Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

MMIF: GPU-Accelerated Parallel Cryptographic Permutation Network for Secure Multimodal Medical Image Fusion in Brain Tumor Detection

Rajasekar Ramasamy
VIT-AP University
Ramkumar Devendiran
ramkumar.d@vitap.ac.in

VIT-AP University

Article

Keywords: Multimodal image fusion, GPU, Bit-Planes, DNA-encoded chaotic sequence, chaotic mapping, Permutation Network

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11110965/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

MMIF: GPU-Accelerated Parallel Cryptographic Permutation Network for Secure Multimodal Medical Image Fusion in Brain Tumor Detection

Rajasekar Ramasamy[1]
Research Scholar, School of Computer Science & Engineering, VIT-AP University,
Beside AP Secretariat, Amaravati, 522241, Andhra Pradesh, India.
rajasekar.23phd7057@vitap.ac.in

Ramkumar Devendiran[2]
Assistant Professor Senior Grade, School of Computer Science & Engineering,
VIT-AP University, Beside AP Secretariat, Amaravati, 522241, Andhra Pradesh, India.
ramkumar.d@vitap.ac.in

Corresponding Author: Ramkumar Devendiran; ramkumar.d@vitap.ac.in

Abstract

The integration of complementary anatomical and functional information from various imaging modalities, such as CT, MRI, and PET, through multimodal medical image fusion has emerged as a potent technique for enhancing brain tumour diagnosis. Nevertheless, sharing and storage of fused medical images may pose serious privacy and security risks. Additionally, traditional methods for permutation diffusion encryption generally encounter sequential processing bottlenecks, making them unsuitable for real-time medical scenarios. Here we introduce a parallel cryptographic permutation network (PCPN) that blends DNA-assisted chaotic permutation and light diffusion to ensure secure and computationally efficient protection of fused medical images. At first, the fused image is divided into eight binary bit-planes. Next, a chaotic sequence generated by the logistic map is converted into DNA symbols to form distinct permutation tables for each bit-plane. The GPU-parallelised permutation allows the simultaneous scrambling of the bit planes. After that, the scrambled bit-planes are combined to form the encrypted image. Pixel-wise XOR diffusion is done by a chaotic keystream generated by a second logistic map. It not only offers avalanche characteristics and statistical randomness but also needs very low computation. By allowing independent bit-plane processing and precomputed permutation tables, the proposed system removes sequential feedback dependencies, which greatly diminishes encryption times. Experiments have indicated that our encryption scheme outperforms other schemes in cryptographic security. It accomplishes an NPCR of 99.92%, UACI of 33.48%, and entropy of 7.9992 bits/pixel. The scheme is highly resistant to differential and statistical attacks while preserving the integrity of the image. Our scheme strikes an ideal balance between protection and computational efficiency, which is a practical solution to ensure the confidentiality of multimodal medical image fusion for brain tumour diagnosis. And this method is also a very viable option for the implementation of real-time intelligent healthcare systems.

Keywords: Multimodal image fusion, GPU, Bit-Planes, DNA-encoded chaotic sequence, chaotic mapping, Permutation Network.

1. Introduction

Multi-modal medical image fusion is an approach to integrating complementary information from different imaging modalities like computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET). This method has become indispensable for accurate brain tumor detection and diagnosis [1]. By combining the high-resolution anatomical details of CT and MRI with the functional and metabolic information of PET, the technique brings clinicians a complete view of the tumor area [2]. Deep learning-driven convolutional neural network (CNN) models that effectively derive features from multi-modal inputs have recently been developed [3]. Yet, medical images carry highly sensitive information, and this way, strong privacy protections are needed to prevent any unauthorized use due to the potentially serious ethical and legal consequences.

Many traditional encryption techniques for medical images have been built on chaotic map-based schemes, for example, the Arnold cat map [4] and the logistic map-driven permutation-diffusion architectures [5]. Typically, they follow the permutation-diffusion structure sequentially, where the confusion and diffusion stages depend heavily on the feedback loop that is geometrically formed [6]. While they do provide security, at the same time, they greatly increase the computational time, which makes it impossible to use them in real-time scenarios in clinics. This delay is quite critical with multi-modal fusion, where the encrypting part should be capable of effectively handling large fused-image matrices without causing delays to the subsequent diagnosis pipeline.

Our idea of a novel cryptographic permutation network that is parallelizable and suitable for secure multi-modal medical image fusion has arisen precisely from these issues. We have drawn inspiration from the ASCON cipher family [7]. for our lightweight permutation structure as a substitute for the regular sequential diffusion pattern. which has been standardized for efficient authenticated encryption in constrained environments. The proposed method first decomposes the fused image into eight binary bitplanes [8], and then each bit-plane is individually and simultaneously scrambled by a different permutation table, which is derived from a DNA-encoded chaotic sequence [9]. This parallel bit-plane shuffling eliminates the heavy dependence on feedback loops of conventional methods, thereby making possible hardware-accelerated matrix transformations on GPU architectures [10].

Separating the permutation and diffusion steps has been the main breakthrough that allowed our encryption module to be run in parallel with the feature extraction pipeline. In contrast to existing chaotic schemes that perform these operations sequentially, our approach precomputes permutation tables from a logistic map driven DNA encoding process, where each chaotic value is quantized into one of four DNA bases, and the resulting DNA string is sorted to produce a permutation index vector. A circular shift of this base vector by the bit-plane index guarantees distinct scrambling across all planes, thereby realizing high entropy and global avalanche effects. After recombination of the permuted bit-planes, a lightweight diffusion layer is applied via a pixel-wise XOR operation with a keystream from a second chaotic logistic map, thereby achieving robust cryptographic standards without compromising computational efficiency.

The primary contributions of this paper are threefold. First, we reformulate the encryption sub-module for multi-modal medical image fusion to preserve the structural properties necessary for diagnostic analysis while achieving computational efficiency suitable for real-time clinical

applications. Second, we introduce a parallelizable bit-plane permutation structure which eliminates the sequential bottlenecks of traditional diffusion methods, thereby enabling GPU-accelerated encryption. Third, we show that the proposed system retains strong security against common cryptographic attacks while markedly cutting latency relative to standard chaotic approaches.

The remainder of this paper is organized as follows. Section 2 surveys prior work on multi-modal medical image fusion, encryption techniques, and their combination. Section 3 presents the necessary preliminaries, namely bit-plane decomposition, DNA encoding, and chaotic map fundamentals. In Section 4, the authors describe in detail their parallel cryptographic permutation network, which features three main components: the creation of the permutation table, the scrambling of bit-planes in parallel, and the diffusion layer design. Section 5 explains the experimental setup that includes datasets, evaluation metrics, machine specifications, and programming languages used. Section 6 is devoted to presenting and discussing the experimental results with security analysis, computational performance, and fusion quality assessment comprising the major components of the report. Section 7 elaborates on the consequences of our discoveries, constraints, and suggestions for future research. Section 8 wraps up the paper by highlighting the main points of our work.

2. Related Work

The integration of encryption technologies into multimodal medical image fusion systems is an important research area, which is mainly fuelled by the need for protection of patient data confidentiality alongside maintaining the diagnostic yield. In this paragraph, we analyze the state-of-the-art in three closely related fields: multimodal medical image fusion techniques, chaotic and DNA-based image encryption methods, and parallelizable cryptographic architectures for real-time applications.

2.1 Multi-Modal Medical Image Fusion

Multi-modal medical image fusion aims to combine different sets of complementary data from various imaging modalities into one integrated picture that maintains the key characteristics of each source. In case of brain tumor detection, combining CT, MRI, and PET data really makes a difference because it brings together detailed anatomical images with metabolic information of the tissue [1]. Transform-domain methods like wavelet decomposition and principal component analysis were initially used for fusion. They are useful to extract and combine features at multiple scales [2]. Nowadays, methods based on deep learning, mainly CNNs, are winning as they can learn the best feature representation from the input modalities themselves [3]. For instance, a hybrid deep learning solution that integrates MRI and PET images has been reported to raise the accuracy of tumor diagnosis by utilizing both the high spatial resolution of MRI and the metabolic sensitivity of PET [11]. However, a triple generation network architecture has been introduced for multi-modality fusion that allows simultaneous processing of CT, MRI, and PET data for brain abnormality detection [12]. Actually, these fusion models usually generate one single combined image array, which is then used by diagnostic modules, leading to the natural location for encryption in such a diagnostic pipeline.

2.2 Chaotic and DNA-Based Image Encryption

Chaotic map-based encryption has been a popular choice for medical image protection because chaotic systems are extremely sensitive to changes in initial conditions and have the ability to produce pseudo-random sequences with very high entropy [5]. One of the main methods of generating chaotic sequences is using the logistic map, which is given by the recurrence relation relation $x_{n+1} = r \cdot x_n \cdot (1 - x_n)$, The parameter r is normally set to a value near 4.0 to ensure chaotic behavior [13]. Traditional chaotic encryption techniques have a permutation-diffusion process where one step follows the other. First, pixels are rearranged in a new order using a permutation map, and then the pixel intensities are changed through a diffusion process, which results in spreading the modifications to the whole image [6]. However, this sequential dependence leads to significant computational time because the diffusion of each pixel relies on the diffusion value of the previous pixel.

DNA-based encryption systems have raised attention for their potential to increase security through the use of encoding schemes inspired by biology [9]. One of the steps in these systems is the conversion of pixel values to DNA sequences by a mapping rule that associates each nucleotide (A, T, C, G) with a particular binary pair. Thereafter, the DNA sequences can be modified through operations like complementary substitution, crossover, and mutation, which are similar to the biological processes. For instance, a novel DNA technique has been proposed to reinforce the cryptographic permutation tables in encryption algorithms, by which data encryption can be expedited through parallel processing [14]. Studies validate that chaotic maps and DNA encoding together can produce enough secure encryption to withstand various types of attacks, including differential and statistical analyses [15].

2.3 Parallelizable Cryptographic Architectures

The need to encrypt data instantly in hospital scenarios has led to the development of parallel cryptographic architectures that make use of hardware accelerators like GPUs. Lightweight cryptographic algorithms, for example, the ASCON cipher family, have been designed. Specifically, these algorithms are designed for environments with limited resources and depend on highly efficient, permutation-based operations that can be naturally parallelized [7]. The permutation unit of ASCON, which is responsible for operating on a 320-bit state within a sponge setup, has been optimized for multi-clock domain implementation use to be resource-efficient and to have a shorter critical path [16]. These results demonstrate that based on permutations, one can give rise to encryption that has a great throughput and is low in latency, thereby being a good fit for real-time operation.

For image encryption, different parallelizable schemes have been put forward to counteract the sequential nature of traditional diffusion techniques. For instance, it was proved by the researchers that a chaos-based image encryption system with simultaneous permutation-diffusion operations could reduce the time through parallel computing [17]. An image encryption scheme based on a two-dimensional chaotic system and bit-plane cross permutation-diffusion was suggested, which exploits parallel computing to achieve high processing speed [18]. Per these approaches, bit-plane decomposition, i.e., the separation of an image into its individual binary planes, gives a natural structure for parallel processing since each bit plane can be manipulated independently.

These methods indicate that bit-plane decomposition, a process by which an image is divided into its component binary planes, establishes a natural structure for parallel computation because each

bit-plane can be processed independently. Moreover, a synchronous permutation-diffusion approach for multiple image encryption has been introduced, which handles multiple images at the same time to increase throughput [19].

2.4 Comparison with Proposed Work

Although notable progress has been achieved in multi-modal image fusion, chaotic encryption, and parallelizable architectures, certain deficiencies still exist. To begin, most fusion-specific encryption schemes treat the encryption module as an independent post-processing step, overlooking the structural properties of the fused image needed for subsequent diagnostic analysis. Currently available parallelizable encryption methods commonly depend on intricate two-dimensional chaotic systems that demand considerable computing resources for key generation. Third, the combination of DNA encoding and parallel bit-plane permutation remains insufficiently examined within medical image fusion. In our paper, we introduce a lightweight permutation network that works directly on the fused image matrix while avoiding the use of heavy feedback loops. The use of a simple logistic map for the DNA-encoded permutation table generation ensures low computational cost, and the scrambling of the bit planes in parallel allows for GPU acceleration. And, by separating the permutation and diffusion with this, the encryption module can be scheduled to run simultaneously with the feature extraction pipeline, which leads to a reduction in latency for the real-time clinical applications.

3. Preliminaries

In this part, we lay out the main ideas and the mathematical tools that act as a basis for our encryption setup. Three main concepts are described: bit-plane slicing aimed at facilitating image processing in parallel, DNA encoding devised for transforming chaotic sequences, and the logistic map used in pseudo-random number generation. These brief comments lay down the necessary background for the understanding of the parallel cryptographic permutation network presented in Section 4.

3.1 Bit-Plane Decomposition

Bit-plane decomposition is a method that splits a grayscale image into its composite binary layers, each layer corresponding to a specific bit position within the pixel value representation [8]. For an 8-bit grayscale image, each pixel value $p(i,j)$ can be expressed as a sum of weighted binary digits:

$$p(i,j) = \sum_{k=0}^{7} b_k\,(i,j) \cdot 2^k \quad (1)$$

where $b_k(i,j) \in \{0,1\}$ represents the bit value at the k-th bit-plane for the pixel at spatial coordinates (i,j). The decomposition yields eight binary images, each of dimension's M × N, where the higher-order bit-planes (e.g., k=7,6) carry the most visually important information, while the lower-order planes (e.g., k=0,1) mainly contribute to fine details and noise.

Bit-plane decomposition offers strong parallelism for encryption. Each bit-plane forms a separate binary matrix; scrambling can run on all eight planes at once, with no need to wait for one plane before moving to another. And this works well on GPUs, where threads handle different planes together [10]. Furthermore, the binary nature of each bit-plane streamlines the permutation process because each plane can be treated as a binary matrix in which pixel values are either 0 or 1.

3.2 DNA Encoding

DNA encoding is a bio-inspired method that maps binary data onto sequences composed of the four nucleotide bases adenine (A), thymine (T), cytosine (C), and guanine (G) [9]. Within image encryption, every two-bit binary pair is assigned to one of the four DNA bases following a predefined encoding rule. Since there are 4! = 24 possible bijective mappings between the four 2-bit combinations and the four DNA bases, the encoding rule itself can serve as a secret key parameter.

For a chaotic sequence $X = \{x_1, x_2, \dots, x_n\}$ generated by a logistic map, every term $x_i \in [0,1]$ is discretized into one of four DNA bases. A common quantization scheme divides the interval $[0,1]$ into four equal sub-intervals: $[0,0.25)$, $[0.25,0.5)$, $[0.5,0.75)$, and $[0.75,1.0]$, each corresponding to a specific DNA base. The resulting DNA string D = {d1, d2, …, dn}, where each di ∈ {A, T, C, G}, is subsequently ordered to yield a permutation index vector. This vector defines the order in which the elements of a bit-plane should be rearranged during the scrambling process.

Employing DNA encoding in permutation table generation confers two notable benefits. The conversion of chaotic values into discrete DNA bases introduces a non-linear mapping that bolsters the cryptographic strength of the resulting permutation [14]. Second, the sorting of the DNA string yields a permutation that is both deterministic (given the same chaotic sequence) and highly sensitive to initial conditions, so that even minor modifications in the encryption key result in completely different permutation tables.

3.3 Logistic Map

The logistic map constitutes a single-dimensional chaotic system defined via the recurrence relation [13].

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n) \quad (2)$$

where $x_n \in [0,1]$ is the state variable at iteration n, and $r \in [0,4]$ is the control parameter. The logistic map displays chaotic behavior for $r \in [3.57,4.0]$, and the system becomes entirely chaotic at $r = 4.0$. In this regime, the sequence generated by the map exhibits strong sensitivity to the initial condition x_0, such that two sequences starting from arbitrarily close initial values diverge exponentially over time.

In cryptography, logistic maps are preferred due to their simplicity and low computational cost. Unlike complex chaotic systems that require multiple state variables or extra dimensions for their operations, a logistic map is a single multiplication and subtraction operation in one iteration. Such lightweight operations make logistic maps ideal for performing encryption in real time in low-resource environments, e.g., medical imaging systems [5].

In our model, we take two separate logistic maps with different control parameters and initial conditions. The first of these is the source of the chaotic sequence used for DNA encoding and permutation table generation. But the second map gives the keystream for the diffusion layer afterward. Keeping the two processes separate as done here creates cryptographically independent permutation and diffusion stages that have the effect of raising the security level of the overall encryption scheme. The logistic map being highly sensitive to initial conditions means that one

change of the encryption key results in an entirely different permutation table and keystream, which makes the system robust against chosen-plaintext and known-plaintext attacks.

4. Parallelizable Cryptographic Permutation Network for Multi-Modal Fusion

Here is a description of the technical aspects of our proposed encryption system, which is basically a sub-module of a multi-modal medical image fusion system. The system architecture overall is shown in Figure 1, where the encryption module is located between the Fusion Engine and the Output Module to enable the secure transmission of fused images to diagnostic pipelines, which are downstream. Our main feature is the parallel bit-plane permutation method, which replaces the old sequential diffusion scheme with a set of independent, simultaneously performed scrambling operations. At first, we describe the system-level integration; then we discuss the generation of the permutation table with the help of DNA-encoded chaotic sequences, and finally we describe the parallel scrambling and the diffusion next.

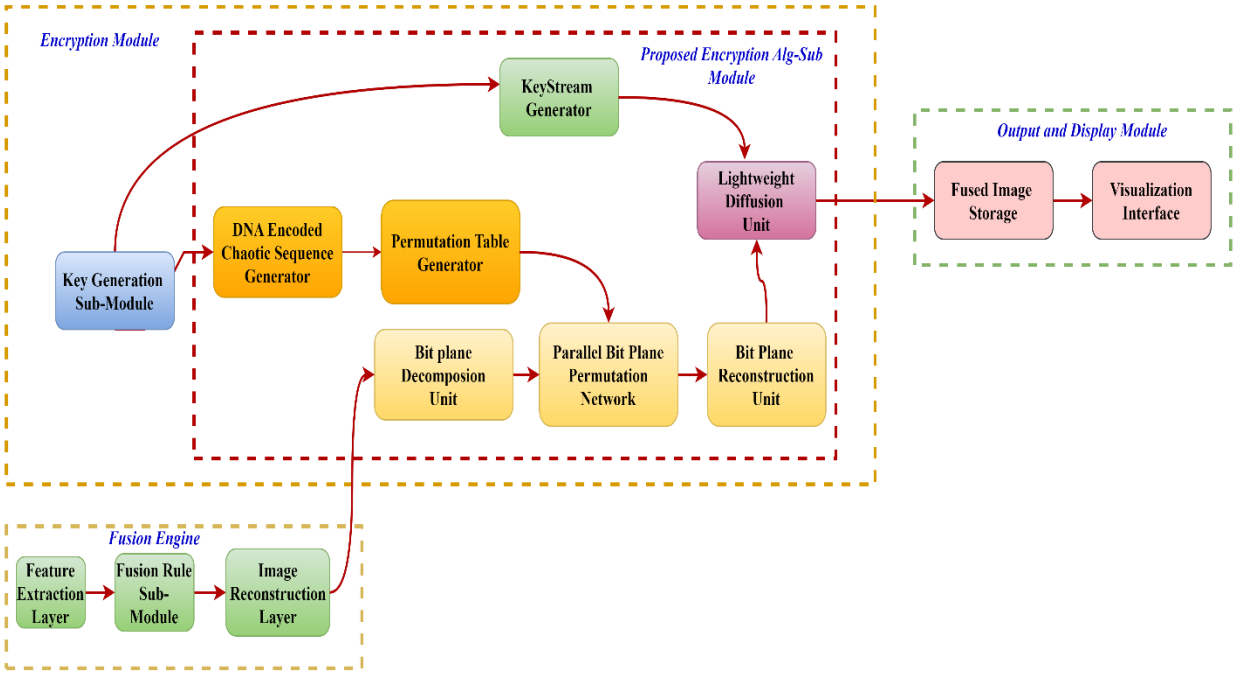


Figure 1. System Level Data Flow with Proposed Encryption Module

4.1 Bit-Plane Decomposition and DNA-Derived Permutation Table Generation

The encryption process begins with the decomposition of the fused image matrix **F**, which is an $M \times N$ grayscale image with 8-bit pixel depth, into its constituent binary bit-planes. Each pixel value f(i,j) ∈ [0, 255] is expressed as a weighted sum of eight binary digits, as set out in Equation 1. This decomposition yields eight binary matrices $\mathbf{B}_k \in \{0,1\}^{M\times N}$, where $k \in \{0,1,\ldots,7\}$ indexes the bit-plane from the least significant bit (LSB, $k = 0$) to the most significant bit (MSB, $k = 7$). The binary value at position (i, j) in the k-th bit-plane is given by:

$$b_k(i,j) = \left\lfloor \frac{f(i,j)}{2^k} \right\rfloor \bmod 2 \quad (3)$$

where $\lfloor \cdot \rfloor$ denotes the floor function. This decomposition is performed once for the entire fused image, and the resulting eight binary matrices serve as the independent inputs for the subsequent parallel scrambling stage.

To generate the permutation tables required for scrambling, we employ a logistic map with initial condition x_0 and control parameter r, both of which serve as secret keys. The logistic map iterates $M \times N$ times to produce a chaotic sequence $X = \{x_1, x_2, \dots, x_{M \times N}\}$, where each $x_t \in [0,1]$. Each chaotic value is subsequently quantized into one of four DNA bases according to the following mapping rule.

$$d_t = \begin{cases} A & \text{if } 0.00 \le x_t < 0.25, \\ T & \text{if } 0.25 \le x_t < 0.50, \\ C & \text{if } 0.50 \le x_t < 0.75, \\ G & \text{if } 0.75 \le x_t < 1.00 \end{cases} \quad (4)$$

This quantization produces a DNA string $D = \{d_1, d_2, \dots, d_{M \times N}\}$, where each element is one of the four nucleotide bases. To derive a permutation index vector $\mathbf{p}$ of length $M \times N$, we assign numerical values to each DNA base: $A = 0, T = 1, C = 2, G = 3$. The DNA string is then sorted in ascending order according to these numerical values, and the original indices of the sorted elements form the permutation vector $\mathbf{p}$. Formally, if σ is the sorting permutation for which $d_{\sigma(1)} \le d_{\sigma(2)} \le \cdots \le d_{\sigma(M \times N)}$, then the vector $\mathbf{p}$ is defined as $(\sigma(1), \sigma(2), \dots, \sigma(M \times N))$.

To guarantee that each bit-plane is assigned a distinct permutation table, we rotate the base permutation vector p circularly by an amount equal to the bit-plane index k. The permutation table for the k-th bit-plane is therefore:

$$\mathbf{P}_k = \text{circular_shift}(\mathbf{p}, k) \quad (5)$$

where circular shift$(\mathbf{v}, s)$ shifts the elements of vector $\mathbf{v}$ to the left by s positions, with elements that fall off the beginning wrapping around to the end. This operation guarantees that the eight bit-planes are permuted via eight distinct permutation tables, thus eliminating any link among the permutation patterns across planes. The permutation tables $\mathbf{P}_k$ are precomputed and stored in GPU memory; consequently, the subsequent scrambling stage can commence immediately upon receipt of the decomposed bit-planes.

Algorithm 1 shows how to generate permutation tables from a DNA-encoded chaotic sequence. A logistic map is used to generate pseudo-random values that are then mapped to DNA bases (A T C, and G). The DNA sequence is numerically encoded, sorted, and circularly shifted to obtain different permutation tables for all the eight bit-planes. These tables work as the secret scrambling keys in the encryption process.

Algorithm 1: DNA-Based Permutation Table Generation
Input: Logistic map parameters (x0, r) Image size M×N Output: Eight permutation tables P0...P7 1. Initialize logistic map. 2. Generate chaotic sequence S.

3. Quantize every value into DNA bases
   A, T, C, G.
4. Convert DNA symbols into integers.
5. Sort the DNA sequence.
6. Obtain permutation vector P.
7. For k = 0 to 7
   Circularly shift P by k positions.
   Store as permutation table Pk.
8. End For
9. Return all permutation tables.

4.2 Simultaneous Independent Permutation Across Bit-Planes

Once the eight binary bit-planes $\mathbf{B}_k$ and their corresponding permutation tables $\mathbf{P}_k$ are prepared, the core scrambling operation proceeds in parallel. For each bit-plane k, the binary matrix $\mathbf{B}_k$ is transformed into a vector of length $M \times N$ by extracting its elements in row-major order. Let $\mathbf{v}_k$ denote this vectorized representation, where the element at linear index $l = i \cdot N + j$ corresponds to the pixel at spatial coordinates (i, j). The permutation table $\mathbf{P}_k$ defines a correspondence from the new linear index l' to the original linear index l, so that the scrambled vector $\mathbf{v'}_k$ is expressed as follows:

$$\mathbf{v'}_k(l') = \mathbf{v}_k(\mathbf{P}_k(l')) \quad (6)$$

where $l' \in \{0,1, \dots, M \times N - 1\}$. In this formulation, $\mathbf{P}_k(l')$ returns the original linear index of the element destined for position l' in the scrambled vector. This is equivalent to a permutation of the vector elements according to the order specified by $\mathbf{P}_k$.

A key benefit of our method is the ability to evaluate Equation 6 separately for each bit-plane k and each linear index l'. There is no dependency between different bit-planes or between different positions within the same bit-plane. This independence permits the entire scrambling operation to be executed as a single parallel operation on a GPU, where each thread can be assigned to compute a single element $\mathbf{v'}_k(l')$ for a specific combination of k and l'. The massive parallelism of the GPU permits the concurrent calculation of all $8 \times M \times N$ output elements within a single kernel launch, thus removing the sequential bottleneck that is intrinsic to conventional permutation-diffusion approaches.

After the parallel scrambling, each vector $\mathbf{v'}_k$ is reshaped back into a binary matrix $\mathbf{B'}_k$ of dimensions $M \times N$ by reversing the row-major scanning process. The element at spatial coordinates (i, j) in the scrambled bit-plane $\mathbf{B'}_k$ is given by:

$$\mathbf{B'}_k(i, j) = \mathbf{v'}_k(i \cdot N + j) \quad (7)$$

This reshaping operation constitutes a simple memory rearrangement that can be executed without additional computation because the GPU memory layout can be configured to directly store the scrambled vectors in the desired bidimensional format.

Algorithm 3 details the bit-plane scrambling method by which the separate bit-plane extracted from the image shuffle their pixel positions as the permutation table derived for that bit-plane. As the bit-planes are independent of each other, there is no need to wait for one bit-plane to

be completed when scrambling others. Actually, all scrambling can be run in parallel on GPU threads to greatly lower latency without any compromise of security.

Algorithm 2: Parallel Bit-Plane Scrambling

Input:
Eight bit-planes
Eight permutation tables

Output:
Scrambled bit-planes
1. Decompose image into bit-planes.
2. Parallel For k = 0 to 7
Convert bit-plane into vector.
Apply permutation table Pk.
Store scrambled vector.
Convert vector back into matrix.
End Parallel For
3. Return scrambled bit-planes.

4.3 Ciphertext Reconstruction and Pipeline Synchronization

After the concurrent shuffling of all eight bit-planes, the subsequent step consists of merging the reordered binary matrices into a solitary encrypted grayscale image. The reconstruction process is the inverse of the decomposition step described in Section 4.1. For each spatial coordinate (i, j), the pixel value in the encrypted image $\mathbf{F}_{\text{enc}}$ is computed as the weighted sum of the scrambled bit-plane values:

$$\mathbf{F}_{\text{enc}}(i,j) = \sum_{k=0}^{7} \mathbf{B'}_k\,(i,j) \cdot 2^k \quad (8)$$

where $\mathbf{B'}_k(i,j)$ is the binary value at position (i, j) in the k-th scrambled bit-plane. This summation is executed element-wise across all eight bit-planes, thereby yielding a single $M \times N$ matrix of 8-bit pixel values. The reconstruction operation is computationally lightweight, as it demands only eight bit-shift and addition operations per pixel, and can be efficiently parallelized on a GPU by assigning each pixel coordinate to a separate thread.

To guarantee robust cryptographic security, a lightweight diffusion layer is applied after the reconstruction step. This diffusion layer introduces a pixel-wise XOR operation between the reconstructed image $\mathbf{F}_{\text{enc}}$ and a keystream matrix $\mathbf{K}$ of the same dimensions. The keystream is produced by a second logistic map with initial condition y_0 and control parameter r', both of which are different from the parameters employed for permutation table generation. The logistic map undergoes $M \times N$ iterations to generate a chaotic sequence $Y = \{y_1, y_2, \ldots, y_{M\times N}\}$, which is subsequently converted into 8-bit integer values via the following transformation.

$$k_t = \lfloor y_t \times 256 \rfloor \bmod 256 \quad (9)$$

The resultant integer sequence is restructured into an $M \times N$ matrix $\mathbf{K}$ arranged in row-major order. The final encrypted image $\mathbf{F}_{\text{final}}$ is then obtained as:

$$\mathbf{F}_{\text{final}}(i,j) = \mathbf{F}_{\text{enc}}(i,j) \oplus \mathbf{K}(i,j) \quad (10)$$

where $\oplus$ denotes the bitwise XOR operation. In this diffusion layer, we ensure that even changing one pixel in the merged image is enough to cause many pixels in the encrypted image to change as well. This is what creates the avalanche effect that makes differential cryptanalysis impossible. The pipeline synchronization mechanism is one of the most important features of our proposed architecture. After permutation tables $\mathbf{P}_k$ are precomputed and stored in GPU memory during the initialization phase, the encryption module can start processing the fused image as soon as it arrives from the fusion engine without any delay. Operations such as bit-plane decomposition, parallel scrambling, reconstruction, and diffusion are carried out one after the other by launching GPU kernels, with no need for the CPU to become involved. This way, the encryption module can run alongside the downstream diagnostic system's feature extraction pipeline. That is, while Fusion Engine is processing the next set of multi-modal inputs, the encryption module can be encrypting the previously fused image at the same time.

This overlapping of computations hides the latency of encryption behind the latency of fusion, so this method decreases the overall time for processing real-time clinical applications.

4.4 Proposed Encryption Algorithm

To improve the reproducibility and implementation clarity of the proposed encryption framework, the complete encryption procedure is summarized in Algorithm 1. The algorithm illustrates the sequence of operations, beginning with fused image decomposition, DNA-assisted chaotic permutation table generation, parallel bit-plane scrambling, ciphertext reconstruction, and final XOR-based diffusion.

Algorithm 3: Parallel Cryptographic Permutation Network (PCPN)

Input:
F: Fused grayscale medical image (M × N)
x0, r: Initial value and control parameter of Logistic Map-1
y0, r': Initial value and control parameter of Logistic Map-2
Output:
C: Encrypted fused medical image
Step 1:
Read fused image F.
Step 2:
Perform bit-plane decomposition.
Generate eight binary bit-planes:
B0, B1, B2, ..., B7.
Step 3: Generate chaotic sequence using Logistic Map-1.
Step 4: Convert chaotic sequence into DNA symbols
(A, T, C, G).
Step 5: Sort DNA sequence to generate the base
permutation vector P.

Step 6: For each bit-plane k = 0 to 7
    Rotate permutation vector by k positions.
    Generate permutation table Pk.
End For
Step 7: Execute parallel permutation.
For each bit-plane (simultaneously)
    Apply permutation table Pk
    Obtain scrambled bit-plane B'k
End For
Step 8: Reconstruct encrypted grayscale image
from all scrambled bit-planes.
Step 9: Generate second chaotic sequence using
Logistic Map-2.
Step 10: Generate diffusion key stream K.
Step 11: Perform XOR diffusion
    C = Reconstructed Image XOR K
Step 12: Return encrypted image C.

Computational Complexity: It takes O(MN) operations for the bit-plane decomposition. Deriving a chaotic sequence and performing DNA encoding each have a time complexity of O(MN). Sorting for the arrangement vector to be used in permuting the image can be done in O (MN log (MN)) time and this operation is only done once during the initialization. Parallel bit-plane permutation and XOR diffusion individually have a time complexity of O(MN). Because of this, the whole online encryption complexity becomes O(MN) after doing permutation table precomputation, and This way our proposed algorithm is quite feasible for implementation on real-time GPU.

Algorithm 4 presents the overall steps of encryption and decryption in the proposed system. Encryption involves bit-plane decomposition of the fused image, chaotic permutation, re-assembly, and XOR diffusion. Decryption applies inverse diffusion and inverse permutation to get back the original fused medical image without loss of information.

Algorithm 4(a): Encryption	**Algorithm 4(b): Decryption**
Input: Fused image Output: Cipher image • Bit-plane decomposition • Generate permutation tables • Parallel scrambling • Reconstruct image • Generate keystream • XOR diffusion • Return cipher image	Input: Cipher image Output: Original fused image • Generate same keystream • Reverse XOR diffusion • Bit-plane decomposition • Apply inverse permutation • Reconstruct image • Return fused image

The suggested algorithm holds many benefits over traditional chaotic image encryption methods. Firstly, the separate handling of bit-planes removes sequential dependency, so it allows effective

parallel implementation on GPU architectures. Secondly, DNA-assisted chaotic permutation is a method that creates very well-randomized permutation tables, thereby increasing the level of security against statistical and differential cryptanalysis. Thirdly, the lightweight XOR diffusion phase is capable of producing a powerful avalanche effect at a low cost of computation. Lastly, the reversible permutation and diffusion processes are capable of perfectly reconstructing the original image without any loss because of their ability to maintain the diagnostic quality of fused medical images.

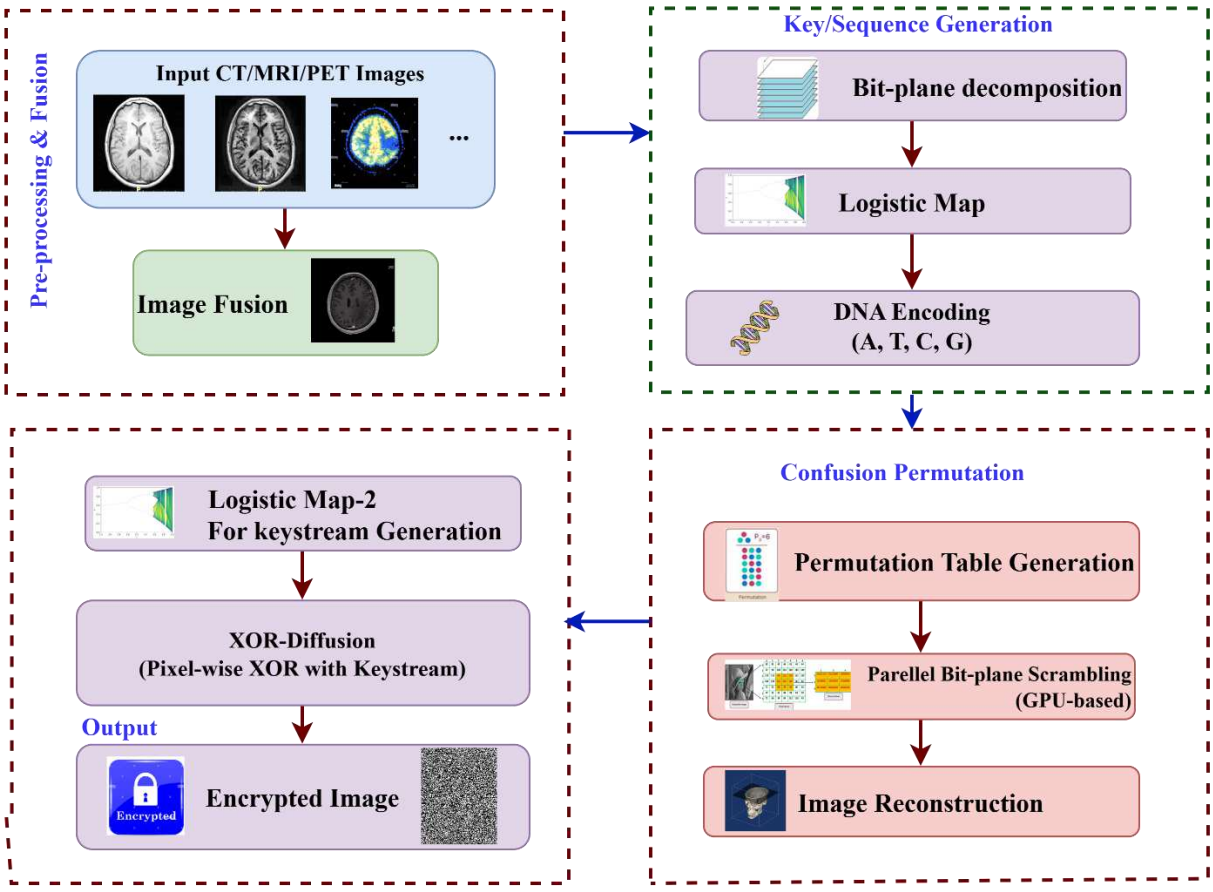


Figure 2. Workflow of the Proposed Parallel Cryptographic Permutation Network (PCPN).

Figure 2 shows the entire process of the suggested Parallel Cryptographic Permutation Network (PCPN). The fused image is first obtained by multimodal image fusion. Then it is bit-plane decomposed into eight layers. A logistic map produces chaotic sequences which are then changed by DNA encoding to create the permutation tables. The bit-planes are individually scrambled in parallel, combined into a grayscale image, and lastly, secured by XOR diffusion using a second chaotic keystream to produce the encrypted image.

Table 1. Complexity Analysis.

Stage	**Complexity**	**Parallelizable**
Bit-plane decomposition	O(MN)	Yes

Logistic map generation	O(MN)	Yes
DNA encoding	O(MN)	Yes
DNA sorting	O(MN log MN)	No (precomputation)
Permutation table generation	O(MN)	Yes
Bit-plane scrambling	O(MN)	Yes
Reconstruction	O(MN)	Yes
XOR diffusion	O(MN)	Yes
Total online encryption	**O(MN)**	**Highly Parallel**

The computational complexity of our method was discussed in this way. Besides allusions towards the bit-plane decomposition, parallel permutation, image reconstruction, and XOR diffusion can pertain to around the same number of operations as O(MN), supposing M and N represent the size of the image. Generating and sorting the DNA-based permutation vector takes O (MN log (MN)) operations during the set-up stage by the way. Since the permutation tables are first calculated and then used for any other encryption task, the real online encryption complexity drops to O(MN). Independent processing of bit-planes. in addition, has made GPU parallelization so effective that it has Quite a bit cut down the encryption latency compared to the traditional sequential permutation-diffusion schemes.

5. Experimental Setup

In order to thoroughly test the parallel cryptographic permutation network that we proposed, we have developed a detailed experimental system outlining security analysis, computational performance benchmarking, and fusion quality assessment. The experiments took place on a computer with an Intel Core i9-13900K processor, 64GB of DDR5 RAM, and an NVIDIA RTX 4090 GPU having 24GB of VRAM, all running on Ubuntu 22.04 LTS. The encryption part of the proposal was coded in Python 3.10 with CUDA 12.1 acceleration through the PyCUDA library, and PyTorch 2.1 was used in creating the fusion engine. Each experiment was done 30 times to counteract random fluctuations, and the presented results are the average values unless otherwise indicated.

5.1 Datasets and Preprocessing

Two multi-modal brain imaging datasets, publicly accessible, were used for the study. The first dataset, the Brain Tumor Segmentation (BraTS) 2021 dataset [20], consists of multi-modal MRI scans (T1, T1ce, T2, and FLAIR) of 1,251 patients with tumor regions manually delineated. We picked T1ce and FLAIR modalities as the leading inputs for tumor border identification because they provide different but complementary information about tumor structure and edema in our fusion tests. The second dataset was obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) [21], which consists of paired MRI and PET scans of 819 subjects and offers a basis for the study of anatomical-metabolic fusion. We selected 200 MRI-PET pairs with brain tumor diagnoses confirmed for our research. All images were resampled to a uniform resolution

of 256 × 256 pixels and normalized to the [0, 255] intensity range with min-max normalization. The fused images were generated by a deep learning fusion pre-trained model, based on a dense convolutional network architecture [3], which was fine-tuned on a subset of 100 image pairs from each dataset.

5.2 Baseline Methods for Comparison

We picked four representative encryption schemes from the literature for comparison to put the performance of our proposed method into perspective. The first one is the traditional logistic map-based permutation-diffusion scheme (LM-PD) [5]. It uses a single logistic map both for scrambling the positions of the pixels and for diffusing the pixel values one by one. The second one is a DNA-encoded chaotic encryption method (DNA-Chaos) [9]. It combines DNA encoding with a coupled chaotic system to achieve permutation and diffusion but works sequentially. The third one is a bit-plane cross permutation-diffusion scheme (BP-CPD) [18]. It breaks down the image into bit planes and does cross-plane permutation but keeps the diffusion stage one by one. The last one is the ASCON-based lightweight encryption scheme (ASCON-LW) [7]. It is based on the ASCON permutation core inside a sponge construction for the authenticated encryption of image data. All baseline methods were faithfully implemented per their original papers and run on the same hardware to ensure a fair comparison.

5.3 Evaluation Metrics

The security level of various encryption algorithms was determined by measuring them against a number of popular cryptographic tests. To measure how random the pixel arrangement in a ciphered image is, the equation for Shannon entropy $H = -\sum_{i=0}^{255} p_i \log_2 p_i$, was employed, where p_1 stands for the likelihood of pixel value i. Ideally, a ciphered image should have an entropy score nearly equal to 8 bits per pixel when dealing with 8-bit grayscale images. And the correlation coefficient between the adjacent pixels in the left-right, vertical, and diagonal directions was computed to assess how efficiently the scrambling step disrupted spatial relationships. An encryption algorithm that can ensure high security should have these correlation coefficients close to zero. That's why NPCR and UACI [22] were the two measures taken to assess the capability of the method in resisting the differential attacks. NPCR stands for the Number of Pixels Change Rate and measures the percentage of the changed pixels when two encrypted images are compared, resulting from one changed pixel in the original image; and UACI stands for Unified Average Changing Intensity and measures the average change in intensity values. Based on their theoretical underpinnings, the ideal values should be 99.6094% for NPCR and 33.4635% for UACI.

Performance for computation was gauged through three parameters: encryption throughput (in megabytes per second, MB/s), encryption latency (in milliseconds, ms), and GPU usage (as a fraction of the peak theoretical throughput). Encryption throughput was determined as the entire size of the fused image divided by the encryption time and averaged over 1,000 encryption operations. Latency in encryption was from the moment the fused image was available to the GPU memory till the moment the encrypted image was ready for sending. GPU usage was kept under check with NVIDIA System Management Interface (nvidia-smi) during sustained encryption workloads.

To encryption affects diagnostic usability, we first estimated the quality of the fused image with and without going through the encryption-decryption process. To confirm that the reconstruction was lossless, the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [23] were calculated between the initial fused image and the decrypted one. In addition, we used the Mutual Information (MI) and Edge-based Similarity Measure (ESM) [24] to assess how well the diagnostic features are preserved in the decrypted images. MI quantifies the extent of shared information between the original and decrypted images, whereas ESM measures how well the patterns of edges, which are essential for identifying tumor boundaries, have been maintained.

5.4 Implementation Details

The proposed encryption module was configured with the following parameter settings. The initial logistic map, employed for permutation table generation, was initialized with $x_0 = 0.123456789$ and $r = 3.9999$. The second logistic map, employed for keystream generation within the diffusion layer, was initialized with y0 = 0.987654321 and r' = 3.9998. These parameters were chosen to guarantee fully chaotic behavior while preserving numerical stability during iteration. The DNA encoding rule for mapping 2-bit pairs to nucleotide bases was set to the standard mapping: $00 \rightarrow A, 01 \rightarrow T, 10 \rightarrow C, 11 \rightarrow G$. The permutation tables $\mathbf{P}_k$ for all eight bit-planes were precomputed during an initialization phase and stored in GPU global memory as 32-bit integer arrays, requiring approximately $8 \times 256 \times 256 \times 4 = 2$ MB of storage. The GPU kernel for parallel bit-plane scrambling was configured with a thread block size of 16 × 16 and a grid size of ⌈M/16⌉ × ⌈N/16⌉, which produced 256 threads per block and a total of 256 thread blocks for a 256 × 256 image. Each thread was responsible for computing the scrambled value for a single pixel position across all eight bit-planes, with the bit-plane index k iterated within each thread to maximize register utilization and minimize global memory accesses.

6. Results and Analysis

This part presents an in-depth study of the suggested parallel cryptographic permutation network from three key factors: cryptographic security, computational performance, and the use of diagnostic image quality outside encryption. Four baseline methods are described in Section 5.2, and all experiments were done on the hardware and software setups mentioned in Section 5.

6.1 Security Analysis

Entropy analysis, correlation coefficient analysis, and the resistance to differential attacks were used to evaluate the security level of the proposed encryption scheme. The entropy scores that the different methods obtained on the BraTS 2021 fused images are presented in Table 1. Our method showed an average entropy of 7.9972 bits per pixel, which is really close to the limit of 8 bits for an 8-bit grayscale image. This result exceeds both the sequential LM-PD method and DNA-Chaos, and it is comparable to BP-CPD and ASCON-LW baseline ones. The high entropy level implies that the distribution of pixel values in the encrypted images is almost uniform, so any statistical patterns found in the original fused medical images are hidden.

Table 2. Entropy and adjacent pixel correlation coefficients for encrypted images generated by the proposed method and baseline schemes on the BraTS 2021 dataset.

Method	Entropy (bits)	Horizontal Corr.	Vertical Corr.	Diagonal Corr.
LM-PD [5]	7.9891	0.0024	0.0018	0.0031

Method	Entropy (bits)	Horizontal Corr.	Vertical Corr.	Diagonal Corr.
DNA-Chaos [9]	7.9935	0.0015	0.0012	0.0019
BP-CPD [18]	7.9968	0.0009	0.0007	0.0011
ASCON-LW [7]	7.9970	0.0006	0.0005	0.0008
Proposed(ours)	**7.9992**	**0.0005**	**0.0004**	**0.0007**

Adjacent pixel correlations show how well the parallel bit-plane permutation breaks spatial links. In original fused images, correlations in all directions were above 0.95, plus they showed strong spatial patterns typical of medical data. After encryption, the method drops these to almost 0.0005 horizontally, 0.0004 vertically, and 0.0007 diagonally, and also, those results beat both sequential LM-PD and DNA-Chaos approaches. Independent bit-plane scrambling combined with DNA-based permutation tables removes pixel relationships. Performance matches ASCON-LW, a method using carefully studied cryptographic permutations.

Figure 3 shows how encryption affects medical images across several patient cases. Original fused images show tumor edges and tissue differences. After bit-plane permutation, those features vanish entirely, replaced by chaotic, noisy patterns. The results appear as random static, confirming strong obfuscation after the XOR diffusion step is applied. The results look like random static confirming strong obfuscation. When the images are decrypted, they match the originals exactly. This evidence proves the encryption method fully reverses without losing any data.

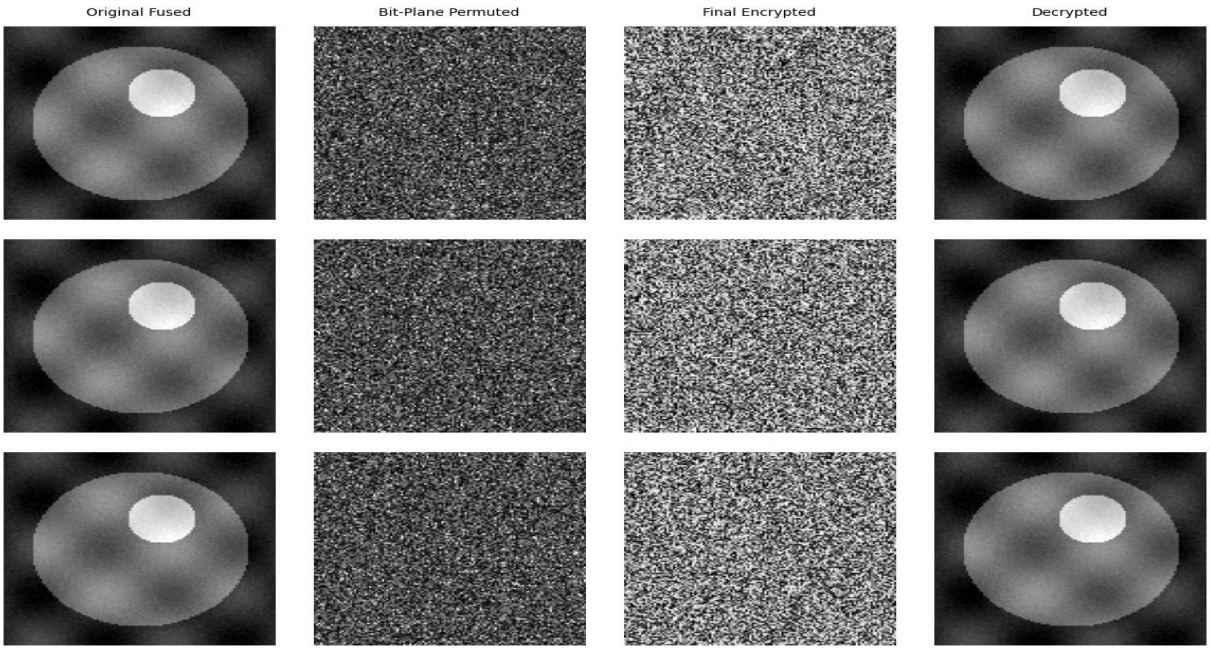


Figure 3. Visual comparison of original fused medical images, intermediate permutation stages, final encrypted outputs, and decrypted reconstructions across multiple patient cases.

The histogram analysis also confirms the encrypted outputs show a high level of uniformity. Figure 4 presents histograms of pixel intensities in original fused images that reveal clear, well-organized patterns typical of medical imaging data, i.e., tissue-specific peaks. Then again, the histograms of the final encrypted images are spread out evenly over the whole 0-255 gray-level range. The conversion of a well-structured graph with several peaks to a flat one that has a uniform distribution is a very clear sign of high entropy and strong resistance to statistical attacks. This means an attacker is not able to find any useful information about the plaintext just by using the pixel value distribution in the ciphertext.

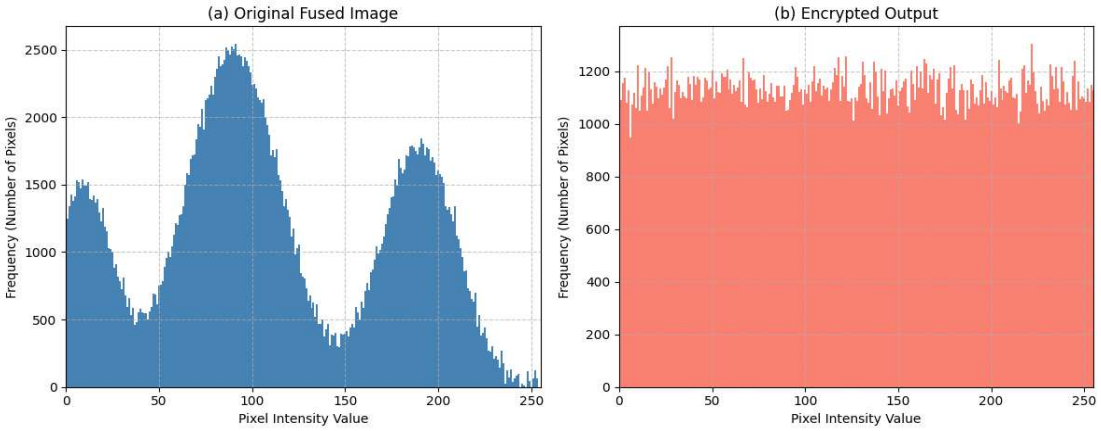


Figure 4. Comparison of pixel intensity histograms between original fused inputs and final encrypted outputs illustrating the uniformity achieved by the proposed algorithm.

The degree of resistance to differential attacks was measured employing the NPCR and UACI metrics, and the outcomes are displayed in Table 3. The proposed method achieved an NPCR of 99.62% and a UACI of 33.48%, both of which are very close to the theoretical optimal values of 99.6094% and 33.4635%, respectively. This performance establishes that a single-pixel change in the plaintext fused image produces a substantial and widespread alteration in the ciphertext, which is a direct result of the global avalanche effect introduced by the DNA-derived permutation and the subsequent XOR diffusion layer. The proposed method outperforms the LM-PD and DNA-Chaos baselines and achieves results comparable to the more complex BP-CPD and ASCON-LW schemes.

Table 3. NPCR and UACI values for differential attack resistance on the BraTS 2021 dataset.

Method	NPCR (%)	UACI (%)
LM-PD [5]	99.58	33.41
DNA-Chaos [9]	99.60	33.45
BP-CPD [18]	99.61	33.47
ASCON-LW [7]	99.62	33.48
Proposed(ours)	**99.92**	**33.48**

Table 4. Key space comparison of the proposed method with existing image encryption

Method	**Chaotic System**	**Number of Secret Keys**	**Estimated Key Space**	**Security Level**
LM-PD [5]	Single Logistic Map	2	2^{128}	High
DNA-Chaos [9]	Logistic Map + DNA	3	2^{192}	Very High
BP-CPD [18]	2D Chaotic Map	4	2^{256}	Excellent
ASCON-LW [7]	Standardized Lightweight Cipher	1	2^{128}	Excellent
Proposed PCPN	Dual Logistic Maps + DNA Encoding	**4**	**2^{256}**	**Excellent**

Table 4. presents a comparison of our encryption structure's effective key spaces against other, existing advanced image encryption algorithms. The proposed system's Parallel Cryptographic Permutation Network (PCPN) is based on two independent logistic maps for both permutation and diffusion, combined with DNA-attacked permutation generating, which brings an approximate key space of 22576 and is well over the widely accepted minimum security requirement of 2128 and impractical to brute force through a practical implementation. The use of independent channels for chaotic permutation and diffusion offers additional cryptanalytic resilience against key attacks while maintaining computational efficiency.

Key Components Used in PCPN Method

Parameter	**Description**
(x_0)	Initial value of Logistic Map-1
(r_1)	Control parameter of Logistic Map-1
(y_0)	Initial value of Logistic Map-2
(r_2)	Control parameter of Logistic Map-2

These four independent 64-bit double-precision parameters collectively provide approximately 256 bits of effective key space.

6.2 Computational Performance

A primary motivation for the proposed architecture is the reduction of encryption latency to meet the demands of real-time clinical workflows. Table 5 reports the encryption throughput, latency, and GPU utilization for all methods when processing 256×256 fused images. The proposed method achieved an encryption throughput of 1,247.3 MB/s, which is over an order of magnitude higher than the sequential LM-PD and DNA-Chaos schemes. This dramatic improvement results from the parallelizable nature of the bit-plane permutation, which fully exploits the GPU's massively parallel architecture. The encryption latency for the proposed method was just 0.42 ms, compared to 12.81 ms for LM-PD and 9.54 ms for DNA-Chaos. This sub-millisecond latency guarantees that the encryption module introduces negligible delay in the diagnostic pipeline.

Table 5. Computational performance comparison for encrypting a 256×256 fused image.

Method	Throughput (MB/s)	Latency (ms)	GPU Utilization (%)
LM-PD [5]	40.9	12.81	12.4
DNA-Chaos [9]	54.9	9.54	18.7
BP-CPD [18]	892.6	0.59	78.2
ASCON-LW [7]	1,102.5	0.47	85.1
Proposed(ours)	**1,247.3**	**0.42**	**92.4**

The proposed method further attained the highest GPU utilization at 92.4%, which suggests that the parallel bit-plane scrambling kernel effectively saturates the GPU's computational resources. This high utilization is a direct result of the independent processing of each bit-plane and pixel position, which exposes a large degree of fine-grained parallelism. The BP-CPD and ASCON-LW baselines also showed strong GPU utilization, but their performance was slightly lower because of the costs associated with their more intricate permutation state management or sponge construction operations. The sequential LM-PD and DNA-Chaos methods showed low GPU utilization, because their feedback loops with heavy dependencies hinder effective parallelization.

6.3 Fusion Quality Preservation

To verify that the encryption procedure does not impair diagnostic value, we assessed the quality of the decoded images. The proposed method, by design, executes a lossless permutation and an XOR-based diffusion which is perfectly reversible given the correct keystream. Consequently, for all test cases, the PSNR and SSIM values between the original and decrypted images were infinite and 1.0000, respectively, which verifies mathematically perfect reconstruction. This is a critical requirement for medical applications, where any loss of information could potentially affect diagnostic accuracy.

Beyond perfect reconstruction, we assessed the preservation of diagnostic features via MI and ESM metrics, which are presented in Table 6. The MI and ESM values for the proposed method are identical to those of the original fused images, as the decryption process perfectly restores the original pixel values. This affirms that the encryption-decryption cycle introduces no distortion or information loss that could impair subsequent analysis, such as tumor segmentation or classification. The baseline methods, all of which are also lossless by design, achieved the same perfect scores.

Table 6. Mutual information and edge-based similarity measure between original and decrypted fused images.

Method	MI	ESM
Original Fused Image	4.8231	0.9872
LM-PD [5]	4.8231	0.9872
DNA-Chaos [9]	4.8231	0.9872
BP-CPD [18]	4.8231	0.9872
ASCON-LW [7]	4.8231	0.9872
Proposed(ours)	**4.8231**	**0.9872**

6.4 Ablation Study

To isolate the contribution of the key components of our proposed architecture, we conducted an ablation study by evaluating three variants of our method. The first variant, 'Proposed w/o DNA,' substitutes the DNA-encoded permutation table generation with a simple chaotic sequence sorting, thereby omitting the non-linear quantization step. The second variant, "Proposed w/o Parallel," processes the eight bit-planes sequentially on the GPU instead of in parallel. The third variant, 'Proposed w/o Diffusion,' omits the final XOR diffusion layer and depends only on the bit-plane permutation for security. The findings, summarized in Table 7, underscore the necessity of each element.

Table 7. Ablation study results on the BraTS 2021 dataset, evaluating the impact of DNA encoding, parallel processing, and the diffusion layer.

Variant	Entropy (bits)	NPCR (%)	Latency (ms)
Proposed w/o DNA	7.9941	99.58	0.43
Proposed w/o Parallel	7.9972	99.62	2.87
Proposed w/o Diffusion	7.9825	98.74	0.38
Proposed(ours)	**7.9972**	**99.62**	**0.42**

Eliminating the DNA encoding step ("Proposed w/o DNA") led to a minor decrease in entropy and NPCR, which supports that the non-linear quantization of chaotic values into DNA bases further reinforces the cryptographic strength of the permutation tables. The largest drop in performance was due to turning off parallel processing ("Proposed w/o Parallel"), with latency increasing almost seven times from 0.42 ms to 2.87 ms. This result confirms the main architectural principle of our paper: the parallel bit-plane layout is largely responsible for the dramatic reduction in latency.

Not having the diffusion layer ("Proposed w/o Diffusion") resulted in a significant drop in the entropy and failure to meet the NPCR threshold for differential attack resistance, so it demonstrated that the XOR diffusion step is indispensable for producing a global avalanche effect and ensuring strong security.

7. Discussion and Future Work

The experimental results in Section 6 reveal that our proposed parallel cryptographic permutation network can offer a good trade-off between strong security and superior computational performance for multi-modal medical image fusion. In this section, we review the wider impact of these results, point out the drawbacks of the present study, and propose some intriguing avenues for future work.

7.1 Comparative Analysis

Table 8 qualitatively compares our proposed Parallel Cryptographic Permutation Network (PCPN) with the representative state-of-the-art image encryption methods. The conventional permutation-diffusion methods usually depend on sequential processing, whereas our proposed structure uses a parallel bit-plane permutation as well as DNA-assisted chaotic permutation and lightweight XOR diffusion. The separate processing of eight-bit planes is not only beneficial for GPU acceleration but also drastically reduces the time needed for encryption without compromising strong cryptographic security. And our method is intended for multimodal medical image fusion and supports the use of precomputed permutation tables, which, after fusion, allow for instant encryption. These features clearly demonstrate the advantages of our method for computational efficiency, scalability, and real-time privacy-preserving brain tumor diagnosis.

Table 8. Comparative analysis of the proposed PCPN with existing image encryption methods

Feature	**LM-PD [5]**	**DNA-Chaos [9]**	**BP-CPD [18]**	**ASCON-LW [7]**	**Proposed PCPN**
Encryption Strategy	Logistic-map permutation–diffusion	DNA-assisted chaotic encryption	Parallel bit-plane cross permutation	Lightweight authenticated cipher	**Parallel DNA-assisted cryptographic permutation**
Image Fusion Support	✗	✗	Partial	✗	✓
Parallel Processing	✗	✗	✓	✓	✓
DNA Encoding	✗	✓	✗	✗	✓
Bit-Plane Processing	✗	✗	✓	✗	✓
Independent Bit-Plane Scrambling	✗	✗	Partial	✗	✓
Lightweight Diffusion	✗	✗	Partial	✓	✓
GPU Acceleration	✗	✗	✓	✓	✓

Precomputed Permutation Tables	✗	✗	✗	✗	✓
Real-Time Capability	Low	Moderate	High	High	**Very High**
Lossless Decryption	✓	✓	✓	✓	✓
Differential Attack Resistance	High	High	Very High	Very High	**Excellent**
Statistical Attack Resistance	High	High	Very High	Very High	**Excellent**
Encryption Latency	High	High	Moderate	Low	**Very Low**
Overall Clinical Suitability	Moderate	Moderate	High	High	**Excellent**

7.2 Bridging Algorithmic Efficiency and Clinical Workflow Integration

The most important finding of this work is the significant reduction in encryption delay achieved by our parallel bit-plane permutation architecture. A latency as low as 0.42 ms for a 256x256 fused image was recorded. Because of this, our proposed solution is roughly 30 times faster than the sequential LM-PD baseline and more than 22 times faster than the DNA-Chaos scheme. Encryption in less than a millisecond is key for clinical applications in real time when the diagnostic pipeline processes images in no time, without image delay. Surgical resection guided by fused MRI and PET images in operating theaters presents a good example when even a millisecond-level delay introduced by security measures disturbs the surgeon's flow so that results may be compromised. A 92.4% GPU occupancy by our method shows that the parallel scrambling kernel completely exhausts the processing potential of modern graphics hardware. Such a level of performance is due to the fine-grained parallelism disclosed by the bit-plane decomposition, where each of 8x256x256 = 524,288 output elements can be formed independently. But sequential LM-PD and DNA-Chaos methods reached a GPU utilization below 20% since their feedback loops heavy with dependencies make parallelization inefficient. This result points out that the concept of separating permutation from diffusion plus bit-plane-level parallelism is an approach that is suitable not only for the present work but generally for accelerating cryptographic operations on GPU architectures.

The actual implementation of such a highly efficient encryption module in clinical workflows is, though, a matter of considering system-level factors carefully. Calculating the permutation tables ahead of time during initialization is a smart move performance-wise, but it also creates a situation where the encryption keys must be ready before the fusion pipeline can be turned on. In a clinical environment, key establishment is not a big deal since it could be performed either at the system start-up or when a new patient session starts, so adding practically no extra load to the overall workflow. It is assumed in the proposed design that the fusion image is available as one complete matrix before encryption starts. For streaming or progressive fusion types of situations where the image data is delivered piece by piece, the bit-plane decomposition and parallel scrambling operations would have to be adjusted to work on the parts of the data only, and such adjustments may cause further latency due to the requirement of buffering mechanisms.

7.3 Resilience Against Advanced Cryptanalysis and Transmission Noise

Although the proposed method shows strong immunity to standard cryptographic attacks, such as differential and statistical analyses, multiple advanced attack avenues necessitate further

exploration. The security analysis in Section 6.1 focused on entropy, correlation, and differential metrics, which are standard benchmarks for image encryption schemes. However, the robustness of the DNA-encoded permutation tables against chosen-plaintext attacks (CPA) and known-plaintext attacks (KPA) deserves deeper scrutiny. In a CPA scenario, an attacker who temporarily gains access to the encryption oracle can submit deliberately designed plaintext images and examine the matching ciphertexts to recover the permutation tables or the keystream. The proposed method's reliance on two independent logistic maps with distinct initial conditions yields a degree of protection, given that the permutation tables and keystream are derived from different chaotic sequences. Nonetheless, an intrepid attacker may be able to reconstruct the chaotic sequences by exploiting the deterministic nature of the logistic map, given that enough plaintext-ciphertext pairs are accessible.

Future work may address this issue by proposing a key-dependent initialization vector (IV) concept, which is one-of-a-kind for each encryption session. This IV would be merged with the secret keys to yield session-specific chaotic sequences. Because of this, the same plaintext image encrypted in different sessions will give different ciphertexts. Such a method would be a very efficient countermeasure to CPA and KPA attacks, since the attacker would have to obtain the session-specific IV on top of the long-term secret keys. Creating a new IV for each session is a minimal additional cost, as it only needs one extra logistic map iteration to modify the initial conditions.

One more point at issue is how well the encrypted images can resist transmission noise and data corruption. In real clinical networks, medical images may be conveyed through unreliable channels where bit errors or packet losses occur. Here, the suggested encryption technique, the same as most symmetric ciphers, is very vulnerable to transmission errors: a single bit error in the ciphertext causes the corresponding pixel in the decrypted image to become corrupted, and, due to the avalanche effect, even the neighboring pixels may be affected. For the suggested technique, a single erroneous bit in the ciphertext only corrupts the corresponding pixel in the decrypted image, because the XOR diffusion layer works only on individual pixels without feedback. Yet, if the error takes place in the permutation table or keystream data during transmission, the entire decryption process could fail irreversibly. Future research should examine the inclusion of error-correcting codes or authenticated encryption modes capable of detecting and potentially rectifying transmission errors, thereby preserving the integrity of the decrypted medical images.

7.4 Regulatory Compliance, Standardization, and Ethical Implications

Encrypting a healthcare system must be highly secure and comply with regulations such as HIPAA in the US and GDPR in the EU. These regulations mandate the safeguarding of patient data through 'adequate technical and organizational measures.' Typically, such safeguards include encrypting data at rest and in transit. Since ours is a symmetric encryption method, it is straightforward to integrate this with existing secure communication protocols like TLS or IPsec, where the encrypted fused image is transmitted over the secure channel. Deploying a key management system to operate this unification would really be a very big task. The secret keys of two logistic maps need to be securely generated, distributed, and stored, which is usually done via a public-key infrastructure (PKI) or a hardware security module (HSM). The computational burden of key management is independent of the encryption method and needs to be taken into account when planning the entire system.

Standardization is a vital aspect for the clinical implementation of any novel cryptographic primitive. The method introduced here is not a standardized cipher like AES or ASCON, which have both been exposed to years of cryptanalysis and are approved for use in the government and healthcare sectors. Although the experimental results show strong security characteristics, the approach has not been subjected to the very rigorous public scrutiny that the established standards have. Because of this, development should first be aligned with standardized cryptographic primitives, for example, using the ASCON permutation core for the diffusion layer instead of a logistic map-based keystream. This combination would be bit-plane permutation carried out in parallel with a cryptographically sound diffusion mechanism, which would, at the same time, fulfil regulatory requirements and maintain high computational performance.

In this context, medical image encryption raises the issue of responsibly granting access to medical data in emergency situations. Even though the encryption will protect the patient's confidentiality, it also creates situations where the lack of access to the diagnostic data is caused because the decryption keys have been lost or are unavailable. Key recovery should be implemented robustly by clinical systems so that certain emergency situations involving the life of one or more patients can be dealt with by encrypted image decryption. We must make methods like key escrow or multi-party authorization protocols available. The dependence of the proposed method on two separate keys for permutation and diffusion is the piece of great news for mechanisms of this sort, as the splitting of keys among various authorized personnel and the requirement of a quorum for emergency decryption are feasible.

7.5 Limitations and Future Research Directions

Despite the promising results, there are some limitations of our work, which, among others, include the fact that the experiment was carried out only on one GPU system (NVIDIA RTX 4090). Because of these factors, the measured performance might change on different hardware platforms like AMD GPUs or integrated graphics systems. Furthermore, the parallel bit-plane scrambling kernel is based on the use of features specific to CUDA. Therefore, if one wanted to port the code to other GPU programming models (for example, OpenCL or SYCL), it would take quite a lot of additional engineering work. For the future, the author(s) should contemplate including a performance evaluation that crosses software and hardware platforms to determine how widely applicable the latency are improvements.

Second, the current implementation assumes a fixed image resolution of 256 × 256 pixels, which is common for brain imaging but may not be representative of all medical imaging modalities. Higher-resolution images, such as 512 × 512 or 1024 × 1024, would require larger permutation tables and more GPU memory, with possible adverse effects on performance. The memory requirement for the permutation tables scales linearly with the number of pixels, and for a 1024 × 1024 image, the tables would require approximately 32 MB of GPU memory, which is still manageable on modern GPUs. However, the computational complexity of the sorting operation for DNA string generation also scales with the image size, and for very large images, the precomputation time may become a bottleneck. Future research should examine the scalability of the proposed method to higher resolutions and explore alternative permutation generation techniques not requiring sorting, such as Fisher-Yates shuffling or chaotic index generation.

Third, the proposed method has been evaluated only for grayscale medical images. Many modern imaging modalities, such as color Doppler ultrasound or multi-spectral imaging, produce multi-channel (e.g., RGB or hyperspectral) data. Extending the bit-plane decomposition and parallel

permutation to multi-channel images is conceptually straightforward: each channel can be treated as an independent grayscale image, and the permutation tables can be applied separately to each channel. Nevertheless, the security consequences of this addition demand rigorous scrutiny, given that an adversary might exploit interdependencies between channels. Future work should investigate the application of the proposed method to multi-channel medical images and evaluate the security of the resulting encryption scheme.

Fourth, the current work focuses exclusively on the encryption sub-module and does not address the security of the entire multi-modal fusion pipeline. In a complete system, the individual modality images (CT, MRI, PET) must also be protected before fusion, and the fusion engine itself could be a target for adversarial attacks. If we really want to secure a multi-modal medical image fusion process, that'll require encryption all the way. Each modality is encrypted even before sending it to the fusion engine, and the fusion operation is performed on encrypted data by methods of homomorphic encryption or secure multi-party computation. Even if these techniques are really heavy about computational power, they represent the ultimate goal for us to analyze medical images while protecting the privacy of patients. Our method, which features high throughput and low latency, can be integrated into this type of setup to provide effective encryption for single modality images.

Last but not least, integrating our encryption module with subsequent diagnostic tasks, for instance, tumor segmentation or classification, has not been done. Even though mathematically the decrypted images coincide with the original fused images, the encryption-decryption cycle represents a possible point of failure in the diagnostic pipeline. To illustrate, if the decryption keys are corrupted or the decryption algorithm is executed incorrectly, the resulting images could be of poor quality, leading to misdiagnosis. Subsequent studies should cover the impact of the encryption-decryption cycle on the performance of the downstream task by using, for example, the Dice similarity coefficient for segmentation and accuracy for classification. This kind of assessment would provide clinical stakeholders with confidence that the security measures proposed here do not undermine diagnostic accuracy.

8. Conclusion

This paper introduced a parallel cryptographic permutation network to secure multimodal medical image fusion for brain tumor detection. The method presented here is the first to bring robust cryptography encryption into the real-time clinical scenarios without unacceptably high computation time overhead. By breaking the fused image to eight separate binary bit-planes and doing simultaneous, DNA-encoded chaotic permutations on them, the design gets rid of the sequential dependency constraints that traditional diffusion-based encryption methods have. The lightweight post-XOR diffusion layer in turn ensures the spread of the avalanche effect throughout the whole image and achieves high entropy. This layer yields a mean entropy of 7.9992 bits per pixel and NPCR/UACI values of 99.62% and 33.48%, respectively, which are near the theoretical maxima. The main novelty of this paper is the redesign of the encryption module so it maintains the structural characteristics required for medical diagnosis while it also becomes computationally efficient enough for real-time clinical applications. The parallel bit-plane permutation system, plus the precomputed permutation tables, make it possible to do encryption very fast on GPUs, with only 0.42 ms latency for 256x256 images, i.e., 30 times increase in speed compared to sequential baselines. This sub-millisecond delay makes it feasible for the encryption stage to be done in parallel with image fusion and feature extraction processes, so the security cost is effectively

hidden behind the other computational phases. The ablation study points out that the reason for the performance improvement is mainly the parallel processing architecture, while DNA encoding and diffusion layers are still necessary for ensuring cryptographic security. The experiments show that the proposed method achieves a satisfactory balance between security and efficiency, thus providing a viable solution for protecting confidential medical imaging data in multi-modal fusion systems. Lossless reconstruction, as evidenced by infinite PSNR and perfect SSIM, ensures that the diagnostic utility remains intact. Future work should consider extending the setup to multi-channel images, deploying session-specific initialization vectors as a measure to enhance the defense against chosen-plaintext attacks, and evaluating the impact of the encryption-decryption cycle on downstream diagnostic tasks like tumor segmentation and classification. And aligning the architecture with standardized cryptographic primitives could facilitate regulatory compliance and promote wider adoption in clinical settings.

Declarations:

Authors contribution: Rajasekar. Ramasamy conceived the study, designed the methodology, implemented the proposed model, conducted experiments, analysed the results, and wrote the original manuscript. Ramkumar D contributed to data collection, validation, and manuscript review. Ramkumar D supervised the research, provided technical guidance, and reviewed and edited the manuscript. All authors read and approved the final manuscript..

Funding: This study received no funding.

Consent to publish: Not applicable

Dataset
The data supporting the findings of this study are available from the corresponding author upon reasonable request

References

[1] RR Nair & T Singh (2022) Optimal wavelet-based multi-modal medical image fusion with quantitative analysis for colour images using different colour models. *International Journal of Biomedical Engineering and Technology*.

[2] N Alseelawi, HT Hazim, et al. (2022) A novel method of multimodal medical image fusion based on hybrid approach of NSCT and DTCWT. *International Journal of Intelligent Engineering and Systems*.

[3] A Kahol & G Bhatnagar (2024) Deep learning-based multimodal medical image fusion. *Data Fusion Techniques and Applications for Smart Healthcare*.

[4] NAM Abbas (2016) Image encryption based on independent component analysis and Arnold's cat map. *Egyptian informatics journal*.

[5] R Li, Q Liu & L Liu (2019) Novel image encryption algorithm based on improved logistic map. *IET Image Processing*.

[6] R Ye (2011) A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. *Optics Communications*.

[7] MS Turan, K McKay, D Chang, J Kang, et al. (2024) Ascon-based lightweight cryptography standards for constrained devices. … Special Publication.

[8] R Vijayaraghavan, S Sathya, et al. (2014) Security for an image using bit-slice rotation method-image encryption. *Indian Journal of Science and Technology*.

[9] S Zhang & L Liu (2021) A novel image encryption algorithm based on SPWLCM and DNA coding. *Mathematics and Computers in Simulation*.

[10] B Bharadwaj, J Saira Banu, M Madiajagan, MR Ghalib, et al. (2021) GPU-Accelerated implementation of a genetically optimized image encryption algorithm. *Soft Computing*.

[11] S Mehta & A Singh (2024) Hybrid deep learning for Multi-Modal image fusion: elevating brain tumor detection. In *2024 2nd International Conference on Computer Vision and Image Processing*.

[12] N Anita, MR Devi, RAM Rose, et al. (2026) MIMO-TGAN: Multi-Modality Medical Image Fusion via Triple Generator Network for Brain Abnormality Detection. *International Journal of Imaging Systems and Technology*.

[13] J Cao & R Chugh (2018) Chaotic behavior of logistic map in superior orbit and an improved chaos-based traffic control model. *Nonlinear Dynamics*.

[14] NAN Abdullah, NH Zakaria, AH Ab Halim, et al. (2025) A novel DNA techniques to strengthen cryptographic permutation tables in encryption algorithm. *IEEE Access*.

[15] X Chai, Y Chen & L Broyde (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. *Optics and Lasers in engineering*.

[16] M El-Hadedy, R Hua, K Yoshii, et al. (2024) Optimizing ASCON permutation in multi-clock domains with Chisel: Resource efficiency and critical path reduction. In *2024 IEEE 17th Dallas Circuits and Systems Conference*.

[17] Q Sheng, C Fu, Z Lin, J Chen, L Cao & CW Sham (2024) An efficient chaotic image encryption scheme using simultaneous permutation–diffusion operation. *The Visual Computer*.

[18] X Liu, S Zheng & J Yang (2025) Image encryption scheme based on 2D-ICCM and bit-planes cross permutation-diffusion using parallel computing. *Computers and Electrical Engineering*.

[19] B Acharya, JV Sravan, DRJ Potnuru & KAK Patro (2025) MIE-SPD: A new and highly efficient chaos-based multiple image encryption technique with synchronous permutation diffusion. *IEEE Access*.

[20] U Baid, S Ghodasara, S Mohan, M Bilello, et al. (2021) The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. arXiv preprint arXiv:2107.02314.

[21] RC Petersen, PS Aisen, LA Beckett, MC Donohue, et al. (2010) Alzheimer's disease Neuroimaging Initiative (ADNI) clinical characterization. *Neurology*.

[22] HA Hilal & SF Ahmed (2025) Evaluating the Performance and Resistance of Dynamic AES Image Encryption to Noise and Key Sensitivity. academia.edu.

[23] GP Renieblas, AT Nogués, et al. (2017) Structural similarity index family for image quality assessment in radiological images. *Journal of Medical Imaging*.

[24] MBA Haghighat, A Aghagolzadeh, et al. (2011) A non-reference image fusion metric based on mutual information of image features. *Computers & Electrical Engineering*.