Indian Journal of Engineering, Management and Technology

https://doi.org/XX.XXXX/ijems.XXXX

Research Article

Multilevel Image Encryption and Decryption Using Image Block Shuffling and Pixel Value Alternation

Honnaraju B.[a], Murali S.[b]

[a]*Department of Computer Science and Business Systems, Maharaja Institute of Technology Mysore, Karnataka-571477, India,*
[b]*Department of Computer Science and Engineering, Maharaja Institute of Technology Mysore, Karnataka-571477, India,*
Email: honnaraju.gowda@gmail.com; murali@mitmysore.in

Received: 27-05-2025, *Accepted*: 20-09-2025, *Published*: 28-09-2025

Abstract

Securing image information has become a key concern with the rapid development of digital data transmission and storage. With the increasing use of images and videos in various fields such as social media, defense, and surveillance systems, it is important to protect personal and sensitive image data from unauthorized access. Image security has thus become a critical challenge. Protecting important and personal images is essential, and many methods have been investigated and developed to preserve data and private information. Among them, image encryption is one of the most widely used techniques to secure image data from unauthorized access. In this paper, an image encryption approach based on pixel value rotation and text file creation is proposed. First, the image size and image name are extracted. Then, the pixel values of each component (R, G, B) are obtained and rotated by a certain number of times. Finally, the rotated pixel values (R, G, B) are stored in a text file. Experiments have been conducted on various sets of images to validate the method.

Keywords: Encryption, Decryption, Rotation, Extraction, Password.

1. Introduction

In recent times, securing image information has become one of the most critical tasks. Nowadays, almost all information is transmitted over computer and mobile networks, which has increased the risk of network attacks and misuse of data. To ensure secure transmission, image data must be encrypted before being sent, and the encrypted image should be stored in such a way that it cannot be easily accessed or extracted by unauthorized attackers. Encryption is the process of converting original data into an unreadable form, where the original pixel values of an image are transformed into cipher text values. Various encryption methods exist that use keys composed of diverse characters for both encryption and decryption. Using these keys, the original data is transformed into cipher text, and decryption performs the reverse process—converting cipher text back to the original data. Image encryption specifically converts image data into an unreadable format, ensuring that only authenticated users with the correct key can reconstruct and display the original image.

This technique is a highly effective method for securing important and sensitive image information, as it relies on mathematical algorithms for both encryption and decryption processes.

2. Literature Review

A large number of articles have focused on different approaches for scrambling image contents, either with or without the use of encryption algorithms. Since the main emphasis of this paper is based on pixel-value rotation, several related works are highlighted here. R. Li et al. [2] employed two chaotic maps: one for scrambling the pixel positions of color and grayscale images, and the other for confusing the relationship between the encrypted and original images. In [3], Arnold's map algorithm was applied as a scrambling method to change the coordinates of the original image pixels into new ones. K. U. Shahna et al. [4] proposed a method combining pixel-level scrambling with bit-level scrambling on the original image, in addition to applying DNA operations (coding, XOR processes, and complementary rules) to enhance performance and security. A similar idea of combining pixel and bit scrambling was presented in [5], where SCAN (a method for visiting all image pixels through different paths) and cyclic shift operations were used to modify both pixel values and positions. To ensure high sensitivity of the encrypted image to any pixel change in the original image, pixel values in this work were determined based on three parameters: the old pixel value, a key stream element, and an unknown secret value. In [6], a chaotic algorithm for image encryption was introduced, where the image was divided into several sub-images and scrambled in three stages. The first stage involved bit-level scrambling of the sub-images to reduce adjacency between pixels.

3. Proposed Method

In the proposed system, the user first provides an image along with a password. The method consists of two main phases:

1. Image Encryption Phase
2. Image Decryption Phase

The image encryption phase is carried out in several stages:

- i Determine the number of pixel rotations, R, from the given password, and identify the values required for shuffling the image blocks.
- ii Extract the name and size of the input image file.
- iii Create a text file with the same name as the image file.
- iv Store the file name, image properties, password, and image size in the text file.
- v Read each pixel value of the image and extract its R, G, and B components.
- vi Split the R, G, and B components into multiple blocks and shuffle the blocks both horizontally and vertically.
- vii Rotate each pixel component value by R rotations.
- viii Swap the R and B components.
- ix Store the rotated values in the text file.

A detailed description of the encryption process (Phase 1) is provided in the following subsection. Figure 1 illustrates the block diagram of the image encryption phase.

a) Finding of pixel rotation value and block shuffling value

In this step, the individual digit extraction process is performed. Digit extraction refers to separating each digit from a numerical value, which can be achieved using mathematical operations such as modulus and division. Before applying digit extraction, the given password is first converted into its ASCII equivalent values. Each ASCII value is then multiplied by 123, and the sum of all resulting values is computed. The sum obtained is subsequently used in the digit extraction process. Ex:

Password: ABCDEFG, ASCII equivalent= 65 66 67 68 69 70 71 72

Sum of all elements: (65 * 121) + (66 * 121) + (67 * 121) + (68 * 121) + (69 * 121) + (70 * 121) + (71 * 121) = 57596

57596 is the value for pixel rotation. Each digit from this number will be extracted and used for pixel value rotation. Individual digit extracted and the digit will be stored in the array for further process.

b) Find the characteristics properties of the image.

Image characteristics refer to the different properties that describe an image's appearance, structure, and quality. Analyzing these characteristics is important for evaluating the differences between the original and the decrypted image. In this work, the following image properties are calculated: i. Contrast, ii. Brightness, iii. Moment, iv. Skewness, v. Kurtosis, vi. Spatial Frequency.

Contrast: Contrast is the difference in luminance that enables objects within an image to be distinguished from one another. A higher contrast enhances visibility of details, whereas lower contrast makes the image appear flat and less distinct.

$$Contrast = \frac{1}{mn}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}[f(i,j) - M]^2}$$

Where M is mean and f(i, j) is input image.

Brightness: A measure of the illumination of the image estimated by the global mean of the image.

$$Brightness, M = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)}{m\ x\ n}$$

Spatial Frequency: Measure of the overall activity level in an image. Spatial frequency is computed as : SF = RF2 + CF2

RF and CF are the row and column frequency

$$RF = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=2}^{N}(f(i,j) - f(i,j-1))^2}$$

$$CF = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=2}^{N}(f(i,j) - f(i-1,j))^2}$$

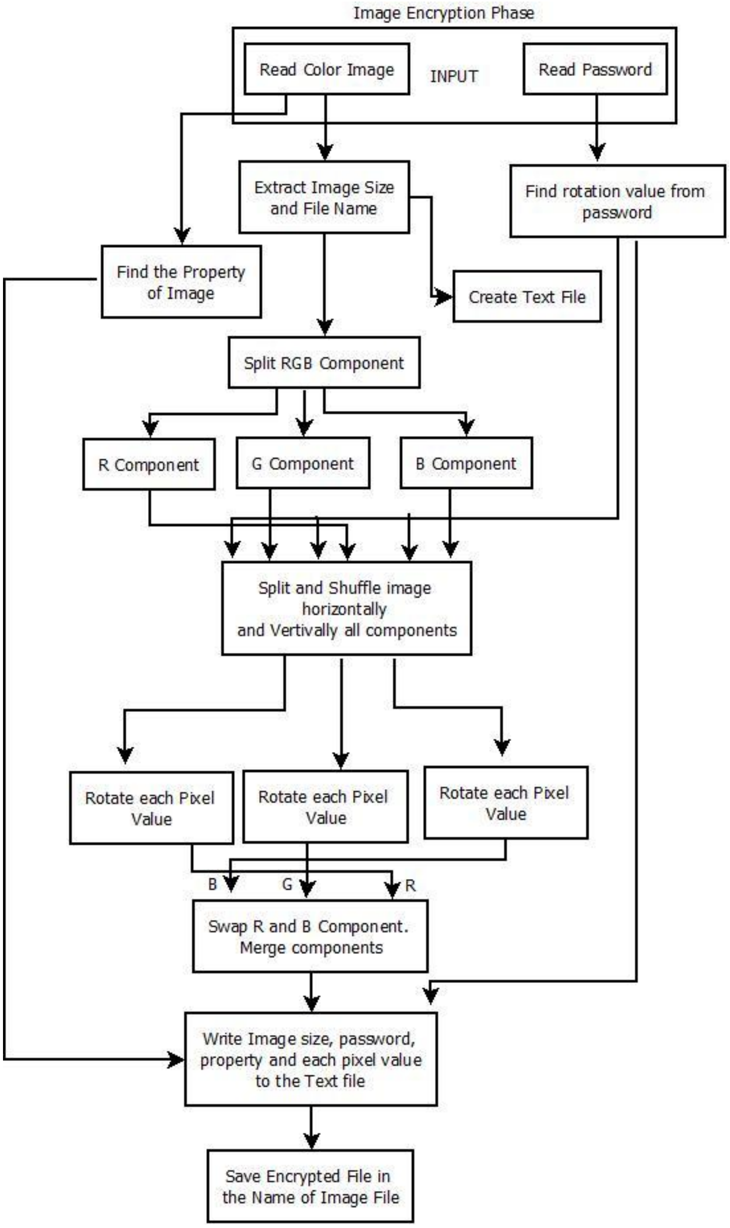
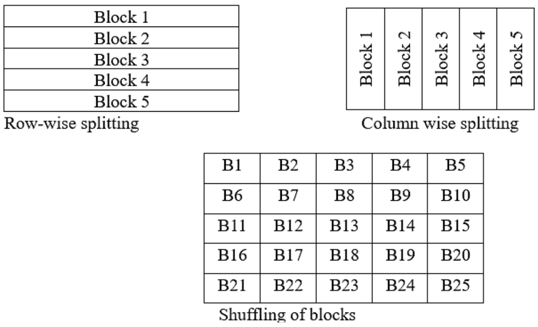


Image Encryption Phase
Read Color Image
INPUT
Read Password
Extract Image Size and File Name
Find rotation value from password
Find the Property of Image
Create Text File
Split RGB Component
R Component
G Component
B Component
Split and Shuffle image horizontally and Vertivally all components
Rotate each Pixel Value
Rotate each Pixel Value
Rotate each Pixel Value
B
G
R
Swap R and B Component. Merge components
Write Image size, password, property and each pixel value to the Text file
Save Encrypted File in the Name of Image File

Fig. 1: Image Encryption Phase

c. Split and Shuffling of R, G, and B Image Blocks:

In this step, the RGB channels of the image are first separated and processed individually, which is a common approach in image processing. Each of the R, G, and B components is divided into blocks of equal size. The splitting process is applied consistently across all three channels to ensure uniformity. The number of blocks is determined based on the digits of the user-provided password. Specifically, a mapping mechanism is defined in which the length or digits of the password are used to calculate the block size or the number of blocks for each channel. Once the blocks are formed, they are shuffled both horizontally and vertically to increase randomness and reduce correlation among adjacent pixels.



Row-wise splitting

Column wise splitting

Shuffling of blocks

d. Pixel value rotation based on the password: Each pixel in a digital image is represented by three values: the Red (R), Green (G), and Blue (B) components. Each component typically ranges from 0 to 255, and together they define a specific color based on the RGB color model.

In the proposed encryption step, the intensity value of each pixel component is transformed within the 0–255 range. This transformation is governed by the digits extracted from the password [1], ensuring that the modification pattern is dependent on the secret key. As a result, the pixel values are altered in a way that makes the encrypted image significantly different from the original, thereby enhancing security.

$\text{Rotated Value}=(\text{Pixel} \gg \text{Di})$

Here, Pixel denotes the intensity value of the pixel, and Di represents the number of right rotations applied to it. Reverse operation will be performed during the decryption process.

After performing the pixel value rotation, the modified pixel values are written to a text file. The file begins with metadata information, including the image file name, the width of the image, and the password (converted into its ASCII values). This is followed by the sequentially stored encrypted pixel values. Finally, the height of the image is appended at the end of the file.

In the decryption phase, the original image is reconstructed from the encrypted text file. The process is carried out through the following steps:

Read the encrypted text file corresponding to the image.

Extract the metadata, including the image file name, image size, image properties, and the password used during encryption.

Retrieve each encrypted pixel value from the file and store them in separate 2×2 matrices for the R, G, and B components.

Perform horizontal and vertical shuffling of the image matrices based on the password.

For each pixel in the R, G, and B matrices, apply reverse rotation using the password-derived value.

Swap the R and B components back to their original positions.

Merge the R, G, and B components to reconstruct the final image file.

The complete decryption process is illustrated in Fig. 2.

a. Extraction of Information from the Encrypted Text File: Extracting the password and image dimensions is a crucial part of the decryption process. If either the extracted password or the image dimensions are incorrect, the decryption will fail, and the original image cannot be reconstructed. The decryption process begins by extracting the image file name, followed by the width and height of the image. Notably, the height value is stored at the end of the text file and must be retrieved from there.

b. Decryption Process: Once the password and image dimensions are correctly extracted, the decryption is performed in a manner similar to the encryption process. This includes block shuffling and reverse pixel value rotation, both of which are governed by the password-derived parameters. These steps restore the encrypted pixel values to their original form, after which the R, G, and B components are merged to reconstruct the image.

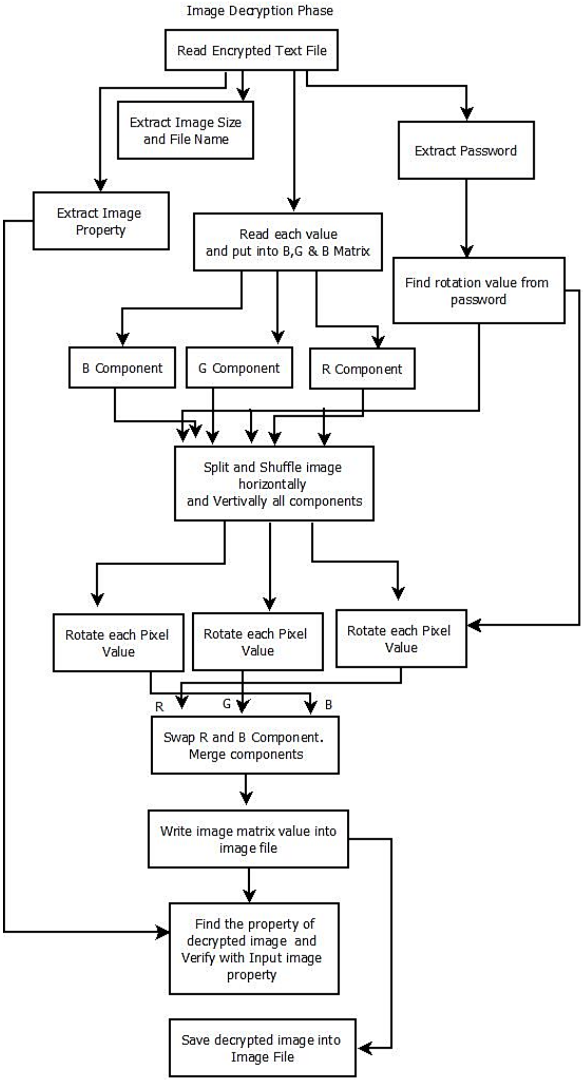


Image Decryption Phase
Read Encrypted Text File
Extract Image Size and File Name
Extract Password
Extract Image Property
Read each value and put into B,G & B Matrix
Find rotation value from password
B Component
G Component
R Component
Split and Shuffle image horizontally and Vertivally all components
Rotate each Pixel Value
Rotate each Pixel Value
Rotate each Pixel Value
R
G
B
Swap R and B Component. Merge components
Write image matrix value into image file
Find the property of decrypted image and Verify with Input image property
Save decrypted image into Image File

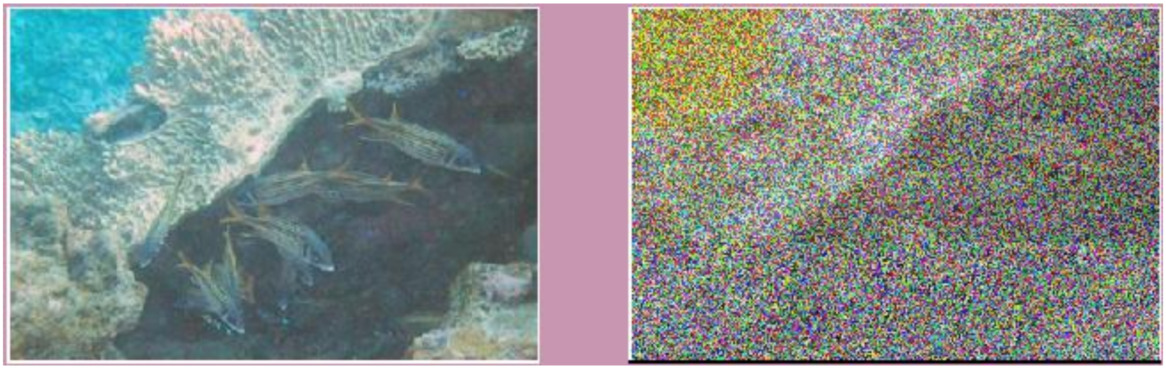
Fig 2: Image decryption process

4. Results and Analysis

The figure below illustrates the output of the encryption and decryption processes. The proposed approach is designed to work with different types of images while ensuring that decryption remains highly challenging without the correct key. In this method, the encrypted data is stored in a text file, which makes it difficult for an attacker to directly identify image data. Furthermore, it becomes challenging to determine whether the stored values correspond to image pixels or to some other type of information.

Even if an attacker successfully identifies the password and image dimensions, the operations performed to extract the digits remain unknown. The block-shuffling process requires **2^n** and **2^n** operations for row-wise and column-wise shuffling, respectively. Similarly, pixel value rotation within each block requires **2^n** operations. Due to these multiple layers of transformation, decrypting the image from the encrypted text file without the proper key remains computationally infeasible.

Metrics such as contrast, brightness, skewness, kurtosis, and spatial frequency are used to analyze differences between original and decrypted images. The decrypted images show values nearly identical to the originals, demonstrating lossless reconstruction. On the other hand, encrypted images display significantly altered statistical characteristics, confirming that the proposed method effectively conceals image information.



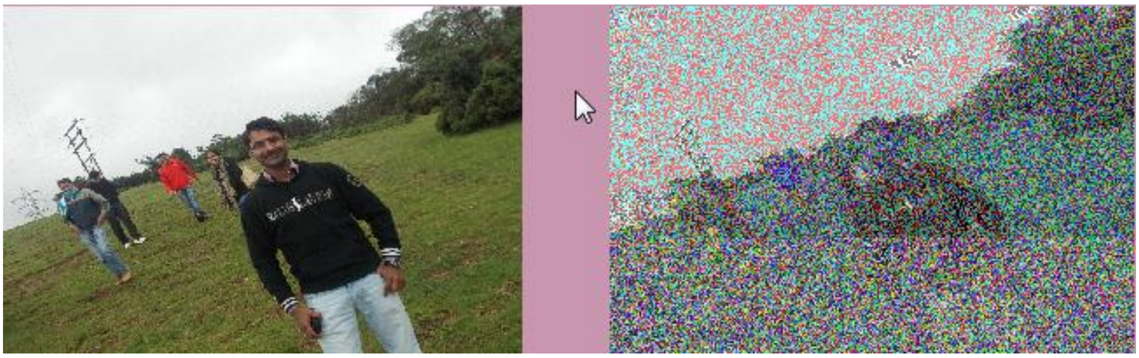


Figure 3: Output of Image Encryption

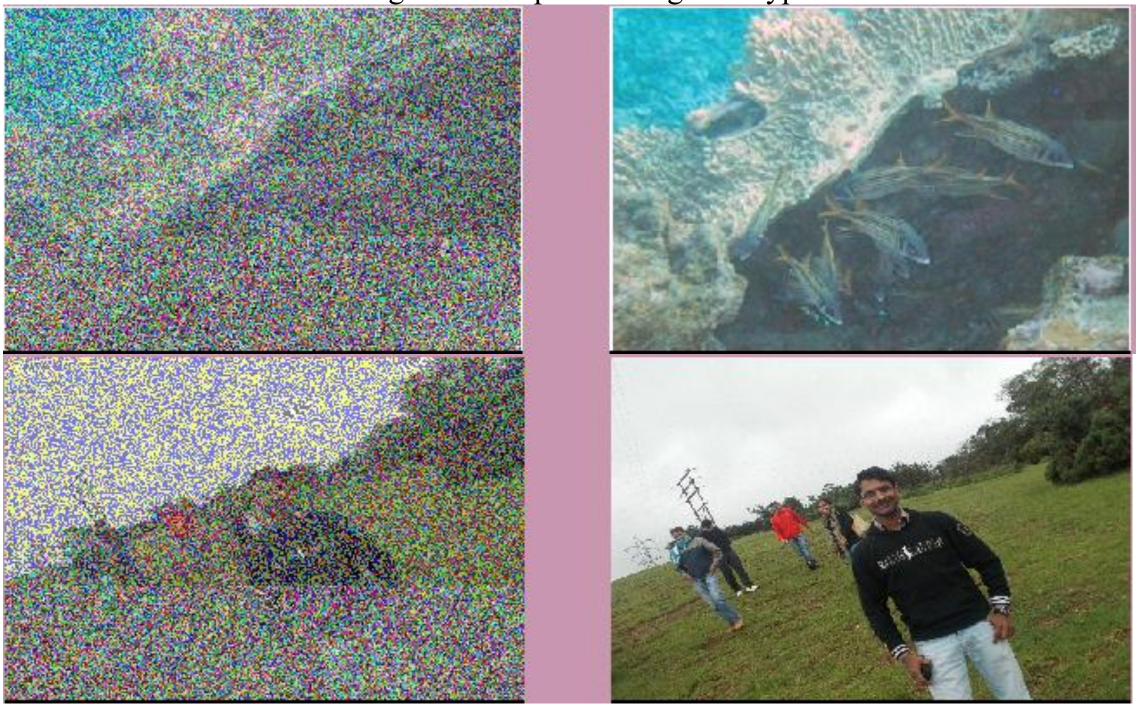


Figure 4: Output Image decryption.

5. Conclusion

The proposed encryption method significantly enhances image security by making unauthorized decryption highly impractical. Since the encrypted pixel values are stored in a text file, it becomes extremely difficult for attackers to recognize or interpret the underlying image data. Even if the password and image dimensions are compromised, the combination of complex operations—such as block shuffling and pixel value rotation—makes it nearly impossible to accurately reconstruct the original image. Moreover, the computational complexity introduced by these operations further strengthens security, providing robust protection against brute-force and analytical attacks. This approach is effective for securing images of various types and formats, thereby establishing it as a reliable and versatile method for image encryption.

References

1. Honnaraju B, “Image Encryption by using pixel value rotation”, Presented in the International Conference on Electrical Engineering and Computer Science” at Trivandrum on 12th May 2012
2. R. Li, Q. Liu and L. Liu, “Novel image encryption algorithm based on improved logistic map,” IET Image Processing, vol.13,no.1, pp.125-134, 2019. doi: 2009. https://doi.org/10.1049/iet-ipr.2018.5900.
3. S. Sun, “A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling,” IEEE Photonics Journal, vol.10,no.2, pp.1-14,2018. https://doi.org/10.1109/JPHOT.2018.2817550.

4. K.U. Shahna and A. Mohamed, “A novel image encryption scheme using both pixel level and bit level permutation with chaotic map,” Applied Soft Computing, vol.90, p.106162,2020. https://doi.org/10.1016/j.asoc.2020.106162.
5. X. Wang, N. Guan and J. Yang,”Image encryption algorithm with random scrambling based on one-dimensional logistic self-embedding chaotic map,” Chaos, Solitons & Fractals, vol.150, p.111117, 2021. https://doi.org/10.1016/j.chaos.2021.111117.
6. S. Anwar and S. Meghana, “A pixel permutation based image encryption technique using chaotic map,” Multimedia tools and applications, vol.78,no.19, pp.27569-27590, 2019.
7. Y. Pourasad, R. Ranjbarzadeh and A. Mardani, “A new algorithm for digital image encryption based on chaos theory,” Entropy, vol.23,no.3, p.341, 2021.
8. A. Alghafis, N. Munir, M. Khan and I. Hussain, “An encryption scheme based on discrete quantum map and continuous chaotic system,” International Journal of theoretical physics, vol.59,no.4, pp.1227-1240, 2020. https://doi.org/10.1007/s10773-020-04402-7.