

Remote sensing image encryption algorithm based on novel hyperchaos and an elliptic curve cryptosystem

Jing-Xi Tian(田婧希), Song-Chang Jin(金松昌), Xiao-Qiang Zhang(张晓强), Shao-Wu Yang(杨绍武), and Dian-Xi Shi(史殿习)

Citation: Chin. Phys. B, 2024, 33 (**5**): 050502. DOI: 10.1088/1674-1056/ad20d9
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

Characteristic analysis of 5D symmetric Hamiltonian conservative hyperchaotic system with hidden multiple stability

Li-Lian Huang(黄丽莲), Yan-Hao Ma(马衍昊), and Chuang Li(李创)
Chin. Phys. B, 2024, 33 (**1**): 010503. DOI: 10.1088/1674-1056/acf9e7

Lossless embedding: A visually meaningful image encryption algorithm based on hyperchaos and compressive sensing

Xing-Yuan Wang(王兴元), Xiao-Li Wang(王晓丽), Lin Teng(滕琳), Dong-Hua Jiang(蒋东华), and Yongjin Xian(咸永锦)
Chin. Phys. B, 2023, 32 (**2**): 020503. DOI: 10.1088/1674-1056/aca149

Memristor hyperchaos in a generalized Kolmogorov-type system with extreme multistability

Xiaodong Jiao(焦晓东), Mingfeng Yuan(袁明峰), Jin Tao(陶金), Hao Sun(孙昊), Qinglin Sun(孙青林), and Zengqiang Chen(陈增强)
Chin. Phys. B, 2023, 32 (**1**): 010507. DOI: 10.1088/1674-1056/ac5e95

Design and FPGA implementation of a memristor-based multi-scroll hyperchaotic system

Sheng-Hao Jia(贾生浩), Yu-Xia Li(李玉霞), Qing-Yu Shi(石擎宇), and Xia Huang(黄霞)
Chin. Phys. B, 2022, 31 (**7**): 070505. DOI: 10.1088/1674-1056/ac4a71

Neural-mechanism-driven image block encryption algorithm incorporating a hyperchaotic system and cloud model

Peng-Fei Fang(方鹏飞), Han Liu(刘涵), Cheng-Mao Wu(吴成茂), and Min Liu(刘旻)
Chin. Phys. B, 2022, 31 (**4**): 040501. DOI: 10.1088/1674-1056/ac272d

---

Remote sensing image encryption algorithm based on novel hyperchaos and an elliptic curve cryptosystem

Jing-Xi Tian(田婧希)[1], Song-Chang Jin(金松昌)[2], Xiao-Qiang Zhang(张晓强)[3], Shao-Wu Yang(杨绍武)[1], and Dian-Xi Shi(史殿习)[2,†]

[1] *School of Computer Science, National University of Defense Technology, Changsha* 410003, *China*
[2] *Intelligent Game and Decision Laboratory (IGDL), Beijing* 100091, *China*
[3] *School of Information and Control Engineering, China University of Mining and Technology, Xuzhou* 221116, *China*

(Received 9 November 2023; revised manuscript received 19 January 2024; accepted manuscript online 22 January 2024)

Remote sensing images carry crucial ground information, often involving the spatial distribution and spatiotemporal changes of surface elements. To safeguard this sensitive data, image encryption technology is essential. In this paper, a novel Fibonacci sine exponential map is designed, the hyperchaotic performance of which is particularly suitable for image encryption algorithms. An encryption algorithm tailored for handling the multi-band attributes of remote sensing images is proposed. The algorithm combines a three-dimensional synchronized scrambled diffusion operation with chaos to efficiently encrypt multiple images. Moreover, the keys are processed using an elliptic curve cryptosystem, eliminating the need for an additional channel to transmit the keys, thus enhancing security. Experimental results and algorithm analysis demonstrate that the algorithm offers strong security and high efficiency, making it suitable for remote sensing image encryption tasks.

Keywords: hyperchaotic system, elliptic curve cryptosystem (ECC), 3D synchronous scrambled diffusion, remote sensing image, unmanned aerial vehicle (UAV)

PACS: 05.45.–a, 05.45.Gg, 95.75.Mn **DOI:** 10.1088/1674-1056/ad20d9

1. Introduction

Remote sensing technology involves employing sensors on platforms like satellites, aircraft or ground stations to gather electromagnetic data and convert it into images or digital information. This technology is utilized for analyzing, monitoring and comprehending the features of Earth or other celestial bodies.[1] Since the 1920s, remote sensing has been employed in diverse military applications, including target detection, positioning and intelligence gathering.[2] With the advent of new data acquisition platforms, spectrum sensors can now be deployed on smaller platforms such as robots and unmanned aerial vehicles (UAVs). The remarkable mobility and cost-effectiveness of these new platforms make them an optimal choice.[3] This has led to the utilization of remote sensing images in various sectors, such as land management, mineral exploration, weather forecasting and air quality monitoring, thereby enhancing people's livelihoods. Remote sensing technology has evolved into a crucial tool for fostering both social and economic sustainability.[4] It is of paramount importance to guarantee the privacy of remote sensing data. Consequently, images containing sensitive information must be encrypted prior to transmission over a network. It is essential to note that most remote sensing images encompass multiple bands, such as multispectral or hyperspectral images. Therefore, designing encryption algorithms tailored for multi-band remote sensing images is particularly crucial.

Encryption algorithms can be classified into two categories: symmetric and asymmetric. Symmetric algorithms utilize a single key for both encryption and decryption. If an unauthorized party acquires the user's key, they can decrypt all data encrypted using that key. Therefore, the symmetric key must be transmitted over a secure channel.[5] Conversely, asymmetric encryption does not require additional key transmission. It employs public and private keys, eliminating the need for both communication parties to exchange keys in advance. Public keys encrypt messages, while private keys decrypt them. Based on the nearly unbreakable elliptic curve discrete logarithm problem,[6,7] the elliptic curve cryptosystem (ECC) achieves the same level of security as the Rivest–Shamir–Adleman system with shorter keys.[8] However, asymmetric algorithms are typically slower compared to symmetric algorithms.[9] Therefore, this paper only conducts asymmetric encryption processing on keys rather than plain images, achieving efficient and targeted encryption performance optimization.

The field of image encryption heavily leverages the advantages of chaos, including its high pseudo-randomness, extreme sensitivity to initial values and unpredictability.[10–12] There are numerous image encryption algorithms based on chaos. The primary focus of research is on the selection and

[†]Corresponding author. E-mail: dxshi@nudt.edu.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd

application of chaos, as its dynamic properties dictate the entire encryption process. While low-dimensional chaos is simpler to implement, it may not offer sufficient security. Conversely, many existing multidimensional systems have more parameters, a broader chaotic range and improved traversability. However, their intricate structure can lead to increased hardware implementation challenges.[13]

To enhance the applicability of chaos in image encryption algorithms, numerous scholars have innovatively modified and enhanced existing chaotic systems. For example, Erkan *et al.*[14] introduced a novel chaotic map based on Euler numbers and the Pi number. Through rigorous cryptanalysis, it was established that this novel chaos exhibits superior hyperchaotic characteristics in terms of ergodicity, complexity and randomness. Combined with the designed "bit reversal" diffusion operation, the image encryption algorithm is implemented. Huang *et al.*[15] utilized the sinusoidal cascade of two identical one-dimensional (1D) chaotic mappings to construct a chaotic system. The validity of the model was verified by combining three classical 1D chaotic systems. Gao *et al.*[16] developed a hyperchaotic map by employing a discrete resistance-coupled sinusoidal map and further enhancing the algorithm's complexity by introducing DNA mutation. Chen *et al.*[17] combined a cubic map with an improved cosine map to realize a selective regional medical image encryption algorithm. However, this algorithm lacks a significant advantage in terms of time efficiency, posing challenges in achieving real-time encryption. Wang *et al.*[18] designed an image encryption algorithm tailored for handling content of irregular sizes. This algorithm introduces a new LS chaos model by perturbing control parameters and introducing delay states based on sinusoidal and logistic maps. However, the paper overlooks the critical aspect of time efficiency, making it challenging to assess the algorithm's real-time performance.

In this paper, we propose a remote sensing image encryption algorithm based on a novel hyperchaos and ECC. This algorithm combines the multi-band characteristics of remote sensing images and designs a Fibonacci sine exponential map (FSEM) and a three-dimensional (3D) synchronous scrambled diffusion operation to realize efficient encryption. Moreover, the utilization of the ECC enables efficient and secure key management, thereby eliminating the need for an additional channel for key transmission.

The main contributions of this paper are as follows:

1) Design of a hyperchaotic system named FSEM, which utilizes Fibonacci's structural characteristics to establish a connection between the current output and the previous two instance inputs, combined with sine and exponential functions. The system exhibits hyperchaotic behavior and successfully passes relevant tests, making it highly suitable for image encryption.

2) Design of a 3D synchronous scrambling operation, which achieves synchronous changes in image pixel values and positions by combining chaotic sequences generated by the FSEM. This innovation not only improves the efficiency of image processing but also greatly enhances the confusion of the image.

3) By using the ECC to implement asymmetric key encryption, the ECC's nearly unbreakable security provides sufficient key protection. During transmission, we use a public key instead of a private key, thus effectively solving the problem of key management and transmission.

4) Experimental results confirm that this algorithm effectively safeguards remote sensing image information.

The remainder of this paper is structured as follows: In Section 2, the designed novel FSEM is introduced and a detailed analysis of its performance is provided. Additionally, the theoretical principles of the ECC and the specific 3D synchronous scrambled diffusion operations are introduced. Section 3 describes the operation of the ECC for asymmetric encryption of keys and presents a remote sensing image encryption algorithm based on a novel hyperchaos and ECC. Section 4 presents experimental results conducted using remote sensing images. Section 5 performs an algorithm analysis. Section 6 draws conclusions.

2. Theoretical principles

2.1. Designed FSEM

This paper introduces an innovative 2D hyperchaotic system named the FSEM. The chaotic system employs a crossover pattern, where the output of the Y sequence is linked to the input of the X sequence and vice versa. Furthermore, the map incorporates the recursive nature of the Fibonacci sequence, connecting the current output to the inputs from two prior moments. The chaos is further characterized by the combination of sinusoidal and exponential functions. The FSEM is defined by

$$\begin{cases} x_n = \sin\left(\mu \dfrac{\mathrm{e}^{x_{n-1}}}{x_{n-1}}(1-y_{n-2})\right), \\ y_n = \sin\left(\dfrac{\mathrm{e}^{y_{n-1}}}{y_{n-1}}(1-x_{n-2})\right), \end{cases} \tag{1}$$

where $x_n, y_n \in [-1,1]$ and the control parameter $\mu \in (3,\infty)$.

2.2. Performance analysis of the FSEM

The performance of the hyperchaotic system FSEM is evaluated across five dimensions: bifurcation diagram, Lyapunov exponent (LE), 0–1 test, sensitivity analysis and the

National Institute of Standards and Technology (NIST) test. In each metric, the FSEM demonstrates exceptional chaotic behavior.

2.2.1. Bifurcation diagram

The bifurcation diagram offers a qualitative description of the system's behavior as parameters vary.[19,20] Figure 1 illustrates the variation in the values of the chaotic sequences X and Y as the parameter μ is varied. As depicted in the figures, when μ exceeds 3, the sequence enters a chaotic state, leading to a dispersal of system output values across the entire plane. Consequently, the proposed FSEM exhibits remarkable chaotic characteristics.

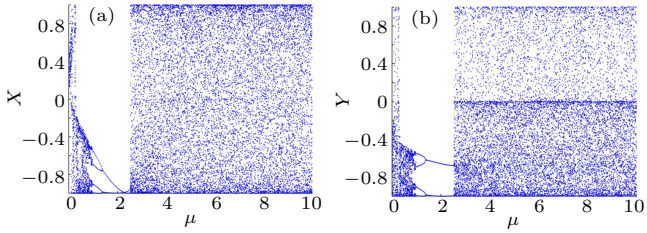


Fig. 1. Bifurcation diagrams of the FSEM: (a) X and (b) Y with $\mu \in [0,10]$, respectively.

2.2.2. LE

The LE serves as a metric for assessing the system's trajectory convergence or divergence rate.[21] A higher LE value indicates a more pronounced display of chaotic behavior. The LE value can be calculated by

$$\psi = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| \frac{\mathrm{d}f(x)}{\mathrm{d}x} \right|_{x=x_i}. \tag{2}$$

When $\psi > 0$, even if the initial conditions of the system are relatively stable, the output drastically changes throughout the course of repetitions, ultimately leading to chaotic behavior. For a multidimensional chaotic system, the presence of multiple LEs greater than 0 signifies a higher degree of complexity and stochastic hyperchaotic dynamics compared to low-dimensional chaotic systems. Figure 2(a) demonstrates that the FSEM has two LEs greater than 0, indicating that the FSEM is hyperchaotic.

To further evaluate the performance of the proposed map we compared it with the 2D-SLMM,[22] the 2D-LSCM[23] and the 2D-LASM.[24] It is evident from Figs. 2(b)–2(d) that the 2D-SLMM, the 2D-LSCM and the 2D-LASM exhibit discontinuities in certain periodic windows and have LEs that are smaller than FSEM. These findings suggest that, compared to other maps, the FSEM exhibits a broader coverage, more intricate hyperchaotic characteristics and faster divergence behavior. Therefore, the designed FSEM is well-suited for application in the encryption process to enhance algorithm security.

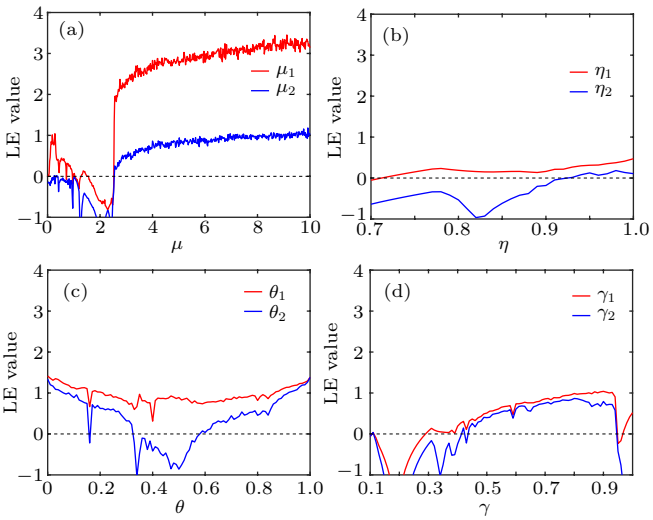


Fig. 2. LEs of the different 2D chaos: (a) the FSEM; (b) the 2D-SLMM; (c) the 2D-LSCM; (d) the 2D-LASM.

2.2.3. The 0–1 test

The 0–1 test assesses the presence or absence of chaos in the time series. Given a sequence $\phi(j)$, $j = 1,2,\ldots,N$, the result of the 0–1 test (denoted as K) can be computed by[25]

$$K = \mathrm{corr}(x, y) = \frac{\mathrm{cov}(x,y)}{\sqrt{\mathrm{var}(x)\mathrm{var}(y)}}, \tag{3}$$

where $x = 1,2,\ldots,n$, $y = M(1)$, $M(2)$, …, $M(n)$, $\mathrm{cov}(x,y)$ represents covariance, $\mathrm{var}(x)$ stands for variance and $M(n)$ denotes mean square displacement. A system can only be called a chaotic system when the test result of 0–1 is $K \approx 1$.

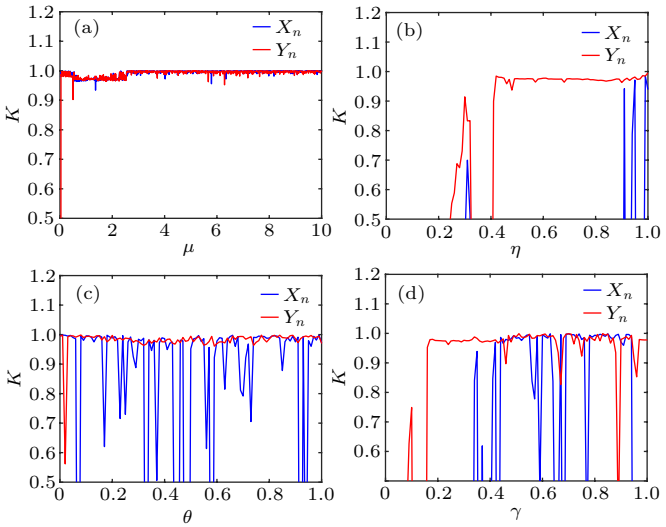


Fig. 3. The 0–1 test results of the different 2D chaos: (a) the FSEM; (b) the 2D-SLMM; (c) the 2D-LSCM; (d) the 2D-LASM.

In the simulation, set $N = 100$ and $r \in [\pi/5, 4\pi/5]$ to calculate the K value of the FSEM. The calculation results are shown in Fig. 3(a). The K value is close to 1 and remains stable in the initial stage. This indicates that the FSEM has a significantly wide chaotic interval. For the other three improved 2D chaotic maps, it can be seen from Figs. 3(b)–3(d)

that their fluctuation is large and cannot maintain the K value of 1 within the parameter range. In comparison, the designed FSEM possesses a significant chaotic interval and is highly suitable for image encryption.

2.2.4. Sensitivity analysis

The system's trajectory with a slight change in initial conditions is displayed in Fig. 4. The initial values are set to $\{x_0,\ x_1,\ y_0,\ y_1\}$ and $\{x_0+10^{-14},\ x_1,\ y_0,\ y_1\}$, and the corresponding four generated chaotic sequences are visualized in the figure. The designed FSEM exhibits a clear divergence after approximately the 10^{th} iteration and the four trajectories can be easily distinguished. This indicates that the designed hyperchaos exhibits strong sensitivity to initial values.

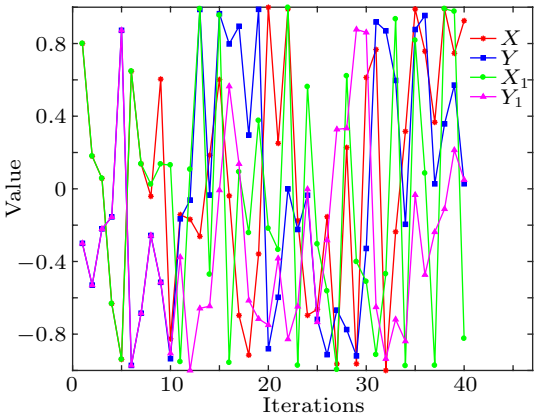


Fig. 4. Sensitivity analysis of the FSEM.

2.2.5. NIST

To further assess the chaotic performance of the designed FSEM, the NIST SP800-22 is utilized for validation. A P-value will be generated for each of the 15 subtests of the test. The sequence is deemed random only when all the P-values exceed 0.01. The test results in Table 1 indicate that both chaotic sequences generated by the FSEM have passed these subtests. Therefore, the designed FSEM is capable of generating excellent pseudo-random sequences.

Table 1. NIST test results of the FSEM.

Test index	P-value		Results
	X	Y	
Frequency	0.437274	0.834308	Passed
Block frequency	0.025193	0.834308	Passed
Cumulative sums	0.090936	0.437274	Passed
Runs	0.012650	0.012650	Passed
Longest run	0.437274	0.437274	Passed
Rank	0.012650	0.275709	Passed
FFT	0.437274	0.162606	Passed
Non-overlapping template	0.048716	0.048716	Passed
Overlapping template	0.637119	0.637119	Passed
Universal	0.162606	0.437274	Passed
Approximate entropy	0.025193	0.090936	Passed
Random excursions	0.350485	0.265901	Passed
Random excursions variant	0.122325	0.018141	Passed
Serial	0.637119	0.437274	Passed
Linear complexity	0.834308	0.437274	Passed

2.3. ECC

An elliptic curve, commonly denoted as E, is a mathematical construct defined within a prime field denoted as F_p. The equation that defines such an elliptic curve is expressed as follows:[8]

$$E: y^2 \equiv x^3 + ax + b (\bmod p), \tag{4}$$

where a and b are elements of the prime field F_p, and the prime number p must satisfy the condition that p is greater than 3 and $4a^3+27b^2 \neq 0 (\bmod p)$. The elliptic curve group, denoted as $E(F_p)$, comprises all points (x,y) that satisfy the equation of E. Additionally, this group includes a special point referred to as the point at infinity, denoted as O.

2.3.1. Point addition

In the domain of the ECC, a fundamental operation involves the addition of two points on an elliptic curve. Consider two points, $A=(x_1, y_1)$ and $B=(x_2, y_2)$, both residing on an elliptic curve denoted as E defined over a prime field. The objective is to perform the point addition operation, resulting in a third point, $R=(x_3, y_3)$. It is essential that the resulting point R lies on the curve E and adheres to its mathematical structure. This process of point addition plays a crucial role in the ECC, serving as the foundation for various cryptographic protocols and algorithms. To clarify, the computational procedure for finding point R is outlined below:

$$x_3 \equiv (\lambda^2 - x_1 - x_2) \bmod p, \tag{5}$$

$$y_3 \equiv (\lambda x_1 - \lambda x_3 - y_1) \bmod p, \tag{6}$$

$$\begin{cases} \lambda = \dfrac{3x_1^2 + a}{2y_1} \bmod p, A = B, \\ \lambda = \dfrac{y_2 - y_1}{x_2 - x_1} \bmod p, A \neq B. \end{cases} \tag{7}$$

2.3.2. Scalar point multiplication

Scalar point multiplication refers to multiplying a scalar value k by a point P, resulting in a point that still lies on the same elliptic curve E. This operation can be performed by repeatedly adding points. For instance, calculating $151P$ can be achieved through a series of point additions $151P = 2^7P + 2^4P + 2^2P + 2^1P + 2P$.

2.4. The 3D synchronous scrambled diffusion operation

Step 1 Resizing image dimension

Take a remote sensing image P with dimensions $M \times N \times K$ as input, as illustrated in Fig. 5. By connecting the beginning and end of each column element in P, and sequentially connecting K bands, a vector L of length MNK is obtained.

Step 2 Integerized chaotic sequences

Utilize the designed FSEM to generate chaotic sequences X and Y. These sequences can be integerized respectively using

$$C_1 = \bmod(\text{abs}(\text{floor}(X \times 10^8)), MNK), \tag{8}$$

$$C_2 = \bmod(\text{abs}(\text{floor}(Y \times 10^8)), 256), \tag{9}$$

then the scrambled matrix C_1 and the diffusion matrix C_2 can be obtained.

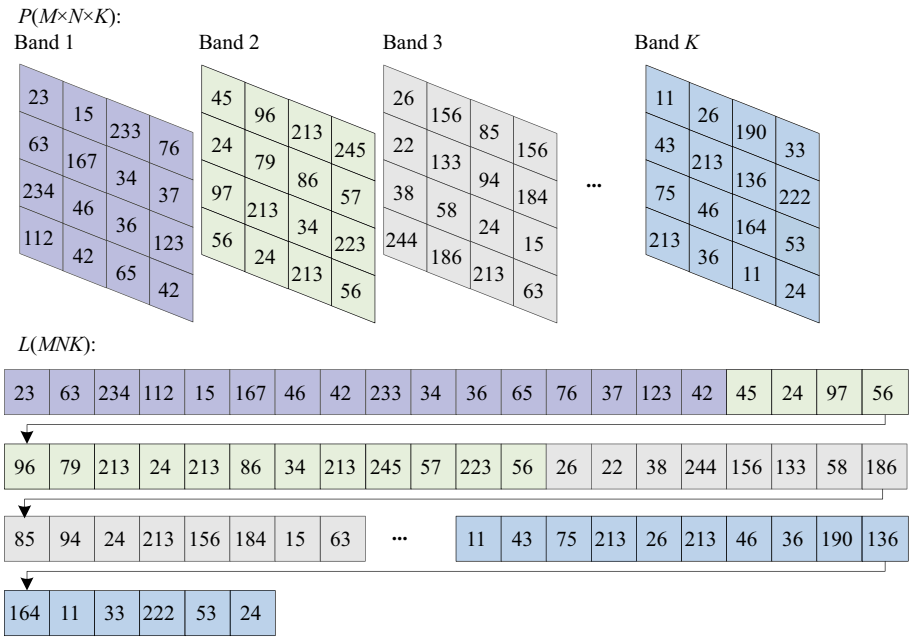


Fig. 5. Schematic diagram of converting P to the vector L.

Step 3 3D synchronous scrambled diffusion operation

Use the following equation to perform 3D synchronous scrambled diffusion operation:

$$H(i,j,k)=\begin{cases} L(C_1(i,j,k))\oplus C_2(i,j,k), & i=1,\ j=1,\\ L(C_1(i,j,k))\oplus C_2(i,j,k)\oplus H(i-1,j,k), & i\neq 1,\ j=1,\\ L(C_1(i,j,k))\oplus C_2(i,j,k)\oplus H(i,j-1,k), & i=1,\ j\neq 1,\\ L(C_1(i,j,k))\oplus C_2(i,j,k)\oplus H(i-1,j-1,k), & i\neq 1,\ j\neq 1, \end{cases} \tag{10}$$

where $i=1,2,\ldots,M$, $j=1,2,\ldots,N$, $k=1,2,\ldots,K$. The 3D synchronous scrambled diffusion operation enables the encrypted value $H(i,j,k)$ to be associated with $L(C_1(i,j,k))$, $C_2(i,j,k)$ and $H(i-1,j-1,k)$. C_1 is the scrambling matrix and the scramble is achieved through $L(C_1(i,j,k))$.

During the decryption process, the inverse operation of 3D synchronous scrambled diffusion can be performed on H using the following equation (where P^1 is the result of the inverse operation and its value is identical to L):

$$P^1(C_1(i,j,k))=\begin{cases} H(i,j,k)\oplus C_2(i,j,k), & i=1,\ j=1,\\ H(i,j,k)\oplus C_2(i,j,k)\oplus H(i-1,j,k), & i\neq 1,\ j=1,\\ H(i,j,k)\oplus C_2(i,j,k)\oplus H(i,j-1,k), & i=1,\ j\neq 1,\\ H(i,j,k)\oplus C_2(i,j,k)\oplus H(i-1,j-1,k), & i>1,\ j>1. \end{cases} \tag{11}$$

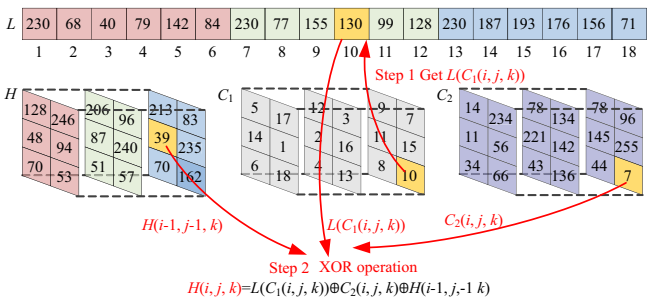


Fig. 6. 3D synchronous scrambled diffusion process with indicator arrows.

To provide a clearer representation of the data relationships, the steps for obtaining an encrypted value have been extracted and presented separately. As shown in Fig. 6, indicator arrows have been added to the diagram to illustrate these steps. The first step involves obtaining $L(C_1(i,j,k))$, and then performing XOR operations on $L(C_1(i,j,k))$, $C_2(i,j,k)$ and $H(i-1,j-1,k)$ to obtain the value of $H(i,j,k)$.

For example, Fig. 7 illustrates the 3D synchronous scrambled diffusion process for a vector L with a length of 18. The highlighted values represent the values that participate in the XOR operation, ultimately obtaining the encrypted matrix H. This process combines both scrambled and diffusion operations, simplifying the operation process and enhancing en-

cryption speed.

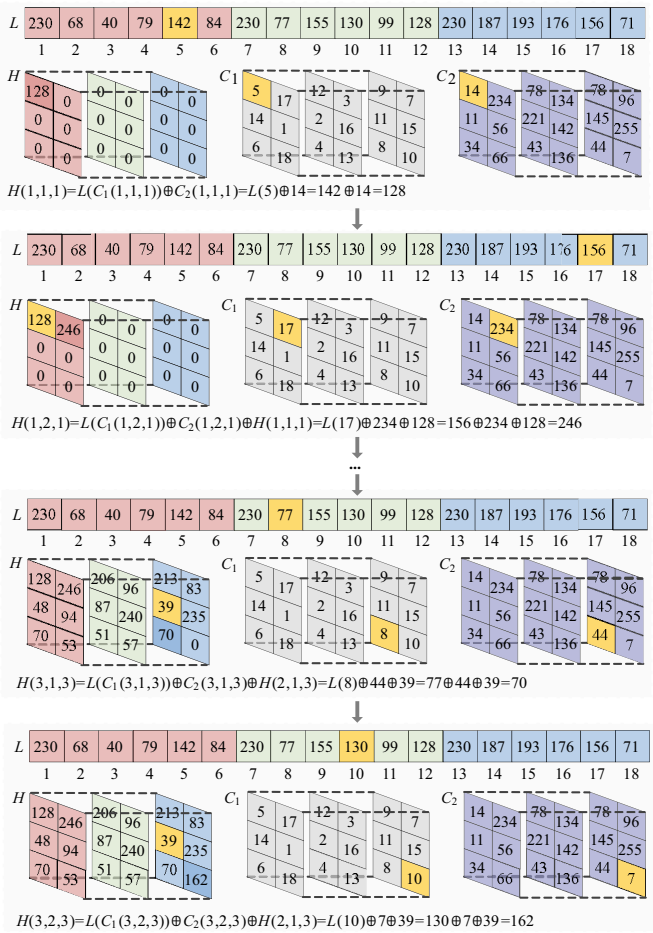


Fig. 7. The 3D synchronous scrambled diffusion process.

3. Proposed encryption algorithm

Establish a scenario with Alice as the sender and Bob as the recipient.

3.1. Key generation and processing

3.1.1. Key generation

To enhance the algorithm's resistance to plaintext attacks, the introduction of SHA-256 in the key generation process leverages its sensitivity to plaintext images. The SHA-256 hash value of the plain image, denoted as H_k, can be divided into

$$H_k = k_1, k_2, \ldots, k_{32}, \tag{12}$$

where $k_1, k_2, \ldots, k_{32}$ are 8-bit blocks. The keys of the designed FSEM can be set by

$$\begin{cases} x^1 = k_1 \oplus k_2 \oplus k_3 \oplus k_4/256, \\ y^1 = k_5 \oplus k_6 \oplus k_7 \oplus k_8/256, \\ x^2 = k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12}/256, \\ y^2 = k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16}/256, \\ \mu = \dfrac{1}{x_1} + \dfrac{1}{y_1} + \dfrac{1}{x_2} + \dfrac{1}{y_2}, \end{cases} \tag{13}$$

where $x^1, y^1, x^2, y^2 \in (0,1)$ and the control parameter $\mu > 4$.

3.1.2. Key processing

The ECC is an asymmetric encryption method that provides strong security, but it has a complex structure and substantial computational demands, making it less suitable for high-performance processing scenarios.[26] In this paper, only asymmetric encryption is performed on the key, so no additional key transmission is required. This approach solves key management and transmission issues, reducing the pressure on key storage. By reducing computational complexity, it can more effectively guarantee the security of encryption systems.

The schematic diagram of the ECC is depicted in Fig. 8, and the key processing steps are elaborated as follows.

Step 1 Setting up the elliptic curve

Bob selects a finite field F_p and an elliptic curve $E_p(a,b)$: $y^2 = x^3 + ax + b \bmod p$, where p is a large prime number. He then selects N on the E_p as the base point and denotes its order as T.

Step 2 Bob's key generation

Bob randomly selects an integer d from the range $[1, T-1]$ as his private key and calculates the corresponding public key $Q = dN$.

Step 3 Alice's encryption

Alice receives $E_p(a,b)$, Q, N and T. She randomly selects an integer u from the range $[1, T-1]$ as her private key. Subsequently, she calculates $C_1(x_1, y_1) = uN$, $C_2(x_2, y_2) = uQ + M$, where M contains the key information including x^1, y^1, x^2, y^2 and μ. Finally, Alice transmits the ciphertext $C(C_1, C_2)$ to Bob.

Step 4 Bob's decryption

Bob calculated $C_2 - dC_1$, and he can obtain the information M.

For easy understanding, specific data is selected as an example to demonstrate the entire ECC encryption key process.

Step 1 Establishment of the elliptic curve

Bob selects the elliptic curve $E_{211}(1,1)$: $y^2 = x^3 + x + 1 \bmod 211$, the base point $N(2,86)$ and its order T is 223.

Step 2 Bob's key generation

Bob selects $d = 86$ as his private key and calculates the corresponding public key $Q = dN = 86(2,86) = (130,153)$.

Step 3 Alice's encryption

The publicly available information includes $E_{211}(1,1)$, $N(2,86)$, $T = 223$ and Bob's public key $Q(130,153)$.

Alice selects $u = 180$ as her private key and calculates $C_1(x_1, y_1) = uN = 180(2,86) = (33,34)$, $C_2(x_2, y_2) = uQ + M = 180(130,153) + (x^1, y^1) = (24 + x^1, 61 + y^1)$.

Step 4 Bob's decryption

Bob calculates $C_2 - dC_1 = (24 + x^1, 61 + y^1) - 86(33,34) = (24 + x^1, 61 + y^1) - (24,61) = (x^1, y^1)$.

For the other keys x^2, y^2 and μ, the same operations are performed, which are omitted here. By performing these calculations, Bob can obtain all the key information.

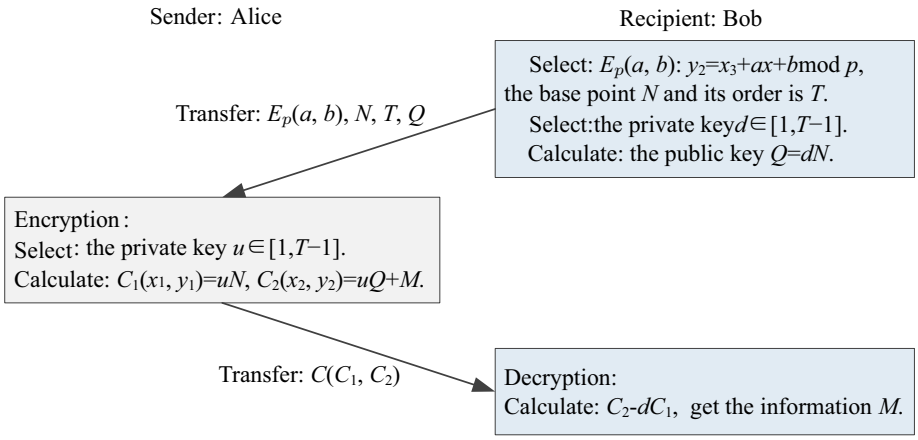


Fig. 8. ECC schematic diagram.

3.2. Encryption process

A flowchart illustrating the encryption and decryption processes is depicted in Fig. 9. The proposed algorithm consists of three stages: key and chaotic sequence generation, image pre-processing, and 3D synchronous scrambled diffusion operation. The keys involved in the algorithm are processed using the ECC and the generated public keys can be transmitted even over insecure channels. This enhancement significantly bolsters the algorithm's security performance. The detailed steps are outlined as follows.

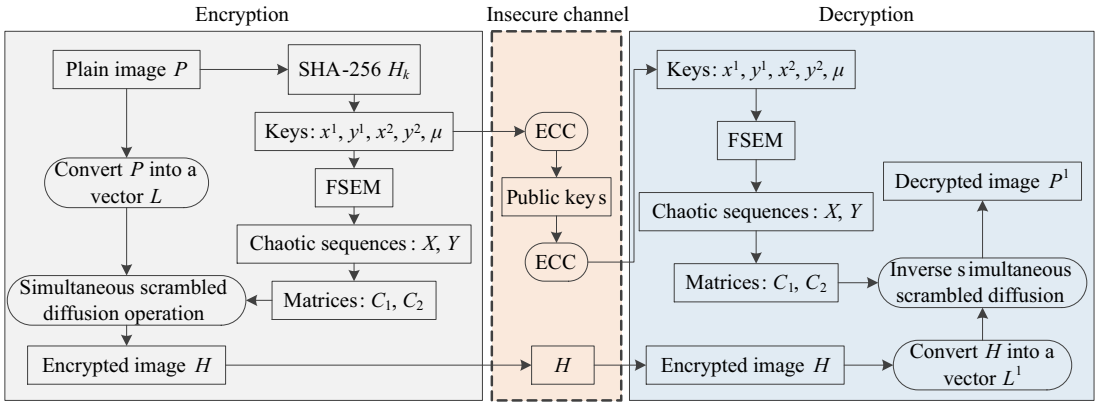


Fig. 9. Flowchart of the encryption and decryption process.

Step 1 Key and chaotic sequence generation

Utilizing the initial values x^1, y^1, x^2 and y^2 and the control parameter μ derived from Eqs. (12) and (13), Alice can obtain two chaotic sequences X and Y generated through the FSEM, whose sizes both are MNK.

Step 2 Image pre-processing

Alice inputs a remote sensing image P with dimensions $M\times N\times K$. By connecting the beginning and end of each column element in P and sequentially connecting K bands, a vector L with a length of MNK is obtained.

Step 3 3D synchronous scrambled diffusion operation

Using the specific operation detailed in Subsection 2.4, Alice employs the chaotic sequences X and Y to generate the scrambled matrix C_1 and the diffusion matrix C_2. Subsequently, Eq. (10) is utilized to achieve the 3D synchronous scrambled diffusion operation, resulting in the encrypted image H.

The pseudocode of the encryption algorithm is presented in Algorithm 1.

Algorithm 1: Encryption algorithm

Input: Plain image P with dimensions $M\times N\times K$
Output: Encrypted image H with dimensions $M\times N\times K$
$H_k \leftarrow \mathrm{SHA256}(P)$;
$k_1, k_2, \ldots, k_{32} \leftarrow H_k$;
$x^1 \leftarrow k_1 \oplus k_2 \oplus k_3 \oplus k_4/256$;
$y^1 \leftarrow k_5 \oplus k_6 \oplus k_7 \oplus k_8/256$;
$x^2 \leftarrow k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12}/256$;
$y^2 \leftarrow k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16}/256$;
$\mu \leftarrow 1/x_1+1/y_1+1/x_2+1/y_2$;
$X, Y \leftarrow$iterate the FSEM under x^1, y^1, x^2, y^2, μ;
for $j=1$: N
  for $i=1$: M
    for $k=1$: K
      $L(q)=P(i,j,k)$
    end
  end
end
$C_1 \leftarrow \mathrm{mod}(\mathrm{abs}(\mathrm{floor}(X\times 10^8)), MNK)$;
$C_2 \leftarrow \mathrm{mod}(\mathrm{abs}(\mathrm{floor}(Y\times 10^8)), 256)$;
$H \leftarrow$using Eq. (10) to realize the 3D synchronous scrambled diffusion operation.

3.3. Decryption process

Bob can utilize ECC encrypted public keys to acquire the decryption keys. Then he can further decrypt the encrypted image H to obtain the decrypted image. Image decryption is the reverse process of encryption. The detailed steps are outlined as follows.

Step 1 Chaotic sequence generation

Using the initial values x^1, y^1, x^2 and y^2 and the control parameter μ obtained through the ECC, Bob can obtain two chaotic sequences X and Y generated by the FSEM, whose sizes both are MNK.

Step 2 Image pre-processing

Bob inputs the encrypted image H with dimensions $M \times N \times K$. By connecting the beginning and end of each column element in H and sequentially connecting K bands, a vector L^1 with a length of MNK is obtained.

Step 3 3D synchronous scrambled diffusion operation

Utilizing the specific operation detailed in Subsection 2.4, Bob employs the chaotic sequences X and Y to generate the scrambled matrix C_1 and the diffusion matrix C_2. Subsequently, Eq. (11) is utilized to achieve the inverse operation of 3D synchronous scrambled diffusion, resulting in the decrypted image P^1.

The pseudocode of the image decryption algorithm is presented in Algorithm 2.

Algorithm 2: Decryption algorithm

Input: Encrypted image H with dimensions $M \times N \times K$

Output: Decrypted image P^1 with dimensions $M \times N \times K$

$x^1, y^1, x^2, y^2, \mu \leftarrow$ decrypted by ECC;

$X, Y \leftarrow$iterate the FSEM under x^1, y^1, x^2, y^2, μ;

for $j = 1: N$

  for $i = 1: M$

    for $k = 1: K$

      $L^1(q) = H(i, j, k)$

    end

  end

end

$C_1 \leftarrow \mathrm{mod}\,(\mathrm{abs}(\mathrm{floor}(X \times 10^8)), MNK)$;

$C_2 \leftarrow \mathrm{mod}\,(\mathrm{abs}(\mathrm{floor}(Y \times 10^8)), 256)$;

$P^1 \leftarrow$ using Eq. (11) to realize the inverse operation of 3D synchronous scrambled diffusion.

4. Experiments

Remote sensing images differ from ordinary images in terms of data volume, presentation information and post-production application. Most remote sensing images include more than one band. Similar to the relationship between grayscale and color images, a multi-band image can be decomposed into multiple single-band images.[27] A single band can be displayed in grayscale, while a multi-band image usually selects three bands based on research needs and uses red (R), green (G), and blue (B) channels to synthesize color maps of pixel values for visualization.[28] While the proposed encryption of remote sensing images is applicable to all bands, remote sensing images can only be displayed in up to three bands. Therefore, this paper uses synthetic color images to display the simulation results of remote sensing images. The test images were sourced from the USC-SIPI image database.

The algorithm was executed utilizing the Matlab R2016b software. The computational environment consisted of a computer running a 64-bit version of Windows 10, powered by a 1.20 GHz M-5Y71 processor and equipped with 8 GB of RAM. Figures 10(a)–10(c) show three test images, each with dimensions $512 \times 512 \times 3$, while Figs. 10(d)–10(f) display the corresponding encrypted images. It is evident that the encrypted images effectively conceal the information of the plain images. Figures 10(g)–10(i) are the decrypted images, which are indistinguishable from the plain images. Therefore, our algorithm is lossless and can effectively protect the detailed content in remote sensing images.

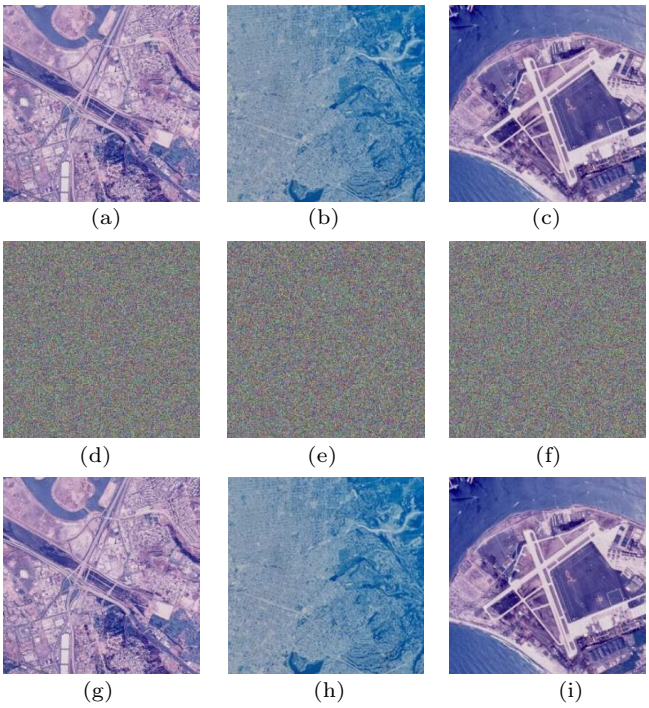


Fig. 10. Experiment results: (a)–(c) plain images of 2.1.02, 2.1.04 and 2.1.05; (d)–(f) cipher images of (a)–(c); (g)–(i) decrypted images of (d)–(f).

5. Algorithm analyses

The ability to withstand certain levels of attack during data transmission is a fundamental requirement for any dependable and reliable encryption system. Therefore, we conducted an extensive evaluation that encompassed multiple facets, including an examination of information entropy, the correlation between neighboring pixels, resistance against occlusion attacks and several other metrics. Meanwhile, our al-

gorithm was compared with several similar algorithms to assess its effectiveness.

5.1. Key space analysis

The key for our algorithm consists of a 256-bit hash referred to as H_k. The key space for our proposed algorithm reaches a total of 2^{256}, greatly exceeding the recommended 2^{100} required for an encryption system.[29] Consequently, the extensive size of our algorithm's key space provides robust safeguards against brute-force attacks due to its immense magnitude.

5.2. Histogram analysis

The histogram serves as a visual representation of the statistical attributes of an image, playing a crucial role in ensuring security standards for encrypted images. To meet these standards, the histogram of an encrypted image must exhibit complete uniformity, preventing attackers from extracting valuable data.[20] There are histograms for both the plain and encrypted images in Fig 11. It is apparent that the histogram of the encrypted image exhibits greater uniformity and dispersion compared to the plain image. Therefore, the proposed algorithm demonstrates superior resistance to statistical attacks.

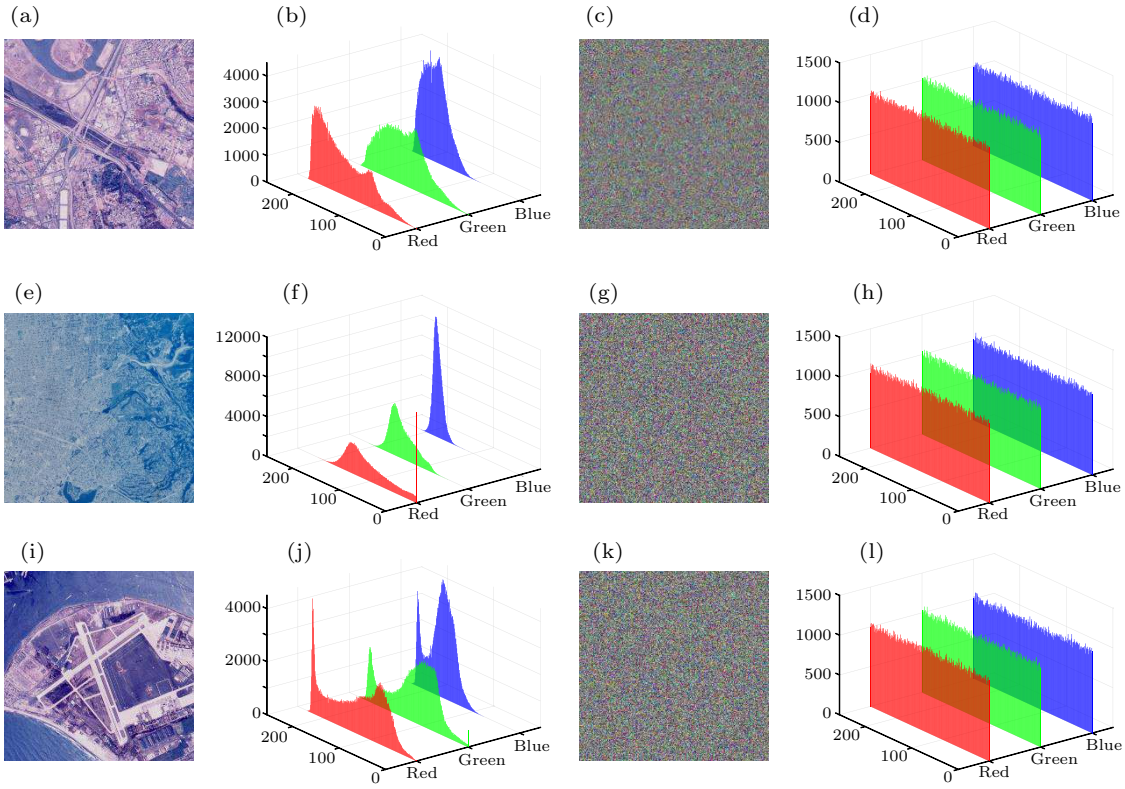


Fig. 11. Histograms of plain and cipher images: (a), (e) and (i) are the plain images; (c), (g) and (k) are the cipher images of (a), (e) and (i); (b), (f), (j) and (d), (h), (l) are the histograms of (a), (e), (i) and (c), (g) and (k), respectively.

5.3. Chi-squared test

The chi-square test assesses the homogeneity of the image histogram by comparing the observed occurrence frequency with the expected occurrence frequency for each gray level.[30] The chi-square is calculated as

$$\chi^2_{\text{test}} = \sum_{i=0}^{255} \frac{(F_i - G)^2}{G}, \tag{14}$$

where G and F_i represent the theoretical and observed frequency distributions, respectively, and $G = MN/256$. Table 2 lists the chi-square values obtained for the encrypted image, confirming its homogeneity. The crucial number for 5% probability with 255 degrees of freedom is 293.2478,[20] while our algorithm is significantly lower. Therefore, statistical attacks are ineffective against our algorithm.

Table 2. The χ^2 test of the pixel histogram.

Image	Component	χ^2_{cipher}
2.1.02	R	254.5781
	G	237.3887
	B	251.2305
2.1.04	R	284.0098
	G	227.5215
	B	234.5000
2.1.05	R	239.2012
	G	244.5762
	B	246.2422

5.4. Correlation of adjacent pixels

To effectively resist statistical attacks, it is ideal for the encrypted image to exhibit a very low or near-zero correlation coefficient.[31] The correlation coefficient between adja-

cent pixels is calculated as follows:

$$r_{x,y} = \frac{E((x-E(x))(y-E(y)))}{\sqrt{D(x)D(y)}}, \tag{15}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{16}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2, \tag{17}$$

where N represents the total number of selected image pixels, x and y denote the pixel values of adjacent pixels. Additionally, $E(\cdot)$ and $D(\cdot)$ stand for the expectation and variance functions, respectively. The computational results are listed in Table 3, which clearly demonstrates that the correlation coefficients in the horizontal (H), vertical (V) and diagonal (D) directions are close to 1 in the plain image, indicating a relatively strong correlation among pixels. Conversely, the correlation coefficient of the encrypted image is close to 0, indicating a significantly reduced correlation.

Table 3. Correlation coefficients of plain and cipher images.

Image	Components	Plain			Cipher		
		H	V	D	H	V	D
2.1.02	R	0.8478	0.8339	0.7682	0.0006	–0.0050	–0.0023
	G	0.7837	0.7634	0.6819	–0.0015	–0.0035	–0.0150
	B	0.7477	0.7253	0.6423	0.0044	–0.000003	–0.0103
2.1.04	R	0.8422	0.8554	0.7825	0.0029	0.0002	–0.0047
	G	0.7550	0.7798	0.6780	0.0009	–0.0060	–0.0120
	B	0.6760	0.7086	0.5845	–0.0004	0.0029	–0.0084
2.1.05	R	0.9577	0.9594	0.9372	–0.0011	–0.0025	–0.0017
	G	0.9368	0.9411	0.9126	0.0028	0.0036	0.0029
	B	0.9221	0.9278	0.8966	0.0035	0.0003	0.0006

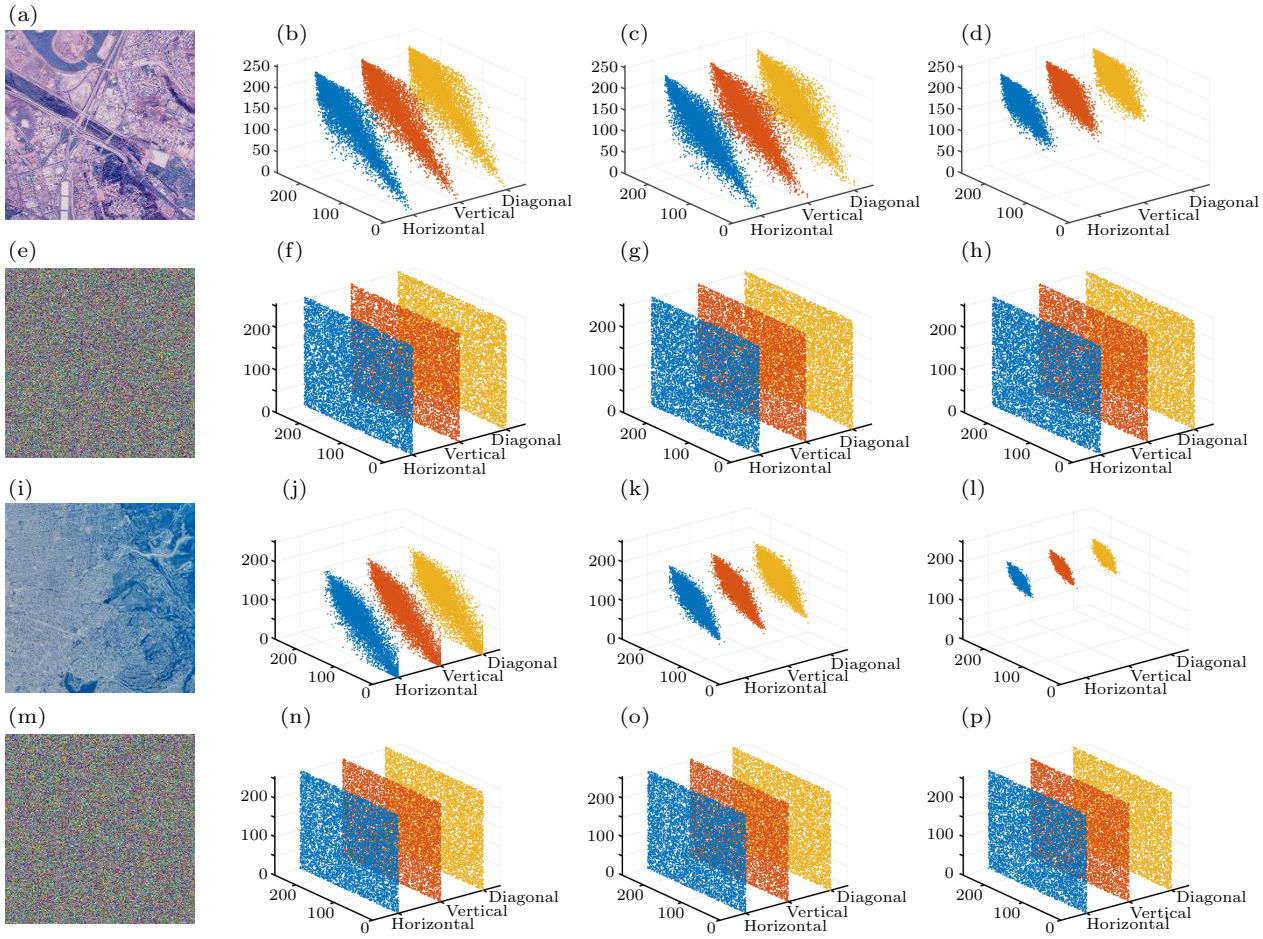


Fig. 12. Correlation analysis of plain and cipher images: (a) and (i) are the plain images; (e) and (m) are the cipher images of (a) and (i); (b)–(d), (j)–(l), (f)–(h) and (n)–(p) are the correlation distributions of the R, G, B components in three directions for (a), (e), (i) and (m), respectively.

Table 4 provides a comparative analysis of our algorithm's data in comparison to other algorithms. It is evident that our algorithm aligns more closely with the ideal value of 0. Figure 12 illustrates the correlation analysis of plain and cipher images. The correlation among the plain images exhibits a closely linear distribution, while the correlation among

the cipher images is randomly distributed and spans the entire range. Therefore, the proposed algorithm effectively achieves significant image scrambling and protection.

Table 4. Correlation coefficients of cipher images with different algorithms.

Algorithm	H	V	D
Proposed	**0.0012**	**–0.0028**	**–0.0092**
Ref. [20]	0.0001	0.0019	0.0124
Ref. [32]	0.0030	0.0023	–0.0116
Ref. [34]	–0.0042	0.0079	–0.0361
Ref. [36]	–0.0039	0.0031	–0.0021
Ref. [37]	0.0018	0.0004	0.0014
Ref. [38]	0.0031	0.0036	0.0008
Ref. [39]	–0.0017	0.0027	0.0002

5.5. Information entropy analysis

The randomness of an 8-bit image is quantitatively measured using information entropy,[40] as defined by

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

where m represents the ith gray level and $P(\cdot)$ denotes the probability. The information entropy values for both plain and cipher images are summarized in Table 5. The ideal value of the information entropy for cipher images is 8, and our algorithm's test value is very close to this value. Table 6 presents the comparative outcomes between our proposed algorithm and other algorithms, highlighting the superior performance of our algorithm relative to most others. Therefore, encrypted images can effectively conceal plain image data while maintaining the security of the transmission process.

Table 5. Information entropy of the plain and cipher images.

Image	Component	Plain image	Cipher image
2.1.02	R	7.4061	7.9993
	G	7.4188	7.9993
	B	6.5931	7.9993
	Average	**7.1393**	**7.9993**
2.1.04	R	7.2622	7.9992
	G	6.6163	7.9994
	B	5.2737	7.9994
	Average	**6.3841**	**7.9993**
2.1.05	R	7.5580	7.9993
	G	7.4597	7.9993
	B	6.6665	7.9993
	Average	**7.2281**	**7.9993**

Table 6. Information entropy of cipher images using different algorithms.

Algorithm	Information entropy
Proposed	**7.9993**
Ref. [20]	7.9972
Ref. [34]	7.9973
Ref. [35]	7.9993
Ref. [36]	7.9968
Ref. [38]	7.9975
Ref. [39]	7.9967
Ref. [41]	7.9993

5.6. Occlusion attack analysis

When transmitting images, data loss is inevitable. Robustness refers to that the majority of a normal image's useful information can still be recovered after an attack.[42] By varying the levels of image chopping on the encrypted images, we simulate data loss during transmission. The simulation output for the occlusion attack is shown in Fig. 13. The primary content of the plain images remains discernible, despite the blurring of the decoded image. Therefore, the proposed algorithm has excellent robustness and ensures the security and integrity of the data.

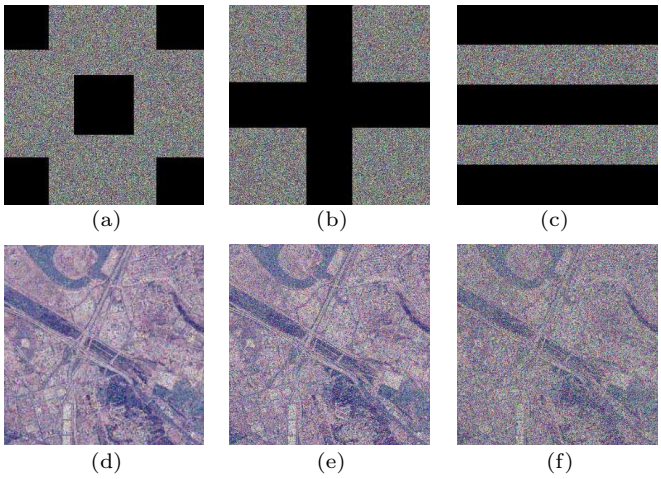


Fig. 13. Occlusion attack: (a)–(d) are the cipher images after cropping the size to 30%, 40%, and 60%, respectively; (d)–(f) are the decrypted images of (a)–(c).

5.7. Noise attack analysis

During the transmission process, noise is unavoidably introduced, which can impact the quality of encrypted images. Therefore, an essential metric for evaluating an image encryption algorithm's performance is its tolerance to noise contamination.[44] We introduced varying levels of pepper and salt and Gaussian noise to the encrypted images. The resulting

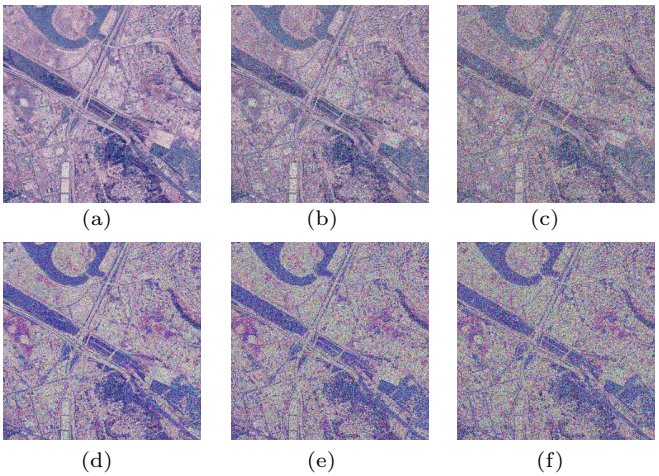


Fig. 14. Noise attack: (a)–(c) are the decrypted images under pepper and salt noise at densities of 0.1, 0.2 and 0.3, respectively; (d)–(f) are the decrypted images under Gaussian noise at densities of 0.01, 0.02 and 0.03, respectively.

decrypted images are depicted in Fig. 14. Despite the noise interference in the encrypted images, it is evident that the decrypted images remain recognizable. Therefore, the proposed algorithm exhibits robustness against noise attacks.

5.8. Key sensitivity analysis

An exceptional algorithm should ensure that even minimal alterations to the decryption key prevent the receiver from accurately decrypting the data.[45] In our algorithm, the keys x^1, y^1, x^2, y^2 and μ are generated using SHA-256. To test its sensitivity, we introduced a slight change of 10^{-14} to each key during decryption. The resulting decrypted images, as depicted in Fig. 15, do not disclose any meaningful information. This demonstrates the proposed algorithm's extreme sensitivity to the encryption key.

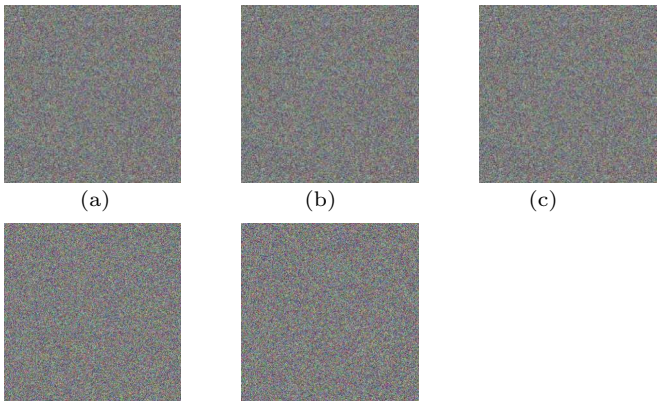


Fig. 15. Key sensitivity: (a)–(e) are decrypted images obtained with decryption keys of x^1+10^{-14}, y^1+10^{-14}, x^2+10^{-14}, y^2+10^{-14} and $\mu+10^{-14}$, respectively.

5.9. Differential attack analysis

The number of pixel change rate (NPCR) and unified average changing intensity (UACI) can test the algorithm's resistance to differential assaults.[46] The NPCR and UACI are defined by

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100\%, \tag{19}$$

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

where $C_1(i,\ j)$ and $C_2(i,\ j)$ represent the encrypted images of two nearly identical images differing by just one pixel, and $D(i,\ j)$ is defined by

$$D(i,j) = \begin{cases} 0, & C_1(i,j) = C_2(i,j), \\ 1, & \text{otherwise}. \end{cases} \tag{21}$$

The optimal benchmarks for NPCR and UACI are 99.6094% and 33.4635%, respectively. To obtain the associated encrypted image, we alter the one-pixel value of the plain image. The result is computed using Eqs. (19)–(21), which are presented in Table 7. A comparative analysis with other algorithms is outlined in Table 8. It is evident that our proposed algorithm aligns more closely with the ideal values than other algorithms. Therefore, our proposed algorithm exhibits resistance to differential attacks.

Table 7. NPCR and UACI values with the different images.

Image	Component	NPCR (%)	UACI (%)
2.1.02	R	99.6063	33.5488
	G	99.6090	33.4000
	B	99.6140	33.4518
	Average	**99.6098**	**33.4669**
2.1.04	R	99.6075	33.4537
	G	99.6216	33.5152
	B	99.6235	33.4299
	Average	**99.6175**	**33.4663**
2.1.05	R	99.6239	33.4936
	G	99.5811	33.4757
	B	99.5930	33.4633
	Average	**99.5993**	**33.4775**

Table 8. NPCR and UACI values with different algorithms.

Algorithm	NPCR (%)	UACI (%)
Proposed	**99.6098**	**33.4669**
Ref. [20]	99.6051	33.5033
Ref. [34]	99.6130	33.5340
Ref. [36]	99.6037	33.5281
Ref. [39]	99.6287	33.4512
Ref. [41]	99.6073	33.4589
Ref. [43]	99.6100	33.4700

5.10. Computational complexity analysis

The proposed algorithm is employed to process K plain images, each with dimensions $M \times N$. By utilizing the designed FSEM to generate two chaotic sequences, each with a length of MNK, this algorithm exhibits linear computational complexity, denoted as $O(MNK)$. The processing time depends on both the size and the number of plain images.

The running time is a crucial metric for evaluating the performance of image algorithms.[38] The time consumption is summarized in Table 9. Our proposed algorithm exhibits substantial advantages compared to other algorithms in terms of handling image size and execution time. Therefore, our algorithm proves to be both feasible and highly efficient.

Table 9. Time consumption.

Algorithm	Image	Image size	Encryption time (s)	Decryption time (s)
Proposed	2.1.02	512×512×3	**0.2938**	**0.2018**
	2.1.04	512×512×3	**0.2934**	**0.2216**
	2.1.05	512×512×3	**0.2882**	**0.2085**
Ref. [20]	Lena	512×512	2.8263	2.3943
Ref. [32]	–	512×512	0.3660	2.7478
Ref. [33]	Baboon	512×512×3	0.6564	0.6625
Ref. [34]	–	256×256	2.3600	3.6600
Ref. [35]	–	512×512	0.4100	0.4400
Ref. [36]	Lena	512×512×3	12.1166	12.1315
Ref. [47]	–	512×512	0.3210	0.3120

6. Conclusions

To secure remote sensing images during transmission, this paper proposes a novel encryption algorithm based on a novel hyperchaos and ECC. Firstly, a novel FSEM is designed, which exhibits remarkable hyperchaos behavior that is well-suited for image encryption algorithms. Secondly, the algorithm combines the multi-band characteristics of remote sensing images and designs a 3D synchronous scrambled diffusion operation to efficiently realize image encryption. Furthermore, to alleviate the burden of key storage and transmission, an asymmetric ECC is used for key processing, eliminating the need for an additional channel to transmit the key. This enhancement significantly improves security. The experimental results and algorithm analysis demonstrate the efficiency and security of the algorithm, showcasing its capability to withstand the most common types of attacks. Therefore, our algorithm is capable of remote sensing image encryption tasks and can serve as an excellent candidate for practical applications such as military and environmental detection.

Data availability statement

The datasets generated and/or analyzed during the current study are not publicly available, but are available from the corresponding author on reasonable request.

Acknowledgement

Project supported was supported by the National Natural Science Foundation of China (Grant No. 91948303).

References

[1] Zhang X Q and Wang X S 2018 *Appl. Sci.* **8** 1540
[2] Liu Z F, Li J Q, Di X Q, Man Z L and Sheng Y H 2021 *Secur. Commun. Netw.* **2021** 1
[3] Wang X M, Li J and Yan H Y 2021 *Enterp. Inf. Syst.* **15** 530
[4] Zhang D H, Ren L J, Shafiq M and Gu Z Q 2022 *Remote Sens.* **14** 6371
[5] Tao Y C, Yang X L, Meng X F, Wang Y R, Yin Y K and Dong G Y 2020 *Journal of Optics* **22** 65701
[6] Lone N P, Singh D, Stoffová V, Mishra D C, Mir U H and Kumar N 2022 *Mathematics* **10** 3878
[7] Khoirom M S, Laiphrakpam D S and Themrichon T 2018 *Optik* **168** 370
[8] Toughi S, Fathi M H and Sekhavat Y A 2017 *Signal Process.* **141** 217
[9] Chowdhary C L, Patel P V, Kathrotia K J, Attique M, Perumal K and Ijaz M F 2020 *Sensors* **20** 5162
[10] Lai Q, Hu G, Erkan U and Toktas 2022 *Appl. Math. Comput.* **442** 127738
[11] Wang M C, Liu H J and Zhao M D 2022 *Eur. Phys. J. Spec. Top.* **231** 3225
[12] Lv W R, Chen J X, Chai X L and Fu C 2023 *Nonlinear Dyn.* **111** 4
[13] Wen J, Xu X M, Sun K H, Jiang Z H and Wang X 2023 *Nonlinear Dyn.* **111** 3887
[14] Erkan U, Toktas A, Toktas F and Alenezi F 2021 *Inf. Sci.* **589** 770
[15] Huang L L, Chai B, Xiang J H, Zhang Z F and Liu J 2023 *Phys. Scr.* **98** 35217
[16] Gao X Y, Sun B, Cao Y H, Banerjee S and Mou J 2023 *Chin. Phys. B* **32** 30501
[17] Chen R, Li X M, Teng L and Wang X Y 2023 *Phys. Scr.* **98** 3
[18] Wang J, Liu L F, Xu M F and Li X J 2022 *J. King Saud. Univ. Comput. Inf. Sci.* **34** 8425
[19] Lai Q and Liu Y 2023 *Expert Syst. Appl.* **223** 119923
[20] Liang Q and Zhu C 2023 *Opt. Laser Technol.* **160** 109033
[21] Briggs K 1990 *Phys. Lett. A* **151** 27
[22] Hua Z Y, Zhou Y C, Pun C M, Chen C.L. P 2015 *Inf. Sci.* **297** 80
[23] Hua Z Y, Jin F, Xu B X, Huang H J 2018 *Signal Process.* **149** 148
[24] Hua Z Y and Zhou Y C 2016 *Inf. Sci.* **339** 237
[25] Gottwald G A and Melbourne I 2009 *SIAM J. Appl. Dyn. Syst.* **8** 129
[26] Wu Y, Zhang L L, Berretti S and Wan S H 2023 *IEEE Trans. Ind. Inform.* **19** 2089
[27] Jiang L, Niu T, Xu Z Q and Xu Y Y 2015 *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **8** 2232
[28] Wang X Y, Liu L L and Song M P 2023 *Nonlinear Dyn.* **111** 14537
[29] Alvarez G and Li S J 2006 *Int. J. Bifurcat. Chaos* **16** 2129
[30] Wu W Q and Wang Q 2023 *Nonlinear Dyn.* **111** 3831
[31] Lai Q, Yang L and Chen G R 2023 *IEEE Trans. Ind. Electron.* **8** 1
[32] Wang X, Liu C and Jiang D 2022 *Inf. Sci.* **610** 300
[33] Zhang X Q, Liu Z W and Yang X C 2023 *Nonlinear Dyn.* **111** 6839
[34] Rani N, Mishra V and Sharma S R 2023 *Nonlinear Dyn.* **111** 2869
[35] Wang X Y and Liu H P 2022 *Chaos, Solitons & Fractals* **164** 112586
[36] Yu J W, Xie W, Zhong Z Y and Wang H 2022 *Chaos, Solitons & Fractals* **162** 112456
[37] Liu P B, Wang X Y and Su Y N 2022 *IEEE T. Circ. Syst. Vid.* **33** 2506
[38] Lone M A and Qureshi S 2023 *Nonlinear Dyn.* **111** 5919
[39] Gao X Y, Mou J, Banerjee S and Zhang Y S 2023 *IEEE T. Cybernetics* **53** 5037
[40] Lin J 1991 *IEEE Trans. Inf. Theory.* **37** 145
[41] Zhao H G, Wang S S and Wang X Y 2022 *Chaos, Solitons & Fractals* **164** 112742
[42] Lai Q, Wan Z Q, Zhang H and Chen G R 2023 *IEEE Trans. Neural Netw. Learn Syst.* **34** 7824
[43] Li D H, Li J Q, Di X Q and Li B 2023 *Nonlinear Dyn.* **111** 2917
[44] Lai Q and Chen Z J 2023 *Chaos, Solitons & Fractals.* **170** 113341
[45] Lin H R, Wang C H, Xu C, Zhang X and Iu H H C 2023 *IEEE T. Comput. Aid. D.* **42** 942
[46] Wu Y and Noonan J P and Agaian S 2011 *Journal of Selected Areas in Telecommunications* **1** 31
[47] Gao X Y, Mou J, Xiong L, Sha Y W, Yan H Z and Cao Y H 2022 *Nonlinear Dyn.* **108** 613