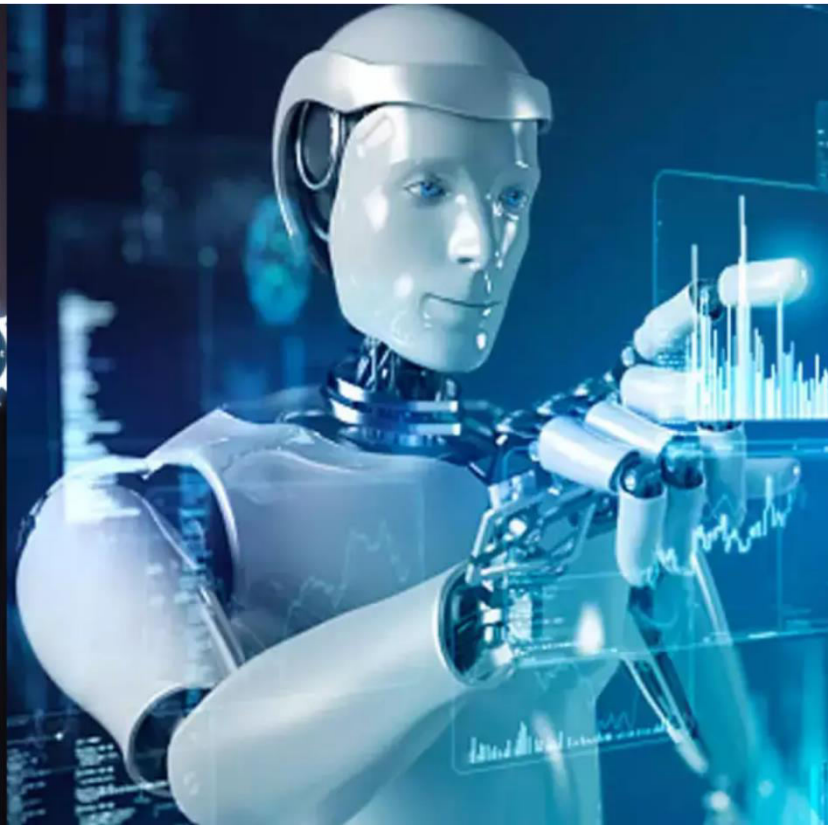
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

Image Encryption and Decryption Algorithm using XOR Operator

Dr. K. Baalaji, Kayala Durga Prasad, Kolapati Vijayabalu, Kethireddy Iswarya, Kesireddy Venkata Mani Lakshmi Narasima

Assistant Professor, Bharath Institute of Higher Education and Research, Chennai, India

IV – B.Tech CSE Students, Bharath Institute of Higher Education and Research, Chennai, India

ABSTRACT: As the result of the rapid development of technology, information science has become much easier, and therefore the problem of information security is growing. This paper deals with the secretiveness of images, so image encryption is that the latest trend in information caching. The novelty of the work lies in generating crucial images for encryption. The crucial image is then created with the assistance of a secret alphanumeric key. Each alphanumeric key will have an 8bit value generated by the binary key table. The challenge is to return up with an image encryption algorithm that is simple yet safe, with featherweight computer processing. This encryption algorithm which mixes Playfair cipher and therefore the Vigenere cipher gives better results. An experiment showed a correlation between the precipitation of the image after encryption and its decline. Supported the standard of quality of encryption, the speed of change of image pixels was high enough for the cipher image to be difficult to spot. The image is meant to be more distorted in this fashion. The deciphered image is obtained by applying the backward process.XOR technique is used in the present paper, for security analysis using XOR technique is an effective method of scrambling, in visual component which may be used as an important visual cryptography. For secure transmission of an image XOR cipher is a best technique.

KEYWORDS: Encryption, Decryption and Cryptography

I. INTRODUCTION

Computer-based image processing methods help manipulate digital images. In the form of imaging dataSatellite platform detectors contain scarcities. Experiencing colorful phases of processing prompts us to look beyond similar excrescences and encourages originality. Pre-processing, improvement and display, and information birth are the three general phases all types of data have to pass through when using digital fashion. A new fashion is needed for preventing information leakage as a result of the advancement of the data age. For digital data fashion had been developed that is cracking it[8]. This is converting data of readable form into non readable form, in order that if a hacker hack the information he cannot understand it until he know the decryption fashion and decryption word. Now a daysDigital revolution has changed the lifestyle of a common man. We are moving towards complete digitalization. Most importantly, transactions done using the digital system have saved our time and efforts. We use different forms of data such as audio voice and images to communicate from one end to another end. These data may carry plenty of important information especially for defense or any other ministry. Thus secure data transmission is a major area of concern for many researchers. Visual encryption is one of the techniques to hide the original message for unwanted persons. There are many techniques reported for visual encryption such as sub-image encryption method
[2]

, Multilevel encoding , R prime shuffle technique Block-based Scrambling, Random Grid technique , Arnold Cat map etc. Our article deals with one of the methods to do the same. We have developed a new algorithm to scramble the image to provide security while sending the information to an individual. We performed various testing on the scrambled image to judge its capability to handle different types of attack.
[3].

There are massive amounts of sensitive information managed and stored online in the cloud or on connected servers. Encryption uses cyber security to defend against brute-force and cyber-attacks, including malware and ransom ware

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

Data encryption works by securing transmitted digital data on the cloud and computer systems. There are two kinds of digital data, transmitted data or in-flight data and stored digital data or data at rest.Modern encryption algorithms have replaced the outdated Data Encryption Standard to protect data . These algorithms guard information and fuel security initiatives including integrity, authentication, and non-repudiation. The algorithms first authenticate a message to verify the origin. Next they check the integrity to verify that contents have remained unchanged. Finally, the nonrepudiation initiative stops sends from denying legitimate activity.

II. METHODOLOGY

Although there are numerous ways out there, but here we tried to build the simplest and most secured way in order to perform Encryption and Decryption.
The proposed methodology to perform this process is using XOR operation. In here XOR operation is considered to be as one of the Symmetrical Encryption method.
In order to protect it from illegal access and keep it private and secure, we simply turn our data or information into secret code.

Firstly, let's understand the functionality of XOR operation:
The basic XOR cipher in cryptography is an example of an additive cipher, an encryption technique that functions in accordance with the principles
A^ 0 = A, A ^ A = 0, A ^B = B^A, (A ^B)^ C = A ^(B^ C), (B ^A)^ A = B^ 0 = B

This process is also known as modulus 2 addition (or subtraction, which is identical). According to this reasoning, a string of text may be encrypted by utilizing a specified key and the bitwise XOR operation on each character. Simply performing the XOR function again with the key will get rid of the encryption and allow you to decode the result. Using the repeating key 11110011, the string "Wiki" (01010111 01101001 01101011 01101001

01010111 01101001 01101011 01101001

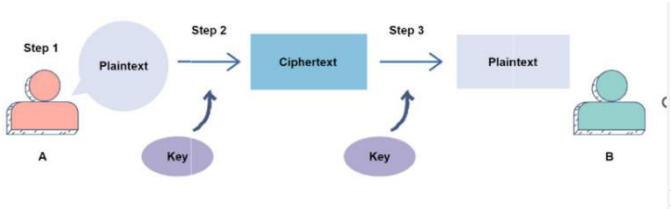
^ 11110011 11110011 11110011 11110011

=10100100 10011010
10011000 10011010
^ 111100111111001

=01010111 01101001

Before delving into the operation of XOR encryption and decryption, it is crucial to establish the foundations of cryptography. Communication between two endpoints is protected by encryption. For instance, in order to encrypt and decode a message that A wants to transmit to B, both A and B need a key. These are the steps involved:
1)The original text message that A intends to convey to B is known as the plaintext.
2)The text that A has encrypted using the key is called the ciphertext.
3)B will use the key to convert the ciphertext back to the plaintext and read the message. The steps above are

shown in the figure below:



Now let's implement the above concept into .The software shown below illustrates the fundamental method of encryption using XOR operator

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

```
# Assign values data
= 1281
key = 27

# Display values
print('Original Data:', data)
print('Key:', key)

# Encryption

data = data ^ key
print('After Encryption:', data)

# Decryption data = data ^ key
print('After Decryption:', data)
```

As we can see, the program above uses two variables, data, and a key, and when we perform an XOR operation on them for the first time, we obtain encrypted data. After that, we obtain the same result as our input variable data when we again do the XOR operation between our data and key (decrypted data). When encrypting and decrypting a byte array of Images, the same reasoning will be used.

In the process of Encryption, we will first choose an image, after which we will convert it into a byte array, converting the entire image data into numeric form so that we can simply do the XOR operation on it. Now, the data will change anytime we apply the XOR function to each value in the byte array, making it impossible for us to access it. But we must always keep in mind that our encryption key is crucial because we cannot decrypt our image without it. It serves as a decryption password.

Now once when the Encryption is done and we have obtained an encrypted output to our image. And here the main thing is that only the sender and receiver will know the Key, which can be manually set by the sender while the process of Encryption. Through this we can obtain more security.

And when the topic comes to Decryption, It only involves transforming our encrypted data into readable form. Here, we'll use the same XOR procedure to decode a picture that has been encrypted. However, never forget that our encryption key and decryption key must match.

III. PROPOSED METHODOLOGY

Encryption key: 22

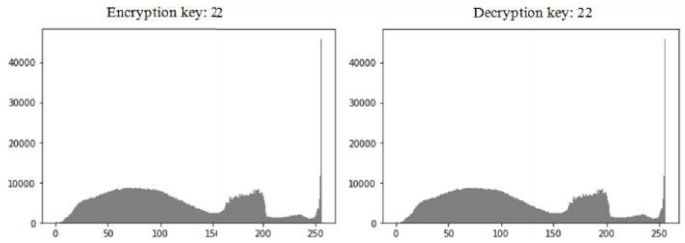
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|



- X Scrambling of Image: Four images are used to analyzing their suitability of scrambling technique. Original images along with their scrambled images are shown in fig 1. Fig 1 is analyzed using XOR technique. we have done vertical correlation and horizontal correlation and histogram for supporting the above statement.

- Histogram Analysis: Generally histogram shows in an image distribution of pixel values. The four images of encrypted images and histogram images are as shown in fig 2. Every histogram of an picture shows the estimation of how the pixels are distributed. Based on results of histogram of fig 2 it is clear that XOR operation is very useful to secure the selected picture.

- Adjacent Pixel Values: The adjacent pixel values of a meaningful image always carry nearby values. So when we scramble an image we always expect those nearby pixel values to be scattered.

- Information Entropy Analysis: Entropy is a process of measure randomness of pixels in an image. If the random distribution of pixel is low then we can say that the image is meaningful. But if the random distribution of an image is high then we can say that the image is less meaningful. We have calculated randomness of different original images and their cipher images. The visual values of picture pixel intensities are determined via histogram analysis. The histograms for the original image and the encrypted image are shown in Fig. 3, allowing us to observe how much the intensity values have been changed. To examine the compatibility of the color distribution between the plain picture and the cipher image, color histogram analysis is utilized. It may be argued that the cipher picture does not provide any hints to undertake a statistical attack on the encryption technique if the histogram value has a considerable distribution of diversity of the cipher image and also has a significant difference from the histogram of the plain image

3.4 Histogram Output for Output Image

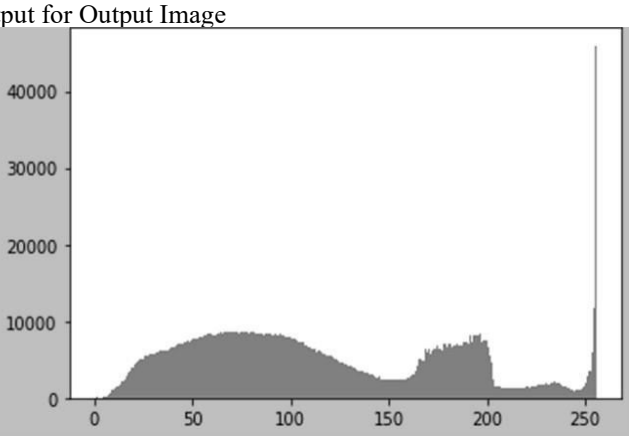


image mean: 118.06344217649648
image variance and std: 4464.0822151870725 66.81378761294013
histogram mean: 118.06344217649648
histogram variance and std: 4464.079084532487 66.81376418472834
histogram variance2 and std2: 4464.079084532486 66.81376418472833
normalized histogram mean: 0.4629938908882215
normalized histogram variance and std: 0.06865173524848147661.2687386
entropy: 5.325538336800033

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

normalized entropy: 0.9603909685706833 when the comparison comes to the Mean, Variance and Entropy values of both Input and Output image files, we obtain results as mentioned below:

Property	Value for Input Image	Value for Output Image
histogram mean	118.0634	118.0634
histogram variance	4464.0790	4464.0790
histogram entropy	5.3255	5.3255

CONCLUSION

This paper proposes an image encryption technique that makes use of the XOR operator. Based on the results XOR cipher is an important tool to encrypt an picture. When we use XOR cipher randomness of pixels of an original picture increase. Suppose randomness is higher , we can say that the picture is high secure. Using a key, the XOR operator is used to odd rows and columns of a picture to muddle the link between the original and encrypted images. Even rows and columns of the picture are treated with the same key that has been flipped. The suggested algorithm's resistance to many forms of assaults, including statistical and differential attacks, has been tested experimentally with comprehensive numerical analysis (visual testing). Furthermore, performance evaluation experiments show how extremely secure the suggested picture encryption technique is. Additionally, it has quick encryption and decryption capabilities, making it appropriate for real-time Internet encryption and transmission applications.

By analyzing histogram, horizontal and vertical correlation, information entropy it can be concluded that after scramble different images the randomness increases in their ciphered images, it means the ciphered images are more secure that it is not possible to decrypt. The security keys which are used to encrypt and decrypt the original images are same. In this process to encrypt an image XOR operation is performed between original image and security key. Then to decrypt the image again XOR operation is performed between encrypted image and Security key. So decryption using security key is totally dependent on the result after encryption. So we can conclude that this process increases security of an image.

DESIGN METHODOLOGY

3.1 Problem Definition:

Data security is a major concern in protecting sensitive images from unauthorized access. Images contain critical information in various domains, including healthcare, finance, military operations, and personal communications. Unauthorized access to confidential images can result in privacy breaches, financial losses, and security risks. Thus, safeguarding image data is essential in modern digital environments.

Encryption techniques help secure image data by converting it into an unreadable format, ensuring only authorized users can access it. While complex cryptographic methods like AES and RSA provide strong security, they often require significant computational resources. A simpler and more efficient alternative is XOR-based encryption, which offers a balance between security and performance.

XOR encryption is a symmetric technique, meaning the same key is used for both encryption and decryption. It functions by flipping bits based on a given key, making the original data unreadable. Applying the same key during decryption restores the original image. This approach is computationally efficient and easy to implement, making it ideal for resource-constrained environments such as IoT devices, mobile applications, and embedded systems.

Despite its advantages, XOR encryption has limitations, particularly its vulnerability to brute force attacks if the key is short or predictable. To enhance security, techniques such as dynamic key generation, key expansion, and multi-layered encryption can be employed.

This study aims to develop and evaluate a simple yet effective XOR-based encryption method for securing images. The research will analyze encryption speed, computational efficiency, and security robustness. By addressing privacy concerns and computational constraints, this study will contribute to lightweight encryption techniques, ensuring secure image transmission and storage across various applications.

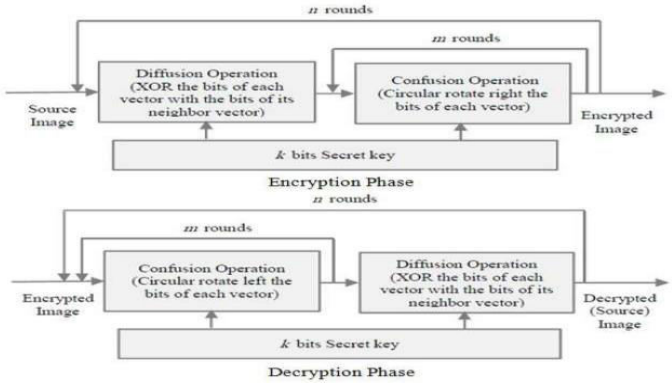
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|



The image represents a secure image encryption and decryption process using diffusion and confusion operations. It ensures high levels of security by making the encrypted image resistant to statistical and brute force attacks. The methodology consists of an encryption phase and a decryption phase, both employing multiple rounds of operations.
Encryption Phase

1.Source Image Input: The process starts with an input image that can be grayscale or color.
2.Diffusion Operation: This step involves modifying pixel values by performing an XOR operation between each vector and its neighboring vector. This process spreads the influence of each pixel over multiple pixels, increasing security.
3.Confusion Operation: After diffusion, a confusion step is performed using a circular right shift of the bits in each vector. This further obfuscates the original pixel values, making the encrypted image difficult to analyze.
4.Multiple Rounds: The diffusion and confusion operations are applied for n rounds to increase encryption strength.
5.Secret Key Application: A k-bit secret key is used to govern the encryption process, ensuring that only authorized users can decrypt the image.
6.Encrypted Image Output: The final output is a transformed, unreadable version of the original image.

Decryption Phase
1.Encrypted Image Input: The encrypted image serves as the input for decryption.
2.Confusion Operation: The circular shift applied during encryption is reversed by a left circular shift of bits in each vector.
3.Diffusion Operation: The original pixel values are restored using the same XORbased diffusion process but in reverse order.
4.Multiple Rounds: The decryption process mirrors the encryption phase, applying mmm rounds of reverse operations.
5.Secret Key Application: The same k-bit secret key is required for successful decryption.
6.Decrypted Image Output: The final result is a perfect reconstruction of the original image.
Security and Advantages
•Diffusion and Confusion Principles: The combination of XOR diffusion and bitwise confusion enhances security by making he encryption highly resistant to attacks.
•Robustness Against Attacks: The circular shifts and XOR operations significantly reduce vulnerabilities to statistical and brute force attacks.
•Applicability: The approach is suitable for secure image storage and transmission in high-security environments like finance, healthcare, and military applications.
This methodology ensures efficient and secure image encryption, balancing computational efficiency with strong security properties

3.3 Modules:
The 4 modules are:
3.3.1 User Interaction Module
The User Interaction Module provides an intuitive interface for users to upload images, input encryption keys, and view the encrypted or decrypted results. It ensures a seamless user experience by allowing easy navigation and interaction. The module

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

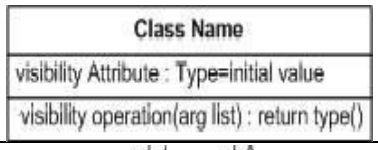
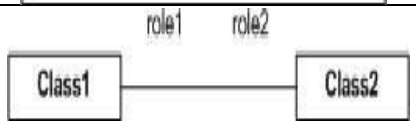
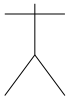
|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

List of UML Notations

Table 3.5.1 UML Notations

S.No	SYMBOL NAME	SYMBOL	DESCRIPTION
1	Class		Classes represent a collec on of similar en es grouped together.
2	Associa on		Associa on represents a sta c rela onship between classes.
3	Aggrega on		Aggrega on is a form of associa on. It aggregates several classes into single class.
4	Actor		Actors are the users of the system and other external en ty that react with the system.
5	Use Case		A use case is an interac on between the system and the external environment.
6	Rela on (Uses)		It is used for addi onal process communicaton.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

7	Communica on		It is the communica on between various use cases.
8	State	State	It represents the state of a process. Each state goes through various flows.
9	Ini al State		It represents the ini al state of the object.
10	Final State		It represents the final state of the object.
11	Control Flow		It represents the various control flow between the states.
12	Decision Box		It represents the decision making process from a constraint.
13	Component	Component	Components represent the physical components used in the system.
14	Node	Node	Deployment diagrams use the nodes for represen ng physical modules, which is a collec on of components.
15	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or ac on.
16	External En ty		It represent any external en ty such as keyboard, sensors etc.
17	Transi on		It represents any communica on that occurs between the processes.
18	Object Lifeline	Object	Object lifelines represents the ver cal dimension that objects communicates.
19	Message	Message	It represents the messages exchanged.

Use Case Diagram

The Use Case Diagram provides an overview of the interactions between different users and the system. The primary actors include the User, System Administrator, and Cloud Server. The user can upload an image, input an encryption key, and retrieve decrypted images. The system administrator ensures secure key management and system maintenance. The cloud server handles image encryption, decryption, and storage. This diagram helps in understanding the system's core functionalities and how users interact with them. users interact with them.

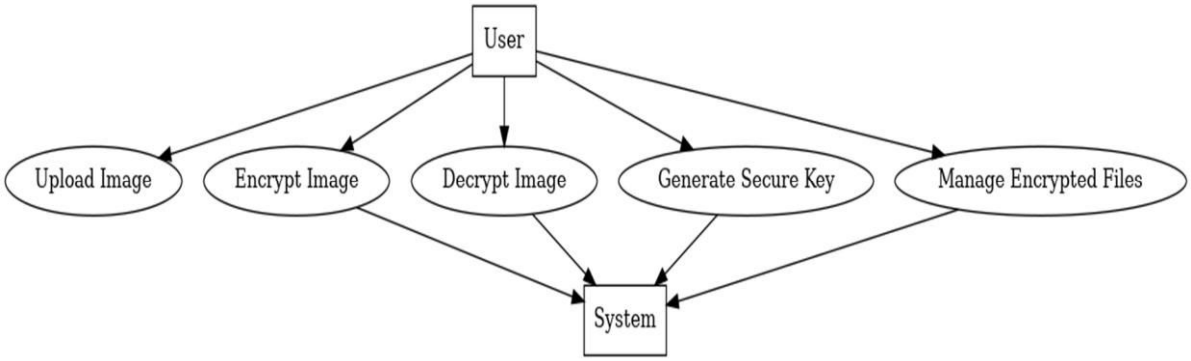
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|



The Use Case Diagram provides an overview of the interactions between different users and the system. The primary actors include the User, System Administrator, and Cloud Server. The user can upload an image, input an encryption key, and retrieve decrypted images. The system administrator ensures secure key management and system maintenance. The cloud server handles image encryption, decryption, and storage. This diagram helps in understanding the system's core functionalities and how users interact with them

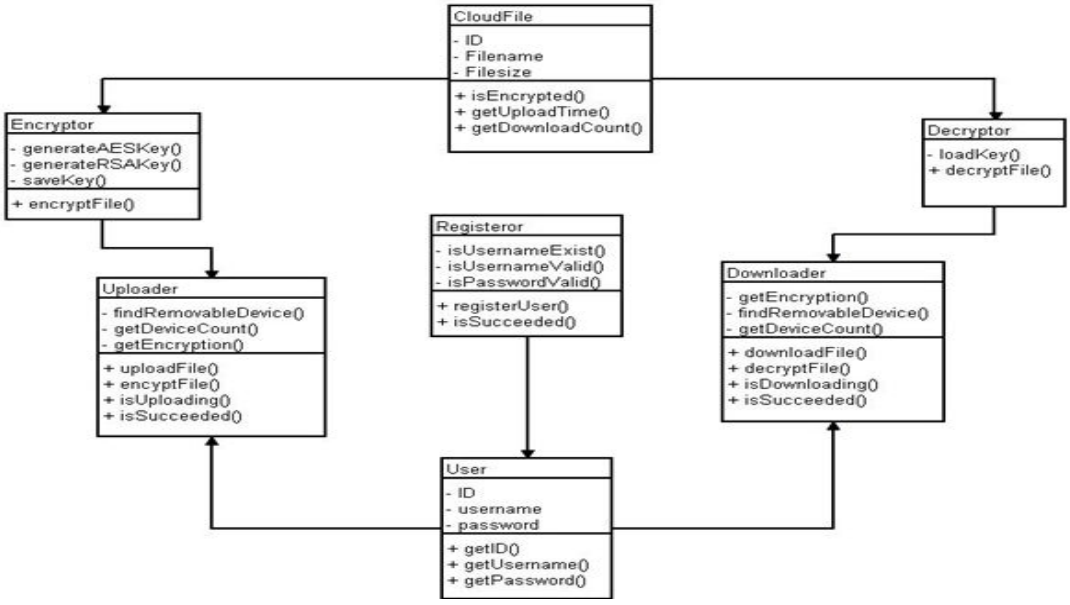


Figure 1: Flowchart of Encryption Algorithm at Sender side.

1 The Class Diagram defines the structural components of the image encryption system. It includes key entities such as User Interaction Module, Processing Module, File Management Module, Encryption & Decryption Module, and Key Generation & Management Module. Each class has specific attributes and methods to handle image input, encryption, key generation, and decryption. The relationships between these modules ensure smooth data flow, secure processing, and efficient storage management.

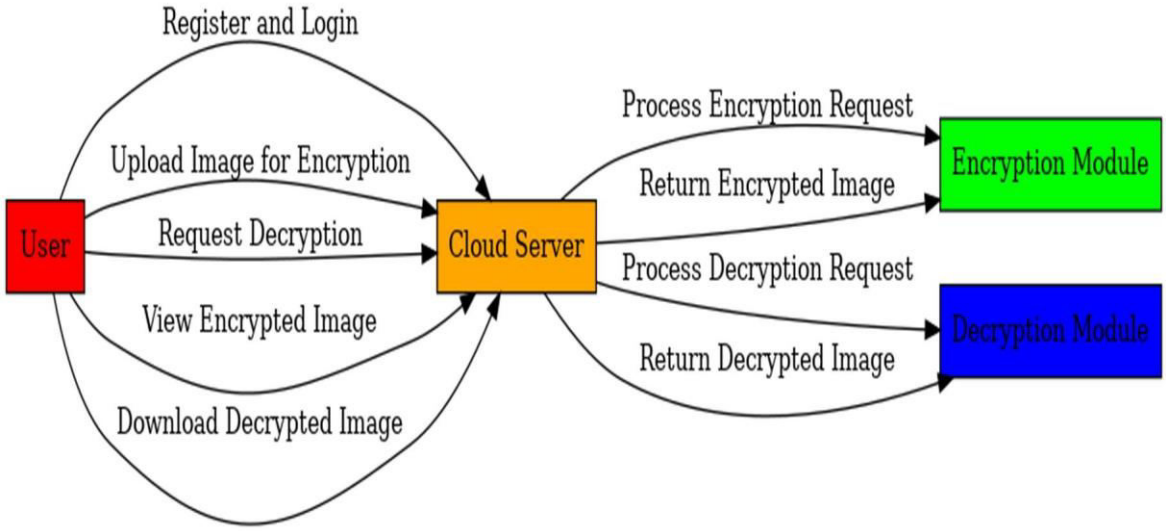
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

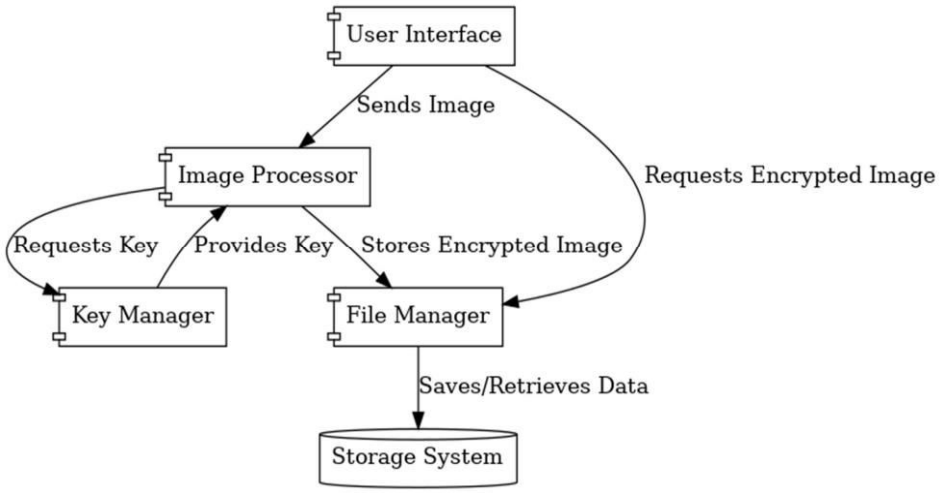
|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|



The Sequence Diagram depicts the interaction between various system components during encryption and decryption. It begins with the User uploading an image and entering an encryption key. The Cloud Server processes the request, applies security checks, and encrypts the image using the Key Management System. The encrypted image is stored securely, and the decryption follows the reverse sequence. This diagram helps visualize the step-by-step communication and processing of data within the system.



Component Diagram

Fig 3.6.4 Component Diagram

The Component Diagram outlines the structural organization of the system's main components and their dependencies. It consists of key modules such as User Interface Module, Encryption & Decryption Module, Key Management System, Storage System, and Processing Module. These components work together, where the User Interface collects input, the Encryption Module processes the data, and the Storage System securely manages files. The Key Management System ensures security through dynamic key generation and validation. This diagram provides a high-level view of the system's architecture and data flow.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

IV. SIMULATION RESULTS

Our Proposal we implement our algorithm in Mat lab 7.8.0 running on windows XP platform. we have used Seven 8-bit gray scale image of size 256×256.One such image of 8-bit gray image of Rose flower of size 256×256 is shown in figure 3. The encrypting process of the Image by taking the initial condition, the encrypted image hides the totality of the information contained therein, as seen in figure 3(a) and figure 3(c), Here the

Implementation is the process of defining how the information system should be built (i.e., physical system design), ensuring that the information system is operational and used, ensuring that the information system meets quality standard.

4.1 TECHNOLOGIES

The technologies used in developing the project are python, Exception Handling, File Handling and Bytearray Manipulation.

4.1.1 Python

Python is a high-level, interpreted programming language known for its simplicity and readability. It supports multiple paradigms, including object-oriented, procedural, and functional programming. With a vast ecosystem of libraries, Python is widely used in web development, data science, automation, and more.

4.1.2 History of Python

Here's a brief history of Python:

Python was created by Guido van Rossum in the late 1980s and officially released in 1991. It was designed as a successor to the ABC language, focusing on simplicity and readability. Over the years, Python has evolved through multiple versions, with major releases like Python 2 (2000) and Python 3 (2008), bringing significant improvements. Today, Python is widely used in web development, data science, AI, and automation, making it one of the most popular programming languages.

•Dynamically Typed – No need to declare variable types; Python automatically detects them.
•Object-Oriented & Multi-Paradigm – Supports object-oriented, procedural, and functional programming.
•Extensive Libraries – Rich ecosystem with libraries like NumPy, Pandas, TensorFlow, Flask, and Django.
• Platform Independent – Python code runs on multiple operating systems without modification.
•Automatic Memory Management – Handles memory allocation and garbage collection efficiently.
•Scalability & Extensibility – Easily integrates with C, C++, Java, and other languages.
•Wide Application Areas – Used in web development, data science, AI, automation, and cybersecurity.
•Large Community Support – A vast global community provides continuous development and support.

4.1.4 Exception Handling

4.1.5 History of Exception Handling

Exception handling evolved from basic error-checking methods in early programming languages like Fortran and Assembly (1950s-60s), which relied on manual error codes. PL/I (1964) and Ada (1980) introduced structured exception handling. C++ (1985) introduced try, catch, and throw, while Java (1995) popularized checked and unchecked exceptions

Modern languages like Python and C# refined exception handling with features like finally blocks and custom exceptions. Today, exception handling is a core part of programming, ensuring robustness and error recovery.

4.1.6 Features of Exception Handling

•Error Detection & Handling – Identifies runtime errors and prevents program crashes.
•try-except Mechanism – Allows handling errors using try, except (or catch in other languages).
•finally Block – Ensures execution of essential code (e.g., closing files, releasing resources) regardless of errors.
•Custom Exceptions – Enables defining user-specific exceptions for better error categorization.
•Propagation & Stack Unwinding – Errors can propagate up the call stack until caught.
•Logging & Debugging – Supports logging error details for easier debugging and troubleshooting.
•Graceful Program Termination – Prevents abrupt crashes and allows controlled recovery.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

•Multiple Exception Handling – Allows handling different types of exceptions separately.
•Performance Optimization – Reduces the need for excessive conditional statements for error checking.
•Cross-Language Support – Found in modern languages like Python, Java, C++, C#, and JavaScript.

4.1.7 Steps for Execution of Exception Handling
1.Identify Risky Code – Determine which part of the code might raise an exception.
2.Write a Try Block (try) – Place the risky code inside a try block to monitor for exceptions.
3.Catch Exceptions (except or catch) – Define one or more except blocks to handle specific exceptions.
4.Use Multiple Exception Handling (Optional) – Handle different types of exceptions separately for better debugging.
5.Execute Cleanup Code (finally) – Use a finally block to run essential cleanup actions like closing files or releasing resources.
6.Raise Custom Exceptions (Optional) – Use raise to generate user-defined exceptions when necessary.
7.Log and Debug Errors (Optional) – Store exception details for future debugging and troubleshooting.
8.Ensure Program Continuation – Prevent crashes and allow the program to recover or exit gracefully.

1. Crypto (for Random Key Generation)
The crypto module in Node.js is a built-in library that provides cryptographic functionalities, enabling secure encryption and decryption. It is commonly used to generate random keys, hash passwords, and secure sensitive data. The module includes various encryption algorithms such as AES, RSA, and SHA, making it a reliable choice for data security applications. Developers use crypto.randomBytes() to create secure random values, essential for cryptographic operations like token generation and data encryption.

2. FS (for File Handling)
The fs (File System) module in Node.js allows interaction with the file system, enabling reading, writing, updating, and deleting files. It provides both synchronous and asynchronous methods to handle file operations efficiently. Developers use fs.readFile() to read file contents, fs.writeFile() to store data, and fs.appendFile() to modify existing files. The module supports handling various file formats, including text, JSON, and binary files, making it an essential tool for backend development.

3. Buffer (for Binary Data Handling)
The buffer module in python is used to manage binary data, especially when dealing with file streams, network communications, or image processing. Unlike regular python strings, buffers allow efficient handling of raw data in memory. Buffer.alloc() is used to create a new buffer, while Buffer.from() converts existing data into buffer format. This module is particularly useful when working with encrypted data, image files, or network protocols requiring direct binary manipulation.

V. CONCLUSION

The image encryption and decryption algorithm is designed and implemented to provide confidentiality and security in transmission of the gray image based data as well as in storage. The scheme presented in this paper has a simple implementation module. The proposed encryption algorithm can ensure multiple criteria such as lossless, maximum distortion, maximum performance and maximum speed. The proposed encryption method in this study has been tested on different gray images and showed good results. Future work will be focused on the development of this algorithm to get color image.

ACKNOWLEDGMENT

The author would like to express their thanks & gratitude to all those who gave suggestions and valuable contributions to this work. The authors would also like to thank the reviewers for their comments to improve this paper. Our special thanks to Prof Vineet Richaria, Head, Computer Science and Engg Dept.,LNCT, Bhopal, India. And Asst Prof Vijay Kumar Trivedi Computer Science and Engg Dept., LNCT, Bhopal, India whose help, stimulating suggestions, experience and encouragement helped us in all the times of study and analysis of this work.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404438|

REFERENCES

[1]. P. Sharma, D. Mishra, V.K. Sarthi, P. Bhatpahri and R. Shrivastava, "Visual Encryption Using Bit Shift Technique", International Journal of Scientific Research in Computer Science and Engineering, Vol. 5, No. 3, pp. 57-61, June 2017.

[2]. XY Wang, YQ Zhang and LT Liu, "An enhanced sub-image encryption method", Optics and Laser in Engineering, Vol. 86, pp. 248-254, November 2016

[3]. CC Lee, HH Chen, HT Liu, GW Chen and CS Tsai, "A new visual cryptography with multi-level encoding", Journal of Visual Languages and Computing, Vol. 2,No. 3, pp. 243-250, June 2016.

[4]. HB Kekre, T Sarode and P. Halarnkar, "Image Scrambling using RPrime Shuffle", International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, Vol. 2, No. 8, pp. 4070 – 4076, August 2013.

[4]. Krishnamurthy, O. (2024). Impact of Generative AI in Cybersecurity and Privacy. International Journal of Advances in Engineering Research, pp. 27, 26–38. https://ijaer.com/admin/upload/04%20Oku%20Krishnamurthy%2001436.pdf.

[5]. Z. Hua and Y. Zhou, "Design of image cipher using block-based scrambling and image filtering", Information Sciences, Vol. 396, pp. 97-113, August 2017.

[6]. T Guo, F Liu and C. Wu, "k out of k extended visual cryptography scheme by random grids", Signal Processing, Vol. 94, pp. 90-101, January 2014.

[7]. A. Soleymani, M. J. Nordin and E. Sundarajan, "A Chaotic Cryptosystem for Images Based on Henon and Arnold Cat Map", The Scientific World Journal, 2014. DOI:-http://dx.doi.org/10.1155/2014/536930.

[9]. Pitkar, H., Bauskar, S., Parmar, D. S., & Saran, H. K. (2024). Exploring model-as-a-service for generative ai on cloud platforms. Review of Computer Engineering Research, 11(4), 140-154.

[8]. X.Y. Wang, F. Chen and T. Wang, "A new compound mode of confusion and diffusion for block encryption of image based on chaos", Communications in Nonlinear Science and Numerical Simulation, Vol. 15,No. 9, pp. 2479-2485, September 2010.

[9]. L. Sui and B. Gao, "Single-channel color image encryption based on iterative fractional Fourier transform and chaos", Optics & Laser Technology, Vol. 48, pp. 117-127, June 2013.

[10]. H. Li, Y. Wang, H. Yan, L. Li, Q. Li and X. Zhao, "Double-image encryption by using chaos-based local pixel scrambling technique and gyrator transform", Optics and Lasers in Engineering, Vol. 51, No. 12, pp. 1327-1331, 2013.

[11]. A. Kanso, M. Ghebleh, "A novel image encryption algorithm based on a 3D chaotic map", Communications in Nonlinear Science and Numerical Simulation, Vol. 17, No. 7, pp. 2943-2959, July 2012.

[12]. C.Y. Song, Y.L. Qiao and X.Z. Zhang, "An image encryption scheme based on new spatiotemporal chaos", Optik, Vol. 124, No.18, pp. 3329-3334, September 2012.

[13]. N. Singh and A. Sinha, "Optical image encryption using improper Hartley transforms and chaos", Optik, Vol. 121, No. 10, pp. 918- 925, June 2010.

[14]. A. Akhshani, S. Behnia, A. Akhavan, H.A. Hassan and Z. Hassan, "A novel scheme for image encryption based on 2D piecewise chaotic maps", Optics Communications, Vol. 283, No. 17, 3259- 3266, September 2010.

[15]. X. Y. Wang, Y.Q Zhang, and L.T. Liu, "An enhanced sub-image encryption method", Optics and Laser in Engineering, Vol. 86, pp. 248-254, November 2016.

[16]. B. Stoyanov, and K. Kordov, "Image Encryption Using Chebyshev Map and Rotation Equation", Entropy, Vol. 17, pp. 2117-2139, April 2015.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details