scientific reports

Check for updates

OPEN

Dynamic analysis of a novel hyperchaotic system based on STM32 and application in image encryption

XueFeng Cheng[1,6], Hongmei Zhu[1], Lian Liu[1], Kunpeng Mao[1] & Juan Liu[2,3,4,5,6]✉

This paper presents a novel 4D hyperchaotic system derived from a modified 3D Lorenz chaotic system. A key aspect of this system is the presence of a single equilibrium point, and its stability is carefully analyzed. The dynamic properties, including the Lyapunov exponent spectrum, bifurcation diagram, and chaotic attractors, are investigated using MATLAB simulations. The results reveal that the system displays hyperchaotic behavior across a wide range of the parameter *d*, with its chaotic attractor transitioning through four states: hyperchaotic, chaotic, periodic, and quasi-periodic, showcasing the system's complex dynamics. Experimental validation using STM32 embedded hardware successfully reproduces these four types of chaotic attractors, confirming the theoretical predictions. The proposed hyperchaotic system is then applied to image encryption, introducing a novel encryption method. The hyperchaotic key sequence generated by this system meets 15 tests of the NIST SP800-22 standard, and further experimental validation with STM32 hardware demonstrates the algorithm's effectiveness, simplicity, non-linearity, and high security. The encrypted images and sequences are rigorously tested key space analysis, histogram similarity analysis, information entropy analysis, statistical attack analysis, differential attack analysis, key sensitivity analysis, and correlation analysis, highlighting the robustness and reliability of the proposed system. This method is versatile and can be extended to various fields, including audio and video encryption, text encryption, IoT security, financial transaction security, and medical data protection.

Since Rössler discovered the first hyperchaotic system[1] in 1979, the evolution of hyperchaotic systems from discovery to application has encompassed multiple stages, including theoretical research[2–5], numerical simulation[6–9], experimental validation[10–14], and application exploration[15–18]. In recent years, the rapid proliferation of digital images containing sensitive information has heightened the need for robust encryption techniques to protect against unauthorized access and illegal copying[19–23]. Traditional image encryption methods often struggle with issues such as limited parameter ranges, slow encryption speeds, and vulnerability to various attacks. To address these challenges, researchers have explored the use of hyperchaotic systems, which offer complex dynamic properties and high sensitivity to initial conditions, making them suitable for secure image encryption applications.

Several novel image encryption frameworks and algorithms have been proposed, leveraging different forms of hyperchaotic systems to enhance security and efficiency. A common approach involves multi-stage encryption processes that combine permutation and diffusion techniques. For instance, one framework[24] utilizes a new hyperchaotic system based on STM32 and single neuron models (SNM), involving three stages that apply a substitution box (S-box) followed by an XOR operation with encryption keys derived from the numerical solutions of hyperchaotic maps and SNM. This method significantly enhances security with a vast key space and improves encryption efficiency through optimized parallel processing.

Another approach[25] employs a 2D hyperchaotic map for designing an efficient pseudo-random number generator (PRNG) and a color image encryption algorithm. The hyperchaotic system generates highly random sequences validated through various analyses, with the PRNG passing all NIST tests. The encryption algorithm

[1]School of Big Data and Information Industry, Chongqing City Management College, Chongqing 401331, China. [2]School of Artificial Intelligence, Chongqing University of Education, Chongqing 400065, China. [3]Chongqing University, Chongqing 400044, China. [4]Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China. [5]Chongqing College, University of Chinese Academy of Sciences, Chongqing 400722, China. [6]These authors contributed equally: XueFeng Cheng and Juan Liu. ✉email: liujuan@cque.edu.cn

uses cross-channel permutation and diffusion to enhance security and speed, though the framework's complexity may require significant computational resources.

Further advancements include the use of a 6D hyperchaotic system combined with random signal insertion. This scheme[26] leverages the sum of all plaintext pixels to generate initial values, ensuring high key sensitivity. The encryption process involves splitting each pixel, scrambling, cycle shifting, and diffusion, resulting in robust security against various attacks. Similarly, a fast and secure image encryption algorithm[27] utilizes a time-delayed combinatorial hyperchaos map to enhance security and efficiency through simultaneous shuffling and diffusion processes.

Color image encryption algorithms[28] have also been developed, incorporating hyperchaotic maps with DNA mutation operations. These methods enhance dynamic properties and introduce additional randomness and complexity, validated through extensive simulations demonstrating robustness against various attacks. A novel image encryption algorithm[29] combining a seven-dimensional hyperchaotic map with stochastic signal injection further increases security, employing SHA-512 for initial value generation to ensure strong correlation with plaintext images.

Additionally, The logistic Feigenbaum nonlinear cross-coupled hyperchaotic map[30] with dynamic correlation of plaintext pixels offers improved chaotic properties and reduced computational complexity, validated through extensive security analyses. Reservoir computing[31] approaches for digital image encryption and hyperchaotic finance model forecasting provide faster and more accurate long-term predictions with reduced execution times, enhancing security and efficiency.

Optical image encryption[32] algorithms combining 4D memristive hyperchaotic systems with compressed sensing reduce image size and transmission burden while maintaining high security, validated through extensive simulations. Deep learning-based compression[33] combined with two-dimensional Sin-Linear-Cos hyperchaotic maps and matrix encoding ensures high-quality cipher images, fast processing speeds, and robustness against attacks.

Despite the advancements in hyperchaotic-based encryption algorithms, challenges remain in terms of implementation complexity and dependency on numerical precision. The development and application of novel four-dimensional hyperchaotic systems for image encryption demonstrate increased complexity and unpredictability, validated through extensive statistical tests and dynamic analyses[16,34,35].

Critically, the cryptanalysis of existing hyperchaotic-based encryption algorithms[36] reveals vulnerabilities to chosen plaintext attacks, highlighting the need for continuous improvement in algorithmic security structures. Lastly, the introduction of reservoir computing approaches and non-degenerate hyperchaotic systems for secure DNA-coding image optical communication underscores the potential for enhanced performance and security in optical access networks[37].

The reviewed literature illustrates significant progress in developing robust, hyperchaotic-based image encryption algorithms, each offering unique enhancements in security, efficiency, and resistance to various attacks. However, the construction of new hyperchaotic systems and the comprehensive analysis of their dynamic behavior, as well as the engineering implementation of image encryption, remain key considerations for future research and practical applications. Additionally, there are few reported studies on utilizing the high randomness and complexity of hyperchaotic systems combined with embedded hardware STM32 for image encryption.

Based on the above analysis, the research content of this paper includes the following. Firstly, a novel 4D hyperchaotic system based on a modified 3D Lorenz chaotic system is devised and analyzed for its dynamic properties, including equilibrium point stability, Lyapunov exponents' spectrum, bifurcation diagram, and chaotic attractors. MATLAB numerical simulations confirm its hyperchaotic behavior, and experimental validation using embedded hardware STM32 demonstrates consistency between the hardware results and MATLAB simulations through oscilloscope displays of the attractor phase portrait. Secondly, leveraging the hyperchaotic key sequence generated by this system, color image pixel encryption is achieved via two bit-XOR operations, with experimental findings revealing high randomness in the hyperchaotic key sequence and the encrypted image making it impossible to discern any information about the original image, meeting 15 tests of the NIST SP800-22 standard. Thirdly, analysis against statistical attacks, differential attacks, key sensitivity, correlation, and key space demonstrates that the encrypted data sequence resembles random noise, underscoring the high complexity and nonlinearity of the hyperchaotic key sequence. Lastly, the conclusions are drawn.

Dynamical analysis of a new hyperchaotic system

An 4D hyperchaotic system[38] is described by (1):

$$\begin{cases} \dot{x} = 40(y - x) + w \\ \dot{y} = 10x + 25y - xz \\ \dot{z} = xy - 3z \\ \dot{w} = dx \end{cases} \tag{1}$$

where x, y, z and w are driving variables, and d is a positive real parameter. Varying the parameter d can cause system (1) to exhibit hyperchaotic, chaotic, periodic, or quasi-periodic motion. Notably, the hyperchaotic system (1) has only one equilibrium point $\boldsymbol{O}(0,0,0,0)$, which is unstable and acts as a saddle point.

Linearizing the system (1) and calculating the Jacobian matrix at the equilibrium point $\boldsymbol{O}(0,0,0,0)$ is shown as follows:

$$J_0=\begin{pmatrix} -40 & 40 & 0 & 1 \\ 10 & 25 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ d & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

Let $det(\boldsymbol{J}_0 - \lambda\mathbf{I}) = 0$, the eigenvalues of system (1) in the equilibrium point $\boldsymbol{O}(0,0,0,0)$ is:

$$\lambda_1 = -40, \lambda_2 = 25, \lambda_3 = -3, \lambda_4 = 0. \tag{3}$$

Herein, $\lambda_2 = 25$ is positive, $\lambda_1 = -40$ and $\lambda_3 = -3$ are negative. So, the equilibrium point $\boldsymbol{O}(0,0,0,0)$ is saddle point of system (1), and it is unstable. It indicates the new system (1) is dissipative, and can be obtained:

$$\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{w}}{\partial w} = -40 + 25 - 3 = -18. \tag{4}$$

That is, each volume containing the system orbit shrinks to zero as $t \to \infty$ at an exponential rate -18. Consequently, all orbits of system (1) will ultimately be confined to a specific subset of zero volume.

Numerical simulation results

MATLAB excels in numerical simulation with advanced mathematical libraries, matrix operation advantages, a flexible programming language, and support for parallel computing. Its extensive toolbox spans multiple domains, and powerful plotting and visualization tools aid in intuitively showcasing simulation results. Therefore, this paper employs MATLAB simulation software for the numerical analysis of the hyperchaotic system.

According to chaos theory, the adjacent trajectories of a chaotic attractor in the state space separate from each other exponentially. The spectrum of Lyapunov exponents quantitatively shows the contraction or expansion of these trajectories.

Figure 1 shows the Lyapunov exponents' spectrum of system (1) with the changing of parameter d. It can be observed that as the parameter d changes, the four Lyapunov exponents of the constructed new system (1) also vary. The largest Lyapunov exponent remains positive over a significant range of parameter d, and the second-largest Lyapunov exponent exhibits positive as parameter d changes. Therefore, the system (1) demonstrates hyperchaotic characteristics.

Figure 2 shows the bifurcation diagram of state variable x of system (1). It is evident that as the parameter d gradually changes, the bifurcation diagram reflects the instability and complexity of the system's solutions x, providing information about the transition from order to chaos. When the bifurcation diagram exhibits more and increasingly complex branches, it indicates that the dynamical behavior of the new system (1) becomes more intricate, potentially entering a chaotic state.

It is found that both of the Lyapunov exponents' spectrum and the bifurcation diagram match, indicating that the numerical simulation results are correct. The variation of system (1) with parameter d is depicted in Table 1. For $1 < d \leq 23.5$, system (1) exhibits hyperchaotic behavior.

Let $d = 15$, the Lyapunov exponents of system (1) is $(1.8451, 0.1832, 0, -19.9925)$, the hyperchaotic attractor is shown in Fig. 3a. Let $d = 70$, the Lyapunov exponents of system (1) is $(1.2823, 0, -0.5963, -18.7196)$, the chaotic attractor is shown in Fig. 3b. Let $d = 100$, the Lyapunov exponents of system (1) is $(0, -1.0039, -0.9961, -16.0151)$, the periodic orbit attractor is shown in Fig. 3c. Let $d = 120$, the Lyapunov exponents of system (1) is $(0, 0, -2.1678, -15.8563)$, the quasi-periodic orbit attractor is shown in Fig. 3d.

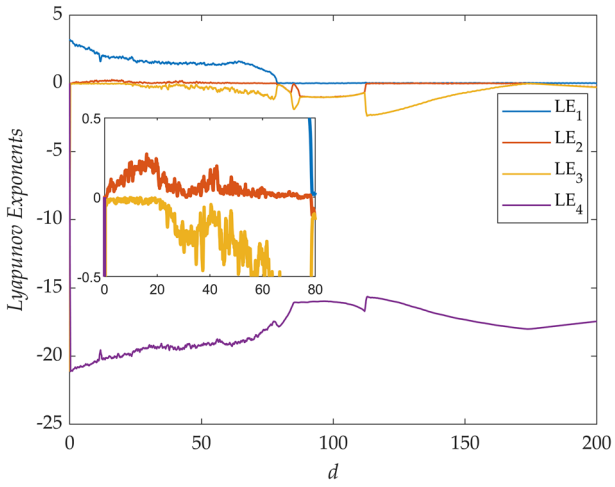

Fig. 1. Lyapunov exponents' spectrum of system (1).

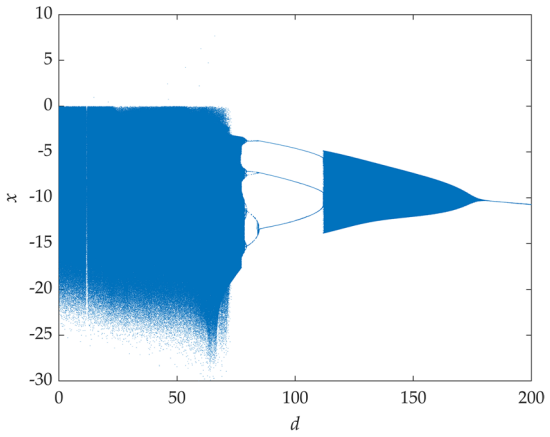

Fig. 2. Bifurcation diagram of x.

Parameter d	Lyapunov exponents	Stability
$0 < d \leq 1$, $23.5 < d \leq 78.6$	$LE_1 > 0, LE_2 = 0, LE_3 < 0, LE_4 < 0$	Chaotic
$1 < d \leq 23.5$	$LE_1 > 0, LE_2 > 0, LE_3 = 0, LE_4 < 0$	Hyperchaotic
$78.6 < d \leq 113$, $177.5 < d \leq 200$	$LE_1 = 0, LE_2 < 0, LE_3 < 0, LE_4 < 0$	Periodic
$113 < d \leq 177.5$	$LE_1 = 0, LE_2 = 0, LE_3 < 0, LE_4 < 0$	Quasi-periodic

Table 1. Stability of system (1) under different parameter d.

Embedded hardware implementation results

In embedded hardware, common components include DSP and FPGA, with DSP widely applied in various embedded implementations due to its flexibility. The STM32 microcontroller, representing a typical embedded hardware, integrates numerous peripherals, prioritizes power optimization, and offers flexible configuration options. Extensively utilized in industrial control, automotive electronics, medical devices, and consumer electronics, it meets the requirements of diverse application scenarios.

In this experiment, an STM32F103ZET6 microcontroller is used to simulate and control the phase diagram of a hyperchaotic system. The results of the embedded system are displayed using a Tektronix TBS1072C digital oscilloscope. The implementation of hyperchaotic systems on STM32 has significant purposes and implications:

a. Hyperchaotic systems are complex nonlinear dynamical systems that exhibit chaotic behavior. Implementing them on STM32 can provide valuable insights into the characteristics and behavior of hyperchaotic systems, as well as challenges related to modeling, simulation, and control of nonlinear dynamical systems.
b. Hyperchaotic systems have wide-ranging applications in communication, encryption, control, and other fields. Implementing them on STM32 can provide a solid foundation for basic research and experimental validation in these fields.

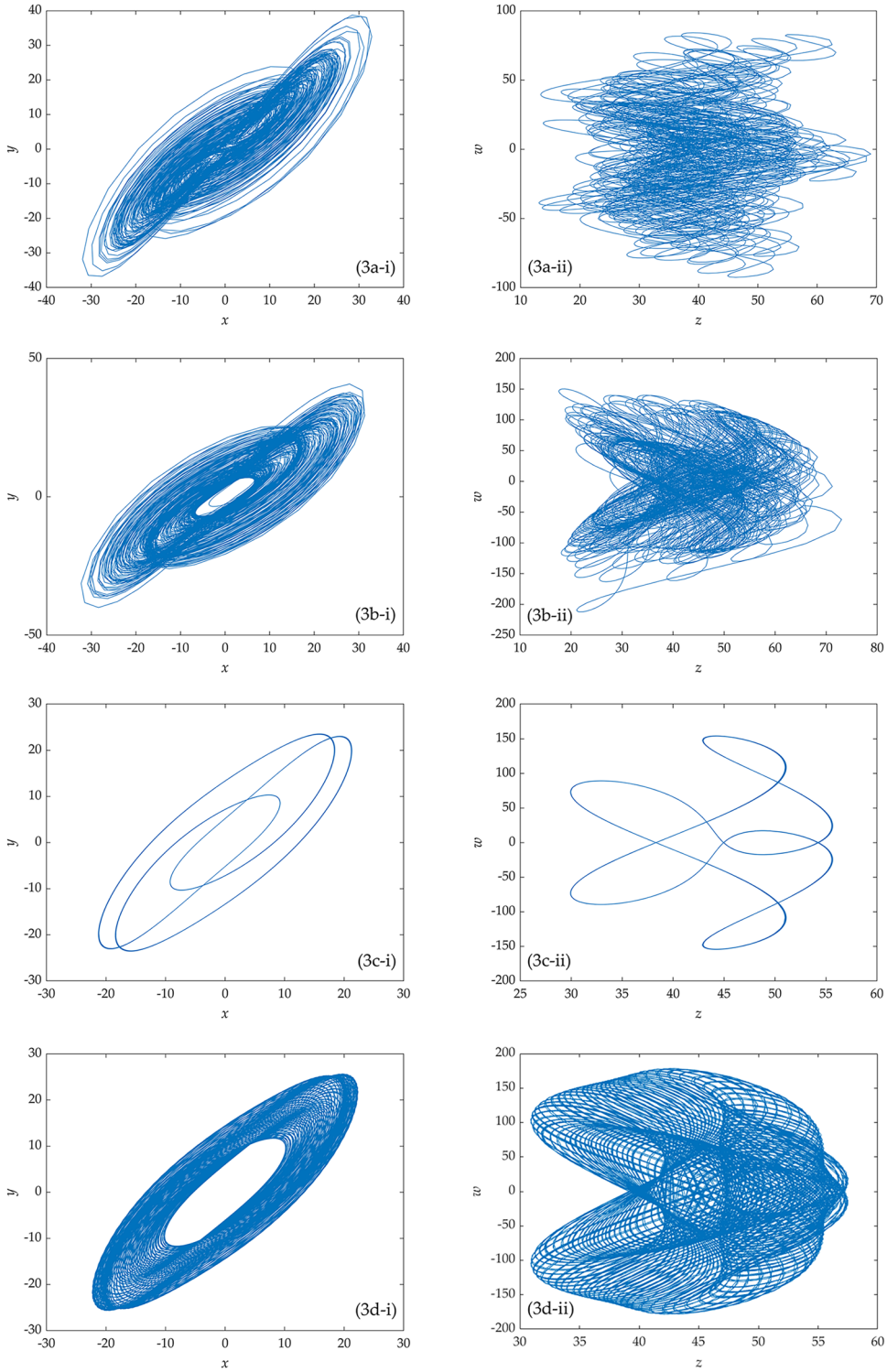

Fig. 3. The attractors of system (1) ((**a**) hyperchaotic attractor ($d = 15$); (**b**) chaotic attractor ($d = 70$); (**c**) periodic attractor ($d = 100$); (**d**) quasi-periodic attractor ($d = 120$)).

$$
\begin{cases}
h_1\_x = \big(40 \times \big(y[\tau] - x[\tau]\big) + w[\tau]\big) \times h \\
h_1\_y = \big(10 \times x[\tau] + 25 \times y[\tau] - x[\tau] \times z[\tau]\big) \times h \\
h_1\_z = \big(x[\tau] \times y[\tau] - 3 \times z[\tau]\big) \times h \\
h_1\_w = (d \times x[\tau]) \times h \\
temp\_x = x[\tau] + h_1\_w/2 \\
temp\_y = y[\tau] + h_1\_y/2 \\
temp\_z = z[\tau] + h_1\_z/2 \\
temp\_w = w[\tau] + h_1\_w/2 \\
h_2\_x = \big(40 \times \big(temp\_y - temp\_x\big) + temp\_w\big) \times h \\
h_2\_y = \big(10 \times temp\_x + 25 \times temp\_y - temp\_x \times temp\_z\big) \times h \\
h_2\_z = \big(temp\_x \times temp\_y - 3 \times temp\_z\big) \times h \\
h_2\_w = \big(d \times temp\_x\big) \times h \\
temp\_x = x[\tau] + h_2\_x/2 \\
temp\_y = y[\tau] + h_2\_y/2 \\
temp\_z = z[\tau] + h_2\_z/2 \\
temp\_w = w[\tau] + h_2\_w/2 \\
h_3\_x = \big(40 \times \big(temp\_y - temp\_x\big) + temp\_w\big) \times h \\
h_3\_y = \big(10 \times temp\_x + 25 \times temp\_y - temp\_x \times temp\_z\big) \times h \\
h_3\_z = \big(temp\_x \times temp\_y - 3 \times temp\_z\big) \times h \\
h_3\_w = \big(d \times temp\_x\big) \times h \\
temp\_x = x[\tau] + h_3\_x \\
temp\_y = y[\tau] + h_3\_y \\
temp\_z = z[\tau] + h_3\_z \\
temp\_w = w[\tau] + h_3\_w \\
h_4\_x = \big(40 \times \big(temp\_y - temp\_x\big) + temp\_w\big) \times h \\
h_4\_y = \big(10 \times temp\_x + 25 \times temp\_y - temp\_x \times temp\_z\big) \times h \\
h_4\_z = \big(temp\_x \times temp\_y - 3 \times temp\_z\big) \times h \\
h_4\_w = \big(d \times temp\_x\big) \times h \\
x[\tau + 1] = x[\tau] + (h_1\_x + 2 \times h_2\_x + 2 \times h_3\_x + h_4\_x)/6 \\
y[\tau + 1] = y[\tau] + \big(h_1\_y + 2 \times h_2\_y + 2 \times h_3\_y + h_4\_y\big)/6 \\
z[\tau + 1] = z[\tau] + (h_1\_z + 2 \times h_2\_z + 2 \times h_3\_z + h_4\_z)/6 \\
w[\tau + 1] = w[\tau] + (h_1\_w + 2 \times h_2\_w + 2 \times h_3\_w + h_4\_w)/6
\end{cases}
\tag{5}
$$

c. Implementing hyperchaotic systems on STM32 requires overcoming technical challenges such as limited hardware resources and complex algorithm implementation, which can promote the development of embedded systems and algorithm optimization.

In experiment, the STM32 is used to verify the 4th-order Runge–Kutta method used for the discretization of the new hyperchaotic system (1), the iterative process is shown in Eq. (5). Where parameter $d = 15$, iteration step $h = 0.02$. τ represents the current moment, $x[\tau]$, $y[\tau]$, $z[\tau]$ and $w[\tau]$ are the values of the current moment, $temp\_x$, $temp\_y$, $temp\_z$ and $temp\_w$ are the intermediate variables, h_1_x, h_1_y, h_1_z and h_1_w are the slopes at the beginning of the time period, h_2_x, h_2_y, h_2_z and h_2_w are the slopes of the midpoint of the time period, the slopes h_1_x, h_1_y, h_1_z and h_1_w are used by Euler's method to determine the values of $x[\tau]$, $y[\tau]$, $z[\tau]$ and $w[\tau]$ at the point $\tau + h/2$, h_3_x, h_3_y, h_3_z and h_4_w are the slopes of the midpoint, but this time the slopes h_2_x, h_2_y, h_2_z and h_2_w are used to determine the $x[\tau]$, $y[\tau]$, $z[\tau]$ and $w[\tau]$ values, h_4_x, h_4_y, h_4_z and h_4_w are

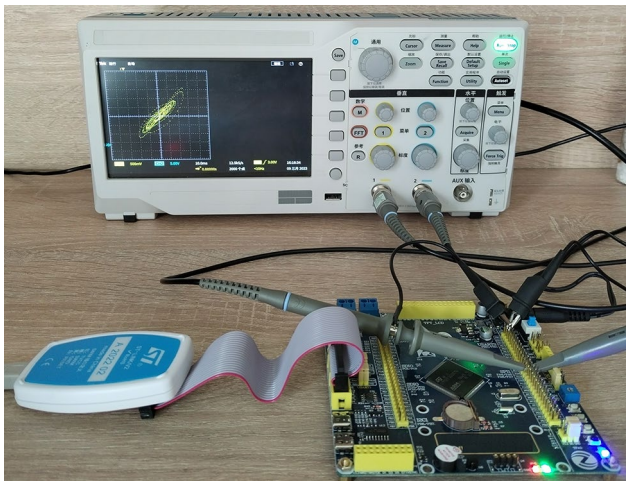

Fig. 4. Experimental platform for STM32 implementation.

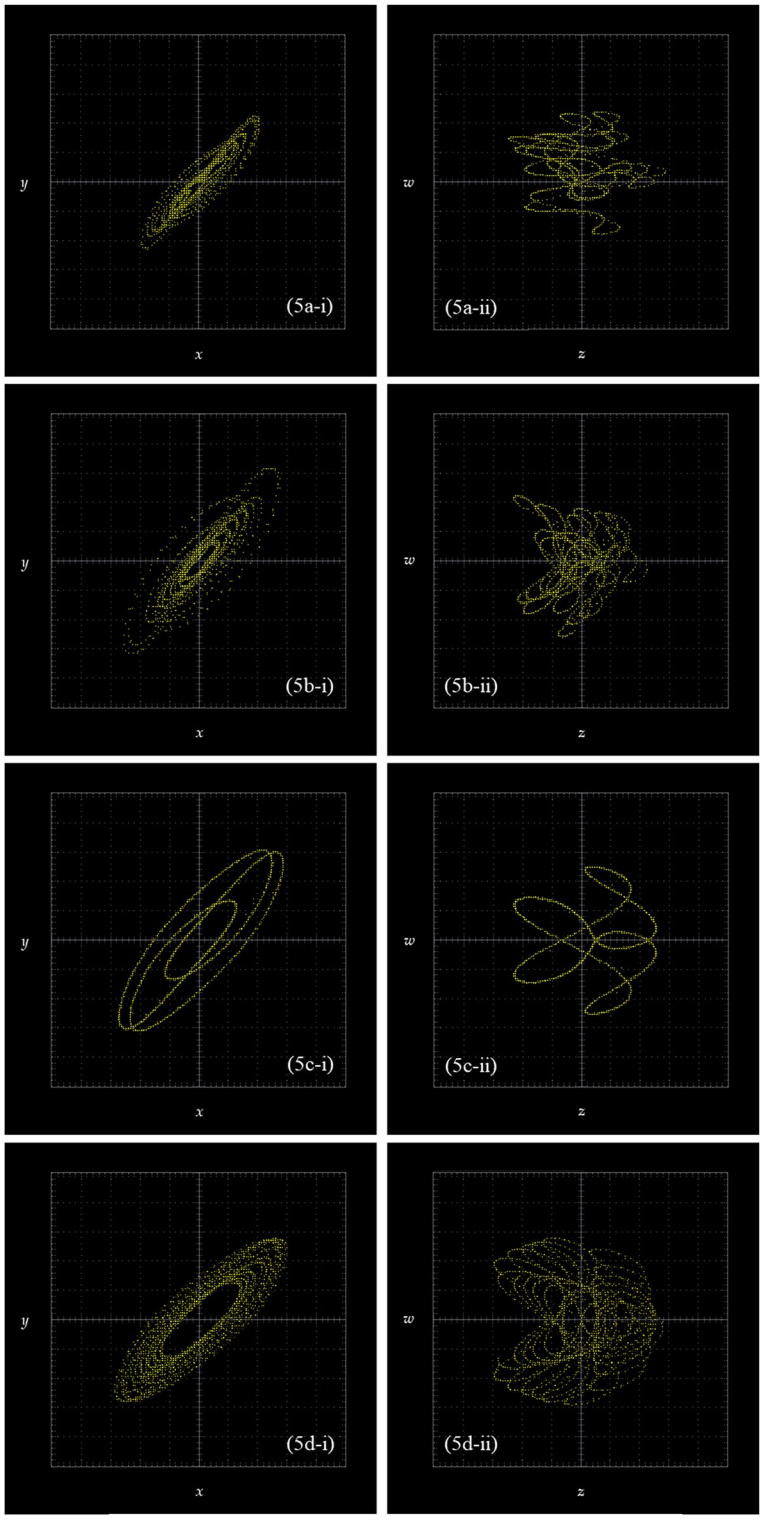

Fig. 5. Experimental results of STM32 implementation ((**a**) hyperchaotic orbit ($d = 15$); (**b**) chaotic orbit ($d = 70$); (**c**) periodic orbit ($d = 100$); (**d**) quasi-periodic orbit ($d = 120$)).

the slopes of the end of the time period, and their $x[\tau]$, $y[\tau]$, $z[\tau]$ and $w[\tau]$ values are determined by h_3_x, h_3_y, h_3_z and h_3_w. Finally, the values $x[\tau + 1]$, $y[\tau + 1]$, $z[\tau + 1]$ and $w[\tau + 1]$ of the $\tau + 1$ moment are calculated.

Then, the digital signals are converted into analog signals through SPI, which is controlled by the DAC converter, and then displayed on the oscilloscope. The embedded system STM32 is depicted in Fig. 4.

When $d = 15$, Fig. 5a shows the phase transition trajectory of hyperchaotic signal by oscilloscope. When $d = 70$, Fig. 5b shows the phase transition trajectory of chaotic signal by oscilloscope. When $d = 100$, Fig. 5c

shows the phase transition trajectory of periodic orbit by oscilloscope. When $d = 120$, Fig. 5d shows the phase transition trajectory of quasi-periodic orbit by oscilloscope.

The alignment of simulation outcomes from MATLAB with the results derived from the hyperchaotic system implemented on the STM32, as illustrated in Figs. 3 and 5, establishes the coherence between the two datasets. This alignment not only affirms the precision and dependability of the simulation model but also underscores the efficacy of deploying the hyperchaotic system on the STM32 platform.

Image encryption

In today's digital society, image encryption, as a crucial branch of information security, plays a key role in safeguarding the privacy of image data and preventing unauthorized access. With the widespread use of images in communication, storage, and sharing, effectively encrypting them has become paramount. Image encryption aims to transform images into unintelligible forms to ensure the security of personal privacy, business secrets, and sensitive information through the utilization of complex mathematical algorithms and keys. The advancement of encryption technology provides essential tools for preventing image information leakage, safeguarding national security, and ensuring confidentiality in fields like medical imaging.

The image encryption technology based on hyperchaotic system-generated hyperchaotic keys exhibits significant necessity and superiority. The hyperchaotic system, with its high dimensionality, complexity, and nonlinear characteristics, forms the basis for generating keys with extremely high randomness, thereby enhancing

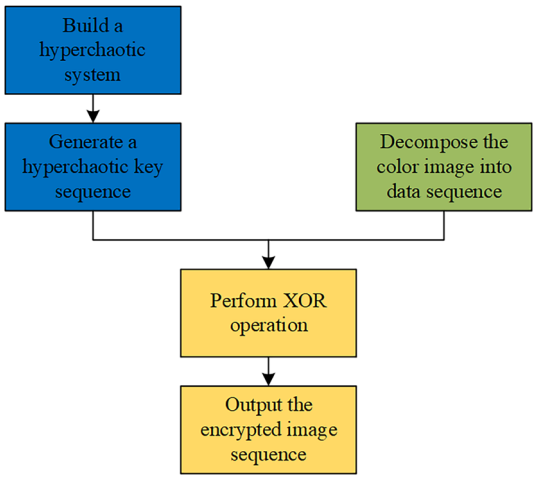

Fig. 6. The image encryption process using the hyperchaotic key sequence.

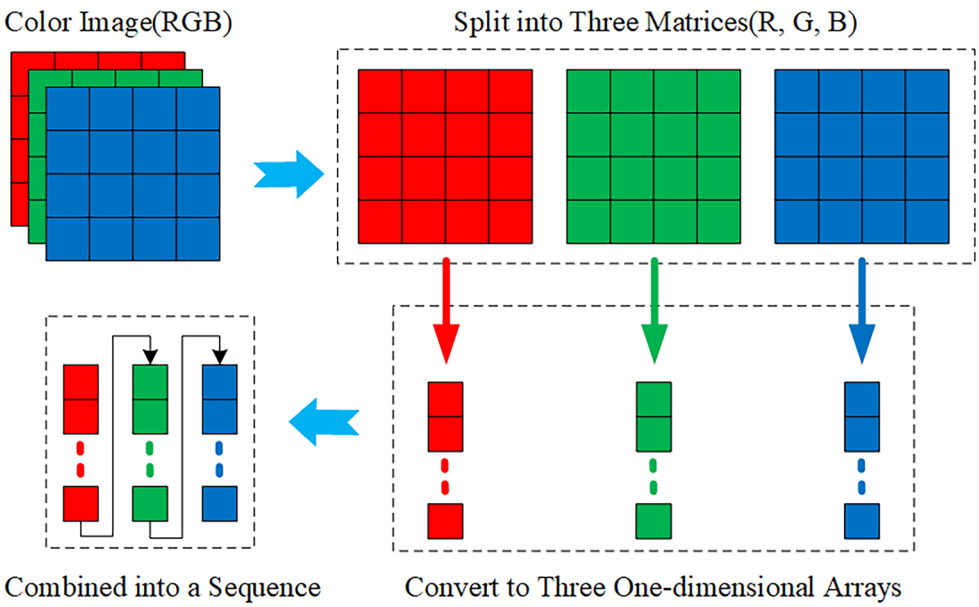

Fig. 7. The process diagram of converting the original image into a data sequence.

the security of encryption algorithms. This technology effectively addresses modern challenges in image security by generating complex and unpredictable key sequences, protecting images from unauthorized access and information leakage. Compared to traditional encryption methods, image encryption based on hyperchaotic systems is more resilient to attacks. Supported by embedded hardware like STM32, it achieves efficient encryption and decryption processes, providing an advanced and practical solution for image encryption in the field of information security.

Encryption method

To highlight the simplicity and non-linear advantages of implementing hyperchaotic keys, as well as fully leveraging the benefits of ciphertext interleaving diffusion technology in image encryption to improve its ability to resist illegal attacks, a method of image encryption bit-XOR operation based on hyperchaotic key sequence is proposed. Characterized by suitability for image encryption, non-linear ciphertext, and easy implementation, this approach enhances the speed of ciphertext diffusion. The image encryption process based on the generation of a hyperchaotic key sequence from the hyperchaotic system is shown in Fig. 6, involves four steps:

Firstly, construct a hyperchaotic system and analyze its dynamic characteristics to demonstrate its complexity, randomness, and nonlinearity, as described in section "Dynamical analysis of a new hyperchaotic system".

Secondly, generate hyperchaotic key sequences that meet the requirements based on the constructed hyperchaotic system, with each hyperchaotic key represented by 16 bits, as detailed in subsection "Hyperchaotic key sequence". Next, decompose the color image to be encrypted (with each pixel represented by 8 bits) into three matrices representing the RGB channels. Convert each matrix into a column vector and concatenate them to form a complete column vector, serving as the data sequence for the image, as shown in Fig. 7.

Then, perform bitwise bit-XOR operations between each value of the image sequence and the corresponding value of the hyperchaotic sequence, obtaining an 8-bit encrypted sequence according to the following formula (6).

$$\boldsymbol{R} = \boldsymbol{S} \oplus \boldsymbol{K}_H \oplus \boldsymbol{K}_L \tag{6}$$

where $\boldsymbol{R}$ sequence is the encrypted image sequence using bit-XOR operation, $\boldsymbol{S}$ sequence is the original image sequence, and $\boldsymbol{K}$ sequence is the hyperchaotic key sequence, $\boldsymbol{K}_H$ and $\boldsymbol{K}_L$ represent the high 8-bit and low 8-bit sub-sequences of the key sequence $\boldsymbol{K}$.

Finally, output the encrypted image sequence through the serial port of the embedded hardware STM32. The decryption of the image follows the reverse process of encryption, with the decryption bit-XOR operation being consistent with the encryption process.

Hyperchaotic key sequence

Due to the inherent high randomness and complexity of the hyperchaotic system, the encryption algorithm's security is significantly strengthened. The generation of the hyperchaotic key sequence involves transforming the hyperchaotic sequence into the required key sequence through a specific quantization algorithm, outlined in the following steps:

State values A_k					
x_k	y_k	z_k	w_k	State values B_k	Key sequence K
+	+	+	+	$\{x_k, y_k, z_k, w_k\}$	$\{bx_1, by_1, bz_1, bw_1, bx_2, by_2, bz_2, bw_2, bx_3, by_3, bz_3, bw_3, \ldots\ldots bx_k, by_k, bz_k, bw_k\}, 0 < k \leq N_2$
+	+	+	−	$\{x_k, y_k, w_k, z_k\}$	
+	+	−	+	$\{x_k, z_k, y_k, w_k\}$	
+	+	−	−	$\{x_k, z_k, w_k, y_k\}$	
+	−	+	+	$\{y_k, x_k, z_k, w_k\}$	
+	−	+	−	$\{y_k, x_k, w_k, z_k\}$	
+	−	−	+	$\{y_k, z_k, x_k, w_k\}$	
+	−	−	−	$\{y_k, z_k, w_k, x_k\}$	
−	+	+	+	$\{z_k, x_k, y_k, w_k\}$	
−	+	+	−	$\{z_k, x_k, w_k, y_k\}$	
−	+	−	+	$\{z_k, y_k, x_k, w_k\}$	
−	+	−	−	$\{z_k, y_k, w_k, x_k\}$	
−	−	+	+	$\{w_k, x_k, y_k, z_k\}$	
−	−	+	−	$\{w_k, x_k, z_k, y_k\}$	
−	−	−	+	$\{w_k, y_k, x_k, z_k\}$	
−	−	−	−	$\{w_k, y_k, z_k, x_k\}$	

Table 2. The mapping relationship illustrating the sorting rule of state values $\boldsymbol{A}$ and $\boldsymbol{B}$ in correlation with the generation of the key sequence $\boldsymbol{K}$.

Step 1: Pre-iterate the hyperchaotic system N_1 times to eliminate transient effects as it enters a hyperchaotic state.

Step 2: Iterate the hyperchaotic system N_2 times to obtain a new set of state values, $\boldsymbol{A} = \{Ax, Ay, Az, Aw\}$, where $Ax = \{x_1, x_2, x_3, x_4\}^T$, $Ay = \{y_1, y_2, y_3, y_4\}^T$, $Az = \{z_1, z_2, z_3, z_4\}^T$, and $Aw = \{w_1, w_2, w_3, w_4\}^T$, $0 < k \leq N_2$. Scale down the state values $\boldsymbol{A}$ to $1/p$ and take q decimal places to generate new state values $\boldsymbol{B} = \{Bx, By, Bz, Bw\}$, where $Bx = \{bx_1, bx_2, bx_3, bx_4\}^T$, $By = \{by_1, by_2, by_3, by_4\}^T$, $Bz = \{bz_1, bz_2, bz_3, bz_4\}^T$, and $Bw = \{bw_1, bw_2, bw_3, bw_4\}^T$, $0 < k \leq N_2$. The conversion formula is given by:

$$\begin{cases} bx_i = x_i * 10^{(p-q)} - \left\lfloor x_i * 10^{(p-q)} \right\rfloor, & if\ x_i \geq 0, i = 1, 2, \cdots, k \\ bx_i = \left\lfloor x_i * 10^{(p-q)} \right\rfloor - x_i * 10^{(p-q)}, & if\ x_i < 0, i = 1, 2, \cdots, k \\ by_i = y_i * 10^{(p-q)} - \left\lfloor y_i * 10^{(p-q)} \right\rfloor, & if\ y_i \geq 0, i = 1, 2, \cdots, k \\ by_i = \left\lfloor y_i * 10^{(p-q)} \right\rfloor - y_i * 10^{(p-q)}, & if\ y_i < 0, i = 1, 2, \cdots, k \\ bz_i = z_i * 10^{(p-q)} - \left\lfloor z_i * 10^{(p-q)} \right\rfloor, & if\ z_i \geq 0, i = 1, 2, \cdots, k \\ bz_i = \left\lfloor z_i * 10^{(p-q)} \right\rfloor - z_i * 10^{(p-q)}, & if\ z_i < 0, i = 1, 2, \cdots, k \\ bw_i = w_i * 10^{(p-q)} - \left\lfloor w_i * 10^{(p-q)} \right\rfloor, & if\ w_i \geq 0, i = 1, 2, \cdots, k \\ bw_i = \left\lfloor w_i * 10^{(p-q)} \right\rfloor - w_i * 10^{(p-q)}, & if\ w_i < 0, i = 1, 2, \cdots, k \end{cases} \tag{7}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, p and q are adjustable parameters, representing scaling down the value by $1/p$ and taking q decimal places as the new state value.

Step 3: Calculate the key sequence. Adjust the order of the state values $\boldsymbol{B}$ based on positive and negative changes in the state values $\boldsymbol{A}$ to generate the key sequence $= \{K_1, K_2, K_3, K_4\}, 0 < k \leq N_2$, following the mapping relationship outlined in Table 2.

The randomness of the hyperchaotic key sequence can be evaluated using the NIST SP800-22 standard. This standard, developed by the National Institute of Standards and Technology (NIST) in the United States, is one of the authoritative standards for pseudo randomness testing. It includes a total of 15 test metrics, examining the deviation of the tested sequence from ideal random sequences from different perspectives in terms of statistical characteristics. With a significance level of 0.01, a test group number of 100, a group length of 1024000 bits, and a confidence interval of [0.93,1], the test results are presented in Table 3. For items with a test frequency not less than 2, Table 3 provides *P*-values and the minimum values of pass rates. It can be observed that the hyperchaotic key sequence generated by the algorithm has successfully passed all 15-test metrics, further confirming its excellent randomness.

Experimental results

Since the STM32 achieves synchronized control of the encryption module and the interface module through the master clock, eliminating the need for separate control of the hyperchaotic system's encryption and decryption processes. This approach avoids the synchronization issues between the encryption and decryption modules that are commonly encountered in traditional encryption systems.

Specifically, in this experiment, image encryption based on the hyperchaotic key sequence was implemented using the embedded hardware STM32F103ZET6. Due to the absence of an integrated image acquisition and display module in the STM32F103ZET6, data input and output were facilitated in the form of a list. The resulting

No	Project name	Number of tests	*P*-values	Pass rates	Test results
1	Frequency	1	0.000017	98%	Yes
2	BlockFrequency	1	0.000014	98%	Yes
3	CumulativeSums	2	0.000058	96%	Yes
4	Runs	1	0.000033	97%	Yes
5	LongestRun	1	0.514124	99%	Yes
6	Rank	1	0.000051	99%	Yes
7	FFT	1	0.071177	95%	Yes
8	NonOverlappingTemplate	148	0.030806	98%	Yes
9	OverlappingTemplate	1	0.021999	100%	Yes
10	Universal	1	0.595549	100%	Yes
11	ApproximateEntropy	1	0.389596	95%	Yes
12	RandomExcursions	8	0.468595	97%	Yes
13	RandomExcursionsVariant	18	0.350485	98%	Yes
14	Serial	2	0.474986	97%	Yes
15	LinearComplexity	1	0.319084	93%	Yes

Table 3. The NIST SP 800-22 test result for hyperchaotic key sequence.

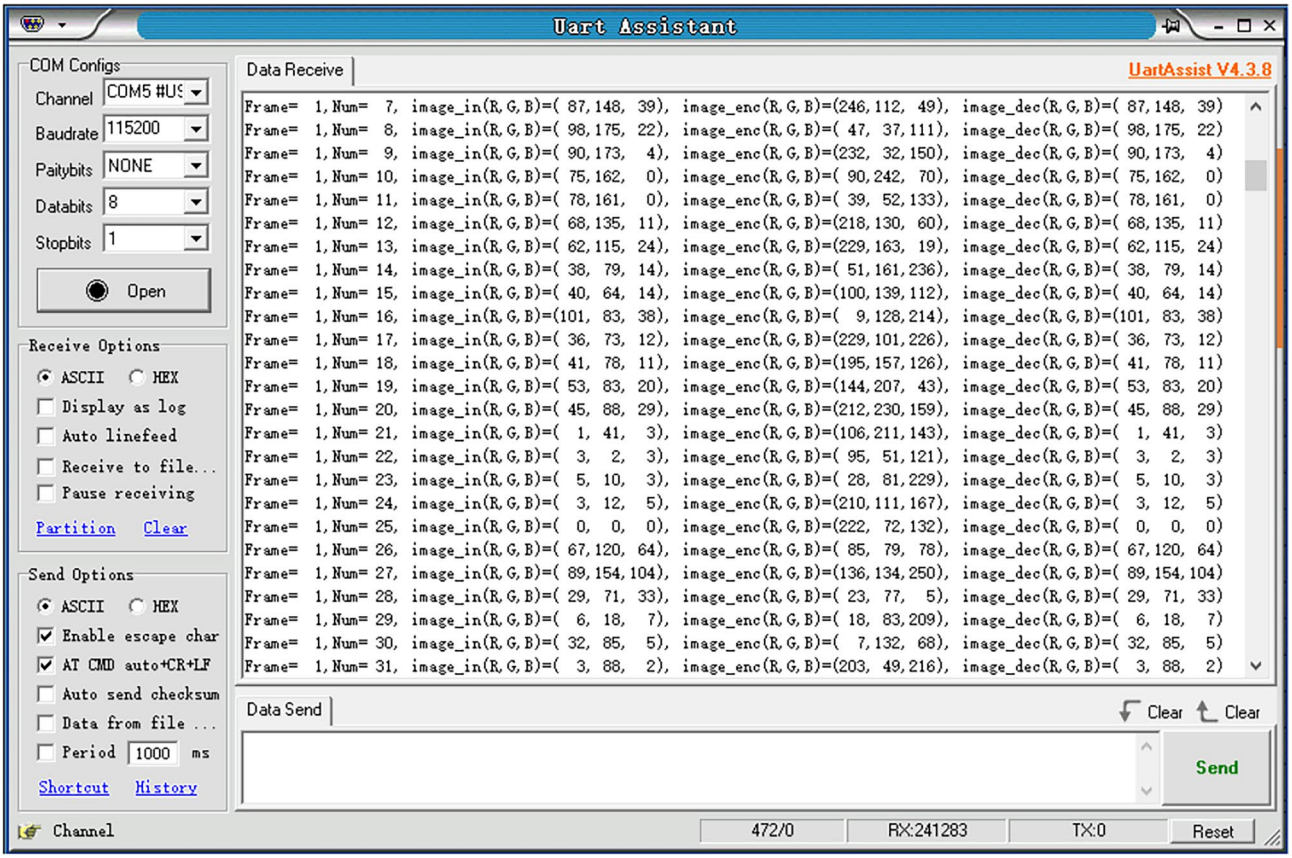

Fig. 8. The image encrypted and decrypted sequences printed by the serial port debugging tool.

encrypted and decrypted image sequences were then transmitted to the PC via the serial port of the embedded hardware STM32F103ZET6. Subsequently, the data was visualized and printed using a serial port debugging tool, as shown in Fig. 8.

Although the embedded hardware STM32F103ZET6 can complete the experimental process, it lacks data analysis tools, making it difficult to visually compare the information before and after image encryption. Therefore, it is necessary to transmit the data sequence generated in the embedded hardware STM32F103ZET6 to the PC through the serial port and use simulation software MATLAB for analysis. Figure 9a and d show the color image before encryption, Fig. 9b and e display the color image after encryption, Fig. 9c and f depict the color image after decryption.

Comparing Fig. 9a, b, c, and d, e, f, the original images are the 256*256 color picture. After encryption with the hyperchaotic key sequences, the images are completely flooded with noise, making them impossible to discern the real information of the original image. After decryption, the images show no differences in the time domain compared to the original images. The serial port data printed by the embedded hardware STM32F103ZET6 also confirms that the image sequences before and after encryption are identical.

The real information of the original image is completely concealed after encryption, and the encrypted image sequence exhibits a Gaussian random white noise distribution, validating the high security of the image encryption method based on the hyperchaotic key sequence. The decrypted image is identical to the original image, demonstrating that image encryption based on the hyperchaotic key sequence does not have a negative impact on the original data, and using the STM32F103ZET6 embedded hardware system as the encryption device does not introduce noise. Therefore, the image encryption method based on the hyperchaotic key sequence generated by the hyperchaotic system is both secure and effective.

Performance evaluation

To evaluate the performance of this scheme, five key aspects were analyzed: key space analysis, histogram similarity analysis, information entropy analysis, statistical attack analysis, differential attack analysis, key sensitivity analysis, and correlation analysis. The performance analysis was conducted on a Windows operating system using Python software, with a PC configuration of an Intel i5-12450H 2.0 GHz CPU, 16 GB RAM, and 512 GB ROM. MATLAB 2019a was used for the analysis software.

Key space analysis

The key space size is a crucial factor in determining whether an encryption algorithm can resist brute-force attacks. Generally, an encryption algorithm is considered secure against such attacks when its key space exceeds

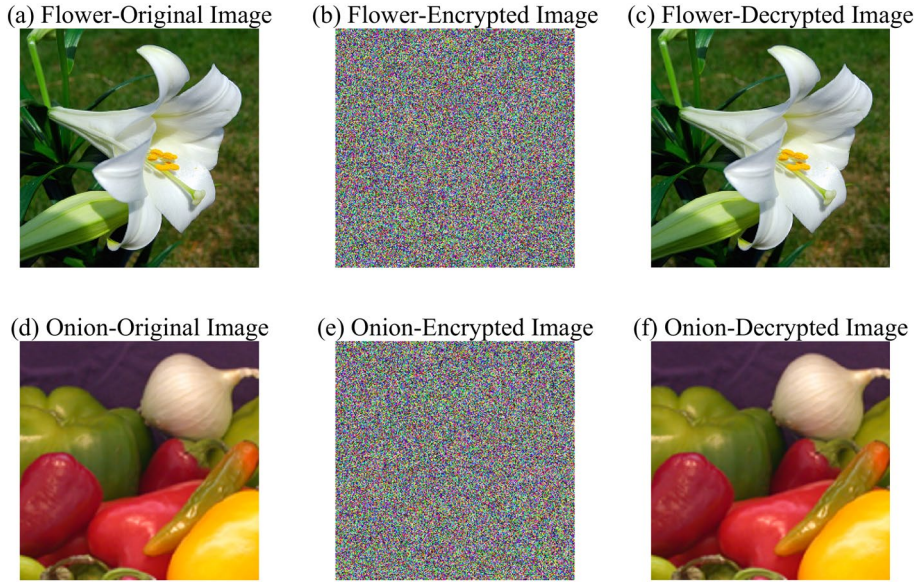

Fig. 9. Image and histogram of encryption/decryption.

2^{100}. For the hyperchaotic mapping, the state values (x, y, z, w) range between $(-100,100)$, and six decimal places are used as the key. With double-precision floating-point data, accurate to 15 decimal places, the key space can reach approximately $2 \times 10^{15} \times 2 \times 10^{15} \times 2 \times 10^{15} \times 2 \times 10^{15} \approx 2^{203}$, which is equivalent to a 203-bit key length. When including system parameters d and pre-iteration times N_0, the key space expands even further, providing strong resistance against exhaustive attacks.

Table 4 presents the key space test results of different schemes, showing that the reference schemes have larger key spaces than the proposed method. However, those systems are more complex, with dimensions exceeding six, which affects encryption efficiency and storage requirements. Although the key space in this scheme is relatively smaller, it still far exceeds the minimum key space requirements for encryption systems.

Histogram similarity analysis

To comprehensively demonstrate the effectiveness of the proposed hyperchaotic system for encryption, 71 images were selected from the MATLAB 2019a toolbox for encryption validation. The effectiveness of the encryption scheme was verified by analyzing the histogram similarity before and after encryption. The test results are shown in Table 5.

Histogram similarity is an effective metric for evaluating the performance of an image encryption scheme. A lower histogram similarity indicates a significant difference between the histograms of the original and encrypted images, suggesting that the encryption algorithm has successfully altered the pixel distribution, making it difficult to find any correlation between the two. In practical applications, the histogram similarity after image encryption should generally be less than 0.3. As shown in Table 5, among the 71 images tested, the highest histogram similarity was 0.2966372, and the lowest was 0.0089738, which meets the requirements for evaluating the effectiveness of image encryption based on histogram similarity.

Information entropy analysis

Information entropy is a crucial metric for measuring the amount of information or uncertainty within an image, reflecting the complexity of the image by evaluating the randomness of pixel values. In image encryption, high entropy indicates that the encrypted image possesses a high degree of randomness and uncertainty, making it extremely difficult to reverse-engineer the original image through statistical methods. Therefore, entropy is commonly used to assess the effectiveness of encryption algorithms. Ideally, the entropy of an encrypted image should approach the maximum value (typically close to 8 for an 8-bit image), indicating a uniform pixel distribution and that the image resembles random noise, effectively safeguarding the original image information.

Table 6 presents the entropy values before and after encryption for 71 images, showing that the entropy of the encrypted images is close to 8, demonstrating the high randomness and uncertainty achieved by the proposed encryption scheme.

Encryption scheme	Proposed	Ref. [28]	Ref. [39]	Ref. [40]	Ref. [41]	Ref. [42]
Key space	$> 2^{203}$	2^{398}	2^{395}	2^{246}	2^{325}	2^{356}

Table 4. Key space of different schemes.

No	Test images	Image sizes	Histogram similarity	No	Test images	Image sizes	Histogram similarity
1	baby	2240*3584*3	0.04420524	37	llama	1280*832*3	0.15345570
2	bag	128*192*3	0.04760596	38	lowlight_1	704*960*3	0.18525304
3	blobs	320*256*3	0.11474805	39	lowlight_2	1984*1280*3	0.13422752
4	car_1	640*448*3	0.04177567	40	micromarket	3456*2304*3	0.04981033
5	car_2	640*448*3	0.18803619	41	office_1	896*576*3	0.10119631
6	car_3	640*448*3	0.1295514	42	office_2	896*576*3	0.16850714
7	car_4	640*448*3	0.05781802	43	office_3	896*576*3	0.2954075
8	car1	3456*2304*3	0.05509924	44	office_4	896*576*3	0.05546688
9	car2	3456*2304*3	0.05189874	45	office_5	896*576*3	0.14683144
10	circles	256*256*3	0.0089738	46	office_6	896*576*3	0.13143341
11	circlesBrightDark	512*512*3	0.09751573	47	onion	256*256*3	0.05183778
12	coins	256*192*3	0.07121657	48	parkavenue	2048*1536*3	0.14046591
13	coloredChips	512*384*3	0.2908124	49	peacock	1024*768*3	0.13711488
14	concordaerial	3008*1984*3	0.14349766	50	pears	704*448*3	0.14576219
15	concordorthophoto	2944*2176*3	0.13939115	51	pillsetc	512*384*3	0.18243526
16	DistortedImage	448*640*3	0.13410182	52	printedtext	1600*896*3	0.14413334
17	eSFRTestImage	3072*1792*3	0.23765657	53	saturn	1152*1472*3	0.29165825
18	fabric	640*448*3	0.14163200	54	sevilla	1600*896*3	0.05219821
19	flamingos	1280*960*3	0.04767290	55	sherlock	960*640*3	0.14022593
20	flower	256*256*3	0.27108517	56	snowflakes	320*64*3	0.13637836
21	foggyroad	2048*1536*3	0.27971324	57	strawberries	1024*640*3	0.14117495
22	foggysf1	3456*2304*3	0.05197918	58	tape	512*384*3	0.13746690
23	foggysf2	3456*2304*3	0.04182551	59	testpat1	256*256*3	0.15047596
24	foosball	3456*2304*3	0.14954731	60	text	256*256*3	0.00921987
25	football	320*256*3	0.13200099	61	threads	448*448*3	0.2376722
26	gantrycrane	384*256*3	0.23377732	62	tissue	768*448*3	0.14848739
27	glass	256*128*3	0.14447569	63	toyobjects	320*320*3	0.08204885
28	greens	448*256*3	0.13989351	64	toysflash	896*640*3	0.23470400
29	hallway	3456*2304*3	0.2446188	65	toysnoflash	896*640*3	0.13711072
30	hands1	192*320*3	0.2966372	66	trailer	1024*640*3	0.14140032
31	hands1-mask	320*192*3	0.08397624	67	wagon	768*1024*3	0.2033131
32	hands2	192*320*3	0.2891822	68	westconcordaerial	320*384*3	0.13978115
33	hestain	256*192*3	0.23509654	69	westconcordorthophoto	320*320*3	0.2870106
34	indiancorn	1024*768*3	0.23153046	70	yellowlily	1216*1600*3	0.27546623
35	kobi	1600*1216*3	0.05287075	71	yellowlily-segmented	1216*1600*3	0.07700115
36	lighthouse	448*640*3	0.13331444				

Table 5. Histogram similarity before and after image encryption.

Table 7 compares the entropy values of images after encryption using the proposed scheme with other schemes, further illustrating that the proposed method effectively confuses pixel values and enhances the security of the encrypted images.

Statistical attack analysis

The distribution histograms of the original image and the encrypted image are shown in Fig. 10a and c, b and d respectively.

From Fig. 10, it can be observed that the distribution of the original image histogram is uneven, with a higher probability density in the low pixel segment and a lower probability density in the high pixel segment. In addition, the distribution of the encrypted image is flat and uniform, with similar probability densities in the low and high pixel segments. This indicates that the probability of pixel values in the encrypted image tends to be equal. Therefore, this encryption algorithm can effectively resist statistical analysis attacks.

Differential attack analysis

According to the principles of cryptography, the stronger the sensitivity of the key to plaintext, the stronger the resistance against differential attacks. The sensitivity of plaintext can be measured using the number of signals change rate (NSCR) and the unified average changing intensity (UACI). They respectively represent the proportion and degree of change in pixel values of the encrypted image when a pixel value of the original image is

No	Test images	Information entropy		No	Test images	Information entropy	
		Original image	Encrypted image			Original image	Encrypted image
1	baby	7.694291	7.996322	37	llama	7.364563	7.999322
2	bag	7.489169	7.998253	38	lowlight_1	6.410127	7.999316
3	blobs	1.303949	7.998321	39	lowlight_2	6.247875	7.996329
4	car_1	7.658223	7.996329	40	micromarket	7.816366	7.997327
5	car_2	6.473951	7.997325	41	office_1	4.870706	7.997320
6	car_3	5.555290	7.996325	42	office_2	6.319726	7.998307
7	car_4	1.109169	7.997319	43	office_3	7.205910	7.996309
8	car1	7.524531	7.996324	44	office_4	7.561897	7.998328
9	car2	7.742344	7.998326	45	office_5	7.667470	7.999335
10	circles	0.797671	7.996289	46	office_6	6.883114	7.999316
11	circlesBrightDark	1.288123	7.999320	47	onion	7.436057	7.997305
12	coins	6.346112	7.998320	48	parkavenue	7.157566	7.998332
13	coloredChips	7.204383	7.996335	49	peacock	7.447847	7.996323
14	concordaerial	7.175397	7.997325	50	pears	7.264007	7.998329
15	concordorthophoto	7.355761	7.999322	51	pillsetc	6.007417	7.998332
16	DistortedImage	7.854813	7.996333	52	printedtext	7.679555	7.999326
17	eSFRTestImage	6.739018	7.996324	53	saturn	4.985464	7.999323
18	fabric	7.348980	7.998306	54	sevilla	7.839214	7.996324
19	flamingos	7.359811	7.996319	55	sherlock	7.366483	7.999317
20	flower	7.508741	7.996326	56	snowflakes	4.872856	7.998238
21	foggyroad	6.612072	7.999319	57	strawberries	7.642410	7.998328
22	foggysf1	7.450419	7.999330	58	tape	6.741722	7.998323
23	foggysf2	7.180980	7.998328	59	testpat1	2.611444	7.999310
24	foosball	7.547328	7.996327	60	text	0.454175	7.997289
25	football	6.713418	7.999297	61	threads	5.186503	7.996315
26	gantrycrane	6.683417	7.996299	62	tissue	7.534090	7.998314
27	glass	7.340917	7.996271	63	toyobjects	1.597250	7.999326
28	greens	7.422649	7.998303	64	toysflash	7.274874	7.998305
29	hallway	7.286544	7.996329	65	toysnoflash	7.404918	7.996324
30	hands1	6.848025	7.997260	66	trailer	7.362810	7.999330
31	hands1-mask	1.052358	7.997301	67	wagon	7.707832	7.996315
32	hands2	6.891528	7.999282	68	westconcordaerial	7.166493	7.999324
33	hestain	7.277335	7.998270	69	westconcordortho-photo	7.739572	7.997343
34	indiancorn	7.737863	7.999322	70	yellowlily	7.039695	7.999315
35	kobi	7.420971	7.997329	71	yellowlily-segmented	1.599543	7.998321
36	lighthouse	7.830373	7.997321				

Table 6. Information entropy before and after image encryption.

No	Encryption scheme	Test images	Information entropy (Encrypted image)
1	Proposed	Average	7.997826
2	Ref. [28]	Average	7.998300
3	Ref. [43]	Baboon	7.977383
4	Ref. [44]	Peppers	7.996367
5	Ref. [45]	Baboon	7.998233

Table 7. Information entropy of different schemes.

randomly altered. Assuming that the values of two ciphertext signals at point i are $C(i)$ and $C\prime(i)$, if $C(i) = C\prime(i)$, then $D(i) = 0$; if $C(i) \neq C\prime(i)$, then $D(i) = 1$. Then the definitions of NSCR and UACI are as follows Eq. (8):

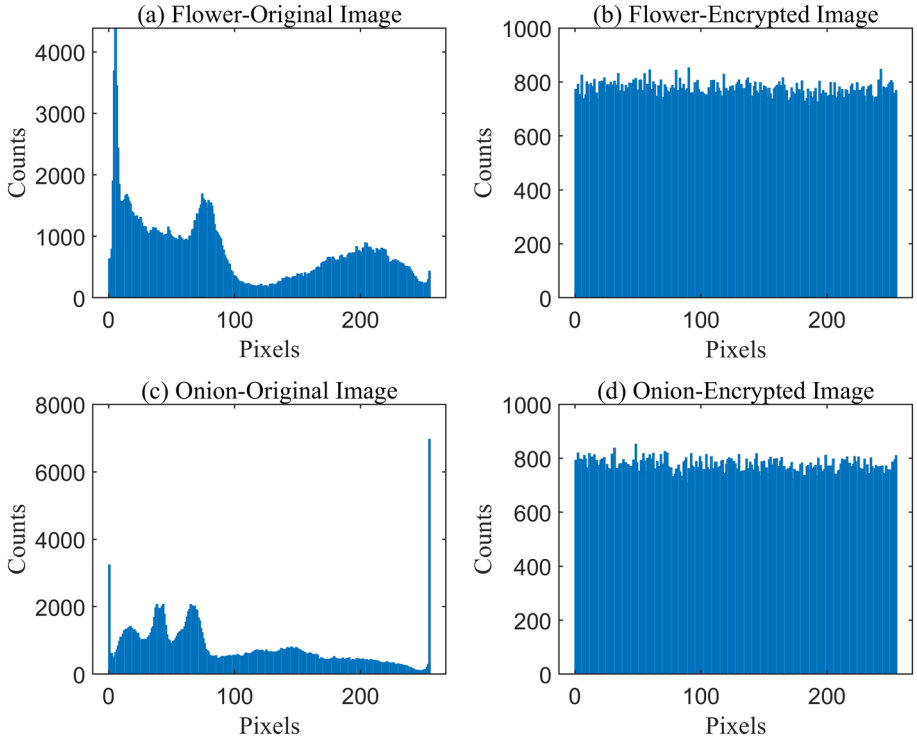

Fig. 10. Histograms distribution of images before and after encryption.

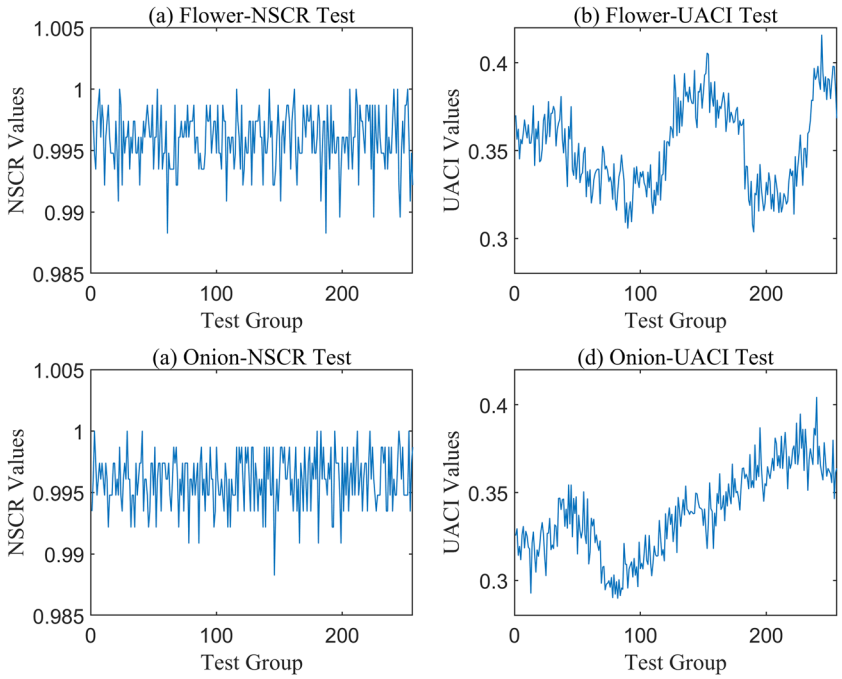

Fig. 11. Plaintext sensitivity test.

$$\begin{aligned} NSCR &= \frac{\sum_{i=1}^{N} D(i)}{N} \times 100\% \\ UACI &= \frac{1}{N} \sum_{i=1}^{N} \frac{|C'(i) - C(i)|}{255} \times 100\% \end{aligned} \tag{8}$$

No	Encryption scheme	Test images	NPCR	UACI
1	Proposed	Flower	99.59%	35.22%
		Onion	99.60%	33.84%
2	Ref. [28]	Average	99.61%	33.47%
3	Ref. [43]	Lena	99.60%	33.88%
4	Ref. [44]	Average	99.62%	33.40%
5	Ref. [45]	Baboon	99.63%	33.51%
6	Ref. [46]	Lena	99.61%	33.44%
7	Ref. [47]	Lena	99.57%	33.44%

Table 8. NPCR and UACI of different schemes.

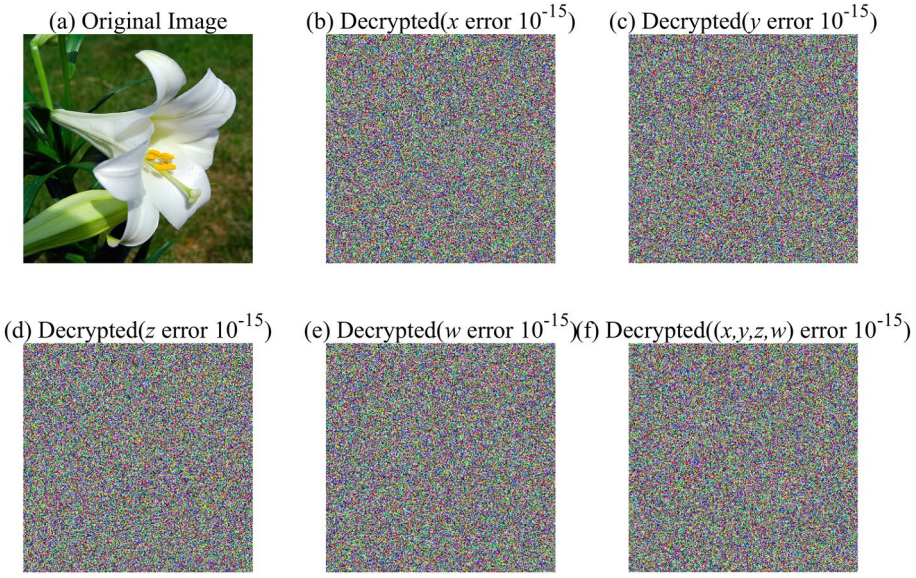

Fig. 12. Correctly decrypted image and incorrectly decrypted images.

where N is the number of image pixels. In this experiment, 256 groups were selected for encryption, with each group consisting of 2 image sequences: one original image sequence and one encrypted image sequence.

The NSCR and UACI value curves for the 256 sets of ciphertext signals obtained from the Flower image are shown in Fig. 11a and b, respectively, with average values of 99.59% for NSCR and 35.22% for UACI. Similarly, the NSCR and UACI value curves for the 256 sets of ciphertext signals obtained from the Onion image are displayed in Fig. 11c and d, with average values of 99.60% for NSCR and 33.84% for UACI. These results indicate that the bit-XOR operation causes slight changes in the lower bits of the plaintext to induce significant changes in the higher bits of the ciphertext, while changes in the higher bits of the plaintext can simultaneously affect both the higher and lower bits of the ciphertext. This enhances the overall variation magnitude of the ciphertext. Additionally, the differential attack test results from other schemes[28,43–47] are summarized in Table 8, demonstrating that the proposed algorithm not only resists differential attacks but also performs stably.

Key sensitivity analysis

A good cryptographic algorithm must be highly sensitive to the key, meaning that even slight differences in encryption keys should result in significant changes in the ciphertext sequence for the same plaintext. Similarly, for the same ciphertext, slight differences in the decryption key should lead to vastly different decryption results.

In this experiment, the initial values (x_0, y_0, z_0, w_0) of the system are used as the key. Initially, the encryption is performed using the key (1,1, 1,1), followed by decryption using five sets of decryption keys with minor differences, including $(1 + 10^{-15}, 1,1, 1)$, $(1,1 + 10^{-15}, 1,1)$, $(1,1, 1 + 10^{-15}, 1)$, $(1,1, 1,1 + 10^{-15})$, and $(1 + 10^{-15}, 1 + 10^{-15}, 1 + 10^{-15}, 1 + 10^{-15})$.

Figure 12a is the original image, Fig. 12b is the decrypted image when the error in the initial decryption key x_0 is 10^{-15}, Fig. 12c is the decrypted image when the error in the initial decryption key y_0 is 10^{-15}, Fig. 12d is the decrypted image when the error in the initial decryption key z_0 is 10^{-15}, Fig. 12e is the decrypted image when the error in the initial decryption key w_0 is 10^{-15}, Fig. 12f is the decrypted image when the errors in the initial decryption keys x_0, y_0, z_0, and w_0 are all 10^{-15}. Where it can be observed that the correctly decrypted

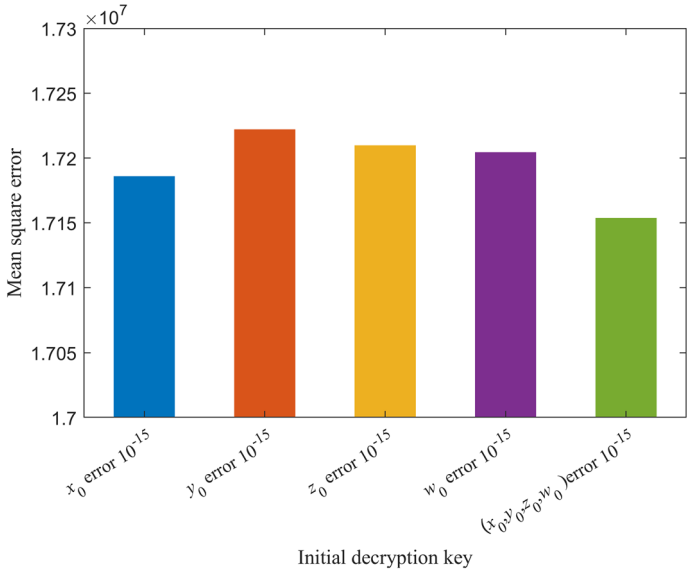

Fig. 13. The mean square error of error decryption to original image.

Number of plaintext-ciphertext pairs *M*	Original image correlation				Encrypted image correlation			
	Red	Green	Blue	Average	Red	Green	Blue	Average
8	0.9964	0.9985	0.9983	0.9977	0.0240	0.4134	−0.3087	0.0429
64	0.9905	0.9796	0.9909	0.9870	−0.0819	0.0687	0.1274	0.0381
128	0.9889	0.9881	0.9947	0.9905	0.0626	0.0299	−0.1489	−0.0188
256	0.9896	0.9797	0.9865	0.9853	0.0350	−0.0140	−0.0784	−0.0191

Table 9. Correlation of adjacent flower image pixels.

signal matches the original image, indicating successful data recovery, while the remaining five sets of erroneous decryption signals resemble noise signals, demonstrating the extreme sensitivity of the algorithm to the key.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(P'(i) - P(i)\right)^2 \tag{9}$$

Let the original image be denoted as P and the decrypted image as $P\prime$. Their mean square error is calculated using Eq. (9), and the mean square error between the erroneous decryption signal and the original image is presented in Fig. 13. It can be observed that even slight errors in the decryption key result in vastly different decryption results, thus numerically proving the sensitivity of the algorithm to the key.

Correlation analysis

Pixel values in images are not independent, with small differences in adjacent pixel amplitudes leading to high correlation. Therefore, one of the objectives of encryption algorithms is to reduce the correlation between adjacent pixel values. Lower correlation implies better confusion effects and higher security. Correlation can be measured using correlation coefficients, calculated between adjacent amplitude values of plaintext and ciphertext, as fallowing Eq. (10).

$$Corr = \frac{\sum_{i=1}^{M}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\left(\sum_{i=1}^{M}(x_i - \overline{x})^2\right)\left(\sum_{i=1}^{M}(y_i - \overline{y})^2\right)}} \tag{10}$$

Let x_i and y_i represent the i-th pair of adjacent pixel values randomly selected from the image pixel value matrix, and let $\overline{x}$ and $\overline{y}$ be their respective averages. M represents the total number of pairs, with M set to 8, 64, 128, and 256. The calculated results are presented in Table 9.

A comparison of the data reveals that adjacent original images have a high correlation ($Corr \to 1$), while adjacent encrypted signals are almost uncorrelated ($Corr \to 0$), indicating good confusion effects. As the correlation coefficient of plaintext increases, the correlation coefficient of ciphertext decreases. This approach effectively

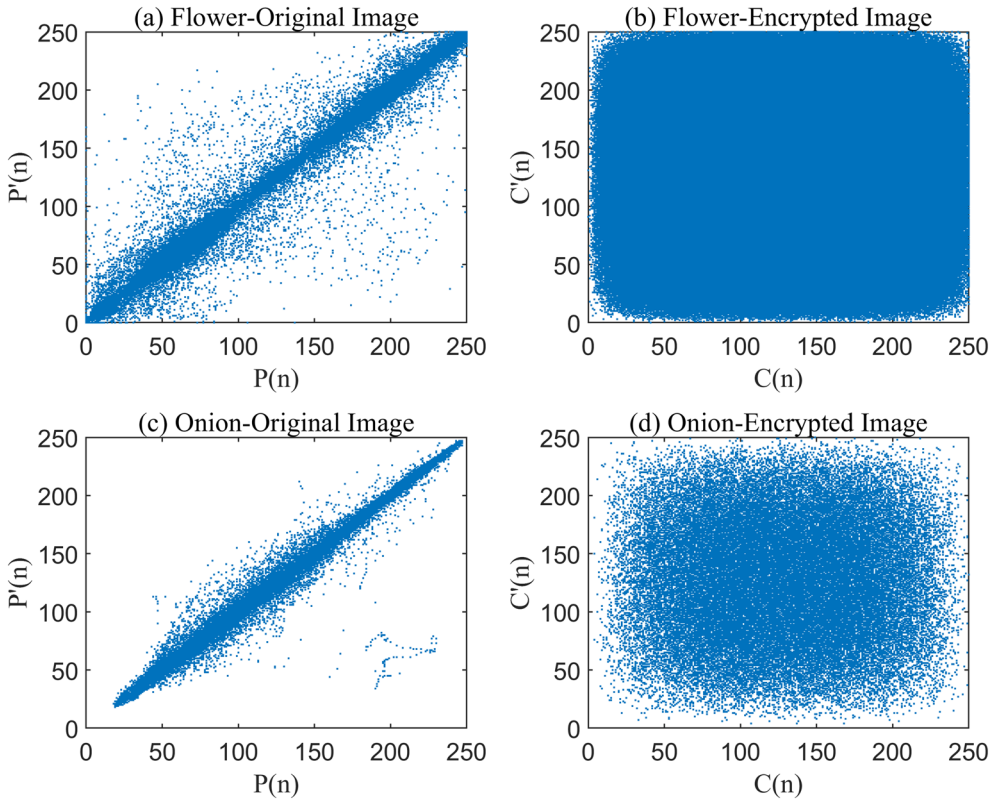

Fig. 14. The distributions of adjacent pixels in the original image and encrypted image.

No	Encryption scheme	Test images	Correlation coefficient values	
			Original image	Encrypted image
1	Proposed	Flower	0.9853	−0.0191
		Onion	0.9959	−0.0004
2	Ref. [28]	Peppers	0.9581	−0.0186
3	Ref. [43]	Lena	0.9226	0.0350
4	Ref. [44]	Baboon	0.8172	0.1093
5	Ref. [45]	Baboon	0.8172	−0.0163
6	Ref. [46]	Lena	0.9226	−0.0250
7	Ref. [47]	Lena	0.9226	0.0830

Table 10. Correlation coefficient values of different schemes.

spreads the influence of the current ciphertext as the encryption initial parameters onto the next set, continuously enhancing the confusion performance. The correlation diagrams of adjacent pixels in the original image and the encrypted signal are shown in Fig. 14.

Figure 14 illustrates the distribution of neighboring pixels in both the original and encrypted images. In the original image, pixels are clustered along a line, indicating a strong correlation between neighboring pixels. In contrast, the encrypted image displays a uniform pixel distribution across the entire range. This uniformity results from the disruption and alteration of the original image's structure during encryption, which effectively conceals patterns and correlations. As a result, statistical attacks are unable to extract meaningful information from the encrypted image, thereby enhancing the security and robustness of the proposed encryption scheme.

The purpose of image encryption is to disrupt the strong correlation between neighboring pixels in the original image. In this study, 256 × 256 randomly selected pixels were analyzed to compute the image correlations. Table 10 presents the calculated correlation coefficients before and after encryption. As shown in the table, the correlation between neighboring pixels is greatly diminished post-encryption, resulting in images with minimal detectable patterns or regularities.

This section evaluates the performance of an image encryption scheme by analyzing key aspects such as key space, histogram similarity, information entropy, resistance to statistical and differential attacks, key sensitivity,

and pixel correlation. The results demonstrate that the scheme effectively disrupts pixel distribution and reduces the correlation between neighboring pixels, exhibiting strong randomness and resistance to attacks, thereby ensuring the security and robustness of the encrypted images.

Conclusions

This paper introduces a novel 4D hyperchaotic system and comprehensively analyzes its fundamental dynamic behaviors, encompassing equilibrium point stability, Lyapunov exponent spectrum, bifurcation diagram, and chaotic attractors. The study reveals that the new hyperchaotic system features a single equilibrium point and can maintain two positive Lyapunov exponents across a wide parameter range, with dynamic behavior ranging hyperchaotic, chaotic, periodic, and quasi-periodic, by adjusting the parameter d. Experimental implementation on embedded hardware STM32 validates these dynamics, with oscilloscope displays illustrating phase trajectories, affirming the system's effectiveness.

Moreover, the paper integrates the designed hyperchaotic system with embedded hardware STM32 for image encryption. This encryption algorithm employs the hyperchaotic key sequence generated by the system, with each image pixel undergoing a bit-XOR operation with corresponding bits of the key sequence. Despite its simple implementation, the method demonstrates high algorithmic complexity. Experimental results show that all hyperchaotic key sequences pass the 15 randomness tests of the NIST SP800-22 standard, confirming their complete randomness. Key space analysis, histogram similarity analysis, information entropy analysis, statistical attack analysis, differential attack analysis, key sensitivity analysis, and correlation analysis further validates the effectiveness of the hyperchaotic key-based encryption algorithm for image encryption.

In future research, the aim is to leverage the highly random and complex nature of hyperchaotic key sequences, in conjunction with multiple embedded hardware STM32 units, to achieve secure communication for data encryption/decryption across various domains, such as audio encryption, video encryption, text encryption, IoT security, financial transaction security, and medical data protection.

Data availability

The data used to support the findings of this study are available from the corresponding author upon request.

Received: 10 May 2024; Accepted: 27 August 2024
Published online: 03 September 2024

References

1. Rössler, O. E. An equation for continuous chaos. *Phys. Lett. A* **57** 397–398. https://doi.org/10.1016/0375-9601(76)90101-8 (1976).
2. Parker, J. P., Ashtari, O. & Schneider, T. M. Predicting chaotic statistics with unsfig invariant tori. *Chaos Interdiscip. J. Nonlinear Sci.* **33**, 083111. https://doi.org/10.1063/5.0143689 (2023).
3. Yu, F. *et al.* Dynamic analysis and FPGA implementation of a new, simple 5D memristive hyperchaotic Sprott-C system. *Mathematics* **11**, 701. https://doi.org/10.3390/math11030701 (2023).
4. Liu, Y., Zhou, Y. & Guo, B. Hopf bifurcation, periodic solutions, and control of a new 4D hyperchaotic system. *Mathematics* **11**, 2699. https://doi.org/10.3390/math11122699 (2023).
5. Cui, N. & Li, J. A new 4D hyperchaotic system and its control. *AIMS Math.* **8**, 905–923. https://doi.org/10.3934/math.2023044 (2023).
6. Li, J. & Cui, N. Dynamical behavior and control of a new hyperchaotic Hamiltonian system. *AIMS Math.* **7**, 5117–5132. https://doi.org/10.3934/math.2022285 (2022).
7. Lin, L., Zhuang, Y., Xu, Z., Yang, D. & Wu, D. Encryption algorithm based on fractional order chaotic system combined with adaptive predefined time synchronization. *Front. Phys.* **11**, 1202871. https://doi.org/10.3389/fphy.2023.1202871 (2023).
8. Karawia, A. Cryptographic algorithm using newton-raphson method and general bischi-naimzadah duopoly system. *Entropy* **23**, 57. https://doi.org/10.3390/e23010057 (2021).
9. Chen, T. H. & Yang, C. H. Region of interest encryption based on novel 2D hyperchaotic signal and bagua coding algorithm. *IEEE Access* **10**, 82751–82765. https://doi.org/10.1109/ACCESS.2022.3190851 (2022).
10. Fu, S. M., Cheng, X. F. & Liu, J. Dynamics, circuit design, feedback control of a new hyperchaotic system and its application in audio encryption. *Sci. Rep.* **13**, 19385. https://doi.org/10.1038/s41598-023-46161-5 (2023).
11. Cao, H., Chu, R. & Cui, Y. Complex dynamical characteristics of the fractional-order cellular neural network and its DSP implementation. *Fractal Fract.* **7**, 633. https://doi.org/10.3390/fractalfract7080633 (2023).
12. Li, X., Mou, J., Banerjee, S., Wang, Z. & Cao, Y. Design and DSP implementation of a fractional-order detuned laser hyperchaotic circuit with applications in image encryption. *Chaos Solitons Fractals* **159** , 112133. https://doi.org/10.1016/j.chaos.2022.112133 (2022).
13. Jia, S. H., Li, Y. X., Shi, Q. Y. & Huang, X. Design and FPGA implementation of a memristor-based multi-scroll hyperchaotic system. *Chin. Phys. B* **31**, 070505. https://doi.org/10.1088/1674-1056/ac4a71 (2022).
14. Wang, Y. *et al.* FPGA-based implementation and synchronization design of a new five-dimensional hyperchaotic system. *Entropy* **24**, 1179. https://doi.org/10.3390/e24091179 (2022).
15. Babu, N. R., Kalpana, M. & Balasubramaniam, P. A novel audio encryption approach via finite-time synchronization of fractional order hyperchaotic system. *Multimed. Tools Appl.* **80**, 18043–18067. https://doi.org/10.1007/s11042-020-10288-8 (2021).
16. Vaidyanathan, S. *et al.* A new 4-D multi-stable hyperchaotic system with no balance point: Bifurcation analysis, circuit simulation, FPGA realization and image cryptosystem. *IEEE Access* **9**, 144555–144573. https://doi.org/10.1109/ACCESS.2021.3121428 (2021).
17. Hou, W., Li, S., He, J. & Ma, Y. A novel image-encryption scheme based on a non-linear cross-coupled hyperchaotic system with the dynamic correlation of plaintext pixels. *Entropy* **22**, 779. https://doi.org/10.3390/e22070779 (2020).
18. Wang, L., Chen, Z., Sun, X. & He, C. Color image ROI encryption algorithm based on a novel 4D hyperchaotic system. *Phys. Scr.* **99**, 015229. https://doi.org/10.1088/1402-4896/ad14d1 (2024).
19. Nguyen, Q. D., Pham, Q. D., Thanh, N. T. & Giap, V. N. An optimal homogenous stability-based disturbance observer and sliding mode control for secure communication system. *IEEE Access* **11**, 27317–27329. https://doi.org/10.1109/ACCESS.2023.3257854 (2023).
20. Rybin, V. *et al.* Prototyping the symmetry-based chaotic communication system using microcontroller unit. *Appl. Sci.* **13**, 936. https://doi.org/10.3390/app13020936 (2023).

21. Wang, P. *et al.* Secure transmission for IoT wireless energy-carrying communication systems. *PLOS ONE* **18**, e0289251. https://doi.org/10.1371/journal.pone.0289251 (2023).
22. Wang, M., Niu, Y., Gao, B. & Zou, Q. Hyperchaotic impulsive synchronization and digital secure communication. *J. Appl. Math. Phys.* **10**, 3485–3495. https://doi.org/10.4236/jamp.2022.1012230 (2022).
23. He, J., Qiu, W. & Cai, J. Synchronization of hyperchaotic systems based on intermittent control and its application in secure communication. *J. Adv. Comput. Intell. Intell. Inform.* **27**, 292–303. https://doi.org/10.20965/jaciii.2023.p0292 (2023).
24. Alexan, W., Chen, Y. L., Por, L. Y. & Gabr, M. Hyperchaotic maps and the single neuron model: A novel framework for chaos-based image encryption. *Symmetry* **15**, 1081. https://doi.org/10.3390/sym15051081 (2023).
25. Zhu, S., Deng, X., Zhang, W. & Zhu, C. Construction of a new 2D hyperchaotic map with application in efficient pseudo-random number generator design and color image encryption. *Mathematics* **11**, 3171. https://doi.org/10.3390/math11143171 (2023).
26. Sun, S. A new image encryption scheme based on 6D hyperchaotic system and random signal insertion. *IEEE Access* **11** , 66009–66016. https://doi.org/10.1109/ACCESS.2023.3290915 (2023).
27. Shen, Y. *et al.* Fast and secure image encryption algorithm with simultaneous shuffling and diffusion based on a time-delayed combinatorial hyperchaos map. *Entropy* **25**, 753. https://doi.org/10.3390/e25050753 (2023).
28. Gao, X., Sun, B., Cao, Y., Banerjee, S. & Mou, J. A color image encryption algorithm based on hyperchaotic map and DNA mutation. *Chin. Phys. B* **32** , 030501. https://doi.org/10.1088/1674-1056/ac8cdf (2023).
29. Sun, S. & Guo, Y. A new hyperchaotic image encryption algorithm based on stochastic signals. *IEEE Access* **9**, 144035–144045. https://doi.org/10.1109/ACCESS.2021.3121588 (2021).
30. Hou, W., Li, S., He, J. & Ma, Y. A novel image-encryption scheme based on a non-linear cross-coupled hyperchaotic system with the dynamic correlation of plaintext pixels. *Entropy* **22**, 779. https://doi.org/10.3390/e22070779 (2020).
31. Elsonbaty, A., Elsadany, A. A. & Adel, W. On reservoir computing approach for digital image encryption and forecasting of hyperchaotic finance model. *Fractal Fract.* **7**, 282. https://doi.org/10.3390/fractalfract7040282 (2023).
32. Du, Y., Long, G., Jiang, D., Chai, X. & Han, J. Optical image encryption algorithm based on a new four-dimensional memristive hyperchaotic system and compressed sensing. *Chin. Phys. B* https://doi.org/10.1088/1674-1056/acef08 (2023).
33. Chen, W., Wang, Y., Xiao, Y. & Hei, X. Explore the potential of deep learning and hyperchaotic map in the meaningful visual image encryption scheme. *IET Image Process.* **17**, 3235–3257. https://doi.org/10.1049/ipr2.12858 (2023).
34. Xu, H. & Wang, J. New 4D hyperchaotic system's application in image encryption. *J. Opt.* **26**, 065503. https://doi.org/10.1088/2040-8986/ad3e0d (2024).
35. Ding, L. & Ding, Q. The establishment and dynamic properties of a new 4D hyperchaotic system with its application and statistical tests in gray images. *Entropy* **22**, 310. https://doi.org/10.3390/e22030310 (2020).
36. Jiang, Q., Yu, S. & Wang, Q. Cryptanalysis of an image encryption algorithm based on two-dimensional hyperchaotic map. *Entropy* **25**, 395. https://doi.org/10.3390/e25030395 (2023).
37. Wen, H. *et al.* Secure DNA-coding image optical communication using non-degenerate hyperchaos and dynamic secret-key. *Mathematics* **10**, 3180. https://doi.org/10.3390/math10173180 (2022).
38. Liu, J., Cheng, X. & Zhou, P. Circuit implementation synchronization between two modified fractional-order Lorenz Chaotic systems via a linear resistor and fractional-order capacitor in parallel coupling. *Math. Probl. Eng.* **2021**, 1–8. https://doi.org/10.1155/2021/6771261 (2021).
39. Chen, G., Mao, Y. & Chui, C. K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **21**, 749–761. https://doi.org/10.1016/j.chaos.2003.12.022 (2004).
40. Luo, Y., Du, M. & Liu, J. A symmetrical image encryption scheme in wavelet and time domain. *Commun. Nonlinear Sci. Numer. Simul.* **20**, 447–460. https://doi.org/10.1016/j.cnsns.2014.05.022 (2015).
41. Wang, X.-Y., Zhang, Y.-Q. & Zhao, Y.-Y. A novel image encryption scheme based on 2-D logistic map and DNA sequence operations. *Nonlinear Dyn.* **82**, 1269–1280. https://doi.org/10.1007/s11071-015-2234-7 (2015).
42. Liu, W., Sun, K., He, Y. & Yu, M. Color image encryption using three-dimensional sine ICMIC modulation map and DNA sequence operations. *Int. J. Bifurc. Chaos* **27**, 1750171. https://doi.org/10.1142/S0218127417501711 (2017).
43. Kaur, G., Agarwal, R. & Patidar, V. Color image encryption scheme based on fractional Hartley transform and chaotic substitution–permutation. *Vis. Comput.* **38**, 1027–1050. https://doi.org/10.1007/s00371-021-02066-w (2022).
44. Kaur, G., Agarwal, R. & Patidar, V. Color image encryption system using combination of robust chaos and chaotic order fractional Hartley transformation. *J. King Saud Univ. Comput. Inf. Sci.* **34**, 5883–5897. https://doi.org/10.1016/j.jksuci.2021.03.007 (2022).
45. Zhang, D., Chen, L. & Li, T. Hyper-chaotic color image encryption based on transformed zigzag diffusion and RNA operation. *Entropy* **23**, 361. https://doi.org/10.3390/e23030361 (2021).
46. Chen, C., Sun, K. & Xu, Q. A color image encryption algorithm based on 2D-CIMM chaotic map. *China Commun.* **17**, 12–20. https://doi.org/10.23919/JCC.2020.05.002 (2020).
47. Farah, M. A. B., Guesmi, R., Kachouri, A. & Samet, M. A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation. *Opt. Laser Technol.* **121**, 105777. https://doi.org/10.1016/j.optlastec.2019.105777 (2020).

Acknowledgements

This work was supported by Science and Technology Project of Chongqing Municipal Education Commission (Grant No. KJZD-K202303301, KJQN202203309 and KJQN202203308).

Author contributions

Conceptualization, X.F.C.; methodology, X.F.C. and J.L.; software, X.F.C.; validation, H.M.Z., L.L. and J.L.; formal analysis, X.F.C.; investigation, K.P.M. and J.L.; resources, X.F.C. and J.L.; data curation, H.M.Z. and L.L.; writing—original draft preparation, X.F.C.; writing—review and editing, J.L.; supervision, X.F.C.; project administration, X.F.C.; funding acquisition, X.F.C. and H.M.Z. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.L.

Reprints and permissions information is available at www.nature.com/reprints.

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