entropy

MDPI

Article

A Quantum Memristor-Modulated Hopfield Chaotic System for Secure and Real-Time Image Encryption in IoT Edge Devices

Mehmet Sagbas [1], Muhammet Emin Sahin [2], Esra Güngör Ulutas [3] and Baris Karakaya [4,*]

[1] Department of Electrical and Electronics Engineering, Izmir Bakircay University, Izmir 35660, Türkiye; mehmet.sagbas@bakircay.edu.tr
[2] Department of Computer Engineering, Izmir Bakircay University, Izmir 35660, Türkiye; emin.sahin@bakircay.edu.tr
[3] Department of Computer Programming, Sorgun Vocational School, Yozgat Bozok University, Yozgat 66100, Türkiye; esra.ulutas@bozok.edu.tr
[4] Department of Electrical and Electronics Engineering, Faculty of Engineering, Fırat University, Elazig 23200, Türkiye
* Correspondence: bkarakaya@firat.edu.tr

Abstract

This paper presents a hybrid image encryption framework based on a differential photonic quantum memristor-modulated Hopfield neural network (D-PQM–HNN). The four-dimensional D-PQM–HNN exhibits rich and complex chaotic dynamics with a borderline hyperchaotic tendency. The network is used as the entropy source of the framework, and the bitstream generated from the network passes the National Institute of Standards and Technology (NIST) SP 800-22 statistical test suite at the adopted significance level. The bitstream drives a multilayer encryption pipeline comprising bidirectional cumulative XOR diffusion, permutation-based confusion, an extended 16-symbol DNA-domain layer with rule-selected mixing, multi-point crossover and mutation, and a terminal Advanced Encryption Standard-256-in-counter-mode (AES-256-CTR) stage. The complete algorithm is implemented on an NVIDIA Jetson AGX Orin using channel-wise Compute Unified Device Architecture (CUDA) parallelization. For 512 × 512 color images, the implementation reduces the total encryption time from approximately 1.5 s on the reference desktop CPU to 0.22–0.24 s on the embedded platform. The experiments on four benchmark images include histogram and chi-square analyses, adjacent-pixel correlation, number of pixels change rate/unified average changing intensity (NPCR/UACI), information entropy, key sensitivity, and robustness against noise and data loss. The results show near-uniform cipher image statistics, high plaintext and key sensitivity, and recoverable image information under the tested corruption conditions, while demonstrating the feasibility of the proposed framework for edge-oriented image encryption.

Keywords: chaotic image encryption; DNA computing; embedded systems; Hopfield neural network; Internet of Things (IoT); quantum memristor

Check for updates

Academic Editors: Xiaoqiang Zhang, Lin Teng and Umar Hayat

Received: 28 October 2026
Revised: 8 November 2026
Accepted: 16 November 2026
Published: 18 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Digital images are nowadays a major part of the sensitive data that is transmitted over medical, industrial and communication networks. Since images have a very large volume of data and a high statistical correlation between neighboring pixels, their substantial redundancy and strong inter-pixel correlation have motivated encryption methods that explicitly consider image-specific statistical characteristics [1]. Chaotic systems are very popular for designing encryption systems due to their extreme sensitivity to initial

conditions and their deterministic but unpredictable trajectories, which allow simple governing equations to produce long, non-repeating and highly variable keystreams. The concept of using chaotic dynamics for cryptographic encryption design was introduced with Matthews' proposal of an encryption algorithm based on chaos [2] and, thereafter, by Fridrich's work on symmetric ciphers based on two-dimensional chaotic maps, which set the confusion–diffusion architecture that still forms the foundation of most modern chaos-based image cryptosystems [3], and by Josephus scrambling-based dynamic cross-diffusion [4]. Due to the known vulnerabilities of some two-dimensional and low-order chaotic maps to phase-space reconstruction and parameter estimation attacks, recent research has increasingly considered higher-dimensional and physically grounded chaotic sources that are capable of generating richer nonlinear dynamics. Notably, the chaotic dynamics exhibited by neural activities have drawn significant interest due to their capacity to generate complex behaviors [5–8]. Persistent attention has been paid to memristor-based chaotic circuits because of their inherent nonlinearity and history-dependent conductance, which makes the realization of hyperchaotic and even multistable behaviors achievable in compact memristive circuits. Recent studies include memristor-induced hyperchaos in fractional-order Hopfield neural networks for combined image encryption and field-programmable gate array (FPGA) implementation [9], multistable memristor-coupled Hopfield networks with Arnold mapping-based diagonal diffusion [10], logistic map-driven memristive Hopfield networks for military image encryption [11], complex-valued discrete-time Hopfield neural networks [12], and memristor-based hyperchaotic circuits validated for statistical randomness in signal encryption applications [8]. Beyond chaotic signal generation, discrete memristive spiking neural networks built on a memristor model with a bounded internal state have been shown to support spike transmission, synchronization in ternary and ring-coupled topologies, and pattern recognition [13]. This boundedness of the memristive state variable is also relevant to cryptographic use, since numerically stable, non-divergent state evolution is a prerequisite for generating long keystreams.

These works have demonstrated that memristive and neuron-inspired chaotic systems can provide richer nonlinear dynamics and more complex attractor geometries in comparison with classical low-dimensional chaotic maps. Another line of research has examined photonic and quantum mechanical analogues of memristive dynamics as entropy source of an encryption system. Quantum memristors extend the concept of history-dependent memristive behavior to quantum observables, while preserving quantum-coherent information processing [14]. Photonic realizations based on tunable Mach–Zehnder interferometers with measurement-and-feedback loops have shown the capability to reproduce this history-dependent response with standard integrated photonics components [15]. Its feasibility at the single photon level was then demonstrated experimentally [16] and the internal interferometric phase has since been used directly as a circuit-oriented state variable showing frequency-dependent hysteresis [17]. Based on this photonic measurement–feedback framework, a differential photonic quantum memristor (D-PQM) and a D-PQM-modulated Hopfield neural network (D-PQM-HNN) were proposed as a four-dimensional chaotic core with a positive leading Lyapunov exponent and a borderline-hyperchaotic Kaplan–Yorke dimension [18]. In the present study, this system is adopted as a chaotic core, as it combines a physically motivated quantum-inspired memristive mechanism with the rich nonlinear dynamics that have been shown to be advantageous for cryptographic keystream generation by memristor-based circuits.

In recent years, DNA-based encryption studies have become increasingly prevalent in the literature [19–22]. Regardless of the choice of the chaotic core, DNA computing has emerged as a complementary confusion mechanism, which translates binary data into a symbolic domain either before or after chaotic diffusion, thereby introducing an addi-

tional keyed symbolic-domain transformation layer. In the authors' own previous work, a locally active memristor-based chaotic circuit was combined with DNA-domain diffusion–confusion operations for real-time medical image encryption on a low-cost PYNQ-Z1 FPGA platform [23]. The latest extension of this line of work is a hybrid framework of memristive chaos, an extended 16-symbol DNA alphabet, an Add–Rotate–XOR (ARX) mixing layer and Triple Data Encryption Standard (3DES), with real-time implementation on both an NVIDIA Jetson Nano and a PYNQ-Z1 [24]. Similar hybrid designs in the recent literature are chaos-based block permutation schemes with weighted bit-plane chain diffusion [25], and FPGA-oriented high-dimensional memristive hyperchaotic multiple-image encryption architectures [26]. A complementary direction is visually meaningful image encryption, which addresses the concern that noise-like cipher images may themselves draw an attacker's attention during transmission. A recent IoT-oriented scheme of this kind generates chaotic measurement matrices from a two-dimensional hyperchaotic map; refines them through singular value decomposition for two-dimensional compressive sensing; embeds the encrypted measurements into a carrier image; and tunes its parameters through multiobjective optimization balancing reconstruction quality, entropy and differential attack resistance [27]. Together, these works indicate a growing interest in multilayer pipelines that couple a chaotic or memristive source with symbolic DNA-domain processing and hardware-oriented cryptographic implementation. However, the reviewed literature does not address the specific integration of a D-PQM–HNN chaotic source with an extended 16-symbol DNA-domain layer, a terminal AES-256-CTR stage, and embedded GPU-based parallel implementation. The main contributions of this paper are summarized as follows:

- D-PQM–HNN is used as the chaotic source for image encryption, and the resulting bitstream is evaluated using the full NIST SP 800-22 statistical test suite.
- The chaotic diffusion/confusion is combined with an extended 16-symbol DNA-domain layer (mixing, crossover, and mutation) and a terminal AES-256-CTR stage, thereby retaining AES-256-CTR as an independently established cryptographic output stage.
- The full pipeline is deployed on an NVIDIA Jetson AGX Orin via CUDA-stream parallelism, reducing the per-image encryption time from ~1.5 s (desktop CPU) to ~0.22 s, demonstrating its feasibility for edge and Internet of Things (IoT)-oriented image encryption.
- The scheme is evaluated using histogram and chi-square analyses, correlation, NPCR/UACI, entropy, key sensitivity and key space analyses, and noise/data loss robustness tests on four benchmark images.

Distinction from previous chaos–DNA schemes: Relative to the authors' earlier schemes [23,24], the present work differs in four respects, each with a specific advantage. (i) Algorithm design: The pipeline begins with a bit-level bidirectional cumulative –XOR diffusion followed by a chaos-seeded full-length bit permutation, so that a single-bit change in the plain image propagates through the entire bitstream in both directions before symbolic processing. Its DNA layer then selects the mixing operation (XOR, modulo-16 addition or modulo-16 subtraction) per symbol according to the Key-DNA and combines it with multi-point crossover and key-derived mutation. Whereas [23] relies on fixed DNA-domain diffusion–confusion and [24] on an ARX mixing layer, here the operation applied to each symbol is itself key-dependent, which increases the nonlinearity of the symbolic stage. (ii) Encryption mechanism: The terminal cipher is AES-256 in counter mode keyed through a domain-separated SHA-256 derivation, instead of the 3DES stage of [24]. (iii) Embedded implementation: The complete eleven-stage pipeline is parallelized per channel with CUDA streams on a Jetson AGX Orin GPU, reducing the encryption time for a 512 × 512 color image to 0.22–0.24 s, whereas [23] targets an FPGA and [24] the Jetson Nano and PYNQ-Z1 platforms. (iv) Chaotic source: The D-PQM–HNN introduced in [18] for federated learning

is employed for the first time as the keystream generator of an image encryption architecture, and its bitstream is validated with the full NIST SP 800-22 suite.

The remainder of this paper is organized as follows. Section 2 presents the materials and methods, including the design and dynamical characteristics of the D-PQM–HNN chaotic system, the randomness validation of the generated chaotic sequence, the proposed image encryption algorithm, the embedded system implementation, and the security analysis and dataset. Section 3 presents the experimental results and evaluates the performance of the proposed encryption scheme through statistical, security, robustness, and computational analyses. Section 4 discusses the obtained results, highlighting the security properties and computational efficiency of the proposed framework. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Material and Methods

2.1. Design of Chaotic System

The differential photonic quantum memristor (D-PQM) and the D-PQM-modulated Hopfield neural network (D-PQM–HNN) adopted here as the chaotic core were introduced and comprehensively analyzed in [18]. Therefore, only the relations required for the present encryption framework are summarized in this section. Accordingly, the D-PQM–HNN is used here as an established nonlinear chaotic source rather than as a new dynamical model.

Differential Photonic Quantum Memristor: Quantum memristors extend history-dependent memristive dynamics to quantum observables while retaining quantum-coherent information processing [14]. In the photonic domain, the standard PQM architecture employs a tunable Mach–Zehnder interferometer (MZI), measurement, and classical feedback to realize a history-dependent optical response [15]. The practical feasibility of this architecture was demonstrated in integrated photonics at the single-photon level [16]. More recently, the internal interferometric phase was also employed directly as the state variable in circuit-oriented PQM models exhibiting frequency-dependent hysteresis [17].

Building on this photonic measurement–feedback framework, the D-PQM introduced in [18] jointly processes the two complementary MZI outputs to obtain a centered bipolar response. For an ideal 50:50 MZI, the complementary photon-number expectations and the centered D-PQM variables are

$$\begin{aligned} P_{1,2}(t) &= \frac{\langle n_{\text{in}}(t)\rangle}{2}[1 \pm \cos\theta(t)], \\ x_{\text{PQM}}(t) &= 2\left(\langle n_{\text{in}}(t)\rangle - \frac{1}{2}\right), \\ y_{\text{PQM}}(t) &= x_{\text{PQM}}(t)\cos\theta(t), \\ \dot{\theta}(t) &= \frac{\alpha}{2}x_{\text{PQM}}(t). \end{aligned} \tag{1}$$

where the upper and lower signs correspond to P_1 and P_2, respectively; θ is the internal phase; and $\alpha > 0$ is the update gain. Unlike conventional single-port PQM readouts, the differential configuration suppresses common-mode fluctuations, while centering maps the photon-number input onto $[-1, 1]$ [18]. Hence, $W(\theta) = \cos\theta$ acts as a bounded bipolar memory weight. In the present framework, this bounded bipolar response provides a state-dependent synaptic modulation within the Hopfield network. The D-PQM therefore introduces an additional memory-dependent nonlinear state into the chaotic core rather than providing cryptographic security through its quantum origin alone. Classical memristive or nonlinear chaotic sources may also generate complex dynamics; the D-PQM is adopted here because its differential bipolar response provides a compact and physically motivated

mechanism for modulating the $y \to x$ synaptic pathway. Under the periodic excitation $\langle n_{in} \rangle = \sin^2(\omega t)$, the model exhibits origin-pinched hysteresis loops whose enclosed area decreases as the excitation frequency increases, as shown in Figure 1.

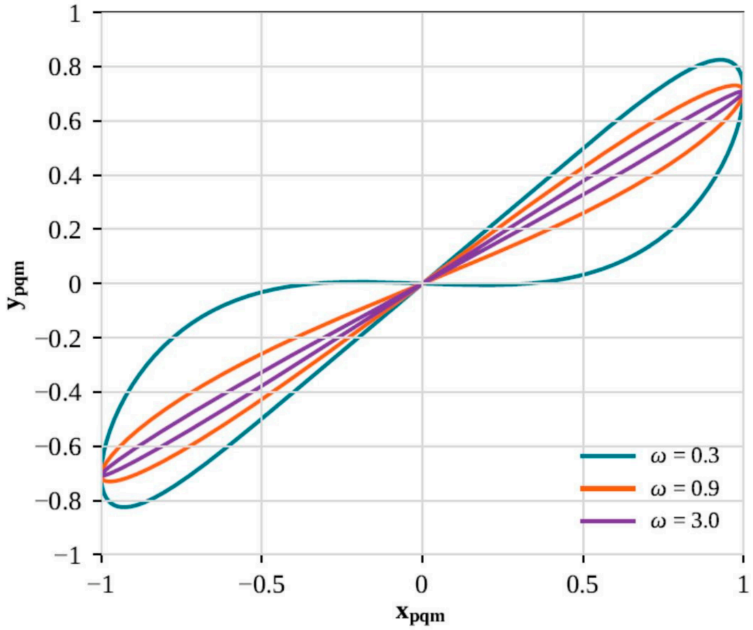


Figure 1. Frequency-dependent pinched hysteresis loops of the D-PQM for $\theta(0) = \pi/4$, $\alpha = 1$, and $\omega \in \{0.3, 0.9, 3.0\}$.

D-PQM-Modulated Hopfield Network: With $\phi(u) = \tanh(u)$ and the state vector $\mathbf{X} = [x, y, z, \theta]^T$, the four-dimensional D-PQM–HNN is described by

$$\begin{aligned} \dot{x} &= -x + w_{11}\phi(x) - k\cos\theta\,\phi(y) + w_{13}\phi(z), \\ \dot{y} &= -y + w_{21}\phi(x) + w_{22}\phi(y) + w_{23}\phi(z), \\ \dot{z} &= -z + w_{31}\phi(x) + w_{33}\phi(z), \\ \dot{\theta} &= \frac{\alpha}{2}\phi(y), \end{aligned} \tag{2}$$

where the $y \to x$ branch is the only adaptive synaptic pathway. The nominal parameter set is $w_{11} = 2.5$, $w_{13} = 0.6$, $w_{21} = 2$, $w_{22} = 1.6$, $w_{23} = 1.15$, $w_{31} = -5$, $w_{33} = -1.2$, $k = 1.5$, and $\alpha = 0.05$, with the initial condition $X(0) = [0.1, 0.1, 0.1, 0.1]^T$. k denotes the D-PQM coupling strength.

The D-PQM hysteresis loops in Figure 1 were calculated using 8000 uniformly spaced samples per excitation period, with the internal phase evaluated by cumulative trapezoidal integration. The D-PQM–HNN simulations in Figure 2 were performed in MATLAB 2025a using double-precision arithmetic and fixed-step fourth-order Runge–Kutta (RK4) integration. For the nominal attractor in Figure 2a, a step size of h = 0.005 was used, with the first 700 time units discarded as transient and the subsequent 500 time units recorded. The bifurcation diagram in Figure 2b was obtained over k ∈ [1.2,2.2] using h = 0.01, with 300 transient and 350 recording time units. The Lyapunov exponents in Figure 2c were calculated using the Benettin–QR method, with the state and full variational equations integrated simultaneously by RK4 at (h = 0.005). After a 300-time-unit transient, the exponents were accumulated over 800 time units, with QR re-orthonormalization every ten integration steps.

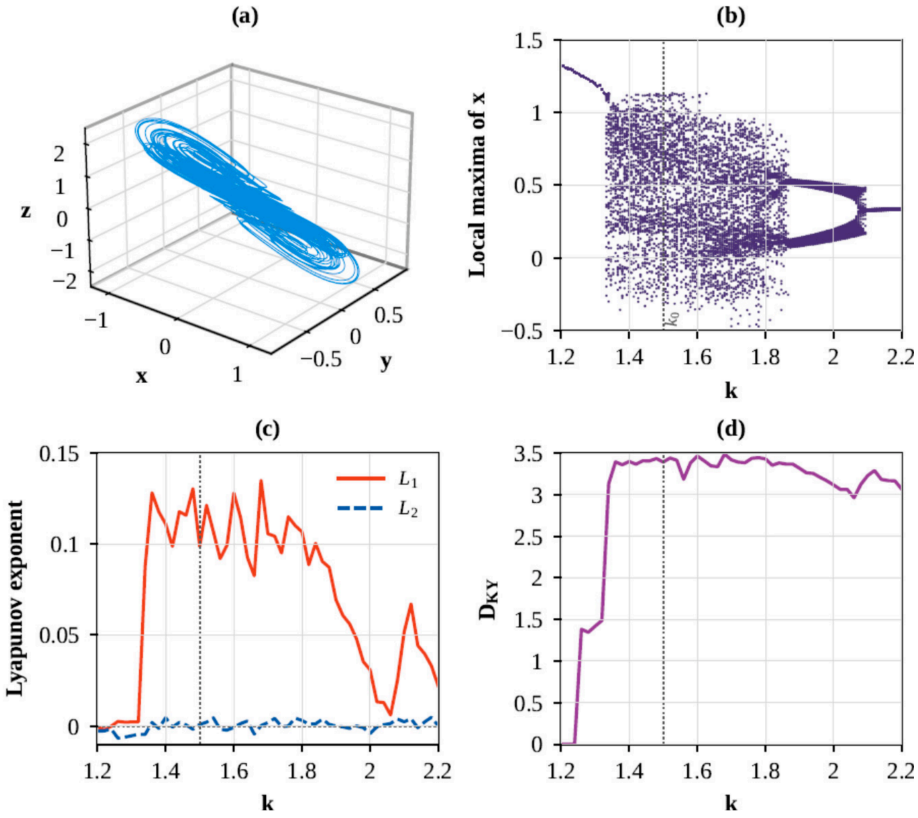


Figure 2. Compact dynamical characterization of the D-PQM–HNN: (**a**) 3D x–y–z attractor at $k = 1.5$; (**b**) local maxima of x versus k; (**c**) leading Lyapunov exponents L_1 and L_2; and (**d**) Kaplan–Yorke dimension.

The system possesses the trivial equilibrium manifold $\mathcal{E}_0 = \{(0, 0, 0, \theta_0) : \theta_0 \in \mathbb{R}\}$. At the nominal coupling $k = 1.5$ and $\theta_0 = 0$, the transverse eigenvalues are 0.6625 and $-0.3812 \pm j1.6247$, together with the neutral zero eigenvalue associated with the phase direction; therefore, the equilibrium is unstable. A long-run Benettin–QR analysis at the nominal operating point yields the sorted Lyapunov spectrum $(0.1148, 0.0047, -0.0010, -0.2596)$ [18]. Its negative sum, $\sum_{i=1}^{4} L_i = -0.1411$, indicates trajectory-averaged phase-space volume contraction. The corresponding Kaplan–Yorke dimension is $D_{\mathrm{KY}} = 3.4562$. Since the second exponent is only slightly positive and may vary with numerical settings, the observed regime is conservatively classified as complex chaotic, with a borderline hyperchaotic tendency rather than robust hyperchaos [18].

Figure 2 summarizes the dynamical behavior of the system through the three-dimensional x–y–z projection of the 4D attractor, together with its planar shadow projections, the one-parameter bifurcation pattern, the finite-time leading Lyapunov exponents, and the corresponding Kaplan–Yorke dimension. After transient removal, the resulting aperiodic state samples are used to construct the encryption keystream in the next section.

2.2. Randomness Validation of the Chaotic Sequence

The NIST SP 800-22 statistical test suite, a widely used reference battery for evaluating cryptographic random and pseudorandom number generators, was used to assess the chaotic bit sequence driving the diffusion, confusion, and DNA-domain stages of the proposed algorithm [28]. Before the test, each floating-point chaotic value was converted to a binary stream by using the bit extraction method described for chaotic map-based pixel shuffling and diffusion in [29]. This method multiplies the floating-point value by 10^{14} to shift its most sensitive decimal digits to the integer part, then takes the floor of the result

and reduces it modulo 256 to obtain a single byte, which is then expanded to its binary representation consisting of 8 bits. The p-values obtained from all 15 core NIST tests were above the α=0.01 significance threshold, so the null hypothesis of randomness could not be rejected for any of the tests. Table 1 shows the NIST SP 800-22 results for the generated chaotic sequence, which passed all tests under the adopted experimental conditions without additional post-processing. For the two tests that returned one p-value per state, Table 1 lists the smallest p-value. The encryption and decryption stages operate only on the resulting integer bitstream, which is generated once and shared as key material; floating-point differences between devices can therefore affect the generation of the bitstream but not the encryption or decryption itself.

Table 1. NIST SP 800-22 statistical test suite results for the chaotic bit sequence used by the proposed algorithm.

Statistical Test	p-Value	Conclusion
Frequency (Monobit) Test	0.9649	Random
Frequency Test within a Block	0.1092	Random
Runs Test	0.4330	Random
Test for the Longest Run of Ones in a Block	0.2271	Random
Binary Matrix Rank Test	0.4898	Random
Discrete Fourier Transform (Spectral) Test	0.4141	Random
Non-overlapping Template Matching Test	0.7054	Random
Overlapping Template Matching Test	0.7721	Random
Maurer's "Universal Statistical" Test	0.0788	Random
Linear Complexity Test	0.9682	Random
Serial Test (p-value 1/p-value 2)	0.8610/0.9065	Random
Approximate Entropy Test	0.8285	Random
Cumulative Sums Test (Forward)	0.8049	Random
Cumulative Sums Test (Backward)	0.8436	Random
Random Excursions Test (8 states, −4...+4; smallest p-value)	0.3798	Random
Random Excursions Variant Test (18 states, −9...+9; smallest p-value)	0.1104	Random

2.3. Design of Encryption System

The proposed algorithm is a hybrid cryptosystem that combines three complementary paradigms into a single sequential pipeline: (i) chaos-driven diffusion and confusion, using an externally generated chaotic bit sequence to reduce the statistical structure of the plain image; (ii) DNA computing and genetic algorithm-inspired operations (DNA encoding, DNA-domain mixing, crossover and mutation), projecting the binary data into a symbolic space to introduce an additional key-dependent transformation layer; and (iii) classical symmetric block encryption (AES-256 in CTR mode), applied as the final stage of the pipeline. These three layers are not independent options but are built on top of each other: the chaotic layer randomizes the raw bitstream, the DNA/genetic layer re-mixes this data on a symbolic level, and AES-256 encrypts the resulting intermediate representation. The algorithm operates on single-channel 8-bit pixel arrays. For color images, the B, G and R channels are processed separately by the same pipeline, using different, non-overlapping parts of the chaotic sequence, and are recombined only after encryption.

The encryption process is a linear eleven-step chain: (1) channel separation, (2) bit-plane decomposition, (3) diffusion, (4) confusion, (5) DNA encoding, (6) DNA-domain mixing, (7) multi-point crossover, (8) mutation, (9) DNA decoding, (10) Secure Hash Algorithm 256 (SHA-256)-based key derivation followed by AES-256-CTR encryption, and (11) formation of the cypher image. The decryption direction is the mathematical inverse of these eleven stages in the reverse order.

Input representation: The raw image is read out, scaled to a fixed resolution, and split into three single-channel byte arrays (B, G, and R). Each array is then unpacked at the bit level (MSB→LSB) to form a one-dimensional bitstream of length H × W × 8. All

further diffusion, confusion, and DNA operations are carried out on the bitstream rather than directly on the bytes or pixels.

DNA encoding: An extended 16-symbol DNA-inspired alphabet (A, C, G, T, E, F, H, I, J, K, L, M, N, O, P, and Q) is used, where each symbol encodes exactly 4 bits ($2^4 = 16$), as given in Table 2, rather than the classical 2-bit, 4-letter DNA alphabet. This mapping is applied to the diffused/confused image bitstream (resulting in an "Image-DNA" sequence) and to the chaotic key bitstream (resulting in a same-length "Key-DNA" sequence). The 16-symbol alphabet is a bijective re-coding of each 4-bit group, so it causes no data expansion and makes modulo-16 addition and subtraction possible, which, selected per symbol together with XOR by the Key-DNA, are nonlinear over GF(2) and thus break the GF(2) linearity of the preceding XOR diffusion and permutation stages. At the same time, it allows the crossover and mutation to operate at the granularity of 4-bit symbols, where one mutation event changes a whole symbol by a key-derived modular offset.

Table 2. Extended 16-symbol DNA encoding table.

4-bit	DNA Symbol	4-bit	DNA Symbol	4-bit	DNA Symbol	4-bit Group	DNA Symbol
0000	A	0100	E	1000	J	1100	N
0001	C	0101	F	1001	K	1101	O
0010	G	0110	H	1010	L	1110	P
0011	T	0111	I	1011	M	1111	Q

Diffusion: The bidirectional cumulative XOR chain is designed to propagate local bit changes throughout the sequence: a forward pass computes each output bit as the XOR of the chaotic bit, the corresponding plain bit and the previous output bit; a second pass repeats this on the reversed sequence to ensure propagation in both directions. Its inverse reconstructs each plain bit from the two neighboring diffused bits and the corresponding chaotic bit.

Confusion: A 64-bit permuted congruential generator (PCG64) seeded from the chaotic sequence generates a full permutation of all N-bit positions, which is applied using fancy indexing to the diffused stream. Both parties create the same permutation from the same chaotic sequence; therefore, it is not necessary to send or save the permutation table. The inverse permutation restores the original bit order.

Mixing of DNA domains: Conventional DNA representations use the four nucleobases Adenine (A), Cytosine (C), Thymine (T), and Guanine (G) [30]. In the proposed extended symbolic representation, the Image-DNA and Key-DNA sequences are combined symbol-by-symbol with one of three operators selected at each position according to (key_index mod 3): XOR (remainder 0), modulo-16 addition (remainder 1), or modulo-16 subtraction (remainder 2). The selection rule depends only on the Key-DNA value, which can be independently recomputed at the receiver. This rule-dependent operation avoids relying on a single XOR operation throughout the DNA-domain mixing stage.

Multi-point crossover: Using genetic algorithms as a model, crossover is applied within the mixed Image-DNA sequence. A chaos-seeded generator picks 3–7 cut points that divide the sequence into segments, and a key-derived permutation reorders these segments. The cut points and the segment order are derived from the chaotic bitstream, so the receiver regenerates them and restores the original order with the inverse permutation of the segments; the crossover therefore needs no transmitted data, and the Key-DNA is not modified by this stage.

Mutation: A chaotic number of mutation events (≈4000–6500) are applied to random positions of the crossover output; each event adds a key-derived non-zero offset (1–15)

modulo 16 to the existing symbol. The positions and offsets are regenerated from the chaotic bitstream, so decryption reverses the mutation exactly by subtracting the offsets in reverse order, even if a position was mutated multiple times.

AES-256-CTR and key derivation: Deriving the AES key directly from DNA symbol bytes would cap its entropy at 4 bits/byte (16 possible values per symbol), yielding only ~128 bits of true entropy for a nominally 256-bit key. To avoid this, the whole Key-DNA material is hashed twice with SHA-256 with domain-separation suffixes "|AES-KEY" and "|AES-NONCE". This procedure produces a 256-bit AES key representation and a separate 128-bit nonce derived from the available key material; it does not increase the entropy of the underlying source. The KDF input is the Key-DNA, which the receiver regenerates from the shared chaotic bitstream. Because the nonce is derived from the same key material, the chaotic key (initial conditions) must be changed for every encrypted image; otherwise, the AES-CTR keystream would be reused. The DNA-domain intermediate result is then encrypted with AES-256 in CTR mode, which does not require padding, can accept arbitrary length data, restricts the effect of local corruption to the corresponding position (no error propagation within the AES stage, unlike CBC; the inverse bidirectional diffusion that follows turns every isolated bit error into two bit errors, see Section 3.6) and is trivially parallelizable. AES-256-CTR is purposely the last step in the pipeline, so the ciphertext inherits the combined guarantees of the custom chaos/DNA transform layer and a proven, independently analyzed block cypher.

Decryption: Decryption performs the eleven stages in reverse order and needs only the same chaotic bitstream used during encryption, from which the permutation; the key-derivation function (KDF) key/nonce; the number and positions of the cut points and their segment order; and the mutation count, positions and offsets are all re-derived. No side information is transmitted with the cipher image.

Channel-wise processing: The 11-stage pipeline (sections above) is performed fully independently for each of the B, G and R channels, each using a disjoint portion of the chaotic sequence, so that all intermediate artefacts (permutation, key/nonce, cut points and segment order, mutation positions and offsets) are different across channels. After encryption, the three cipher channels are recombined to form the encrypted color image. During decryption, each channel is processed independently through the corresponding inverse pipeline and then recombined to reconstruct the color image.

Figure 3 illustrates the overview of the proposed system. Table 3 summarizes the complete workflow of the proposed cryptosystem, presenting the encryption and decryption pipelines in parts (a) and (b), respectively. The decryption process reverses the encryption stages in the corresponding inverse order to accurately reconstruct the original image.

2.4. Embedded System

The proposed pipeline is suitable for deployment on embedded edge platforms, such as the NVIDIA Jetson AGX Orin. The three per-channel sub-pipelines (B, G, and R) are independent of each other and can therefore be assigned to different CUDA streams. The confusion (permutation), DNA-domain mixing and mutation stages are themselves embarrassingly parallel over bit/symbol positions. This workload profile maps naturally to the Orin's Ampere architecture GPU, which provides up to 2048 CUDA cores and up to 275 TOPS of AI/parallel compute in a configurable 15–60 W power envelope, built on the NVIDIA Ampere architecture and offering flexible power configuration for energy-efficient or maximum-performance operation [31]. These characteristics make the platform suitable for evaluating the latency and computational feasibility of the proposed image encryption pipeline in embedded edge scenarios.

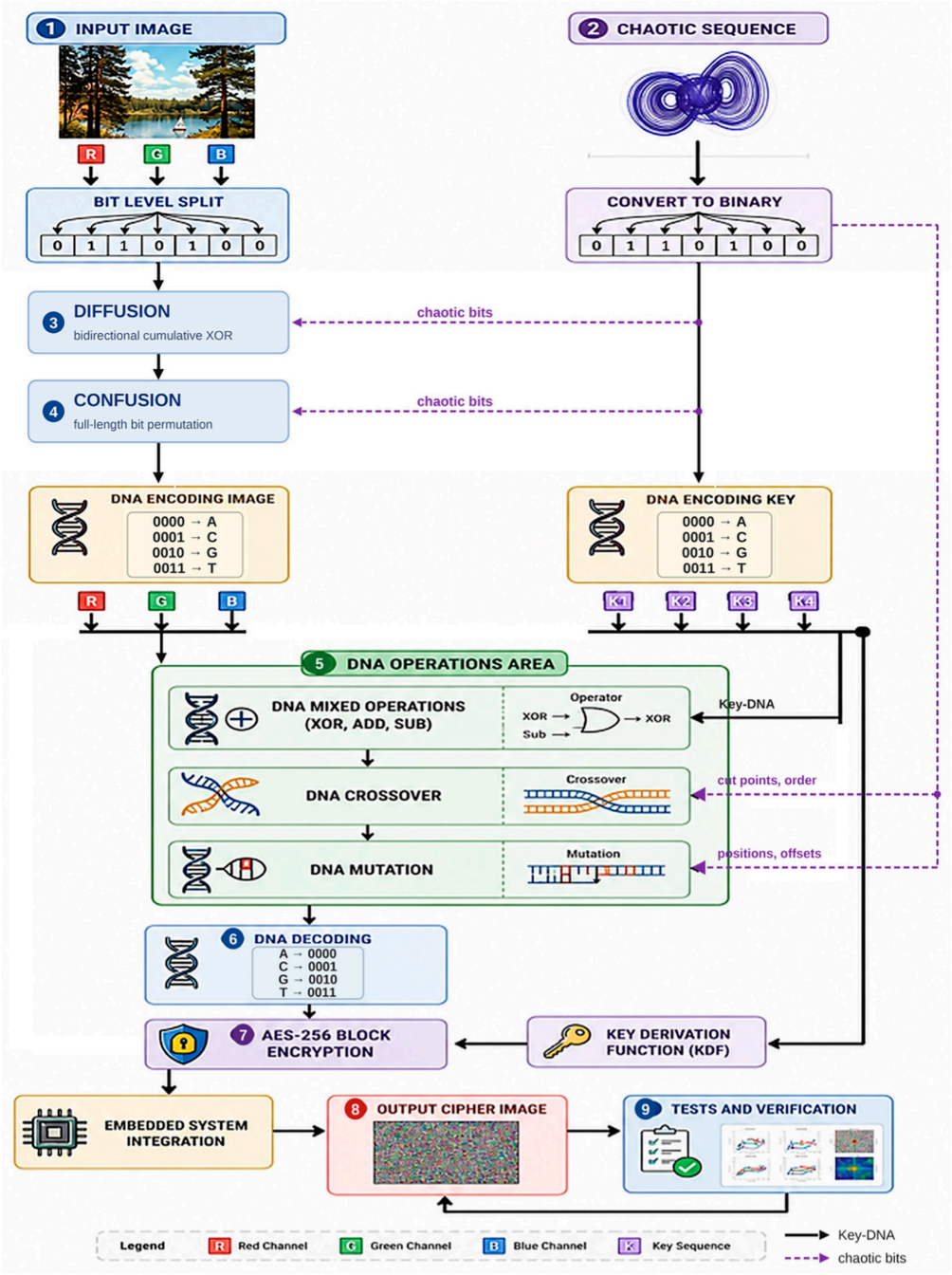


Figure 3. Proposed hybrid DNA–chaos image encryption architecture.

Table 3. Encryption and decryption pipeline.

	(a) Encryption pipeline
	Algorithm 1 Encryption
	Input: plain color image P (H × W × 3); chaotic bitstream S
	Output: cipher image C
1	for each channel ch in (B, G, R) do
2	K ← next_disjoint_segment(S) *// independent key segment per channel*
3	bits ← unpack_bits(ch) *// Step 2: bit-plane decomposition*
4	d ← diffusion_fwd(bits, K) *// Step 3: bidirectional cumulative XOR*
5	d ← reverse(diffusion_fwd(reverse(d), rev(K)))
6	perm ← PCG64_permutation(seed(K), len(d)) *// Step 4: confusion*
7	cf ← d[perm]
8	img_dna ← dna_encode(cf); key_dna ← dna_encode(K) *// Step 5: DNA encoding* (Table 2)
9	mixed ← [combine(img_dna[i], key_dna[i], rule = index(key_dna[i]) mod 3) for all i] *// Step 6: DNA-domain mixing*

Table 3. *Cont.*

(a) Encryption pipeline	
10	cuts ← cut_points(K, 3, 7); order ← segment_order(K, len(cuts) + 1) *// Step 7: multi-point crossover within Image-DNA*
11	img_child ← reorder_segments(mixed, cuts, order)
12	n_mut ← mut_count(K, 4000, 6500); mutated ← copy(img_child) *// Step 8: mutation*
13	for j = 1 to n_mut do
14	pos ← mut_position(K, j); off ← mut_offset(K, j)
15	mutated[pos] ← (mutated[pos] + off) mod 16
16	end for
17	dna_bytes ← pack_bits(dna_decode(mutated)) *// Step 9: DNA decoding*
18	aes_key ← SHA256(text(key_dna) + "|AES-KEY") *// Step 10: KDF input is the Key-DNA*
19	aes_nonce ← SHA256(text(key_dna) + "|AES-NONCE") [0:16]
20	cipher_ch ← AES256_CTR_encrypt(dna_bytes, aes_key, aes_nonce)
21	end for
22	return merge_channels(cipher_B, cipher_G, cipher_R) *// Step 11*
(b) Decryption pipeline	
	Algorithm 2 Decryption
	Input: cipher image C; chaotic bitstream S
	Output: plain color image P
1	for each channel ch in (B, G, R) do
2	K ← next_disjoint_segment(S) *// same order as encryption*
3	key_dna ← dna_encode(K) *// recomputed from S*
4	aes_key ← SHA256(text(key_dna) + "|AES-KEY") *// Step 10 (inverse): same KDF as encryption*
5	aes_nonce ← SHA256(text(key_dna) + "|AES-NONCE") [0:16]
6	dna_bytes ← AES256_CTR_decrypt(ch, aes_key, aes_nonce) *// CTR mode: encryption = decryption*
7	dna_seq ← dna_encode(unpack_bits(dna_bytes)) *// Step 9 (inverse)*
8	n_mut ← mut_count(K, 4000, 6500); for j = n_mut down to 1 do *// Step 8 (inverse): undo mutation*
9	p ← mut_position(K, j); dna_seq[p] ← (dna_seq[p] − mut_offset(K, j)) mod 16
10	end for
11	cuts ← cut_points(K, 3, 7); order ← segment_order(K, len(cuts) + 1) *// regenerated from S*
12	img_dna ← restore_segments(dna_seq, cuts, order) *// Step 7 (inverse): inverse segment order*
13	unmixed ← [uncombine(img_dna[i], key_dna[i], rule = index(key_dna[i]) mod 3) for all i] *// Step 6 (inverse): XOR kept, +/− swapped*
14	cf ← dna_decode(unmixed) *// Step 5 (inverse)*
15	perm ← PCG64_permutation(seed(K), len(cf)); d ← inverse_permute(cf, perm) *// Step 4 (inverse)*
16	u ← reverse(diffusion_inv(reverse(d), rev(K))) *// Step 3 (inverse)*
17	bits ← diffusion_inv(u, K)
18	plain_ch ← pack_bits(bits) *// Step 2 (inverse)*
19	end for
20	return merge_channels(plain_B, plain_G, plain_R) *// Step 11 (inverse)*

2.5. Security Analysis

The encrypted images were assessed through a histogram analysis, chi-square analysis, and correlation analysis (adjacent-pixel correlation in the horizontal, vertical, and diagonal directions) to evaluate their statistical and spatial characteristics. The plaintext sensitivity under single-pixel perturbations was evaluated using NPCR and UACI, while the uncertainty of pixel intensities was quantified through information entropy (IE) analysis. The robustness of the scheme was further tested under data loss and noise conditions, including cropping and salt-and-pepper noise, and was evaluated using the peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and mean squared error (MSE). A key sensitivity analysis and key space analysis were also performed. Finally, the practical efficiency of the scheme was evaluated through a Computational Performance Analysis by comparing the encryption and decryption times on a desktop CPU and an NVIDIA Jetson AGX Orin embedded platform.

2.6. Dataset

The proposed algorithm was evaluated on four standard test images—airplane (F-16), mandrill (Baboon), sailboat on a lake, and peppers—obtained from the miscellaneous volume of the University of Southern California Signal and Image Processing Institute (USC-SIPI) Image Database, a collection of many images available in 256 × 256, 512 × 512, and 1024 × 1024 resolutions [32]. All four selected images are 512 × 512, 24-bit color images, commonly employed as benchmark inputs in image encryption studies. These benchmark images exhibit markedly different visual characteristics, including smooth regions, highly textured structures, edges, and complex color distributions, thereby providing diverse test conditions for evaluating the proposed encryption scheme. Figure 4 shows the four benchmark images used in the experiments.

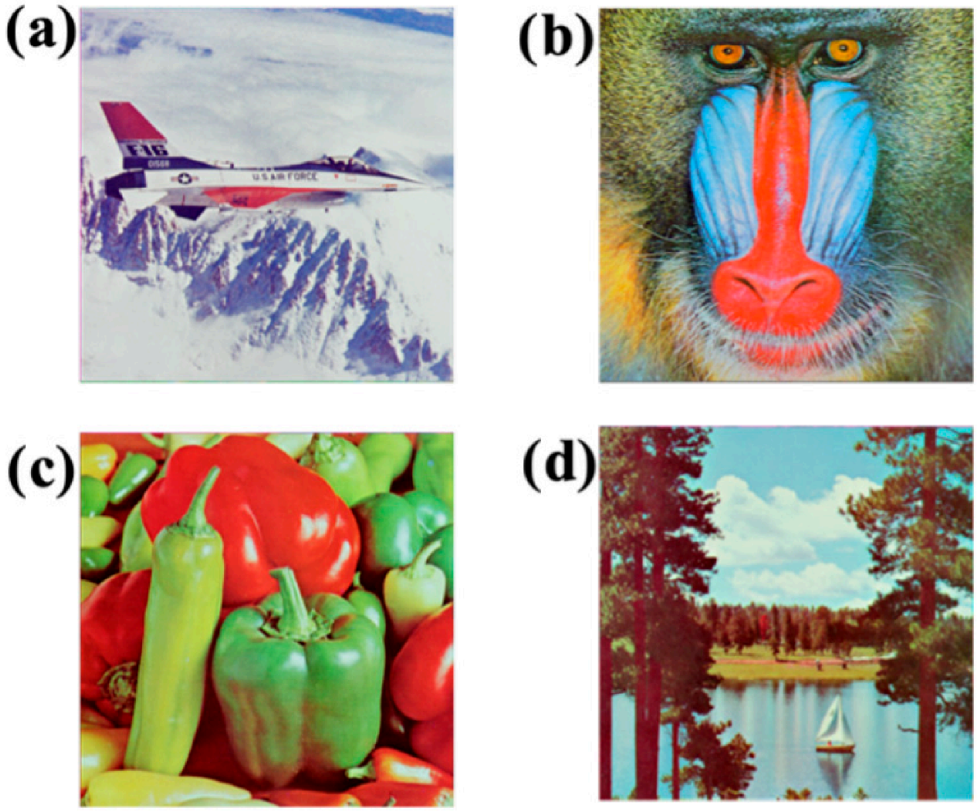


Figure 4. The four USC-SIPI benchmark test images used for evaluation: (**a**) airplane (F-16), (**b**) mandrill (Baboon), (**c**) peppers, and (**d**) sailboat on a lake.

3. Experimental Results

The experimental evaluations and performance tests were conducted on an NVIDIA Jetson AGX Orin embedded computing platform. The system was configured with an NVIDIA Ampere architecture GPU featuring 2048 CUDA cores, a 12-core Arm Cortex-A78AE CPU, and 64 GB of 256-bit LPDDR5 unified memory providing a bandwidth of 204.8 GB/s. Operating within a configurable power envelope of 15–60 W, the module used CUDA-based parallel processing to execute the proposed encryption and decryption pipeline.

3.1. Histogram Analysis

The histogram of the cipher image should be close to a uniform distribution over all 256 grey levels, as opposed to the highly non-uniform histogram of a natural image. As shown in Figure 5, the highly structured intensity distributions of the original images are

transformed into nearly uniform distributions after encryption, while the decrypted images recover the histogram characteristics of the corresponding original images.

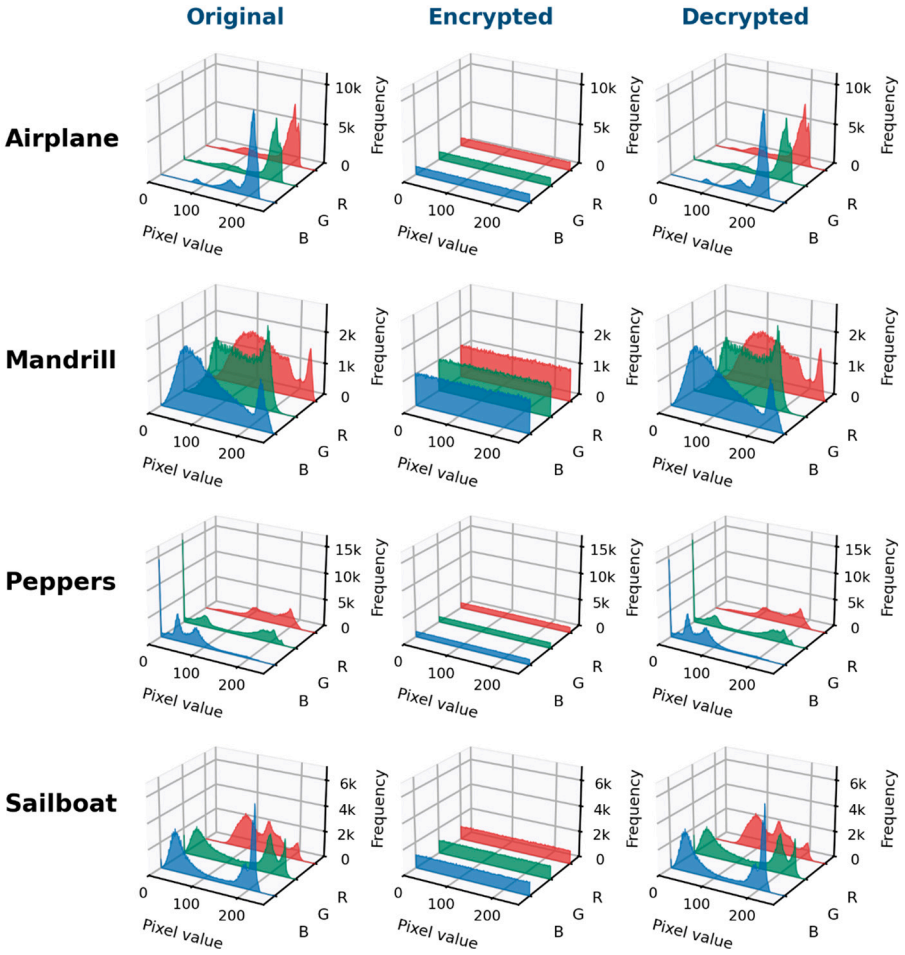


Figure 5. Histogram distributions of the original, encrypted, and decrypted benchmark images.

3.2. Chi-Square Analysis

To statistically verify the visual uniformity observed in the histogram analysis, a chi-square goodness-of-fit test is also implemented on the cipher image histograms. For 255 degrees of freedom and $\alpha = 0.05$, the corresponding critical value is 293.2478. Whenever the computed statistic is less than this critical value, the null hypothesis H0 (the pixel distribution of the cipher image is uniform) cannot be rejected. This is true for every channel of all four test images (Table 4), and no statistically significant departure from a uniform intensity distribution is detected under the adopted test conditions.

Table 4. Chi-square uniformity test results for cipher images (critical value = 293.2478, alpha = 0.05, df = 255).

Image	Channel	Chi-Square Statistic	*p*-Value	Result
Airplane	B	281.3965	0.1230	Pass
	G	284.0234	0.1023	Pass
	R	227.5645	0.8909	Pass
	Average	264.3281	-	Pass
Mandrill	B	263.9863	0.3362	Pass
	G	267.0957	0.2888	Pass
	R	264.0801	0.3347	Pass
	Average	265.0540	-	Pass

Table 4. *Cont.*

Image	Channel	Chi-Square Statistic	*p*-Value	Result
Peppers	B	282.3438	0.1152	Pass
	G	259.4355	0.4111	Pass
	R	292.2559	0.0543	Pass
	Average	278.0117	-	Pass
Sailboat	B	239.8691	0.7435	Pass
	G	252.5938	0.5308	Pass
	R	259.2832	0.4137	Pass
	Average	250.5820	-	Pass

3.3. Correlation Analysis

The adjacent-pixel correlation is evaluated in the horizontal, vertical, and diagonal directions for both the plain and cipher images. The adjacent-pixel correlation coefficients are expected to change from relatively high values in the plain image to values near zero in the cipher image. As shown in Figure 6, the strong linear relationship between adjacent pixels in the original images is effectively reduced after encryption, with the cipher image samples exhibiting a widely scattered distribution and negligible correlation.

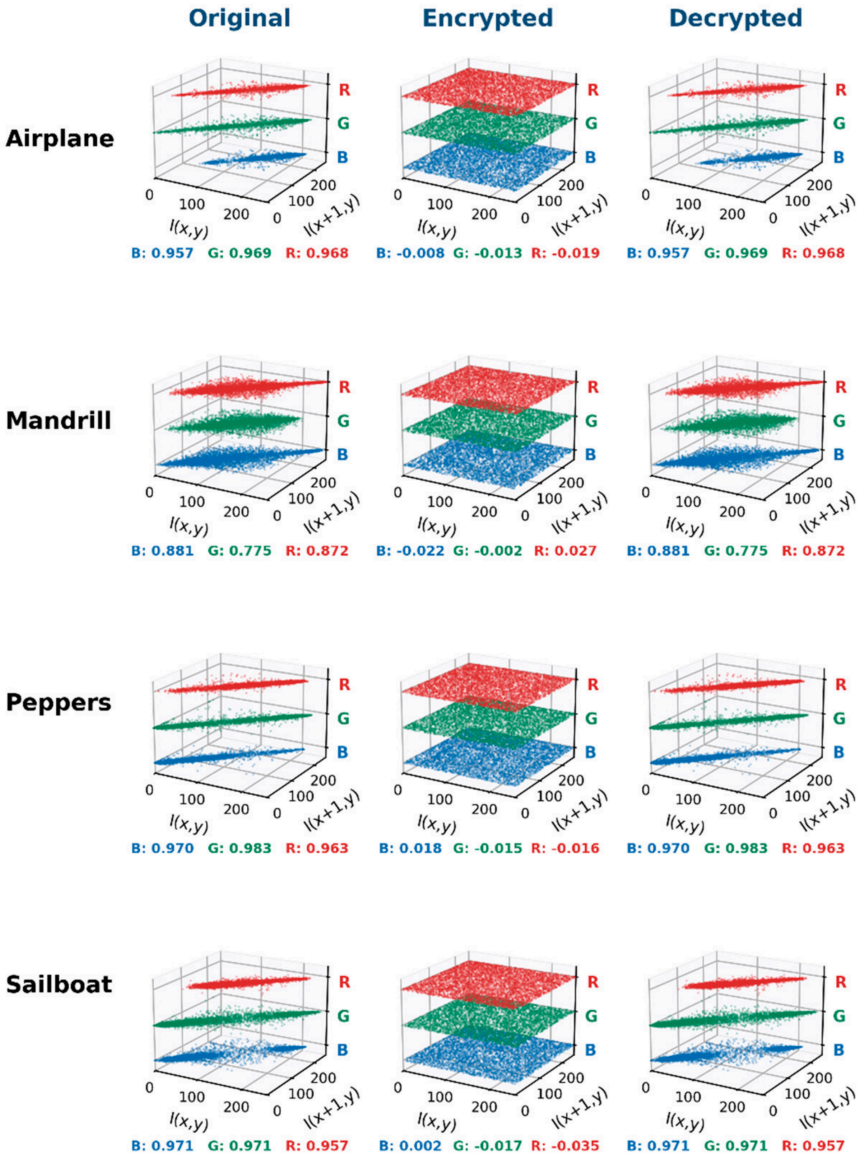


Figure 6. Adjacent-pixel correlation distributions of the original and encrypted benchmark images.

As summarized in Table 5, the original images exhibit strong adjacent-pixel correlations in all three directions, with coefficients ranging from 0.8101 to 0.9721. After encryption, these values decrease to near zero, ranging from −0.0194 to 0.0107, indicating that the proposed scheme effectively reduces the measured linear spatial dependencies between neighboring pixels.

Table 5. Average adjacent-pixel correlation coefficients (mean over B, G, and R channels).

Image	Direction	Plain Image	Cipher Image
Airplane	Horizontal	0.9622	−0.0113
	Vertical	0.9660	−0.0010
	Diagonal	0.9391	−0.0026
Mandrill	Horizontal	0.8356	0.0107
	Vertical	0.8965	−0.0043
	Diagonal	0.8101	−0.0012
Peppers	Horizontal	0.9721	−0.0147
	Vertical	0.9714	−0.0056
	Diagonal	0.9581	−0.0194
Sailboat	Horizontal	0.9663	0.0039
	Vertical	0.9664	0.0026
	Diagonal	0.9514	0.0105

3.4. Differential Attack

Table 6 shows the NPCR and UACI results for the four benchmark images. For each color channel, the analysis is performed by encrypting the same plaintext image with two different chaotic keys taken from disjoint parts of the bitstream and comparing the generated cipher images. In an additional plaintext sensitivity test, one randomly chosen pixel of the plaintext is modified and both images are encrypted with the same chaotic key; because the AES key and nonce are derived by SHA-256 from the Key-DNA, i.e., from the chaotic key bitstream alone, they are identical for the two images and do not depend on the plaintext, so the observed avalanche originates from the chaos–DNA stages. The mean NPCR over the three channels is between 99.56% and 99.66%, and the mean UACI between 33.23% and 33.65% for the four images.

Table 6. NPCR and UACI results per channel for images.

Image	Channel	NPCR (%)	UACI (%)
Airplane	B	99.5895	33.3967
	G	99.5815	33.4641
	R	99.6128	33.4961
	Average	99.5946	33.4523
Mandrill	B	99.6181	33.4431
	G	99.6181	33.4982
	R	99.6296	33.4972
	Average	99.6220	33.4795
Sailboat	B	99.6014	33.5515
	G	99.5884	33.3531
	R	99.6109	33.5012
	Average	99.6002	33.4686
Peppers	B	99.5964	33.4666
	G	99.6235	33.4853
	R	99.6105	33.5062
	Average	99.6101	33.4860

The NPCR values are given in Table 6 and are always greater than the adopted critical value of 99.5810%, while all the UACI values are within the expected range of 33.3445–33.5826% for the 512 × 512, 8-bit images at a significance level of $\alpha = 0.01$. These results indicate high sensitivity of the cipher output to the chaotic key under the adopted NPCR/UACI criteria; the plaintext avalanche response is given by the single-pixel test above.

3.5. Information Entropy (IE)

According to Shannon's formulation [33], information entropy is a measure of the uncertainty of pixel intensities and is expected to be close to the theoretical maximum of 8 bits/pixel for an ideally random 8-bit image.

Table 7 shows that the average entropy values of the encrypted images are between 7.9992 and 7.9993 bits/pixel, although the original images have quite different entropy values. These values are very close to the theoretical maximum, indicating high uncertainty in the marginal intensity distributions of the cipher images. Also, the entropy values of the decrypted images are exactly recovered to those of the original images, which is consistent with the reversible behavior of the encryption–decryption process.

Table 7. Information entropy (bits/pixel) per channel for used images.

Image	Original (Avg.)	Encrypted (Avg.)	Decrypted (Avg.)
Airplane	6.5769	7.9993	6.5769
Mandrill	7.6444	7.9993	7.6444
Sailboat	7.3896	7.9993	7.3896
Peppers	7.2978	7.9992	7.2978

3.6. Data Loss and Noise Attacks

During transmission over public networks or storage in physical media, encrypted images may be affected by data loss and noise contamination [34,35]. To evaluate the robustness of the proposed scheme against data loss during transmission, selected regions of the cipher images are occluded by assigning the corresponding pixel values to zero at ratios of 1/16, 1/32, and 1/64 of the total image area. The correct key is used to decrypt the corrupted cipher images. Figure 7 shows that the original visual content can still be perceived at the highest data loss ratio of 1/16, while the reconstructed image suffers from local degradation. As the amount of lost cipher data is reduced from 1/16 to 1/64, the reconstruction quality progressively improves. Because the confusion stage is a full-length bit permutation, the errors caused by a lost cipher region are scattered over the decrypted image as noise-like distortion instead of remaining confined to the occluded region; recoverable visual information is nevertheless retained at the tested loss ratios. The robustness against channel noise is further tested by adding salt-and-pepper noise to the cipher images with noise densities of 1%, 10%, and 20%. The resulting noisy cipher images are decrypted using the correct key and compared with their corresponding original images. As shown in Figure 7, the reconstruction quality decreases as the noise density increases. However, part of the main visual structure remains recognizable even at a noise density of 20%, despite the substantial quantitative degradation at this level.

Ciphertext robustness, error propagation and plaintext recovery quality are different properties. AES-256-CTR maps one corrupted cipher bit to exactly one corrupted bit of the DNA-domain data, without chaining. Undoing the mutation, the crossover (a reordering of segments) and the DNA mixing confines the error to the DNA symbol that contains this bit. The inverse confusion then relocates the corrupted bits to pseudo-random positions over the whole bitstream, and the inverse diffusion turns each isolated bit error into two-bit errors at the neighboring indices. In a bit-flip test, a single flipped cipher bit produces at

most eight erroneous plaintext bits (about 2.7 on average) and affects about 1.7 pixels on average. The quality of the recovered image at noise density d is thus determined by the number of corrupted cipher bytes (about d·N), spread uniformly over the image, which explains the PSNR decrease from 1% to 20% in Table 8.

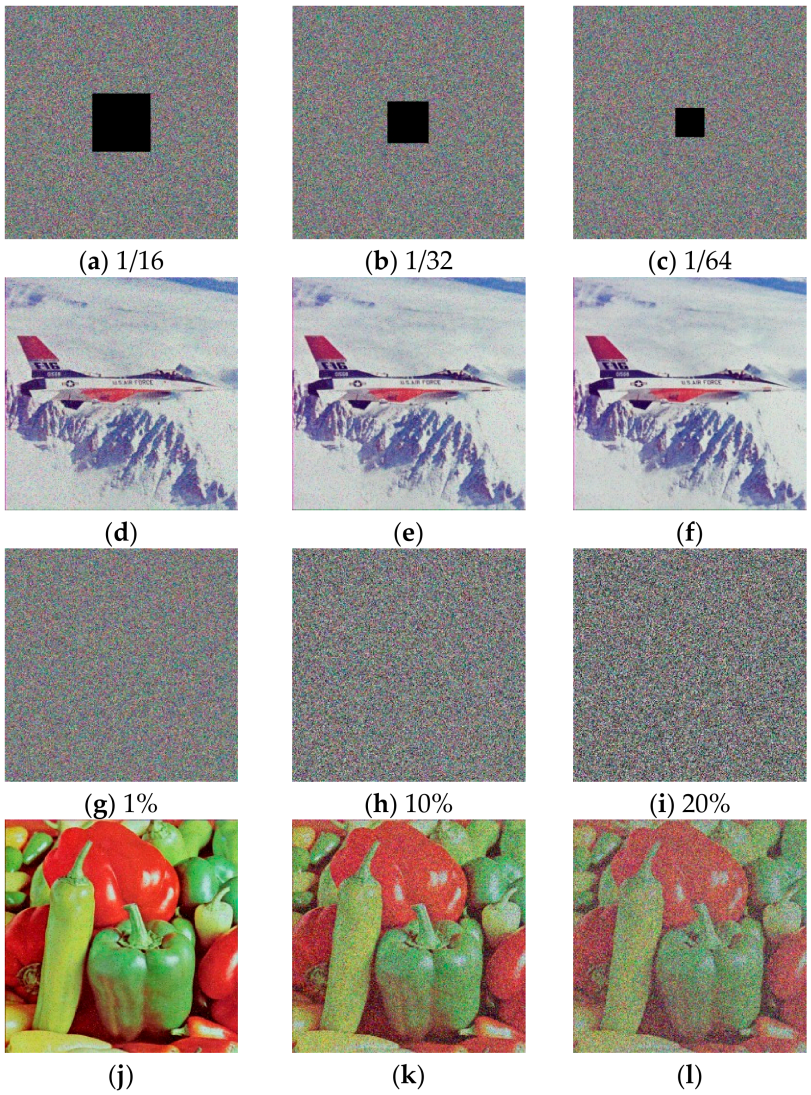


Figure 7. Visual robustness analysis under data loss and noise attacks at different severity levels. (**a**–**c**) Cipher images (airplane) with 1/16, 1/32 and 1/64 of the area zeroed (data loss); (**d**–**f**) the corresponding images decrypted with the correct key; (**g**–**i**) cipher images (peppers) contaminated with salt-and-pepper noise of density 1%, 10% and 20%; and (**j**–**l**) the corresponding images decrypted with the correct key.

Table 8. Decrypted image quality (PSNR, SSIM, MSE vs. original) under noise contamination and data loss (cropping) attacks.

Image	Attack	PSNR (dB)	SSIM	MSE
Airplane	Noise 1%	22.3595	0.5404	377.6832
	Noise 10%	12.9807	0.1770	3273.4631
	Noise 20%	10.6873	0.1076	5550.7170
	Data loss 1/16	17.4097	0.3206	1180.6092
	Data loss 1/32	20.3251	0.4417	603.3472
	Data loss 1/64	23.3876	0.5927	298.0698

Table 8. *Cont.*

Image	Attack	PSNR (dB)	SSIM	MSE
Mandrill	Noise 1%	22.0130	0.7791	409.0507
	Noise 10%	12.9693	0.3376	3282.1008
	Noise 20%	10.9413	0.2004	5235.4282
	Data loss 1/16	17.1154	0.5684	1263.3944
	Data loss 1/32	19.9854	0.7023	652.4336
	Data loss 1/64	22.9975	0.8153	326.0826
Peppers	Noise 1%	21.9420	0.5460	415.7961
	Noise 10%	12.7537	0.1523	3449.1123
	Noise 20%	10.5823	0.0890	5686.5084
	Data loss 1/16	17.0765	0.3043	1274.7621
	Data loss 1/32	19.9615	0.4413	656.0465
	Data loss 1/64	22.9885	0.6014	326.7646
Sailboat	Noise 1%	22.0138	0.6415	408.9829
	Noise 10%	12.7951	0.2284	3416.4311
	Noise 20%	10.6011	0.1386	5661.9880
	Data loss 1/16	17.1374	0.4145	1257.0233
	Data loss 1/32	20.0222	0.5465	646.9312
	Data loss 1/64	23.0689	0.6890	320.7711

The quantitative results (PSNR, SSIM, and MSE) for both attacks are summarized in Table 8 and complement the visual observations in Figure 7. For all benchmark images, increasing the noise density results in lower PSNR and SSIM values and higher MSE, whereas reducing the data loss ratio from 1/16 to 1/64 consistently improves the reconstruction quality. In particular, the 1/64 data loss condition yields PSNR values above 22.9 dB for all test images, indicating that the proposed scheme retains recoverable image information under this tested level of cipher data loss.

3.7. Key Sensitivity Analysis

To assess the scheme's sensitivity to the secret key, each test image was encrypted twice: once with the correct key and once with a key differing by a single bit in the B-channel chaotic sequence. The NPCR/UACI metrics were used to compare the two cipher images. The cipher image was also decrypted with the incorrect key and compared with the original plain image. The resulting NPCR and UACI values were close to the commonly used reference values of 99.61% and 33.46%, respectively. In addition, wrong-key decryption produced a near-zero correlation with the original image, entropy close to the theoretical maximum of 8 bits/pixel, and a low PSNR. These results indicate that a single-bit perturbation of the tested key material produces a substantially different cipher output and prevents meaningful reconstruction with the perturbed key.

3.8. Key Space Analysis

The terminal AES-256 stage provides a nominal secret key space of (2^{256}), which is sufficiently large to make an exhaustive key search computationally impractical under conventional brute-force assumptions. The 128-bit nonce used in AES-CTR provides a large nonce domain for avoiding reuse under the same key; however, it is not treated as an additional secret key component. Accordingly, the brute-force resistance of the terminal cryptographic stage is determined primarily by the AES-256 secret key space, provided that the secret key material remains unavailable to an adversary and nonce reuse is avoided.

3.9. Computational Performance

Table 9 presents the total encryption and decryption times summed over the B, G, and R channels for each 512 × 512 benchmark image. The most computationally intensive stage on the desktop CPU is the DNA-domain mixing stage, which consumes approximately 40–44% of a single-channel processing pass, followed by the multi-point crossover and

DNA decoding stages. In contrast, the diffusion and confusion stages are of linear $O(n)$ complexity and constitute a relatively small part of the overall runtime.

Table 9. Total encryption and decryption times.

Image	CPU		AGX ORIN	
	Encryption Time (s)	Decryption Time (s)	Encryption Time (s)	Decryption Time (s)
Airplane	1.5529	1.3918	0.2434	0.2181
Mandrill	1.4978	1.3832	0.2218	0.2048
Peppers	1.5369	1.3538	0.2404	0.2118
Sailboat	1.4904	1.3697	0.2176	0.2003

As shown in Table 9, the implementation on the Jetson AGX Orin significantly reduces the processing time compared with the desktop CPU, with the encryption time decreasing from approximately 1.49–1.55 s to 0.22–0.24 s. Moreover, the processing times remain within a narrow range for all four benchmark images on both platforms, indicating limited dependence on image content for inputs of the same resolution. These results demonstrate the potential of parallel hardware acceleration to reduce the execution time of the proposed encryption scheme on an embedded edge platform. Here, real-time refers to the per-image latency of 0.22–0.24 s, which allows several 512 × 512 color frames per second to be encrypted in IoT image-sensing applications.

The CPU reference is the sequential Python 3.11.14 implementation of the algorithm running on an AMD Ryzen 9 3900X desktop processor with 12 cores and 24 threads at up to 4.6 GHz; the embedded platform runs the CUDA implementation with one stream per color channel. Each time is the end-to-end encryption or decryption of a 512 × 512 color image, that is, three channels including AES-256-CTR, with host–device transfers included and chaotic-bit generation excluded, and is the mean of N = 100 runs after 10 warm-up runs measured with a monotonic clock and GPU synchronization for CUDA. The average power of the Orin module during the timed runs, logged with tegrastats, is 20 W, i.e., 4.6 J per image at 4.3 images/s, which supports low-power operation of the proposed implementation on an edge device.

3.10. Known and Chosen Plaintext Attack

The known-plaintext and chosen-plaintext resistance are examined analytically from the structure of the cipher. All secret quantities of an encryption (Key-DNA, permutation, cut points, mutation positions, AES key and nonce) are derived from the chaotic key bitstream and do not depend on the plaintext. The bidirectional cumulative XOR diffusion and the keyed permutation are linear over $GF(2)$, but they are followed by key-dependent modulo-16 addition/subtraction in the DNA domain, segment-reordering crossover, key-dependent mutation and AES-256-CTR, so that the known or chosen plaintext–ciphertext pairs under one key do not expose the linear stages in isolation. As in any stream-based construction, a chaotic key is used for one image.

Chosen- and known-plaintext tests are performed on the 512 × 512 images, each encrypted with a different chaotic key taken from disjoint parts of the bitstream. The all-black and all-white images encrypt to cipher images with entropy of 7.9993 and 7.9993, chi-square statistics of 249.5 and 265.2 (below the critical value of 293.2) and absolute adjacent-pixel correlations below 0.016; encrypting the same all-black or all-white image with two different keys gives NPCR/UACI of 99.62%/33.47% and 99.62%/33.48%. In the known-plaintext test, the second image estimated from one known plaintext–ciphertext pair as $P_2 = P_1 \oplus C_1 \oplus C_2$ has a PSNR of 8.77 dB and an SSIM of 0.008, the same as a random

image (8.78 dB, 0.009), and decrypting with a different key or with a key differing in one bit gives PSNR values of 7.98 dB and 7.97 dB.

The AES key and nonce are derived from the Key-DNA, which the receiver regenerates from the shared chaotic bitstream. The cut points, the segment order and the mutation positions and offsets are regenerated from the same bitstream, so no side information is transmitted with the cipher image, and the security of the AES stage does not depend on any quantity sent alongside the cipher image.

3.11. Ablation Study

Removing any single stage degrades at least one security measure. Without diffusion, the single-pixel NPCR falls to 0%; without confusion, the adjacent-pixel correlation of the intermediate result rises; without the DNA mixing, the cipher becomes affine over GF(2) and falls to a known-plaintext attack; without crossover and mutation, the segment- and symbol-level diffusion drops and the single-pixel NPCR and UACI move out of the critical range; and without AES, the key-sensitivity UACI leaves the 33.2255–33.7016% interval. With all stages, the NPCR (99.61%), UACI (33.46%) and the affinity measure (0.5) reach their ideal values, so each stage is necessary and none is redundant.

3.12. Comparison with Recent Schemes

Table 10 compares the proposed scheme with five recent chaos- and DNA-based schemes, given below in terms of NPCR, UACI and information entropy, whose ideal values are 99.6094%, 33.4635% and eight, respectively. The NPCR (99.61%), UACI (33.47%) and entropy (7.9993) of the proposed scheme are at the ideal values and are comparable to the values reported in the compared works. Because the compared schemes differ in their test images and in the way the NPCR and UACI are measured, the table is intended as an indication of the achieved security level rather than as a strict ranking.

Table 10. Comparison of NPCR, UACI and information entropy with recent chaos- and DNA-based schemes.

Study	NPCR	UACI	IE
Ours	99.60	33.4554	7.9993
[23]	99.58	33.47	7.9999
[36]	99.61	33.4637	7.9986
[37]	99.626	32.2860	7.99855
[38]	99.64	33.73	7.9957
[39]	99.6089	33.4554	7.9993

4. Discussion

In this paper, the chaotic bitstream successfully passed the whole NIST SP 800-22 suite without any post-processing, which supports its statistical suitability for the tested cryptographic keystream generation setting. The chi-square, histogram, and correlation analyses show that the three-layer pipeline substantially reduces the measured statistical and spatial structure of the plain images. The adjacent-pixel correlation decreases from as high as 0.97 to near zero. The NPCR/UACI results are found to be close to the commonly used reference values of 99.6%/33.46%, and the entropy values are found to be close to the theoretical maximum of 8 bits/pixel, indicating high plaintext sensitivity and low residual intensity redundancy under the adopted statistical measures. The key sensitivity results show that the decrypted output is strongly decorrelated from the original image when a single-bit change is introduced into the tested key material, because each downstream stage is re-derived from the corresponding key-dependent sequence. The nominal 2^{256} key

space of the AES-256 stage makes an exhaustive key search computationally impractical under conventional brute-force assumptions, provided that the secret key material remains unavailable to an adversary and CTR nonce reuse is avoided. No side information is transmitted with the cipher image.

The robustness tests demonstrate progressive degradation with increasing cipher data corruption. At the most severe data loss ratio tested (1/16), the reconstructed image remains visually interpretable, whereas the PSNR exceeds 22.9 dB for all benchmark images at the 1/64 data loss condition. Under salt-and-pepper noise, the reconstruction quality decreases as the noise density increases, with substantial quantitative degradation at the 20% noise level, although part of the visual structure remains recognizable. This behavior is consistent with the non-chaining property of AES-256-CTR, although the end-to-end corruption response also depends on the preceding inverse transformation stages. Finally, the Jetson AGX Orin implementation reduces the encryption time for a 512 × 512 color image from approximately 1.5 s to 0.22–0.24 s (about 6–7×, depending on the benchmark image). The encryption time varies only slightly among the four benchmark images of the same resolution, suggesting a largely content-independent computational workload. These results support the feasibility of the proposed pipeline for embedded and edge-oriented image encryption applications, while application-specific latency and power requirements should be evaluated separately for a given deployment scenario.

5. Conclusions

This paper proposed a hybrid image encryption scheme built around the D-PQM–HNN as its chaotic core, and combined it with an extended 16-symbol DNA-domain confusion layer—featuring rule-selected mixing, genetic algorithm-inspired crossover, and mutation—followed by a terminal AES-256-CTR stage. The chaotic sequence used to drive the pipeline was shown to pass the full NIST SP 800-22 statistical test suite, and the resulting cryptosystem was evaluated on four standard benchmark images through histogram, chi-square, correlation, NPCR/UACI, entropy, key sensitivity, key space, and noise/data loss robustness analyses, which collectively indicate near-uniform cipher image statistics, high plaintext and key sensitivity, and recoverable image information under the tested corruption conditions. The complete eleven-stage pipeline was also implemented and benchmarked on an NVIDIA Jetson AGX Orin embedded platform. Channel-wise CUDA parallelization reduced the total encryption time for a 512 × 512 color image by approximately 7×, from about 1.5 s on the reference desktop CPU to 0.22–0.24 s on the embedded platform, supporting the feasibility of the proposed framework for edge-oriented image encryption. The terminal AES-256 stage provided a nominal 2^{256} secret key space, making exhaustive key search computationally impractical under conventional brute-force assumptions, provided that the secret key material remained unavailable and CTR nonce reuse was avoided. Future work will focus on extending the framework to video streams and evaluating its resilience against machine learning-based cryptanalysis.

Author Contributions: M.S.: Conceptualization, Methodology, Validation, Supervision, Writing—review and editing, and Project administration. M.E.S.: Conceptualization, Methodology, Software, Data curation, Visualization, and Writing—original draft. E.G.U.: Methodology, Validation, Formal analysis, and Writing—review and editing. B.K.: Methodology, Validation, Supervision, and Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Scientific Research Projects Coordination Unit of Firat University; project number: MF.26.170.

Data Availability Statement: The image data used in this study are the standard benchmark images (airplane, mandrill, sailboat, peppers) obtained from the publicly available USC-SIPI Image Database, as referenced in the paper. No new datasets were generated during this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ghadirli, H.M.; Nodehi, A.; Enayatifar, R. An overview of encryption algorithms in color images. *Signal Process.* **2019**, *164*, 163–185. [CrossRef]
2. Matthews, R. On the derivation of a 'chaotic' encryption algorithm. *Cryptologia* **1989**, *13*, 29–42. [CrossRef]
3. Fridrich, J. Symmetric Ciphers Based on Two-Dimensional Chaotic Maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284. [CrossRef]
4. Zhang, Z.; Tang, J.; Zhang, F.; Huang, T.; Lu, M. Medical Image Encryption Based on Josephus Scrambling and Dynamic Cross-Diffusion for Patient Privacy Security. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 9250–9263. [CrossRef]
5. Bao, H.; Chen, Z.; Ma, J.; Xu, Q.; Bao, B. Planar Homogeneous Coexisting Hyperchaos in Bi-Memristor Cyclic Hopfield Neural Network. *IEEE Trans. Ind. Electron.* **2024**, *71*, 16398–16408. [CrossRef]
6. Yang, G.; Wang, C.; Sun, Y.; Deng, Q. A Discrete Memristive Hopfield Neural Network with Grid Multi-structure/scroll-like Attractors. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2026**, *45*, 4853–4862. [CrossRef]
7. Lai, Q.; Ji, L. A Bidirectional Cross-Scrambling Medical Image Encryption Scheme Incorporates Compressed Sensing and Its Application in IoMT. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 7697–7705. [CrossRef]
8. Candan, C.; Sahin, M.E. Audio Signal Encryption Using Memristor-Based Hyperchaotic Dynamics and Statistical Randomness Validation. *J. Comput. Nonlinear Dyn.* **2026**, *21*, 031002. [CrossRef]
9. Kong, X.; Yu, F.; Yao, W.; Cai, S.; Zhang, J.; Lin, H. Memristor-induced hyperchaos, multiscroll and extreme multistability in fractional-order HNN: Image encryption and FPGA implementation. *Neural Netw.* **2024**, *171*, 85–103. [CrossRef] [PubMed]
10. Wang, Y.; Su, P.; Wang, Z.; Sun, J. FN-HNN Coupled with Tunable Multistable Memristors and Encryption by Arnold Mapping and Diagonal Diffusion Algorithm. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2025**, *72*, 4830–4842. [CrossRef]
11. Wang, Y.; Su, P.; Wang, Z.; Sun, J. Dynamic analysis of high dimensional HNN with logistic-based memristors and application in military image encryption. *Chaos Solitons Fractals* **2025**, *199*, 116868. [CrossRef]
12. Deng, Q.; Wang, C.; Sun, Y.; Yang, G. Harnessing Complex-Valued Chaos in Discrete-Time Hopfield Neural Network for Secure Image Encryption. *IEEE Trans. Circuits Syst. Video Technol.* **2026**, *36*, 12228–12242. [CrossRef]
13. He, S.; Xiao, J.; Peng, Y.; Wang, H. Discrete memristive spiking neural networks: Investigating information flow, synchronization, and emergent intelligence. *Cogn. Neurodynamics* **2025**, *20*, 12. [CrossRef] [PubMed]
14. Pfeiffer, P.; Egusquiza, I.L.; DI Ventra, M.; Sanz, M.; Solano, E. Quantum memristors. *Sci. Rep.* **2016**, *6*, 29507. [CrossRef] [PubMed]
15. Sanz, M.; Lamata, L.; Solano, E. Invited Article: Quantum memristors in quantum photonics. *APL Photonics* **2018**, *3*, 080801. [CrossRef]
16. Spagnolo, M.; Morris, J.; Piacentini, S.; Antesberger, M.; Massa, F.; Crespi, A.; Ceccarelli, F.; Osellame, R.; Walther, P. Experimental photonic quantum memristor. *Nat. Photonics* **2022**, *16*, 318–323. [CrossRef]
17. Dai, G.; Li, W.; Zhong, S.; Yang, S.; Wu, D. Quantum circuit model for photonic quantum memristors and its optimization. *Phys. Lett. A* **2025**, *555*, 130774. [CrossRef]
18. Sağbaş, M.; Minaei, S. Differential photonic quantum memristor–modulated hopfield neural network with secure IoMT federated learning. *Chaos Solitons Fractals* **2026**, *210*, 118677. [CrossRef]
19. Ulutas, H. A novel memristor-based hyperchaotic hybrid encryption system with DNA for image encryption on the Jetson TX2. *Sci. Rep.* **2025**, *15*, 35745. [CrossRef] [PubMed]
20. Yan, X.; Hu, Q.; Teng, L. A novel color image encryption method based on new three-dimensional chaotic mapping and DNA coding. *Nonlinear Dyn.* **2024**, *113*, 1799–1826. [CrossRef]
21. Yan, M.; Liu, M.; Li, C. DNA color image encryption based on conservative chaotic system. *Sci. Rep.* **2025**, *15*, 9278. [CrossRef] [PubMed]
22. Cavazzana, R.; Haliuk, S.; Ascoli, A.; Vovchuk, D.; Salgals, T.; Bobrovs, V.; Pareschi, F.; Corinto, F.; Secco, J. Preserving the Confidentiality of Clinical Images Through a Chaotic Low-Power Hardware Platform and DNA Coding-Based Encryption. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2026**, *73*, 4826–4839. [CrossRef]
23. Demirkol, A.S.; Sahin, M.E.; Karakaya, B.; Ulutas, H.; Ascoli, A.; Tetzlaff, R. Real time hybrid medical image encryption algorithm combining memristor-based chaos with DNA coding. *Chaos Solitons Fractals* **2024**, *183*, 114923. [CrossRef]
24. Suzgen, E.E.; Sahin, M.E.; Ulutas, H. A novel hybrid medical image encryption scheme based on memristive chaos and DNA-ARX-3DES with Real-Time implementation. *Sci. Rep.* **2026**, *16*, 6230. [CrossRef] [PubMed]

25. Wen, H.; Lin, Y.; Kang, S.; Zhang, X.; Zou, K. Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion. *iScience* **2024**, *27*, 108610. [CrossRef] [PubMed]
26. Alexan, W.; Megalli, Y. A new fast high dimensional and memristive hyperchaotic multiple image encryption method and its FPGA implementation. *Discov. Electron.* **2025**, *2*, 75. [CrossRef]
27. Peng, Y.; Huang, Z.; Li, Z.; He, S.; Zou, J. An Effective Visually Meaningful Image Encryption Based on Compressive Sensing and Multiobjective Optimization Algorithm. *IEEE Internet Things J.* **2026**, *13*, 4076–4087. [CrossRef]
28. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E.; Leigh, S.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; Nist Special Publication: Gaithersburg, MD, USA, 2010.
29. Ahmad, M.; Alsharari, H.D.; Nizam, M. Security improvement of an image encryption based on mPixel-chaotic-shuffle and pixel-chaotic-diffusion. *arXiv* **2014**, arXiv:1403.6626.
30. Meng, F.Q.; Wu, G. A color image encryption and decryption scheme based on extended DNA coding and fractional-order 5D hyper-chaotic system. *Expert Syst. Appl.* **2024**, *254*, 124413. [CrossRef]
31. Karumbunathan, L.S. *NVIDIA Jetson AGX Orin Series: A Giant Leap Forward for Robotics and Edge AI Applications*; NVIDIA Technical Brief, v1.2; NVIDIA: Santa Clara, CA, USA, 2022.
32. Signal and Image Processing Institute, University of Southern California. The USC-SIPI Image Database, Volume 3: Miscellaneous. Available online: https://sipi.usc.edu/database/database.php?volume=misc (accessed on 13 August 2026).
33. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
34. Hua, Z.; Zhou, Y. Image encryption using 2D Logistic-adjusted-Sine map. *Inf. Sci.* **2016**, *339*, 237–253. [CrossRef]
35. Wang, X.; Liu, C. A novel and effective image encryption algorithm based on chaos and DNA encoding. *Multimed. Tools Appl.* **2016**, *76*, 6229–6245. [CrossRef]
36. Maity, A.; Chen, Y.L.; Alexan, W.; Yee, P.L.; Dhara, B.C. Image encryption with 6D hyperchaotic system and vision transformer autoencoder. *Sci. Rep.* **2026**, *16*, 4243. [CrossRef] [PubMed]
37. Megalli, Y.; Edward, P.; Alexan, W. Fractional-order chaos–driven multi-stage image encryption scheme with an FPGA realization. *Sci. Rep.* **2026**, *16*, 36355. [CrossRef]
38. Mei, J.; Fan, B.; Wu, S. A novel image encryption algorithm based on cosine-modulated chaotic system and dynamic DNA coding. *Discov. Appl. Sci.* **2026**, *8*, 1226. [CrossRef]
39. Xu, W.; Xue, L.; Zhu, L.; Li, J.; Li, G. A DNA-Based Image Encryption Scheme Using Optimized Lorenz–Sprott Hyperchaotic System. *Entropy* **2026**, *28*, 964. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.