DYNAMIC DATA STRUCTURE-BASED IMAGE SECURITY METHOD

By

Swapan Kumar Shee

(B.Tech 3rd Year CSE, Enrollment No.:12021002001009)

Shivam Kumar Mishra

(B.Tech 3rd Year AIML, Enrollment No.:12021002001042)

Under the Supervision of

Prof. Jyoti Khandelwal

Dept. of Computer Science & Engineering

University of Engineering & Management, Jaipur

Outlines

- Introduction
- Literature Review
- Objectives
- Proposed Method
- Experimental Setup
- Experimental Results
- Conclusion & Future Scope
- Reference
- Acknowledgement

Introduction

- The goal is to devise a method for scrambling images for visual incomprehensibility while maintaining structural integrity, alongside creating an efficient descrambling mechanism for image reconstruction.
- The report will delve into the theoretical foundations of the methodology, including binary tree construction, hash table implementation, and their application to image manipulation.
- Detailed implementation strategies will be discussed, covering image preprocessing, data structure instantiation, and algorithmic optimization.
- The project aims to contribute to image security and privacy by offering a pragmatic solution grounded in mathematical rigor and practical utility, with the intention of fostering further research and development in the field.

Literature Review

Serial No.	Authors	Topic	Conclusion
1.	Wu, Y., Xiang, Y., Guo, Y., Tang, J., & Yin, Z. (2020) [1]	An improved reversible data hiding in encrypted images using parametric binary tree labeling	The IPBTL-RDHEI method effectively enhances data hiding capacity in encrypted images while ensuring reversible extraction without loss.
2.	Abbas Fadhil Al-Husainy, M., A. A. Al-Sewadi, H., & Al-Shargabi, B. (2022) [2]	Image encryption using a binary search tree structure-based key	BST-based image encryption algorithm demonstrates robust security with flexible key lengths and comparable performance to AES and DES in various metrics.

Literature Review(Cont.)

Serial No.	Author	Topic	Conclusion
3.	Gopalakrishnan T. (2018) [3]	Image Encryption Using Multiple Chaotic Maps In Hybrid Domain	Utilizing different keys per round and an internal key generation process enhances image security against chosen plain text.
4.	Wang, X., & Liu, L. (2020) [4]	Image encryption based on hash table scrambling and DNA substitution.	An image encryption algorithm employing hash table scrambling, DNA substitution, and hyper-chaotic Chen system-generated pseudo-random sequences enhances security with high key sensitivity.

Literature Review(Cont.)

Serial No.	Author	Topic	Conclusion
5.	Al-Husainy, M. A., & Al-Sewadi, H. A. (2019) [5]	Implementing binary search tree concept for image cryptography.	The proposed symmetric image encryption method utilizing a binary search tree mechanism for key generation achieves high security with flexible key length, fulfilling Shannon's diffusion and confusion concept
6.	K. CHITRA. (2019) [6]	Enhancement of image security using improved visual cryptography visual cryptography and digital watermarking techniques.	Visual Cryptography concept facilitates secure secret sharing among participants via secret image distribution, with the (2, 2) sharing scheme and diagonal sharing pattern demonstrating efficient runtime and high security, particularly suited for PNG file format.

Objectives

- To design a robust method for image manipulation that could effectively conceal the content of digital images while enabling accurate reconstruction.
- To exploit the inherent advantages of Binary Tree and Hash Table data structures to achieve efficient image scrambling and descrambling.
- To Implement the developed algorithms and conduct comprehensive evaluations to assess their performance.
- To Design an algorithm capable of accurately reconstructing the original image from its scrambled counterpart, thus ensuring fidelity in the descrambling process.

Proposed Method

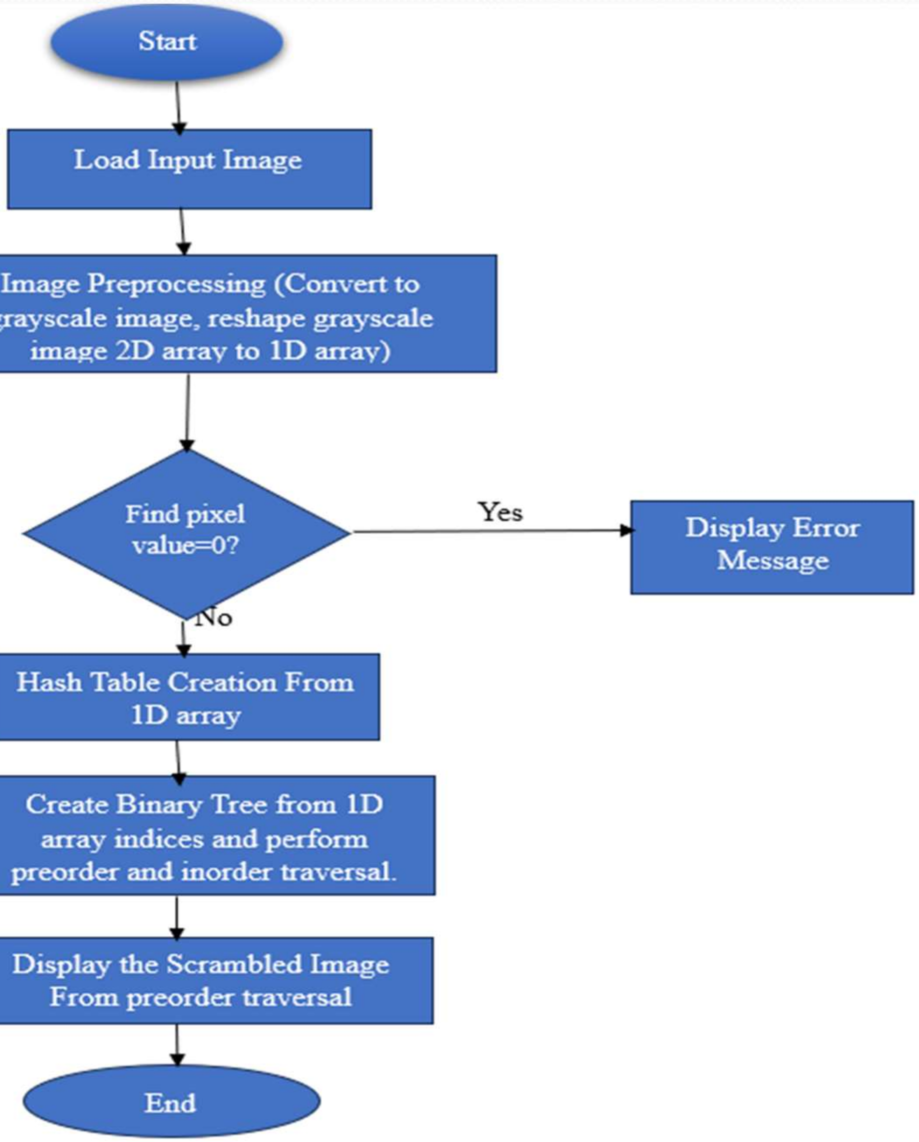


Fig .1 Proposed model for Scrambling.

Proposed Method(Cont.)

Proposed Image Scrambling Algorithm:

Input: Input Image(S).

Output: Scrambled Image (SS1).

Begin

1. Read the image(S) and Convert the RGB image to grayscale using "rgb2gray". And then Reshape the 2D grayscale image into a 1D array using reshape.
2. Check if the 1D array contains any zeros. If zeros are present, display an error message indicating the image cannot be scrambled and exit the program; otherwise, proceed to step 3.
3. Initialize an empty hash table with 255 cells.
   - Iterates through each pixel value (1-255):
   - Find occurrences and either create a new entry with occurrences if cell empty or chain with linked list if collision.
4. Create a binary tree using recursive helper function construct_tree_helper.
   - Helper takes 1D array, start/end indices.
   - If start > end, return empty node.
   - Else: Create node with value at start. call helper for left child and right child.
5. Perform an in-order traversal of the binary tree with a recursive function inorder.
6. Perform a pre-order traversal of the binary tree with a recursive function preorder.
7. Reshape the in-order and pre-order traversal lists back to the original 2D image dimensions using reshape and save the reshaped arrays as separate ". MAT" as a key.

Return scrambled image SS1 from reshaped 2D pre-order traversal.

End

Proposed Method(Cont.)

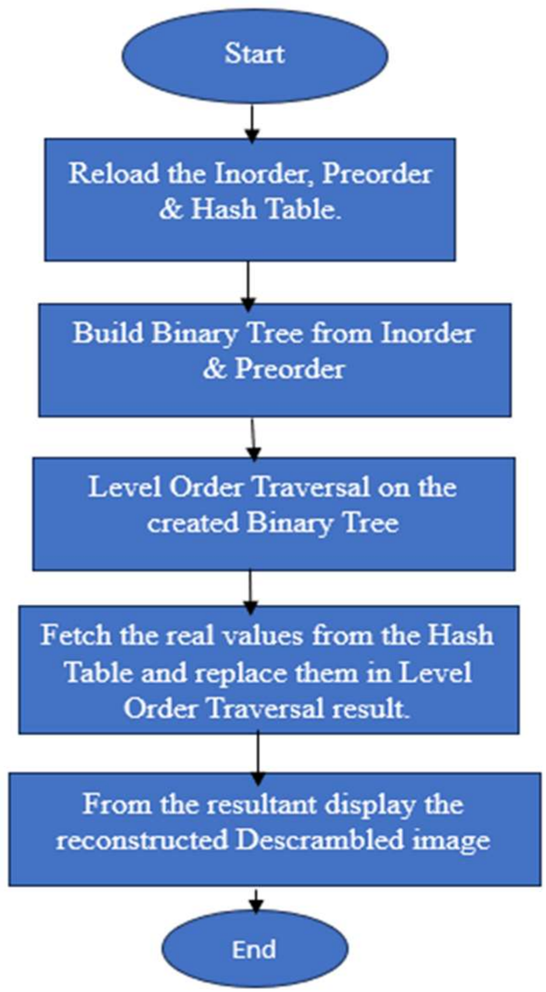


Fig.2 Proposed Model for Descrambling.

Proposed Method(Cont.)

Proposed Image Descrambling Algorithm:

Input:

a)inorder: 2D array representing a structure in inorder traversal.

b)preorder: 2D array representing the same structure in preorder traversal.

c)hash_Table: Hash table for efficient value lookups.

Output: regenrate_2D: Regenerated 2D array to its original grayscale Image.

Begin

1. Load inorder, preorder, and hash_Table from their respective “MAT”-files.
2. Create an empty binary tree and Call buildBinaryTree(inorder, preorder) to recursively construct the tree:
   - If preorder is empty then return an empty node.
   - Else: extract root data, find root's index in inorder to split into left/right subtrees, recursively build binary tree on them, and set current node's children accordingly.
3. Level-Order Traversal Binary Tree:
   - Create an empty queue and Enqueue the root node.
   - While the queue is not empty, dequeue a node, add its data to the result list, and enqueue its left and right children if they exist.
4. Hash Table Lookups:
   - Iterate through each element in the result, call search (hash_Table, element) to find the corresponding value in the hash table, and replace the element in the result with the retrieved value.
5. Reshape result back into a 2D array (regenrate_2D).

Return regenerate_2D

End

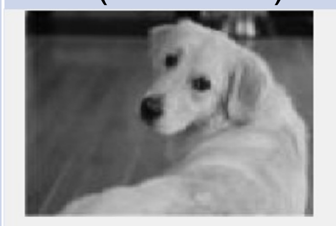
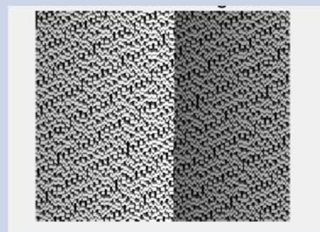
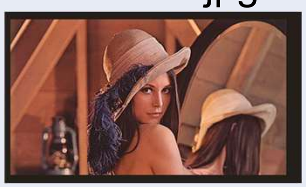
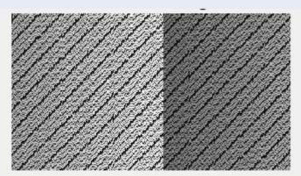
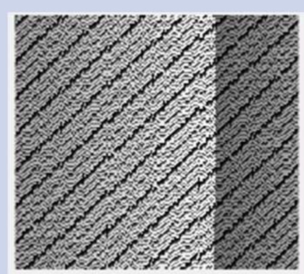
Experimental Set-up

Table 1: Requirements of the Projects

Software Requirements	Hardware Requirements
Operating system: Windows (10 or above) or Linux or MAC	Processor: intel core i5
Programming language: MATLAB	Hard disk:10 GB minimum
IDE: R2023b Update 6(23.2.0.2485118)	RAM:256 MB or more

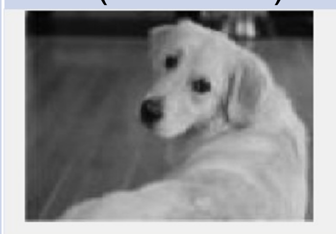
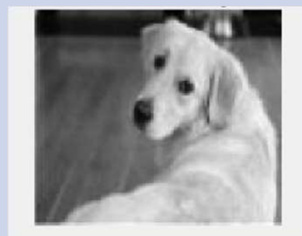
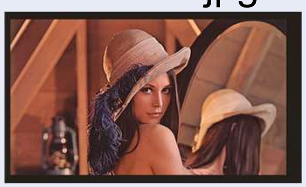
Experimental Results

Table 2: Result Analysis between Grayscale & Scrambled Images

Original Image (Name)	Input Grayscale Image (Size)	Scrambled Image	SSIM	PSNR (dB)
Dog.jpg 	(150X150) 		0.31387	4.51
Lenna.jpg 	(225X400) 		0.32003	3.87
Goldhill.jpeg 	(225X225) 		0.224903	4.54

Experimental Results

Table 2: Result Analysis between Grayscale & Descrambled Images

Original Image (Name)	Input Grayscale Image (Size)	Descrambled Image	SSIM	PSNR (dB)
Dog.jpg 	(150X150) 		1	Inf
Lenna.jpg 	(225X400) 		1	Inf
Goldhill.jpeg 	(225X225) 		1	Inf

Conclusions & Future Scope

- This project successfully developed a novel image **scrambling and descrambling** technique utilizing binary trees and hash tables.
- The method effectively **rearranges pixel intensities** within a binary tree structure, achieving a level of image concealment.
- The use of hash tables facilitates efficient **retrieval of original pixel indices** during descrambling, leading to accurate image reconstruction.
- Design and develop a user-friendly **graphical user interface (GUI)** for the image scrambling algorithm.
- Explore the integration of **Deep learning techniques**, such as **Convolutional Neural Networks (CNNs)**, for enhancing the image scrambling algorithm.
- Continuously optimize the algorithm for **speed and efficiency**, aiming to minimize computational time and memory usage without compromising scrambling quality.

References

[1]. Wu, Y., Xiang, Y., Guo, Y., Tang, J., & Yin, Z. (2020). An improved reversible data hiding in encrypted images using parametric binary tree labeling. IEEE Transactions on Multimedia, 22(8), 1929–1938. https://doi.org/10.1109/tmm.2019.2952979.

[2]. Abbas Fadhil Al-Husainy, M., A. A. Al-Sewadi, H., & Al-Shargabi, B. (2022). Image encryption using a binary search tree structure-based key. International Journal of Computing and Digital Systems, 12(1), 823–836. https://doi.org/10.12785/ijcds/120168.

[3]. Gopalakrishnan t. (2018). Image encryption using multiple chaotic maps in hybrid domain [doctor of philosophy, anna university]. http://hdl.handle.net/10603/253307.

[4]. Wang, X., & Liu, L. (2020). Image encryption based on hash table scrambling and DNA substitution. IEEE Access, 8, 68533–68547. https://doi.org/10.1109/access.2020.2986831.

[5]. Al-Husainy, M. A., & Al-Sewadi, H. A. (2019). Implementing binary search tree concept for image cryptography. International Journal of Advanced Science and Technology, 130, 21–32. https://doi.org/10.33832/ijast.2019.130.03.

[6]. K. Chitra. (2019). Enhancement of image security using improved visual cryptography visual cryptography and digital watermarking techniques [Doctor of philosophy, Pondicherry university]. http://hdl.handle.net/10603/436682.

Acknowledgement

UNIVERSITY OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

The endless thanks go to Lord Almighty for all the blessings he has showered on us, which has enabled us to write this last note in our research work. During the period of our research, as in the rest of our life, we have been blessed by Almighty with some extraordinary people who have spun a web of support around us. Words can never be enough to express how grateful we are to those incredible people in our life who made this thesis possible. We would like to thank them for making our time during my research in the Institute a period We will treasure. We deeply indebted to our research supervisor, Professor Jyoti Khandelwal for such an interesting thesis topic. Each meeting with her added invaluable aspects to the implementation and broadened our perspective. She has guided us with his invaluable suggestions, lightened up the way in our darkest times, and encouraged us a lot in academic life.


Thank You!