Computers and Electrical Engineering 118 (2024) 109354

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Data security in the Industrial Internet of Things (IIoT) through a triple-image encryption framework leveraging 3-D NEAT, 1DCJ, and 4DHCFO techniques

Abid Mehmood [a], Arslan Shafique [b,*], Neeraj Kumar [c], Muhammad Nasir Bhutta [a]

[a] Department of Computer Science, Abu Dhabi University, United Arab Emirates
[b] School of Electronic and Nanoscale Engineering, University of Glasgow, Glasgow, G12 8QQ, UK
[c] Department of CSE, Thapar Institute of Engineering and Technology (Deemed University), Patiala, Punjab, India

ARTICLE INFO

Keywords:
Triple grayscale images
Secure data transmission
Bit planes
Internet of things

ABSTRACT

In the Industrial Internet of Things (IIoT) era, protecting vast data volumes, including sensitive information, poses a significant security challenge. To address this issue, this research proposes a novel triple-image encryption method tailored for IIoT applications. Unlike conventional algorithms processing a single grayscale image to produce a corresponding single ciphertext, the proposed approach generates a single color encrypted image corresponding to three grayscale input images. This complexity adds an extra layer of challenge for unauthorized individuals attempting to recover plaintext data. Leveraging the 3-D non-equilateral Arnold transform (NEAT), extended one-dimensional chaotic jumping (1DCJ), and a four-dimensional hyperchaotic Chen map of fractional order (4DHCFO), the proposed method begins by processing three grayscale images—R_{gray}, G_{gray}, and B_{gray}—with a 3-D NEAT to scramble their pixel positions. Employing three distinct scrambling operations, multilayer permutation, multiround permutation, and diagonal permutation, enhances scrambling complexity. Subsequently, binary bit planes are extracted and subjected to bit-level scrambling via 1DCJ. Further, a 4DHCFO generates a 16×16 substitution box for diffusing scrambled bit planes using XOR operations. Experimental analyses encompassing entropy, correlation, energy, histogram, key sensitivity, key space, NPCR, and UACI reveal the efficacy of the proposed scheme. The scheme demonstrates significant statistical values (entropy: 7.9999, correlation: 0.0001, NPCR: 33.96, UACI: 96.79) and operates efficiently with a computational time of 0.002 for encrypting triple grayscale images simultaneously which shows its suitability for real-time applications.

1. Introduction

The rapid development in communication technology has given rise to the Internet of Things (IoT) that enables the realization of interconnected devices [1]. This development not only enhances the convenience of daily life, but also has a significant appeal for the industrial sector due to the smooth connectivity it offers. The integration of industry and IoT environment has led to the creation of the Industrial Internet of Things (IIoT) [2]. The IIoT incorporates perception, acquisition, monitoring, sensors, and the analysis of wireless communications and intelligent technologies. These elements are smoothly integrated into various stages of the industrial production process that propels traditional industries towards a new era of intelligence. In addition, the IIoT

* Corresponding author.
E-mail address: Arslan.shafique@glasgow.ac.uk (A. Shafique).

https://doi.org/10.1016/j.compeleceng.2024.109354
Received 14 March 2024; Received in revised form 21 April 2024; Accepted 25 May 2024
Available online 7 June 2024

interconnected environment plays a vital role in improving industrial manufacturing efficiency to improve the quality of products and minimize the costs and consumption of resources [3].

However, the development of multimedia files, specifically image data, within the IIoT poses challenges in terms of security [4,5]. The sharing of these industrial big data files that often include sensitive information requires careful consideration of the robust security. The complex connections between physical and network components in the IIoT present significant security challenges in such applications that involve real-time data exchange. To address privacy and security concerns in the IIoT, a safety measure involves encrypting data captured by IIoT smart devices before transmission to other nodes. Therefore, all transmitted or stored data in the IIoT environment becomes encrypted and only available to authorized users with the correct secrete key. Continuously developing such technologies that secure the sensitive information is necessary for ensuring the secure data transmission and storage in the IIoT environment.

To ensure the confidentiality of the information contained in digital images and prevent unauthorized transmission of sensitive data, commonly employed technologies in image information security encompass image hiding [6,7] and image encryption [8–11]. Information hiding involves concealing confidential data within the other plaintext data, leveraging the data redundancy of the information itself and the sensory redundancy of human sensory organs. This method effectively conceals plaintext within common media, making it challenging for attackers to detect the presence of secret information and achieving the goal of secure communication. Consequently, this technology has garnered scholarly attention in recent years [12–14].

Chaotic systems exhibit tremendous features such as pseudorandomness, and high sensitivity to secrete keys and ergodicity makes them suitable for cryptographic applications [15–17]. Due to their potential to enhance the robustness of encryption schemes, several chaotic systems including, chaotic logistic map, chaotic gauss iterated map and chaotic sine map are commonly employed in recent years on gray as well as color image encryption. Chaotic maps are classified as either low-dimensional (LD) [18] or high-dimensional (HD) [19]. The selection of these classes of chaotic systems in image cryptosystems involves a trade-off between complexity and the level of security. Low-dimensional chaotic systems provide simple hardware implementations while providing a satisfactory level of security to the digital data. This makes them a preferred option for applications that demand real-time image encryption efficiency. In contrast, high-dimensional chaotic systems provides robust security, but at the same time the implementation of such HD chaotic systems are more complex.

The proposed research introduces a new image encryption system that utilizes 3D-NEAT and 3D-LS for transmitting image information in the Industrial Internet of Things (IIoT). The proposed cryptosystem can encrypt three grayscale images simultaneously into a single color encrypted image that ensures both image privacy protection and reduced transmission data. Unlike the approach proposed in [20,21]that employs AT for two-step pixel position and value changes, the proposed cryptosystem converts plaintext images into bit-planes and each scrambled is applied on each bit in the extracted bit-planes through 3D-NEAT. This process alters the position and pixel values of the plaintext image simultaneously in a single step which also helps to reduce the encryption computational time. Following the scrambling procedure, the pixel values of the permuted images are substituted with values from a substitution box (S-box). The S-box of size 16 X 16 is generated using 4DHCFO in this research. Finally, HDNAC and IDCJ are employed to create diffusion in the substituted image in order to generate the color encrypted image. The proposed scheme has several advantages:

- Encryption involves combining two color images into a single color cipher image, introducing complexity for unauthorized individuals attempting to decipher the information illicitly.
- Diverging from most AT-based encryption approaches that merely permute the image pixels, the proposed scheme alters the pixel values, enhancing the effectiveness and security of image information protection.
- The chaotic sequences are employed in the proposed encryption scheme and the resulting color ciphertext image exhibit commendable randomness. Evaluation through the NIST SP 800-22 test battery confirms their ideal level of randomness.
- The proposed encryption scheme demonstrates high sensitivity to both the encryption key and the plaintext image. The mean square error (MSE) curve of the key confirms the heightened sensitivity of the scheme, and the results of the number of pixels change rate (NPCR) and unified average changing intensity (UACI) show its sensitivity to variations in the plaintext image.

1.1. Organization of the paper

The remaining sections of the paper are structured as follows: Section 2 provides a brief overview of existing encryption schemes in which their strengths, weaknesses, and potential solutions are highlighted. Section 3 delves into the preliminary knowledge that is essential for developing the proposed encryption framework. Section 4 presents the comprehensive flow of the proposed framework. Section 5 is dedicated to experimental results and analysis in which several security parameters are used to gauge the performance of the proposed encryption framework. Section 6 provides a few future recommendations that can be used to enhance the proposed encryption framework. Finally, Section 7 concludes the proposed work and outlines a few future recommendations for enhancing the proposed encryption scheme.

2. Related work

In recent years, researchers have shown significant interest in the challenge of image encryption which leads to the investigation of various methodologies proposed in the domain of chaotic systems [22,23], DNA computing, [24] Bit-plane extraction [25], S-box based encryption [26], confusion-diffusion operation [27], and scrambling operations. Chaotic systems, DNA encoding and bit-plane

extraction have emerged as a main point in the proposed research due to some specific properties including dynamic sensitivity, high information density and non-linear mapping. These characteristics render chaotic systems, DNA encoding and bit-plane extraction method highly suitable for image encryption applications that can offer a robust security for ensuring the privacy and protection of sensitive visual information.

In [28], Ravichandran et al. Proposed an encryption algorithm that incorporates integer wavelet transform (IWT), DNA, and chaos, exhibits potential vulnerabilities in the DNA-based diffusion phase, where the security of DNA coding and XOR operations is uncertain. The use of circular bitwise shifting in the first stage of diffusion may also introduce vulnerabilities, especially if not resilient against advanced cryptographic attacks. In [29], Hassan et al. proposed two high-payload embedding methods utilizing the Hue-Saturation-Value color model to achieve superior steganographic image quality. However, information encryption technology employs mathematical or physical means to safeguard electronic information during transmission and storage, preventing information leakage.

In [30], Dagadu et al. proposed an encryption scheme to secure medical images using chaotic systems and DNA coding. The scheme involves chaotic key generation, DNA diffusion, and the Message Digest Algorithm 5 hash function. The resulting cipher image demonstrates robustness against various attacks as it is shown by statistical analysis. Vulnerabilities in the proposed encryption scheme include weaknesses in the Message Digest Algorithm 5 hash function and susceptibility to advancements in cryptographic attacks on chaotic systems. The highly reliance on DNA encoding may also make the encryption scheme susceptible to emerging DNA-based cryptanalysis techniques. In [31], Zefreh et al. proposed an encryption scheme to secure digital images. Their scheme depends on hybrid DNA computing, chaotic systems, and hash functions. Moreover, the scheme involves DNA-level permutation using a logistic map-based mapping function and DNA-level diffusion with new algebraic DNA operators. Experimental results showed its effectiveness and resistance to known attacks. The vulnerability in their proposed encryption scheme lies in the DNA computing components, which are susceptible to cryptographic attacks. In [32], Mansoor et al. proposed Hybrid Adaptive Image Encryption (HAIE) scheme that incorporated statistical parameters to modify chaotic system conditions. The hybrid scheme permutes half of the plain image with a logistic map and the other half with a chaotic tent map. DNA encoding and subsequent operations contribute to a robust encryption effect, as shown by statistical measures and resilience against noise and known plaintext attacks. The selection of statistical parameters for chaotic systems in HAIE is not optimized, which makes the encryption scheme susceptible to statistical attacks. In [33], Maniyath et al. proposed a computationally efficient DNA-based encryption/decryption algorithm to secure data. Moreover, the primary goal of the scheme was to reduce the encryption computational time for large-size data. Using natural DNA sequences as secret keys, their algorithm employed double encryption on images containing hidden secret data before transmission. Leveraging reversible data hiding (RDH) for telemedicine, the technique supports authentication of robot-captured images and enables the embedding of electronic patient records (EPR) data into medical images. The plaintext images are divided into blocks before encryption, which employs machine learning, specifically support vector machines (SVM), for classifying and recovering original images from encrypted versions. As the natural DNA sequences used as secrecy keys are used in their encryption scheme, it can be suspectable to biological factors and potential errors in the DNA encoding process. Additionally, the use of reversible data hiding (RDH) and machine learning also introduces risks associated with advancements in cryptanalysis techniques targeting these specific methodologies.

In [34], Shafique et al. proposed a bit-plane extraction-based image encryption scheme to provide a suitable level of security to the digital images. Moreover, the purpose of incorporating the bit-plane extraction method is to reduce the overall encryption computational time. Their method begins with the extraction of binary bit planes from the plaintext image to achieve both confusion and diffusion. Moreover, chaotic cubic logistic and logistic maps and a new random image also contributed to the efficiency of the encryption algorithm. The bit-level permutation is weak due to the simple scrambling operation. In [35], Kumar et al. introduced a new approach for bit-plane embedding using a median edge detector and a difference predictor. Their method efficiently transforms the plaintext image into an enciphered 2D matrix. In addition, quad-tree-based bit-plane representation and redundancy mitigation strategies are also incorporated to enhance the security of digital images. The simple bit-plane embedding method in the median edge detector (MED) and the difference predictor make the technique susceptible to specific image processing attacks. In [36], Song et al. presented a parallel image encryption algorithm using intra-bitplane scrambling. With four threads for bit-level permutation and multiple threads for keystream generation in diffusion, the scheme is demonstrated to be secure and efficient through cryptographic analyses. The reliance on multiple threads for keystream generation introduces risks associated with vulnerabilities in the diffusion phase. In [37], Wu et al. proposed a high-capacity reversible data hiding (RDH) scheme to secure digital images using a new pre-processing method based on bit-plane extraction. This method allows for the reversible hiding of pixel values in least significant bit planes. The scheme utilized stream-cipher encryption and an efficient most significant bit (MSB) prediction method to achieve higher embedding capacity compared to existing RDH schemes for encrypted images. In [38], Morankar et al. proposed a robust encryption scheme for biometric fingerprint images using bit plane extraction and discrete wavelet transform (DWT). The process involves extracting the 8-bit planes form biometric image and applied DWT for encryption in the frequency domain through a cubic map.

In [39], Pourasad et al. Proposed a highly secure and fast image encryption method utilizing random chaos sequences. Despite limited accuracy, their scheme presented an improved technique by creating chaotic sequences and wavelet transform values. The high reliance on wavelet transform values creates challenges in the encryption process if not secured against advanced cryptographic attacks. In [40], Man et al. presented a dual-channel image encryption method (digital and optical channels) to secure double grayscale images with improved efficiency and resistance to attacks. The scheme uses a chaotic map to enhance key security and employs a chaotic sequence in a convolutional neural network to resist known-plaintext and chosen-plaintext attacks. A new image fusion method and dual-channel encryption enhance security and efficiency. The reliance on optical and digital channels poses

several challenges related to cybersecurity against advanced cryptographic attacks in the domain of image encryption. In [41], Yasser et al. proposed a hybrid encryption/decryption scheme for securing medical images in the domain of e-healthcare and IoT-enabled healthcare systems. The system utilizes chaotic maps for both confusion and diffusion rounds. Moreover, the authors have addressed the drawbacks of traditional chaos-based architectures. Their proposed scheme demonstrates efficiency, robustness, and protection against cyberattacks in various test images. Quantitative results with benchmark images reveal higher security levels of the encrypted images. The unproven randomness of chaotic sequences makes the encryption scheme vulnerable against differential attacks. In [42], Farhan et al. proposed a novel optical image encryption method that utilizes fractional Fourier transform, DNA sequence operation, and chaos theory. The approach involves generating random phase masks using an iterative Lorenz map that transforms the plain image into a DNA matrix and applying three fractional Fourier transforms. Their algorithm exhibits effective encryption with a large secret key space of size 2^{169} that makes it suitable for real-time applications. In [43], Alexan et al. presented a new image encryption framework utilizing two hyperchaotic maps and the single-neuron model. The process involves two stages: (i) substitution boxes and (ii) XORing with encryption keys. The method demonstrates superior performance and high encryption efficiency (8.54 Mbps). The framework was evaluated with NIST SP 800 standards, and evaluation metrics indicate its effectiveness and have an MSE value of 9626, a PSNR of 8.3 dB, MAE of 80.99, an entropy of 7.999, and a NPCR of 99. 6% and a UACI of 31. 49%. However, the use of a single neuron model (SNM) may pose a vulnerability related to the security of the entire system, as it relies heavily on the robustness of this specific model. The summary of the existing encryption scheme is provided in Table 1.

2.1. Contributions of the paper

To overcome the vulnerabilities present in Table 1, the contributions of the proposed research are as follows:

- The proposed research presents a new approach for data security in the Industrial Internet of Things (IIoT) through a triple-image encryption method. Unlike conventional encryption algorithms that operate on single images, the proposed approach processes three grayscale images simultaneously to generate a single encrypted color image. This innovative approach adds complexity to the encryption process in order to enhance security against cyberattacks.
- The proposed encryption framework leverages multiple encryption methods, such as the 3-D non-equilateral Arnold transform (3D-NEAT), extended one-dimensional chaotic jumping (1DCJ), and a four-dimensional hyperchaotic Chen map of fractional order (4DHCFO). These techniques are strategically employed to scramble pixel positions, perform bit-level scrambling, and introduce diffusion into the encrypted data, respectively.
- The combination of multi-layer permutation, multi-round permutation, diagonal permutation, and bit-level scrambling operations adds robustness to the encryption process, which makes it more challenging for attackers to receive plaintext data from the encrypted information.
- Despite the complexity of the encryption process, the proposed scheme exhibits efficient computational performance, with an overall encryption time of 0.002 for encrypting triple grayscale images simultaneously. This shows the suitability of the encryption scheme for real-time applications in IIoT environments.

3. Preliminaries

Based on the studies proposed by Shao et al. and Li et al. in [44,45], respectively, Wu et al. [46] proposed the 3D-NEAT and its corresponding inverse transformation. The mathematical form of the 3D-NEAT is given in Eq. (1).

$$\begin{cases} \begin{pmatrix} x'_\alpha \\ y'_\alpha \\ z'_\alpha \end{pmatrix} = \begin{pmatrix} 1 & b'_\alpha & 0 \\ c'_\alpha & 1 + b'_\alpha c'_\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x'_\gamma \\ y'_\gamma \\ z'_\gamma \end{pmatrix} \mod \begin{pmatrix} A \\ B \\ C \end{pmatrix} \\ \begin{pmatrix} x_\beta \\ y'_\beta \\ z'_\beta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & b'_\beta \\ 0 & c'_\beta & 1 + b'_\beta c'_\beta \end{pmatrix} \begin{pmatrix} x'_\alpha \\ y'_\alpha \\ z'_\alpha \end{pmatrix} \mod \begin{pmatrix} A \\ B \\ C \end{pmatrix} \\ \begin{pmatrix} x'_{\gamma+1} \\ y'_{\gamma+1} \\ z'_{\gamma+1} \end{pmatrix} = \begin{pmatrix} 1 + b'_{y'} c'_{y'} & 0 & c'_{y'} \\ 0 & 1 & 0 \\ b'_{y'} & 0 & 1 \end{pmatrix} \begin{pmatrix} x'_\beta \\ y'_\beta \\ z'_\beta \end{pmatrix} \mod \begin{pmatrix} A \\ B \\ C \end{pmatrix} \end{cases} \tag{1}$$

Where, $c'_\alpha = r_\alpha \left[\frac{B}{gcd(A,B)} \right]$, $c'_\beta = r_\beta \left[\frac{C}{gcd(B,C)} \right]$, and $c'_{y'} = r_{y'} \left[\frac{A}{gcd(C,A)} \right]$. gcd(·) is the greatest common divisor. $b'_\beta, b'_{y'}, b'_\alpha, r_\beta, r_{y'}$ and r_α are the integer values in the positive domain. The inverse transformation can be achieved using Equations and its inverse transformation of Eq. (1) can be achieved using Eqs. (2)–(4).

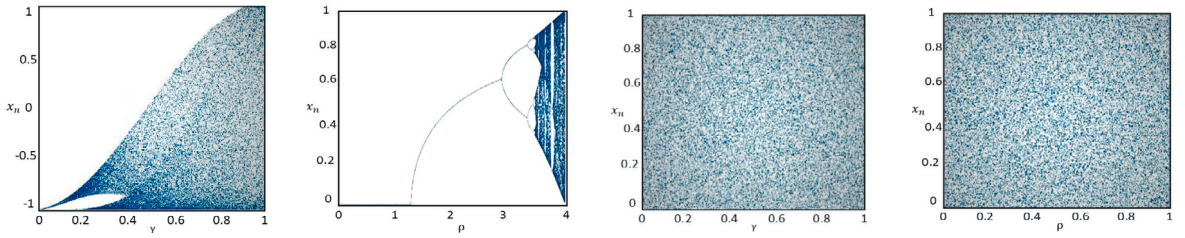
$$\begin{cases} y'_\beta = y'_{\gamma+1} \\ x'_\beta = (x'_{\gamma+1} - c'_{y'} z'_{\gamma+1}) \mod(A) \\ z'_\beta = (z'_{\gamma+1} - b'_{y'} x'_\beta) \mod(C) \end{cases} \tag{2}$$

Table 1
Summary of the existing encryption scheme.

Methodology	Application domain	Robustness against attacks	Advantages	Disadvantages	Potential solutions
Ravichandran et al. [28]	–	Uncertain (DNA phase)	–	Vulnerable DNA coding, XOR operations	Improve DNA-based diffusion, Address circular bitwise shifting
Hassan et al. [29]	Steganographic image quality	High	High-quality steganographic images	Prevents information leakage	–
Dagadu et al. [30]	Secure medical images	Robust	Statistical analysis, DNA diffusion	Weaknesses in Message Digest Algorithm 5, Vulnerable to cryptanalysis	Improve Message Digest Algorithm 5, Address vulnerabilities in chaotic systems
Zefreh et al. [31]	Secure digital images	Resistant	Effectiveness against attacks	Vulnerable DNA computing components	Strengthen DNA computing components
Mansoor et al. [32]	Hybrid Adaptive Image Encryption	Robust	Robust encryption, Resilience against noise	Non-optimized statistical parameters	Optimize statistical parameters for chaotic systems
Maniyath et al. [33]	DNA-based encryption/decryption	Efficient	Computational time reduction, Support for telemedicine	Susceptible to biological factors	Enhance security against biological factors
Shafique et al. [34]	Image encryption	Suitable	Reduced computational time	Weak bit-level permutation	Strengthen bit-level permutation
Kumar et al. [35]	Bit-plane embedding	Enhanced	2D matrix encryption, Redundancy mitigation	Susceptible to image processing attacks	Improve resistance against image processing attacks
Song et al. [36]	Parallel image encryption	Secure	Efficiency through cryptographic analyses	Risks with multiple threads for keystream generation	Mitigate risks in keystream generation
Wu et al. [37]	High-capacity RDH scheme	Secure digital images	Higher embedding capacity	–	–
Morankar et al. [38]	Robust encryption for biometric fingerprint images	Robust	Encryption in frequency domain	–	–
Pourasad et al. [39]	Image encryption	Fast	Improved technique	Challenges with wavelet transform values	Address challenges in wavelet transform values
Man et al. [40]	Dual-channel image encryption	Efficient	Improved efficiency	Challenges with optical and digital channels	Enhance security in optical and digital channels
Yasser et al. [41]	Hybrid encryption/decryption	E-healthcare and IoT-enabled healthcare	Efficiency, Robustness	Unproven randomness of chaotic sequences	Enhance randomness in chaotic sequences
Farhan et al. [42]	Optical image encryption	Real-time applications	Effective encryption	–	–
Alexan et al. [43]	Image encryption framework	Superior performance	High encryption efficiency	Vulnerability with single neuron model	Address vulnerability in single neuron model

$$\begin{cases} x'_{\alpha} = x'_{\beta} \\ z'_{\alpha} = (z'_{\beta} - c'_{\beta} y'_{\beta}) \mod(C) \\ y'_{\alpha} = (y'_{\beta} - b'_{\beta} z'_{\alpha}) \mod(B) \end{cases} \tag{3}$$

$$\begin{cases} z'_{\gamma} = z'_{\alpha} \\ y'_{\gamma} = (y'_{\alpha} - c'_{\alpha} x'_{\alpha}) \mod(B) \\ x'_{\gamma} = (x'_{\alpha} - b'_{\alpha} y'_{\gamma}) \mod(A) \end{cases} \tag{4}$$



(a) Bifurcation representation of Chebyshev map

(b) Bifurcation representation of Sine map

(c) Bifurcation representation of Chebyshev-Chebyshev map

(d) Bifurcation representation of Sine-Sine map

Fig. 1. Bifurcation representation of different chaotic systems.

3.1. 1DCJ based bit-scrambling

The Chebyshev map depicts a chaotic one-dimensional system characterized by a single control parameter, as illustrated in Eq. (5) [47]. In Fig. 1(a), the bifurcation diagram of the Chebyshev map is presented. Notably, when the parameter γ exceeds 0.8, the system exhibits chaotic behavior.

$$x_{i+1} = \cos(\gamma \arccos(x_i)), \quad i \in [0, \infty] \tag{5}$$

Here, x_0 is the initial condition that can be in the range $x \in (0, 1)$. γ is the control parameter. Chaos-based image encryption algorithms frequently employ sine mapping, a well-established one-dimensional chaotic function, as depicted in Eq. (6) [48].

$$x_{i+1} = \rho y_i(\pi y_i), \quad i \in [0, \infty] \tag{6}$$

Here, ρ is the control parameter. Through the comparison of output sequences from two identical one-dimensional (1D) chaotic maps, Pak et al. [49] introduced an algorithm for constructing an efficient chaotic system with an extended parameter range and enhanced chaotic performance. Eq. (7) expands on traditional one-dimensional chaos.

$$F(\rho, x_n, k) = F_{\text{chaos}}(\rho, x_n) \cdot G(k) - \lfloor F_{\text{chaos}}(\rho, x_n) \cdot G(k) \rfloor \tag{7}$$

where F_{chaos} represents one of the 1D chaotic functions, as specified in Eqs. (5) and (6), while $F(\rho, x_n, k)$ corresponds to a newly proposed chaotic system proposed in [49]. The function $F(\rho, x_n, k)$ retains its chaotic behavior within the extended range of $\rho \in [0, 10]$. The symbol $\lfloor \rfloor$ denotes the floor function. Additionally, $G(k)$ stands for an adjustable function parameterized by k. Optimal chaotic properties are observed when the parameter k is selected from the interval $k \in [8, 20]$.

The bifurcation diagram in Fig. 1(c) and 1(d) demonstrates the expansion of Fig. 1(a) and 1(b), respectively, indicating an increased parameter selection range in the extended one-dimensional chaotic model.

This paper presents a new random DNA encoding operation group diffusion algorithm aiming to propose an optimal or nearly optimal ciphertext solution within an extensive solution space. The algorithm generates sequences of lengths $8 \times m \times n$ and $4 \times m \times n$ through a parallel chaotic sequence, where m and n denote the length and width of the encrypted image. These sequences are organized into units of 8 and 4, undergo sorting within each group, and result in the generation of an index sequence. The process forms a total of 32 encoding and operation combinations, from which the most secure image is selected among the ciphertext images generated. To visually exemplify this process, the paper considers the pixel block encryption of a 4×4 pair, illustrating the randomly generated sequences of lengths $8 \times 4 \times 4$ and $4 \times 4 \times 4$.

4. Proposed encryption framework

The proposed encryption framework for securing digital images is divided into three major stages. In the first stage, the pixels of the plaintext image undergo scrambling through the application of 3D-NEAT. Recognizing the vulnerability of single-stage scrambling in [34], a three-step scrambling approach is implemented in this stage. This includes the incorporation of multi-layer permutation (ML-P), multi-round permutation (MR-P), and diagonal permutation (D-P). Moving on to the second stage, before the utilization of 1DCJ, bit-planes are extracted from the image that underwent scrambling in the first stage. The 1DCJ is then employed to permute the bits within each extracted bit plane. The third stage involves the generation of an S-box through 4DHFCO, which aims to introduce diffusion in the image pixels. According to the principle of Shannon, an encryption algorithm that encompasses both confusion and diffusion is considered more secure than one containing only confusion or diffusion [50]. Although various S-boxes are available in the literature, the reason to create a new S-box is its superior strength compared to existing ones. The proposed S-box also contributes to enhancing diffusion in the image pixels, thus improving the overall security of the plaintext data. The generalized flow diagram of the proposed encryption framework is shown in Fig. 2.

The major steps involved in the proposed encryption framework are given below:

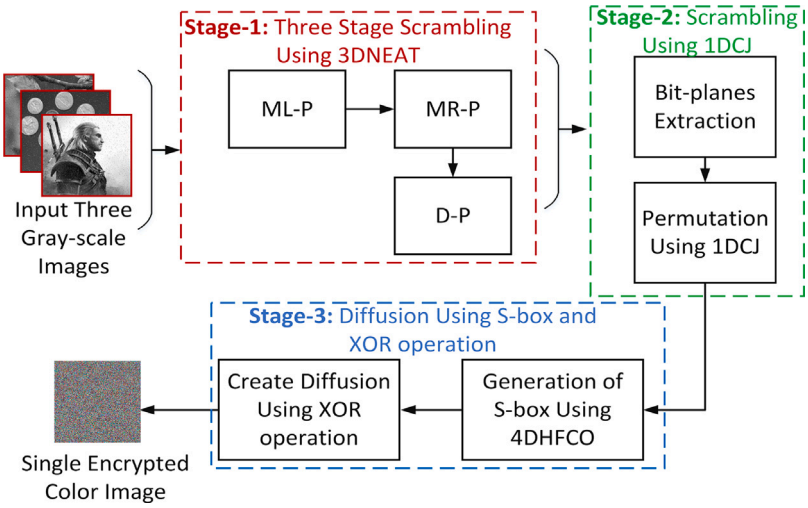


Fig. 2. Generalized flow work of the proposed encryption framework.

- Implement 3D NEAT to incorporate the scrambling process. The scrambling process involves ML-P, MR-P, and D-P
- Extraction of bit planes and apply permutation to each bit-plane. The most significant bitplanes (MSBPs) and the least significant bitplanes (LSBPs) are permuted according to jumping patterns x_i and y_i, respectively.
- Generation of a new S-box.
- Perform substitution of pixel with the respective S-box values.
- Convert the S-box into 4 blocks of size 4×4.
- Perform an XOR operation on the resultant image with each block.

The detailed version of the proposed encryption framework is given in the next subsections. The block diagram that illustrates the proposed encryption framework can be found in Fig. 3. The following subsections provide a detailed version of the proposed encryption framework.

4.1. Implementation of 3D NEAT in permutation process

The chaotic behavior in the Arnold transform is generally associated with rotation angles that are irrational multiples of π. Specifically, rotation angles such as $\frac{\pi}{2}$, $\frac{\pi}{3}$, $\frac{\pi}{4}$ $\cdots$, are known to induce chaotic behavior in the Arnold transform. In the proposed work, the initial values for $y'_\beta, x'_\beta, z'_\beta, x'_\alpha, z'_\alpha, y'_\alpha, z'_\gamma, y'_\alpha$, and x'_γ are set to be 0.34, 0.29, 0.24, 0.39, 0.46, 0.33, 0.27, 0.41, and 0.38, respectively. Moreover, the values of the other parameters such as $c'_{y'_\gamma} = 1.64$, $b'_{y'_\gamma} = 2.16$, $c'_\beta = 1.91$, $b'_\beta = 2.31$, $c'_\alpha = 3.01$, and $b'_\alpha = 2.57$ are taken from the chaotic interval ([03.1415]) of the Arnold transform. There are three pairs of chaotic sequences that will be generated such as Pair-1: x'_β, z'_β, Pair-2: z'_α, y'_α, Pair-3: y'_γ, and x'_γ will be generated. Each pair will be used for each type of scrambling. There are three types such as ML-P, MR-P, and D-P of scrambling are used in the proposed work to permute the pixel of an input image. ML-P, MR-P, and R-P are defined below:

- **ML-P:** In ML-P, the image pixels undergo permutation between different layers. For example, in the proposed approach, three different layers of permutation are implemented, column-wise, row-wise, and block-wise.
- **MR-P:** In MR-P, the permutation process involves multiple iterations.
- **D-P:** A diagonal permutation involves rearranging elements along the diagonal of a matrix according to a specified random sequence, while the remaining elements of the matrix retain their original positions.

4.1.1. ML-P

Consider a small portion of the image pixels $I = \begin{bmatrix} 150 & 29 & 231 & 137 \\ 6 & 137 & 201 & 193 \\ 83 & 91 & 155 & 190 \\ 156 & 173 & 91 & 13 \end{bmatrix}$ as an example. To permute the pixel of I, the first pair of random sequences i.e. x'_β, z'_β will be used. For example, x'_β = [2 4 1 3], and z'_β = [3 1 4 2]. The sequence x'_β will be used for row-wise and column-wise scrambling, while the sequence z'_β will be used for block-wise permutation. After applying the sequence x'_β for row permutation the resultant image will be : $I_r = \begin{bmatrix} 6 & 137 & 201 & 193 \\ 156 & 173 & 91 & 13 \\ 150 & 29 & 231 & 137 \\ 83 & 91 & 155 & 190 \end{bmatrix}$, and for column permutation, the extended version

of I_r will be: $I_c = \begin{bmatrix} 137 & 193 & 6 & 201 \\ 173 & 13 & 156 & 91 \\ 29 & 137 & 150 & 231 \\ 91 & 190 & 83 & 155 \end{bmatrix}$. Once, row-wise and column-wise permutation is performed, I_c will be divided into four blocks such as $B_1 = \begin{bmatrix} 137 & 193 \\ 173 & 13 \end{bmatrix}$, $B_2 = \begin{bmatrix} 6 & 201 \\ 156 & 91 \end{bmatrix}$, $B_3 = \begin{bmatrix} 29 & 137 \\ 91 & 190 \end{bmatrix}$, and $B_4 = \begin{bmatrix} 150 & 231 \\ 83 & 155 \end{bmatrix}$. After applying block-wise permutation using the random sequence z'_β, the resultant image will be: $I_B = \begin{bmatrix} 29 & 137 & 137 & 193 \\ 91 & 190 & 173 & 16 \\ 150 & 231 & 6 & 201 \\ 83 & 155 & 150 & 91 \end{bmatrix}$.

4.1.2. MR-P

In multi-round permutation, there are three iterations of pixel permutation that are performed. To perform the pixel permutation, the processed image I_B is converted into a 1D matrix i.e. I'_B [29 137 137 193 91 190 173 16 150 231 6 201 83 155 150 91], and then converted into four blocks such as $B'_1 = \begin{bmatrix} 29 & 137 \\ 137 & 193 \end{bmatrix}$, $B'_2 = \begin{bmatrix} 91 & 190 \\ 173 & 16 \end{bmatrix}$, $B'_3 = \begin{bmatrix} 150 & 231 \\ 6 & 201 \end{bmatrix}$, and $B'_4 = \begin{bmatrix} 83 & 155 \\ 150 & 91 \end{bmatrix}$. The two iterations of the permutation will be performed using the random sequence pair-2: z'_α, y'_α. Let, the random sequences z'_α, y'_α are [1 3 4 2] and [4 2 3 1], respectively. The resultant image after first permutation iteration will be: $I_{i1} = \begin{bmatrix} 29 & 137 & 150 & 231 \\ 137 & 193 & 6 & 201 \\ 83 & 155 & 91 & 190 \\ 150 & 91 & 173 & 16 \end{bmatrix}$, and after second permutation iteration will be: $I_{i2} \begin{bmatrix} 91 & 190 & 150 & 231 \\ 173 & 16 & 6 & 201 \\ 83 & 155 & 29 & 137 \\ 150 & 91 & 137 & 193 \end{bmatrix}$.

4.1.3. D-P

Two diagonal permutations are performed according to the pair-3 of the random sequence: one along the main diagonal ([91, 16, 29, 193]) and another along the secondary diagonal ([231, 6, 155, 150]). Let y'_γ, and x'_γ be [4 1 2 3], and [3 4 2 1], respectively. The resulting images following the main diagonal permutation and secondary diagonal permutation are $I_m = \begin{bmatrix} 193 & 190 & 150 & 231 \\ 173 & 91 & 6 & 201 \\ 83 & 155 & 16 & 137 \\ 150 & 91 & 137 & 29 \end{bmatrix}$ and $I_s = \begin{bmatrix} 193 & 190 & 150 & 155 \\ 173 & 91 & 150 & 201 \\ 83 & 6 & 16 & 137 \\ 231 & 91 & 137 & 29 \end{bmatrix}$, respectively. Algorithm 1 illustrates the scrambling process utilized within the proposed encryption framework.

4.2. Extraction of bit-planes

After completing the scrambling process, the bit planes are extracted from the scrambled images using Eq. (8).

$$\begin{cases} I_{sb_1} = (\frac{I_s}{1}) mod 2, & I_{sb_2} = (\frac{I_s}{2}) mod 2, & I_{sb_3} = (\frac{I_s}{4}) mod 2, & I_{sb_4} = (\frac{I_s}{8}) mod 2, \\ I_{sb_5} = (\frac{I_s}{16}) mod 2, & I_{sb_6} = (\frac{I_s}{32}) mod 2, & I_{sb_7} = (\frac{I_s}{64}) mod 2, & I_{sb_8} = (\frac{I_s}{128}) mod 2 \end{cases} \tag{8}$$

I_{sb_8} to I_{sb_5} correspond to the most significant bit planes ($MSBPs$), while I_{sb_4} to I_{sb_1} pertain to the least significant bit planes ($LSBPs$). Each specific bit-plane contains unique information levels, which can be calculated using Eq. (9). The visual representation of the distribution of different information levels can also be seen in Fig. 4.

$$I_{(sb_i)} = \frac{2^{i-1}}{\sum_{i=1}^{8} 2^{i-1}} \tag{9}$$

Two chaotic jumping patterns (x_i, and x'_i) are generated using Eqs. (5) and (6), respectively. Let, x_i and x'_i be [3 4 1 2], and [3 1 2 4], respectively. Before applying the chaotic jump patterns on the given image, bit-planes are extracted from the image I_s. The binary version of I_s will be: $I_{bin} = \begin{bmatrix} 11000001 & 10111110 & 10010110 & 10011011 \\ 10101101 & 01011011 & 10010110 & 11001001 \\ 01010011 & 00000110 & 00010000 & 10001001 \\ 11100111 & 01011011 & 10001001 & 00011101 \end{bmatrix}$. The bit-planes which are extracted from I_{bin}

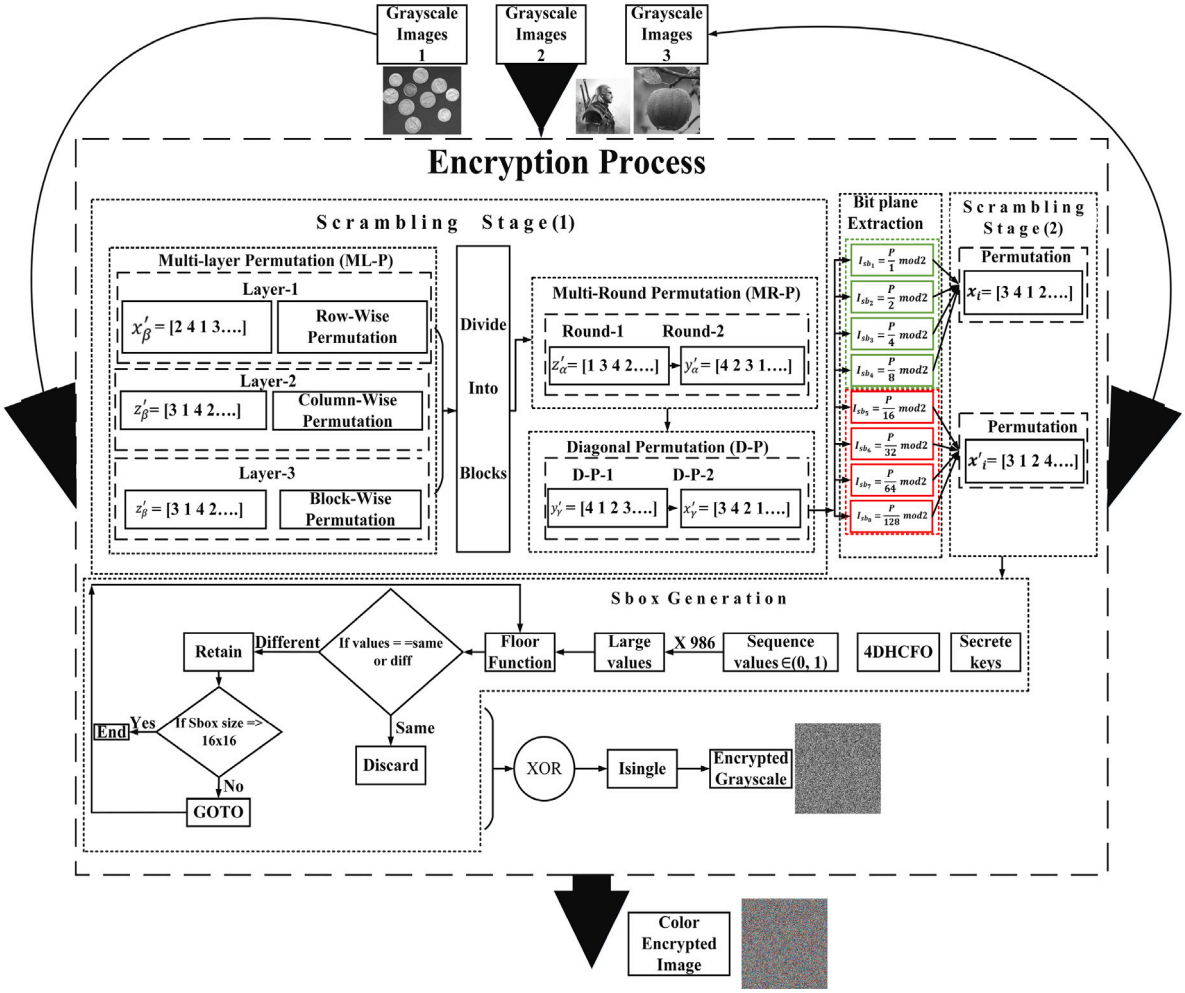
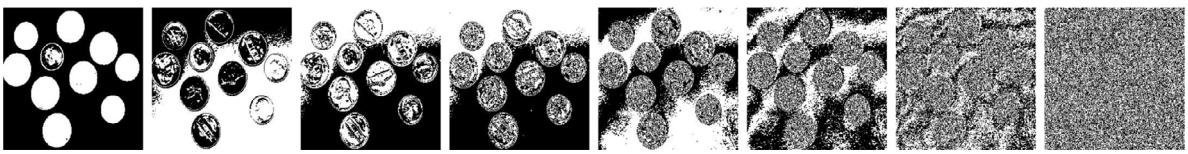


Fig. 3. The proposed encryption framework.

(a) Plaintext image ($I_s(i,j)$)



(b) I_{sb_8} (c) I_{sb_7} (d) I_{sb_6} (e) I_{sb_5} (f) I_{sb_4} (g) I_{sb_3} (h) I_{sb_2} (i) I_{sb_1}

Fig. 4. Bit-plane decomposition ((I_{sb_j})).

Algorithm 1 Scrambling process

Start
Input: img = plaintext image
1) Multi-layer permutation
Layer-1: Row-wise permutation
[row, col] = size(img)
    for R1 = 1:row
        row_perm(x'_β(R1),:) = img(R1,:);
    end
Layer-1: Column-wise permutation
    for for C1= 1:col
        col_perm(:,z'_β(C1)) = row_perm(:,col1);
    end
2) Multi-round permutation
num_rounds = 3; ▷ Define the number of rounds
    for round = 1:num_rounds
permuted_indices = randperm(numel(permuted_blocks)); ▷ Shuffle the pixels of the image
MR_permuted_img = reshape(img(permuted_indices), size(img));
    end
3) Diagonal permutation
diagonal_seq = randperm(min(rows, cols)) ▷ Define the diagonal sequence
    for i = 1:length(diagonal_seq)
        diagonal_elements = diag(MR_permuted_img, diagonal_seq(i) - 1) ▷ Extract the diagonal elements
        shuffled_diagonal = diagonal_elements(randperm(length(diagonal_elements))) ▷ Permute the diagonal elements
        D-P_img = MR_permuted_img - diag(diagonal_elements, diagonal_seq(i) - 1) ▷ Replace the diagonal elements with shuffled ones
        D-P_img = MR_permuted_img + diag(shuffled_diagonal, diagonal_seq(i) - 1)
    end
End

are given below:

$$I_{bin8}=\begin{bmatrix}1&1&1&1\\1&0&1&1\\0&0&0&1\\1&0&1&0\end{bmatrix}, I_{bin7}=\begin{bmatrix}1&0&0&0\\0&1&0&1\\1&0&0&1\\1&1&0&0\end{bmatrix}, I_{bin6}=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&0\\1&0&0&0\end{bmatrix}, I_{bin5}=\begin{bmatrix}0&1&1&1\\0&1&1&0\\1&0&1&0\\0&1&0&1\end{bmatrix}$$

$$I_{bin4}=\begin{bmatrix}0&1&0&1\\1&1&0&1\\0&0&0&1\\0&1&1&1\end{bmatrix}, I_{bin3}=\begin{bmatrix}0&1&1&0\\1&0&1&0\\0&1&0&0\\1&0&0&1\end{bmatrix}, I_{bin2}=\begin{bmatrix}0&1&1&1\\0&1&1&0\\1&1&0&0\\1&1&0&0\end{bmatrix}, I_{bin1}=\begin{bmatrix}1&0&0&1\\1&1&0&1\\1&0&0&1\\1&1&1&1\end{bmatrix}$$

Now, the binary bit-planes I_{bin8} to I_{bin5} are permuted row-wise according to the jumping pattern x_i, while the bit-planes I_{bin4} to I_{bin1} are also permuted row-wise according to the jump pattern x'_i. The permuted versions of the bit-planes I_{bin8} to I_{bin1} are given below:

$$I_{per8}=\begin{bmatrix}0&0&0&1\\1&0&1&0\\1&1&1&1\\1&0&1&1\end{bmatrix}, I_{per7}=\begin{bmatrix}1&0&0&1\\1&1&0&0\\1&0&0&0\\0&1&0&1\end{bmatrix}, I_{per6}=\begin{bmatrix}0&0&0&0\\1&0&0&0\\0&1&0&0\\1&0&0&0\end{bmatrix}, I_{per5}=\begin{bmatrix}1&0&1&0\\0&1&0&1\\0&1&1&1\\0&1&1&0\end{bmatrix}$$

$$I_{per4}=\begin{bmatrix}0&0&0&1\\0&1&0&1\\1&1&0&1\\0&1&1&1\end{bmatrix}, I_{per3}=\begin{bmatrix}0&1&0&0\\0&1&1&0\\1&0&1&0\\1&0&0&1\end{bmatrix}, I_{per2}=\begin{bmatrix}1&1&0&0\\0&1&1&1\\0&1&1&0\\1&1&0&0\end{bmatrix}, I_{per1}=\begin{bmatrix}1&0&0&1\\1&0&0&1\\1&1&0&1\\1&1&1&1\end{bmatrix}$$

To convert the binary versions of the bit-planes (I_{per8}-I_{per1}) into a single image (I_{single}), apply Eq. (10).

$$\mathbf{I_{single}} = 2\times(2\times(2\times(2\times(2\times(2\times I_{per8}+I_{per7})+I_{per6})+I_{per5})+I_{per4})+I_{per3})+I_{per2})+I_{per1} \tag{10}$$

The image $\mathbf{I}_{\text{single}}$ will be after applying Eq. (10):

$$\mathbf{I}_{\text{single}} = \begin{bmatrix} 01010011 & 00000110 & 00010000 & 11001001 \\ 11100001 & 01011110 & 00011011 & 00011011 \\ 11001101 & 10111011 & 10010110 & 10011001 \\ 10100111 & 01011011 & 10011001 & 11001101 \end{bmatrix} \rightarrow \mathbf{I}_{\text{single}} = \begin{bmatrix} 83 & 6 & 16 & 201 \\ 225 & 94 & 27 & 27 \\ 205 & 187 & 150 & 153 \\ 167 & 91 & 153 & 205 \end{bmatrix}$$

The summary process of 1DCJ is given in Algorithm 2.

Algorithm 2 1DCJ process

Start
Input: Any bit-plane (B) of size $m \times n$; ▷ m: pixel rows, n pixel columns
Generate two sequences S_1 and S_2 using Eqs. (5) and (6), respectively.
  for i = 1: m
    perm(S_1(i),:) = B(i,:) ▷ For $MSBs$
  end
  for i = 1: n
    perm$_2$(:,$S_2(n)$) = B(:,n) ▷ For $LSBs$
  end
End

4.3. Generation of S-box

In image encryption, incorporating an S-box is a vital step for introducing diffusion. This work not only includes an S-box for confusion, but also enhances the overall encryption technique by expanding the key space. A seed-based, randomly generated S-box utilizes the 4D hyperchaotic Chen system of fractional order. Hyperchaotic systems, known for having more than one positive Lyapunov exponent, are chosen for generating Pseudo-Random Number Generator (PRNG) sequences, transforming into an S-box. The fractional order 4D Chen system introduces multiple control variables, effectively increasing the key space. Moreover, the Chen system achieves a balance with high ergodicity, improved phase-space distribution, and acceptable computational complexity. This choice is highlighted when comparing it to alternatives like the Lorenz system, which has poorer ergodicity, or the hyperchaotic memristor circuit with intricate non-linearities and software implementation challenges. The hyperchaotic fractional-order 4D Chen system is represented by four equations for variables i, j, k, and l, respectively [51].

$$\begin{cases} D_i^{\gamma} = b(j - i) + l \\ D_j^{\gamma} = \xi i - ik + dj \\ D_k^{\gamma} = ij - ck \\ D_l^{\gamma} = jk + el \end{cases} \tag{11}$$

Eq. (11) categorizes the control variables into three groups: initial values (i_0, j_0, k_0, l_0), scale coefficients (b, c, d, ξ, e), and differential order (γ), totaling 10 variables. Fig. 5 illustrates a sample plot of the fractional order 4D Chen system.

The behaviour of hyperchaotic system is analyzed through bifurcation plots (Figs. 6 and 7 for b and c) and four Lyapunov characteristic exponents (Fig. 8), indicating the rate of divergence from perturbed initial conditions.

To generate an S-box with 10 keys, the fractional order 4D Chen system is numerically solved, resulting in a 4D geometry. The i, j, k, and l coordinates are then flattened into a single 1D array for further processing as follows:

$$\left[[i_0, j_0, k_0, l_0], [i_1, j_1, k_1, l_1], \ldots\right] \rightarrow \left[i_0, j_0, k_0, l_0, i_1, j_1, k_1, l_1, \ldots\right] \tag{12}$$

Using the flattened sequence, the median is computed, and this result is employed to transform the sequence into a bit stream through the following process:

$$Sream(S_i) = \begin{cases} 1, & if\, S_i \geq median \\ 0, & otherwise \end{cases} \tag{13}$$

Utilizing a set of 2048 bits, each 8 bits are employed to create a decimal number within the range [0, 255], resulting in a list of size 256. With a given list L of 256 numbers (unsorted and may contain repetitions) and a sorted set S [0, 255], the S box is efficiently constructed in constant time. For example, using the keys [i, j, k, l] = 0.25, b = 37, c = 3.5, d = 14, ξ = 33, e = 0.6, and γ = 0.93, the initial 8 bits in the generated bit stream are [0, 0, 0, 0, 1, 1, 1, 0]. Upon conversion to a decimal number, the result is 14, representing the first entry in the S-box, as illustrated in Table 1.

As an illustration, employing seed values [i, j, k, l] = 0.25, b = 37, c = 3.5, d = 14, ξ = 33, e = 0.6, and γ = 0.93 results in the Sbox generated, as shown in Table 2.

Although it is a 4D system, the computational cost of such a system is expected to be high in terms of time and memory allocation [52]. However, in the context of this work, this is not the case, as it is utilized to generate a fixed-size set of bits (specifically, 2048 bits). Consequently, the computation cost for this step remains constant. On the other hand, the significant

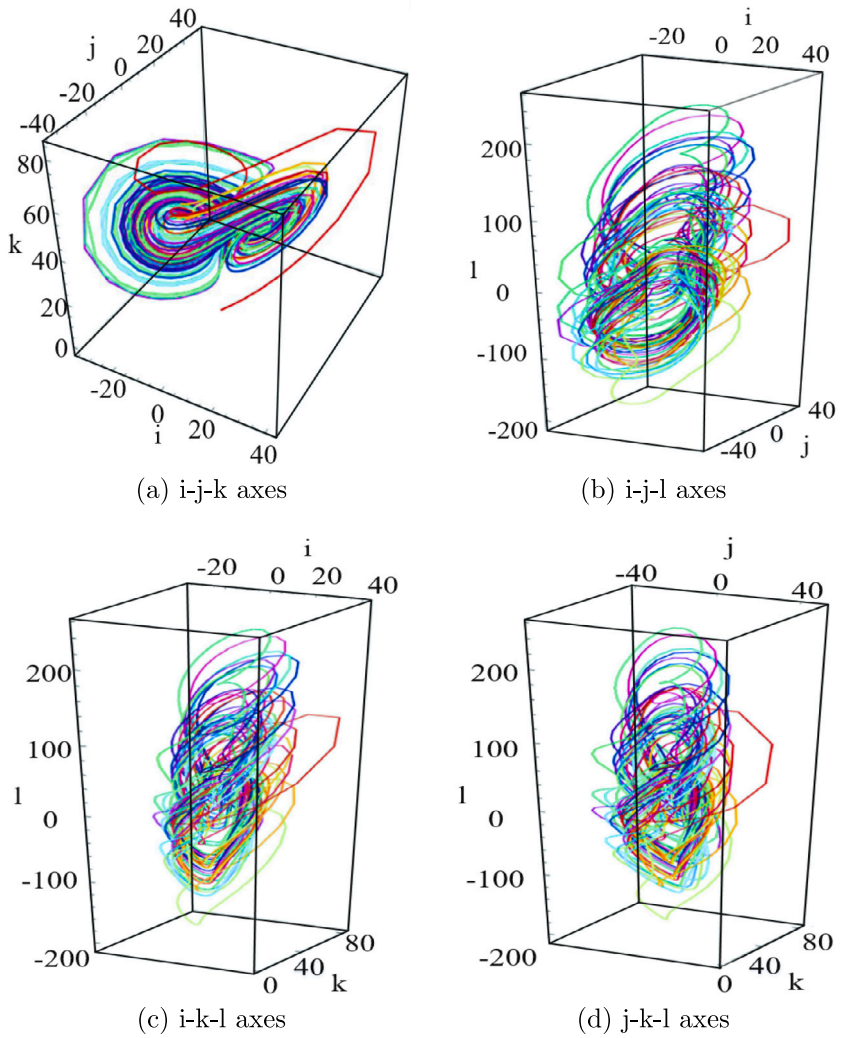


Fig. 5. 3D plots of the 4DHFCO with specified parameters, exploring different axis combinations ($\{i, j, k, l\}$ = 0.34, b = 41, c = 2.6, d = 16, ξ = 31, e = 0.49, γ = 0.95). Initial values are required for all four axes in the 4D space, but for clarity, one axis is excluded in each illustration.

Table 2
Arrangement of the values of the proposed Sbox.

90	103	168	74	114	159	30	60	16	76	135	182	160	20	13	221
181	148	109	75	71	144	66	256	33	69	187	241	142	143	218	22
116	97	83	164	228	94	108	73	78	183	65	15	174	172	6	213
112	162	198	107	27	170	132	197	212	242	139	206	216	210	173	207
68	106	123	136	146	110	85	23	36	157	180	5	211	239	214	1
237	138	93	89	167	127	193	39	29	163	236	192	115	175	220	2
130	67	49	185	131	81	54	48	21	137	17	31	234	219	8	203
72	238	26	118	84	205	14	117	52	184	58	195	179	38	177	233
24	80	125	63	196	88	32	247	113	64	217	18	70	243	244	141
79	155	86	248	122	42	82	200	235	245	61	35	232	249	165	225
100	102	41	149	50	40	202	92	153	147	208	222	169	209	134	176
154	151	43	98	22	121	124	47	111	186	46	105	201	19	190	223
101	158	119	161	126	55	45	255	87	62	251	254	28	224	229	215
152	104	59	140	37	12	34	96	120	250	166	57	7	231	3	11
99	145	2	91	51	128	25	194	56	199	133	171	227	230	189	188
156	95	129	178	150	77	44	53	9	10	191	252	204	240	4	253

advantage offered by this representation of the S-box lies in the incorporation of 10 keys, substantially enhancing the key space of the overall image cryptosystem.

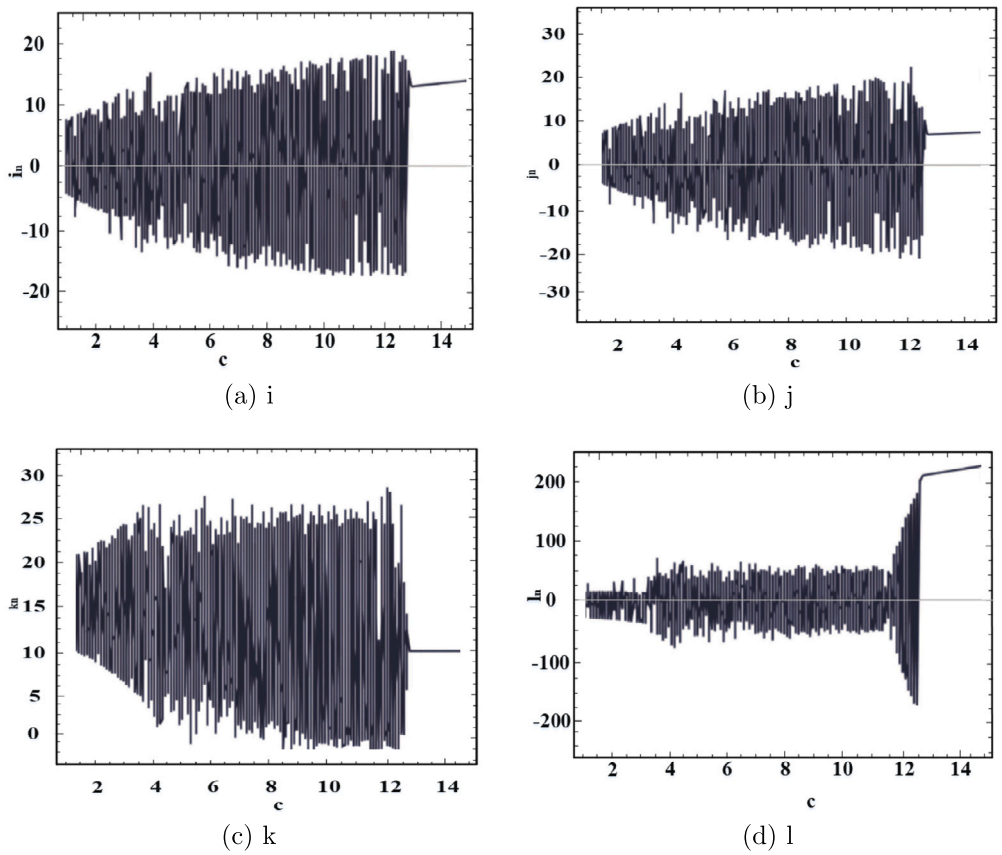


Fig. 6. Bifonal order 4DHFCO for i, j, k and l against c.

The construction of a new S-box serves two major roles. Firstly, it aims to introduce diffusion in the processed image through the substitution process as given in Algorithm 3. Secondly, it aims to create diffusion by employing the XOR operation as outlined in the process described in Algorithm 4. The outcome obtained by applying the XOR operation represents the initial encrypted image corresponding to the first grayscale image. Similarly, the second and third encrypted images are generated in correspondence to the second and third input grayscale images. To create the final color encrypted image, combine the three encrypted images. Fig. 9 displays three grayscale images along with their corresponding single-color images.

The selection rationale for the 3DNEAT, 1DCJ, and 4DHCFO techniques in our proposed encryption framework stems from their suitability for enhancing data security within the Industrial Internet of Things (IIoT) context. Specifically, such techniques are leveraged to provide multi-layered scrambling and diffusion processes, thereby adding confusion and increasing the complexity of the encryption scheme. This approach enhances resistance against potential attacks and ensures the confidentiality and integrity of sensitive IIoT data. Additionally, the comprehensive experimental analyses conducted demonstrate the effectiveness of our encryption framework in achieving high levels of security while maintaining low computational overhead, making it suitable for real-time IIoT applications. The NEAT, 1DCJ, and 4DHCFO are more suitable than other low-dimensional chaotic systems due to their specific attributes, such as multidimensional scrambling capabilities, increased complexity and confusion in encryption, and resilience against statistical attacks. The multidimensional scrambling capability of 3-D NEAT introduces higher complexity, while 1DCJ efficiently introduces randomness at the bit level, which contributes to enhancing the robustness of the encryption process. Additionally, the high-dimensional nature and fractional-order dynamics of 4DHCFO enhance the diffusion properties of the encryption, making it more resilient to attacks. These selected techniques strike a balance between complexity, efficiency, and security, making them well-suited for the demands of IIoT data security.

5. Experimental results and analysis

The main purpose of an encryption framework is to guarantee that the resulting encrypted image does not contain any plaintext information and that it can decrypt the plaintext image from the encrypted version with minimal loss of information. To evaluate the effectiveness of the proposed encryption framework, analysis and simulations are carried out using MATLAB 2019 to assess both the encryption and decryption frameworks. Furthermore, the experimental setup utilized to test the efficiency of the proposed

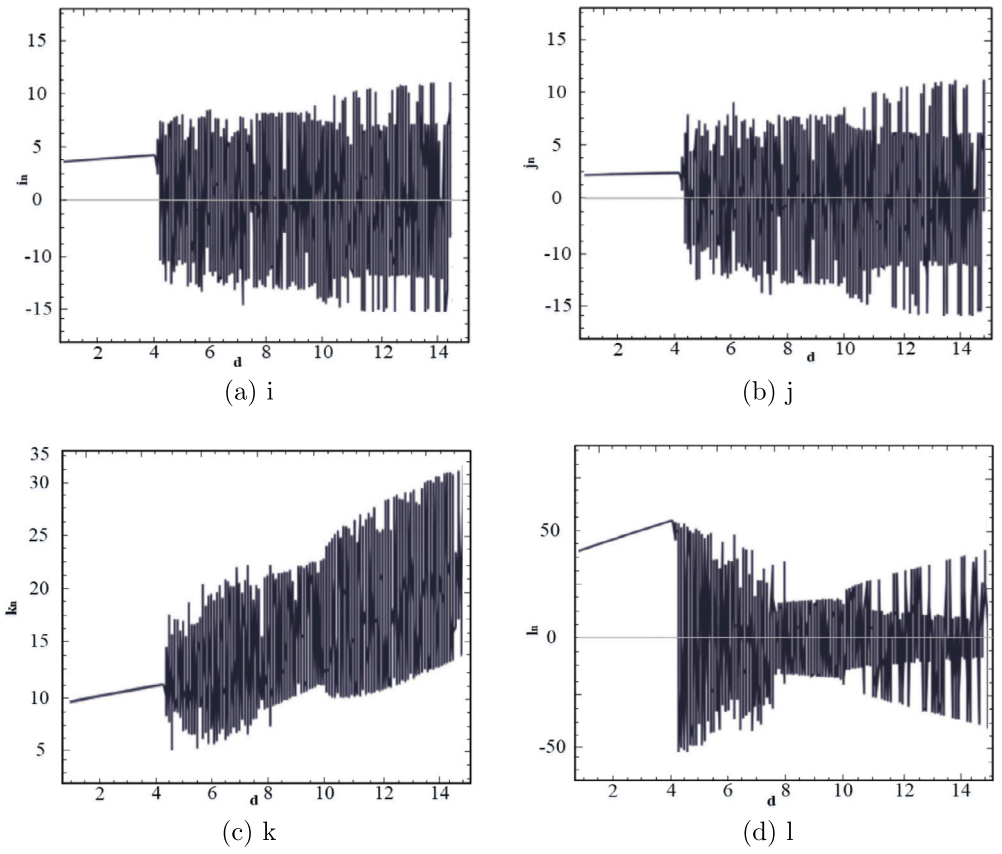


Fig. 7. Bifurcation plots of the 4DHFCOr for i, j, k and l against d.

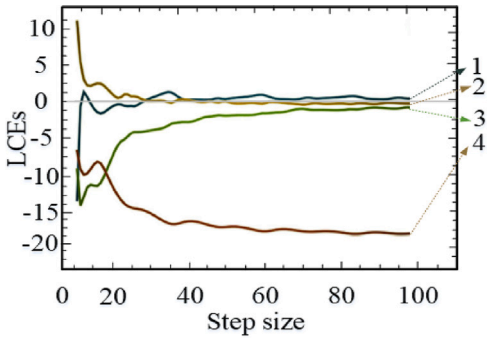


Fig. 8. Four Lyapunov exponents of 4dHFCO.

encryption framework involved a computer with the following specifications: 8 GB of RAM, 512 GB of solid-state drive (SSD), CoreTM i5-1135G7, 2.40 GHz, 2.42 GHz, and Windows 11.

The test images include traditional grayscale images as shown in Fig. 9(a, b, c, e, f, g, i, j, k). One encrypted color image is generated corresponding to each group of three grayscale images. When analyzing the images in Fig. 9(d, h, l), it is clear that the proposed encryption framework adequately conceals the pixels in the plain text images. Furthermore, the algorithm adeptly decrypts and restores the original images, ensuring no loss of visual data, as illustrated in Fig. 10 (b, c, d, f, g, h, j, k, l).

5.1. Keyspace analysis

Increasing the size of the key space is crucial for enhancing the resistance against brute force attacks. The key space of the proposed encryption framework encompasses various parameters such as y'_β, x'_β, z'_β, x'_α, z'_α, y'_α, z'_γ, y'_α, x'_γ, $c'_{y'}$, $b'_{y'}$, c'_β, b'_β, c'_α, and b'_α are

Algorithm 3 Diffusion using substitution process

Start

Input: A complete image: As an example, take a previous processed portion of an input image $\mathbf{I_{single}} = \begin{bmatrix} 83 & 6 & 16 & 201 \\ 225 & 94 & 27 & 27 \\ 205 & 187 & 150 & 153 \\ 167 & 91 & 153 & 205 \end{bmatrix}$, and Sbox.

```
for i=1:x
    for j=1:y
        bin = dec2bin(I_single(i,j),8);                    ▷ Converting decimal to binary
        S_R_b = [bin(1) bin(2) bin(3) bin(4)];             ▷ Separating most significant bits (MSBs)
        S_C_b = [bin(5) bin(6) bin(7) bin(8)];             ▷ Separating least significant bits (LSBs)
        S_R_d = bin2dec(S_R_b) + 1;                        ▷ Converting MSBs into decimal
        S_C_b = bin2dec(S_C_b) + 1;                        ▷ Converting LSBs into decimal
        Sbox_image(i,j)= sbox(S_R_d,S_C_b);  ▷ Substitution with the S-box values placed at the respective row and column index of the S-box
    end
end
```

End

Algorithm 4 Diffusion using XOR operation process

Start

Input: $Sbox_{image}$ and Sbox.

```
[rows, cols] = size(Sbox)                                  ▷ Get the size of the original image
blockSize = 4                                              ▷ Define the block size
Calculate the number of blocks in each dimension
    numBlocksRows = rows / blockSize;
    numBlocksCols = cols / blockSize;
    for i = 1:numBlocksRows
        for j = 1:numBlocksCols
        currentBlock = Sbox_image((i-1) × blockSize + 1 : i ×blockSize, (j-1) ×blockSize + 1 : j ×blockSize); ▷ Extract the current block from the larger image
        E_image = bitxor(currentBlock, originalBlock);     ▷ Perform XOR operation with the original 4x4 block
    end
end
```

End

taken from 3D-NEAT, x_0, γ, and ρ are taken from 1DCJ, and D_0^{γ}, D^{γ}, and ξ are taken form 4DHCM. Maintaining these parameters constant during encryption, and introducing a slight alteration of 10^{-15} during decryption, results in uniformly noise like decrypted images, as illustrated in Fig. 11.

The magnitude of the key space for a single key is (10^{15}). Consequently, the overall key space reaches ($10^{15\times21}$). This value significantly exceeds the minimum threshold of 2^{100} which is reported by Alvazari in [53], indicating the robustness of the ability of the proposed encryption framework to effectively resist brute-force attacks.

5.2. Correlation analysis

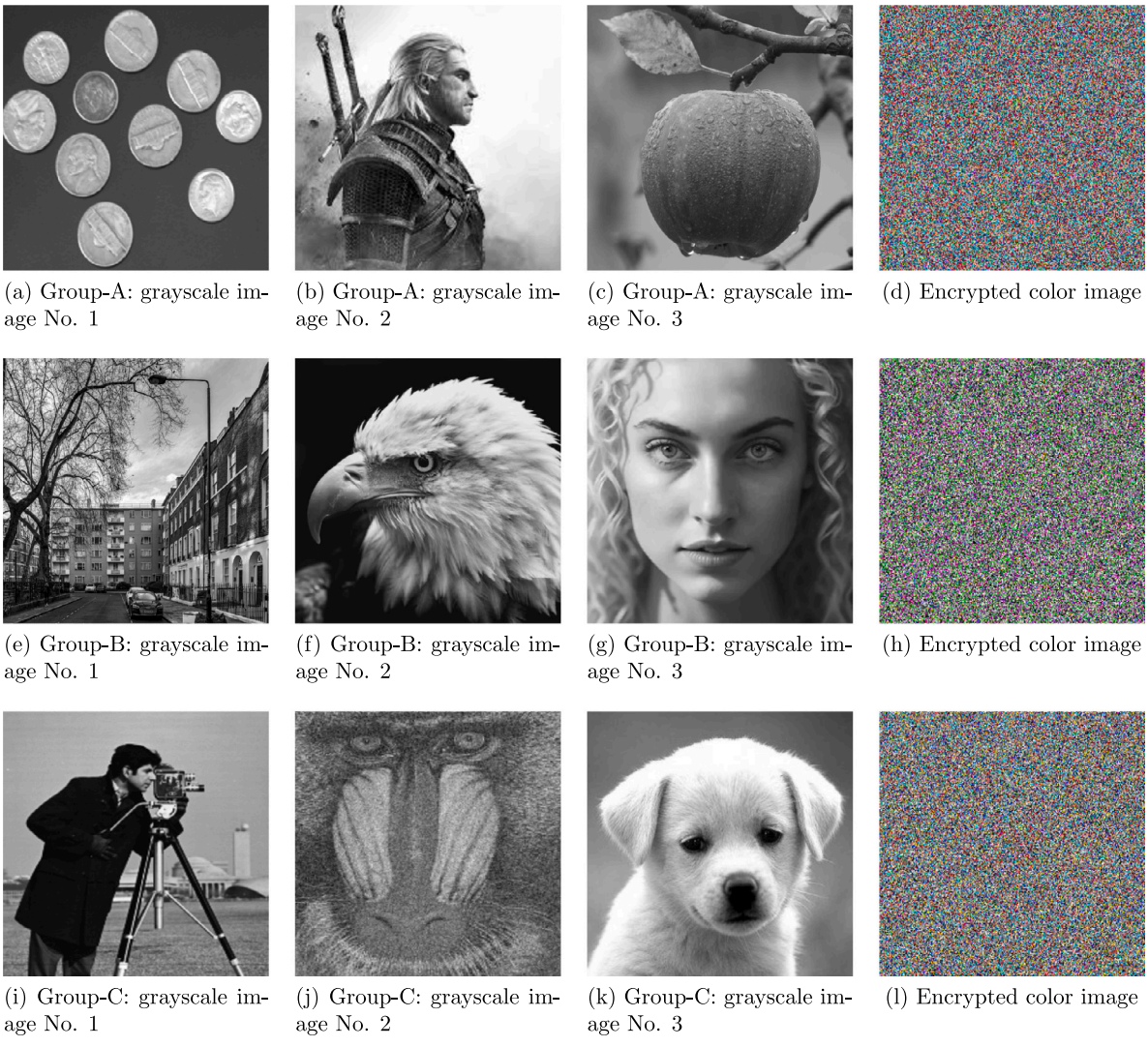
Neighbor pixel correlation refers to the degree of correlation between pixels situated in neighboring positions within an image. Ordinarily, plain images manifest a substantial degree of neighbor pixel correlation, while enciphered images should ideally exhibit correlations approaching minimum. Mathematically, the correlation is computed using Eq. (14).

$$s_{c,d} = \frac{cov(c,d)}{\sqrt{D(c)}\sqrt{D(d)}} \tag{14}$$

Where,

$$\mathrm{D(c)} = \frac{1}{M}\sum_{j=1}^{M}(c_j - C(c))^2, \quad cov(c,d) = \frac{1}{M}\sum_{j=1}^{M}(c_j - C(l))(d_j - C(d)),$$

$$\mathrm{C(c)} = \frac{1}{M}\sum_{j=1}^{M} c_j$$



(a) Group-A: grayscale image No. 1 (b) Group-A: grayscale image No. 2 (c) Group-A: grayscale image No. 3 (d) Encrypted color image

(e) Group-B: grayscale image No. 1 (f) Group-B: grayscale image No. 2 (g) Group-B: grayscale image No. 3 (h) Encrypted color image

(i) Group-C: grayscale image No. 1 (j) Group-C: grayscale image No. 2 (k) Group-C: grayscale image No. 3 (l) Encrypted color image

Fig. 9. Grayscale images and their corresponding color images.

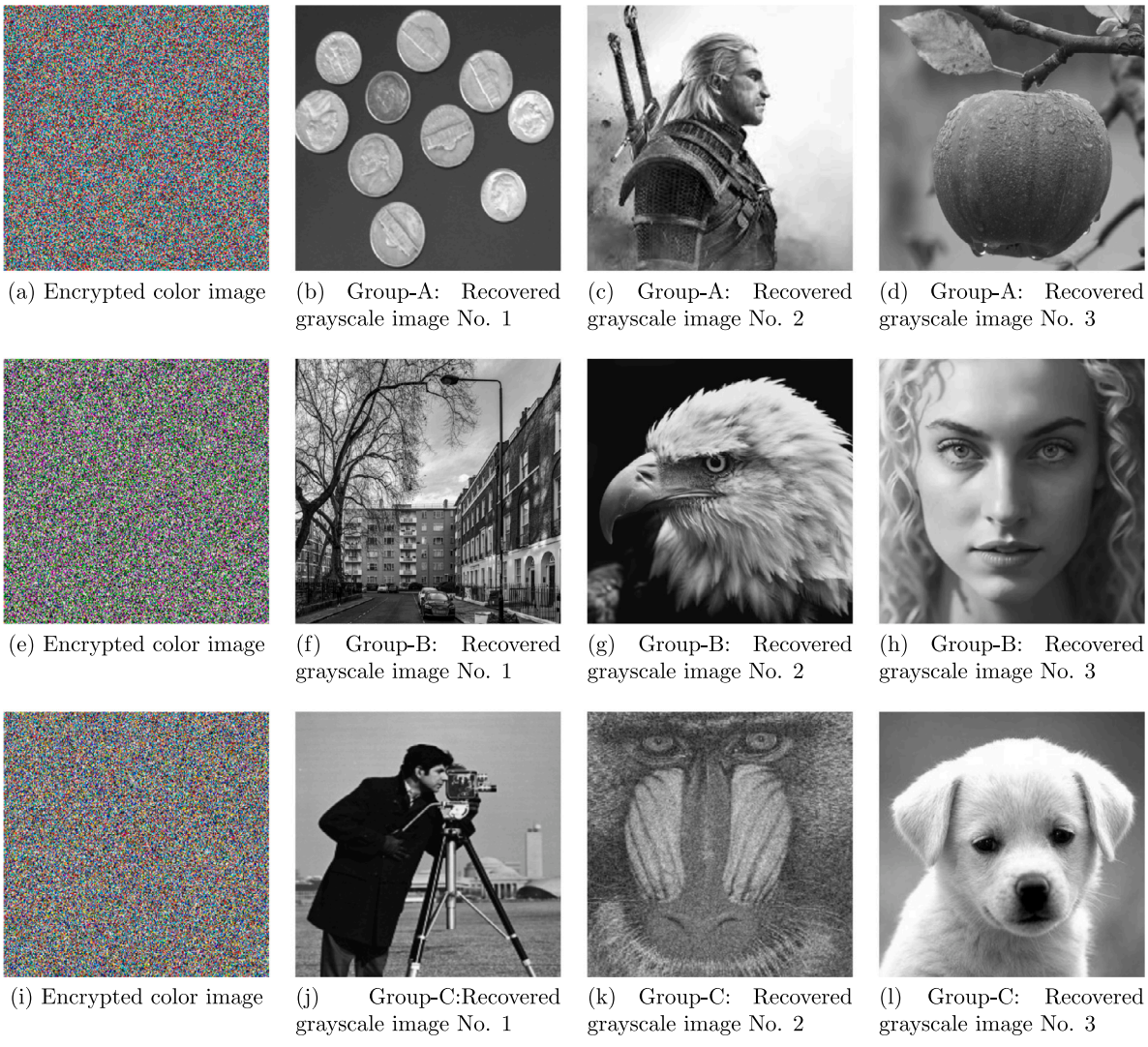
where c and d represent the grayscale values of two neighbor pixels in enciphered image.

In Fig. 12, pixel correlation in plaintext and enciphered images are illustrated in three different directions such as diagonal (D), vertical (V), and horizontal (H). The green and red dots in the figure depict the pixel distribution of the original and enciphered images, respectively. As observed in Fig. 12, the blue dots represent the plaintext images are significantly closer compared to the black dots representing the enciphered image. This signifying a lower correlation in the enciphered image pixels compared to the plaintext images.

Table 3 presents various correlation values for analyzing the similarity of the pixel values. In addition, a comparison with the existing encryption scheme is conducted on the basis of correlation values. Analysis of Table 3 reveals that the correlation values for the enciphered images produced by the proposed encryption framework are lower compared to those of the existing ones. This reveals that the proposed encryption framework has a greater capability to disrupt pixel correlation than the existing methods.

5.3. Entropy analysis

Information entropy serves as a common metric for assessing the volume of information present in an image. Large entropy values implies a greater amount of information present in the image. On the contrary, a plain text image contains more visual information but exhibits lower entropy. When encrypting an image, the entropy should be high. Statistically, it can be calculated



(a) Encrypted color image (b) Group-A: Recovered grayscale image No. 1 (c) Group-A: Recovered grayscale image No. 2 (d) Group-A: Recovered grayscale image No. 3

(e) Encrypted color image (f) Group-B: Recovered grayscale image No. 1 (g) Group-B: Recovered grayscale image No. 2 (h) Group-B: Recovered grayscale image No. 3

(i) Encrypted color image (j) Group-C:Recovered grayscale image No. 1 (k) Group-C: Recovered grayscale image No. 2 (l) Group-C: Recovered grayscale image No. 3

Fig. 10. Grayscale images and their corresponding color images.

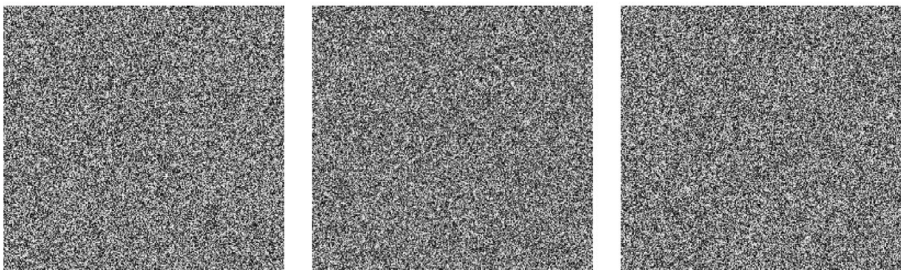
Table 3
Correlation analysis.

Encrypted images	Directions	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
Group-A: Encrypted image No. 1	H	0.0032	0.0025	0.0028	0.0031	0.0001
	V	0.0029	0.0031	−0.0331	−0.0046	−0.0013
	D	0.0033	−0.0011	−0.0035	0.0041	−0.0001
Group-A: Encrypted image No. 2	H	0.0037	−0.0025	−0.0033	−0.0053	−0.0018
	V	0.0037	−0.0055	−0.0023	−0.0023	−0.0018
	D	0.0037	−0.0055	−0.0043	−0.0053	−0.0018
Group-A: Encrypted image No. 3	H	0.0037	−0.0025	−0.0033	−0.0043	−0.0021
	V	0.0034	−0.0025	−0.0043	−0.0035	−0.0017
	D	0.0027	−0.0044	−0.0034	−0.0031	−0.0011

using Eq. (15):

$$Entropy = -\sum_{a=0}^{255} \rho(j) log_2 \rho(a) \tag{15}$$

(a) Group-A: grayscale image No. 1 (b) Group-A: grayscale image No. 2 (c) Group-A: grayscale image No. 3



(d) Group-A: Recovered image No. 1 (e) Group-A: Recovered image No. 2 (f) Group-A: Recovered image No. 3

Fig. 11. Demonstration of the sensitivity of initial keys.

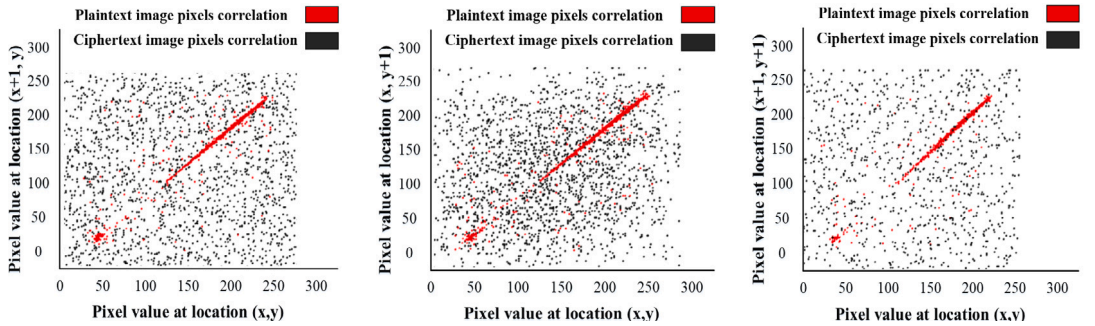
Table 4
Entropy analysis.

Groups	Encrypted images	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
Group-1	Encrypted image No. 1	7.9961	7.9863	7.9690	7.8915	7.9991
	Encrypted image No. 2	7.9832	7.9853	7.9855	7.9888	7.9990
	Encrypted image No. 3	7.9866	7.9871	7.9954	7.9920	7.9990
Group-2	Encrypted image No. 1	7.9871	7.9896	7.9898	7.9976	7.9991
	Encrypted image No. 2	7.9686	7.9756	7.9876	7.9779	7.9996
	Encrypted image No. 3	7.9886	7.9877	7.9887	7.9886	7.9993
Group-3	Encrypted image No. 1	7.9877	7.9875	7.9876	7.9876	7.9991
	Encrypted image No. 2	7.9876	7.9876	7.9946	7.9875	7.9991
	Encrypted image No. 3	7.9977	7.9849	7.9886	7.9876	7.9993
Group-4	Encrypted image No. 1	7.9876	7.9874	7.9876	7.9876	7.9993
	Encrypted image No. 2	7.9974	7.9965	7.9975	7.9872	7.9993
	Encrypted image No. 3	7.9876	7.9864	7.9971	7.9775	7.9991

In the experimentation phase, a range of grayscale images and their corresponding encipher images are employed to analyze the entropy values. The outcomes are detailed in Table 4, revealing a noteworthy increase in the entropy of the encipher image compared to the original images. The entropy approaching 7.999 in case of the proposed encryption framework, indicating close proximity to the optimal theoretical value. Furthermore, Table 4 provides a comparative overview of the entropy results of the proposed encryption framework. The finding shows the superior performance of the proposed encryption framework, signifying enhanced security compared to existing schemes.

5.4. Sensitivity analysis

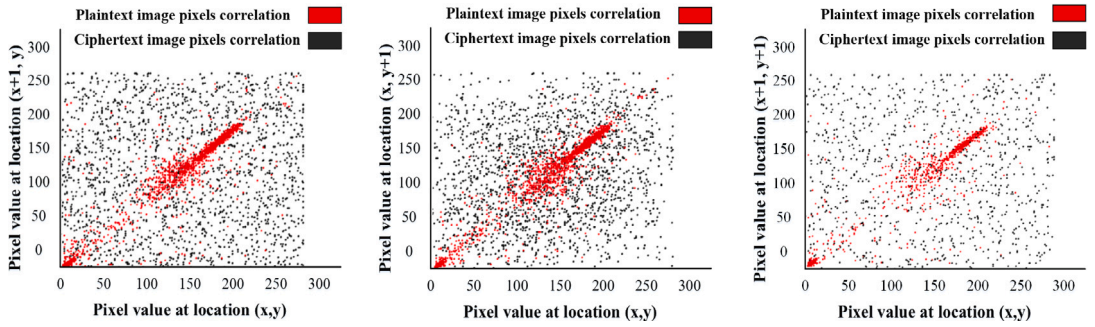
Attackers use differential attacks by making minor modifications to a plaintext image and comparing the encrypted versions before and after the change to discern relationships between them. To resists such attacks, two metrics, the number of pixels' change rate (NPCR) and unified average changing intensity (UACI), are employed. These metrics, which can be calculated using



(a) Group-A: Plaintext and encrypted image No. 1 (Horizontal correlation)

(b) Group-A: Plaintext and encrypted image No. 1 (Vertical correlation)

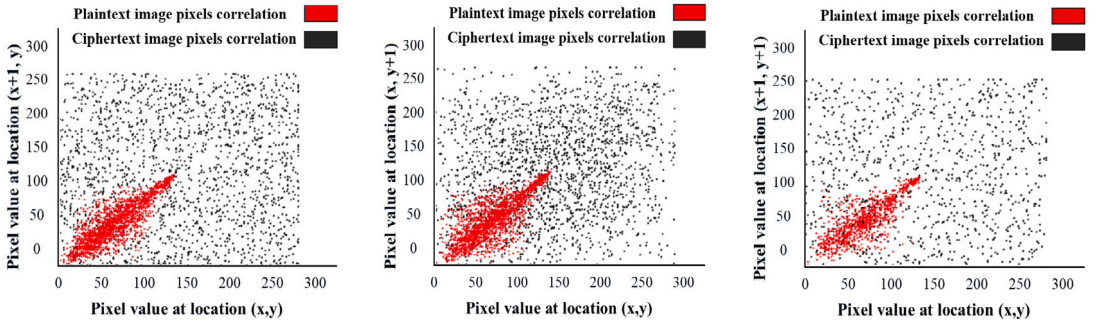
(c) Group-A: Plaintext and encrypted image No. 1 (Diagnoal correlation)



(d) Group-A: Plaintext and encrypted image No. 2 (Horizontal correlation)

(e) Group-A: Plaintext and encrypted image No. 2 (Vertical correlation)

(f) Group-A: Plaintext and encrypted image No. 2 (Diagnoal correlation)



(g) Group-A: Plaintext and encrypted image No. 3 (Horizontal correlation)

(h) Group-A: Plaintext and encrypted image No. 3 (Vertical correlation)

(i) Group-A: Plaintext and encrypted image No. 3 (Diagnoal correlation)

Fig. 12. Pixels scattered plots for the plaintext and enciphered images of group-A.

specific equations, help measure the effectiveness of encryption methods in resisting differential attacks.

$$NPCR = \frac{\sum_{t,y} D(t,y)}{F \times G} \times 100\% \tag{16}$$

$$UACI = \frac{1}{F \times G} \sum_{t,y} \frac{|C_1(t,y) - C_2(qt,y)|}{255} \times 100\% \tag{17}$$

Where, F and G represent the rows and columns of pixels in an image, while C_1 and C_2 are two ciphertext images derived from plaintext images that differ by only one pixel. The difference between C_1 and C_2 is captured in a difference matrix $D(t,y)$, where

Table 5
Sensitivity analysis.

Groups	Encrypted images	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
UACI analysis						
Group-1	Encrypted image No. 1	33.3456	33.4253	33.4687	33.53161	33.6462
	Encrypted image No. 2	33.3946	33.4938	33.4223	33.5336	33.6318
	Encrypted image No. 3	33.4541	33.5600	33.2678	33.4946	33.6638
Group-2	Encrypted image No. 1	33.5302	33.4150	33.5271	33.4336	33.5983
	Encrypted image No. 2	33.3438	33.3250	33.4697	33.4998	33.6792
	Encrypted image No. 3	33.3655	33.5491	33.3336	33.4146	33.6006
Group-3	Encrypted image No. 1	33.3008	33.4993	33.4368	33.4563	33.6133
	Encrypted image No. 2	33.3804	33.4346	33.5631	33.4668	33.6667
	Encrypted image No. 3	33.3563	33.4265	33.5610	33.4793	33.5913
Group-4	Encrypted image No. 1	33.4037	33.3789	33.4367	33.5116	33.6669
	Encrypted image No. 2	33.5237	33.4416	33.4960	33.4667	33.6109
	Encrypted image No. 3	33.3291	33.3062	33.4369	33.4963	33.6723
NPCR analysis						
Group-1	Encrypted image No. 1	99.5136	99.5468	99.5786	99.561.	99.6674
	Encrypted image No. 2	99.4697	99.5223	99.5596	99.5613	99.6199
	Encrypted image No. 3	99.5102	99.4496	99.4631	99.5632	99.6931
Group-1	Encrypted image No. 1	99.5310	99.4463	99.4368	99.4493	99.6996
	Encrypted image No. 2	99.4693	99.4339	99.4996	99.5309	99.6188
	Encrypted image No. 3	99.5569	99.4631	99.4996	99.5310	99.5961
Group-1	Encrypted image No. 1	99.5136	99.4831	99.4669	99.4863	99.6698
	Encrypted image No. 2	99.4419	99.4916	99.5413	99.5612	99.6701
	Encrypted image No. 3	99.5361	99.6178	99.4136	99.5316	99.6730
Group-1	Encrypted image No. 1	99.5196	99.4996	99.5136	99.5559	99.6649
	Encrypted image No. 2	99.4687	99.5531	99.5613	99.4687	99.6550
	Encrypted image No. 3	99.4663	99.5310	99.4669	99.4963	99.6679

$D(t, y) = 0$ if $C_1 = C_2$; otherwise, it has a non-zero value. The effectiveness of the encryption is higher when the values of UACI and NPCR.

For strong encryption, UACI and NPCR are always required high. The UACI, and NPCR, metrics for both the proposed and existing encryption schemes are presented in Table 5. The analysis of these metrics shows that the proposed encryption scheme outperforms the existing ones in terms of resistance to differential attacks.

5.5. Noise attack analysis

Images are susceptible to noise when transmitted through communication channels, which can adversely affect their quality. To secure them against noise attack, robust encryption framework become essential. To assess the efficacy of the proposed encryption framework, numerous enciphered images undergo modification by introducing random noise. Subsequently, the original images are decrypted from the noisy versions of the enciphered images to analyze the patterns of the original image in the decrypted results using the peak signal-to-noise ratio (PSNR). Mathematically, PSNR can be expressed as given in Eq. (18).

$$\begin{cases} MSE = \frac{1}{R \times S} \sum_{i=0}^{R-1} \sum_{j=0}^{R-1} \left| I(i,j) - C(i,j) \right| \\ PSNR = 20log_{10}\left(\frac{Max_I}{\sqrt{MSE}} \right) \end{cases} \tag{18}$$

$R \times S$ denotes the dimensions of the digital image, while $I(i, j)$ and $C(i, j)$ refer to the plaintext and decrypted images, respectively.

PSNR values are presented in Table 6. A greater PSNR value shows a reduced level of degradation. According to [58], a minimum PSNR value of 30 dB is recommended to mitigate noise attacks. It is evident in Table 6 that the PSNR values of the decrypted images exceed 40 dB, even when subjected to varying intensities of Salt and Pepper noise. Furthermore, as depicted in Fig. 13, the even patterns of the plaintext image remain visible in the decrypted images, despite the presence of noise intensities at 0.09, 0.10, and 0.11.

Table 6
Noise attack analysis.

Noise intensity	Groups	Encrypted images	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
0.09	Group-1	Encrypted image No. 1	30.36	30.47	31.19	35.33	45.98
		Encrypted image No. 2	32.11	34.30	62.49	36.22	45.35
		Encrypted image No. 3	41.36	41.10	35.02	39.39	46.22
0.10	Group-2	Encrypted image No. 1	39.22	39.36	41.09	41.23	45.28
		Encrypted image No. 2	39.37	35.10	38.09	39.11	45.22
		Encrypted image No. 3	39.78	38.30	41.27	42.37	47.19
0.11	Group-3	Encrypted image No. 1	40.36	39.15	38.39	39.28	46.75
		Encrypted image No. 2	39.37	37.20	39.19	41.37	49.26
		Encrypted image No. 1	37.88	38.20	39.16	39.27	49.75

(a) Group-A: Recovered image No. 1 when noise intensity level is 0.09

(b) Group-A: Recovered image No. 2 when noise intensity level is 0.10

(c) Group-A: Recovered image No. 3 when noise intensity level is 0.11

Fig. 13. Noise attack analysis.

Table 7
Clipping attack analysis.

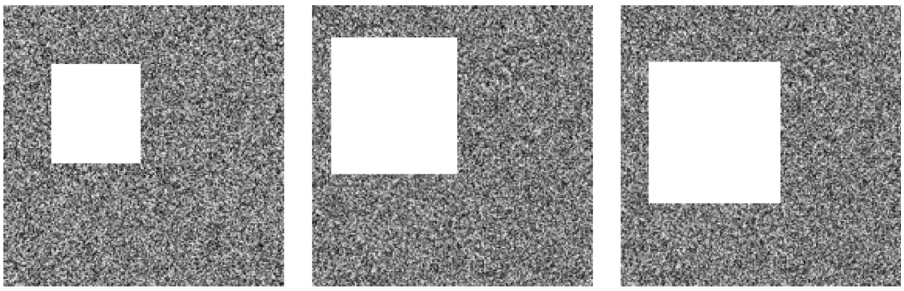
Clipping area percentage (CAP)	Groups	Encrypted images	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
8.19%	Group-1	Encrypted image No. 1	85.33	89.16	90.29	88.22	96.67
		Encrypted image No. 2	89.67	89.98	90.22	90.38	96.15
		Encrypted image No. 3	89.24	88.64	89.90	90.18	96.37
15.31%	Group-2	Encrypted image No. 1	89.37	89.15	90.38	90.37	95.15
		Encrypted image No. 2	89.19	88.46	88.28	89.79	95.01
		Encrypted image No. 3	90.20	89.33	90.21	90.41	95.76
18.26%	Group-3	Encrypted image No. 1	89.16	90.22	90.14	91.39	94.46
		Encrypted image No. 2	89.13	90.74	91.22	92.14	94.16
		Encrypted image No. 3	89.20	90.34	89.44	90.50	94.06

5.6. Clipping attack

To evaluate the resilience of the encryption algorithm against clipping attacks, where an attacker crops a section of the enciphered image to induce decryption failure, rigorous testing is performed. The quality of the recovered images is assessed by comparing them with the original image. Different percentage areas (8.19%, 15.31%, 18.26%, and 19.89%) of the enciphered image are clipped, and the plaintext image is then recovered from these clipped versions of the enciphered images. The results are detailed in Table 7. According to the findings in Table 7, more than 92% of the original information is recovered when 18.26% of the enciphered image is clipped. Furthermore, as depicted in Fig. 14, even with a clipping percentage of 18.26%, a substantial portion of the original image remains visible. This shows the robustness of the proposed encryption framework against clipping attacks.

5.7. Computational complexity analysis

In addition to its high effectiveness, it is essential for the encryption algorithm to demonstrate computational efficiency so that it can be utilized in real-time applications. To compare with existing algorithms, Table 8 presents the values of the computational



(a) Group-A: Cropped image No. 1 when CAP= 8.19

(b) Group-A: Cropped image No. 2 when CAP= 15.31

(c) Group-A: Cropped image No. 3 when CAP= 28.26

(d) Group-A: Recovered image No. 1

(e) Group-A: Recovered image No. 2

(f) Group-A: Recovered image No. 3

Fig. 14. Clipping attack analysis.

Table 8
Computational time analysis (sec)

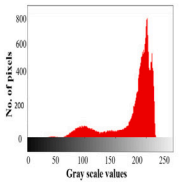
Encrypted groups (256 × 256) ×3	Ref [54]	Ref [56]	Ref [55]	Ref [57]	Proposed
Encrypted group-1	1.126	0.462	0.864	0.0002	0.0001
Encrypted group-2	0.649	0.761	0.861	0.0004	0.0005
Encrypted group-3	0.861	0.531	0.701	0.0007	0.0006

time which is required to encrypt original image of size 256 × 256. The results reveal that, except for the algorithm proposed in [57], the proposed encryption framework exhibits better performance. The encryption scheme [57] employs only confusion and diffusion processes to encrypt a color image of size 256 × 256 × 3, resulting in reduced computational time. However, due to fewer encryption operations, the overall security of the scheme proposed in [57] is not comparatively robust.

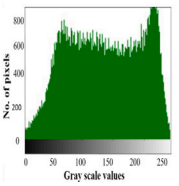
The proposed encryption framework integrates various encryption techniques, including 3D-NEAT, 1DCJ, and 4DHCM. These techniques collectively generate diffusion and confusion effects in the enciphered image. The simultaneous modification of both pixel values and positions during the 3D-NEAT and 1DCJ processes aims to decrease computational complexity within the proposed encryption scheme.

5.8. Histogram analysis

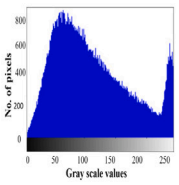
A histogram serves as a visual representation illustrating the distribution of pixels within an image. This information is especially valuable when evaluating the efficacy of encryption methods. An effective encryption technique should generate an encrypted image histogram that appears uniform and level, devoid of noticeable patterns in comparison to the original image. For example, Fig. 15 offers a contrast between the grayscale histograms and the enciphered images. The histogram of the enciphered image reveals a consistent pixel distribution, distinctly different from the histogram of the original image, which shows no discernible patterns. This suggests that the proposed encryption method is resilient against histogram cyberattacks.



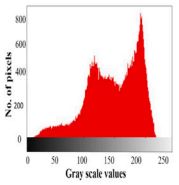
(a) Group-1: Histogram of grayscale image No. 1



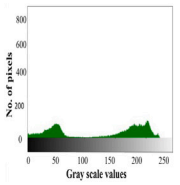
(b) Group-1: Histogram of grayscale image No. 2



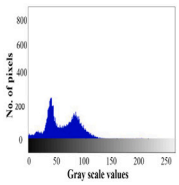
(c) Group-1: Histogram of grayscale image No. 3



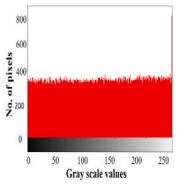
(d) Group-2: Histogram of grayscale image No. 1



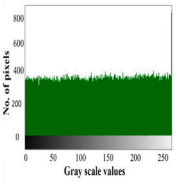
(e) Group-2: Histogram of grayscale image No. 2



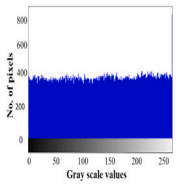
(f) Group-2: Histogram of grayscale image No. 3



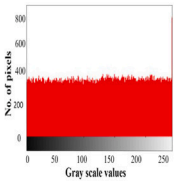
(g) Group-1: Histogram of an encrypted image No. 1



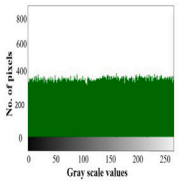
(h) Group-1: Histogram of an encrypted image No. 2



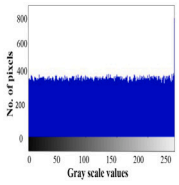
(i) Group-1: Histogram of an encrypted image No. 3



(j) Group-2: Histogram of an encrypted image No. 1



(k) Group-2: Histogram of an encrypted image No. 2



(l) Group-2: Histogram of an encrypted image No. 3

Fig. 15. Histogram analysis.

Table 9
Histogram variance.

Groups	Encrypted images	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
Group-1	Encrypted image No. 1	270.235	273.219	266.178	267.216	257.158
	Encrypted image No. 2	273.157	266.56	269.186	269.724	258.345
	Encrypted image No. 3	271.316	2.78.399	270.197	268.348	259.993
Group-2	Encrypted image No. 1	271.185	273.379	271.198	266.78	258.319
	Encrypted image No. 2	271.491	270.491	267.167	266.648	253.498
	Encrypted image No. 3	266.398	269.994	270.469	274.678	259.956
Group-3	Encrypted image No. 1	273.167	278.168	277.498	269.468	259.994
	Encrypted image No. 2	269.314	268.167	267.198	270.198	260.164
	Encrypted image No. 3	264.198	268.167	269.189	270.198	259.486
Group-4	Encrypted image No. 1	269.197	269.451	270.198	270.568	258.198
	Encrypted image No. 2	263.168	269.157	270.198	268.199	259.168
	Encrypted image No. 3	270.168	271.464	271.167	267.167	259.167

5.9. Histogram variance analysis

Variance can also be used to statistically quantify the uniformity of the histogram. Eq. (19) illustrates how mathematically it can be calculated.

$$Var(D) = \frac{1}{256}\sum_{L=1}^{256}[v_i - O(D)]^2 \tag{19}$$

where D represent the sequence of pixels, defined as $D = \{ d_1, d_2, d_3 \cdots, d_{256}\}$, d_i is the pixel value at the Lth position. The expression $O(D)$ is calculated as $\frac{1}{256}\sum_{L=1}^{256} d_i$. A low variance value signifies a higher level of robustness in the encryption algorithm. Table 9 offers a comparison of the different variance value for various encrypted images. The relatively high statistical variance in the proposed encryption framework signifies its robustness and resilience.

Table 10
Maximum deviation.

Groups	Encrypted images	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
Group-1	Encrypted image No. 1	256.156	259.168	257.119	250.102	261.236
	Encrypted image No. 2	251.019	251.165	253.167	258.197	262.134
	Encrypted image No. 3	251.167	253.189	259.498	253.168	261.124
Group-2	Encrypted image No. 1	259.167	257.189	253.046	256.167	261.891
	Encrypted image No. 2	253.198	257.168	253.190	251.164	261.236
	Encrypted image No. 3	251.468	256.198	259.100	253.222	267.198
Group-3	Encrypted image No. 1	251.0978	250.137	259.167	259.168	263.168
	Encrypted image No. 2	257.168	257.168	256.198	255.102	267.168
	Encrypted image No. 3	259.116	250.157	250.112	256.168	261.468
Group-4	Encrypted image No. 1	259.168	259.468	259.164	259.164	264.198
	Encrypted image No. 2	259.145	256.168	252.198	259.467	261.557
	Encrypted image No. 3	256.198	252.134	251.168	253.164	267.678

5.10. Maximum deviation

The effectiveness of a encryption algorithm can be evaluated by examining the variation in original and encrypted pixel values. High deviation values show the high robustness of the encryption schemes. Mathematically, it can be represented by Eq. (20).

$$D_M = \frac{Z_0 + Z_{D-1}}{2} + \sum_{L=1}^{D-2} Z_L \tag{20}$$

Where Z_L is the amplitude of the difference histogram at the index L, and D_M is the dissimilarity measure. The comparison of maximum deviations between the proposed method and existing algorithms, as shown in Table 10, indicates superior performance of the proposed encryption algorithm.

5.11. Irregular deviation

Evaluating encryption quality solely through maximum deviation (MD) is considered inadequate. An additional metric, Irregular deviation (ID), can be employed to evaluate the quality of the enciphered image by examining the statistical distribution of deviations from the original image to a uniform statistical distribution. An alternative evaluation metric, the irregular deviation (ID), can be utilized to gauge the quality of the encoded image by analyzing the statistical distribution of deviations from the original image. The mathematical form of the ID is given in Eq. (21).

$$I_D = \sum_{L=0}^{D-1} |B_M - F_G| \tag{21}$$

where B_M denotes the peak of the histogram at position L, while F_G represents the average sum of the histogram values. A reduced I_D value means an improved quality in the encrypted image. The results provided in Table 11 show the robustness of the proposed encryption framework, as ID values are lower than those of existing encryption algorithms.

6. Future work

The proposed encryption framework can be enhanced in terms of security as well as computational speed. Below are two potential solutions that can be employed in the future.

- **Multilevel discrete wavelet transform encryption (for enhancing security):** Apply multilevel DWT to the three grayscale images before the scrambling phase. This involves decomposing each image into multiple frequency sub-bands, capturing both low- and high-frequency components. After that, apply the 3D-NEAT, 1DCJ, and 4DHCFO operations separately to each sub-band. This ensures that the encryption process is applied at multiple scales, which increases its complexity and strengthens its security.
- **Selective Encryption in Frequency Domains (for enhancing computational speed):** Selectively apply encryption to specific frequency bands based on the importance of the information contained in those bands. This allows for a more adaptive and efficient encryption process, focusing computational efforts where they are most needed.
- **Strengthening security against side-channel attacks and future solutions:** Side-channel attacks, such as timing attacks or power analysis attacks, exploit unintended information leakage from the encryption process. Implementing countermeasures in real-world applications to mitigate side-channel vulnerabilities can further enhance the security of the proposed framework. Moreover, ensuring end-to-end security, including encryption of communication channels, is vital to protecting data integrity

Table 11
Irregular deviation.

Groups	Encrypted images	Ref [54]	Ref [55]	Ref [56]	Ref [57]	Proposed
Group-1	Encrypted image No. 1	46 315	47 613	46 324	46 312	45 012
	Encrypted image No. 2	46 830	46 301	46 620	465 831	45 310
	Encrypted image No. 3	463 012	46 831	46 831	46 831	45 931
Group-2	Encrypted image No. 1	46 315	49 301	46 312	47 315	45 991
	Encrypted image No. 2	46 663	47 132	46 813	46 810	45 931
	Encrypted image No. 3	46 831	46 532	46 861	45 316	45 162
Group-3	Encrypted image No. 1	46 315	47 613	47 633	47 613	453 164
	Encrypted image No. 2	46 333	46 210	46 301	47 016	453 001
	Encrypted image No. 3	46 345	47 610	46 103	46 130	45 330
Group-4	Encrypted image No. 1	46 312	46 710	46 312	47 610	45 301
	Encrypted image No. 2	46 781	46 512	47 613	46 315	45 316
	Encrypted image No. 3	46 531	46 764	47 613	4761 3	453 160

and confidentiality. Such vulnerabilities can be mitigated in the future by integrating an extra layer of security using quantum encryption into the proposed encryption framework.

- **Resource constraints:** Numerous IoT devices face constraints regarding processing power, memory, and battery life. Enhancements to the encryption framework are necessary for applications requiring exceedingly rapid encryption processing, notably achieving speeds of less than 0.00001 s.

7. Conclusion

In this research, a robust triple-image encryption framework is designed to address the escalating challenges of data security in the Industrial Internet of Things (IIoT). Unlike conventional algorithms, which encrypt one grayscale image at a time, the proposed approach generates a single color encrypted image from three grayscale input images, significantly enhancing the complexity for potential decryption attempts by unauthorized users. Leveraging the 3-D non-equilateral Arnold transform (NEAT), an extended one-dimensional chaotic jumping (1DCJ), and a four-dimensional hyperchaotic Chen map of fractional order (4DHCFO), the proposed algorithm underwent a multi-layered encryption process. The encryption process involved the application of 3D-NEAT to scramble pixel positions, incorporating diverse scrambling operations such as multi-layer permutation, multi-round permutation, and diagonal permutation. Subsequently, binary bit planes were extracted and underwent a bit-level scrambling operation using 1DCJ. The introduction of diffusion into scrambled bit planes is achieved using 4DHCFO to generate a substitution box of size 16×16 through XOR operations. The proposed encryption framework demonstrated exceptional performance, as evidenced by statistical evaluation parameters such as entropy, correlation, energy, histogram analysis, key sensitivity analysis, and key space analysis, which showed remarkable scores of 7.9999 for entropy and 0.0001 for correlation and $> 2^{200}$ for key space. Furthermore, the proposed encryption framework is computationally efficient, as evidenced by the analysis revealing a processing time of less than 1 s to encrypt triple grayscale images into a single color image. This level of efficiency makes the proposed encryption framework highly suitable for real-time applications.

Funding statement

This work was supported by Abu Dhabi University's Office of Sponsored Programs in the United Emirates under Grant 19300752.

Declaration of competing interest

The authors declare no conflicts of interests.

The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

Data availability

The data that has been used is confidential.

References

[1] Abdul-Qawy AS, Pramod P, Magesh E, Srinivasulu T. The internet of things (IoT): An overview. Int J Eng Res Appl 2015;5(12):71–82.
[2] Xu H, Wu J, Pan Q, Guan X, Guizani M. A survey on digital twin for industrial internet of things: Applications, technologies and tools. IEEE Commun Surv Tutor 2023.
[3] Rath KC, Khang A, Roy D. The role of internet of things (IoT) technology in industry 4.0 economy. In: Advanced IoT technologies and applications in the industry 4.0 digital economy. CRC Press; 2024, p. 1–28.
[4] Magaia N, Fonseca R, Muhammad K, Segundo AHFN, Neto AVL, de Albuquerque VHC. Industrial internet-of-things security enhanced with deep learning approaches for smart cities. IEEE Internet Things J 2020;8(8):6393–405.
[5] Tange K, De Donno M, Fafoutis X, Dragoni N. A systematic survey of industrial internet of things security: Requirements and fog computing opportunities. IEEE Commun Surv Tutor 2020;22(4):2489–520.
[6] Maniriho P, Ahmad T. Information hiding scheme for digital images using difference expansion and modulus function. J King Saud Univ-Comput Inf Sci 2019;31(3):335–47.
[7] Anees A, Siddiqui AM, Ahmed J, Hussain I. A technique for digital steganography using chaotic maps. Nonlinear Dynam 2014;75:807–16.
[8] Anees A, Siddiqui AM, Ahmed F. Chaotic substitution for highly autocorrelated data in encryption algorithm. Commun Nonlinear Sci Numer Simul 2014;19(9):3106–18.
[9] Shafique A, Hazzazi MM, Alharbi AR, Hussain I. Integration of spatial and frequency domain encryption for digital images. IEEE Access 2021;9:149943–54.
[10] Ahmed F, Anees A, Abbas VU, Siyal MY. A noisy channel tolerant image encryption scheme. Wirel Pers Commun 2014;77:2771–91.
[11] Rehman MU, Shafique A, Khan KH, Khalid S, Alotaibi AA, Althobaiti T, et al. Novel privacy preserving non-invasive sensing-based diagnoses of pneumonia disease leveraging deep network model. Sensors 2022;22(2):461.
[12] Alexan W, Elkandoz M, Mashaly M, Azab E, Aboshousha A. Color image encryption through chaos and kaa map. IEEE Access 2023;11:11541–54.
[13] Zhu S, Deng X, Zhang W, Zhu C. Image encryption scheme based on newly designed chaotic map and parallel DNA coding. Mathematics 2023;11(1):231.
[14] Huang X, Dong Y, Ye G, Shi Y. Meaningful image encryption algorithm based on compressive sensing and integer wavelet transform. Front Comput Sci 2023;17(3):173804.
[15] Shafique A, Ahmed F. Image encryption using dynamic S-box substitution in the wavelet domain. Wirel Pers Commun 2020;115:2243–68.
[16] Hussain I, Anees A, AlKhaldi AH, Algarni A, Aslam M. Construction of chaotic quantum magnets and matrix lorenz systems S-boxes and their applications. Chinese J Phys 2018;56(4):1609–21.
[17] Rehman MU, Shafique A, Khalid S, Hussain I. Dynamic substitution and confusion-diffusion-based noise-resistive image encryption using multiple chaotic maps. IEEE Access 2021;9:52277–91.
[18] Hussain I, Anees A, Alkhaldi AH, Aslam M, Siddiqui N, Ahmed R. Image encryption based on Chebyshev chaotic map and S8 S-boxes. Opt Appl 2019;49(2):317–30.
[19] Liu J, Wang Y, Han Q, Gao J. A sensitive image encryption algorithm based on a higher-dimensional chaotic map and steganography. Int J Bifurcation Chaos 2022;32(01):2250004.
[20] Jithin K, Sankar S. Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set. J Inf Secur Appl 2020;50:102428.
[21] Sneha P, Sankar S, Kumar AS. A chaotic colour image encryption scheme combining Walsh–Hadamard transform and Arnold–Tent maps. J Ambient Intell Humaniz Comput 2020;11:1289–308.
[22] Hussain I, Anees A, Aslam M, Ahmed R, Siddiqui N. A noise resistant symmetric key cryptosystem based on S 8 S-boxes and chaotic maps. Eur Phys J Plus 2018;133:1–23.
[23] Rehman MU. Quantum-enhanced chaotic image encryption: Strengthening digital data security with 1-D sine-based chaotic maps and quantum coding. J King Saud Univ-Comput Inf Sci 2024;101980.
[24] Rehman MU, Shafique A, Usman AB. Securing medical information transmission between IoT devices: An innovative hybrid encryption scheme based on quantum walk, DNA encoding, and chaos. Internet Things 2023;24:100891.
[25] Roselinkiruba R, Sharmila TS. Dynamic optimal pixel block selection data hiding approach using bit plane and image encryption. Int J Inf Technol 2023;15(7):3441–8.
[26] Shafique A. A new algorithm for the construction of substitution box by using chaotic map. Eur Phys J Plus 2020;135(2):194.
[27] Hussain I, Anees A, Al-Maadeed TA. A novel encryption algorithm using multiple semifield S-boxes based on permutation of symmetric group. Comput Appl Math 2023;42(2):80.
[28] Ravichandran D, Banu S A, Murthy B, Balasubramanian V, Fathima S, Amirtharajan R. An efficient medical image encryption using hybrid DNA computing and chaos in transform domain. Med Biol Eng Comput 2021;59:589–605.
[29] Hassan FS, Gutub A. Improving data hiding within colour images using hue component of HSV colour space. CAAI Trans Intell Technol 2022;7(1):56–68.
[30] Dagadu JC, Li J-P, Aboagye EO. Medical image encryption based on hybrid chaotic DNA diffusion. Wirel Pers Commun 2019;108:591–612.
[31] Zefreh EZ. An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. Multimedia Tools Appl 2020;79(33–34):24993–5022.
[32] Mansoor S, Parah SA. HAIE: A hybrid adaptive image encryption algorithm using chaos and DNA computing. Multimedia Tools Appl 2023;1–28.
[33] Maniyath SR, Thanikaiselvan V. DNA coding and RDH scheme hybrid encryption algorithm using SVM. Int J Cloud Comput 2021;10(1–2):144–57.
[34] Shafique A, Shahid J. Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps. Eur Phys J Plus 2018;133(8):331.
[35] Kumar R, Sharma AK, et al. Bit-plane based reversible data hiding in encrypted images using multi-level blocking with quad-tree. IEEE Trans Multimed 2023.
[36] Song W, Fu C, Zheng Y, Tie M, Liu J, Chen J. A parallel image encryption algorithm using intra bitplane scrambling. Math Comput Simulation 2023;204:71–88.
[37] Wu H-T, Yang Z, Cheung Y-M, Xu L, Tang S. High-capacity reversible data hiding in encrypted images by bit plane partition and MSB prediction. IEEE Access 2019;7:62361–71.
[38] Morankar G. Robust image encryption scheme for biometric finger print images using bit planes, DWT and cubic map. Int J Next-Gener Comput 2022;13(5).
[39] Pourasad Y, Ranjbarzadeh R, Mardani A. A new algorithm for digital image encryption based on chaos theory. Entropy 2021;23(3):341.
[40] Man Z, Li J, Di X, Sheng Y, Liu Z. Double image encryption algorithm based on neural network and chaos. Chaos, Solitons Fractals 2021;152:111318.
[41] Yasser I, Khalil AT, Mohamed MA, Samra AS, Khalifa F. A robust chaos-based technique for medical image encryption. IEEE Access 2021;10:244–57.
[42] Farah MB, Guesmi R, Kachouri A, Samet M. A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation. Opt Laser Technol 2020;121:105777.
[43] Alexan W, Chen Y-L, Por LY, Gabr M. Hyperchaotic maps and the single neuron model: A novel framework for chaos-based image encryption. Symmetry 2023;15(5):1081.
[44] Shao L-P, Qin Z, Gao H-J, Heng X-C. 2-dimension non equilateral image scrambling transformation. Acta Electon Sin 2007;35(7):1290.
[45] Li Y-K, Feng Q-S, Zhou F, Li Q. 2-D Arnold transformation and non-equilateral image scrambling transformation. Comput Eng Des 2009;30(13).

[46] Wu C, Tian X. 3-dimensional non-equilateral Arnold transformation and its application in image scrambling. J Comput-Aided Des Comput Graph 2010;22(10):1831–9.
[47] Liu L, Jiang D, Wang X, Rong X, Zhang R. 2D logistic-adjusted-Chebyshev map for visual color image encryption. J Inf Secur Appl 2021;60:102854.
[48] Liu J, Wang Y, Liu Z, Zhu H. A chaotic image encryption algorithm based on coupled piecewise sine map and sensitive diffusion structure. Nonlinear Dynam 2021;104(4):4615–33.
[49] Pak C, Huang L. A new color image encryption using combination of the 1D chaotic map. Signal Process 2017;138:129–37.
[50] Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27(3):379–423.
[51] Hegazi A, Matouk A. Dynamical behaviors and synchronization in the fractional order hyperchaotic Chen system. Appl Math Lett 2011;24(11):1938–44.
[52] Wen J, Xu X, Sun K, Jiang Z, Wang X. Triple-image bit-level encryption algorithm based on double cross 2D hyperchaotic map. Nonlinear Dynam 2023;111(7):6813–38.
[53] Alvarez G, Li S. Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurcation Chaos 2006;16(08):2129–51.
[54] El-Shafai W, Hemdan EE-D. Robust and efficient multi-level security framework for color medical images in telehealthcare services. J Ambient Intell Humaniz Comput 2023;14(4):3675–90.
[55] Deb S, Bhuyan B. Chaos-based medical image encryption scheme using special nonlinear filtering function based LFSR. Multimedia Tools Appl 2021;80:19803–26.
[56] Wang S, Peng Q, Du B. Chaotic color image encryption based on 4D chaotic maps and DNA sequence. Opt Laser Technol 2022;148:107753.
[57] Wang X, Zhang X, Gao M, Tian Y, Wang C, Iu HH-C. A color image encryption algorithm based on hash table, Hilbert curve and hyper-chaotic synchronization. Mathematics 2023;11(3):567.
[58] Ye G, Pan C, Dong Y, Shi Y, Huang X. Image encryption and hiding algorithm based on compressive sensing and random numbers insertion. Signal Process 2020;172:107563.