryption process, a final masking layer is introduced. A random matrix of the same size as the encrypted image is created using a fixed seed (an example using the randi function in MATLAB is included). The last XOR operation between the image and the mask will add an additional layer of confusion to the encryption process so that unauthorized decryption is nearly impossible without knowledge of the masking key.

Figure 8 below shows the original and encrypted images, demonstrating the effectiveness of the proposed encryption process. With an encryption time of approximately 0.583539 s, showing the whole proposed scheme produced a good level of security with efficiency.

In Fig. 9, we present a synoptic diagram summarizing the above-mentioned steps.

4.2. Decryptographic approach

In order to decrypt, it is necessary to employ the receiver equations in reverse order of the encryption procedure to recover the original image. To properly recover the original image, decryption must be executed in reverse order using synchronized chaotic sequences and secret keys in relation to the encryption. It is important to highlight that a coherent decryption has been designed for every correspondence to the encryption step so that the decryption is the correct inverse of the encryption, to allow for data recovery, confidentiality, and computational efficiency.

The following describes the complete decrypting framework including a further description of the objectives and purpose of each step.

4.2.1. Step 1: Initial reverse masking and data reorganization

The decryption process begins by removing the final encryption layer applied during masking. A fresh random matrix is recreated using the same fixed seed as the last encryption mask. Then, the original encrypted image and the recreated mask are combined using the XOR function to reverse the final encryption layer.

Next, a backward pixel scan is performed horizontally and vertically, simulating the reversal of the spatial scrambling introduced by the encryption process. Next, a second XOR operation is performed using the first secret key to remove the first layer of masking.

To restore the image's structural integrity, a double reflection transformation is applied, first in the vertical orientation and then in the horizontal orientation, to align the spatial layout of the image. The new image is then converted to a 1D vector with zero-padding added to the prefix so that the linearized output aligns with the spatial layout of the original image. Lastly, the image is converted to standard 8-bit binary, which allows for bit-oriented decryption in subsequent stages.

4.2.2. Step 2: Receiver's chaotic sequence generation and binary conversion

At this stage, the receiver uses a system similar to the transmitter to generate its own chaotic sequences. This way, synchronization has been achieved as well as consistency.

These sequences are produced by iterating nonlinear equations initialized with shared secret parameters. These parameters are scaled, quantized, and converted into 8-bit binary sequences. These sequences serve as decryption keys that allow for the exact undoing of the diffusion operations that took place during encryption.

4.2.3. Step 3: Dual XOR-based bitwise decryption

With the binary image vector and synchronized chaotic sequences, the decryption process uses a two-phase XOR operation: First, the binary image data undergoes a XOR operation with the first chaotic binary sequence. Second, the binary image data is subjected to a XOR with the second chaotic binary sequence.

The XOR duality cancels out two layers of confusion generated during the encryption process. The final result is a binary vector data that is identical to the original pre-encrypted image data.

4.2.4. Step 4: Image reconstruction and channel realignment

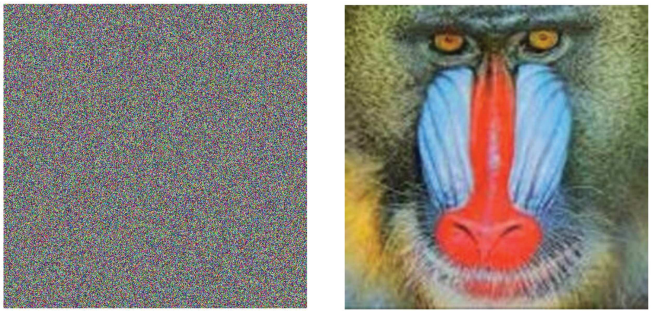
The decrypted binary vector is converted to decimal values, and the padding introduced during encryption is removed. Then, the vector is returned to its original form as a 2D array.

Then, a reverse permutation is applied to reverse the pixel order. The RGB channels are then reordered and inverse geometric transformations are applied to realign the image to its previous orientation.

The resulting image is a fully decrypted image that is visually and structurally identical to the original input. The

recovered image is shown in Fig. 10. Decryption took only 9.682926 s, which clearly demonstrates the method's efficiency and the practicality of secure image transmission.

Figure 11 illustrates a synoptic diagram that summarizes the steps previously detailed.



(a) (b)

Fig. 10. (a) The encrypted image. (b) The clear image.

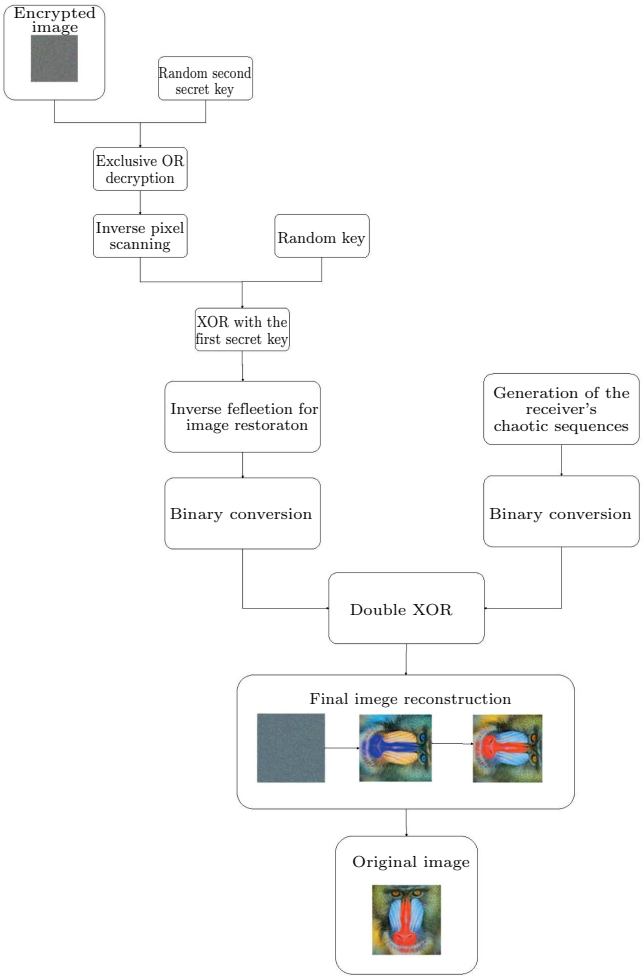


Fig. 11. Synoptic diagram summarizing the decryption steps.

5. Evaluation and analysis of encryption robustness

Cryptanalysis significantly impacts the evaluation of the scalability and overall security of cryptographic functions against malicious attacks. In an increasingly hostile digital world, the ability to evaluate encryption algorithms is important for determining the level of resistance to unauthorized decryption of data. The purpose is not only to verify a system's ability to prevent unauthorized data removal but also to protect the secrecy and integrity of conveyed information.

We strongly support all cryptographic indicators, namely, entropy, number of pixel change rate (NPCR), unified average changing intensity (UACI), key space, etc. with sound theoretical or experimental justification, as suggested in recent literature.[24,25] This makes the credibility of the proposed method more apparent and easier to replicate.

In considering the above, we will examine additional important evaluation parameters to help us assess the efficacy, stability, and strength of the proposed encryption scheme.

5.1. Statistical analysis

Statistical analysis of image encryption evaluates the effectiveness of algorithms using histogram, entropy, pixels correlation and χ^2 test. These tools allow us to measure data distribution, uncertainty and pixel relationships, ensuring that the encrypted image is difficult to analyze, and thus guaranteeing its security.

5.1.1. Histogram analysis with the χ^2 test

Histogram analysis involves observing the amounts of gray levels of the pixels in the images before and after encryption. As suggested in recent studies,[67] histogram uniformity is a widely accepted indicator of good encryption, revealing the algorithm's capacity to hide statistical features of the original image. Secure encryption would produce a histogram that is random and uniform, with no visible patterns. A comparison between the histograms of the original image and the encrypted image confirms the correctness of the encryption and detects possible weaknesses.

Figure 12 shows a histogram of the encrypted image. When compared to that of the original image, represented in Fig. 13, it is noticeable that the histogram of the encrypted image is nearly uniform and completely distinct from that of the original image, thus ensuring the effectiveness of the encryption applied to the image. Furthermore, Fig. 14 illustrates the histogram of the recovered image, which closely matches that of the original image, demonstrating the successful decryption process and the ability to restore the original image without significant distortion.

The χ^2 test is a statistical method for measuring the difference between an observed distribution and an expected (theoretical) distribution. It is an essential tool for determining whether differences between these two distributions are due to chance or whether they are significant. Furthermore, to confirm the effectiveness of our encryption, a χ^2 test was integrated into the histogram analysis to assess the randomness

of the pixel distribution in the encrypted image. This test compares the histogram of the encrypted image with an ideal uniform distribution, where each pixel value should have an equal probability of occurring. As shown in recent work by Setiadi *et al.*, the χ^2 test is widely employed alongside histogram uniformity to assess encryption randomness and resistance against statistical attacks.[68]

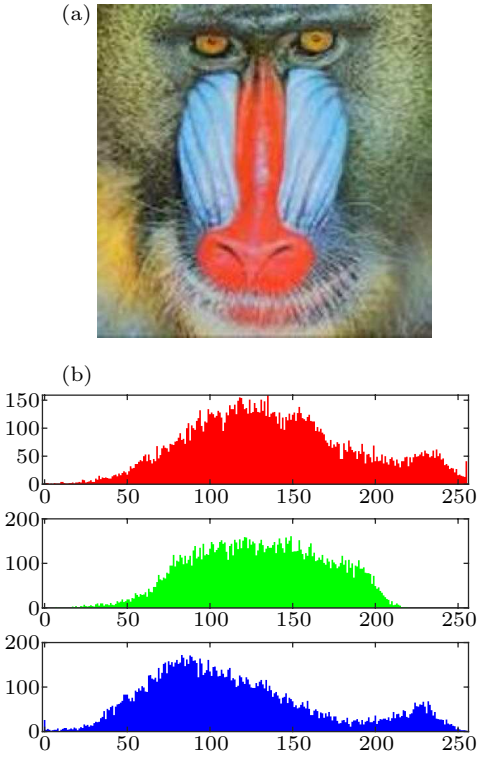


Fig. 12. (a) Original image. (b) Histogram of the original image.

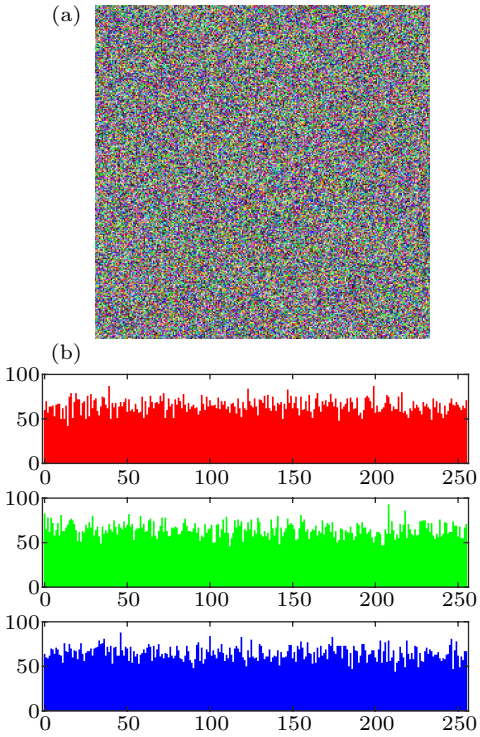


Fig. 13. (a) Encrypted image. (b) Histogram of the encrypted image.

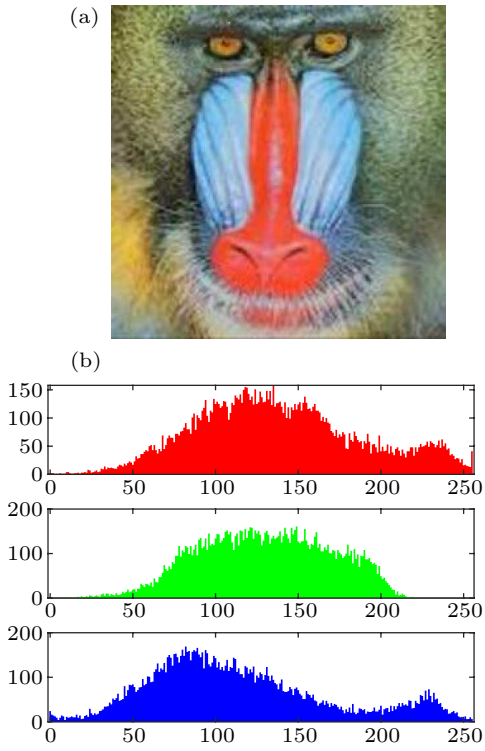


Fig. 14. (a) Recovered image. (b) Histogram of the recovered image.

The χ^2 statistic is defined as

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}, \tag{17}$$

where O_i is the observed frequency of pixel value i in the encrypted image, E_i is the expected frequency assuming a uniform distribution of all possible pixel intensities, calculated as $E_i = N/n$, where N is the total number of pixels and n is the number of possible pixel values.

The χ^2 values for each color channel (red, green, blue) of the encrypted image were calculated and compared with the expected values for a uniform distribution. The obtained results are presented below: red channel: χ^2 value of 253.0586; green channel: χ^2 value of 237.4902; blue channel: χ^2 value of 267.3613. The histogram for each channel (red, green, blue) is illustrated in Fig. 15.

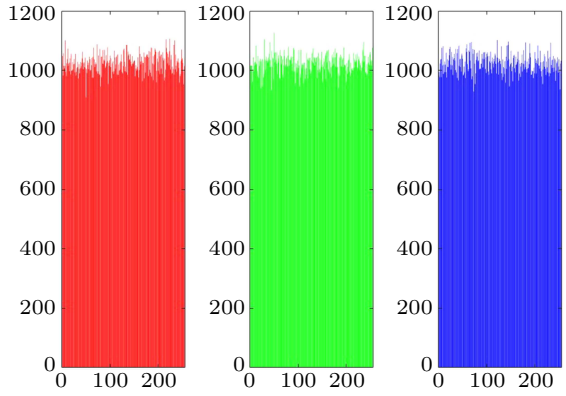


Fig. 15. Histograms of the encrypted image for each color channel (red, green, blue).

These values, approaching 256, indicate an almost uniform distribution of gray levels in the encrypted image, thus indicating the effectiveness of the encryption in terms of the statistical distribution of pixels.

Table 2 presents the χ^2 test values calculated for different image sizes and channels of encrypted images, allowing the evaluation of their conformity to a perfectly random pixel distribution.

The χ^2 values are all below the critical value, confirming that the distribution of encrypted pixels is statistically uniform and close to random noise, thereby strengthening resistance to statistical attacks.

Table 2. Values of χ^2 for different image sizes.

Size	Name	Channel	χ^2
256×256	4.1.04	Red	231.4766
	4.1.04	Green	237.2031
	4.1.04	Blue	280.7656
512×512	2.1.02	Red	274.9551
	2.1.02	Green	215.3809
	2.1.02	Blue	271.7344
1024×1024	2.2.01	Red	273.4267
	2.2.01	Green	235.9356
	2.2.01	Blue	265.4887

5.1.2. Entropy analysis

Entropy, as measured by Shannon's index,[69] quantifies the level of randomness in an image, a key indicator of the security of encryption. An entropy value close to 8 bits per pixel indicates a random and uniform distribution of pixels, which strengthens the encryption's resistance to statistical attacks. As also suggested in recent studies,[67] a high entropy close to 8 is essential to prevent statistical inference in encrypted images.

The formula for entropy is as follows:

$$H = -\sum_{i=0}^{255} p_i \log_2(p_i), \tag{18}$$

where p_i is the probability of the occurrence of grayscale level i in the image.

In our study, we calculated the entropy for each color channel (red, green, blue) of the original and encrypted images using MATLAB. The results are shown below. Red channel: entropy of 7.9993 for the encrypted image vs. 7.5728 for the original image. Green channel: entropy of 7.9993 for the encrypted image vs. 7.2531 for the original image. Blue channel: entropy of 7.9992 for the encrypted image vs. 7.5652 for the original image.

These results show that the entropy of the encrypted image is close to the ideal value of 8 bits per pixel, indicating strong pixel randomization in each color channel. In contrast, the original image has significantly lower entropy, suggesting a more exploitable and less secure structure. These observations, obtained using MATLAB, confirm the effectiveness of the applied encryption, ensuring better protection of visual data.

Information entropy utilizing the plaintext and ciphertext images of different size and color channels is shown in Table 3. This demonstrates the proposed fractional-order memristor-based system produces ciphertexts of overall entropy approaching the theoretical upper limit of 8 (for all sizes or types of images).

Table 3. Statistical information entropy results for different image sizes.

Image size	Image name	Channel	Plaintex	Ciphertext
256×256	4.1.04	Red	7.2549	7.9974
	4.1.04	Green	7.2704	7.9974
	4.1.04	Blue	6.7825	7.9969
512×512	2.1.02	Red	7.4061	7.9992
	2.1.02	Green	7.4188	7.9994
	2.1.02	Blue	6.5931	7.9993
1024×1024	2.2.01	Red	7.7575	7.9998
	2.2.01	Green	7.3387	7.9998
	2.2.01	Blue	6.9561	7.9998

Table 4 presents a detailed comparison of the entropy values obtained with our approach and those derived from other algorithms. This analysis enables us to assess the ability of our method to generate an intensity distribution close to randomness, an essential criterion for secure image encryption. By comparing these results, we highlight the performance and potential improvements of our algorithm compared with existing solutions.

Table 4. Comparison of entropy results between our algorithm and other methods.

Method	Red channel	Green channel	Blue channel
Our algorithm	7.9993	7.9993	7.9992
[70]	7.9881	7.9882	7.9875
[71]	7.9992	7.9993	7.9993
[72]	7.9800	7.9800	7.9800
[73]	7.9993	7.9993	7.9993
[74]	7.9994	7.9992	7.9992
[75]	7.9994	7.9994	7.994

5.1.3. Pixel correlation analysis

The study of adjacent pixel correlation is very important when evaluating an encryption algorithm for hiding the visual structure of an image. Adjacent unencrypted pixels of an image typically exhibit a strong form of correlation due to continuity in patterns and textures. However, effective encryption schemes need to break such relationships, thus producing a distribution that approximates a random pixel distribution where the correlation is approximately 0 in all directions (horizontal, vertical, and diagonal).[53]

The formula for the correlation between two sets of pixels X and Y is given by

$$r = \frac{E[(X-E(X))(Y-E(Y))]}{\sigma_X \sigma_Y}, \tag{19}$$

where $E(X)$ and $E(Y)$ are the means of the pixel sets X and Y. σ_X and σ_Y are their standard deviations. $E[(X-E(X))(Y-E(Y))]$ is the covariance between X and Y.

Table 5. Correlation coefficients of adjacent pixels in the original and encrypted images.

Image	Direction	Red	Green	Blue
Original image	Horizontal	0.9947	0.9917	0.9944
	Vertical	0.9946	0.9904	0.9953
	Diagonal	0.9898	0.9824	0.9899
Encrypted image	Horizontal	0.0384	–0.0073	–0.0167
	Vertical	–0.0159	–0.0126	–0.0318
	Diagonal	0.0097	–0.0099	–0.0298

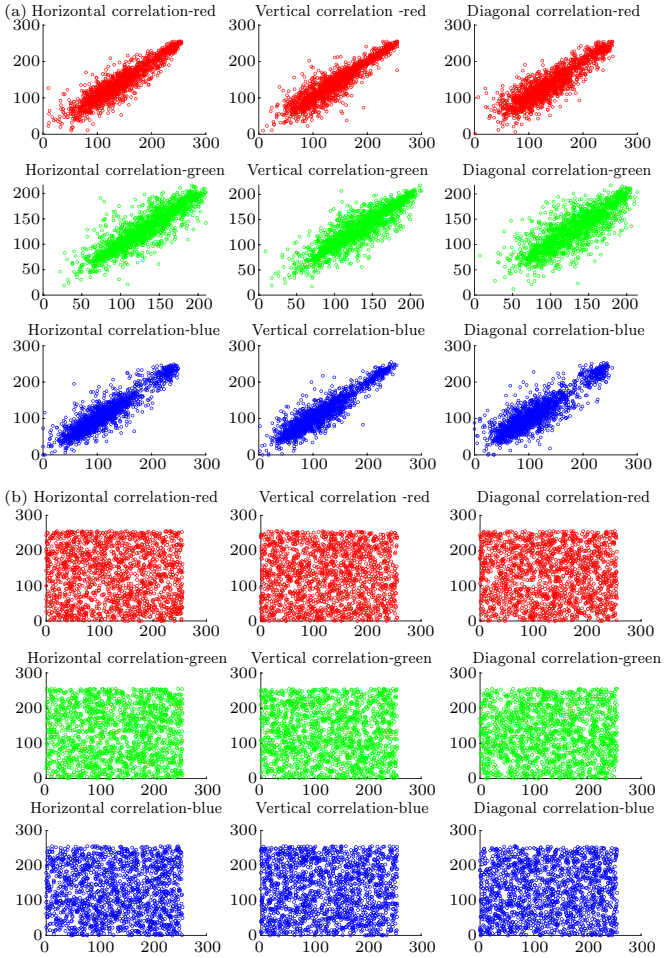


Fig. 16. Adjacent pixels correlation of (a) the original image and (b) the encrypted image.

In our study, we computed the correlation of adjacent pixels in these three directions for both the original and encrypted images. The results obtained using MATLAB are summarized in Table 5. Figure 16 shows the correlation distribution of pixels horizontally, vertically, and diagonally in the original image and in the encrypted image of the three channels.

The computed values indicate that the original image showed a strong correlation between adjacent pixels, which is consistent with natural structures in visual data. However, after exclusive encryption, the correlation drops drastically and approaches 0, suggesting that the correlation between adjacent pixels has been canceled and proving that the encryption process was successful. The very low correlation prevents any attempt to extract information from the image's redundancies, providing higher security against statistical attacks.

Tables 6 and 7 compare the correlation coefficients of adjacent pixels in encrypted images, comparing the performance of our algorithm with that of previous studies.

The correlation results show that our algorithm effectively reduces the correlations between adjacent pixels in encrypted images. Compared to Refs. [71,72,74,75], our method achieves correlation coefficients close to zero or even negative, indicating better pixel diffusion after encryption.

Table 6. Comparison of correlation coefficients (part 1).

Direction	Channel	Our algorithm	[71]
Horizontal	Red	0.0122	0.0023
	Green	0.0243	–0.0022
	Blue	0.0143	0.0086
Vertical	Red	–0.0602	0.0068
	Green	0.0089	0.0023
	Blue	0.0143	–0.0013
Diagonal	Red	–0.0105	0.0012
	Green	0.0364	0.0018
	Blue	0.0202	0.0042

Table 7. Comparison of correlation coefficients (part 2).

Direction	Channel	[72]	[74]	[75] (avrg)
Horizontal	Red	0.0016	–0.0016	
	Green	0.0021	0.0000	0.0013
	Blue	0.0542	0.0002	
Vertical	Red	0.0063	–0.0011	
	Green	–0.0024	0.0020	0.0025
	Blue	0.0063	0.0000	
Diagonal	Red	–0.0044	–0.0024	
	Green	–0.0065	–0.0012	0.0010
	Blue	–0.0044	0.0004	

Table 8. Correlation coefficients between adjacent pixels for plaintext and ciphertext images of different sizes.

Image size	Image name	Channel	Plaintext			Ciphertext		
			Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
256×256	4.1.04	Red	0.9808	0.9904	0.9666	–0.0003	–0.0344	–0.0069
	4.1.04	Green	0.9644	0.9853	0.9484	–0.0134	–0.0010	–0.0201
	4.1.04	Blue	0.9577	0.9784	0.9295	–0.0228	0.0028	0.0412
512×512	2.1.02	Red	0.8304	0.8446	0.7579	0.0010	–0.0193	–0.0210
	2.1.02	Green	0.7920	0.7447	0.6739	–0.0474	–0.0446	–0.0069
	2.1.02	Blue	0.7297	0.7154	0.6486	–0.0124	–0.0120	0.0020
1024×1024	2.2.01	Red	0.9243	0.9295	0.9031	0.0035	–0.0012	0.0042
	2.2.01	Green	0.9142	0.9169	0.8987	–0.0125	0.0024	0.0330
	2.2.01	Blue	0.9035	0.9112	0.8865	0.0134	–0.0006	0.0056

Our approach differs significantly from others in the vertical and diagonal dimensions because we tend to have lower coefficients. The most pertinent example is the red channel in the vertical direction; -0.0602 is the value of our algorithm, while others are closer to zero. This reflects that our approach actually creates a large perturbation in the configuration of adjacent pixels, and thus has greater resistance to statistical attacks.

To demonstrate the adaptability of our system to different image sizes, Table 8 presents the horizontal, vertical, and diagonal correlation coefficients of adjacent pixels, calculated for each color channel before and after encryption.

In conclusion, these results confirm the effectiveness of our encryption scheme in reducing the correlation between neighboring pixels, making the encrypted text more secure against statistical analysis attacks.

5.2. Perturbation analysis

Undoubtedly, perturbation analysis is an obvious tool for evaluating the resilience of image encryption algorithms against attacks that rely on small changes in the input data. The number of pixel change rate (NPCR) and UACI are two of the most widely used measures. They represent the percentage of pixels affected by a small change and the average intensity of the induced changes, respectively. These measures are important to ensure that the slightest change in the original image causes a very pronounced change in the encrypted image, thwarting any differential attack attempt.

In this study, we carried out an experimental evaluation of the cryptosystem's sensitivity by introducing a minimal modification to the source image. Specifically, we modified a single pixel in the original image — the value of the pixel located at position $(49, 123)$ in the third channel was 101 before alteration, and increased to 153 after modification. Following this alteration, we applied the encryption algorithm to both images — the original and the modified — generating two distinct encrypted versions.

To know how much this change affects the encryption process, we calculated NPCR and UACI values on the obtained encrypted images. NPCR and UACI are used to assess the sensitivity and diffusion of an encryption algorithm. High values indicate that the algorithm is resistant to differential attacks. The experimental results obtained confirm the ability of the system to produce statistically independent images, even with a minimal variation of the input image to the system, further increasing its robustness against differential attacks.

5.2.1. Analysis of NPCR

The NPCR is an important metric that measures how sensitive an encryption scheme is to minor changes in the original image. In this sense, the NPCR is the percentage of pixels that change when two encrypted images are generated from source images with only one pixel differing between them. A robust encryption scheme should produce an NPCR between 99.5% and 99.8% for color images, demonstrating strong diffusion properties and a high level of protection against differential attacks.[79]

The NPCR is defined by the following formula:

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%, \tag{20}$$

where M and N are the dimensions of the image. $D(i,j)$ is defined as

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

where C_1 and C_2 represent the two encrypted images obtained from slightly different original images.

The evaluation of our encryption algorithm provided the results in Table 9 for each color channel.

Table 9. NPCR values for each color channel.

Channel	Obtained NPCR	Ideal value
Red (R)	99.61%	~ 99.5%–99.8%
Green (G)	99.59%	~ 99.5%–99.8%
Blue (B)	99.62%	~ 99.5%–99.8%

These results demonstrate that our encryption method ensures high sensitivity to input image variations. Even a minimal change results in an almost total transformation in the encrypted image, making any differential attack attempt ineffective.

5.2.2. Analysis of UACI

The UACI is a metric that indicates how sensitive an encryption scheme is to tiny variations in the original image. It measures the average difference in intensity between two encrypted images that differ by only one pixel in the plaintext image. For an encryption algorithm, it must have a UACI value close to 33.33% for maximum diffusion and strong resistance to differential attacks.[79] The UACI is defined as follows:

$$\mathrm{UACI} = \frac{1}{M \times N} \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\%, \tag{21}$$

where M and N are the dimensions of the image, $C_1(i,j)$ and $C_2(i,j)$ are the pixel values at coordinates (i,j) in two encrypted images derived from slightly different plaintext images.

The UACI values obtained from our experiments are presented in Table 10.

Table 10. UACI results for each color channel.

Channel	Obtained UACI	Ideal value
Red (R)	33.40%	~ 33.33%
Green (G)	33.40%	~ 33.33%
Blue (B)	33.51%	~ 33.33%

These values, computed using MATLAB, demonstrate that our encryption method ensures excellent diffusion of initial variations. The obtained results are very close to the ideal 33.33% value, confirming the robustness of the encryption against differential attacks.

Table 11 presents a comparative analysis of resistance to differential attack between our method and those presented in other studies.

Table 12 presents the NPCR values obtained for images of various resolutions, allowing a comparative evaluation between our algorithm and recent state-of-the-art methods. It should be noted that the 256×256 test image corresponding to reference datasets could not be found in the cited works, and is therefore only reported for our method.

As shown in Table 12, all NPCR values for the three color channels and different image sizes exceed 99.6%, confirming the strong sensitivity and robustness of the proposed method against differential attacks.

Table 13 presents the UACI values obtained for different image sizes, allowing a comparative evaluation between the proposed algorithm and several existing methods.

Table 11. Comparison of NPCR and UACI values for different encryption algorithms.

Metric	Channel	Our algorithm	[70]	[71]	[74]	[75]
NPCR (%)	Red	99.6143	99.6550	99.6080	99.6060	99.6140
	Green	99.5994	99.6723	99.6140	99.6144	99.6073
	Blue	99.6273	99.6747	99.6095	99.6085	99.6292
UACI (%)	Red	33.4067	33.3987	33.4450	33.4317	33.4843
	Green	33.4025	33.3903	33.5354	33.5394	33.4690
	Blue	33.5116	33.3644	33.4520	33.4437	33.4965

Table 12. Comparison of NPCR results for images of different sizes between our algorithm and other methods.

Image size	Image name	Channel	Our algorithm	[25]	[76]	[77]	[78]
256×256	4.1.04	Red	99.6203	/	/	/	/
	4.1.04	Green	99.6123	/	/	/	/
	4.1.04	Blue	99.6312	/	/	/	/
512×512	2.1.02	Red	99.6252	99.5887	99.6201	99.6246	99.6100
	2.1.02	Green	99.6201	99.6116	99.6506	99.5899	99.6054
	2.1.02	Blue	99.6133	99.6139	99.6414	99.5987	99.6057
1024×1024	2.2.01	Red	99.6207	99.6112	99.6277	99.6099	99.6166
	2.2.01	Green	99.6022	99.6144	99.5575	99.6076	99.6164
	2.2.01	Blue	99.6301	99.6146	99.6490	99.6110	99.6141

Table 13. Comparison of UACI results for images of different sizes between our algorithm and other methods.

Image size	Image name	Channel	Our algorithm	[25]	[76]	[77]	[78]
256×256	4.1.04	Red	33.5509	/	/	/	/
	4.1.04	Green	33.3212	/	/	/	/
	4.1.04	Blue	33.5123	/	/	/	/
512×512	2.1.02	Red	33.5138	33.4623	33.5137	33.4399	33.4839
	2.1.02	Green	33.4801	33.4349	33.4412	33.4847	33.4263
	2.1.02	Blue	33.4276	33.4619	33.3946	33.4360	33.4408
1024×1024	2.2.01	Red	33.4301	33.4941	33.4753	33.5109	33.5007
	2.2.01	Green	33.5120	33.4700	33.4831	33.4686	33.4449
	2.2.01	Blue	33.4566	33.4736	33.5122	33.4622	33.4479

The UACI results show that the proposed encryption scheme produces values close to the theoretical ideal of 33.33% for all tested image sizes and color channels. These values indicate that even minor changes result in significant differences, reflecting high sensitivity to initial conditions. These results confirm the robustness of the algorithm against differential attacks.

5.3. Key space analysis

When evaluating the strength of an encryption system, it is crucial to examine the size and distribution of its key space. A large, well-designed key space makes brute-force attacks nearly impossible and provides a higher level of protection. As a general rule, the key space should be sufficiently large that launching multiple decryption attempts, known as a trial-and-error process, becomes impractical. The recommended size for a key space in modern systems is 128 bits or more. Currently, it is commonly believed that the key space of a secure cryptosystem must be greater than 2^{128} to resist brute-force attacks.[80]

The system we described earlier is based on chaotic dynamics and characterized by a large, unpredictable key space. This guarantees a large diversity of keys, providing enhanced

overall resilience to attacks.

To achieve this, we consider a copy of the master system (4), characterized by the parameters l_{20}, l_{21}, l_{22}, l_{23}, l_{24}, l_{25}, l_{26}, l_{27}, l_{28}, l_{29}, l_{210}, l_{211}, and l_{212}. We also take into account the fractional derivative orders k_1, k_2, k_3, and k_4 to generate a secret key for the transmission mechanism. However, the constraint imposed by the system dynamics excludes the order λ, and the parameters β_1 and β_2. We then determine the key space size Δ, which represents the finite set of all possible keys.

The key space estimation follows from a sensitivity analysis of the parameters involved.[80] In each step, we vary one parameter while keeping all others fixed and estimate the UACI and NPCR for encrypted images obtained from the sender and its copy. Through such an estimation, we assess the sensitivity of each parameter, and thus compute the key space with accuracy.

As illustrated in Table 14, and in the context of double-precision floats, an accuracy of the order of 10^{-16} can be achieved in general. To evaluate the key space, we speak of a precision of at least 10^{-15}, which corresponds to the threshold often considered as the relevant reference frame.

Based on the results illustrated in Table 14, we can determine the dimension of the key space associated with our scheme as follows:

$$\Delta = \prod_{i=1}^{13} 10^{(14\times 9+13\times 3+12)} = 10^{177} \approx 2^{589} \gg 2^{128}.$$

The key space thus determined is extremely vast, with an estimated size of $10^{177} \approx 2^{589}$, well in excess of 2^{128}.[80] This scale gives the scheme significant resistance to brute-force attacks, making exhaustive exploration unfeasible with current computing capacities.

Table 14. Sensitivity to parameter variations.

Parameters l_i	Sensitivity ϵ_i	No. of possibilities
$l_{21} = 1.5$	$\epsilon_1 = 10^{-15}$	$\Delta_1 = 10^{14}$ $(\approx 2^{46})$
$l_{22} = 0.3$	$\epsilon_2 = 10^{-15}$	$\Delta_2 = 10^{14}$ $(\approx 2^{46})$
$l_{23} = 9$	$\epsilon_3 = 10^{-15}$	$\Delta_3 = 10^{14}$ $(\approx 2^{46})$
$l_{24} = 330$	$\epsilon_4 = 10^{-14}$	$\Delta_4 = 10^{13}$ $(\approx 2^{43})$
$l_{25} = 0.07$	$\epsilon_5 = 10^{-15}$	$\Delta_5 = 10^{14}$ $(\approx 2^{46})$
$l_{26} = 0.3$	$\epsilon_6 = 10^{-15}$	$\Delta_6 = 10^{14}$ $(\approx 2^{46})$
$l_{27} = 2.1$	$\epsilon_7 = 10^{-14}$	$\Delta_7 = 10^{13}$ $(\approx 2^{43})$
$l_{28} = 2.6$	$\epsilon_8 = 10^{-13}$	$\Delta_8 = 10^{12}$ $(\approx 2^{40})$
$l_{29} = 10^{-7}$	$\epsilon_9 = 10^{-15}$	$\Delta_9 = 10^{14}$ $(\approx 2^{46})$
$l_{210} = 0.00003$	$\epsilon_{10} = 10^{-15}$	$\Delta_{10} = 10^{14}$ $(\approx 2^{46})$
$l_{211} = 0.008$	$\epsilon_{11} = 10^{-14}$	$\Delta_{11} = 10^{13}$ $(\approx 2^{43})$
$l_{212} = 0.5$	$\epsilon_{12} = 10^{-15}$	$\Delta_{12} = 10^{14}$ $(\approx 2^{46})$
$l_{213} = 0.3$	$\epsilon_{13} = 10^{-15}$	$\Delta_{13} = 10^{14}$ $(\approx 2^{46})$

To illustrate this point, the high sensitivity of each key remains a major advantage when it comes to security. Any infinitesimal variation in any system parameter will produce large changes in the generation of encryption sequences. This increases the unpredictability and strength of the encryption mechanism against reconstruction attempts by an attacker.

Table 15 below provides a comparison of the key space, showing how the proposed scheme performs compared to other currently available encryption techniques.

Table 15. Comparison of key space size between different encryption schemes.

Reference	Key space size
Our method	10^{182}
[70]	10^{112}
[71]	10^{120}
[72]	10^{72}
[74]	10^{56}
[75]	10^{64}

5.4. Robustness to noise in image cryptography: A PSNR-based study

Mean squared error (MSE) and PSNR are essential measures for evaluating the quality of images after encryption and decryption, particularly with regard to the impact of noise.[81] The MSE quantifies the error between the original image and the decrypted image by calculating the sum of the squares of the differences between the corresponding pixels and then normalizing it by the total number of pixels, according to the following formula:

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j) - K(i,j)]^2, \tag{22}$$

where $I(i,j)$ and $K(i,j)$ represent the pixel values in the original and decrypted image, respectively, and M and N are the image dimensions.

PSNR measures the ratio of maximum signal power to noise power, and is calculated using the following formula:

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right), \tag{23}$$

where MAX is the maximum possible value of a pixel, typically 255 for an 8-bit grayscale image.

In the following, we will evaluate the robustness of our algorithm to noise attacks. For each variation in density, we will illustrate the impact of noise on the restored image. The results are shown in Fig. 17. Table 16 shows the results for PSNR and MSE as a function of salt-and-pepper noise density variation. Table 16 highlights the increasing impact of salt-and-pepper noise on the quality of the restored image, with the MSE rising from 309.9962 to 1282.3665 and the PSNR decreasing from 19.43 dB to 13.41 dB. These results demonstrate that the higher the noise density, the more pronounced the degradation, confirming the effectiveness of our algorithm against perturbations.

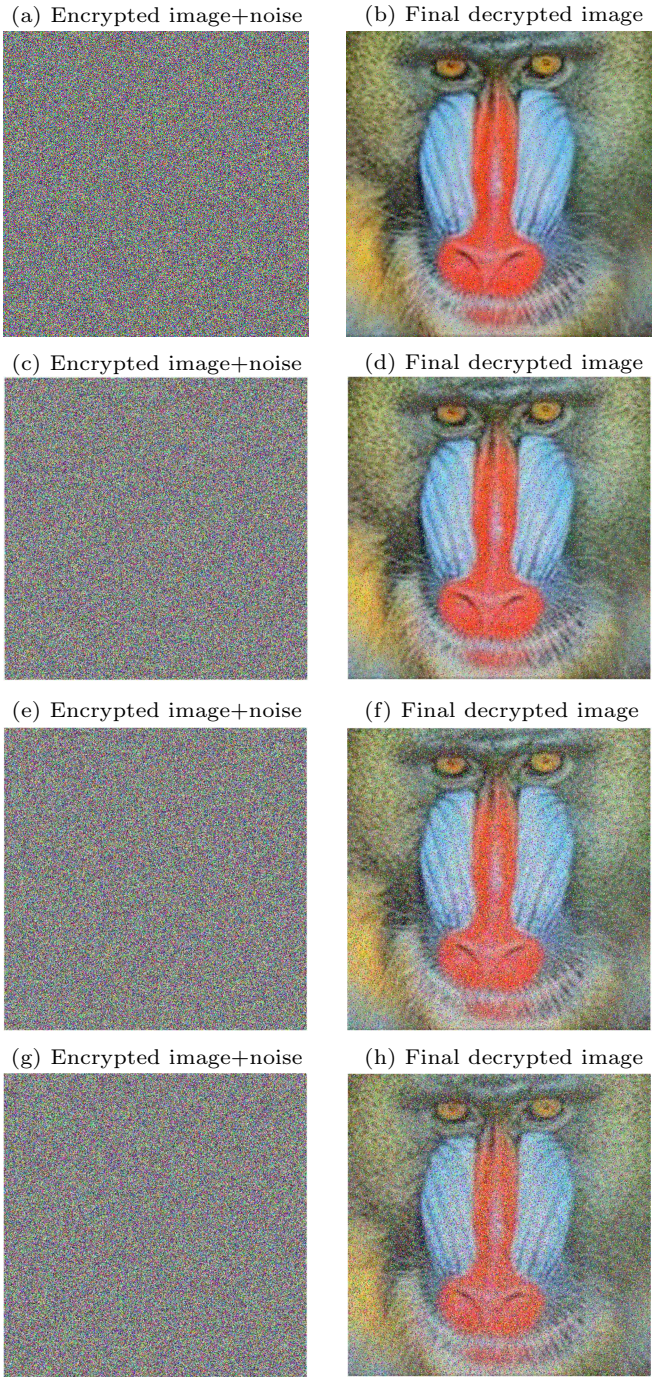


Fig. 17. The left column shows the encrypted images with noise, while the right column presents the decrypted images after the addition of salt-and-pepper noise: (a) and (b) salt-and-pepper noise density 0.05, (c) and (d) salt-and-pepper noise density 0.1, (e) and (f) salt-and-pepper noise density 0.17, (g) and (h) salt-and-pepper noise density 0.24.

Table 16. Results of PSNR and MSE for varying salt-and-paper noise densities.

	(a), (b)	(c), (d)	(e), (f)	(g), (h)
Salt-pepper noise density	0.05	0.1	0.17	0.24
MSE	309.9962	594.3394	967.9154	1282.3665
PSNR(dB)	19.43	16.65	14.60	13.41

6. Conclusion

This paper proposes a new fractional-order chaotic system that incorporates a memristor emulator. With increasing demands for secure and lightweight cryptographic schemes, such architectures that combine the high complexity of fractional dynamics and the intrinsic nonlinearity of memristive devices are being actively explored. The proposed system takes advantage of memristors' non-volatility, ultra-low power, and small geometry for advanced encryption applications.

The following summarizes the main contributions and results of this work: (i) Design and analysis of a novel memristor-based fractional-order chaotic system. (ii) Implementation of two complementary synchronization approaches to guarantee that signals will align securely. (iii) Development of an image encryption framework based on the proposed system, where multiple cryptographic keys were used and fractional dynamics added. (iv) Validation of the systems security performance through entropy evaluation (≈ 7.993), correlation tests, NPCR/UACI metrics (e.g., 33.40% and 99.61% respectively), and noise-resilience analysis for different images and different image sizes.

Nevertheless, this work has certain limitations. Due to the large amount of data to be processed, the execution time is particularly long for high-resolution images. Moreover, the algorithm requires substantial memory, making it difficult to use on resource-limited systems.

Future research will focus on hardware implementation, paving the way for the system's use in real-world conditions. Subsequent work will complement these implementations by integrating the system into compact or embedded devices (i.e., IoT devices). Research will also focus on optimizing real-time performance for encryption. Additionally, entropy values close to 7.999 and an NPCR of 99.6% will serve as relevant indicators of the effectiveness and robustness of the adopted encryption approach. Overall, this work establishes a foundation for a new approach to secure, low-power cryptographic architectures based on chaotic, fractional-order memristor dynamics.

References

[1] Chua L O 1971 *IEEE Trans. Circ. Theory* **18** 507
[2] Strukov D B, Snider G S, Stewart D R and Williams R S 2008 *Nature* **453** 80
[3] Williams R S 2008 *IEEE Spectr.* **45** 28
[4] Chouchane L, Hamiche H, Kemih K, Azar A T, El-shafai W and Kamal N A 2025 *Proceedings of the 2025 International Conference on Control, Automation and Diagnosis (ICCAD)* July 1–3, 2025, Barcelona, Spain
[5] Qian K, Xiao Y, Wei Y, Liu D, Wang Q and Feng W 2023 *Micromachines* **14** 2090
[6] Yu F, Zhang S, Su D, Wu Y, Gracia Y M and Yin H 2025 *Fractals Fract.* **9** 115
[7] Xiao P D, Fang J J, Wei Z M, Dong Y, Du S and Wen S 2025 *IEEE Trans. Autom. Sci. Eng.* **22** 15163
[8] Hong Q, Jiang H, Xiao P, Du S and Li T 2025 *IEEE Trans. Comput.* **74** 703
[9] Lin R, Shi G, Qiao F, Wang C and Wu S 2023 *Microelectron. J.* **133** 105702

[10] Gupta R K, Joshi M, Bisen A, Agarwal A and Singh A 2025 *Chaos Solitons Fractals* **192** 115964
[11] Neifar A, Barraj I, Mestiri H and Masmoudi M 2025 *Micromachines* **16** 848
[12] Barraj I, Neifar A, Mestiri H and Masmoudi M 2025 *Micromachines* **16** 269
[13] Zhou L, Wang C, Qin H and Wang Q 2023 *AEU Int. J. Electron. Commun.* **162** 154593
[14] Ananda Y R, Raj N and Trivedi G 2023 *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **31** C3
[15] Li M J, Zhou X F, Wang F, Bi M H, Yang G W, Xu M M, Hu M and Li H Z 2024 *Chaos Solitons Fractals* **182** 114720
[16] Ma J, Zhang H, Zhang J and Wang L 2024 *Chaos Solitons Fractals* **188** 115534
[17] Hamiche H, Kemih K, Addouche S A, Azar A T, Saddaoui R and Laghrouche M 2021 *Int. J. Adv. Intell. Paradigms* **20** 58
[18] Zouad F, Kemih K and Hamiche H 2019 *Analog Integr. Circuits Signal Process.* **99** 619
[19] Megherbi O, Guermah S, Hamiche H, Djennoune S and Ghanes M 2007 *Proceedings of the 3rd International Conference on Systems and Control (ICSC)*, October 29–31, 2007, Algiers, Algeria, p. 117
[20] Su J, Fang S, Hong Y and Wen Y 2024 *Chin. Phys. B* **33** 070502
[21] Hamiche H, Guermah S, Kassim S, Lahdir M, Djennoune S and Bettayeb M 2015 *Proc. 3rd Int. Conf. Control Eng. Inf. Technol. (CEIT)* Tlemcen, Algeria
[22] Babu N R, Balasubramaniam P and Meng J E 2024 *Multimed. Tools Appl.* **83** 26055
[23] Ye C, Tan S, Wang J, Shi L, Zuo Q and Feng W 2025 *Entropy* **27** 276
[24] Feng W, Zhang J, Chen Y, Qin Z, Zhang Y, Ahmad M and Woniak M 2024 *Expert Syst. Appl.* **246** 123190
[25] Feng W, Wang Q, Liu H, Ren Y, Zhang J, Zhang S, Qian K and Wen H 2023 *Fractal Fract.* **7** 887
[26] Jiang M and Yang H 2023 *Entropy* **25** 1516
[27] Setiadi D R I M and Rijati N 2023 *Computation* **11** 178
[28] Shao S, Li J, Shao P and Xu G 2023 *IEEE Access* **11** 27477
[29] Qian K, Xiao Y, Wei Y, Liu D, Wang Q and Feng W 2023 *Micromachines* **14** 2090
[30] Wang Q, Sang H, Wang P, Yu X and Yang Z 2024 *Sci. Rep.* **14** 29615
[31] Zhang Q and Han J 2021 *Multimed. Tools Appl.* **80** 13841
[32] Kaur M, Singh D and Kumar V 2020 *Appl. Phys. B* **126** 147
[33] Zhang J and Liu E 2024 *Eur. Phys. J. B* **97** 100
[34] Xiong L, Wang X, Zhang X G and He T D 2024 *Optoelectron. Lett.* **20** 183
[35] Duan Z K, Chen J H, He S B, Yu X M, Wang Q, Zhang X and Xiong P 2025 *Micromachines* **16** 246
[36] Volos C K, Pham V T, Nistazakis H E and Stouboulos I N 2020 *Int. J. Bifurcat. Chaos* **30** 2030036
[37] Messadi M, Kemih K, Moysis L and Volos C 2023 *Integration* **88** 91
[38] Xiong Z, Qu S and Luo J 2019 *Complexity* **2019** 3827201
[39] Yuan B, Xu H, Hu L and Wu J 2024 *Front. Phys.* **12** 1445805
[40] Wang Y and Li K 2025 *Math. Model. Control* **5** 164
[41] Fan J, Lai Q, Wang Q and Wang L 2024 *Chin. Phys. B* **33** 110501
[42] Zhou L, Wang C H, Du S and Zhou L 2016 *IEEE Trans. Neural Netw. Learn. Syst.* **28** 570
[43] Zhou L, Tan F, Yu F and Liu W 2019 *Neurocomputing* **359** 264
[44] Zhou L, Lin H and Tan F 2023 *Chaos Solitons Fractals* **175** 113977
[45] Zhou L and Tan F 2019 *Nonlinear Dyn.* **96** 869
[46] Tan F, Zhou L, Zong G, Wang Z, Zhuang G and Shangguan X 2025 *IEEE Trans. Autom. Sci. Eng.* **22** 10827
[47] Zhou L, Tan F and Yu F 2020 *IEEE Syst. J.* **14** 2508
[48] Lin L, Zhuang Y, Xu Z, Yang D and Wu D 2023 *Front. Phys.* **11** 1202871
[49] Yang F, Mou J, Liu J, Ma C and Yan H 2020 *Signal Process.* **176** 107373
[50] Yang F, Mou J, Ma C and Cao Y 2020 *Opt. Lasers Eng.* **135** 106031
[51] Megherbi O, Kassim S, Hamiche H, Djennoune S, Bettayeb M and Barbot J P 2017 *Proc. SIAM Conf. Control Appl. (CT17)* Pittsburgh, USA
[52] Kassim S, Hamiche H, Djennoune S and Bettayeb M 2019 *Proc. IEEE Symp. Signal Process. Inf. Technol. (ISSPIT)* Ajman, UAE
[53] Hannoun K, Hamiche H, Lahdir M, Megherbi O, Laghrouche M and Bettayeb M 2024 *Phys. Scr.* **99** 055255
[54] Megherbi O, Hamiche H and Bettayeb M 2024 *Comput. Electr. Eng.* **116** 109224
[55] Wang X, Zhang Y and Li M 2021 *Multimed. Tools Appl.* **80** 7431
[56] Feng W, Qin Z T, Zhang J and Ahmad M 2021 *IEEE Access* **9** 145459
[57] Feng W, He Y, Li H and Li C 2019 *IEEE Access* **7** 12584
[58] Zhao Y, Shi Q and Ding Q 2025 *Entropy* **27** 40
[59] Fan H, Zhang C, Lu H, Li M and Liu Y 2021 *Entropy* **23** 158
[60] Shinriki M, Yamato M and Mori 1981 *Proc. IEEE* **69** 394
[61] Chouchan L, Hamiche H, Kemih K and Ouslimani A 2024 *Nonlinear Phenom. Complex Syst.* **27** 135
[62] Bao H and Cao J 2016 *Nonlinear Anal. Model. Control.* **21** 306
[63] Zhang D, Mei J and Miao P 2017 *Appl. Math. Model.* **48** 303
[64] Wang E, Yan S, Sun X and Wang Q 2023 *Nonlinear Dyn.* **111** 3869
[65] Chen D, Zhang R, Liu X and Ma X 2014 *Commun. Nonlinear Sci. Numer. Simul.* **19** 4105
[66] Aguila-Camacho N, Duarte-Mermoud M and Gallegos J 2014 *Commun. Nonlinear Sci. Numer. Simul.* **19** 2951
[67] Alghamdi Y, Munir A and Ahmad J 2022 *Entropy* **24** 1344
[68] Setiadi D R I M and Rijati N 2023 *Computation* **11** 178
[69] Shannon C E 1948 *Bell Syst. Tech. J.* **27** 379
[70] Niu Y, Sun X, Zhang C and Liu H 2020 *Math. Probl. Eng.* **2020** 6795964
[71] Dong Z, Zhang Z, Zhou H and Chen X 2024 *Comput. Mater. Continua* **78** 1977
[72] Kanwal S, *et al.* 2022 *Secur. Commun. Netw.* **2022** 4152683
[73] Li R, Liu T and Yin J 2024 *Sci. Rep.* **14** 20733
[74] Li T, Shi J and Zhang D 2021 *J. Electron. Imaging* **30** 013008
[75] Mohamed H G, ElKamchouchi D H and Moussa K H 2020 *Entropy* **22** 158
[76] Hua Z, Zhu Z, Yi S, Zhang Z and Huang H 2021 *Inf. Sci.* **546** 1063
[77] Hua Z, Zhou Y and Huang H 2019 *Inf. Sci.* **480** 403
[78] Qian K, Feng W, Qin Z, Zhang J, Luo X and Zhu Z 2022 *Front. Phys.* **10** 718
[79] Wu Y, Noonan J P and Agaian S 2011 *Cyber J. Sel. Areas Telecommun. (JSAT)* **1** 31
[80] Ozkaynak F 2018 *Nonlinear Dyn.* **92** 305
[81] Luo Y, Wang F, Xu S, Zhang S, Li L, Su M and Liu J 2022 *Expert Syst. Appl.* **208** 118133