**3D Chaos‑Enhanced Block‑Permutation & Diffusion**

Project submitted for the partial fulfillment of the requirements for the course

**CSE 459: Cryptography and Network Security**

Offered by the

**Department Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

1. P Vishnu Vardhan, AP22110010094

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[April,2025]**

**Contents:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Contents** | **Page number** |
| **1** | **Introduction** | **3** |
| **2** | **Background** | **3** |
| **3** | **Proposed Approach** | **4** |
| **4** | **Results & Discussion** | **6** |
| **5** | **Conclusion** | **6** |

1. **Introduction**

Secure encryption of digital images ensures confidentiality and integrity when images are stored or transmitted over untrusted channels. Conventional block‐cipher methods deliver high security but impose heavy computational demands, which can hinder performance in systems with limited processing power. Chaos‐based schemes exploit the unpredictable behavior of nonlinear maps to generate pseudo‑random sequences for pixel‑level operations, achieving lightweight encryption. However, one‑ and two‑dimensional chaotic maps often suffer from restricted key‑space and inadequate diffusion, leaving encrypted images vulnerable to statistical and differential attacks.

A three‑dimensional logistic‐map approach addresses these limitations by producing three independent chaotic sequences with enhanced complexity. The first sequence drives row‑level rotations, redistributing pixels across horizontal lines; the second sequence governs column‑level rotations, shuffling pixels along vertical axes; the third sequence supports intra‑block permutations within fixed‑size blocks, creating fine‑grained confusion at the local level. A final XOR diffusion layer applies the same third sequence to pixel intensities, erasing residual statistical patterns.

This multi‑stage design intertwines global and local permutations with diffusion, maximizing both confusion and diffusion as defined by Shannon’s principles of cryptographic security. The method remains computationally efficient, requiring only simple arithmetic and array‐shifting operations, and is tailored for 256×256 grayscale images. Results demonstrate near‑ideal entropy and high sensitivity metrics, confirming robustness against standard cryptanalytic attacks.

1. **Background** 
   1. **Standard Block Ciphers (AES, DES)**

 Encrypt fixed‑size blocks (e.g. 128 bits for AES) ss rounds of substitution and permutation.

 High CPU and memory requirements, which can slow down devices with limited resources.

**2.2 Row/Column Chaos Permutation**

* Generate two separate pseudo‑random sequences from a chaotic map.
* Use the first sequence to circularly shift each entire row (even shifts right, odd shifts left).
* Use the second sequence to shift each entire column (even shifts up, odd shifts down).
* Quickly breaks large‑scale image patterns with simple array rotations.
* Leaves relative order of pixels within each row or column intact, so local details may still be visible.

**2.3 Chaos‑Based Pixel Permutation**

* Produce a single pseudo‑random sequence from a chaotic map, scaled to cover every pixel position.
* Flatten the image into a one‑dimensional list of pixels and reorder them according to the chaotic indices.
* Achieves full‐image confusion in one pass, scattering all pixels.

1. **Proposed Approach**

A four‑stage pipeline combines global and local permutations with diffusion:

1. **Key Generation**
   * Three sequences generated by iterating the 3D logistic map

xₙ₊₁ = c·xₙ(1–xₙ) + b·yₙ²·xₙ + a·zₙ³

yₙ₊₁ = c·yₙ(1–yₙ) + b·zₙ²·yₙ + a·xₙ³

zₙ₊₁ = c·zₙ(1–zₙ) + b·xₙ²·zₙ + a·yₙ³

After discarding transients, extract

* **x\_key**: 256 entries mod 256
* **y\_key**: 256 entries mod 256
* **z\_key**: 65 536 entries mod 256

 **Row Rotation**

* Each row *i* is circularly shifted by x\_key[i] positions: even→right, odd→left.

 **Column Rotation**

* Each column *j* is shifted by y\_key[j]: even→up, odd→down.

 **Intra‑Block Permutation & Diffusion**

* Partition image into 16×16 blocks (256 total).
* For block *b*, take z\_key[b·256 … b·256+255], compute order = argsort(chunk), and permute its 256 pixels.
* Reshape z\_key to 256×256 and apply XOR with the permuted image.

**System Design**

* A person using a camera

  AI-generated content may be incorrect.**Input**:

256×256 grayscale image

* **Key Store**: Seeds and parameters for 3D chaotic map
* **Processing Units**:  
  – Row rotator  
  – Column rotator  
  – Block permuter  
  – XOR diffuser
* **Output**:

A close-up of a television screen

AI-generated content may be incorrect.

Ciphertext image

1. **Results & Discussion:**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Image 1** | **Image 2** |
| Entropy (cipher) | 7.9975 | 7.9970 |
| NPCR (orig vs cipher) | 99.5972 % | 99.6002 % |
| UACI (orig vs cipher) | 31.7662 % | 28.7791 % |

 **Entropy** near 8 confirms high randomness.

 **NPCR** above 99.5 % shows strong sensitivity to input changes.

 **UACI** around 30 % indicates excellent diffusion of pixel intensities.

 **Encryption time**: 0.0194 seconds

 **Decryption time:** 0.0171 seconds

1. **Conclusion**

The 3D Chaos‑Enhanced Block Permutation & Diffusion method achieves a balance of strong security metrics and computational efficiency. Layered global (row/column) and local (block) permutations, followed by XOR diffusion, maximize confusion and diffusion. Future work will explore adaptive block sizes and hardware acceleration to support higher resolutions and real‑time streaming.