Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

New Pseudo-Random Image Encryption Through Spatio-Temporal Method Assignments

GMIRA Faiq

LT2I Laboratory, ESTF, Sidi Mohamed Ben Abdellah University, Fez, Morocco.
C3S Laboratory, ESTC, Hassan II University, Casablanca, Morocco.

E-mail: faiq.gmira@usmba.ac.ma

Abstract

This paper proposes an image encryption method using a spatio-temporal cryptographic diversification strategy. The implemented method, called Pseudo-Random Image Encryption System (PRIES), is based on three conceptual components: What (choice of methods), Where (spatial segmentation) and How (temporal assignment). The What component uses a cryptographic repository of image encryption methods (CRIEM) to pseudo-randomly provide, at each encryption cycle, a sequence of several different methods. The Where component employs a radial segmentation strategy inspired by the wavelet domain, segmenting the image into various angular segments to break spatial correlations. Finally, the How component implements a Markov model to pseudo-randomly assign the selected sequences, defined in the What component, to the angular segments defined in the Where component. On the operational level, experimental data demonstrate high robustness and cryptographic performance: an entropy close to 7.999, a pixel change rate (NPCR) greater than 99.6%, a uniqueness of pixel change index (UACI) approaching 33.46% and a very reduced processing time.

Keywords: Image Encryption, Pseudo-Randomness, Cryptographic Diversity, Spatio-Temporal Cryptography, Security Analysis, Image Segmentation, Markov Chain, NPCR, UACI, Information Entropy.

1. Introduction

In recent years, images have been exposed to evolving and sophisticated threats, which require a defense that is both robust and adaptable [1-3]. Image encryption, which has been receiving increased attention in light of its use, has traditionally relied on single and static cryptographic schemes [4]. These classic approaches, which have proven themselves with some success, nevertheless suffer from a structural flaw: their predictable and pre-coded nature, which is susceptible to cryptanalysis and attacks based on statistical methods It is, in particular, to overcome this weakness that work in image encryption is pursuing more random and variational approaches [4, 5].

Pseudo-random image encryption is an innovative approach that aims to overcome the vulnerabilities of conventional static methods with a vision of continuous improvement, by randomizing the techniques [6], keys, and parameters used in the encryption process. In fact,

DOI: https://doi.org/10.36548/jiip.2025.4.003
Received: 16.08.2025, received in revised form: 15.09.2025, accepted: 29.09.2025, published: 10.10.2025

the fundamental idea of this approach is to make all operations unpredictable through an uncertain method, which is detrimental to any analysis and cryptanalytic attack [1,5].

Image encryption pseudo-randomization is highly promising for many reasons, including its variability, its resistance to classic attacks based on statistical cryptanalysis and its ability to adapt to evolving threats, which contributes to its greater reliability [2, 7].

Pseudo-random image encryption should therefore, in principle, meet image security requirements by necessitating adaptability that can significantly increase resistance to escalating attacks [3, 7-9].

Practically, the use of this pseudo-random approach proves very useful, particularly in sensitive areas such as biometric systems or embedded devices [6, 7], where it would be inappropriate to base security on mono-algorithmic static encryption.

This article is organized as follows: The second part presents the methodology and pseudo-codes for the pseudo-random image encryption process. The third section experiments and validates its robustness and theoretical value using appropriate metrics. The fourth part of the article concludes by summarizing the results and discussing future research prospects.

2. Proposed Encryption System Architecture

An innovative pseudo-random image encryption system is proposed, based on an approach of joint spatio-temporal cryptographic diversification. Its architecture is structured according to a conceptual model articulated around three complementary components: What (method selection), Where (spatial segmentation), and How (temporal orchestration). The main idea using a UML sequence diagram, is illustrated below (Figure 1).

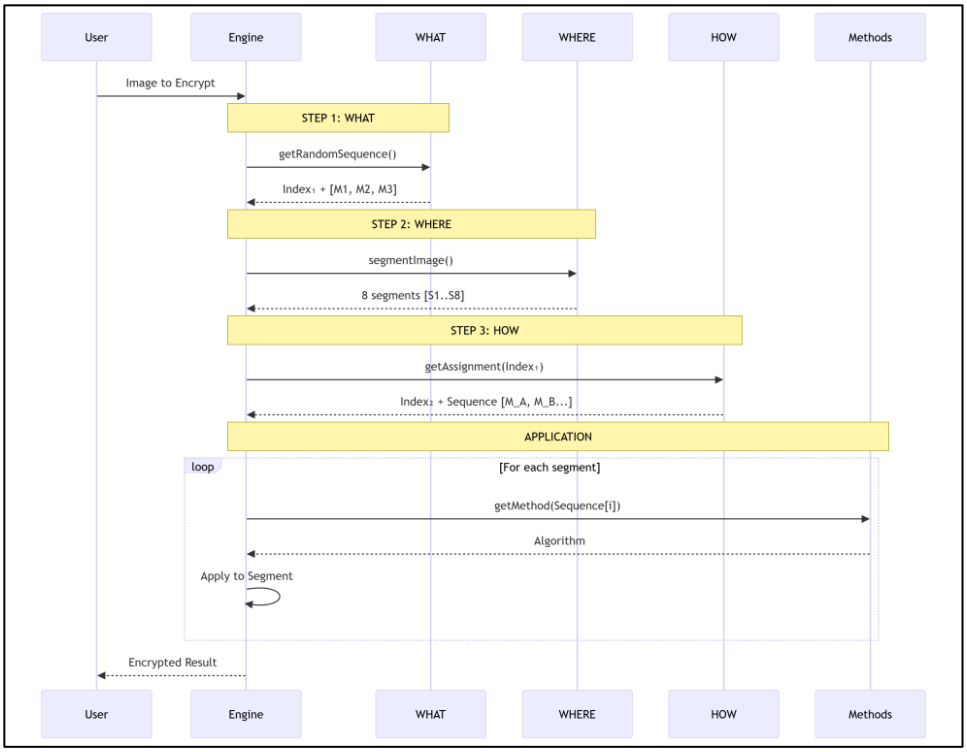

Figure 1. UML Sequence Diagram of the Proposed System

This UML sequence diagram depicts an entire encryption cycle by demonstrating how queries to Register1 and Register2, configure the system to produce a pseudo-random and unique image encryption. The encryption cycle begins with an input image; then Register1 is queried using Index 1 to return a pseudo-random sequence of encryption methods (What). In parallel, the input image is subdivided into an 8 symmetric overlay segments (Where); Following that, Register2 is queried using Index 2 to return a pseudo-random mapping of the encryption methods for each segment (How). Each segment is encrypted in parallel (Encrypt); finally, all segments are merged to create the output Cipher-Image (Merge & Output).

The pseudo-random spatio-temporal behaviour, resistant to statistical analyses [8-11], meets the basic cryptographic standards of PRIES.

2.1 Method Selection Component

To avoid the vulnerability of static single-method encryption, a Cryptographic Repository of M Image Encryption Methods (M-CRIEM) is implemented. At each cycle, an ordered sequence of N distinct methods is pseudo-randomly selected from this repository, thus generating a cryptographic combination that is pseudo-random , unique, and unpredictable, which is then applied according to spatio-temporal assignments to introduce robust image encryption [9,12].

2.1.1 Optimal Parameter Selection Methodology

First, the search for the parameters M (repository size) and N (Selection Size**)** is formulated as a constrained combinatorial optimization problem. The objective is to maximize cryptographic diversity, which directly depends on the number of permutations of N distinct elements chosen from M [9, 10].

The configuration space using the combinatorial arrangement function is modeled as follows:

$$A\,(M, N) = M! / (M - N)!) \tag{1}$$

This function quantifies the resistance to brute-force attacks, where an adversary would have to explore a space of size A (M, N) to guess the correct sequence [13].

For the selection of parameter pairs (M, N), three important constraints must be simultaneously satisfied:

- Storage constraint:

To ensure that each configuration is represented by a 1-byte index, it is imperative to respect the following condition:

$$A(M,N) \leq 255 \tag{2}$$

- Diversity constraint:

To ensure sufficient cryptographic diversity:

$$A(M,N) \geq 100 \tag{3}$$

- Complexity constraint:

$$N \cdot T_{enc} \leq T_{max} \quad (4)$$

In order to limit the computational complexity induced by algorithmic diversification:

Or:

- T_{enc}: Average execution time per encryption method
- T_{max}: Maximum acceptable processing time per encryption cycle

Although an analytical solution is possible, a graphical solution proves more intuitive for visually identifying optimal parameters under constraints. The optimal pair (M,N) that maximizes the arrangement function A(M,N) is determined by systematically exploring the parameter space.

To achieve this, the exploration strategy combines [14]:

- a scan for M constant and N variable: evaluation of all valid N values for each M.
- a scan for N constant and M variable: evaluation of all valid M values for each N.

A reasonable search space is chosen as follows:

- $M \in \{3, 4, 5, 6, 7\}$
- $N \in \{2, 3, 4, 5\}$ with $N \leq M$

The corresponding visualization of the cardinality of the arrangements A(M,N) is shown in (Figure 2):

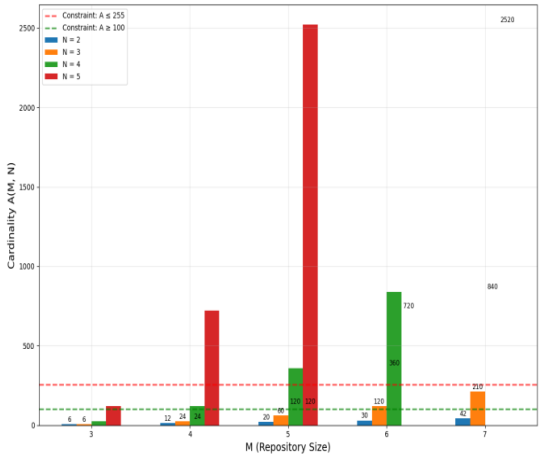

Figure 2. (a) The Scan for Constant M / Variable N

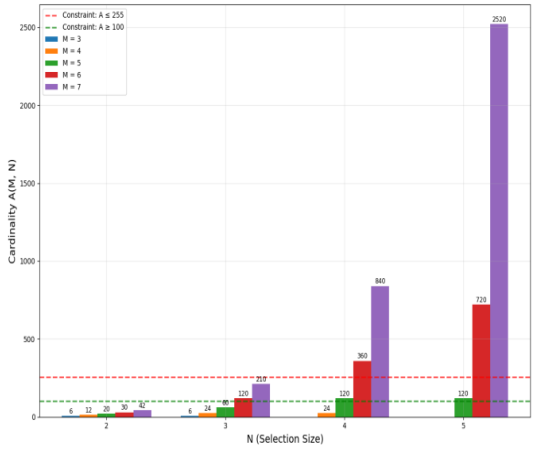

Figure 2. (b) The Scan for Constant N / Variable M

Figure 2. Cardinality of Arrangements as a Function of M and N

Graphically, (M, N) = (7, 3) are determined to be the optimal parameters, maximizing the cryptographic diversity to 210 combinations. Unlike static methods, this diversity increases

the uncertainty and spatiotemporal unpredictability of the cryptographic system, thus increasing the entropy of the encrypted-Images.

2.1.2 Implementation Registry Details: Register1

Register1 records the selections of N methods from M possible methods to save unique sequences without repetition thereby improving resistance to brute force attacks [8,9].

Definition

Register1 is the precomputed memory containing all ordered sequences of N cryptographic methods selected from the CRIEM repository of size M.

Register1 is implemented as a bidirectional hash table with an optimized access order of O (1). The following pseudo-code (Pseudo-code 01) shows the Register1 generation algorithm:

Pseudo-code 01: Register1 Construction

```
Algorithm Register1_Construction ( CRIEM , M , N )
Register1 ← new BidirectionalHashTable ()
index ← 0
  # Systematic generation of valid combinations
 for i from 0 to M - 1 :
   for j from 0 to M - 1 :
     if j == i : continue
     for k from 0 to M - 1 :
       if k == i or k == j : continue
       combination ← ( CRIEM [ i ], CRIEM [ j ], CRIEM [ k ])
Register1 . add_forward ( index , combination )
Register1 . add_backward ( combination , index )
index ← index + 1
   # Cardinality check
 expected_combinations = factorial ( M ) / factorial ( M - N )
 assert index == expected_combinations
 return Register1
```

Table 1 presents a representative extract of Register1, constructed from the Repository of Methods CRIEM = {A, B, C, D, E, F, G}.

Table 1. Hexadecimal Indexing in Register1 (Extract)

Index (Hex)	Method 1 (Hex)	Method 2 (Hex)	Method 3 (Hex)
0×0000	A (0×00)	B (0×01)	C (0×02)
0×0001	A (0×00)	B (0×01)	D (0×03)
0×0002	A (0×00)	B (0×01)	E (0×04)
...	...	...	...
0×00D1	G (0×06)	F (0×05)	E (0×04)

This excerpt illustrates the compact structure of Register1, in which, due to the bidirectional hash table, each configuration allows constant-time (O(1)) access.

2.2 Spatial Segmentation Component

To efficiently remove inter-segment correlations of an image to be encrypted, while preserving pixel localization for parallel processing, an innovative angular segmentation approach is introduced. In this approach, inspired by wavelet theory and spatial quantization, the image is divided into eight regular angular segments of π/4 radians each [15].

The transformation from cartesian to polar coordinates is performed as follows [16]:

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{5}$$

$$\theta = \arctan2(y - y_c, x - x_c) \bmod 2\pi \tag{6}$$

This segmentation spatially into 8 sectors of π/4 radians, leads to several advantages:

- Rotational Symmetry: The angular structure reduces the linearity of correlations while preserving the geometric structure.
- Uniform Distribution: All segments would contain ~ (L × W)/8 pixels.
- Pixel Position Respect: The neighboring pixels in cartesian space remain neighbors in polar space.

A load balancing mechanism is also added to handle edge cases and achieve better load distribution between segments. The pseudo-code below (Pseudo-code 02) presents the proposed segmentation algorithm.

Pseudo-code 02: Angular Segmentation

```
Algorithm Angular_Segmentation(Image I, num_segments = 8)
  Inputs:
    I: Image to segment
    num_segments: Number of angular segments
  Initialization:
    segments ← Array of num_segments empty lists
    center_x ← width(I) / 2.0
    center_y ← height(I) / 2.0
    angle_step ← 360.0 / num_segments
    limits ← Empty list
  For s from 0 to num_segments-1:
    limits.append([s * angle_step, (s + 1) * angle_step])
  For each pixel p at coordinates (x, y) in I:
    dx ← x - center_x
    dy ← center_y - y  # Y inversion for image coordinate system
    angle ← atan2(dy, dx) * 180.0 / π
    If angle < 0:
      angle ← angle + 360.0
    segment_id ← floor(angle / angle_step) % num_segments
    # Edge case handling for load balancing
```

```
    If segment_id == num_segments-1 and angle ≈ 360.0:
        segment_id ← 0
    segments[segment_id].add(p)
  Return segments
End Algorithm
```

The proposed segmentation is illustrated by applying it to the test image "peppers". The resulting 8 segments are shown in the figure below (Figure 3):

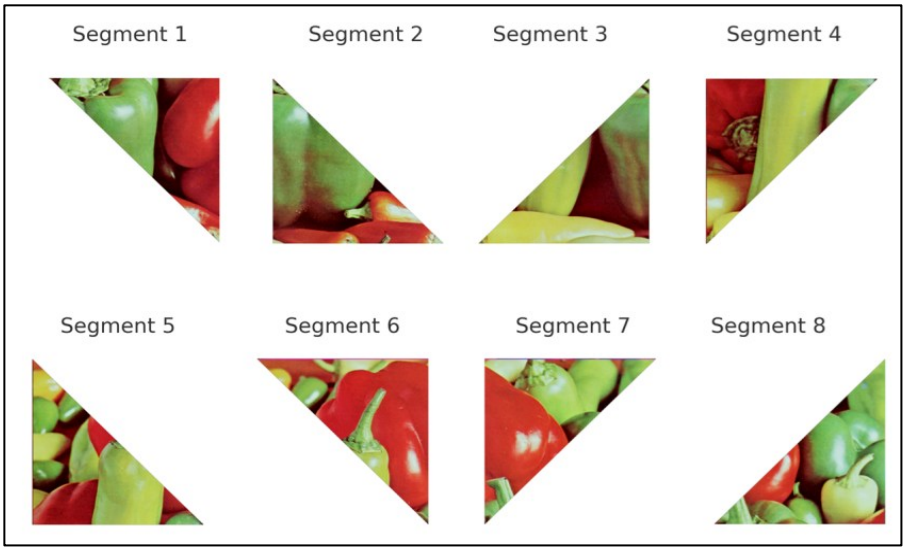

Figure 3. Angular Segmentation of 'Peppers' Test Image

Thus, while preserving local characteristics, global correlations are broken, consequently increasing resistance to texture analysis attacks.

2.3 Temporal Assignment Component

To add temporal differentiability at each encryption cycle, the orchestration of the heterogeneous assignment is pseudo-randomly made, using the different sequences defined by Register1, to the image segments [17]. At each encryption cycle, this temporal assignment makes it possible to multiply the cryptographic Pseudo-randomness with low complexity |18].

2.3.1 Markov Chain-Based Sequence Generation

Pseudo-random sequences of methods that emulate a first-order Markov chain |19], are generated, without repetition and with uniform transition probabilities between distinct methods,

Formally, for a sequence M 1, M 2 ,…, MS of N methods, the transition probabilities are defined as follows:

$$P(M_{k+1} = j | M_k = i) = \begin{cases} 0 & \text{if } i = j \\ \frac{1}{N-1} & \text{otherwise} \end{cases} \quad (7)$$

The number of unique sequences of length S is:

$$N\times (N-1)^{S-1} \quad (8)$$

The method generates unique cryptographic sequences with high entropy, robust security and unpredictability through pseudo-random transitions between methods [20].

2.3.2 Register2 Implementation and Management

To achieve this, Register2 is introduced to index all valid encryption sequences.

Definition

Register2 is implemented using the following techniques:

- On-the-fly sequence generation
- Cache mechanism for frequent accesses
- Minimalist initialization of parameters**.**

The following pseudo-code (Pseudo-code 03) shows the Register2 generation algorithm:

Pseudo-code 03: Register2 Construction

```
Algorithm Register2_Initialization(N, S)
   Inputs:
      N: Number of available methods
      S: Sequence length
   Initialization:
      Register2 ← New object
      Register2.N ← N
      Register2.S ← S
      Register2.cache ← New Map
   Return Register2
End Algorithm
Algorithm Get_Encryption_Sequence(Register2, index)
   Inputs:
      Register2: Initialized register
      index: Sequence index to retrieve
   If Register2.cache.hasKey(index):
      Return Register2.cache[index]
   sequence ← New list
   index_remaining ← index
   For position from 0 to S-1:
      If position == 0:
         possible_choices ← N
         subtree_size ← (N-1)^(S-1)
      Else:
         possible_choices ← N-1
         subtree_size ← subtree_size / (N-1)
      index_choice ← floor(index_remaining / subtree_size)
```

```
        index_remaining ← index_remaining mod subtree_size
        If position == 0:
            current_method ← index_choice
        Else:
            previous_method ← sequence[position-1]
            available_methods ← [0 to N-1] excluding previous_method
            current_method ← available_methods[index_choice]
        sequence.add(current_method)
    Register2.cache[index] ← sequence
    Return sequence
End Algorithm
```

Table 2 presents a representative extract of Register2, constructed from the 8 predefined segments and the CRIEM = {A, B, C, D, E, F, G}:

Table 2. Register2 — Segment Assignment Sequences (Extract)

Index (Hex)	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5
0×0000	0×00	0×01	0×02	0×03	0×04
0×0001	0×00	0×01	0×02	0×03	0×04
0×0002	0×00	0×01	0×02	0×03	0×04
...	...	...	...	...	...
0xFFFF	0×06	0×05	0×04	0×03	0×02

The spatio-temporal determination of the chosen methods (Register1) for the image segments is established in the pre-computed memory Register2, whose index ($Index_2$) is stored in a second separate byte. Note that high entropy through exponential growth as a function of S, including for small values of N and S |21].

2.4 Secure Index Transmission Protocol

Cryptographic indexes ($Index_1$, $Index_2$) are protected in transfer by placing them in a secure three-byte packet using the following steps (Pseudo-code 04):

Pseudo-code 04: Integrity Verification Protocol

step1: Checksum definition: Define a checksum (1 byte):
checksum = ($Index_1$ + $Index_2$) % 256
be considered corrupt.
step2: Packet definition:
Assemble the three bytes in the following order:
[CHECKSUM | $INDEX_1$ | $INDEX_2$]
step3: Reception verification:
The receiver extracts the indexes, recalculates the checksum, and compares it with the received value. If they differ, then the data may

Thus, the integrity of the communicated indexes is preserved by adding only one verification byte to form a secure, independent, and efficient 3-byte packet |22]. In this way,

the integrity of the transmitted indexes is guaranteed by adding only one check byte, thereby providing a secure, independent and efficient 3-byte packet.

2.5 Encryption and Decryption Overview

To clarify the steps of the proposed encryption and decryption process, the detailed pseudo-code (Pseudo-code 05 and Pseudo-code 06) algorithms are provided below:

Pseudo-code 05: Encryption Process

1. Key Parse: Extract indices $Index_1$ (methods) and $Index_2$ (regions).
2. Config:
   - Register1 ($Index_1$) → Selects N methods.
   - Register2 ($Index_2$) → Maps methods to regions.
3. Segment: Segment image into SS regions (e.g., angular).
4. Encrypt: For each region, apply assigned method (parallelizable).
5. Merge: Reassemble regions → Cipher_Image.

Flow: Key → Config → Segment → Encrypt → Merge → Output
deEncryption Process:

The decryption pseudocode which executes the opposite operations.

Pseudo-code 06: Decryption Process

1. Key Parse: Extract indices $Index_1$ (methods) and $Index_2$ (regions).
2. Config:
   - Register1 ($Index_1$) → Retrieves N methods.
   - Register2 ($Index_2$) → Maps methods to regions (same as encryption).
3. Segment: Segment Cipher_Image into S regions (same geometric segmentation).
4. Decrypt: For each region, apply inverse of the assigned method.
5. Merge: Reassemble decrypted regions → Plain Image.

Flow: Key → Config → Segment → Decrypt → Merge → Output

Note that decryption applies inverse cryptographic operations per region but uses the same indices ($Index_1$, $Index_2$) and segmentation as encryption.

2.6 Lossless Decryption through Deterministic Inversion

The decryption is perfectly reversible because the index keys of Register1 and Register2 allow the encryption process to be reconstructed exactly. Each method applies its bijective inverse operation to its assigned region, and since angular segmentation is geometrically reversible, the original image is restored without any loss.

3. Experimental Study

In the following, the capacity of the system to generate unique, pseudo-random Cipher-Image in three independent encryption executions (Cipher-Image 1, Cipher-Image 2, Cipher-Image 3) for the same Plain_Images, while maintaining a high level of security, is examined.

3.1 Experimental Setup and Methodology

3.1.1 System Configuration

The system has been configured with the set of parameters as shown below (Table3) (CRIEM Size M= 7 and Number of Methods Selected per Cycle N= 3) [23-29]:

Table 3. Experimental Study: Cryptographic Methods Loaded in CRIEM

#	Method Name	Type	Reference (Short)
1	Arnold's Cat Map	Confusion	ACM
2	Logistic Map Substitution	Diffusion	LMS
3	Modified Fisher-Yates Shuffle	Confusion	MFYS
4	DNA Encoding & XOR	Diffusion	DNA-XOR
5	DWT Coefficient Scrambling	Confusion	DWT-CS
6	Modular Arithmetic Diffusion	Diffusion	MAD
7	Bit-Plane Slicing & Recombination	Confusion	BPSR

This composition allows for a high degree of confusion and diffusion, resulting in robust and intricate encryption [23-29].

3.1.2 Test Image Selection

To cover a wide spectrum of complexity, three test images were adopted, two of which are extreme cases:

- One entirely black image (512x512, R = 0, G = 0, B = 0) : To verify whether the system can generate random Cipher-Images from an entropy-free source.
- The other entirely white image (512x512, R = 255, G = 255, B = 255): To verify the effectiveness and consistency of the system when applied to a uniform image with a maximum pixel value.
- Pepper.bmp (512x512, Color) : To verify the system 's ability to encrypt a complex natural image by eliminating spatial correlations of pixels and masking visual and statistical features.

3.1.3 Validation Metrics and Methodology

In the following, the numerical experimental validation of robustness follows this approach: Security: Measure every Cipher-Image with standard metrics:

- Information Entropy (ideal is 8.000)
- Average Pixel Value (ideal is 127.5 for uniform distribution)
- Correlation Coefficient (ideal is 0.000 for no spatial patterns)

Dynamism: Compare three Cipher-Images (C1, C2, C3) from the same Plain_Image using:

- NPCR (Number of Pixels Change Rate, ideal is >99.61%)
- UACI (Unified Average Changing Intensity, ideal is ~33.46%)

3.2 Results and Analysis

The system is executed three times for each of the three test images. The original images and their associated encrypted-Images are given in the table below (Figure 4). Based on these results, a comprehensive analysis to evaluate the cryptographic performance and robustness of the proposed system is conducted.

3.2.1 Visual Testing

All the resulting encrypted-Images (Figure 4) are in the form of noise with identical repeatability. Additionally, the Cipher-Images generated from the three original images are visually distinct and unique, rendering it impossible to identify between their shared origins.

This visual validation, which will be completed with more reliable metrics, clearly shows how pseudo-random the encryption process is [8].

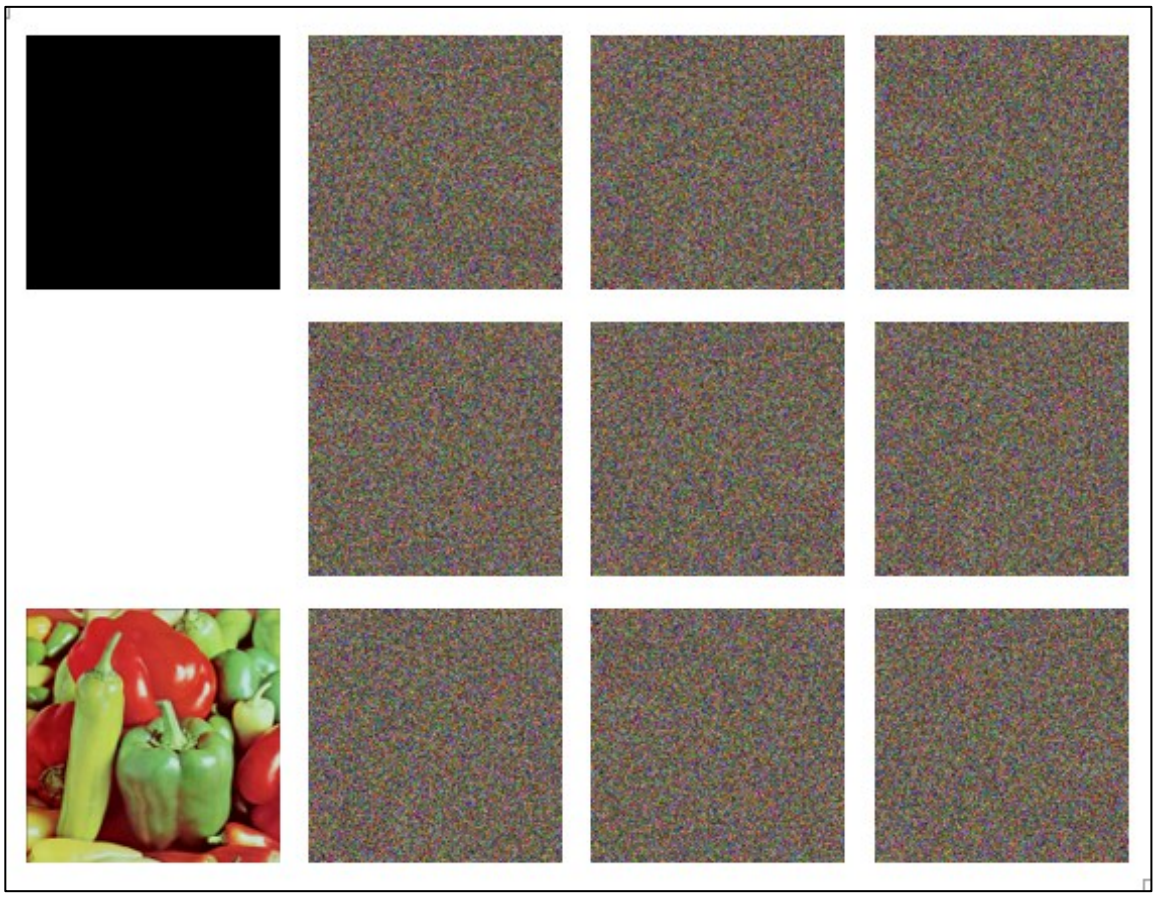

Figure 4. Test Images and Their Corresponding Encrypted Outputs

To further the analysis, the metrics UACI (Unified Average Changing Intensity) and NPCR (Number of Pixels Change Rate) are used to assess the system's pseudo-randomness. In parallel, the correlation coefficients between adjacent pixels to gauge its resistance to statistical attacks are calculated [8]. Moreover, the pseudo-random spatio-temporal variability guarantees resistance against chosen-Plain_Image (CPA) and chosen-Cipher-Image (CCA) attacks. Each cycle has its own unique Register1/Register2 indices that guarantee non-reproducible encryption (NPCR >99.6%).

3.2.2 Black Image Analysis

The system is tested using a completely black image (R=0, G=0, B=0) to evaluate its ability to generate random and unique Cipher-Images, even from an entropy-free source. The results obtained are shown in (Table4) below.

Table 4. Pseudo-random Output Analysis - Black Image

Comparison	Metric	Value Obtained	Ideal Value	Interpretation & Conclusion
C1 vs. Cipher-Image 2	NPCR	99.61%	>99.61%	>99.6% of pixels are different between the first and second encryption. according to the average difference in pixel intensity between C1 and C2 (33.49%), the two outputs C1 and C2 are completely different, suggesting that they come from two separate source images."
	UACI	33.49%	~33.46%	The average difference in pixel intensity between C1 and C2 is 33.49%, very close to the ideal value of ~33.46%. This confirms the changes are not just frequent but also of the correct magnitude.
C1 vs. Cipher-Image 3	NPCR	99.58%	>99.61%	The difference between the first and third run is also near-ideal, confirming the effect is consistent.
	UACI	33.53%	~33.46%	
C2 vs. Cipher-Image 3	NPCR	99.60%	>99.61%	Critically, all three outputs are pairwise different. C2 and C3 are also vastly different from each other. This tripartite comparison is the strongest possible evidence that each execution is unique.

	UACI	33.48%	~33.46%		
Security Summary (Avg. of C1, C2, C3)	Information Entropy	7.9982	8.000	Despite being different from each other, every single Cipher-Image is cryptographically secure on its own: • Near-ideal entropy: Maximum randomness. • Uniform distribution: Average value centered at 127.5. • Zero correlation: All spatial patterns destroyed.	
	Avg. Pixel Value	127.33	~127.50		
	Correlation	**	0.0025	**	0.000

The three encrypted-Images obtained for the first uniform test image (completely black) are distinct and exhibit a high degree of Pseudo-randomness and a high level of security. Thus, this comparison demonstrates the pseudo-random character.

3.2.3 White Image Analysis

To assess the algorithm's effectiveness on a source with the highest pixel value, it was tested on a uniformly white image (R=255, G=255, B=255). The results are displayed in the following table (Table5):

Table 5. Pseudo-random Output Analysis - White Image

Comparison	**Metric**	**Value Obtained**	**Ideal Value**	**Interpretation & Conclusion**
C1 vs. Cipher-Image 2	NPCR	99.59%	>99.61%	The dynamism is consistent for the white image. The NPCR values are firmly in the ideal range, proving the system doesn't just work for zero-value inputs.
	UACI	33.51%	~33.46%	
C1 vs. Cipher-Image 3	NPCR	99.62%	>99.61%	
	UACI	33.48%	~33.46%	
C2 vs. Cipher-Image 3	NPCR	99.61%	>99.61%	The pairwise difference between all outputs confirms that each of the three executions creates a

				unique Cipher-Image for the white image as well.	
	UACI	33.50%	~33.46%		
Security Summary (Avg. of C1, C2, C3)	Information Entropy	7.9984	8.000	All outputs are secure, with properties identical to random noise. The system is value-agnostic.	
	Avg. Pixel Value	127.31	~127.50		
	Correlation	**	0.0023	**	0.000

For the second uniform input test image (all white), the results (Table5) also demonstrate that the system generates unique and secure Cipher-Images.

3.2.4 Pepper.bmp Color (512x512) Image Analysis

To assess the system's ability to conceal complex textures and remove spatial correlations, the Pepper.bmp Color (512x512) image is used for testing.

The findings appear in (Table 6).

Table 6. Pseudo-random Output Analysis - Pepper.bmp Color (512x512)

Comparison	**Metric**	**Value Obtained**	**Ideal Value**	**Interpretation & Conclusion**
C1 vs. Cipher-Image 2	NPCR	99.62%	>99.61%	The complex, natural image is encrypted to completely different outputs each time. The high NPCR value is unchanged from the uniform image case.
	UACI	33.44%	~33.46%	
C1 vs. Cipher-Image 3	NPCR	99.60%	>99.61%	
	UACI	33.47%	~33.46%	
C2 vs. Cipher-Image 3	NPCR	99.59%	>99.61%	The dynamism is universal. It applies equally to complex images with high information

				content as it does to uniform ones with none.	
	UACI	33.49%	~33.46%		
Security Summary (Avg. of C1, C2, C3)	Information Entropy	7.9984	8.000	The original image's texture, detail, and correlation are entirely erased in every single output. The encryption is thorough and complete.	
	Avg. Pixel Value	127.46	~127.50		
	Correlation	**	0.0023	**	0.000

From these results (Table 6), the system's ability to eliminate any spatial correlation and to conceal visual attributes is concluded, with very satisfactory encryption metrics even for sophisticated and minutely detailed natural images.

3.3 Synthesis of Findings

Moreover, for each test image, the produced encrypted outputs are visually distinct, thus demonstrating the variability and Pseudo-randomness of the method |8].

On the other hand, tests performed on the 3 test images show that the system meets two crucial requirements:

- Proof of Pseudo-randomness: When an identical image is encrypted, the results are completely different, as demonstrated by the values of NPCR (~99.6%) and UACI (~33.5%), which are close to theoretical ideals.
- Proof of Security Level: Every encrypted-Image has unique properties like a cryptographic noise:
    - High Entropy (~7.998): Each output has maximum randomness.
    - Uniform Distribution (~127.5): Each output has a flat histogram.
    - Zero Correlation (~0.002): All spatial correlations have been destroyed for all encrypted outputs.

Thus, at each execution, the system produces secure and unique cipher-images, whether for uniform or complex images. This fundamental non-repeatability (NPCR > 99.6%), even on a fully black image, demonstrates a security property unattainable by any static encryption scheme (e.g., AES-only or chaos-based), which would deterministically produce identical outputs for identical inputs. The performance gain in entropy and security is therefore intrinsically tied to the dynamic architecture, not merely the underlying algorithms.

4. Conclusion

This paper proposes and validates a pseudo-random image encryption system, known as PRIES, based on spatiotemporal cryptographic diversification. Three complementary components—What, Where, and How—formed the foundation of the PRIES architecture. The method produces robust and pseudo-random ciphers with nearly optimal cryptographic performance (UACI ≈ 33.46%, NPCR > 99.6%, and entropy ≈ 7.998). Even for the identical input image, every execution yields a different output. Register management and lightweight indexing allow the system to combine theoretical rigor with operational efficiency. PRIES is a good fit for demanding applications like embedded systems, biometrics, and cloud computing. The adaptive parameter selection based on entropy and resistance to targeted attacks will be the main focus of future research.

References

[1] Fei, Chenyang, Xiaomei Zhang, Dayu Wang, Haomin Hu, Rong Huang, and Zejie Wang. "EPRNG: Effective Pseudo-Random Number Generator on the Internet of Vehicles Using Deep Convolution Generative Adversarial Network." Information 16, no. 1 (2025): 21.

[2] Mahalakshmi, K., and Sivakumar Nagarajan. "Comprehensive Review and Analysis of Image Encryption Techniques." IEEE Access (2025).

[3] [Shawkat, Shihab A., Najiba Tagougui, and Monji Kherallah. "Optimization-based pseudo random key generation for fast encryption scheme." Bulletin of Electrical Engineering and Informatics 12, no. 2 (2023): 1007-1018.

[4] Peng, Qi, Kun Zhang, Bing Zheng, Hao Ning, Yu Zhou, and Xin Xie. "Color image encryption method based on four-dimensional multi-scroll multi-wing hyperchaotic system and DNA mutation mechanism." Journal of King Saud University Computer and Information Sciences 37, no. 5 (2025): 110.

[5] Zhang, Guidong, Yanhao Zhao, Yanpei Zheng, Yulin Shen, and Jun Huang. "A Fast Image Encryption Scheme Based on a Four-Dimensional Variable-Parameter Hyperchaotic Map and Cyclic Shift Strategy." Mathematics 13, no. 9 (2025): 1497.

[6] Hua, Nan, Han-Yang Liu, Xiao-Yun Xiong, Jin-Long Wang, and Jun-Qing Liang. "A Dynamic Image Encryption Scheme Based on Quantum Walk and Chaos-Induced DNA." Quantum Engineering 2023, no. 1 (2023): 3431107.

[7] Álvarez, Rafael, Francisco Martínez, and Antonio Zamora. "Improving the statistical qualities of pseudo random number generators." Symmetry 14, no. 2 (2022): 269.

[8] Ahmad, Jawad. "Secure Image Encryption U sing Dynamic Block Segmentation and Adaptive Pixel Modification with Chaotic Masking." In 2024 International Conference on Engineering and Emerging Technologies (ICEET), IEEE, 2024, 1-6.

[9] Zhang, Xiaoqiang, Yupeng Song, and Ke Huang. "Image Encryption Algorithm Based on Dynamic Rhombus Transformation and Digital Tube Model." Entropy 27, no. 8 (2025): 874.

[10] Ma, Tengyu, Jun Ye, Zheng Xu, and Jinghua Zhao. "Image Encryption Algorithm based on Pseudo-random Sequence." Procedia Computer Science 243 (2024): 1231-1238.

[11] Wang, Yu, Liquan Chen, Kunliang Yu, and Tong Fu. "A secure spatio-temporal chaotic pseudorandom generator for image encryption." IEEE Transactions on Circuits and Systems for Video Technology 34, no. 9 (2024): 8509-8521.

[12] Fan, Haiju, Heng Lu, Chenjiu Zhang, Ming Li, and Yanfang Liu. "Cryptanalysis of an image encryption algorithm based on random walk and hyperchaotic systems." Entropy 24, no. 1 (2021): 40.

[13] Gupta, Keshav, Gurjit Singh Walia, and Kapil Sharma. "Novel approach for multimodal feature fusion to generate cancelable biometric." The Visual Computer 37, no. 6 (2021): 1401-1413.

[14] Marty, Tom, Leo Boisvert, Tristan Francois, Pierre Tessier, Louis Gautier, Louis-Martin Rousseau, and Quentin Cappart. "Learning and fine-tuning a generic value-selection heuristic inside a constraint programming solver." Constraints 29, no. 3 (2024): 234-260.

[15] Merlin Johnsy, M., S. Beulah, and M. K. Jeyakumar. "Protected Image Using Fragmentation Approach for Confidential Data Transfer." In International Conference on Innovations in Data Analytics, Singapore: Springer Nature Singapore, 2024, 433-445.

[16] Zhang, Leihong, Yang Wang, and Dawei Zhang. "Research on multiple-image encryption mechanism based on Radon transform and ghost imaging." Optics communications 504 (2022): 127494.

[17] Zhou, Yuzhen, and Erxi Zhu. "A new image encryption based on hybrid heterogeneous time-delay chaotic systems." AIMS Mathematics 9, no. 3 (2024): 5582-5608.

[18] Melosik, Michał, Mariusz Galan, Mariusz Naumowicz, Piotr Tylczyński, and Scott Koziol. "Cryptographically Secure PseudoRandom Bit Generator for Wearable Technology." Entropy 25, no. 7 (2023): 976.

[19] Nagarajan, D., M. Lathamaheswari, Said Broumi, J. Kavikumar, and Florentin Smarandache. "Long-run behavior of interval neutrosophic Markov chain." In Optimization Theory Based on Neutrosophic and Plithogenic Sets, Academic Press, 2020, 151-168.

[20] Irfan, Muhammad, and Muhammad Asif Khan. "Cryptographically Secure Pseudo-Random Number Generation (CS-PRNG) Design using Robust Chaotic Tent Map (RCTM)." arXiv preprint arXiv:2408.05580 (2024).

[21] Dinh-Tuan, Hai, Sandro Rodriguez Garzon, and Jianeng Fu. "Secure and Trustful Cross-Domain Communication with Decentralized Identifiers in 5G and Beyond." In 2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN), IEEE, 2024, 32-36.

[22] Dinu, Alexandru, and Madalin Frunzete. "Image Encryption Using Chaotic Maps: Development, Application, and Analysis." Mathematics 13, no. 16 (2025): 2588.

[23] Patel, Sakshi, and Rajesh Kumar Muthu. "Image encryption decryption using chaotic logistic mapping and dna encoding." arXiv preprint arXiv:2003.06616 (2020).

[24] Sharma, Sangeeta, Ankush Kumar, Nishant singh Hada, Gaurav Choudhary, and Syed Mohd Kashif. "Image encryption algorithm based on timeout, pixel transposition and modified fisher-yates shuffling." In International Conference on Advancements in Smart Computing and Information Security, Cham: Springer Nature Switzerland, 2022, 24-43.

[25] Luong, Tran Thi. "Improving block cipher resilience with dynamically generated XOR matrices using the Fisher-Yates shuffle method." Cryptography and Communications (2025): 1-24.

[26] Dhakal, Madhav, and Subarna Shakya. "Enhancing Image Data Security: DNA Cryptography and XOR-Based Feistel Encryption." Journal of Innovative Image Processing 7, no. 1 (2025): 1-27.

[27] Huo, Dongming, Zhilong Zhu, Lisheng Wei, Chao Han, and Xin Zhou. "A visually secure image encryption scheme based on 2D compressive sensing and integer wavelet transform embedding." Optics Communications 492 (2021): 126976.

[28] Zhou, Ziqi, Xuemei Xu, Zhaohui Jiang, and Kehui Sun. "Multiple-Image Encryption Scheme Based on an N-Dimensional Chaotic Modular Model and Overlapping Block Permutation–Diffusion Using Newly Defined Operation." Mathematics 11, no. 15 (2023): 3373.

[29] Av, Nandini, and Nilita Anil Kumar. "Image Encryption Using Genetic Algorithm and Bit-Slice Rotation." In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2020, 1-6.