



ARTICLE

Cylinder XOR-Cascade: Lightweight Image Encryption Using Autoencoder-Based Representation

Rania Al-Ali¹, Mustafa Al-Fayoumi^{1,2} and Saleem Alsaraireh^{3,*}

¹Department of Cybersecurity, King Hussein School of Computing Sciences, Princess Sumaya University for Technology (PSUT), Amman, Jordan

²Department of Computer Science, Faculty of Information Technology, Middle East University (MEU), Amman, Jordan

³Department of Computer Science, King Hussein School of Computing Sciences, Princess Sumaya University for Technology (PSUT), Amman, Jordan

*Corresponding Author: Saleem Alsaraireh. Email: sal20219008@std.psut.edu.jo

Received: 19 May 2026; Accepted: 25 June 2026; Published: 15 September 2026

ABSTRACT: The rapid growth of the Internet of Things (IoT) and edge computing has increased the demand for secure and lightweight image encryption suitable for resource-constrained environments. This paper proposes a hybrid framework combining a residual-based pretrained autoencoder with a novel Cylinder XOR-Cascade (CXC) encryption scheme. The autoencoder compresses images into a compact latent representation while a residual branch preserves fine spatial details for accurate reconstruction. Both representations are encrypted using CXC, a two-pass column-wise stream cipher that enhances confusion and diffusion through sequential SHA3-256-based chaining and a cylinder-like feedback mechanism. Experiments on the USC-SIPI dataset demonstrate strong statistical security: Number of Pixels Change Rate (NPCR) $\approx 99.5\%$, Unified Average Changing Intensity (UACI) $\approx 33.5\%$, and entropy ≈ 7.5 . The full pipeline processes images in 280 ms on GPU and 928 ms on CPU, with the cryptographic stage requiring only 25 ms. For noisy channels, integrating Reed–Solomon forward error correction restores Structural Similarity Index Measure (SSIM) > 0.85 under Gaussian noise ($\sigma = 20$), making the framework suitable for edge-assisted IoT and real-time visual transmission across both reliable and noisy channel conditions.

KEYWORDS: Lightweight encryption; IoT security; image encryption; Cylinder XOR-Cascade (CXC); autoencoder

1 Introduction

The rapid growth of Internet of Things (IoT) systems, Unmanned Aerial Vehicles (UAVs), and cloud-based visual communication platforms has significantly increased the demand for secure and efficient image transmission. Images often contain sensitive information whose exposure during transmission or storage can lead to severe privacy breaches, making robust encryption essential for ensuring confidentiality and integrity of visual data. Traditional algorithms such as AES and RSA provide strong security but are computationally expensive and unsuitable for resource-constrained environments, motivating the development of lightweight encryption schemes that balance security with efficiency. Chaos-based and XOR-based methods have been widely explored for their simplicity and high-speed performance [1], while deep learning approaches have gained significant attention in image encryption. Autoencoders, in particular, demonstrate strong capability in learning compact latent representations suitable for dimensionality reduction before encryption [2]. Hybrid approaches integrating deep learning with classical cryptographic primitives have demonstrated strong resistance against statistical and differential attacks while maintaining high reconstruction quality [3],

with lightweight XOR-based cascade diffusion mechanisms further adopted for their effectiveness in destroying pixel correlations [4]. Despite these advances, methods relying on complex neural networks increase computational overhead, while those depending solely on chaotic systems fail to exploit learned feature representations, motivating a hybrid framework that combines deep learning with lightweight cryptographic operations. To address these challenges, this paper proposes CXC, a lightweight image encryption framework integrating a residual-based autoencoder with a XOR-based cascade encryption scheme. The autoencoder compresses the input image into a compact latent representation while preserving key visual features, and the encryption stage applies a cylinder-inspired XOR cascade to ensure strong confusion and diffusion, providing robust resistance against statistical and differential attacks.

Main Contributions—The key contributions of this research are:

- A novel CXC image encryption scheme is proposed, achieving strong diffusion with low computational and memory overhead.
- Images are standardized to 256×256 pixels and converted from TIFF to PNG, reducing storage requirements while preserving lossless quality and ensuring consistent data representation across the pipeline.
- A residual autoencoder-based reconstruction model is developed to accurately recover original images, improving robustness against noise and partial data loss.
- By operating on compressed latent representations, the framework achieves faster encryption and decryption compared to pixel-domain methods, making it highly efficient for resource-constrained IoT environments.

The remainder of the paper is organized as follows. [Section 2](#) covers preliminary concepts including IoT constraints, autoencoders, and image encryption techniques. [Section 3](#) reviews related work on encryption methods and latent-space image encryption. [Section 4](#) presents the proposed framework, including the autoencoder architecture and CXC encryption scheme. [Section 5](#) details the experimental setup, dataset, and analyses covering computational performance, statistical security, and differential attacks. [Section 6](#) compares the proposed method against existing works. [Section 7](#) concludes the paper.

2 Preliminary Concepts

IoT systems are characterized by limited computational resources, restricted memory, and low energy availability [5,6], posing significant challenges for secure image processing and transmission. Applications such as healthcare monitoring, UAV surveillance, and smart environments demand real-time secure transmission, where unsecured communication channels introduce significant security risks that motivate the development of lightweight encryption solutions [7]. However, conventional encryption approaches incur high computational overhead, making lightweight solutions essential for resource-constrained IoT systems. Autoencoders are neural networks designed for unsupervised learning of compact data representations [8], consisting of an encoder that compresses the input into a lower-dimensional latent space and a decoder that reconstructs the original data. By capturing the most relevant input features, autoencoders enable dimensionality reduction while preserving structural information, making them useful for compression, feature extraction, denoising, and anomaly detection [9], though some information loss during reconstruction is inevitable. Image encryption protects visual data from unauthorized access during storage and transmission [10], addressing the unique properties of images such as high redundancy and strong pixel correlation [11]. Encryption transforms the original image into an unintelligible form, recovered only upon authorized decryption, though balancing security strength with computational efficiency remains a key challenge in resource-constrained environments.

3 Literature Review

This section reviews image encryption studies published between 2022 and 2026, summarized in [Table 1](#). The literature is categorized into four groups: Lightweight and IoT-Friendly Encryption, High-Speed and High-Throughput Encryption, Advanced and Security-Focused Encryption, and AI-Driven and Learning-Based Encryption.

3.1 Lightweight and IoT-Friendly Encryption

In [5], the SED-UAV framework integrates a Lightweight, Block-based, and Improved Chaotic Image Encryption Algorithm (LB-ICA) with YOLO-LFP object detection for UAVs in 6G edge environments, achieving a key space exceeding 2^{100} , 20.3 ms encryption time per block, and 44.7% mAP@0.5 on Vis-Drone2019, with robustness against differential and chosen-plaintext attacks. In [12], a visual cryptography framework protects biometric IoT data using QR-code gray-level features to reduce reconstruction noise under lossy transmission. Ref. [6] proposes an enhanced AES scheme combining chaotic key generation, dual S-box structures, and bit-level permutation, achieving sub-millisecond execution and passing NIST randomness tests, though single-round key reuse may affect long-term robustness. For Industrial IoT, ref. [7] proposes a lightweight chaotic scheme performing permutation and diffusion in a single round with resistance against statistical and differential attacks, while ref. [13] presents a lightweight image encryption approach for IoT using scan-based block compression and selective pixel encryption, it reduces computational complexity and communication overhead, achieving average reductions of 15% in power consumption and 26% in packet rate compared with existing methods. On the compression-encryption front, ref. [14] combines compressive sensing with a seven-dimensional hyperchaotic map and SHA-512 key initialization for medical IoT, while [15] integrates a five-dimensional chaotic system, DNA computing, and Salsa20, achieving entropy close to 7.99 with strong differential attack resistance but without real-world IoT validation. Finally, ref. [16] employs nonlinear feedback shift registers and DNA encoding for lightweight encryption, though the absence of runtime analysis limits real-time suitability assessment.

3.2 High-Speed and High-Throughput Encryption

In [17], the Multiple-Image Encryption Algorithm (MIEA) encrypts image batches simultaneously by combining images into a cube, applying a 4D hyperchaotic Chen system for key generation, and using six novel S-boxes across all RGB channels. It achieves 96.4 Mbps encryption rate, key space of 2^{4624} , entropy of 7.999, NPCR = 99.62%, UACI = 33.3%, and NIST compliance, making it suitable for real-time industrial transmission. In [18], a one-dimensional fractional chaotic map paired with parallel DNA coding and SHA-3 hashing achieves resistance to chosen-plaintext attacks with lower computational overhead and robustness against noise. While both exploit parallelism, MIEA prioritizes security through complex key design whereas the fractional chaotic approach favors efficiency, highlighting a security-speed trade-off that motivates hybrid solutions.

3.3 Advanced and Security-Focused Encryption

In [19], a multi-layer scheme for sensitive applications such as satellite imagery integrates hyperchaotic systems, SVD, RC5, a chaos-based Hill cipher, and a BBS-based S-box, merging images into an augmented structure to resist traffic analysis, though the multiple cryptographic components introduce high complexity that limits real-time use. In [1], a recoverable image-mosaic technique achieves perceptual security using small-sized keys and a DCT-based similarity metric without full encryption, improving key management but offering weaker protection against sophisticated attacks. Similarly, ref. [20] proposes a Thumbnail-Preserving Encryption (TPE) scheme retaining partial visual semantics via cross-plane substitution and permutation,

though the preserved visual information risks leakage in high-security contexts. For stronger cryptographic guarantees, ref. [21] combines chaotic maps, ECC, and genetic algorithms, achieving entropy of 7.99, near-zero correlation, and a key space of 2^{511} , at the cost of increased computational overhead. Ref. [22] implements confusion and diffusion via multiple chaotic maps and the KAA map following Shannon's principles, passing NIST SP 800 tests, though parameter sensitivity may affect practical stability. Finally, ref. [23] presents a hyperchaos and DNA-based scheme with fine-grained encoding, a novel DNA subtraction operation, and pixel-level crossover, enhancing the avalanche effect and noise resistance, though at higher computational cost due to multi-stage DNA operations.

3.4 AI-Driven and Learning-Based Encryption

In [3], a color image encryption scheme combines a convolutional autoencoder with DNA encoding and chaotic systems, compressing images from $N \times M \times 3$ to $P \times Q$ before encryption via DNA operations, chaotic sequences, and S-boxes, achieving ~97% training and ~95% validation accuracy with strong resistance to statistical and differential attacks. In [24], a Convolutional Neural Network (CNN) autoencoder paired with Henon and Ikeda chaotic maps compresses images from 256×256 to 128×128 before multi-stage chaotic confusion and diffusion, achieving NPCR = 99.65%, UACI = 33.65%, and Peak Signal-to-Noise Ratio (PSNR) = 33.72 dB. In [25], MAN-C integrates masked autoencoders with neural cryptography combining tree parity machine and Shamir's secret sharing for secure medical image sharing, achieving improved Root Mean Square Error (RMSE) on TCIA CT scan datasets with strong resistance to data leakage. In [2], an IoT-based framework applies a convolutional autoencoder for compression prior to encryption in edge computing, achieving Mean Absolute Error (MAE) = 0.221 and classification accuracy of 75%, though performance varies with architectural choices, underscoring hyperparameter sensitivity.

While the above studies report strong performance using metrics such as PSNR, NPCR, UACI, and RMSE, they are evaluated under high-resource assumptions and do not fully reflect real IoT constraints. These metrics alone are insufficient for IoT feasibility assessment, where memory, energy, latency, and communication overhead are critical. In particular, the convolutional autoencoder-based methods in [3,24] incur high computational and memory costs due to large model sizes and multi-stage pipelines (compression, chaotic maps, DNA encoding, and diffusion), making them unsuitable for constrained IoT devices. Similarly, ref. [25] relies on multi-party cryptographic synchronization, introducing communication overhead that is inefficient for low-bandwidth IoT networks. Although [2] considers an edge setting, it still highlights sensitivity to model design, reinforcing the need for lightweight architectures. Overall, there remains a gap between strong cryptographic performance and practical IoT feasibility, motivating resource-aware encryption design.

Table 1: Comparative analysis of related work.

Study	Domain	Dataset/Images	Encryption Method	Feature Representation Technique	Limitation
[24]	IoT	USC-SIPI dataset	Hybrid (Autoencoder + Chaotic Maps + CNN-based Encryption)	Learned Latent Representation (Autoencoder-based dimensionality reduction from 256×256 to 128×128)	Potential increase in computational overhead due to the integration of Autoencoder, CNN, chaotic maps, and multi-stage processing.
[5]	UAV/6G Edge Computing	VisDrone2019 dataset	Lightweight Block-Based Chaotic (LB-ICA)	Raw Pixels + Block-based + Chaotic (PLSM)	- No compression stage, higher communication overhead. - Separate encryption and detection pipelines (no end-to-end design)
[17]	Industrial Informatics/Batch Image Transmission	Standard images: Fl6, Girl, House, House2, Lena, Mandrill, Peppers, Sailboat, Tree	Hyperchaotic 4D Chen + Mersenne Twister + 6 S-boxes	Raw pixels (bit-stream/image cube representation)	- High memory consumption due to image cube formation - Key management overhead (large key space)
[19]	Satellite Imagery/National Security	MAR20 dataset	Hyperchaotic systems + SVD + RC5 (counter mode) + Chaotic Hill Cipher + BBS S-box	Raw pixels/spatial domain encryption (block-based + chaotic + matrix-based operations)	Multi-layer pipeline introduces high computational overhead.
[1]	IoT	USC-SIPI dataset	Bit-level circular shift + XOR (DCT-CSIM metric)	Raw pixel/bit-level spatial encryption (mosaic scrambling method)	- Weaker cryptographic security than full encryption - No diffusion mechanism
[12]	Biometric IoT/Customer IoT	NA	Visual Cryptography (VC) + QR-code embedding + GAN-based image recovery (HGAN)	Converts image to mathematical structure to encrypted image	- Lossy decryption degrades image quality. - High computational cost (GAN training not IoT suitable)
[20]	General/Industrial Informatics	Helen dataset	Cross-Plane Substitution-Permutation Network (SPN) + PRF-driven encryption	Deterministic encrypted feature space (SPN-based)	Intentionally preserves partial visual content (thumbnail), inherently leaking coarse image semantics; not suitable for applications requiring full confidentiality
[6]	IoT/Edge-Fog-Cloud	NA	Enhanced AES + Chaotic Key Generation (novel map + 3D Lorenzo) + Dual Dynamic S-boxes + Bit-level Circular Permutation	Raw pixels (block cipher substitution/permutation on the byte values)	Not validated for large images or video

(Continued)

Table 1 (continued)

Study	Domain	Dataset/Images	Encryption Method	Feature Representation Technique	Limitation
[21]	General	NA	Hybrid image encryption scheme based on Arnold's Cat Map, Elliptic Curve Cryptography (ECC), and Genetic Algorithm optimization	Raw pixel (Arnold + ECC + GA)	• High computational complexity due to ECC and genetic algorithm integration • Increased processing time from GA-based key optimization • Not well-suited for real-time or low-power devices
[7]	General	Standard benchmark test images (Baboon, Lena, Barbara, Cameraman, and Peppers)	A hybrid chaotic image encryption scheme based on a 1-D perturbed logistic map and cyclic construction model	Raw pixel (CPLM-based algorithm operates directly on pixel intensity values (0–255 per channel) without any prior transformation.)	• High computational complexity due to multiple chaotic and diffusion steps • Increased memory usage for storing permutation and chaotic matrices • Not fully suitable for very low-resource Industrial Internet of Things (IIoT) devices
[13]	IoT	Images (Drop, Airplane, Lena, Peppers, House, Sailboat, Baboon)	Scan-based block compression and selective pixel encryption	Selective pixels (significant and insignificant pixel representation)	• Preprocessing overhead: Extra stages (pixel extraction, permutation, compression, selective encryption) add computational cost. • Block-size sensitivity: Execution time doesn't always drop with larger blocks — sizes 4 and 8 outperform 2 and 16.
[14]	Medical	Biomedical images	Compressive sensing + modified 7D hyperchaotic map + NSCT-based transform + permutation–diffusion encryption	Transform domain via NSCT + raw pixel representation	• High computational complexity (NSCT + compressive sensing + chaos) • Time-consuming for large images • Complex key generation (MSD hyperchaotic + SHA-512)
[15]	General	Standard images (Lena, Baboon, and Pepper)	5D chaotic system + DNA computing + Salsa20 stream cipher + S-Box substitution	Raw pixel-based + S-Box/DNA transformation + keystream mixing	• High computational complexity due to multiple combined modules (5D chaos, Salsa20, DNA operations) • Increased encryption/decryption time because of many iterative rounds (e.g., 20–32 rounds) • Limited suitability for real-time or resource-constrained environments

(Continued)

Table 1 (continued)

Study	Domain	Dataset/Images	Encryption Method	Feature Representation Technique	Limitation
[16]	General	Standard images (Lena, Baboon, Boat, and Pepper)	Nonlinear Feedback Shift Register (NLFSR) + DNA computation + Cipher Block Chaining (CBC) mode	Raw pixels (with DNA-based transformation)	• Time-consuming for large images • Complex key management (k1, k2, k3 + IV + PRNs)
[22]	General	Standard images (Lena, Mandrill, and Peppers)	KAA map + multiple chaotic maps (Logistic Sine map, LCG, Tent map, Bernoulli map) + bit-level confusion and diffusion	Raw pixels (bit-level representation)	• High computational complexity (multiple chaotic maps + XOR operations) • Multi-stage processing overhead • Time-consuming for large images • Complex key generation (multiple chaotic + LCG components)
[23]	General	Standard images (Lena)	Hyperchaos system + DNA encoding + DNA subtraction operation + pixel-based DNA crossover + left/right diffusion + XOR-based diffusion	Raw pixels (with DNA-based encoding)	• High complexity • Slow for large images
[18]	General	Standard test images datasets, CVG-UGR and USC-SIP	1D fractional chaotic map (sine map + fractional operation) + parallel DNA coding + SHA-3 hash-based key generation	Raw pixels (with DNA encoding transformation)	• High complexity • Slow for large images
[3]	General	Kodak dataset	Convolutional autoencoder (for dimensionality reduction) + DNA encoding + multiple chaotic sequences + substitution boxes (S-boxes)	Latent space representation	• The autoencoder introduces reconstruction loss, affecting fine image details.
[25]	Medical	CT scan images dataset	Masked Autoencoder (MAE) + Neural Cryptography (Tree Parity Machine) + Shamir Secret Sharing scheme	Latent space representation via a masked autoencoder	• The autoencoder introduces reconstruction loss, affecting fine image details.

(Continued)

Table 1 (continued)					
Study	Domain	Dataset/Images	Encryption Method	Feature Representation Technique	Limitation
[2]	IoT medical	NIH Chest X-ray dataset	Convolutional autoencoder-based image compression + encryption + decoder reconstruction (CNN-based secure transmission framework)	Latent space representation	• The encryption is lightweight and not cryptographically strong, relying on simple mathematical operations. • The autoencoder introduces reconstruction loss, affecting fine image details.
Proposed Approach	IoT	USC-SIPI dataset	CXC encryption + autoencoder-based image representation (lightweight encryption framework)	Autoencoder-based latent feature representation	—

3.5 Research Gap and Novelty

[Table 1](#) summarizes existing image encryption methods and reveals several critical gaps: reliance on raw pixel-domain encryption causing high computational and communication overhead; reconstruction loss in autoencoder-based approaches; excessive complexity from chaotic systems and multi-stage operations that limits IoT suitability; key management and memory overhead from parallel chaotic processing; limited cross-platform scalability in lightweight techniques; and predominantly simulation-based evaluations lacking real-world energy and latency measurements. These gaps collectively motivate a latent-feature-based encryption approach combining geometric transformations and block-level processing as a computationally efficient alternative with strong privacy guarantees for real-world IoT deployment. To address these limitations, this work proposes a lightweight image encryption framework built on a CXC mechanism tailored for resource-constrained IoT environments. A residual autoencoder preprocesses the image, preserving quality with minimal reconstruction loss by capturing global semantic structure in the latent space while an external residual branch retains fine-grained spatial details (edges, textures) lost during compression. CXC encryption is then applied independently to both the latent representation and residual features, with the decoder reconstructing the image at high perceptual fidelity (SSIM > 0.95), enabling secure and efficient IoT image transmission.

4 Methodology

4.1 Framework Overview

The overall architecture of the proposed framework is illustrated in [Fig. 1](#). The system integrates a residual-based pretrained autoencoder with a lightweight encryption mechanism to securely compress, encrypt, and reconstruct images, targeting high-quality (near-lossless) reconstruction while protecting intermediate representations during transmission or storage. The input is a fixed-size RGB or grayscale image of 256×256 pixels, which is first processed by the autoencoder encoder to produce a compact latent representation z . Simultaneously, an intermediate encoder feature map is extracted as a residual (skip) representation s , preserving fine-grained spatial details essential for accurate reconstruction. The framework operates through two parallel branches on both the transmitter and receiver sides. On the transmitter side, both z and s are independently encrypted using the proposed CXC method, producing corresponding ciphertexts and ensuring that neither the compressed representation nor the auxiliary residual is exposed in plain form. On the receiver side, the inverse CXC process recovers the approximated representations $\tilde{z}$ and $\tilde{s}$, which are fused via a bitwise XOR operation and passed through the decoder to generate the final reconstructed image. It is important to distinguish the roles of the autoencoder and the cipher. The autoencoder serves solely as a dimensionality-reduction and bandwidth-saving module and provides no cryptographic security; confidentiality is guaranteed entirely by CXC encryption of both z and s . While the latent representation offers a secondary obscurity benefit—an attacker who breaks the cipher obtains a compressed feature tensor rather than a human-interpretable image—this is not a formal security guarantee and must not be relied upon as such. [Table 2](#) summarizes the notation used in this section.

4.2 Autoencoder Stage

4.2.1 Architecture and Design

The proposed framework employs a pretrained lightweight Variational Autoencoder (LightVAE) [26] as the core feature extractor, adapted for single-image processing due to its strong compression capability and low computational overhead. The autoencoder is used strictly for compression and feature representation, while cryptographic security is exclusively provided by the CXC encryption stage. Its encoder employs depthwise-separable convolutions and a shallow bottleneck ($1 \times 4 \times 64 \times 64$ latent shape for 256×256

inputs), yielding approximately 3.2M parameters and 1.8 GFLOPs per encode–decode cycle, mapping $I \in \mathbb{R}^{256 \times 256 \times 3}$ into a compact latent vector z encoding global semantic structure suitable for efficient transmission. As a novel design contribution, a residual representation s is explicitly extracted from intermediate encoder features—external to the pretrained model—to capture fine-grained spatial details such as edges, textures, and high-frequency components partially lost during compression. Together, z and s preserve complementary information at different abstraction levels, enabling high-fidelity reconstruction, and are independently encrypted via the CXC mechanism prior to transmission. Autoencoder compression introduces a small but measurable reconstruction error, as the latent bottleneck discards high-frequency information; the residual s compensates for this loss. The combined latent-residual reconstruction achieves $\text{SSIM} > 0.95$, indicating excellent perceptual quality suitable for IoT transmission. Since CXC is XOR-based (involutive), it introduces no information loss; all reconstruction error originates solely from the compression stage, making the end-to-end pipeline near-lossless. To validate the backbone choice, LightVAE is compared against a standard 4-layer CNN autoencoder (CNN-AE) and a deeper ResNet-based autoencoder (Res-AE) on the USC-SIPI dataset (Table 3). LightVAE achieves the best latency–quality trade-off ($\text{SSIM} = 0.96$, encoding latency < 90 ms on NVIDIA T4): CNN-AE is slightly faster but suffers from visible blocking artifacts ($\text{SSIM} = 0.91$), whereas Res-AE provides only marginal quality improvement at $4\times$ more parameters and $2.3\times$ higher latency, confirming LightVAE as the most suitable backbone for IoT-oriented latent-space encryption.

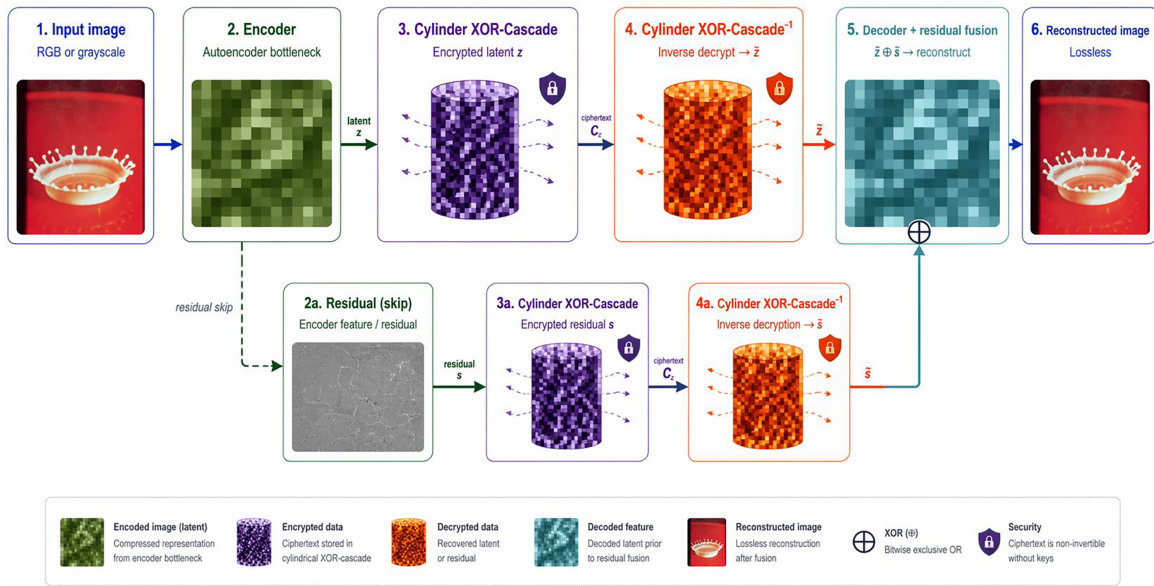


Figure 1: Methodology process flow.

4.2.2 Ablation Study

To isolate the contribution of each architectural component, three configurations are evaluated on the USC-SIPI dataset under noise-free conditions: (A) the full proposed pipeline (LightVAE + residual + CXC); (B) LightVAE without the residual branch (latent-only reconstruction); and (C) a standard CNN autoencoder with residual branch. Table 4 reports reconstruction quality and computational cost. Removing the residual branch (B) causes a 4.3 dB PSNR drop and 0.08 SSIM degradation, confirming that the residual stream is essential for recovering high-frequency details lost during latent compression. Replacing LightVAE with CNN-AE (C) reduces latency but degrades reconstruction quality by 1.7 dB PSNR, validating LightVAE as

the optimal backbone. The CXC encryption stage is identical across all configurations and does not affect reconstruction quality, as XOR is information-preserving.

Table 2: Notation summary.

Symbol	Description
$I \in \{0, \dots, 255\}^{H \times W \times C}$	Input image (RGB or grayscale)
$z \in \mathbb{R}^{d_z}$	Latent vector from encoder bottleneck
$s \in \mathbb{R}^{d_s}$	Residual feature map (fine details)
$w, n = \lfloor W/w \rfloor$	Column width; number of vertical columns
$P_k, C_k^{(i)}$	Plaintext/ciphertext column k , pass $i \in \{1, 2\}$
σ	Hash seed: secret key $\parallel$ pass ID $\parallel$ chaining seed
$s_k^{(i)}, K_k^{(i)}$	Chaining seed/keystream for column k , pass i
K	256-bit secret encryption key
$N, \mathcal{S}$	128-bit nonce; 128-bit KDF salt
$\oplus, \parallel$	Bitwise XOR; concatenation

Table 3: Autoencoder backbone comparison (256×256 RGB images, NVIDIA T4 GPU).

Model	Params	FLOPs	Enc. (ms)	Dec. (ms)	PSNR (dB)	SSIM
CNN-AE	1.8M	0.9G	62.1	98.4	28.4	0.91
Res-AE	12.4M	4.2G	201.3	312.7	32.1	0.97
LightVAE	3.2M	1.8G	87.0	142.1	31.5	0.96

Table 4: Ablation study: component contribution analysis.

Configuration	Enc. (ms)	Dec. (ms)	PSNR (dB)	SSIM	Params
Full (LightVAE + Res + CXC)	89.5	142.1	31.5	0.96	3.2M
LightVAE only (no Res)	87.0	138.4	27.2	0.88	3.2M
CNN-AE + Res + CXC	62.1	98.4	29.8	0.92	1.8M

4.3 Proposed Encryption Method

The proposed encryption scheme, CXC, is a lightweight two-pass XOR-based column-wise stream cipher with sequential inter-column dependency, operating directly on pixel intensities by partitioning the image into fixed-width vertical columns across two sequential passes.

4.3.1 Key Components

Keystream Generation: A deterministic pseudo-random keystream is generated using SHA3-256 in counter mode. Given a concatenated seed $\sigma = \text{key} \parallel \text{pass_label} \parallel s$, the keystream is constructed as in Eq. (1), where ctr_i is a 4-byte incrementing counter and s is an intermediate chaining seed. The output is truncated and reshaped to match the exact byte count required for each image column.

$$K = \text{SHA3-256}(\sigma \parallel ctr_0) \parallel \text{SHA3-256}(\sigma \parallel ctr_1) \parallel \dots \quad (1)$$

Column Segmentation: Given an input image $I \in \{0, \dots, 255\}^{H \times W \times C}$, it is divided into $N = \lfloor \frac{W}{w} \rfloor$ non-overlapping vertical columns of width w pixels. If W is not divisible by w , the image is cropped to $N \cdot w$;

in practice, all images are resized to 256×256 with w chosen to divide W exactly, eliminating the need for cropping.

4.3.2 Encryption Process

Fig. 2 presents the step-by-step encryption process of the proposed CXC scheme, where two sequential passes are performed from left to right.

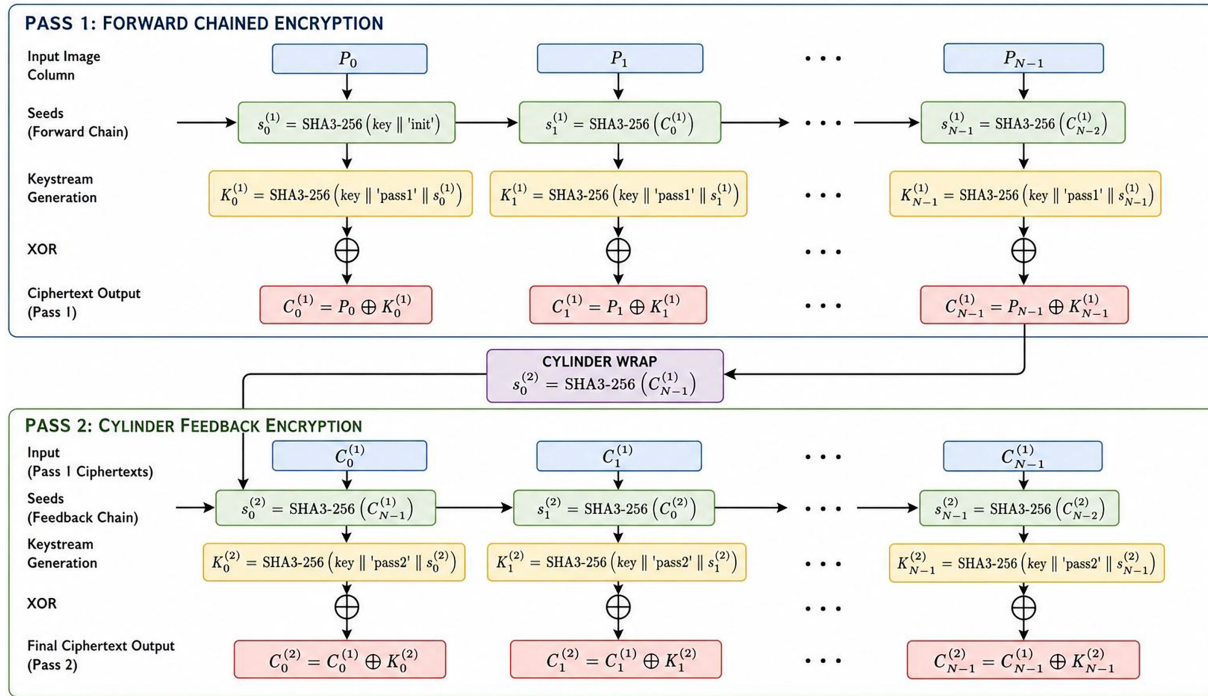


Figure 2: Detailed encryption workflow of the proposed CXC scheme.

Pass 1: Forward Chained Encryption: The initial chaining seed is defined as:

$$s_0^{(1)} = \text{SHA3-256}(\text{key} \parallel \text{"init"}) \quad (2)$$

For each column $k \in \{0, \dots, N-1\}$, the recursive seed update (3), keystream generation (4), and ciphertext formation (5).

$$s_k^{(1)} = \text{SHA3-256}(C_{k-1}^{(1)}), \quad C_{-1}^{(1)} = s_0^{(1)} \quad (3)$$

$$K_k^{(1)} = \text{SHA3-256}(\text{key} \parallel \text{"pass1"} \parallel s_k^{(1)}) \quad (4)$$

$$C_k^{(1)} = P_k \oplus K_k^{(1)} \quad (5)$$

Pass 2: Cylinder Feedback Encryption: The second pass introduces global diffusion via a cylinder-like seed feedback mechanism. The initial seed is derived from the last ciphertext column of Pass 1 as defined in Eq. (6) while the seed update, keystream generation, and ciphertext formation are defined in Eqs. (7)–(9).

$$s_0^{(2)} = \text{SHA3-256}(C_{N-1}^{(1)}) \quad (6)$$

This represents a single wrap-around dependency at the seed level only. For each column $k \in \{0, \dots, N-1\}$:

$$s_k^{(2)} = \text{SHA3-256}(C_{k-1}^{(2)}) \quad \text{with } C_{-1}^{(2)} = s_0^{(2)} \quad (7)$$

$$K_k^{(2)} = \text{SHA3-256}(\text{key} \parallel \text{"pass2"} \parallel s_k^{(2)}) \quad (8)$$

$$C_k^{(2)} = C_k^{(1)} \oplus K_k^{(2)} \quad (9)$$

The final ciphertext is given by the set $\{C_k^{(2)}\}$. Fig. 3 illustrates the step-by-step encryption process of the image of class aerials.

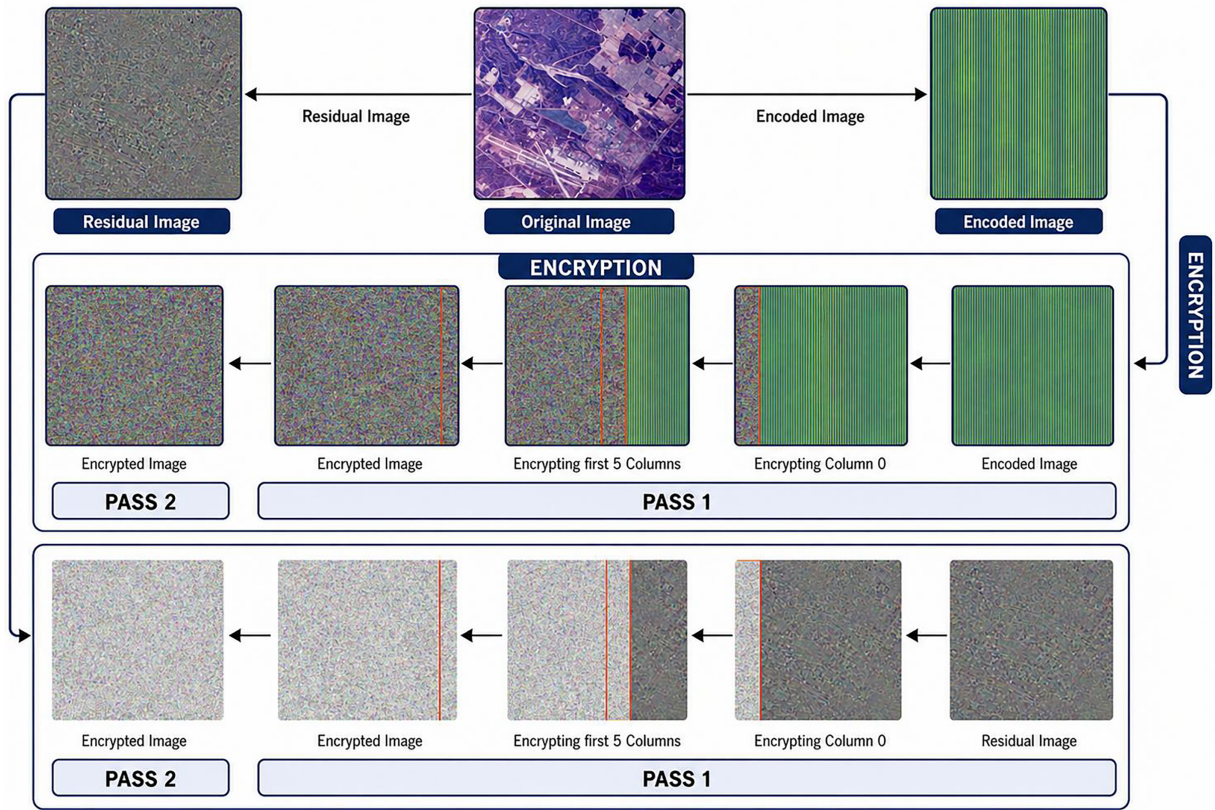


Figure 3: Encrypting image using the encoder-residual and CXC encryption.

4.3.3 Decryption Process

Decryption reverses both encryption passes in reverse order using the same deterministic keystream generation mechanism. During encryption, per-column seeds are generated and stored for both Pass 1 and Pass 2 to ensure deterministic keystream reconstruction during the decryption process. Fig. 4 presents the step-by-step decryption process of the proposed CXC scheme. The detailed operations of Inverse Pass 2 and Inverse Pass 1 are given in Eqs. (10) and (11), respectively.

- **Inverse Pass 2 (right → left):** for each column k , retrieve the stored seed $s_k^{(2)}$:

$$C_k^{(1)} = C_k^{(2)} \oplus K_k^{(2)} \quad (10)$$

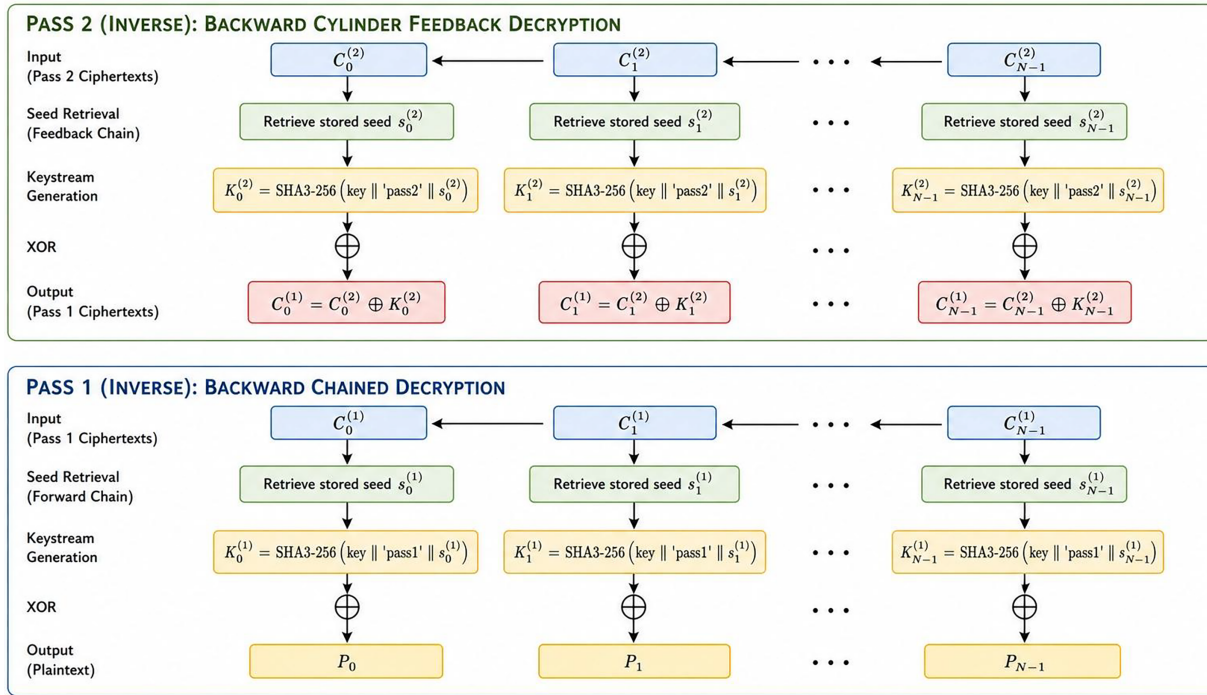


Figure 4: Detailed decryption workflow of the proposed CXC scheme.

- **Inverse Pass 1 (right $\rightarrow$ left):** for each column k , retrieve the stored seed $s_k^{(1)}$:

$$P_k = C_k^{(1)} \oplus K_k^{(1)} \quad (11)$$

Since XOR is involutive $(X \oplus K) \oplus K = X$, perfect reconstruction of the original image is guaranteed.

Fig. 5 illustrates the step-by-step decryption process of the image of class aerials.

4.3.4 Key Management and Session Nonce

Raw passwords are never used directly as encryption keys. Instead, a 256-bit key K is derived via a KDF: either PBKDF2-HMAC-SHA256 (100,000 iterations, 128-bit salt S) or Argon2id (64 MB memory, 3 iterations, parallelism = 4) for stronger resistance against GPU-based brute-force attacks.

Each session uses a fresh 128-bit nonce N from a cryptographically secure random number generator, with the initial seed computed as Eq. (12).

$$s_0^{(1)} = \text{SHA3-256}(K \parallel N \parallel \text{'init'}). \quad (12)$$

The nonce N is transmitted in plaintext alongside the ciphertext; without K , it reveals nothing about the keystream. Reusing (K, N) across distinct plaintexts would allow an adversary to recover $P \oplus P'$ by XORing ciphertexts, so nonce uniqueness is strictly enforced per session. To prevent replay attacks and tampering, a MAC is appended as in Eq. (13); the receiver recomputes the tag and discards any frame with a mismatch, ensuring integrity.

$$\text{Tag} = \text{HMAC-SHA3-256}(K, C^{(2)} \parallel N). \quad (13)$$

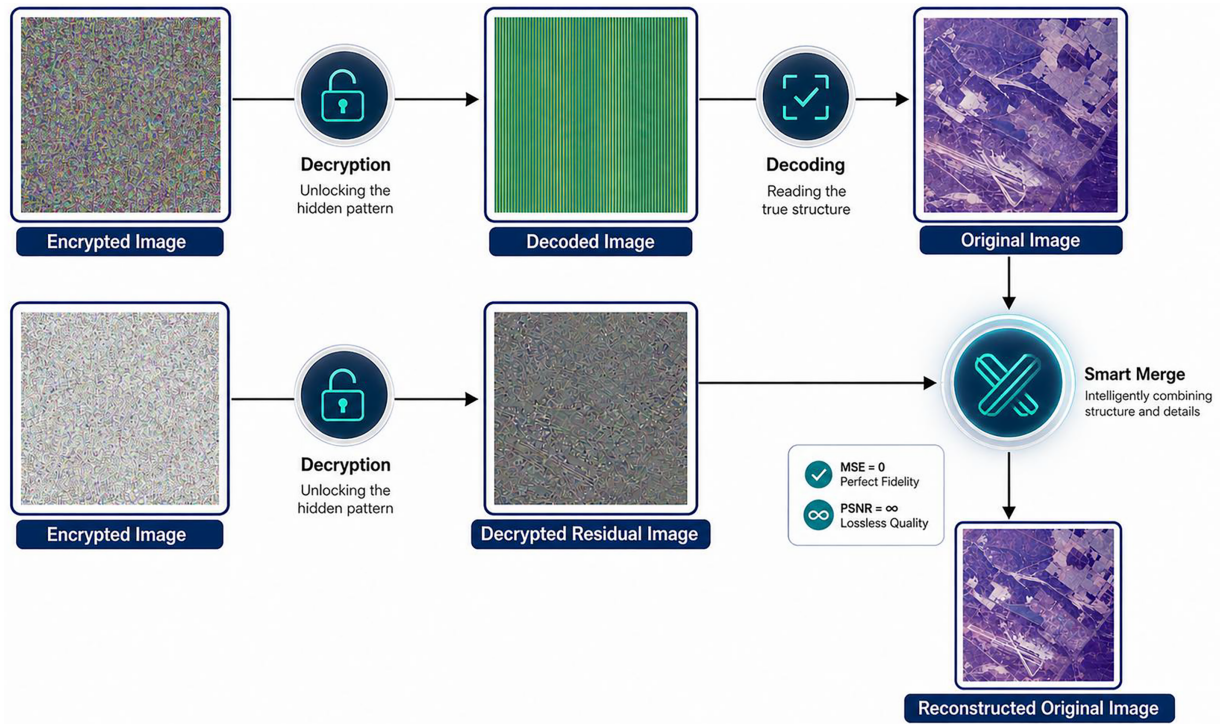


Figure 5: Reconstructed image using encoder-residual and CXC decryption process.

4.3.5 Comparison with Conventional Stream Ciphers

Table 5 contrasts CXC with widely adopted stream ciphers. Unlike RC4 or ChaCha20, which generate keystream independently of plaintext, CXC employs a ciphertext-dependent chaining mechanism. This creates a non-linear diffusion path: a single-bit plaintext change alters the entire seed chain via the cylinder wrap-around, a property not present in simple counter-mode or LFSR-based designs. The per-column SHA3-256 derivation also provides a 256-bit security level and inherent avalanche, whereas lightweight chaotic ciphers often rely on 32-bit or 64-bit state variables vulnerable to statistical reconstruction.

Table 5: Qualitative comparison of CXC with standard stream ciphers.

Property	RC4	ChaCha20	Chaotic XOR	CXC
Keystream dependency	Key only	Key+Nonce	Key+Chaos params	Key+Nonce+Ciphertext chain
Global diffusion	No	No	Limited (permutation)	Yes (cylinder wrap)
Key space	$2^{128-2048}$	2^{256}	2^{32-64}	2^{256}
Per-column avalanche	N/A	N/A	Weak	Strong (SHA3-256)

Note: N/A = Not Applicable.

5 Experimental Results

5.1 Experimental Setup

The experiments were conducted using Google Colab with Python 3.12.13. The implementation was executed on a T4 GPU for accelerated computation. The framework was developed using Python-based deep learning libraries, including PyTorch and HuggingFace Transformers (version 5.7.0). The residual feature s is extracted from the third encoder layer of LightVAE (extttencoder.conv3) before the final downsampling

operation, producing a tensor of shape $64 \times 64 \times 64$. This tensor is quantized to 8-bit unsigned integers via per-channel min-max scaling and flattened into a 1D byte stream for CXC encryption. All experiments use PyTorch 2.2.0, CUDA 12.2, and HuggingFace Transformers 5.7.0. The source code and pre-trained weights will be made publicly available upon acceptance.

5.2 Dataset

The experiments were conducted using the USC-SIPI dataset [27], a standard benchmark in image processing containing four categories: Textures, Aerials, Miscellaneous (e.g., Mandrill and Peppers), and Sequences (e.g., moving objects and dynamic scenes), totalling 210 images (38 Aerial, 39 Miscellaneous, 69 Sequence, and 64 Texture). Detailed characteristics are presented in Table 6. The original TIFF images span resolutions from 256×256 to 2250×2250 pixels in grayscale and RGB formats; for consistency, all images were converted to PNG and resized to $256 \times 256 \times 3$ RGB, ensuring a standardized input structure for the proposed framework.

Table 6: USC-SIPI dataset distribution and image characteristics.

Category	Total	RGB	Gray Scale	Sizes Distribution
Textures	64	0	64	$512 \times 512 \rightarrow 39, 1024 \times 1024 \rightarrow 25$
Sequences	69	0	69	$256 \times 256 \rightarrow 59, 512 \times 512 \rightarrow 10$
Aerials	38	37	1	$512 \times 512 \rightarrow 12, 1024 \times 1024 \rightarrow 25, 2250 \times 2250 \rightarrow 1$
Miscellaneous	39	14	25	$256 \times 256 \rightarrow 14, 512 \times 512 \rightarrow 22, 1024 \times 1024 \rightarrow 3$
Total	210	51	159	256×256 to 2250×2250

5.3 Security Model and Heuristic Cryptanalysis

5.3.1 Attacker Model

The security of the CXC scheme is analyzed under three standard adversarial models:

- **Ciphertext-Only Attack (COA):** The adversary $\mathcal{A}$ observes only the ciphertext stream $\{C_k^{(2)}\}$.
- **Known-Plaintext Attack (KPA):** $\mathcal{A}$ has access to one or more plaintext-ciphertext pairs $(P_k, C_k^{(2)})$ generated under the same secret key.
- **Chosen-Plaintext Attack (CPA):** $\mathcal{A}$ can adaptively query an encryption oracle $\mathcal{E}_K(\cdot)$ and obtain corresponding ciphertexts.

SHA3-256 is modeled as a random oracle $\mathcal{H}$, and the secret key $K \in \{0, 1\}^{256}$ is assumed to be uniformly random. A fresh 128-bit nonce N is concatenated to the initial seed for each encryption session; reuse of the pair (K, N) is strictly prohibited (see Section 4.3.4).

5.3.2 Keystream One-Wayness (Lemma 1)

Given any ciphertext column $C_k^{(i)}$ and its corresponding plaintext P_k , the keystream segment $K_k^{(i)} = P_k \oplus C_k^{(i)}$ can be computed by an adversary. However, recovering the secret key K from $K_k^{(i)}$ requires inverting, where Eq. (14) represents a one-way cryptographic hash function and $s_k^{(i)}$ is derived from a ciphertext-dependent hash chain. Under the random oracle assumption, finding a valid preimage $(K, s_k^{(i)})$ for a 256-bit output requires approximately $\mathcal{O}(2^{256})$ operations, rendering key recovery computationally infeasible.

$$K_k^{(i)} = \mathcal{H}(K \parallel \text{pass}_i \parallel s_k^{(i)}) \quad (14)$$

5.3.3 Propagation under CPA (Lemma 2)

Let P and P' differ in a single bit at column j . During Pass 1, the seed chain $s_k^{(1)}$ for all $k > j$ depends on $C_j^{(1)}$ Eq. (15): a one-bit change in P_j alters $C_j^{(1)}$ and propagates through all subsequent seeds and keystream values. In Pass 2, the cylinder wrap-around feeds $C_{N-1}^{(1)}$ into $s_0^{(2)}$, spreading the disturbance across all ciphertext positions and producing a global avalanche effect. Experimentally, this yields NPCR $> 99.4\%$ and UACI $\approx 33.4\%$.

$$C_j^{(1)} = P_j \oplus K_j^{(1)}. \quad (15)$$

The above analysis provides heuristic evidence of resistance against KPA and CPA through strong diffusion and keystream irreversibility, but does not constitute a formal IND-CPA security proof. CXC should therefore be regarded as a lightweight, empirically validated scheme for latent-space image protection rather than a general-purpose or standards-grade encryption primitive.

5.4 Computational Analysis

5.4.1 Computational Time Analysis

To evaluate computational efficiency, one image from each dataset class (Textures, Aerials, Miscellaneous, and Sequences) was tested after resizing to 256×256 , executing the full pipeline including autoencoder encoding, residual extraction, CXC encryption/decryption, and decoding. As shown in Table 7, encoding ranges from 86.79 to 97.24 ms, residual extraction remains minimal (1.01–1.77 ms), CXC encryption and decryption require 23.49–28.25 ms and 21.82–23.45 ms respectively, and decoding is the most time-consuming stage (128.02–150.98 ms) due to autoencoder complexity. Overall execution time ranges from 265.22 to 289.75 ms; minor variations across classes reflect image content differences, confirming the framework's efficiency and suitability for resource-constrained environments.

Table 7: Computational time (ms) analysis of the proposed framework.

Image Type	Encoding (Autoencoder)	Residual Extraction	Encryption (CXC)	Decryption	Decoding	Total Time
Textures	86.89	1.02	23.80	21.82	148.73	282.26
Aerials	87.01	1.12	25.62	23.45	128.02	265.22
Miscellaneous	97.24	1.77	23.49	22.03	140.59	285.12
Sequences	86.79	1.01	28.25	33.4	150.98	289.75
Total Average	89.48	1.23	25.29	22.51	142.08	280.59

To assess feasibility on true edge hardware, the paper evaluates the full pipeline on three platforms: (1) NVIDIA T4 GPU (Google Colab), (2) Intel Core i5-10400 CPU @ 2.9 GHz, and (3) Raspberry Pi 4 (ARM Cortex-A72, 1.5 GHz, 4 GB RAM). Table 8 reports the results. While the VAE dominates latency across all platforms, the CXC encryption remains lightweight (<120 ms even on the Raspberry Pi). This confirms that CXC is suitable for microcontroller-class devices, whereas the VAE stage should be offloaded to an edge gateway or an embedded AI accelerator (e.g., Google Coral TPU, Intel NCS2) in a realistic IoT deployment.

Table 8: Cross-platform execution time (ms) per 256×256 image.

Stage	T4 GPU	i5 CPU	Raspberry Pi 4
VAE Encoding	89.5	312.4	1240.2
Residual Extract	1.2	4.1	18.7
CXC Encryption	25.3	58.6	118.4
CXC Decryption	22.5	54.2	112.9
VAE Decoding	142.1	498.3	2103.5
Total	280.6	927.6	3593.7

5.4.2 Big-O Analysis

The proposed CXC scheme operates on images of size $H \times W \times C$ ($256 \times 256 \times 3$). Each of the two sequential passes processes every pixel exactly once via XOR and keystream generation, giving $T_1 = T_2 = O(H \times W \times C)$, and a total complexity of $T_{\text{total}} = T_1 + T_2 = O(2 \cdot H \times W \times C)$, which simplifies to $T_{\text{total}} = O(H \times W \times C)$. Therefore, CXC exhibits linear time complexity in the general case and constant time complexity for fixed-size images.

5.4.3 Throughput Analysis

[Table 9](#) summarizes the dataset used in the experiments: 210 images with a total size of 39.38 MB, sufficient to evaluate computational performance under realistic conditions. [Table 10](#) presents throughput analysis across all major pipeline stages. VAE encoding and decoding are the most computationally expensive stages, achieving 20.60 and 7.22 images/sec respectively, due to deep neural network operations. The proposed CXC encryption and decryption achieves significantly higher throughput at 40.83 and 44.20 images/sec, confirming that lightweight hashing and XOR operations introduce minimal overhead. Residual computation achieves the highest throughput at 2206.58 images/sec, as it involves only a pixel-wise difference without neural network inference or cryptographic transformation, resulting in negligible overhead relative to other pipeline components.

Table 9: Dataset statistics.

Metric	Value
Total Images	210
Total Size	39.38 MB

Table 10: Performance and throughput analysis.

Operation	Time (s)	Img/s	MB/s
VAE Encoding	10.20	20.60	3.86
VAE Decoding	29.09	7.22	1.35
CXC Encryption	5.14	40.83	7.66
CXC Decryption	4.75	44.20	8.29
Residual Enc/Dec	0.095	2206.58	413.73

5.5 Statistical Analysis

5.5.1 Histogram Analysis

Histogram analysis evaluates pixel intensity distributions to assess the removal of statistical patterns across four image categories: aerial (Fig. 6), miscellaneous (Fig. 7), sequence (Fig. 8), and texture (Fig. 9). Original images exhibit highly non-uniform histograms reflecting strong spatial correlations, whereas encrypted images show a nearly uniform distribution across all intensity levels, confirming effective confusion and diffusion with improved resistance to statistical attacks.

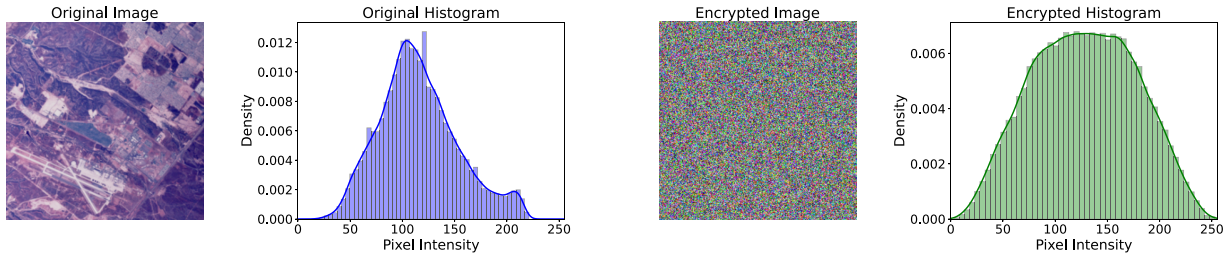


Figure 6: Comparison of histograms for original and encrypted aerial images.

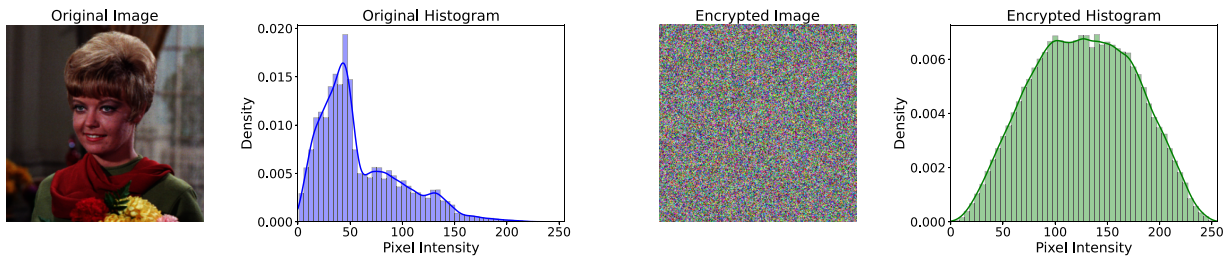


Figure 7: Comparison of histograms for original and encrypted miscellaneous image.

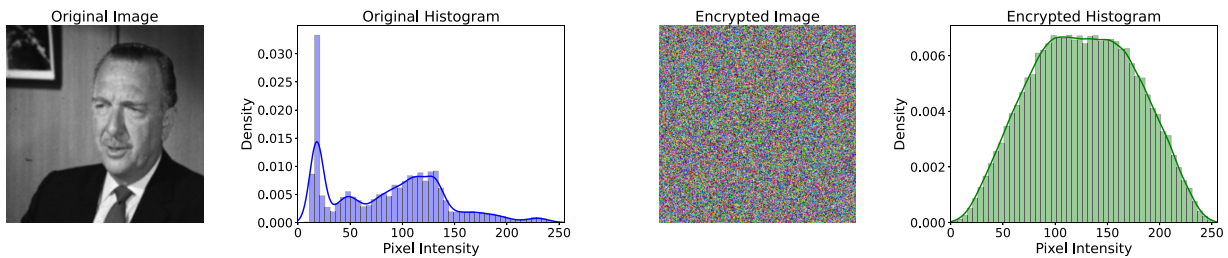


Figure 8: Comparison of histograms for original and encrypted sequence image.

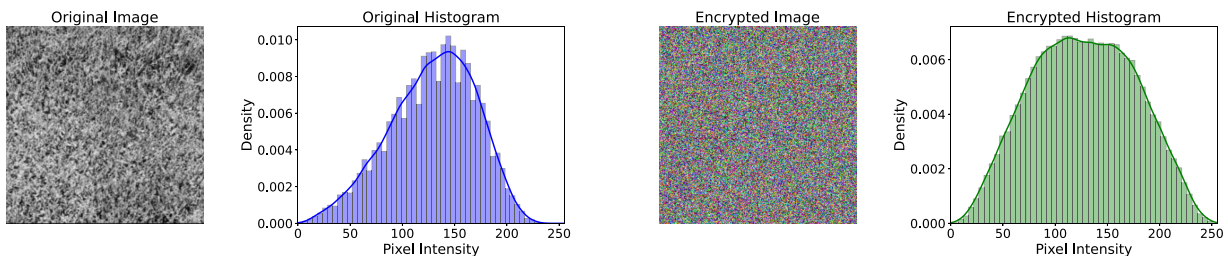


Figure 9: Comparison of histograms for original and encrypted texture image.

5.5.2 Chi-Square Analysis

The Chi-square test evaluates the uniformity of pixel intensity distribution, measuring how far the observed distribution deviates from an ideal uniform distribution. A lower Chi-square value in encrypted images indicates a more random-like distribution, desirable for resisting statistical attacks. As illustrated in Table 11, a significant reduction is observed across all tested datasets: the Aerial dataset drops from 92,099.77 to 240.96, Miscellaneous from 161,648.19 to 267.39, Sequence from 122,303.23 to 267.25, and Texture from 48,552.86 to 272.95. These consistent reductions confirm that the proposed CXC scheme effectively randomizes pixel distributions, eliminating identifiable structural patterns and significantly enhancing statistical security.

Table 11: Chi-square analysis of original and encrypted images.

Dataset Category	Original Image	Encrypted Image
Aerial	92,099.77	240.96
Miscellaneous	161,648.19	267.39
Sequence	122,303.23	267.25
Texture	48,552.86	272.95

5.5.3 Correlation Analysis

Correlation coefficients between adjacent pixels (horizontal, vertical, and diagonal) are computed for all image categories. As shown in Table 12, original images exhibit high correlation (0.4700–0.9865) due to strong spatial redundancy, while encrypted images produce near-zero values (≈ -0.0062 to 0.0046) across all directions and datasets. This confirms that the proposed CXC scheme effectively removes pixel dependencies and disrupts spatial relationships, achieving strong diffusion and confusion properties with robust resistance against statistical attacks.

Table 12: Comparison of pixel correlation coefficients between original and encrypted images.

Metric	Textures		Aerials		Miscellaneous		Sequences	
	Original	Encrypted	Original	Encrypted	Original	Encrypted	Original	Encrypted
Vertical	0.7160	−0.0062	0.9612	0.0002	0.9773	−0.0051	0.9579	−0.0039
Horizontal	0.6692	−0.0046	0.9401	0.0019	0.9710	−0.0013	0.9865	0.0034
Diagonal	0.4700	0.0046	0.9100	0.0004	0.9485	0.0045	0.9495	−0.0045

5.6 Differential Attack Analysis

5.6.1 Avalanche Effect

The avalanche effect measures encryption sensitivity to a one-bit plaintext change under the same secret key. Here, a single pixel value is modified while all other pixels remain unchanged; a strong cipher should produce significant, unpredictable ciphertext changes. Table 13 reports NPCR, UACI, Entropy, MSE, and PSNR results for different test images under this single-pixel modification scenario. The obtained NPCR values consistently exceed 99.4%, confirming that more than 99% of ciphertext pixels change upon a single-pixel plaintext modification, demonstrating strong diffusion and high resistance to differential attacks. UACI values are close to the ideal 33%, indicating that intensity variation between ciphertexts is statistically close to random behavior. Entropy values above 7.3 approach the theoretical maximum of 8 for 8-bit images,

reflecting a high degree of randomness in the encrypted outputs. Finally, high MSE and low PSNR values (around 11 dB) confirm significant divergence between encrypted images, collectively demonstrating that the proposed CXC scheme exhibits strong avalanche characteristics with robust resistance against statistical and differential attacks.

Table 13: Avalanche effect analysis under single-pixel modification.

Image Type	NPCR (%)	UACI (%)	Entropy	MSE	PSNR
Aerial	99.45	33.44	7.60	4898.86	11.23
Miscellaneous	99.43	33.46	7.43	4910.14	11.22
Sequence	99.42	33.45	7.38	4874.96	11.25
Texture	99.43	33.41	7.72	4893.91	11.23

5.6.2 Key Sensitivity

Key sensitivity evaluates encryption responsiveness to a minimal key change under the same plaintext image. Here, a one-bit modification is introduced in the encryption key, and the same image is encrypted using both the original and modified keys. Table 14 reports NPCR, UACI, Entropy, MSE, and PSNR results under this one-bit key variation scenario. NPCR values consistently exceed 99.4%, confirming that more than 99% of ciphertext pixels change upon a one-bit key modification. UACI values close to the ideal 33% confirm significant intensity variation across ciphertext images. Entropy values above 7.4 approach the theoretical maximum of 8 for 8-bit images, reflecting high randomness in ciphertexts generated under different keys. Finally, high MSE and low PSNR (≈ 11 dB) confirm that ciphertexts produced by slightly different keys are highly dissimilar, collectively demonstrating strong key sensitivity and robust resistance against key-based attacks.

Table 14: Performance metrics under one-bit key variation.

Image Type	NPCR (%)	UACI (%)	Entropy	MSE	PSNR
Aerial	99.46	33.47	7.81	4886.67	11.24
Miscellaneous	99.44	33.56	7.77	4861.82	11.26
Sequence	99.48	33.55	7.45	4883.36	11.24
Texture	99.49	33.48	7.61	4865.33	11.26

5.6.3 Noise Attack

To evaluate robustness against channel disturbances, a Gaussian noise attack with standard deviation $\sigma = 20$ is applied to the encrypted images. The corrupted ciphertext is then decrypted and decoded to analyze the effect on reconstructed images. Fig. 10a illustrates the decrypted encoded image under normal vs. noisy conditions, while Fig. 10b presents a visual comparison between the original and noisy reconstructed image. Table 15 reports MSE, PSNR, and SSIM between decrypted and fully decoded reconstructed images under this attack. MSE values reach up to 4801.14 for decoded images and 1800–1900 for decrypted images, PSNR ranges between 11–12 dB and ~ 15 dB, respectively, and SSIM falls between 0.12 and 0.18, with consistent visual degradation observed across all reconstructed outputs. These results confirm that the proposed framework is sensitive to channel noise, as expected for a lightweight scheme without error correction. The CXC encryption itself does not amplify noise; degradation occurs because bit errors in the ciphertext propagate through the deterministic decryption and decoding pipeline. Accordingly, the current design

is suitable primarily for noise-free or low-noise channels, and integration of forward error correction (FEC) or channel coding is recommended for deployment in typical IoT wireless environments. Under noise-free conditions, Table 16 confirms strong reconstruction performance across image categories (aerials, miscellaneous, sequences, textures), with PSNR ranging from 29.7 to 32.4 dB, SSIM between 0.94 and 0.97, and low MSE values throughout (average: PSNR = 31.2 dB, SSIM = 0.955, MSE = 50.1), collectively demonstrating high reconstruction fidelity under ideal conditions.

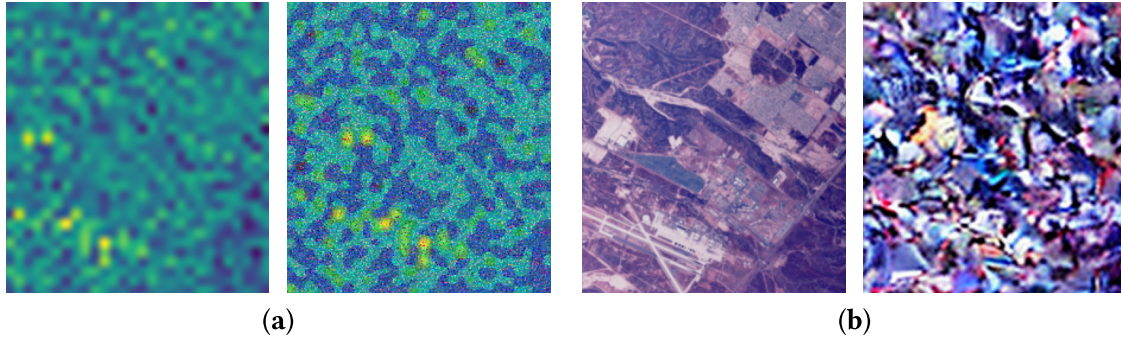


Figure 10: Visual comparison of the proposed framework under normal and Gaussian noise attack conditions. (a) Decrypted image under normal (left) and noise attack (right) conditions. (b) Comparison between original (Left) and noisy reconstructed image (right) after attack.

Table 15: Comparison of decrypted and decoded image quality under Gaussian noise attack.

Metric	Textures		Aerials		Miscellaneous		Sequences	
	Decoded	Decrypted	Decoded	Decrypted	Decoded	Decrypted	Decoded	Decrypted
MSE	4801.14	1820.73	3903.32	1870.08	4222.26	1880.14	3426.14	1840.14
PSNR	11.32	15.53	12.22	15.41	11.88	15.39	12.78	15.48
SSIM	0.17	0.13	0.18	0.12	0.13	0.12	0.17	0.17

Table 16: Baseline reconstruction quality under noise-free conditions.

Category	PSNR	SSIM	MSE
Aerials	31.8	0.96	42.3
Miscellaneous	32.4	0.97	37.1
Sequences	30.9	0.95	52.6
Textures	29.7	0.94	68.4
Average	31.2	0.955	50.1

5.6.4 Channel Coding Mitigation

To mitigate the degradation observed under Gaussian noise ($\sigma = 20$), a Reed–Solomon code RS (255,223) over GF (2^8) is applied to the ciphertext bitstream prior to transmission, adding 32 bytes of redundancy per 223-byte payload and enabling correction of up to 16 byte errors per block. Table 17 presents the results: with RS coding, reconstructed images achieve SSIM > 0.85 and PSNR > 24 dB, compared to SSIM < 0.20 without FEC, confirming that noise sensitivity can be substantially mitigated at approximately 14%

bandwidth overhead. Consequently, uncoded CXC is suitable for reliable channels (e.g., wired Ethernet or low-noise LoRa links), whereas RS-coded CXC is recommended for typical wireless IoT environments.

Table 17: Noise attack ($\sigma = 20$) with and without Reed–Solomon FEC.

Category	Without FEC			With RS (255,223)		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Aerials	3903.32	12.22	0.18	312.4	24.31	0.87
Miscellaneous	4222.26	11.88	0.13	340.1	23.95	0.86
Sequences	3426.14	12.78	0.17	298.7	24.58	0.88
Textures	4801.14	11.32	0.17	356.2	23.71	0.85

5.6.5 Security Analysis Against Known-Plaintext and Chosen-Plaintext Attacks

The CXC encryption employs a deterministic keystream in which each column's keystream depends on the secret key, a unique nonce, and a chaining seed derived from previous ciphertext columns, providing confidentiality under ciphertext-only assumptions. We analyze two active attack scenarios:

- **KPA:** If an adversary obtains a plaintext–ciphertext pair $(P_k, C_k^{(2)})$ for column k , the keystream segment can be recovered as $K_k^{(2)} = P_k \oplus C_k^{(2)}$. However, recovering the secret key K requires inverting Eq. (16), where $s_k^{(2)}$ is derived from a hash chain of previous ciphertext columns. Under the random oracle model, finding a valid preimage $(K, s_k^{(2)})$ for a 256-bit output requires $\mathcal{O}(2^{256})$ operations, leaving the secret key protected by the one-way property of SHA3-256 even with multiple known pairs.

$$K_k^{(2)} = \text{SHA3-256}(K \parallel \text{"pass2"} \parallel s_k^{(2)}) \quad (16)$$

- **CPA:** Resistance to CPA is ensured by the sequential dependency across columns. A single-bit modification in plaintext column j changes $C_j^{(1)}$, which affects the seed $s_{j+1}^{(1)}$ and consequently all subsequent keystream values in Pass 1. The cylinder feedback mechanism in Pass 2 further propagates this effect globally, since the last column of Pass 1 influences the initialization of Pass 2. Therefore, a single-bit change affects all ciphertext columns across both passes. Experimental results confirm this behavior, yielding NPCR > 99.4% and UACI $\approx$ 33.4%, indicating a strong avalanche effect. The use of a unique nonce per session prevents replay and codebook reuse attacks.

5.6.6 Key Space and Brute-Force Resistance

The security of the CXC scheme is primarily determined by the entropy of the 256-bit secret key and the nonlinear transformation induced by SHA3-256. The initial seed defined in Eq. (17) ensures strong key sensitivity, where a one-bit change in K or N produces entirely different keystream trajectories, experimentally confirmed by NPCR > 99.4% and UACI $\approx$ 33.4%. The 256-bit key combined with SHA3-256 provides a total key space of $2^{256} \approx 1.15 \times 10^{77}$, significantly exceeding the minimum cryptographic requirement of 2^{100} and rendering both brute-force attacks and key recovery through differential analysis computationally infeasible. For practical deployment, keys should be derived using a secure KDF such as PBKDF2 or Argon2id with appropriate salt and iteration parameters, rather than using raw passwords directly.

$$s_0^{(1)} = \text{SHA3-256}(K \parallel N \parallel \text{"init"}) \quad (17)$$

6 Discussion

Tables 18 and 19 highlight the effectiveness of the proposed framework in achieving a balanced trade-off between security, reconstruction quality, and computational efficiency. From a security perspective, the proposed method achieves NPCR and UACI values of 99.43% and 33.43%, respectively, which are comparable to those reported by state-of-the-art image encryption schemes, indicating strong resistance against differential attacks. The entropy value exceeding 7.3 further demonstrates a high degree of randomness in the encrypted images. In terms of reconstruction quality, the proposed framework achieves a PSNR of 31.5 dB, an MSE below 0.05, and an SSIM greater than 0.95. These results indicate that the decrypted images preserve structural information and visual fidelity with minimal distortion. Compared with existing autoencoder-based approaches, the proposed method provides competitive reconstruction performance while simultaneously maintaining strong encryption characteristics. It should be noted that some studies report PSNR values computed between plaintext and ciphertext images, whereas the proposed framework evaluates reconstruction quality between the original and decrypted images. Therefore, direct numerical comparison of PSNR values across all studies may not always be meaningful. Unlike conventional pixel-level encryption methods, the proposed framework employs autoencoder-based dimensionality reduction prior to encryption, reducing transmission bandwidth and computational requirements. The average end-to-end processing time is 0.28 s per image, including VAE encoding, residual extraction, CXC encryption/decryption, and decoding, while the CXC stage alone requires only 0.03 s. Although certain schemes report lower encryption times (e.g., 0.003 s in [12]), these methods do not include feature extraction, latent-space processing, or image reconstruction. Compared with other autoencoder-based image protection approaches [2,3,24,25], the proposed framework offers competitive execution time while uniquely combining reversible XOR-based encryption with high-fidelity reconstruction through residual compensation. Several limitations should be acknowledged. The framework is sensitive to channel noise: Gaussian noise ($\sigma = 20$) causes substantial degradation in reconstruction quality, reducing SSIM to below 0.2. Consequently, practical deployment over noisy wireless IoT networks would benefit from integrating forward error correction (FEC) mechanisms such as Reed–Solomon or LDPC codes. Furthermore, although the security analysis demonstrates strong statistical and avalanche characteristics, a formal IND-CPA security proof has not yet been established. In addition, scalability to higher image resolutions (e.g., 4K) requires further investigation regarding memory consumption and latency. These limitations will be addressed in future work through channel coding integration, formal cryptographic analysis, and hardware-specific optimization.

Table 18: Comparison of the proposed method with previous works.

Paper	NPCR (%)	UACI (%)	Entropy	Time (s)	HW Platform	PSNR	MSE	SSIM
[24]	99.65	33.65	7.99	3.5	Not specified	33.72	–	–
[25]	–	–	–	–	Not specified	28.70	–	–
[12]	–	–	–	≤ 0.003	CPU	31.3	49.6	0.73
[16]	99.67	33.45	7.99	0.05	Not specified	–	83.6	–
[3]	99.8	34.7	7.99	–	Not specified	≈ 29	–	–
[13]	–	–	–	0.12	Not specified	13.7	–	–
[2]	–	–	–	14	Not specified	–	0.0012	–

(Continued)

Table 18 (continued)

Paper	NPCR (%)	UACI (%)	Entropy	Time (s)	HW Platform	PSNR	MSE	SSIM
Proposed (CXC only—Reversible crypto)	99.43	33.43	>7.3	0.03	T4 GPU	–	–	–
Proposed (Full)	99.43	33.43	>7.3	0.28	T4 GPU	31.5	<0.05	>0.95
Proposed (Full CPU)	99.43	33.43	>7.3	0.93	i5 CPU	31.5	<0.05	>0.95

Note: Reported PSNR for [12,24,25], and the proposed method denotes reconstruction quality (decrypted vs. original).

Table 19: Feature-based comparative analysis of related work.

Study	Autoencoder	Dimensionality Reduction	IoT Suitability	Image Retrieved as Original
[24]	✓	✓	✗	✗
[5]	✗	✗	✓	✓
[1]	✗	✗	✓	✗
[12]	✓	✗	✓	✗
[20]	✗	✗	✓	✗
[6]	✗	✗	✓	✗
[21]	✗	✗	✓	✗
[13]	✗	✗	✓	✗
[3]	✓	✓	✗	✗
[25]	✓	✓	✗	✗
[2]	✓	✓	✓	✗
[16]	✗	✗	✓	✗
Proposed Approach	✓	✓	✓	✓*

Note: *Near-lossless reconstruction (SSIM > 0.95) due to autoencoder compression. XOR-based encryption is information-preserving (involution property).

For practical IoT deployment, a realistic *edge-assisted* architecture is assumed. In this setting, the resource-constrained sensor node (e.g., ARM Cortex-M4) executes only the lightweight CXC encryption stage, which requires approximately 120 ms on a Raspberry Pi 4, while computationally intensive VAE encoding and decoding operations are offloaded to an edge gateway or embedded AI accelerator (e.g., Google Coral TPU or Intel NCS2). This architecture enables the framework to balance security, reconstruction quality, and computational efficiency within resource-constrained IoT environments. Future work will investigate INT8 quantization and VAE knowledge distillation to enable fully standalone deployment on microcontroller-class devices.

7 Conclusion

This paper presents a lightweight and secure image encryption framework integrating a pretrained residual autoencoder with the proposed CXC scheme for efficient image transmission in edge-assisted IoT environments. The CXC scheme operates in two phases: a SHA3-256-based forward chained pass encrypts latent and residual representations via reversible XOR operations, followed by a cylinder-like feedback pass

that enforces inter-column dependency for enhanced global diffusion and heuristic resistance against differential and statistical attacks. Experimental results on the USC-SIPI dataset confirm strong security properties (NPCR $\approx$ 99.4%, UACI $\approx$ 33.5%, entropy $\approx$ 7.5), effective pixel decorrelation, and low computational overhead (<290 ms end-to-end on GPU; 25 ms for CXC alone). Under Gaussian noise, Reed–Solomon Forward Error Correction (FEC) restores reconstruction quality to SSIM > 0.85. The security analysis is explicitly heuristic and empirical, and should not be interpreted as a formal IND-CPA proof. Future work will pursue formal security analysis, adaptive chosen-plaintext attack evaluation, INT8 quantization for edge devices, and advanced channel coding for ultra-low-power wireless IoT networks.

Acknowledgement: The authors sincerely acknowledge Princess Sumaya University for Technology for supporting the steps of this research.

Funding Statement: The authors received no specific funding for this research.

Author Contributions: The authors confirm contributions to the paper as follows: Conceptualization: Rania Al-Ali and Mustafa Al-Fayoumi; methodology: Rania Al-Ali and Mustafa Al-Fayoumi; software: Saleem Alsaraireh; validation: Saleem Alsaraireh and Rania Al-Ali; formal analysis: Saleem Alsaraireh and Rania Al-Ali; investigation: Rania Al-Ali; data curation: Saleem Alsaraireh; writing—original draft preparation: Rania Al-Ali; writing—review and editing: Mustafa Al-Fayoumi, Rania Al-Ali, and Saleem Alsaraireh; supervision: Mustafa Al-Fayoumi. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The database used in this study is available from the corresponding author upon reasonable request.

Ethics Approval: Ethics Statement: This study did not involve the recruitment of human participants or the collection of personal data. All human images used in the experiments were obtained from the publicly available USC-SIPI Image Database, which is a widely used benchmark dataset for image processing research. The images were used solely for algorithm evaluation in accordance with the dataset's terms of use. Therefore, institutional ethics approval and informed consent were not required for this study.

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

References

1. Yung C, Huang SCH, Wu HC, Li CH. Novel secure and robust recoverable cryptographic mosaic technique. *IEEE Trans Multimed.* 2025;27:7137–51. doi:10.1109/tmm.2025.3590922.
2. El-Kafrawy P, Aboghazalah M, Ahmed AM, Torkey H, El-Sayed A. An efficient encryption and compression of sensed IoT medical images using auto-encoder. *Comput Model Eng Sci.* 2023;134(2):909–26. doi:10.32604/cmes.2022.021713.
3. Ahmed F, Rehman MU, Ahmad J, Khan MS, Boulila W, Srivastava G, et al. A DNA based colour image encryption scheme using a convolutional autoencoder. *ACM Trans Multimed Comput Commun Appl.* 2023;19(3s):1–21. doi:10.1145/3570165.
4. Chai X, Zheng X, Gan Z, Han D, Chen Y. An image encryption algorithm based on chaotic system and compressive sensing. *Signal Process.* 2018;148(9):124–44. doi:10.1016/j.sigpro.2018.02.007.
5. Li J, Sun L, Lin C, Wang X, Wang Z, He Q, et al. SED-UAV: a synergistic framework of lightweight chaotic encryption and multi-scale feature detection for secure UAV applications. *IEEE Internet Things J.* 2025;13(5):8050–63. doi:10.1109/jiot.2025.3607318.
6. Ali AH, Gbashi EK, Alaskar H, Hussain AJ. A lightweight image encryption algorithm based on secure key generation. *IEEE Access.* 2024;12(1):95871–83. doi:10.1109/access.2024.3414334.
7. Alawida M. A novel image encryption algorithm based on cyclic chaotic map in industrial IoT environments. *IEEE Trans Ind Inform.* 2024;20(8):10530–41. doi:10.1109/tii.2024.3395631.

8. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science*. 2006;313(5786):504–7. doi:10.1126/science.1127647.
9. Goodfellow I. Deep learning. Cambridge, MA, USA: MIT Press; 2016.
10. Stallings W. Cryptography and network security: principles and practice. 7th ed. London, UK: Pearson; 2017.
11. Acharya T, Ray AK. Image processing: principles and applications. Hoboken, NJ, USA: Wiley; 2009.
12. Zhang D, Shafiq M, Srivastava G, Gadekallu TR, Wang L, Gu Z. STBCIoT: securing the transmission of biometric images in customer IoT. *IEEE Internet Things J*. 2024;11(9):16279–88. doi:10.1109/jiot.2024.3351988.
13. Hedayati R, Mostafavi S. A lightweight image encryption algorithm for secure communications in multimedia Internet of Things. *Wireless Pers Commun*. 2022;123(2):1121–43. doi:10.1007/s11277-021-09173-w.
14. Kaur M, AlZubi AA, Singh D, Kumar V, Lee HN. Lightweight biomedical image encryption approach. *IEEE Access*. 2023;11:74048–57. doi:10.1109/access.2023.3294570.
15. Mohammed RA, Khodher MAA, Alabaichi A. A novel lightweight image encryption scheme. *Comput Mater Contin*. 2023;75(1):2137–53. doi:10.32604/cmc.2023.036861.
16. Kumari P, Mondal B. Lightweight image encryption algorithm using NLFSR and CBC mode. *J Supercomput*. 2023;79(17):19452–72. doi:10.1007/s11227-023-05415-9.
17. Alexan W, Hosny K, Gabr M. A new fast multiple color image encryption algorithm. *Clust Comput*. 2025;28(5):325.
18. Zhu S, Deng X, Zhang W, Zhu C. Image encryption scheme based on newly designed chaotic map and parallel DNA coding. *Mathematics*. 2023;11(1):231. doi:10.3390/math11010231.
19. Alexan W, Youssef M, Hussein HH, Ahmed KK, Hosny KM, Fathy A, et al. A new multiple image encryption algorithm using hyperchaotic systems, SVD, and modified RC5. *Sci Rep*. 2025;15(1):9775.
20. Zhang Y, Zhao R, Zhang Y, Yi S, Lan R. Visually semantics-aware color image encryption based on cross-plane substitution and permutation. *IEEE Trans Ind Inform*. 2024;20(8):10576–86. doi:10.1109/tii.2024.3395646.
21. Kumar S, Sharma D. A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm. *Artif Intell Rev*. 2024;57(4):87.
22. Alexan W, Elkandoz M, Mashaly M, Azab E, Aboshousha A. Color image encryption through chaos and KAA map. *IEEE Access*. 2023;11:11541–54. doi:10.1109/access.2023.3242311.
23. Li L. Image encryption algorithm based on hyperchaos and DNA coding. *IET Image Process*. 2024;18(3):627–49. doi:10.1049/ipr2.12974.
24. Manikandan S, Linkkesh A, Sreenivasan S, Thanikaiselvan V, Subashanthini S, Amirtharajan R. Autoencoder-based image encryption using hybrid scrambling, diffusion, and dimensionality reduction. *Results Eng*. 2026;29(1):108977. doi:10.1016/j.rineng.2026.108977.
25. Kumar K, Tanwar S, Kumar S. MANC: a masked autoencoder neural cryptography based encryption scheme for CT scan images. *MethodsX*. 2024;12:102738. doi:10.1016/j.mex.2024.102738.
26. LightX2V Team. LightVAE: efficient video autoencoder (VAE) model collection. Hugging Face; 2025 [cited 2026 Apr 8]. Available from: <https://huggingface.co/LightX2V>.
27. Signal and Image Processing Institute (SIPI), University of Southern California. USC-SIPI image database; 2026 [cited 2026 Apr 3]. Available from: <https://sipi.usc.edu/database/>.