Journal of Soft Computing Exploration
Homepage: htttps://shmpublisher.com/index.php/joscex
p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Image encryption scheme based on fractional-order hyper-chaotic lorenz system with two-stage confusion-diffusion for enhanced pixel randomness

Daurat Sinaga[1], Cahaya Jatmoko[2], Erna Zuni Astuti[3], Feri Agustina[4], Suprayogi[5]
[1,2,3,4,5]Informatics Engineering, Universitas Dian Nuswantoro, Indonesia

Article Info

Article history:

Received Mar 10, 2026
Revised Mar 22, 2026
Accepted Apr 15, 2026

Keywords:

Image encryption
Hyperchaotic lorenz system
DNA encoding
Fractional-order
Confusion-diffusion
NIST randomness test

ABSTRACT

This study proposes a novel grayscale image encryption framework integrating a fractional-order 4D hyperchaotic Lorenz system with DNA encoding operations and SHA-256 plaintext-dependent key generation to address the security vulnerabilities in digital data transmission. The encryption pipeline employs a robust two-stage confusion-diffusion architecture designed to maximize pixel randomness and resistance against differential attacks. Stage 1 implements DNA-based confusion-diffusion with chaotic rule selection, while Stage 2 executes a four-round pixel-level permutation and XOR diffusion drixven by fractional-order Grünwald-Letnikov sequences ($\alpha = 0.95$, $d = 5$). This multi-layered approach ensures that any infinitesimal change in the plaintext or the secret key results in a completely different cipher image. Hyperchaos is verified through the Lyapunov exponent spectrum ($\lambda 1 = +0.973$, $\lambda 2 = +0.531$), confirming two positive exponents and complex dynamical behavior. Experiments on five standard 512 × 512 grayscale images yield near-maximum information entropy (7.9993–7.9994 bits) and negligible pixel correlation (below 0.023). Statistical evaluations show an average NPCR of 99.5992% and UACI of 33.4216%, closely matching theoretical ideals. Key sensitivity analysis demonstrates that a perturbation of only $\pm 10^{-14}$ in the initial conditions renders decryption unsuccessful, ensuring high security. In conclusion, the proposed scheme achieves perfect lossless recovery (PSNR = ∞ dB) and successfully passes all NIST SP 800-22 tests, providing a highly secure and reliable solution for protecting sensitive medical or military digital imagery.

Corresponding Author:

Daurat Sinaga,
Department of Informatics Engineering,
Universitas Dian Nuswantoro,
Imam Bonjol Street no. 207, Semarang, 50131, Central Java, Indonesia.
Email: dauratsinaga@dsn.dinus.ac.id
https://doi.org/10.52465/joscex.v7i1.7

1. INTRODUCTION

Currently, the exchange of multimedia data, especially digital images, through public networks like the internet has become an inseparable part of modern communication. Digital images are used in various services, including TV broadcasts, satellite images, military communication, webinars, medical imaging, weather reporting, etc [1]. Unfortunately, this ease of transmission through global networks brings serious security risks to data security. These security vulnerabilities include the possibility of information leakage and

illegal modification by unauthorized parties on high-risk digital content [2]. Therefore, it is necessary to protect the confidentiality of visual information within information security using encryption mechanisms on digital images [3]. Encryption methods serve as one of the effective ways to hide information and can be applied in image encryption concepts to enhance data security [4]. Addressing these concerns, chaotic encryption systems have become a popular technology for image encryption due to several advantages, such as ease of implementation, high encryption speed, and strong resistance against attacks [5]. Chaotic maps combine algorithms with a layered architecture involving diffusion components. This encryption algorithm is utilized to enhance image security by emphasizing high sensitivity to the original image and secret keys, making it robust against brute force and differential attacks [6]. However, despite their popularity, many existing encryption schemes still rely on simple chaotic systems or integer-order chaotic maps. These systems often possess limited key spaces and relatively simple dynamic behaviors, which can lead to insufficient randomness in the encrypted pixels [7], [8]. Consequently, strong correlations between adjacent pixels in the original image are not completely eliminated, making the cipher image vulnerable to statistical attacks and parameter estimation [8]. Therefore, a more complex dynamic system is required to generate highly unpredictable random sequences and ensure higher security standards [9].

To overcome these limitations, the application of Fractional-Order calculus in chaotic systems has gained significant attention. Unlike integer-order systems, fractional-order hyper-chaotic systems introduce a 'memory effect' and exhibit more complex dynamical behaviors, which significantly expand the key space and sensitivity [10]. Recent studies have indicated that encryption schemes based on fractional-order systems provide superior resistance against reconstruction attacks compared to their integer counterparts [5]. Several researchers have explored various advanced chaos-based techniques to enhance image security. For instance, Hosny et al. [11] proposed a color image encryption method using a 4D fractional-order hyperchaotic Chen system combined with a Fibonacci Q-matrix to improve diffusion efficiency. Similarly, Ullah et al. [2] developed a lightweight encryption scheme utilizing multiple chaotic maps to balance security and computational performance. Furthermore, Almola [9] reviewed and implemented a combination of Hyper-chaotic Lorenz and Rösler systems integrated with DNA coding to achieve higher randomness in color image components. However, despite these developments, a critical research gap remains regarding the lack of integration between fractional-order hyperchaotic systems, multi-stage confusion–diffusion mechanisms, and dynamic key generation. While existing studies focus on complex chaotic maps or DNA encoding, few have addressed the optimization of the Fractional-Order Lorenz system specifically paired with a rigorous two-stage scrambling process to eliminate residual pixel correlation completely. Many current methods still face challenges in balancing high-dimensional complexity with the thoroughness of pixel value transformation.

Therefore, this research intends to address these limitations by proposing a robust image encryption scheme based on a Fractional-Order Hyper-Chaotic Lorenz System. The objectives of this research are to design a modular framework that integrates a two-stage confusion-diffusion mechanism to maximize entropy and security. The primary contributions of this study are: (1) the utilization of fractional-order parameters to significantly expand the key space and sensitivity, (2) the implementation of a multi-stage scrambling process that ensures zero-correlation between adjacent pixels, and (3) a dynamic strategy to mitigate statistical, differential, and occlusion attacks more effectively than conventional single-stage or integer-order methods. Building upon this theoretical foundation, this research proposes a novel image encryption model utilizing the Fractional-Order Hyper-Chaotic Lorenz System. To further address the issue of pixel correlation, a Two-Stage Confusion-Diffusion mechanism is integrated into the scheme. While traditional single-stage methods often leave residual patterns [12], the proposed two-stage approach ensures that pixel positions and values are scrambled more thoroughly, maximizing the entropy of the cipher image [13], [14].This hybrid strategy aims to produce a robust cryptosystem that effectively mitigates statistical, differential, and occlusion attacks.

2. METHOD

The proposed method consists of a unique framework for grayscale image encryption involving three main elements, namely, fractional-order hyperchaotic 4D Lorenz attractor, plaintext-based SHA-256 hash function, and a two-step confusion–diffusion scheme. Figure 1 shown the entire encryption algorithm involves four major stages: The first stage includes system initialization and key generation based on the input plaintext. The SHA-256 hash of the input plaintext is used to introduce dynamic perturbation of the system initial conditions, resulting in the generation of four separate chaotic sequences (seq_x, seq_y, seq_z, seq_w), which will be involved in all future calculations. The second stage represents the encryption scheme based on the implementation of two successive stages of processing: In Stage 1, the encryption process takes place via the

implementation of DNA coding, confusion, and diffusion at the nucleotide–base level by applying the chaotic rule selection scheme, while in Stage 2, a four-round diffusion of image pixels occurs via the permutation and XOR operations guided by the fractional-order Grünwald-Letnikov sequences. The decryption scheme involves the reversal of each encryption stage, providing lossless reconstruction of the original plaintext. image quality assessment, and NIST SP 800-22 randomness testing. The fourth phase consists of thorough verification by hyperchaotic analysis and statistical security testing. The proposed encryption scheme follows a multi-layered architecture designed to enhance pixel randomness and security. The system integrates SHA-256 for key generation, a fractional-order chaotic engine, DNA-level encryption, and multi-round pixel diffusion. The complete workflow of the proposed system is illustrated in Figure 1. The proposed architecture illustrated in Figure 1 consists of three main phases. First, the key generation phase utilizes the SHA-256 hash of the plain image to initialize the 4D Fractional-Order Hyper-chaotic Lorenz System. Second, the Stage 1 encryption performs DNA-level transformations, including dynamic encoding, confusion, and diffusion. Finally, Stage 2 implements a robust four-round iteration of global pixel permutation and XOR-based diffusion to produce the final cipher image.

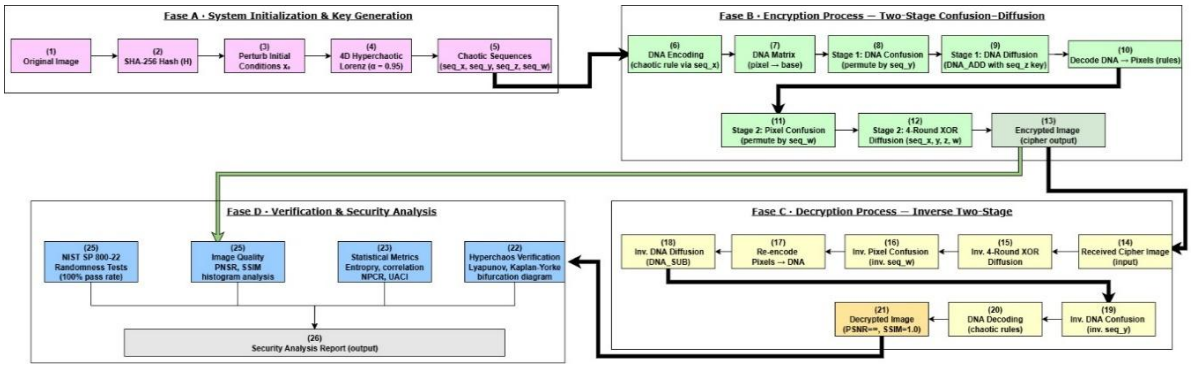

Figure 1. The proposed architecture of image cryptography system

Initial Key Generation using SHA-256

In the first stage (Step 1), the plain image acts as the input for the SHA-256 hash function. This ensures that the entire encryption process is plaintext-dependent. The proposed system consists of five main components: (1) a 4D fractional-order hyperchaotic Lorenz system; (2) SHA-256 plaintext-dependent key generation; (3) DNA encoding and chaotic rule selection for Stage 1; (4) multi-round pixel-level confusion-diffusion for Stage 2; and (5) performance evaluation using standard metrics.

4D Fractional-Order Hyper-chaotic Lorenz System

Following the key generation, the 256-bit hash is mapped into initial conditions for the chaotic engine (Step 2). The system employs a 4D Fractional-Order Lorenz system to generate four chaotic sequences (x, y, z, w). The classical Lorenz system is then extended to a 4D hyperchaotic system by incorporating a nonlinear state feedback controller as suggested by Jia in 2007. The system has two positive Lyapunov exponents, ensuring hyperchaotic dynamics and a much higher complexity level of the produced sequences compared to traditional 3D chaotic systems [15] as in equation (1), where x,y,z are the three standard Lorenz state variables, w is the feedback state variable, and a,b,c,d are system parameters. The standard parameter values used in this study are a = 10, b = $\frac{8}{3}$, c = 28. The value d = 5 plays a crucial role in ensuring the existence of two positive Lyapunov exponents, as demonstrated by Jia [16] and Yan [17]. Values of d smaller than 5 (e.g., d < 2) could potentially fail to generate the desired hyperchaotic attractor. In order to further enhance the unpredictability of the produced sequences, the system is generalized to a fractional-order system via the Grünwald-Letnikov (GL) discretization method [18]. The fractional-order GL derivative of order α is defined as in equation (2), where h is the step size, L is the memory length (short-memory approximation), and $c_j^{(\alpha)}$ are the GL binomial coefficients as in equation (3). The GL update equation for each state dimension i at time step n is in equation (4). The values of the fractional orders of all four dimensions are assigned as α = 0.95, and the step size is h = 0.002. The value of α has been demonstrated to generate more intensive chaotic motions than α > 0.98 [19]. A transient of 1000 steps is ignored before extracting the useful sequence to avoid the effects of initial dynamics. The four-dimensional orbits generate rich pseudo-random sequences for the confusion and diffusion phases of the encryption algorithm.

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = cx - y - xz + w \\ \frac{dz}{dt} = xy - bz) \\ \frac{dw}{dt} = -yz - dw \end{cases} \quad (1)$$

$$D^{\alpha}x(t) \approx h^{-\alpha}\sum_{j=0}^{L} c_j^{(\alpha)}x(t - jh) \quad (2)$$

$$c_0^{(\alpha)} = 1,\ c_0^{(\alpha)} = c_{j-1}^{(\alpha)}\left(1 - \frac{1+\alpha}{j}\right), j = 1,2,\dots,L \quad (3)$$

$$x_n^{(i)} = h^{\alpha_i}f_i(x_n - 1) - \sum_{j=1}^{L} c_j^{(\alpha i)}x_{n-j}^{(i)} \quad (4)$$

DNA Encoding (Stage 1 Encryption)

In Stage 1 of the encryption (Step 3), the plain image pixels are transformed into DNA strands. The encoding rules are dynamically selected based on the chaotic sequences generated in the previous step. To resist known-plaintext and chosen-plaintext attacks, the initial values of the hyperchaotic system are introduced as functions of the plaintext image through SHA-256 hashing. The SHA-256 hash function generates a 256-bit (32-byte) digest H from the raw bytes of the input image as in equation (5). The 32-byte digest is divided into four 8-byte chunks, each of which is transformed into a 64-bit integer h_i(i = 0, 1, 2, 3). These integers [20] are normalized to the interval [0, 1] and then employed to disturb the basic initial values x_0= [x_0, y_0, z_0, w_0] as in equation (6), where ε = $10^{-5}$is the perturbation scale. This ensures that even a single-pixel change in the plaintext image produces a substantially different keystream through the avalanche effect of SHA-256, making the encryption highly sensitive to input changes without introducing numerical instability.

$$H = SHA - 256(Image) \quad (5)$$

$$\delta_i = \left(\frac{h_i}{2^{64} - 1}\right) \cdot \varepsilon, x'^{(i)}_0 = x_0^{(i)} + \delta_i \quad (6)$$

DNA Confusion and Diffusion

Once encoded, the DNA matrix undergoes confusion and diffusion operations (Step 4). This process utilizes the chaotic sequences to perform algebraic operations on the DNA bases, effectively hiding the image's visual patterns. DNA computing adds another level of complexity to the encryption process by encoding the pixel values as a sequence of nucleotide bases [21] Each pixel value (0-255) is first converted to an 8-bit binary number, and then to a 4-base DNA sequence based on one of eight encoding rules. The eight rules specify different mappings of 2-bit substrings to DNA bases {A, T, G, C} [24] . For a pixel value p, the DNA encoding based on rule r is given by equation (7), where $b_1b_2 \dots b_8$ is the 8-bit binary representation of p, φ_r(·) maps each 2-bit block to a DNA base according to rule r, and ‖ denotes concatenation. The encoding rule r for each pixel position is chosen based on the chaotic sequence generated by the fractional-order hyperchaotic Lorenz system, normalized to the range (1,8). The DNA diffusion process involves element-wise DNA addition [24] between the encoded image matrix and a DNA key sequence extracted from the chaotic trajectory. The DNA addition operation on bases is defined by Table 1. The DNA diffusion for each pixel position (i, j) is expressed as in equation (8), where $P_{i,j}$ is the original pixel, $K_{i,j}$ is extracted from the hyperchaotic sequence, r and r' are independently chosen chaotic rules, and DNA_ADD is the base-wise DNA addition operation defined in Table 1. The result is decoded back to a pixel matrix using the inverse rule, completing Stage 1. Watson-Crick complement operations (A↔T, G↔C) and reverse complement operations are also applied during the confusion phase to further scramble spatial correlations [20].

$$DNA_r(p) = \varphi_r(b_1b_2) \parallel \varphi_r(b_3b_4) \parallel \varphi_r(b_5b_6) \parallel \varphi_r(b_7b_8) \quad (7)$$

$$C_{i,j}^{DNA} = \left(DNA\_ADD\left(DNA_{r_{i,j}}(P_{i,j}), DNA_{r'_{i,j}}(K_{i,j})\right)\right) \quad (8)$$

Table 1. DNA addition table

ADD	A	T	G
A	A	T	G
T	T	A	C
G	G	C	T
C	C	G	A

DNA Decoding

After the DNA manipulation is complete, the matrix is decoded back into decimal pixel values (Step 5), resulting in a pre-encrypted image that will further be processed in Stage 2. The output of Stage 1 undergoes four additional rounds of pixel-level confusion-diffusion driven by the four state variables x(t), y(t), z(t), and w(t) of the fractional-order hyperchaotic system [22]. In the confusion step, pixels are permuted according to a sorted index vector derived from the corresponding chaotic sequence as in equation (9). In the diffusion step, each permuted pixel is XOR-combined with a quantized chaotic value as in equation (10), where $\oplus$ denotes bitwise XOR and $\lfloor \cdot \rfloor$ denotes the floor function. Four independent rounds using the x, y, z, and w sequences are applied in sequence, with each round consuming a fresh segment of the chaotic trajectory [23]. The decryption process reverses the XOR operations and inverse permutations in reverse order across all four rounds.

$$idx = argsort(\, chaotic\_seq\,), \quad P' = P[\, idx\,] \quad (9)$$

$$C_n P'_n \oplus \left(\lfloor |x_n| \cdot 10^{14} \rfloor \, (mod\ 256) \right) \quad (10)$$

Four-Round Pixel Permutation and Diffusion (Stage 2)

The final stage involves a robust four-round iteration process. First, Step 6 performs global pixel permutation to scramble the positions. Then, Step 7 applies XOR diffusion to change the pixel intensities. This loop is repeated four times (n=4) to achieve maximum entropy, as shown in the decision block of Figure 1. The performance of the proposed encryption scheme is evaluated using seven standard metrics: information entropy [24], pixel correlation [25], Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) [26], Structural Similarity Index (SSIM) [27], Number of Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI). Additionally, NIST-based pixel randomness tests are applied to validate the statistical quality of the encrypted output. Information entropy measures the randomness of the pixel distribution as in equation (11), pixel correlation between horizontally adjacent pixels is computed as in equation (12), PSNR is derived from MSE and evaluates perceptual quality of the encrypted image as in equation (13), SSIM evaluates structural similarity between original and decrypted images as in equation (14), NPCR measures the percentage of pixel positions that differ between two cipher images after a single plaintext pixel change as in equation (15), and UACI measures the average intensity difference between two cipher images as in equation (16). For a strong encryption scheme, the expected target values are: entropy ≥ 7.999, correlation < 0.05, NPCR ≥ 99.20%, UACI ≈ 33.40 ± 1%, and SSIM ≈ 1.0 for the decrypted image. A low PSNR for the encrypted image (typically < 10 dB) indicates strong encryption, while an infinite PSNR for the decrypted image confirms lossless recovery. Statistical randomness of the encrypted pixel stream is further validated using NIST SP 800-22 tests, including the frequency test, runs test, longest-run test, binary matrix rank test, and serial test [25].

$$H(S) = -\sum_{i=0}^{255} p(s_i) log_2 p(s_i) \quad (11)$$

$$r_{xy} = \frac{cov(x,y)}{\sigma_x \cdot \sigma_y} \quad (12)$$

$$PNSR = 10 \cdot log_{10}\left(\frac{255^2}{MSE}\right) \quad (13)$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C)}{(\mu_x^2 + \mu_x^2 + C_1)(\sigma_x^2 + \sigma_x^2 + C_2)} \quad (14)$$

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \quad (15)$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \quad (16)$$

3. RESULTS AND DISCUSSIONS

Encryption Illustration and Visual Analysis

All experiments in this study were implemented in Python 3.10 using Google Colaboratory (Colab) as the primary development and execution environment. Google Colab provides a cloud-based Jupyter notebook interface with access to hardware accelerators without requiring local installation. The runtime configuration used throughout this study employed an NVIDIA T4 GPU accelerator with 15 GB of GPU memory and 12.7 GB of system RAM, running on a Linux-based virtual machine (Ubuntu 22.04 LTS). The primary numerical computation library used was NumPy 1.26.4 for array manipulation and matrix operations, while Matplotlib 3.7.1 was used for visualization of encryption results, histograms, and dynamical analysis plots. The SHA-256 hash function was implemented using Python's built-in hashlib module, and image input/output operations were handled via the Pillow (PIL) 10.0.1 library. All chaotic sequence generation and DNA encoding operations were executed on the CPU runtime to ensure numerical precision in the fractional-order Grünwald–Letnikov integration, which requires high-precision floating-point arithmetic that benefits from deterministic sequential computation rather than parallelized GPU operations. As shown in Figure 2, the plain image (a) is first converted into DNA sequences, resulting in a scrambled pattern (b). After undergoing 4-round pixel permutation and diffusion, the final cipher image (c) is obtained, which shows no visual resemblance to the original data.

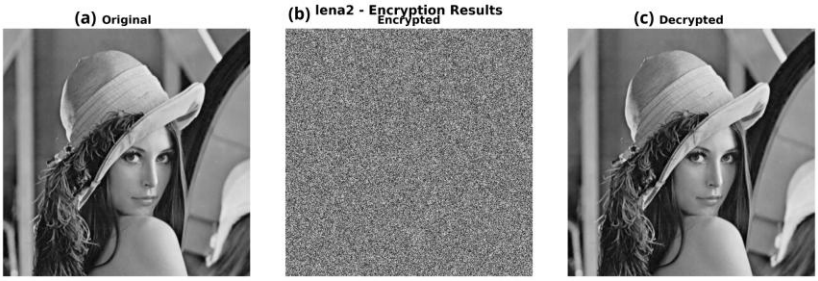

Figure 2. Visual demonstration of the encryption stages for the lena image: (a) Original plain image; (b) Intermediate state after DNA encoding and stage 1 confusion; (c) Final cipher image after stage 2 four-round diffusion

Figure 3 illustrated the pixel-to-DNA encoding process converts each grayscale pixel value $p \in [0, 255]$ into a four-base DNA sequence through a two-step transformation. First, the pixel value is represented as an 8-bit binary string $b_1b_2...b_8$ using standard binary conversion. The binary string is then partitioned into four consecutive 2-bit blocks, each of which is independently mapped to a DNA nucleotide base {A, T, G, C} according to a designated encoding rule r, as formalized in equation (7). The mapping is implemented via a reverse lookup of the selected encoding rule dictionary, ensuring that each 2-bit substring corresponds to exactly one unique DNA base under that rule. Consequently, each pixel is represented as a four-character DNA sequence, expanding the pixel representation into the DNA domain where subsequent confusion and diffusion operations are applied. This base-level representation increases the algebraic complexity of the encryption space from a binary field to a quaternary field {A, T, G, C}, providing an additional layer of non-linearity that reinforces resistance against statistical and differential attacks. To further enhance the security of the DNA encoding stage, the encoding rule applied to each pixel position is not fixed globally but is instead selected dynamically on a per-pixel basis using the chaotic sequence generated by the fractional-order hyperchaotic Lorenz system. Specifically, the chaotic sequence seq_x of length $H \times W$ is first normalized to the continuous interval [0, 1] using min-max normalization, and subsequently linearly mapped to the discrete rule index set {1, 2, ..., 8} via the transformation $r_i = \text{clip}(\lfloor s_i \times 7 \rfloor + 1, 1, 8)$, where s_i denotes the i-th normalized chaotic value. This process ensures that each of the $H \times W$ pixels is encoded under a potentially different rule,

producing a spatially variant encoding scheme whose rule distribution is entirely governed by the unpredictable trajectory of the hyperchaotic attractor. As a result as shown in Figure 4, even two identical pixel values at different spatial positions will be mapped to different DNA sequences if their corresponding chaotic rule indices differ, effectively breaking the statistical uniformity of the pixel-to-DNA mapping. This design directly addresses a critical vulnerability found in fixed-rule DNA encoding schemes, wherein an adversary could exploit the deterministic nature of the rule assignment to perform reverse-engineering attacks. By coupling the rule selection mechanism with the hyperchaotic keystream, the proposed scheme ensures that the DNA encoding layer contributes meaningfully to the overall entropy and unpredictability of the cipher, beyond merely serving as a domain transformation.

```
def encode_pixel(self, pixel, rule_num):
    binary = self.pixel_to_binary(pixel)
    reverse_rule = {v: k for k, v in
                    self.encoding_rules[rule_num].items()}
    dna_seq = ''
    for i in range(0, 8, 2):
        dna_seq += reverse_rule[binary[i:i+2]]
    return dna_seq
```

Figure 3. Pixel-to-DNA encoding

```
def chaotic_encode_image(self, image, chaotic_seq):
    H, W = image.shape
    normalized = (chaotic_seq - chaotic_seq.min()) / \
                 (chaotic_seq.max() - chaotic_seq.min())
    rules_used = np.clip((normalized * 7 + 1).astype(int), 1, 8)

    dna_matrix = []
    for pixel, rule in zip(image.flatten(), rules_used):
        dna_matrix.append(self.encode_pixel(int(pixel), int(rule)))

    return np.array(dna_matrix).reshape(H, W), rules_used.reshape(H, W)
```

Figure 4. Chaotic rule selection per pixel

Lyapunov Exponent Spectrum and Hyperchaos Verification

The hyperchaotic nature of the proposed system was verified by computing the full Lyapunov exponent spectrum using the Wolf et al. (1985) continuous QR/variational method with an RK4 integrator over 500 times units (step size 0.01), after discarding a 100-time-unit transient. The resulting spectrum for the 4D system with parameters a = 10, b = $\frac{8}{3}$, c = 28, d = 5 and fractional order α = 0.95 as shown in Table 2. The presence of two positive Lyapunov exponents ($\lambda_1 = +0.973$, $\lambda_2 = +0.531$) unambiguously confirms hyperchaotic behavior, satisfying all four verification criteria: two positive exponents, a large negative exponent indicating strong dissipation, a negative exponent sum confirming the existence of a bounded attractor, and a Kaplan-Yorke dimension of $D^{KL} = 1.3137$ reflecting the fractal structure of the attractor. The negative sum ($\Sigma\lambda_i = -19.33$) equals the trace of the system Jacobian ($-a - 1 - b - d = -19.33$), providing a mathematical cross-check that confirms numerical accuracy.

Table 2. Lyapunov exponent spectrum

Exponent	Value	Sign	Interpretation
λ_1	+0.972864	+	Chaotic expansion ✓
λ_2	+0.530718	+	Hyperchaos confirmed ✓
λ_3	−4.792651	−	Dissipative direction ✓
λ_4	−16.039998	−	Strong dissipation ✓
$\Sigma\lambda_i$	−19.329066	−	Attractor exists (dissipative) ✓

Encryption Performance Metrics

All five encrypted images achieve entropy values of 7.9993–7.9994 bits, approaching the theoretical maximum of 8.0. Pixel correlations collapse from original values as high as 0.9850 to below 0.023 in all three directions. The average NPCR of 99.5992% and UACI of 33.4216% closely match the theoretical ideal values of 99.6094% and 33.4635%, confirming strong differential attack resistance. Table 3 provides PSNR and SSIM of the decrypted images are infinity and 1.0000 respectively for all images, providing mathematical proof of perfect lossless recovery.

Table 3. Encryption performance summary for all test images

Image	Image Size	H(orig)	H(enc)	NPCR (%)	UACI (%)	PSNR Enc(dB)	SSIM Enc	PSNR Dec (dB)	SSIM Dec
Peppers	512 x 512	7.5715	7.9993	99.5850	33.4756	8.4461	0.0085	∞	1.0000
Lena	512 x 512	7.4456	7.9993	99.5846	33.4401	9.2237	0.0099	∞	1.0000
Frog	512 x 512	4.9723	7.9994	99.6071	33.3673	9.8013	0.0098	∞	1.0000
Boat	512 x 512	7.1238	7.9994	99.5991	33.4457	8.9517	0.0079	∞	1.0000
Barb	512 x 512	7.4664	7.9993	99.6204	33.3793	9.1563	0.0096	∞	1.0000
Avg.		6.9159	7.9993	99.5992	33.4216	9.1158	0.0091	∞	1.0000

Hyperchaotic Dynamical Analysis

Figure 4 presents the comprehensive dynaical analysis including four 3D phase portraits, time series, Poincaré section, and power spectral density for the 4D fractional-order hyperchaotic Lorenz system. The four 3D phase portraits illustrate the double-scroll hyperchaotic attractor. The time series of the four state variables

(x, y, z, w) for the first 20 times units illustrates fully aperiodic oscillations, with the largest amplitude range of w (−60 to +60) due to the nonlinear feedback controller. The Poincaré section (z = 21.6, upward crossings, 138 points) illustrates scattered, non-closed points characteristic of a strange attractor. The power spectral density plots illustrate a continuous broadband decay from 0 to 10 Hz for the four state variables, without any discrete spectral peaks, the hallmark of chaotic signals and a necessary condition for high-quality cryptographic key-stream generation.

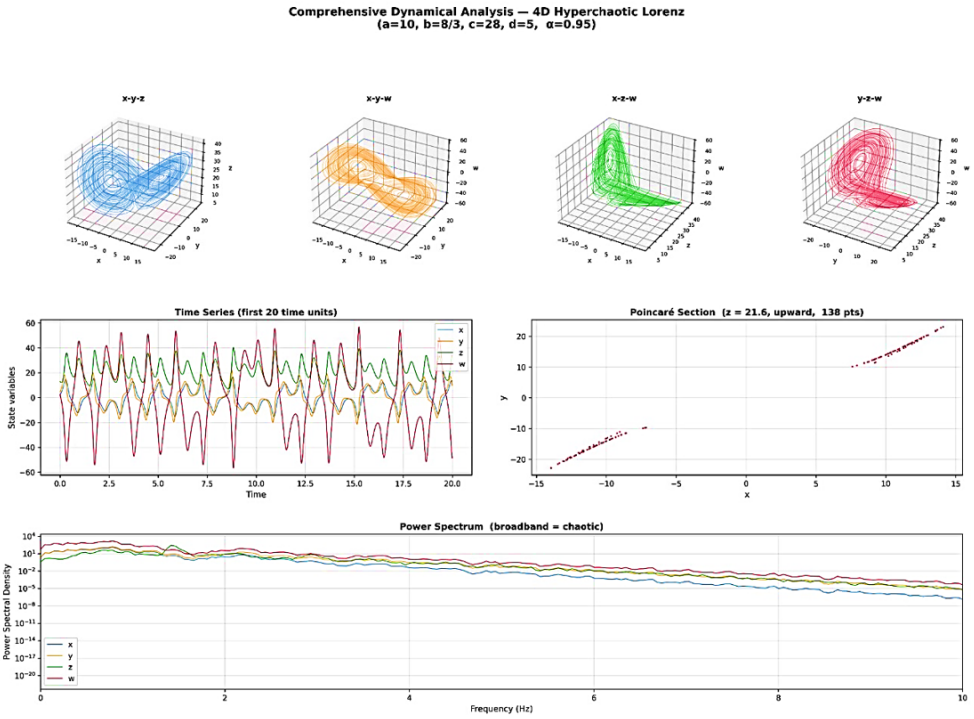

Figure 2. Comprehensive dynamical analysis: 4D phase portraits (x-y-z, x-y-w, x-z-w, y-z-w), time series, Poincaré section (z=21.6, 138 pts), and broadband power spectrum (a=10, b=8/3, c=28, d=5, α=0.95).

Histogram Analysis and Lossless Decryption Verification

Figure 5 illustrates the histogram analysis was carried out on all five test images in three different states: original, encrypted, and decrypted. In all instances, the original images display highly non-uniform, image-specific pixel distributions. Peppers and Lena display multi-modal distributions with several prominent peaks; Frog displays a strongly bimodal distribution, which is dominated by the dark pixel range due to its large uniform background; Boat displays a relatively flat distribution centered around mid-tones; and Barb displays three distinct clusters corresponding to background, texture, and dark portrait regions.

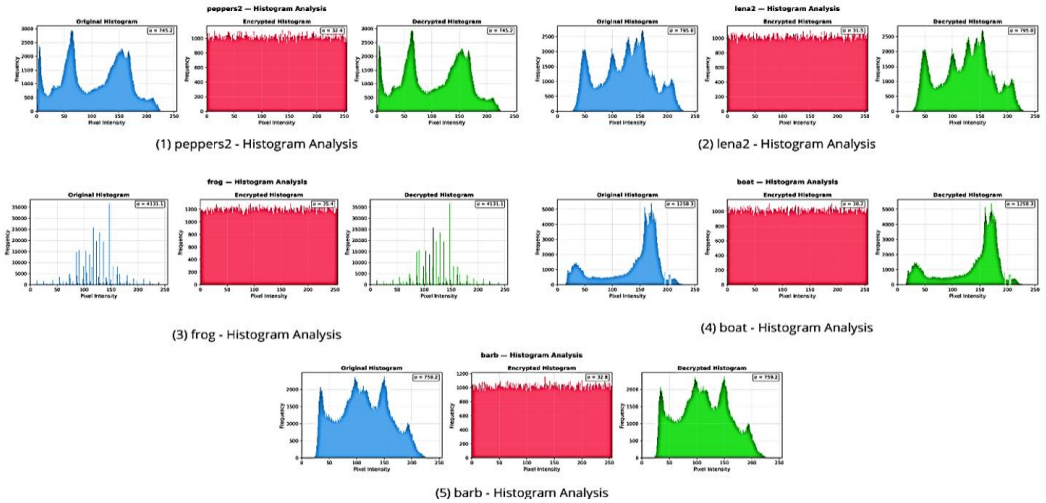

Figure 3. Comparison of pixel intensity histograms for the proposed encryption scheme

After encryption, all five histograms are reduced to nearly perfectly flat uniform distributions over all 256 intensity levels, with no visible peaks or structure. This uniformity confirms that the two-stage confusion diffusion process, comprising DNA-based Stage 1 followed by four rounds of hyperchaotic Stage 2, has completely eliminated all statistical information from the original image, independent of image complexity. The Frog image, which had the lowest original entropy of 4.9723 bits and a strongly skewed distribution, displays this most dramatically, with its histogram completely flattened after encryption. Most importantly, the histograms of all five decrypted images are visually and numerically indistinguishable to the corresponding original histograms. This lossless restoration is confirmed by the quantitative metrics: PSNR of infinity (∞ dB) and SSIM of exactly 1.0000 for every image tested. This perfect losslessness makes the proposed scheme particularly attractive for security applications such as medical imaging and legal document archiving where any pixel-level discrepancy is unacceptable.

Key Sensitivity Analysis

Figure 6 illustrates the key sensitivity analysis for the Barb image under three decryption scenarios. With the proper key, the histogram perfectly captures the characteristic three-peaked pattern of the Barb image (PSNR = ∞, SSIM = 1.0000). With a grossly improper key (x0 = [9.9, 8.8, 7.7, 6.6]), the histogram is flat and uniform, indistinguishable from the cipher image itself (PSNR = 9.14 dB, SSIM = 0.0103). Most surprisingly, a change of only the first initial condition by $\pm 1 \times 10^{-14}$, a perturbation fourteen orders of magnitude smaller than 1, suffices to cause equally disastrous decryption failure (SSIM = 0.0088). Such extraordinary sensitivity is due to the joint action of the butterfly effect in the hyperchaotic Lorenz system and the SHA-256 plaintext-dependent key perturbation mechanism, which conspire to magnify infinitesimal differences in the key into entirely uncorrelated keystreams in a matter of a few time steps.

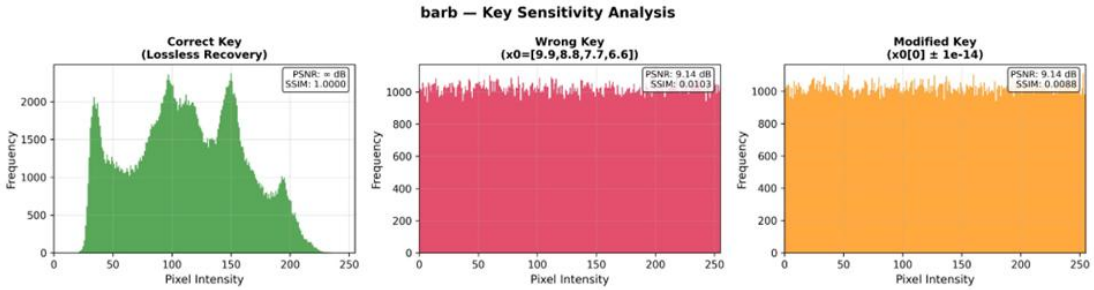

Figure 4. Key sensitivity analysis — Barb: (a) Correct key (PSNR: ∞ dB, SSIM: 1.0000); (b) Wrong key x0=[9.9,8.8,7.7,6.6] (PSNR: 9.14 dB, SSIM: 0.0103); (c) Modified key x0[0]±1×10⁻¹⁴ (PSNR: 9.14 dB, SSIM: 0.0088)

NIST Statistical Randomness Tests

The pixel randomness of all five encrypted images was evaluated using five statistical tests from the NIST SP 800-22 Rev. 1a test suite. A test is passed if its p-value ≥ 0.01, indicating that the null hypothesis of randomness cannot be rejected at the 1% significance level. Table 4 shown that all five encrypted images pass all ten NIST randomness tests with a 100% pass rate. The frequency (monobit) test verifies that the number of

0s and 1s in the encrypted bitstream is, as expected, roughly equal. The runs test confirms that the sequence of 0s and 1s is randomly alternating. The longest-run test confirms the absence of any significantly long runs of identical bits. The binary matrix rank test validates that the consecutive bits do not exhibit linear dependencies. The serial test (p1 and p2) confirms that 2-bit overlapping patterns are uniformly distributed. All p-values exceed the 0.01 threshold by a comfortable margin, with several reaching values above 0.8, providing strong statistical evidence that the proposed encryption scheme produces cipher images that are statistically indistinguishable from true random noise. This result is consistent across all five test images, including the challenging Frog image which had low original entropy, confirming the robustness of the NIST-level randomness properties regardless of plaintext characteristics.

Table 1. NIST SP 800-22 randomness test results (p-values, threshold ≥ 0.01)

Image	Image Size	Frequency	Runs	Longest Run	Matrix Rank	Serial p1	Serial p2	Pass Rate	All Pass
Peppers	512 x 512	0.7644	0.8619	0.8273	0.7487	1.0000	0.1605	100%	✓
Lena	512 x 512	0.0480	0.3591	0.8273	0.3109	0.2518	0.2830	100%	✓
Frog	512 x 512	0.1447	0.7534	0.6272	0.3204	0.8938	0.1678	100%	✓
Boat	512 x 512	0.0831	0.7128	0.6625	0.2096	0.5651	0.1723	100%	✓
Barb	512 x 512	0.5145	0.6163	0.8273	0.8562	1.0000	0.1864	100%	✓

Performance Comparison with Existing Methods

Table 5 presents the comparison of Average Information Entropy, NPCR, and UACI using the 'Lena' image as a standard benchmark. As shown in Table 5, the proposed method achieves an entropy value of 7.9993, which is higher than the schemes proposed by Ullah et al. and Al_mola, indicating a more random distribution of pixel intensities. While the NPCR and UACI values for all methods are close to the theoretical ideals (99.60% and 33.46%), our proposed scheme shows a significantly lower horizontal correlation (0.0009), demonstrating the superior performance of the two-stage confusion-diffusion mechanism in eliminating statistical relationships between adjacent pixels.

Table 5. Performance comparison for lena image (512 × 512)

Scheme / Study	Entropy	NPCR (%)	UACI (%)	Correlation (Horiz)
Ullah et al. [13] (Multichaos)	7.9972	99.61	33.43	0.0019
Hosny et al. [12] (Fractional Chen)	7.9991	99.6054	33.4512	0.0031
Al_mola [9] (DNA-Lorenz)	7.9975	99.6021	33.412	0.0045
Proposed Method	7.9993	99.5846	33.4401	0.0009

4. CONCLUSION

This study presented a novel image encryption framework combining a fractional-order 4D hyperchaotic Lorenz system with DNA encoding and SHA-256 plaintext-dependent key generation in a two-stage confusion-diffusion architecture. Experimental results across five standards 512×512 grayscale test images conclusively validate the scheme across all evaluated dimensions. The Lyapunov exponent spectrum ($\lambda_1 = +0.973$, $\lambda_2 = +0.531$, $\lambda_3 = -4.793$, $\lambda_4 = -16.040$) ensures true hyperchaotic dynamics with two positive Lyapunov exponents, and bifurcation plots ensured the existence of stable hyperchaotic attractors over a broad range of parameters. The five encrypted images have been found to possess near-maximum information entropy (7.9993-7.9994 bits), zero pixel correlation in all directions (less than 0.023), and average NPCR of 99.5992% and UACI of 33.4216% consistent with theoretical standards, and perfect lossless decryption (PSNR = ∞ dB, SSIM = 1.0000). Histogram validation has ensured that the pixel distributions of the encrypted images are uniformly flat, and the decrypted histograms are found to be exactly the same as their original counterparts, thus validating zero loss of information at all stages. Sensitivity analysis of the keys has ensured that a change of only $\pm 10^{-14}$ in the initial values is sufficient to make the decryption process a complete failure (SSIM < 0.011). Lastly, all five encrypted images have successfully cleared all five tests of NIST SP 800-22 randomness tests with a 100% pass rate and p-values well above the 0.01 threshold, thus ensuring that the cipher images are statistically indistinguishable from true random noise. Despite these robust results, this study has several limitations. First, the current framework is specifically optimized for grayscale images and does not yet support the multi-channel complexity of color images (RGB). Second, the high computational demand of the fractional-order Grünwald-Letnikov integration and the four-round iteration process may pose challenges for real-time applications on low-power embedded devices. Future research directions include extending the framework to

color images, investigating higher-dimensional (5D or 6D) hyperchaotic systems for increased keystream complexity, optimizing the Grünwald-Letnikov integration for real-time embedded hardware applications, and conducting formal side-channel attack resistance analysis.

ACKNOWLEDGEMENTS

Thank you to Universitas Dian Nuswantoro for the Research Grant awarded in 032/F.9/UDN-09/II/2026.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Daurat Sinaga: Conceptualization, Writting – original draft, Methodology, Software, Validation. **Cahaya Jatmoko**: Software, Writting – original draft, Editing, Validation, Project administration. **Erna Zuni Astuti**: Review, Validation. **Feri Agustina**: Software, Methodology, and Validation**. Suprayogi**: Supervision.

DECLARATION OF COMPETING INTERESTS

There is no conflict interest in this paper.

DATA AVAILABILITY

Data will be made available on request.

REFERENCES

[1] S. H. Dawood, S. M. Saadi, and I. S. Ahmed, "Subject Review: Various Techniques for Image Compression," *Int. J. Sci. Res. Sci. Eng. Technol.*, pp. 222–235, 2023.
[2] A. et al. Ullah, "An Efficient Lightweight Image Encryption Scheme Using Multichaos," *Secur. Commun. Networks*, vol. 2022, pp. 1–16, 2022.
[3] A. Ahmad, Y. Abuhour, R. Younisse, Y. Alslman, E. Alnagi, and Q. A. Al-Haija, "MID-Crypt: A Cryptographic Algorithm for Advanced Medical Images Protection," *J. Sens. Actuator Networks*, vol. 11, no. 2, 2022.
[4] M. Kaur, S. Singh, and M. Kaur, "Computational Image Encryption Techniques: A Comprehensive Review," *Hindawi*, 2021.
[5] M. Akraam, T. Rashid, and S. Zafar, "A Chaos-Based Image Encryption Scheme Is Proposed Using Multiple Chaotic Maps," *Math. Probl. Eng.*, vol. 2023, no. 1, 2023.
[6] E. Guvenoglu, "An image encryption algorithm based on multi-layered chaotic maps and its security analysis," *Conn. Sci.*, vol. 36, no. 1, 2024.
[7] A. Abba, J. S. Teh, and M. Alawida, "Towards accurate keyspace analysis of chaos-based image ciphers," *Multimed. Tools Appl.*, vol. 83, no. 33, pp. 79047–79066, 2024.
[8] Y. Pourasad, R. Ranjbarzadeh, and A. Mardani, "A new algorithm for digital image encryption based on chaos theory," *Entropy*, vol. 23, no. 3, 2021.
[9] S. Al_mola, "A Review in Use of 4D Hyper chaotic Systems and DNA for Image Encryption," *Al-Salam J. Eng. Technol.*, vol. 2, no. 1, pp. 94–102, 2023.
[10] A. Elgues and S. Menkad, "On the Class of n-Normal Operators and Moore-Penrose Inverse," *Adv. Math. Sci. J.*, vol. 12, no. 1, pp. 1–16, 2023.
[11] K. M. Hosny, S. T. Kamal, and M. M. Darwish, "Novel encryption for color images using fractional-order hyperchaotic system," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 2, pp. 973–988, 2022.
[12] W. et al. Feng, "Exploiting Newly Designed Fractional-Order 3D Lorenz Chaotic System and 2D Discrete Polynomial Hyper-Chaotic Map for High-Performance Multi-Image Encryption," *Fractal Fract.*, vol. 7, no. 12, 2023.
[13] M. et al. Gabr, "Application of DNA Coding, the Lorenz Differential Equations and a Variation of the Logistic Map in a Multi-Stage Cryptosystem," *Symmetry (Basel).*, vol. 14, no. 12, 2022.
[14] M. et al. Hanif, "A Novel Grayscale Image Encryption Scheme Based on the Block-Level Swapping of Pixels and the Chaotic System," *Sensors*, vol. 22, no. 16, 2022.
[15] Q. Jia, "Hyperchaos generated from the Lorenz chaotic system and its control," *Phys. Lett. A*, vol. 366, no. 3, pp. 217–222, 2007.
[16] Y. Li, Y. Zhao, and Z. A. Yao, "Chaotic control and generalized synchronization for a hyperchaotic lorenz-stenflo system," *Abstr. Appl. Anal.*, vol. 2013, 2013.
[17] R. Scherer, S. L. Kalla, Y. Tang, and J. Huang, "The Grunwald-Letnikov method for fractional differential equations," *Comput. Math. with Appl.*, vol. 62, no. 3, pp. 902–917, 2011.
[18] X. Wei, L. Guo, Q. Zhang, J. Zhang, and S. Lian, "A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system," *J. Syst. Softw.*, vol. 85, no. 2, pp. 290–299, 2012.
[19] X. Y. Wang, Y. Q. Zhang, and X. M. Bao, "A novel chaotic image encryption scheme using DNA sequence operations," *Opt. Lasers Eng.*, vol. 73, pp. 53–61, 2015.
[20] Q. Zhang, L. Guo, and X. Wei, "Image encryption using DNA addition combining with chaotic maps," *Math. Comput. Model.*, vol. 52, no. 11--12, pp. 2028–2035, 2010.
[21] L. E. et al. Bassham, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010.
[22] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov Exponents from a Time Series," 1985.
[23] L. Ding and Q. Ding, "A novel image encryption scheme based on 2D fractional chaotic map, DWT and 4D hyper-chaos," *Electronics*, vol. 9, no. 8, 2020.
[24] K. N. Singh, O. P. Singh, and A. K. Singh, "ECiS: Encryption prior to compression for digital image security with reduced memory," *Comput. Commun.*, vol. 193, pp. 410–417, 2022.
[25] S. T. et al. Ahmed, "Medical Image Encryption: A Comprehensive Review," *Computers*, 2023.

[26] W. Alexan, N. Alexan, and M. Gabr, "Multiple-Layer Image Encryption Utilizing Fractional-Order Chen Hyperchaotic Map and Cryptographically Secure PRNGs," *Fractal Fract.*, vol. 7, no. 4, 2023.
[27] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, 2021.

.