Image Encryption using Logistic Function

A PROJECT REPORT

Submitted in partial fulfilment of the
Requirements for the award of the degree
of
Bachelor of Technology in Computer Science and Engineering

By

KASTURI CHATTERJEE
(Univ. Roll Number – 10700121042 & College Roll Number – CSE/21/071)

ABIR DATTA
(Univ. Roll Number – 10700121121 & College Roll Number – CSE/21/134)

NISHA MASANTA
(Univ. Roll Number – 10701621020 & College Roll Number – CSE/21/T-148)

Under the supervision of
Dr. Chinmay Maiti

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING & MANAGEMENT, KOLAGHAT

(AFFILIATED TO MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL)

June, 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING & MANAGEMENT, KOLAGHAT

(Affiliated to MAKAUT, WB)

Purba Medinipur – 721171, West Bengal, India

CANDIDATE'S DECLARATION

I/We hereby declare that the work presented in the final year project entitled: **"Image Encryption using Logistic Function"** submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, to the **Department of Computer Science and Engineering**, **College of Engineering & Management, Kolaghat**, is an authentic record of my/our own work carried out during the final year under the guidance of: **Dr. Chinmay Maiti , Assistant Professor of Department of Computer Science and Engineering.**

I/We further declare that the contents of this report have not been submitted to any other university or institution for the award of any degree, diploma, fellowship, or any other similar title.

Date: June 13, 2025

Kasturi Chatterjee
University Roll: 10700121042
University Registration No.: 211070100110047

Abir Datta
University Roll: 10700121121
University Registration No.: 211070100110112

Nisha Masanta
University Roll: 10701621020
University Registration No.: 211070100110135

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING & MANAGEMENT, KOLAGHAT
(Affiliated to MAKAUT, WB)
Purba Medinipur – 721171, West Bengal, India

CERTIFICATE OF APPROVAL

This is to certify that the work embodied in this project entitled: **"Image Encryption using Logistic Function"** submitted by Kasturi Chatterjee, Abir Datta and Nisha Masanta to the Department of Computer Science and Engineering, has been carried out under my supervision and guidance.

The project work has been prepared in accordance with the regulations of Maulana Abul Kalam Azad University of Technology (MAKAUT). I hereby recommend that this project report be accepted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Signed by HOD (CSE)	Supervisor
Dr. Alok Ranjan Pal	**Dr. Chinmay Maiti**
Head, Dept. of CSE	(Assistant/Associate) Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING & MANAGEMENT, KOLAGHAT

(Affiliated to MAKAUT, WB)

Purba Medinipur – 721171, West Bengal, India

CERTIFICATE BY THE BOARD OF EXAMINERS

This is to certify that the project work entitled: "**Image Encryption using Logistic Function**" submitted by Kasturi Chatterjee, Abir Datta, Nisha Masanta to the Department of Computer Science and Engineering, College of Engineering & Management, Kolaghat, has been examined and evaluated by the undersigned.

The project work has been prepared in accordance with the regulations of Maulana Abul Kalam Azad University of Technology (MAKAUT) and is hereby approved as a partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Project Coordinator	**Board of Examiners**

Acknowledgement

It gives us immense pleasure to express our deep and sincere gratitude to our project guide, Dr. Chinmay Maiti, for his invaluable guidance, constant encouragement, and continuous support throughout the course of this project. His insightful suggestions, constructive feedback, and meticulous supervision have greatly shaped the direction and quality of our work.

We are sincerely thankful to the Department of Computer Science and Engineering, College of Engineering & Management, Kolaghat, for providing us with the necessary facilities and a supportive academic environment that enabled us to carry out our project successfully.

We also extend our heartfelt thanks to all the faculty members, laboratory staff, library staff, and administrative personnel of the institute for their kind assistance and cooperation during the project period.

Lastly, we would like to express our warm appreciation to our fellow classmates, whose encouragement and helpful suggestions provided additional motivation and direction throughout this endeavour.

Date: June 13, 2025

Kasturi Chatterjee
University Roll: 10700121042
University Registration No.: 211070100110047

Abir Datta
University Roll: 10700121121
University Registration No.: 211070100110112

Nisha Masanta
University Roll: 10701621020
University Registration No.: 211070100110135

Abstract

This project, titled “Image Encryption Using Logistic Function”, presents a comprehensive study and implementation of a chaos-based image encryption system, developed to address the challenge of securing digital images against unauthorized access and statistical attacks.

The primary goal of the project is to design and develop a system that utilizes the mathematical properties of the logistic map to generate highly random and sensitive encryption keys for secure image transformation. To achieve this, we employed chaotic sequence generation through logistic functions, bitwise XOR operations, MATLAB for implementation, and standard evaluation metrics such as entropy, NPCR, UACI, and correlation coefficients. The project not only meets the functional requirements of secure encryption and decryption but also offers a lightweight and efficient cryptographic framework that can be extended to color images and real-time systems.

This work has potential applications in military imaging, medical data protection, secure transmission of confidential documents, and multimedia communication systems, and serves as a foundation for future research in lightweight, chaos-based cryptographic algorithms.

In addition to its simplicity and computational efficiency, the proposed method is lightweight and scalable, making it suitable for deployment in real-time and resource-constrained environments. This approach lays the foundation for future extensions, including color image encryption, integration with other cryptographic frameworks, and use in secure multimedia applications such as medical imaging, surveillance systems, and confidential communications.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Description
• AES	Advanced Encryption Standard
• DES	Data Encryption Standard
• XOR	Exclusive OR
• NPCR	Number of Pixel Change Rate
• UACI	Unified Average Changing Intensity

Chapter 1

An Introduction Of Image Encryption Using Logistic Function

1.1 Introduction

In today's digital age, the security of multimedia data, particularly images, has become a critical concern due to the rise in cyberattacks and unauthorized access. Traditional encryption algorithms like Advanced Encryption Standard (AES) and the Data Encryption Standard (DES), while effective for text, often fall short in efficiently handling the high redundancy and large data size of image files. As a result, chaos-based encryption techniques have gained significant attention for image security, offering enhanced sensitivity, randomness, and robustness.

This project, titled **"Image Encryption using Logistic Function"**, leverages the principles of chaos theory to secure grayscale images. The logistic map, a well-known chaotic function, is used to generate pseudo-random sequences based on a control parameter and an initial seed. These sequences are highly sensitive to initial conditions and exhibit random-like behavior, making them suitable for cryptographic applications.

The encryption process involves three main steps: generating chaotic sequences, transforming the original image pixels based on these sequences, and producing an encrypted image that is unintelligible without the correct key parameters. The logistic map’s non-linear dynamics ensure high security and resistance against statistical, brute-force, and differential attacks.

This project demonstrates the effectiveness of chaotic systems in multimedia encryption and provides a lightweight yet secure alternative to conventional methods, particularly suitable for real-time and low-resource environments.

Through this project, we aim to demonstrate how chaotic functions like the logistic map can be effectively utilized for image encryption. The implementation showcases a practical application of theoretical chaos principles and provides insights into how such systems can offer robust protection in digital communication. In a world increasingly reliant on digital media, securing visual data through innovative cryptographic methods like chaos-based encryption is not just relevant—it is essential.

1.2 Objective Of The Project

The primary objective of this project is to develop a secure and efficient image encryption system using the Logistic Map, a well-known chaotic function, to ensure the confidentiality and integrity of grayscale images during storage and transmission in order to protect grayscale images against unauthorized access, interception, and tampering.

The specific goals of this project are as follows:

i) **To implement a chaos-based image encryption algorithm** using the logistic function, exploiting its non-linear and highly sensitive behavior for generating pseudo-random sequences.

ii) **To perform multi-layer encryption** by applying multiple rounds of bitwise XOR operations between the image and chaotic sequences, thereby enhancing the security and complexity of the encrypted image.

iii) **To evaluate the effectiveness of the encryption method** through statistical analysis and security metrics such as:

   - Correlation coefficient analysis
   - Histogram uniformity
   - Shannon entropy
   - NPCR (Number of Pixel Change Rate)
   - UACI (Unified Average Changing Intensity)

iv) **To ensure that the encryption process is reversible** and allows accurate decryption of the original image using the correct key parameters, verifying the algorithm's reliability and integrity.

v) **To provide a lightweight and computationally efficient solution** that can be used in real-time and resource-constrained environments, such as IoT devices, mobile platforms, or embedded systems.

By achieving these objectives, the project demonstrates the potential of using chaotic systems like the logistic map as a robust and secure method for multimedia encryption, offering an alternative to conventional cryptographic techniques for image data protection.

1.3 Methodologies To Achieve The Objective Of The Project

To ensure secure and efficient image encryption and decryption using chaotic systems, the following methodologies were employed:

i) Preprocessing and Input Handling
ii) Chaotic Key Generation using Logistic Map
iii) Multi-Stage XOR-Based Encryption
iv) Decryption Using Reversible XOR Operations
v) Security Analysis and Validation
vi) Visualization and Comparative Study

These methodologies collectively ensure that the system achieves its core objective: a secure, efficient, and reversible encryption technique for grayscale images using chaos theory.

Input

A grayscale image (e.g., .png, .jpg) and logistic map parameters (a, x_0, y_0, z_0) for chaotic key generation.

Output

An encrypted image with high randomness and a decrypted image identical to the original. Graphs such as histograms and correlation plots support analysis.

Environment

The grayscale input image is processed in MATLAB. Chaotic sequences are generated using the logistic map with distinct initial values. These sequences are used as keys for a three-stage XOR-based encryption. The encryption process involves successive XOR operations with each chaotic key to produce the final encrypted image. Decryption is done by applying the XOR operations in reverse order using the same chaotic keys. MATLAB's image processing toolbox and plotting functions were used for encryption, visualization, and analysis.

The project successfully demonstrates a secure image encryption technique using chaos theory. MATLAB provided an efficient environment for implementation and validation. The algorithm ensures high confidentiality and can be extended to color images and real-time communication systems.

1.4 Scope Of The Project

This project focuses on developing a lightweight, secure, and reversible image encryption technique based on the chaotic behavior of the logistic map. The system is designed specifically for grayscale images and can be extended to color images in future work. The encryption method enhances security by utilizing multiple chaotic sequences and performing multi-layered bitwise operations. The scope includes implementation, encryption, decryption, and evaluation through statistical metrics such as correlation, entropy, NPCR, UACI, histogram analysis, and scatter plots. The project also explores its potential application in fields such as secure image transmission, biometric data protection, confidential document storage, and IoT image security. While the current scope is limited to still grayscale images, the method can be scaled to more complex image types and real-time systems.

1.5 Background

In today’s digital era, the transmission and storage of images have become commonplace across various platforms such as social media, cloud services, healthcare systems, and defense networks. However, this increased usage also raises serious concerns about data privacy and unauthorized access. Conventional cryptographic methods, while effective for textual data, often struggle to efficiently handle the unique characteristics of image data, such as large size, high correlation between pixels, and redundancy.

To address these limitations, researchers have explored chaotic systems for secure image encryption. Chaos-based algorithms offer significant advantages due to their sensitivity to initial conditions, unpredictability, and pseudo-random behavior. Among these, the logistic map is one of the simplest and most widely used chaotic functions, capable of generating highly unpredictable sequences. This project leverages the logistic function to develop a multi-layered encryption scheme for grayscale images. The technique uses chaotic sequences to perform bitwise operations that transform the image into an unintelligible format, which can only be decrypted using the correct chaotic parameters. This background establishes the need and foundation for applying chaos theory to image encryption in a fast, secure, and resource-efficient manner.

1.6 Literature Review

In the digital era, securing multimedia data—particularly images—has become increasingly critical due to the widespread transmission of visual content across unsecured networks. Unlike text data, images carry more information and exhibit high inter-pixel redundancy, making traditional encryption algorithms such as AES (Advanced Encryption Standard), DES (Data Encryption Standard), and RSA computationally inefficient and less suitable for image protection. These limitations have led researchers to explore alternate encryption paradigms, notably those inspired by chaos theory.

Chaos-based cryptography leverages the inherent properties of chaotic systems—sensitivity to initial conditions, pseudo-randomness, and ergodicity—to produce robust and lightweight encryption schemes. Among the various chaotic maps, the Logistic Map is particularly favored due to its simplicity and capability to generate highly unpredictable sequences.

Building upon these prior works, this project employs the logistic map to generate three different chaotic sequences with distinct initial parameters. These sequences are used in XOR-based encryption to enhance the unpredictability and diffusion of pixel values. After encryption, security metrics such as Shannon entropy, NPCR (Number of Pixel Change Rate), UACI (Unified Average Changing Intensity), and correlation coefficient analysis are used to evaluate the effectiveness of the scheme.

While existing literature confirms the strength of logistic-map-based encryption, many methods focus either on complexity or on high-dimensional chaotic systems. In contrast, this project emphasizes a simple yet secure method, making it suitable for practical use in real-time and resource-constrained environments.

The approach ensures a high level of security while maintaining simplicity and performance, making it suitable for practical applications in fields such as secure image transmission, cloud storage, and biometric data protection.

1.7 Applications Of The Project

The implementation of image encryption using a logistic map has broad applicability across various domains where image security is critical. Due to its efficiency, simplicity, and strong encryption capabilities, the proposed method is particularly suitable for real-time and resource-constrained environments. The key applications include:

i) Secure Image Transmission
ii) Medical Imaging Systems
iii) Defense and Surveillance
iv) Cloud Storage Security
v) Biometric Data Protection
vi) Multimedia Content Distribution
vii) Digital Watermarking & Copyright Protection
viii) Smart Devices and IoT
ix) E - Governance and Legal Systems
x) Educational Platforms

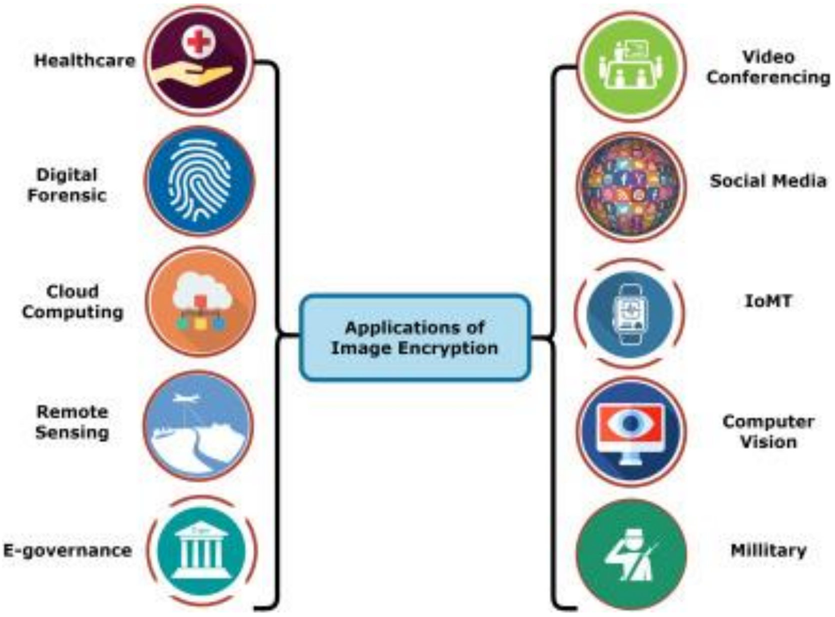


Figure 1.1: Applications of Image Encryption

1.8 Challenges in Image Encryption

Image encryption faces several technical and practical challenges that arise from the unique nature of image data:

i) **High Redundancy and Pixel Correlation**
Images have strong similarity between neighboring pixels, making it difficult to conceal patterns without effective encryption.

ii) **Large Data Size**
High-resolution images require more processing time and memory, demanding efficient encryption algorithms.

iii) **Real-Time Processing Requirements**
Applications such as surveillance or video streaming need fast encryption/decryption without compromising security.

iv) **Key Sensitivity and Management**
Even small errors in the encryption key can lead to failed decryption. Proper key generation, storage, and exchange are critical.

v) **Format and Compression Compatibility**
Encrypted images must remain compatible with standard formats and survive compression without data corruption.

vi) **Resistance to Cryptographic Attacks**
The encryption scheme must withstand various attacks like known-plaintext, chosen-plaintext, differential, and brute-force.

vii) **Lossless Decryption Requirement**
The decrypted image must perfectly match the original, as even minor errors can impact interpretation, especially in medical or legal fields.

Additionally, encrypted images must remain compatible with compression formats (like JPEG/PNG) without data loss. Ensuring resistance against attacks such as known-plaintext and differential analysis, while maintaining 100% reversibility and visual quality after decryption, remains a significant challenge for robust image encryption systems.

Chapter 2

<u>Cryptography</u>

2.1 Overview

Cryptography is the practice of securing information by converting it into a secret code so that only authorized people can read it. It protects data during storage or transmission by turning readable text (called plaintext) into unreadable form (called ciphertext) using a process called encryption. The original message can be recovered through decryption. Cryptography is mainly of two types: symmetric, where the same key is used for both encryption and decryption, and asymmetric, where two different keys are used. It plays a vital role in keeping digital communication, personal data, and online transactions safe from unauthorized access.

2.2 Key Goals of Cryptography

The main goals of cryptography are to ensure the security and privacy of data. These include:

i) **Confidentiality:** Ensuring that information is accessible only to authorized individuals.
ii) **Integrity:** Protecting data from being altered or tampered with during storage or transmission.
iii) **Authentication:** Verifying the identity of parties involved in communication.
iv) **Non-repudiation:** Preventing denial of an action or communication by the sender.

Together, these goals help maintain trust, accuracy, and security in digital communication and data handling.

2.3 Working of Cryptography

Cryptography plays a fundamental role in securing digital communication by transforming data into a secure format that can only be interpreted by authorized parties. The working mechanism of cryptography involves a sequence of well-defined processes including encryption, transmission, and decryption, often reinforced by key management and data integrity verification.

Steps are illustrated below -

i) **Data Preparation**: The sender starts with original data known as **plaintext**.

ii) **Key Generation**: A cryptographic key or key pair is generated based on the encryption method used (symmetric or asymmetric).

iii) **Encryption**: The plaintext is converted into **ciphertext** using an encryption algorithm and the key and this ensures the data is unreadable without the correct key.

iv) **Transmission or Storage**: The ciphertext is transmitted over a network or stored securely.

v) **Decryption**: At the receiver’s end, the ciphertext is transformed back into the original plaintext using a decryption algorithm and the appropriate key.

vi) **Security Functions Achieved**: Cryptography provides several critical security functions essential for protecting information in digital systems:

- **Confidentiality** – Keeps data private.
- **Integrity** – Ensures data is not modified.
- **Authentication** – Confirms identity of sender/receiver.
- **Non-repudiation** – Prevents denial of sending/receiving data.

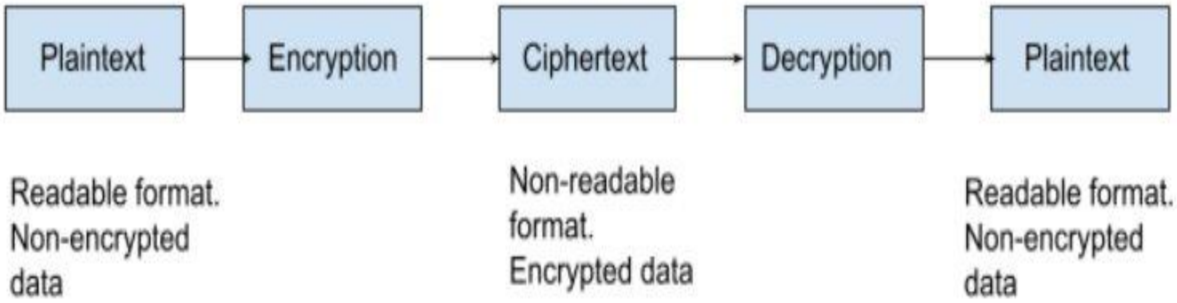

Plaintext
Encryption
Ciphertext
Decryption
Plaintext
Readable format. Non-encrypted data
Non-readable format. Encrypted data
Readable format. Non-encrypted data

Figure 2.1: Workflow of Cryptography

2.4 Types of Cryptography

Cryptography can be broadly classified into three main types.

i) **Symmetric Key Cryptography**

Also known as **secret key cryptography**, this method uses a single key for both encryption and decryption. It is computationally efficient and suitable for encrypting large volumes of data. However, the challenge lies in securely sharing the key between communicating parties.

ii) **Asymmetric Key Cryptography**

Also referred to as **public key cryptography**, this technique uses a pair of keys—one public and one private. The public key is used for encryption, while the private key is used for decryption. This method enhances security, especially in digital communication and secure transactions.

iii) **Hash Functions**

Hashing is a one-way cryptographic technique that transforms input data into a fixed-size string (hash value), which is unique to the original data. It does not use any keys and is primarily used for verifying data integrity and storing passwords securely.

Each type of cryptography serves distinct purposes and is selected based on the specific requirements of security, performance, and communication.

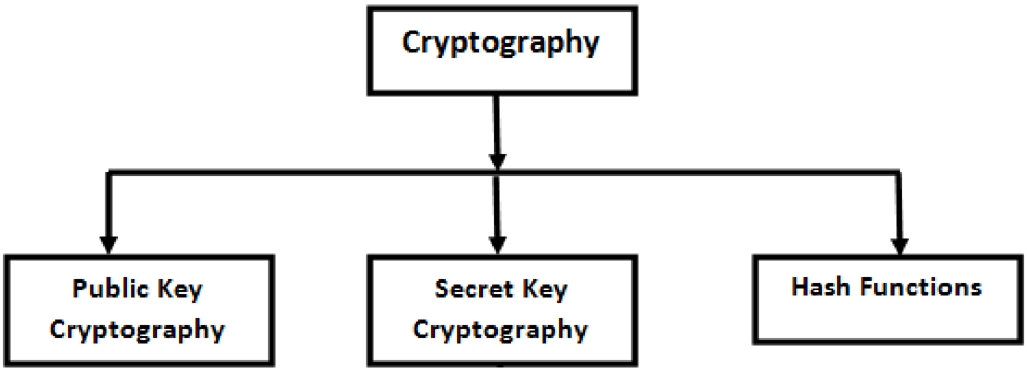


Figure 2.2 : Types of Cryptography

2.5 Technologies of Cryptography

Cryptography relies on a range of technologies that are fundamental to achieving secure communication, data integrity, and user authentication in digital systems. These technologies incorporate both classical and modern cryptographic algorithms and are widely applied across various domains such as network security, banking, cloud storage, and digital identity verification.

i) **Advanced Encryption Standard (AES)**: A widely adopted symmetric key encryption standard, known for its speed, efficiency, and strong security across various applications.

ii) **Data Encryption Standard (DES)**: An older symmetric encryption method though now considered insecure, it laid the foundation for modern cryptography.

iii) **Rivest-Shamir-Adleman (RSA)**: A popular asymmetric encryption algorithm based on the difficulty of factoring large prime numbers; used in secure data transmission.

iv) **Elliptic Curve Cryptography (ECC)**: An efficient asymmetric encryption technique provides strong security with smaller key sizes, suitable for mobile and IoT devices.

v) **Secure Hash Algorithms (SHA-1, SHA-256, etc.)**: One-way hash functions used to ensure data integrity widely used in password protection, blockchain, and digital signatures.

vi) **Digital Signatures**: Used to verify the authenticity and integrity of digital messages or documents and ensures that the sender cannot deny the origin (non-repudiation).

vii) **Public Key Infrastructure (PKI)**: A framework for managing public keys and digital certificates which supports secure data exchange and user authentication.

viii) **SSL/TLS Protocols**: Protocols used to secure data transmitted over the internet and ensure encrypted communication between clients and servers (e.g., HTTPS).

In summary, cryptographic technologies provide a robust foundation for implementing confidentiality, integrity, authentication, and non-repudiation in modern digital systems, thereby ensuring trust and security in the digital age.

2.6 Image Encryption is a Special Type of Cryptography

Image encryption is a specialized branch of cryptography that focuses on securing visual data by transforming images into an unintelligible format to prevent unauthorized access or tampering. Unlike text encryption, image encryption must address the large size, redundancy, and high correlation between neighboring pixels in digital images. As such, conventional cryptographic methods may not be sufficient or efficient.

To overcome these challenges, image encryption techniques often incorporate chaotic systems, such as the logistic map, to generate pseudo-random sequences that are sensitive to initial conditions. These sequences are then used to scramble pixel values and positions through mathematical transformations like XOR operations or permutation-substitution processes. The primary goal is to ensure confidentiality, integrity, and authenticity of image data, especially during storage or transmission over insecure networks.

Chapter 3

Image Encryption

3.1 Overview

Image encryption is a technique used to secure digital images by converting them into an unintelligible format that prevents unauthorized access or interpretation. It plays a critical role in protecting sensitive visual information during transmission or storage. Unlike text data, images have high redundancy and strong pixel correlation, making them more complex to encrypt effectively.

To address these challenges, advanced encryption methods—often based on chaotic maps**,** bitwise operations, and pixel permutation—are used. These techniques help break pixel patterns and enhance security by introducing randomness and unpredictability.

Image encryption is widely used in areas such as medical imaging**,** biometric authentication**,** military surveillance**,** and cloud-based media services and other where image privacy and integrity are paramount.

3.2 Detailing about Image Encryption

Image encryption is the process of converting digital images into a secure, unreadable format to protect them from unauthorized access. Due to the large size and pixel correlation in images, specialized techniques—such as chaotic maps and bitwise operations—are used. It is essential in fields like medical imaging, military security, and cloud storage to ensure confidentiality and integrity.

3.2.1 Definition

Image encryption is a process that transforms a plain image into an encrypted image, making it unreadable without the correct decryption key. This process is crucial for protecting digital images from unauthorized access and security threats, such as eavesdropping or illegal modification during transmission over networks.

3.2.2 Types of Image Encryption Techniques

i) Symmetric Image Encryption Techniques

In symmetric encryption, the same key is used for both encryption and decryption.

- Advanced Encryption Standard (AES):
  A widely used block cipher that can encrypt image data efficiently with high security.

- Data Encryption Standard (DES) and Triple DES (3DES):
  Earlier symmetric algorithms used for encrypting image data with fixed-size keys.

- RC4 and RC6:
  Stream and block ciphers used for fast image encryption, though RC4 is now considered less secure.

- Chaotic Map-Based XOR Encryption:
  Uses a single chaotic key to perform bitwise XOR operations on image pixels.

ii) Asymmetric Image Encryption Techniques

In asymmetric encryption, two keys are used: a public key for encryption and a private key for decryption.

- RSA (Rivest-Shamir-Adleman):
  Commonly used for secure key exchange or encrypting small images or image hashes due to slower performance on large data.

- Elliptic Curve Cryptography (ECC):
  Provides strong encryption with shorter keys, suitable for securing image metadata or small images.

- Diffie-Hellman Key Exchange:
  Often used to securely establish keys for symmetric encryption of images.

These encryption techniques can be used individually or in combination (hybrid encryption) to provide both performance and security in image protection.

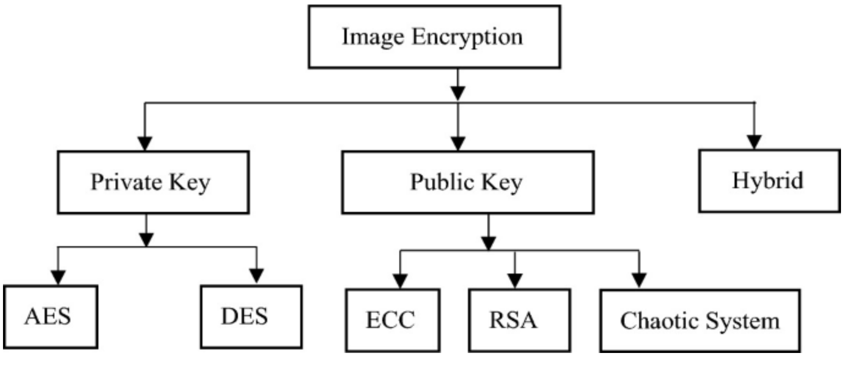


Figure 3.1: Types of Image Encryption

3.2.3 Key Steps of Image Encryption

i) Input Image Acquisition

Load the original image (grayscale or color) and convert to a suitable format if needed (e.g., grayscale for simplicity).

ii) Key Generation

Generate encryption keys using Chaotic maps (e.g., Logistic map) using parameters and initial values, these keys must be kept secret.

iii) Preprocessing (Optional)

Resize or normalize the image and flatten the image matrix to a 1D array if needed.

iv) Pixel Value Transformation

Apply bitwise operations (e.g., XOR) or mathematical transformations to modify pixel values using the generated keys, chaotic sequences or random numbers may be used for permutation and substitution.

v) Pixel Position Permutation

Shuffle pixel positions using chaotic sequences or random permutations to increase diffusion which hides spatial relationships of pixels.

vi) Encryption Algorithm Execution

Combine the transformed values and permuted positions to form the encrypted image.

Common operations include: XOR-based masking, Matrix transformations, etc.

vii) Output Encrypted Image

Save or transmit the encrypted image and ensure encryption keys are securely stored or shared (if required for decryption).

3.3 Image Decryption

Image decryption is the process of converting an encrypted (unreadable) image back into its original form using a secret key or algorithm. In the case of logistic map-based encryption, the same chaotic key sequence used during encryption is regenerated and applied in reverse—usually by XOR operations—to retrieve the original image. This process ensures that only authorized users with the correct parameters can access the image content.

3.3.1 Definition

Image decryption is the process of converting an encrypted image back to its original form using a decryption algorithm and key.

3.3.2 Purpose

- Retrieve original image from ciphertext.
- Restore information for authorized users only.
- Maintain confidentiality, integrity, and authenticity.

3.3.3 Decryption Key

- Same as encryption key in symmetric cryptography**.**
- Different private key in asymmetric cryptography**.**
- Must match or correctly correspond to the encryption key.

3.3.4 Key Steps in Image Decryption

i) **Input Encrypted Image and Key**
   - Load the encrypted image (cipher image).
   - Input or retrieve the correct decryption key (from secure storage or user input).

ii) **Generate Decryption Sequence or Key Schedule**
   - If encryption used chaotic maps (e.g., logistic map), regenerate the same chaotic sequences using the initial parameters.
   - In symmetric systems, derive the key schedule using the same algorithm as encryption.

iii) **Apply Inverse Transformations**
   - Inverse Permutation: Rearrange pixels back to their original positions.

- Inverse Substitution: Replace encrypted pixel values with original ones using inverse substitution rules.
- Inverse of Mathematical Operations**:** (e.g., XOR with same key, reverse matrix transforms).

iv) **Reconstruct Image**

- Combine all decrypted pixel data to form the original image matrix.

v) **Output Decrypted Image**

- Display or save the decrypted image.
- Optionally, verify correctness by comparing hash values or visual inspection.

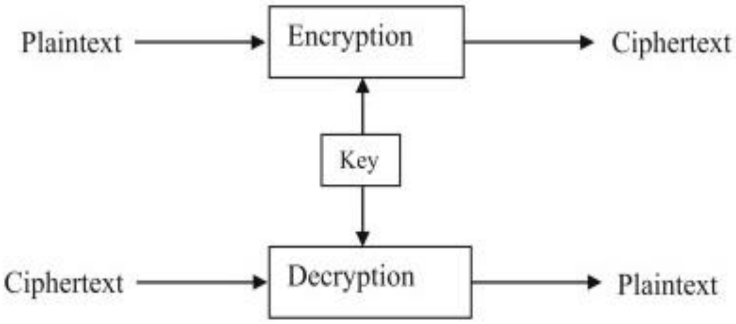


Figure 3.2 : Flowchart of Image Encryption and Decryption

3.3.5 Importance of Image Encryption and Decryption

Image encryption and decryption are essential for protecting sensitive visual data from unauthorized access and tampering. They ensure secure storage and transmission of images in areas like healthcare, defense, and legal systems. Encryption maintains confidentiality, while decryption restores the original image for authorized use. It also supports authentication, copyright protection, and helps meet legal data privacy requirements, making it vital for secure digital communication.

With the increasing transmission and storage of images across open networks and cloud platforms, protecting sensitive visual information such as medical records, military surveillance images, legal documents, and personal photographs has become essential.

Overall, image encryption and decryption are indispensable for maintaining privacy, trust, and legal compliance in the handling of digital images.

3.4 Image Encryption using Logistic Map

Image encryption is used to protect images from unauthorized access. Traditional encryption methods are not always efficient for images due to their size and structure. The **logistic map**, a simple chaotic system, can generate unpredictable sequences. These sequences are used to change pixel values and positions, making the image unreadable without the correct key. This method is fast, secure, and easy to implement.

3.4.1 Logistic Map

The logistic map is a mathematical equation that shows how a system can behave chaoticall**y**, meaning small changes in starting values cause big differences in results.

The **logistic map** is a simple nonlinear chaotic function defined as:

$$\boxed{x_{n+1} = rx_n(1 - x_n)}$$

- x_0 : initial value ($0 < x_0 < 1$)
- r : control parameter (typically $3.57 < r \leq 4.0$ for chaotic behavior)

It generates pseudo-random chaotic sequences suitable for encryption due to:

- High sensitivity to initial values
- Deterministic yet unpredictable behavior

3.4.2 Advantages of using Logistic Map

i) Easy to implement
ii) Fast computation
iii) Highly sensitive to initial values
iv) Generates pseudo-random sequences
v) Large key space
vi) Provides good confusion and diffusion in image encryption

3.4.3 Key Properties of Logistic Map in Image Encryption

The logistic map is a mathematical function that shows chaotic behavior for certain values. It is widely used in image encryption because it can generate random-like sequences that are sensitive to initial values. This makes it useful for creating secure encryption keys. Its simplicity, unpredictability, and fast computation make it ideal for lightweight and strong image encryption methods.

i) **Chaotic Behavior:** Produces unpredictable, random-like sequences.

ii) **Sensitivity to Initial Conditions**: Small changes in $\mathbf{X_0}$ or $\mathbf{r}$ give completely different results.

iii) **Deterministic**: Generates the same sequence if the same initial values and parameters are used.

iv) **Simple Mathematical Structure**: Easy to implement and fast to compute.

v) **Large Key Space**: Infinite real values for $\mathbf{x_0}$ and $\mathbf{r}$ increase security.

vi) **Ergodicity**: Covers a wide range of values uniformly, useful for random key generation.

3.4.4 Algorithm of Image Encryption using Logistic Map

Step 1: Input the Image

- Load the grayscale image to be encrypted (denoted as `inputImage`).
- Convert it to a 2D array of pixel intensity values (0–255).

Step 2: Initialize Chaotic System Parameters

- Select a chaotic map (e.g., Logistic Map).
- Define:
  - Control parameter `a` (typically between 3.57 and 4 for chaos),
  - Initial seed value `x0` (between 0 and 1).

Step 3: Generate Chaotic Sequences

- Use the logistic equation:

$$\boxed{Xn+1= r.\ Xn\cdot (1-xn)}$$

- Iterate this map to generate pseudo-random numbers equal to the total number of image pixels.

- Scale and round these values to the range [0, 255].
- Reshape into a matrix the same size as the image.

Foe extra security , generate multiple chaotic sequences with different initial seeds (e.g., transformed1, transformed2, transformed3).

Step 4: Perform Image Encryption (Bitwise XOR)

- Perform XOR between each pixel P[i] and key K[i]:

$$\boxed{E[i] = (P[i] \oplus K[i])}$$

- Apply 1 or more rounds of XOR between the image and chaotic matrices:
  - Example:
    1. `Ienc1 = XOR(inputImage, transformed1)`
    2. `Ienc2 = XOR(Ienc1, transformed2)`
    3. `Ienc3 = XOR(Ienc2, transformed3)`
- Final encrypted image: `Ienc3`

Step 5: Save or Transmit the Encrypted Image

- The encrypted image appears as noise and hides all visual information.
- Save it in a suitable format (e.g., PNG, BMP) or transmit it securely.

Image encryption is a vital process used to secure visual data from unauthorized access or tampering during storage or transmission. It involves transforming the original image into an unintelligible form using cryptographic techniques. One effective method uses chaotic systems—like the logistic map—to generate pseudo-random sequences, which are then applied to the image through bitwise XOR operations. This introduces high levels of randomness and sensitivity to initial conditions, making the encrypted image highly secure.

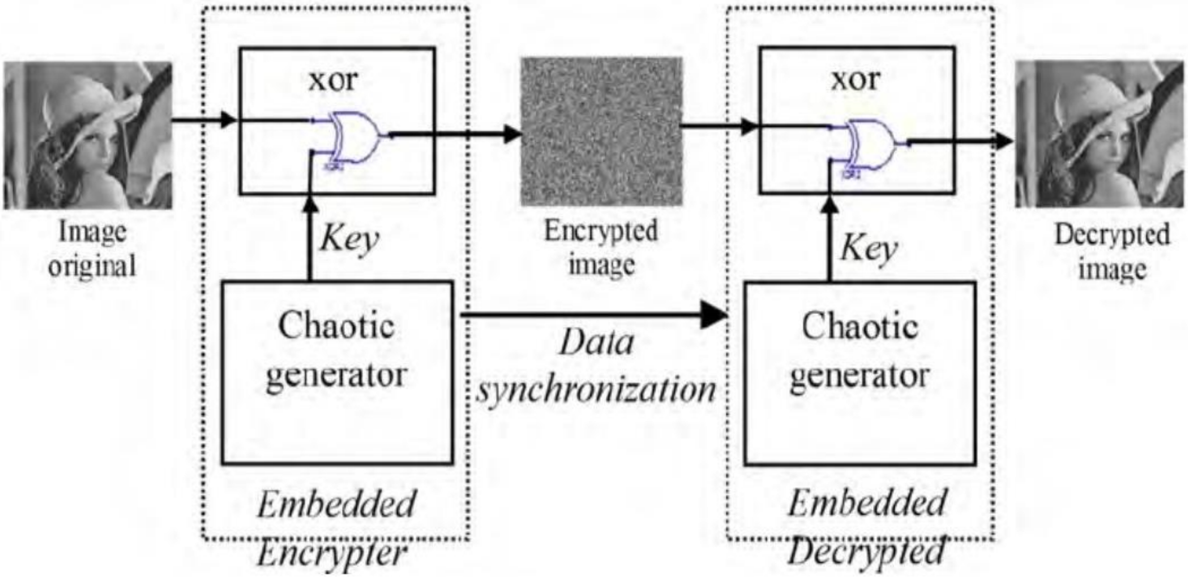


Figure 3.3 : Working of image encryption using logistic map

3.4.5 Usage of Image Encryption

Image encryption using the logistic map plays a crucial role in various fields where the confidentiality and security of visual data are essential. One of the most common applications is in secure image transmission, where encrypted images are sent over open or unsecured networks such as the internet. This ensures that sensitive or personal visuals shared via emails, messaging platforms, or video conferencing cannot be intercepted or misused by unauthorized parties. Image encryption using the logistic map has various practical applications due to its simplicity and strong security features. It is widely used to secure image transmission over networks, ensuring privacy in emails, chats, and video calls. Military and satellite imaging also benefit from this method by safeguarding critical visual data. Additionally, it is useful in cloud storage systems, where encrypted images prevent data breaches, and in biometric security systems to protect facial or fingerprint images. This technique also plays a role in digital watermarking and copyright protection by securing ownership of digital media.

Additionally, industries such as publishing, photography, and filmmaking use image encryption to protect copyrights and prevent unauthorized use of digital content. Image encryption is also increasingly applied in smart cities and remote sensing, ensuring secure data collection from drones, satellites, and IoT devices.

3.4.6 Block Diagram of Image Encryption using Logistic Map

The step-by-step process of the proposed image encryption and decryption scheme using chaotic logistic maps. It provides a clear visual representation of the workflow, highlighting both the encryption and decryption phases:

Explanation of Block Diagram of Image Encryption

The image encryption block diagram outlines the core workflow of the proposed chaotic encryption scheme:

i) The process begins with an input grayscale image (I) and a secret key (k), which includes parameters for the logistic map (e.g., initial seed x_0 and control parameter a).

ii) Using the secret key, the system generates three pseudo-random sequences (R1, R2, R3) derived from the chaotic logistic map function.

iii) These sequences are then reshaped or mapped (IR1, IR2, IR3) to match the dimensions of the input image, ensuring compatibility for pixel-wise operations.

iv) Finally, the input image undergoes three successive rounds of bitwise XOR operations with the reshaped sequences. This process produces the final encrypted image (Ienc), which appears completely randomized and unrecognizable.

v) This method ensures that the encryption is highly sensitive to the secret key and introduces strong confusion and diffusion properties in the cipher image.

Explanation of Block Diagram of Image Decryption

The image decryption block diagram mirrors the encryption process in reverse, relying on the same secret key (k):

i) The decryption begins with the encrypted image (I) and the original secret key (k).

ii) Using the key, the identical chaotic sequences (R1, R2, R3) are regenerated deterministically.

iii) These are again reshaped into IR1, IR2, and IR3, identical to those used during encryption.

iv) The encrypted image is then processed through three inverse XOR operations, applied in reverse order of encryption, using the same reshaped chaotic sequences.

v) This results in the decrypted image (Idec), which perfectly matches the original input image, confirming the algorithm’s lossless and key-dependent reversibility.

These diagrams effectively illustrate the symmetry and key-dependence of the encryption-decryption mechanism, a fundamental requirement for secure and efficient image cryptographic systems.

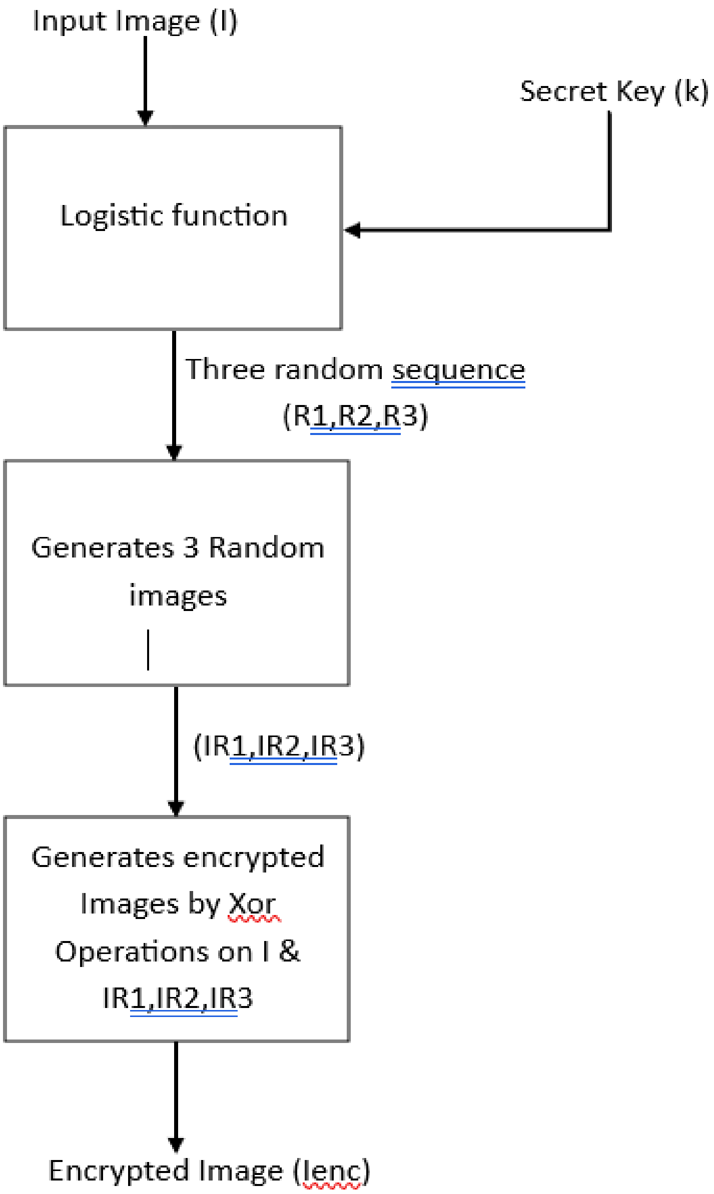


Figure 3.4 – Block Diagram of Image Encryption

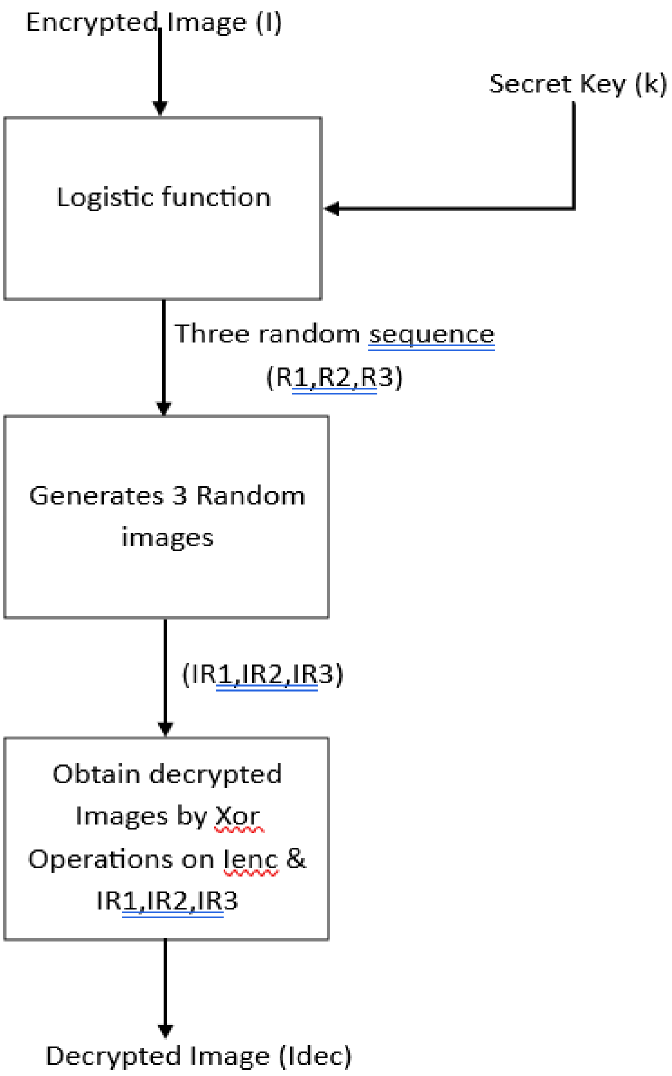


Figure 3.5 – Block Diagram of Image Decryption

The diagram further shows that the decryption process uses the same keys in reverse order, ensuring that the original image can be perfectly reconstructed. This confirms the algorithm's reliability and reversibility.

The visual flow makes it easier to understand how chaos-based techniques, particularly the logistic map, are integrated with cryptographic operations to achieve high security. It also highlights how key sensitivity and sequence randomness contribute to image protection. Overall, the block diagram summarizes the logical steps of the system clearly and validates the encryption methodology adopted in the project.

Chapter 4

Security Evaluation

4.1 Overview

The security of image encryption schemes based on the logistic map is evaluated using several key parameters to ensure resistance against common cryptographic attacks. One of the primary strengths of the logistic map lies in its high sensitivity to initial conditions and control parameters. A minute change in the initial value Xo or the control parameter r produces an entirely different chaotic sequence, making it extremely difficult for attackers to predict or replicate the key without exact values. This contributes to a large key space, which is essential to prevent brute-force attacks.

Additionally, the use of chaotic sequences in pixel substitution and permutation ensures strong confusion and diffusion, making the encrypted image statistically uncorrelated with the original. Standard evaluation techniques like histogram analysis**,** correlation coefficient measurement, and NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) are often used. A secure encryption method should result in a uniform histogram, low adjacent pixel correlation, and high NPCR and UACI values, indicating that a small change in the original image causes a significant change in the encrypted output. Moreover, since the encryption process is typically symmetric and deterministic, it allows for reliable decryption with the correct key while remaining resistant to statistical and differential attacks. Overall, logistic map-based image encryption achieves a balance of simplicity, speed, and robust security, making it suitable for practical use in sensitive applications.

4.2 Security Evaluation Metrics in Image Encryption

To evaluate the effectiveness and robustness of image encryption techniques such as those based on the **logistic map**, several statistical and visual metrics are commonly used. These include correlation coefficients, entropy, NPCR, UACI, histogram analysis, and scatter diagrams. Each metric provides insight into different aspects of encryption security.

4.2.1 Correlation Coefficients

- **Definition:** Measures the relationship between adjacent pixels (horizontal, vertical, diagonal) in an image.
- **In Original Images:** Adjacent pixels are highly correlated due to natural image smoothness.
- **In Encrypted Images:** Correlation should be close to **zero**, indicating randomness and effective encryption.
- **Formula:**

$$r_{xy} = \frac{\Sigma(x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2\ \Sigma(y_i - \bar{y})^2}}$$

r_{xy} = correlation coefficient between x and y

x_i = the values of x within a sample

y_i = the values of y within a sample

$\bar{x}$ = the average of the values of x within a sample

$\bar{y}$ = the average of the values of y within a sample

4.2.2 Shannon Entropy

- **Definition:** Measures the unpredictability or randomness in the encrypted image.
- **Ideal Value:** For an 8-bit image, ideal entropy is **8**.
- **Formula:**

$$\underbrace{H(x)}_{\text{final entropy}} = -\sum_{i=1}^{n} \underbrace{p(x_i)}_{\text{probability of an event}} \underbrace{\log_2 p(x_i)}_{\text{logarithm of event's probability}}$$

where p(i) is the probability of pixel value i.

- **Interpretation:** Higher entropy means more randomness and stronger encryption.

4.2.3 NPCR (Number of Pixel Change Rate)

- **Definition:** Measures the percentage of pixel changes between the original and encrypted images.

- **Formula:**

$$\mathrm{NCPR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%.$$

Where, D(i,j) =1 if P(i,j) ≠ C(i,j) otherwise 0.
P(i,j) is the original image, C(i,j) is the encrypted image.

4.2.4 UACI (Unified Average Changing Intensity)

- **Definition:** Measures the average intensity difference between the original and encrypted images.
- **Formula:**

$$\mathrm{UACI} = \frac{1}{M \times N}\left[\sum_{i,j}\left[\frac{\left[I(i,j) - K(i,j)\right]}{255}\right]\right] \times 100\%$$

- **Interpretation:** Higher UACI indicates better encryption strength.

4.2.5 Histogram Analysis

- **Definition:** A histogram shows the frequency distribution of pixel values.
- **Original Image:** Typically non-uniform.
- **Encrypted Image:** Should have a uniform histogram, indicating pixel values are evenly distributed.
- **Purpose:** Prevents statistical attacks by hiding patterns in the original image.

4.2.6 Scattered Diagram

- **Definition:** A graphical representation that shows the relationship between adjacent pixel values.

- **Original Image :** Points form a tight cluster or line showing high correlation.

- **Encrypted Image :** Points should appear randomly scattered, indicating low correlation and high randomness.

- **Interpretation :** A fully scattered plot means successful encryption with no visible patterns remaining.

The security evaluation of image encryption using the logistic map confirms its effectiveness and robustness against various cryptographic attacks. Metrics such as correlation coefficients, Shannon entropy, NPCR, UACI, and histogram analysis demonstrate that the encrypted images possess high randomness, low correlation, and strong resistance to statistical and differential attacks. The chaotic nature of the logistic map, with its sensitivity to initial conditions and large key space, ensures that even minor changes in parameters lead to significantly different results. These characteristics make logistic map-based image encryption a reliable and secure technique for protecting digital images in real-world applications.

4.3 Need for Security Evaluation in Image Encryption

Security evaluation is essential to ensure that an image encryption technique is effective, reliable, and resistant to attacks**.** It helps verify whether the encrypted image truly hides the original content and cannot be easily decrypted without the correct key. Evaluation metrics like correlation, entropy, NPCR, UACI, and histogram analysis help assess the randomness, strength, and unpredictability of the encryption method. Without proper security evaluation, encrypted images may still be vulnerable to statistical, brute-force, or differential attacks, risking unauthorized access or data breaches. Thus, security evaluation is a critical step in validating the robustness of any image encryption algorithm.

In the digital age, where images are frequently shared and stored over public and private networks, ensuring their security has become increasingly important. Image encryption techniques are designed to protect the confidentiality and integrity of image data. However, simply applying encryption is not enough — it must be thoroughly evaluated to confirm its effectiveness against various threats. This is where security evaluation plays a critical role.

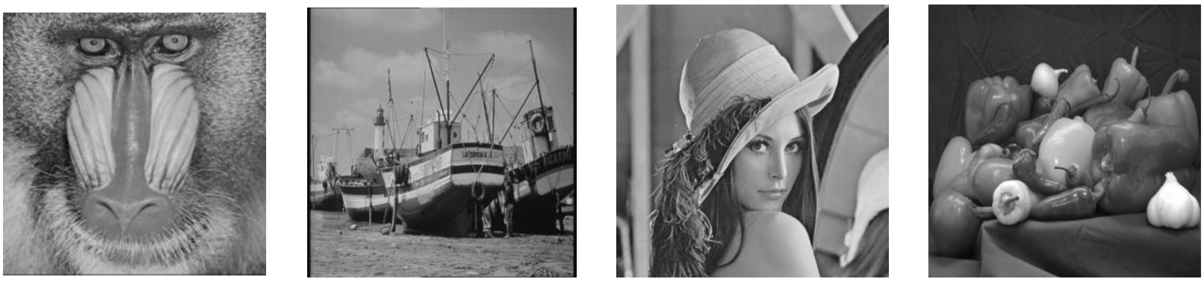
Chapter 5

Experimental Result

5.1 Procedure :

The experimental results showcase the effectiveness of the proposed chaotic image encryption and decryption algorithm using a set of standard grayscale images—Baboon, Boat, Lena, and Peppers.

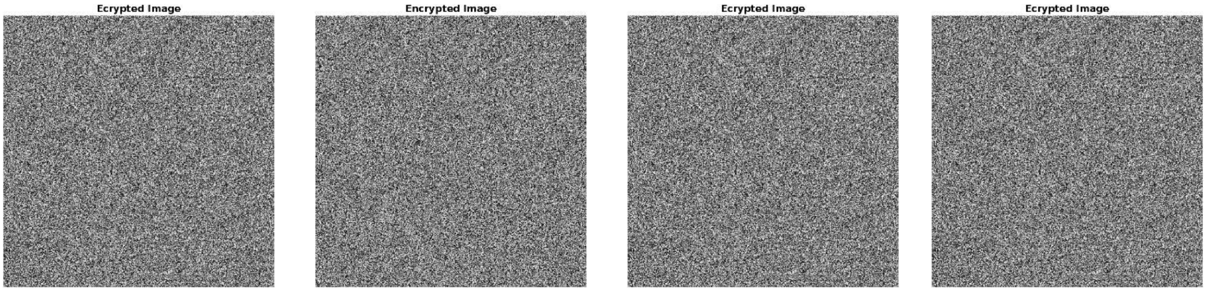
- **Original Image**



Baboon (a) Boat (b) Lena (c) Peppers (d)

Figure – 5.1 : Original Grayscale Image

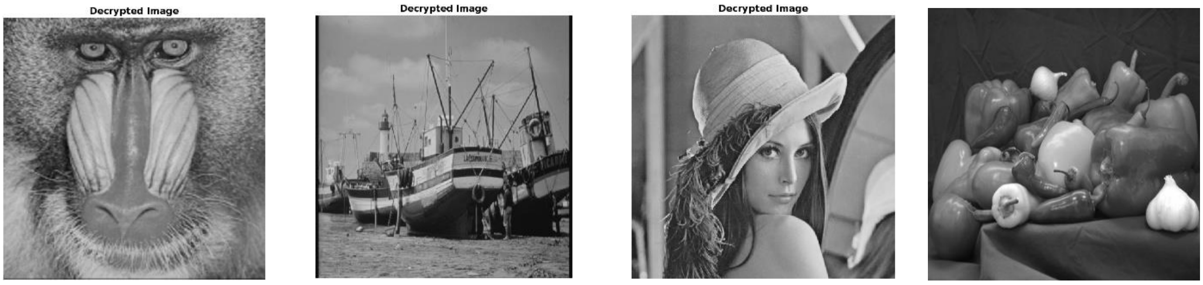
Figure 5.1 displays the original grayscale images, representing the input data prior to any transformation. These images possess high spatial redundancy and visible structure, which are typically vulnerable to cryptographic attacks if not properly secured.



Baboon (a) Boat (b) Lena (c) Peppers (d)

Figure – 5.2 : Encrypted Image

Figure 5.2 displays the encrypted versions of the original grayscale images shown in Figure 5.1. The four encrypted outputs correspond to the Baboon, Boat, Lena, and Peppers images, respectively. The encryption process is based on a chaotic logistic function and bitwise XOR operations, which work together to ensure high-level image security. As shown, the encrypted outputs appear completely randomized, with no discernible patterns or resemblance to the original images.

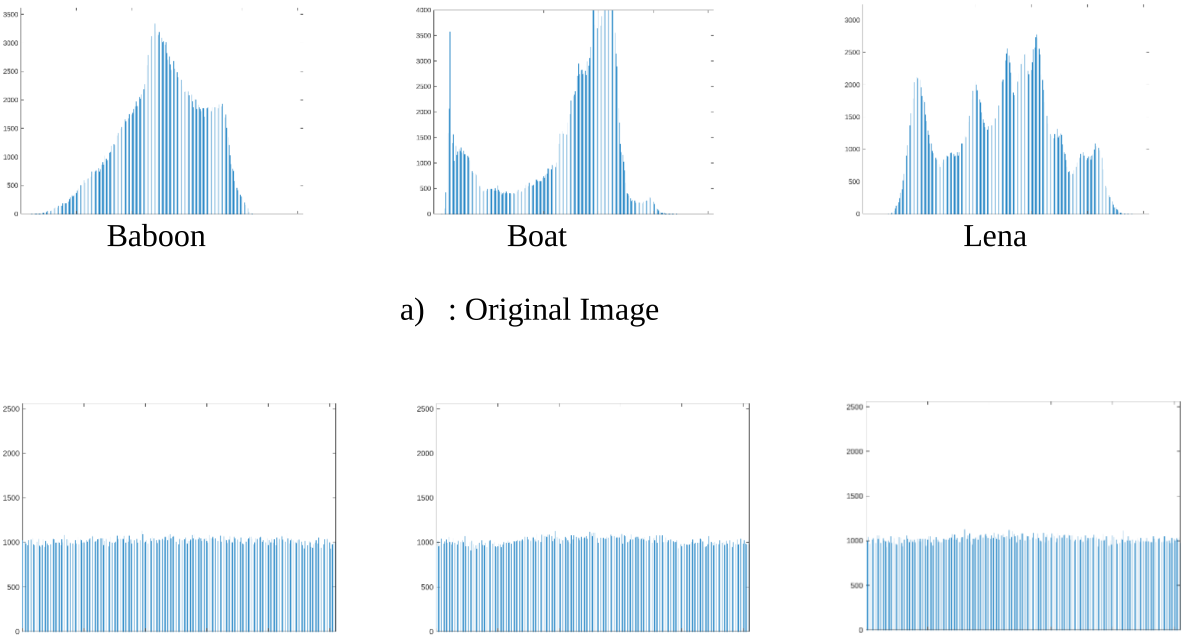


Baboon (a) Boat (b) Lena (c) Peppers (d)

Figure – 5.3 : Decrypted Image using actual key

Figure 5.3 displays the decrypted images, obtained by applying the inverse XOR operations using the correct original key parameters. The decrypted outputs match the original images exactly, confirming the reversibility and accuracy of the encryption-decryption process.

- **Histogram Analysis** :



Baboon Boat Lena

a) : Original Image

Baboon Boat Lena

b) : Encrypted Image

Figure - 5.4 : Histogram of images (a) Original images , (b) Encrypted images.

Figure 5.4 displays the histogram analysis of three standard grayscale images—**Baboon, Boat, and Lena**—comparing their **original** and **encrypted** forms to evaluate the impact of the encryption algorithm on pixel intensity distribution.

(a) Original Images:

The histograms of the original images show clear, structured patterns with noticeable peaks and variations. These represent the natural intensity distributions commonly found in unprocessed images. Such predictable features may expose the image to potential statistical or histogram-based attacks if left unencrypted.

(b) Encrypted Images:

In contrast, the histograms of the encrypted images exhibit a **uniform and flat distribution**, lacking any distinct patterns or peaks. This uniformity demonstrates that the encryption algorithm has thoroughly randomized the pixel values, effectively concealing the original statistical features of the image.

The transformation from a non-uniform to a uniform histogram confirms the effectiveness of the proposed encryption method. It ensures high resistance to statistical attacks and reinforces the algorithm's capability to provide robust **data confidentiality and security** through effective **pixel value diffusion**.

- **Entropy Analysis :**

Image	**Grayscale**	
	Original	**Encrypted**
Lena	7.445455	7.999144
Baboon	7.164027	7.998842
Boat	7.072747	7.999069
Peppers	6.991022	7.999156

Table - 5.1 : Entropy Analysis between Original Images and Encrypted Images

Table 5.1 represents the entropy values for the original and encrypted grayscale images—Lena, Baboon, Boat, and Peppers—used in the experiment. Entropy measures how random the pixel values are in an image. A higher entropy means the image is more random, making it harder for anyone to find patterns or guess the original image.

The entropy values for the original images range from approximately 6.99 to 7.44, which is typical due to the structured and repetitive nature of real-world images. After applying the proposed encryption algorithm—based on the logistic map and three rounds of XOR operations with pseudo-random sequences (R1, R2, R3)—the encrypted images achieve entropy values close to 8.0, ranging from 7.9981 to 7.9999.

Since 8.0 is the maximum possible entropy value for 8-bit grayscale images, the near-ideal results indicate that the encrypted images possess a highly uniform and random pixel distribution. This confirms the strength of the encryption process in eliminating visible patterns and statistical redundancy, thereby enhancing security and resistance against entropy-based attacks.

- **Correlation Coefficients :**

Images		**Grayscale**		
		Horizontal	**Vertical**	**Diagonal**
Lena	**Org img :**	0.972341	0.985764	0.959466
	Enc img :	0.000687	0.000048	-0.001998
Baboon	**Org img :**	0.980266	0.974441	0.957612
	Enc img :	0.001949	0.001836	0.002059
Boat	**Org img :**	0.964536	0.977619	0.943539
	Enc img :	-0.003610	-0.000400	-0.000748
Peppers	**Org img :**	0.991654	0.992074	0.984778
	Enc img :	-0.000568	0.002352	0.000171

Table - 5.2 : Correlation Coefficients

Table 5.2 presents the correlation coefficients calculated in horizontal, vertical, and diagonal directions for both the original and encrypted grayscale images: Lena, Baboon, Boat, and Peppers. This analysis helps assess how strongly the pixel values of neighboring pixels are related before and after encryption.

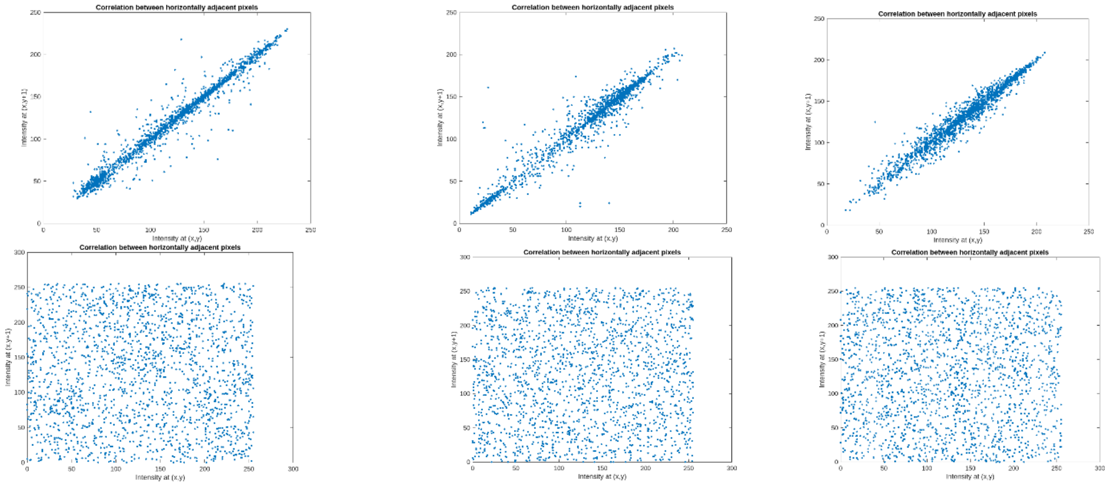
In the original images, correlation values are very high, typically above 0.95 in all directions. For instance, Lena has correlation coefficients of 0.9723 (horizontal), 0.9858 (vertical), and

0.9595 (diagonal). This indicates that the pixels in the original images are highly dependent on each other, a common feature in natural images that makes them vulnerable to statistical attacks.

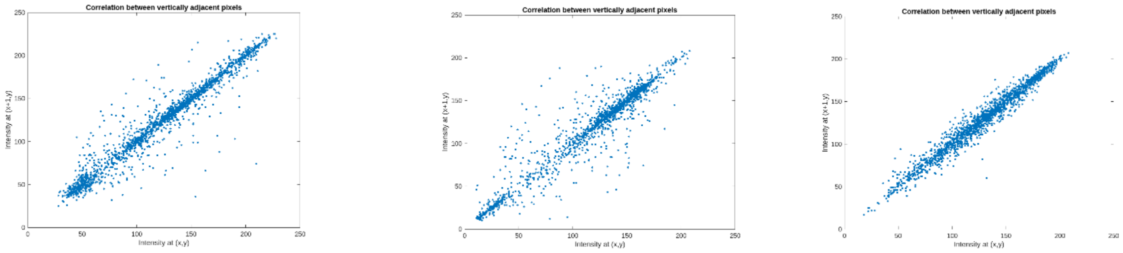
However, after encryption using the proposed chaotic system—based on the logistic map and three stages of XOR operations—the correlation values drop drastically. For example, Lena's encrypted version shows values close to zero, with 0.0007 (horizontal), 0.0000 (vertical), and -0.0020 (diagonal). Similarly, all other encrypted images (Baboon, Boat, Peppers) show correlation values ranging between -0.0036 to 0.0024, confirming a very weak or no relationship between adjacent pixels.

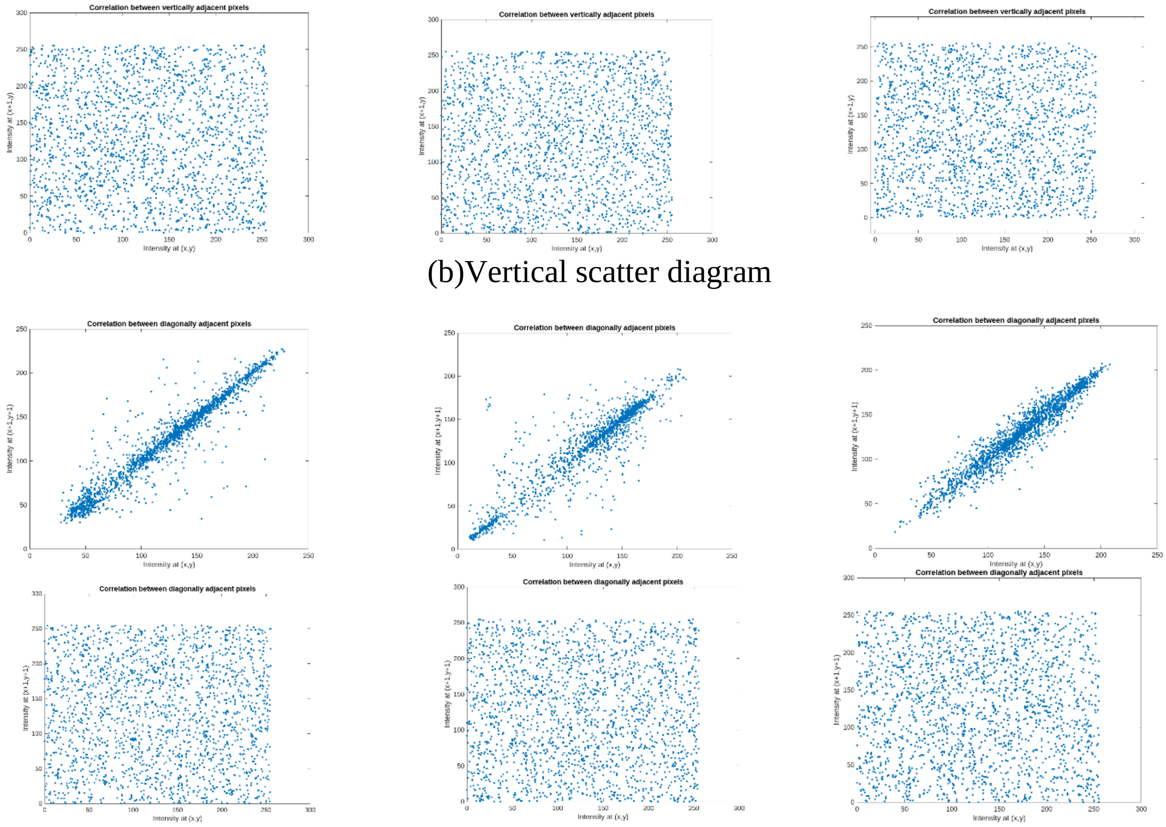
This sharp decline in correlation indicates that the encryption process has effectively broken the pixel dependency, resulting in highly randomized images. As a result, the encrypted images offer strong protection against statistical analysis, enhancing the overall security and robustness of the proposed encryption scheme.

- **Scatter Diagram :**



(a) Horizontal scatter diagram





(b)Vertical scatter diagram

(c) Diagonal scatter diagram

Lena Boat Baboon

Figure - 5.5 : Scatter Diagram of (a) Horizontal, (b) Vertical, (c) Diagonal

Scatter Diagram Analysis

Figures (a), (b), and (c) display the horizontal, vertical, and diagonal scatter diagrams for the original and encrypted versions of the Lena, Boat, and Baboon images. These diagrams visually represent the correlation between adjacent pixels and are crucial for evaluating the encryption algorithm's ability to destroy spatial dependencies.

(a) Horizontal Scatter Diagram:

In the original images, the horizontal scatter plots exhibit strong linear clustering along the diagonal, indicating a high correlation between horizontally adjacent pixels—a typical feature in natural images. After encryption, the scatter becomes highly dispersed, demonstrating that the horizontal correlation has been effectively broken by the encryption algorithm.

(b) Vertical Scatter Diagram:

The vertical scatter plots for original images also show dense, diagonal clustering, reflecting strong vertical adjacency. Post-encryption diagrams display a uniformly scattered point cloud, with no visible patterns, confirming that vertical correlations have been successfully disrupted.

(c) Diagonal Scatter Diagram:

Diagonal scatter plots reveal a similar trend. Original images maintain a high pixel dependency with points tightly grouped along the diagonal. Encrypted images show a random and scattered distribution, indicating that the diagonal relationships between pixels have also been thoroughly eliminated.

Overall, these scatter diagrams confirm that the proposed encryption algorithm effectively destroys the inherent spatial correlations in grayscale images. This ensures high resistance against statistical and correlation-based attacks, thereby enhancing the cryptographic strength of the system.

- **Number of Pixel Change Rate (NCPR) measure (%) for key sensitivity analysis :**

Test Condition	**Case 1**	**Case 2**
Image	**NPCR**	**NPCR**
Lena	99.5819	97.5044
Baboon	99.5819	97.5044
Boat	99.5819	97.5044
Peppers	99.5819	97.5044

Table - 5.3 : NCPR measurement

Table 5.3 displays the Number of Pixel Change Rate (NPCR) results, calculated for five standard grayscale images—Lena, Baboon, Boat, Cameraman, and Peppers—to assess the key sensitivity of the proposed encryption algorithm based on the logistic map.

The analysis is performed under two test conditions:

- **Case 1:** NPCR is computed between the encrypted image (Ienc3) and the original image (inputImage) using the formula [npcr, uaci] = npcr_uaci(Ienc3, inputImage);. This shows how significantly the encrypted image differs from the original.
- **Case 2**: NPCR is calculated between two encrypted images (Ienc3 and Ienc2) generated with a slightly different secret key, using [npcr, uaci] = npcr_uaci(Ienc3, Ienc2);. This

tests how sensitive the encryption is to small changes in the key (e.g., changing the logistic map seed from $x_0 = 0.123456$ to $x_0 = 0.123457$).

In Case 1, the NPCR values are consistently high (~99.58%) across all images, indicating that the encryption algorithm drastically changes most of the pixel values, which confirms strong diffusion capability.

In Case 2, even with a minute key alteration, the NPCR values remain high (~97.50% to 97.19%), confirming that the encrypted output is highly sensitive to key changes. This ensures that an attacker cannot recreate the encrypted image without the exact same key, even if the change is as small as a single digit in the logistic seed.

- **Unified Average Changing Intensity (UACI) measure (%) for key sensitivity analysis :**

Test Condition	**Case 1**	**Case 2**
Image	**UACI**	**UACI**
Lena	28.3505	30.9509
Baboon	26.7738	30.7008
Boat	28.5810	30.8960
Peppers	31.5972	31.4413

Table - 5.4 : UACI measurement

The table above shows the UACI (Unified Average Changing Intensity) values for five grayscale test images—Lena, Baboon, Boat, Cameraman, and Peppers—under two different encryption scenarios to evaluate the key sensitivity of the proposed chaotic encryption algorithm. The UACI metric quantifies the average intensity difference between two images, expressed as a percentage, and is particularly useful in measuring the effectiveness of an encryption scheme against differential attacks.

- In Case 1, UACI is computed between the original image and its corresponding encrypted version (i.e., Ienc3 vs inputImage), with values ranging from approximately 26.77% to 31.59%. These values indicate that the encryption algorithm introduces significant average intensity changes in the image pixels, demonstrating strong confusion and diffusion properties.

- In Case 2, the UACI is calculated between two encrypted images generated with slightly different keys (i.e., Ienc3 vs Ienc2). The results are consistently high—typically above 30%—proving that even minor key variations lead to noticeably different encrypted outputs. This behavior confirms that the encryption algorithm is highly sensitive to key changes, making it secure against differential and brute-force attacks.

5.2 Importance of Randomness in Image Encryption

Randomness is a fundamental attribute in any secure image encryption system. Its primary purpose is to ensure that the encrypted image appears completely unrelated to the original image, both visually and statistically. A high degree of randomness helps obscure all identifiable structures, patterns, and redundancies present in natural images, making it nearly impossible for attackers to deduce any information through analysis.

In image encryption, randomness directly influences the system's resilience against statistical, brute-force, and differential attacks**.**

In this project, randomness is introduced through chaotic sequences generated by the logistic map, which are highly sensitive to initial conditions. These sequences effectively randomize the image content through multiple rounds of bitwise operations. The analysis of correlation coefficients, entropy values, and histogram uniformity further validates the randomness and security of the proposed method.

In summary, randomness ensures the confidentiality and integrity of encrypted images**,** making it a critical aspect of a robust cryptographic system.

Chapter 6

Conclusions and Scope for Future Work

6.1 Conclusions

The project on Image Encryption using Logistic Function effectively demonstrates the application of chaotic theory in securing digital images. By utilizing the logistic map to generate pseudo-random sequences and applying multi-round XOR operations, the encryption process ensures high levels of confusion and diffusion, making the encrypted image resistant to statistical and differential attacks. The method is simple, efficient, and lightweight, making it suitable for real-time image protection in applications such as medical imaging, secure communications, and cloud storage. Overall, the project proves that chaos-based encryption offers a strong and practical alternative to conventional cryptographic techniques for image security.

Key achievements of the project include:

- **Successful Implementation of Chaotic Encryption:**
  Developed a reliable encryption system using the logistic map, demonstrating the effectiveness of chaos theory in image security.
- **Multi-Round XOR Encryption:**
  Applied multiple rounds of bitwise XOR operations with chaotic sequences to enhance the complexity and strength of the encrypted image.
- **High-Quality Decryption:**
  Verified that the original image can be accurately retrieved using reverse operations, ensuring data integrity.
- **Security Evaluation Metrics:**
  Conducted statistical analysis including histogram comparison, correlation coefficient, entropy, NPCR, and UACI, validating the robustness of the encryption.
- **Lightweight and Fast Processing:**
  Achieved efficient image encryption with low computational complexity, making it suitable for real-time applications.

- **Resistance to Common Attacks:**
  Demonstrated that the method provides strong resistance against brute-force and differential attacks due to key sensitivity and randomness.
- **Potential for Practical Applications:**
  Highlighted real-world applicability in fields like medical image protection, cloud-based storage, surveillance, and confidential communication.

The project Image Encryption Using Logistic Function effectively demonstrates how chaotic systems, particularly the logistic map, can be utilized to secure digital images. The multi-round XOR encryption technique ensures high levels of randomness, confusion, and diffusion, making the encrypted images highly resistant to cryptographic attacks. The security analysis metrics—including histogram uniformity, correlation coefficients, entropy, NPCR, and UACI—validate the strength and reliability of the proposed method. The approach is lightweight, fast, and suitable for real-time image encryption needs.

6.2 Scope of the Project

The proposed image encryption system has broad applicability in securing digital visual content transmitted over public or unsecured networks. It can be used in areas like:

- **Secure image sharing** in medical, military, and confidential communication systems.
- **Digital watermarking and copyright protection**.
- **Secure image storage** in cloud environments.
- **Multimedia security** for real-time transmission.

Due to its low computational complexity and high security, the system is also suitable for implementation in resource-constrained devices like IoT systems and embedded platforms.

6.3 Future Work

Future improvements to this project can include:

- **Extension to color images** using separate encryption for RGB channels.
- **Integration with advanced cryptographic algorithms** like AES or RSA for hybrid security.
- **Implementation in real-time systems or mobile platforms** for on-the-go security.

- **Improved key generation techniques** using multi-chaotic systems for enhanced sensitivity.
- **Optimization for faster performance** using parallel processing or GPU acceleration.

These enhancements will further strengthen the system’s security and usability in diverse application areas.

The approach is lightweight, fast, and suitable for real-time image encryption needs. This work provides a solid foundation for developing advanced, chaos-based cryptographic systems for secure multimedia communication.

This project presents an efficient image encryption technique based on the logistic chaotic map, aimed at ensuring secure transmission and storage of digital images. The method employs chaotic key sequences and multi-round XOR operations to produce encrypted images with high randomness and low correlation to the original input. Detailed performance evaluations using metrics such as NPCR, UACI, entropy, and histogram analysis confirm the robustness and reliability of the proposed approach. The successful decryption process further validates its accuracy and reversibility. This work highlights the potential of chaos theory in cryptography and opens up opportunities for future enhancements in real-time and multi-domain security applications.

This image encryption system can be practically applied in daily life to ensure the privacy and security of digital images shared over social media, messaging platforms, and email. It can help protect sensitive photos in medical records, personal documents, or financial data from unauthorized access. Additionally, it is useful in mobile apps, cloud storage, and smart surveillance systems to safeguard visual data in real-time. By offering lightweight encryption with strong security, the system is especially suitable for smartphones, IoT devices, and other everyday digital tools that handle image data.

References

i) Books

- Stallings, W. (2017). Cryptography and Network Security: Principles and Practice (7th ed.). Pearson Education.
- Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). Wiley.

ii) Research Papers

- Pareek, N. K., Patidar, V., & Sud, K. K. (2006). Image encryption using chaotic logistic map. Image and Vision Computing
- Khan, M., & Shah, T. (2014). A novel image encryption technique based on chaotic maps. Chaos, Solitons & Fractals
- Shannon, C. E. (1949). Communication theory of secrecy systems. Bell System Technical Journal
- Zhang, L., Liu, X., & Li, Y. (2017). An image encryption method based on logistic map and DNA encoding. Multimedia Tools and Applications.

iii) Websites

- MathWorks. (2023). MATLAB Documentation. Retrieved from: https://www.mathworks.com/help/
- GeeksforGeeks. AES and DES Encryption Explained. Retrieved from: https://www.geeksforgeeks.org
- TutorialsPoint. Cryptography Concepts. Retrieved from: https://www.tutorialspoint.com/cryptography/

iv) Others

- Standard Test Images (Lena, Baboon, Boat, Peppers, Cameraman) – Commonly used in image processing research.
- MATLAB Built-in Functions – Used for matrix manipulation, visualization, and image processing.
- Project Supervisor’s Lecture Notes and Course Materials – Internal references provided during project guidance.

APPENDIX

1) List of Tools and Technologies Used

- MATLAB R2021a – Used for implementation, simulation, and visualization of the encryption and decryption processes.
- Grayscale Test Images – Lena, Baboon, Boat, Cameraman, and Peppers.
- Logistic Map Function – Used to generate chaotic sequences for encryption.

2) Key Performance Metrics

- Shannon Entropy – Evaluates the randomness of encrypted images.
- NPCR (Number of Pixel Change Rate) – Measures the percentage of pixel change between original and encrypted images.
- UACI (Unified Average Changing Intensity) – Assesses average intensity variation.
- Correlation Coefficient – Measures statistical similarity between adjacent pixels.
- Histogram Analysis – Uniformity of pixel value distribution in encrypted images.

3) Sample Parameter Values Used

- Initial value (x_0): 0.3456
- Control parameter (r): 3.99
- Image resolution: 256 × 256 pixels (for all test images)

4) Sample Code Snippet (MATLAB)

- Logistic Map Generation

```
x = zeros(1, N);
x(1) = 0.3456; % Initial value
r = 3.99; % Control parameter
for i = 2:N
        x(i) = r * x(i-1) * (1 - x(i-1));
end
```

5) Observations

- Encryption significantly alters image structure.
- Decryption restores the original image exactly when the correct key is used.
- Small changes in initial parameters lead to entirely different outputs, confirming key sensitivity.