iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.62570

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Android Image Steganography

Mrs. Bharti Sahu[1], Om Kawane[2], Sanket Watt[3], Vaibhav Rahangdale[4], Prathmesh Totkar[5]
Dept. of Computer Engineering Dr. D.Y Patil Institute of Technology, Pimpri, Pune, India

Abstract: The internet's expansion and technological advancements have facilitated rapid access to multimedia information, yet they have also heightened concerns regarding privacy and security. This has led researchers to explore digital image steganography as a means of securely storing sensitive data within images. Despite the advantages of existing techniques, challenges persist, necessitating the development of novel approaches. Our study proposes an innovative image steganography method to address these challenges, aiming to balance information concealment with visual quality. By concealing messages within images, our approach enhances security while enabling seamless transmission of confidential information.
Keywords: Digital steganography, Privacy, Security, Challenges, Image steganography, Information concealment.

I. INTRODUCTION

In today's digital landscape, where information flows freely across networks and cyber threats loom large, the importance of safeguarding sensitive data has become paramount. While encryption serves as a stalwart defense against unauthorized access to information, steganography offers a nuanced approach to concealment, operating under the premise that the best way to protect data is to hide its existence altogether. Steganography, derived from the Greek words "steganos" (meaning covered or hidden) and "graphein" (meaning to write), is an ancient practice dating back to ancient Greece, where messages were concealed within wax tablets or tattooed onto the shaved heads of messengers. In the digital realm, steganography entails embedding secret messages within innocuous carrier files, such as images, audio files, or even text, in a manner that eludes detection by casual observers.

At its core, steganography exploits the limitations of human perception, capitalizing on the fact that our senses are not inherently equipped to discern hidden information within seemingly ordinary files. Unlike encryption, which relies on complex algorithms to scramble data, steganography operates under the guise of normalcy, concealing information in plain sight. The clandestine nature of steganography makes it an invaluable tool for covert communication, allowing individuals to transmit sensitive information without alerting potential adversaries to its presence.

This paper introduces a novel approach to steganography, focusing specifically on the concealment of data within digital images. Digital images, with their vast storage capacity and ubiquity in online communication, serve as ideal carrier files for hidden messages. By embedding data within the pixels of an image, our proposed system enables users to transmit confidential information securely with minimal risk of interception or detection. Furthermore, our system incorporates encryption mechanisms to further fortify the security of the hidden data, ensuring that only authorized recipients possess the means to access and decipher the concealed messages.

The objectives of this research paper are threefold: firstly, to explore the principles and techniques underlying image steganography, shedding light on its historical evolution and contemporary applications; secondly, to propose a novel steganographic method for concealing data within digital images, elucidating the technical intricacies of the proposed system; and thirdly, to evaluate the efficacy and security of the proposed system through rigorous testing and analysis. By advancing our understanding of steganography and offering practical insights into its implementation, this research aims to contribute to the broader discourse on cybersecurity and data privacy in the digital age.

II. LITERATURE SURVEY

Mohammed Abod Hussein [1] mentioned background (family income, parents' occupation, gender, educational environment/language), education not consistency (scores in several tests), psychology (satisfaction with engineering studies) are important factors in determining education. factor. This study uses Naive Bayes Classifier, KNN, Decision Tree, etc. to classify students as A and Grade. compares classifiers. [2] Younis Mohammed Younis, Ramadhan J. Mstafa1,2, Haval I. Hussein, Ahmed L. Alyousify, Department of Computer Science.

The paper introduces a two-layer randomness strategy, enhancing security by introducing unpredictability in both pixel selection and the number of bits hidden in each pixel.

[3] Suraj Kumar, Santosh Kumar, Neeraj Kumar Singh, Anandaprova Majumder, Suvamoy Changder. The paper introduces a selective embedding approach, focusing on pixels where color components are unequal Visual Quality: The histograms and Peak Signal-to-Noise Ratio (PSNR) results suggest that the proposed method has better visual quality compared to generic LSB substitution.

[4] Nandhini subramanian 1, (member, IEEE), omar elharrouss1 , somaya al-maadeed 1 , (senior member, IEEE), and ahmed bouridane 2 , (senior member, IEEE), discusses CNN-based methods, especially those using GANs, demonstrate superior hiding capacity compared to traditional methods. Security and Robustness: GAN-based methods exhibit higher security, and their discriminators are trained to overcome steganalysis attacks.

[5] Kevin A. Zhang, Alfredo Cuesta-Infante, Lei Xu, Kalyan Veeramachaneni, the paper claims to achieve a state-of-the-art payload of 4.4 bits per pixel, which is significantly higher than competing deep learning-based approaches.

[6] Puteri Awaliatush Shofro, Kiki Widia, Dwi Dian Ayu Puji Astuti, Eko Hari Rachmawanto, De Rosal Ignatius Moses Setiadi, Christy Atika Sari, the dual encryption approach enhances the security of the embedded messages. The LSB 3-3-2 technique balances imperceptibility and payload capacity. The use of bitwise operations contributes to imperceptibility.

[7] C. Gayathri #1, V. Kalpana, mentions that when combined with cryptography, steganography enhances the security of communication. Versatility: Multiple Formats: Steganography can be applied to various types of data, including text, images, audio, and protocols, making it versatile. Invisibility: Visual Imperceptibility: Techniques like LSB coding aim for visual imperceptibility, making it hard for the human eye to detect changes.

[8] Savitha Bhallamudi, tells us that the LSB substitution steganographic method is acknowledged for its impressive results in hiding data within images. Applicability to Bitmap Images: The method is particularly effective for bitmap images, which involve lossless compression techniques. Extension to Color Images: The method can be extended to color images by performing bit-plane slicing individually for the top four bit-planes of each color channel (R, G, B).

[9] Parberry, I. paper shares that using the Knight's Tour problem for image steganography adds a creative and novel aspect to the technique. It may attract attention and make the steganographic method less predictable. Algorithmic Complexity: The Knight's Tour problem is known for its algorithmic complexity, and leveraging this complexity can contribute to the robustness of the steganographic technique.

[10] Setiadi, D. R. I. M, displays that PSNR has been widely used in various digital image measurements and is considered tested and valid. Simplicity: PSNR is easy to understand and calculate, making it a straightforward metric. Perceptual Consideration: SSIM is designed based on three factors—luminance, contrast, and structure—to better suit the workings of the human visual system. Human-Centric: SSIM takes into account aspects that are more aligned with how humans perceive image quality

III. PROPOSED METHODOLOGY

A. Sender's Part

1) Sender loads an image which he wants to send
2) Then he enters the text, chooses the Image or PDF
3) He sets the password for text and finally encrypts it.
4) Sender can share the Process id or Image directly though the app.

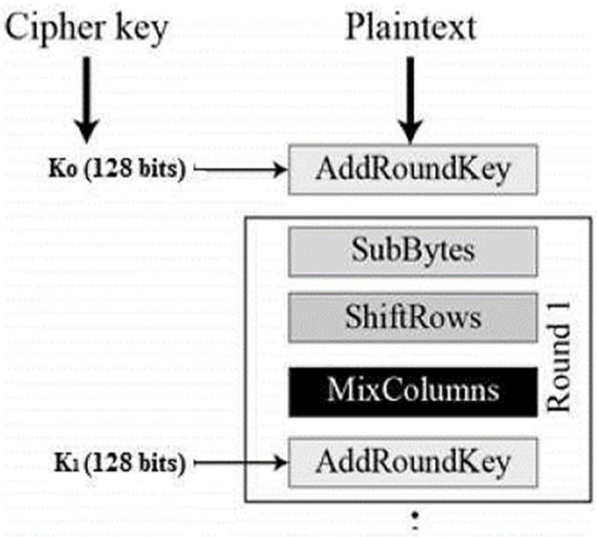
B. Receiver's Part

1) Receiver opens the image in the application or enters process id.
2) Enter password which was used for encrypting (Password can be pre-decided or shared)
3) After typing the password press Decrypt.
4) Text/image or PDF will be shown as it was sent by the Sender.

C. Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprises of four sub-processes. The first-round process is depicted below:

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com



D. Byte Substitution (Sub Bytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

E. Shift rows

Each of the four rows of the matrix is shifted to the left. Any entries that "**fall off"** are re-inserted on the right side of row. Shift is carried out as follows −

1) First row is not shifted.
2) Second row is shifted one (byte) position to the left.
3) Third row is shifted two positions to the left.
4) Fourth row is shifted three positions to the left.

The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

F. Mix Columns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

G. Add round key

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round, then the output is the cipher text. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

H. LSB

In computing, the least significant bit (LSB) is the bit position in a binary integer giving the units value, that is, determining whether the number is even or odd. The LSB is sometimes referred to as the right-most bit, due to the convention in positional notation of writing less significant digits further to the right. It is analogous to the least significant digit of a decimal integer, which is the digit in the ones (right-most) position.

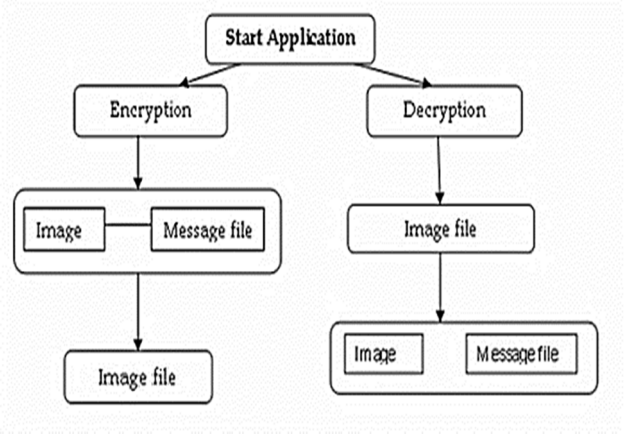
It is common to assign each bit a position number, ranging from zero to N-1, where N is the number of bits in the binary representation used. Normally, this is simply the exponent for the corresponding bit weight in base-2 (such as in 231...20). Although a few CPU manufacturers assign bit numbers the opposite way (which is not the same as different endianness), the term least significant bit itself remains unambiguous as an alias for the unit bit.

By extension, the least significant bits (plural) are the bits of the number closest to, and including, the LSB.

The least significant bits have the useful property of changing rapidly if the number changes even slightly.

For example, if 1 (binary 00000001) is added to 3 (binary 00000011), the result will be 4 (binary 00000100) and three of the least significant bits will change (011 to 100). By contrast, the three most significant bits (MSBs) stay unchanged (000 to 000).
Least significant bits are frequently employed in pseudorandom number generators, steganographic tools, hash functions and checksums.

IV. SYSTEM ARCHITECTURE



V. TYPES OF ALGORITHMS USED

A. *AES (Advanced Encryption Standard)*

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.
A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.
The features of AES are as follows:

1) Symmetric key symmetric block cipher
2) 128-bit data, 128/192/256-bit keys
3) Stronger and faster than Triple-DES
4) Provide full specification and design details
5) Software implementable in C and Java

Operation of AES
AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).
Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −
Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key. The schematic of AES structure is given in the following illustration –

B. *DES (Data Encryption Standard)*

DES works by using the same key to encrypt and decrypt a message, so both the sender and the receiver must know and use the same private key.
Originally designed by researchers at IBM in the early 1970s, DES was adopted by the U.S. government as an official Federal Information Processing Standard (FIPS) in 1977 for the encryption of commercial and sensitive yet unclassified government computer data. It was the first encryption algorithm approved by the U.S. government for public disclosure. This ensured that DES was quickly adopted by industries such as financial services, where the need for strong encryption is high.

The simplicity of DES also saw it used in a wide variety of embedded systems, smart cards, SIM cards and network devices requiring encryption like modems, set-top boxes and routers.

DES key length and brute-force attacks The Data Encryption Standard is a block cipher, meaning a cryptographic key and algorithm are applied to a block of data simultaneously rather than one bit at a time. To encrypt a plaintext message, DES groups it into 64-bit blocks. Each block is enciphered using the secret key into a 64-bit cipher text by means of permutation and substitution.

The process involves 16 rounds and can run in four different modes, encrypting blocks individually or making each cipher block dependent on all the previous blocks. Decryption is simply the inverse of encryption, following the same steps but reversing the order in which the keys are applied. For any cipher, the most basic method of attack is brute force, which involves trying each key until you find the right one. The length of the key determines the number of possible keys and hence the feasibility of this type of attack. DES uses a 64-bit key, but eight of those bits are used for parity checks, effectively limiting the key to 56-bits. Hence, it would take a maximum of 2^56, or 72,057,594,037,927,936, attempts to find the correct key.

Even though few messages encrypted using DES encryption are likely to be subjected to this kind of code-breaking effort, many security experts felt the 56-bit key length was inadequate even before DES was adopted as a standard. (There have always been suspicions that interference from the NSA weakened IBM's original algorithm). Even so, DES remained a trusted and widely used encryption algorithm through the mid-1990s. However, in 1998, a computer built by the Electronic Frontier Foundation (EFF) decrypted a DES-encoded message in 56 hours. By harnessing the power of thousands of networked computers, the following year EFF cut the decryption time to 22 hours.

Apart from providing backwards compatibility in some instances, reliance today upon DES for data confidentiality is a serious security design error in any computer system and should be avoided. There are much more secure algorithms available, such as AES. Much like a cheap suitcase lock, DES will keep the contents safe from honest people, but it won't stop a determined thief.

VI. CHALLENGES

A. *Performance Constraints*

Resource Limitations: Mobile devices have limited processing power and memory compared to desktops. Efficiently embedding and extracting hidden messages in images can be computationally intensive. Battery Consumption: Intensive processing can drain the device's battery quickly, impacting user experience negatively.

Solution: Implement efficient algorithms optimized for mobile devices. Utilize native libraries and hardware acceleration where possible. For instance, use C++ with the Android NDK to offload intensive processing tasks from the Java layer, improving performance and reducing memory usage.

B. *Image Quality and Size*

Preserving Image Quality: Embedding hidden data should not significantly degrade the image quality. Balancing between data capacity and visual imperceptibility is crucial. File Size Management: Embedded data can increase the image file size, which may be problematic for storage and sharing, especially over mobile networks.

Solution: Use advanced embedding techniques like Least Significant Bit (LSB) substitution and adaptive steganography, which selectively embed data in less noticeable regions of the image. Implement quality assessment tools to ensure minimal degradation.

C. *Data Security and Integrity*

Robustness Against Attacks: Ensuring that the embedded data remains intact even if the image undergoes common transformations like compression, resizing, or format conversion. Encryption: Simply hiding data is often not enough; the hidden data should also be encrypted to prevent unauthorized access. Solution: Use robust steganographic algorithms like Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT) that withstand common image transformations. Implement redundancy and error-correction codes to maintain data integrity.

D. *User Experience*

Ease of Use: The application should be user-friendly, making it easy for non-technical users to embed and extract hidden messages without understanding the underlying complexity. Speed: Operations like embedding and extracting data should be quick enough to not frustrate users.

Solution: Design a user-friendly interface with clear instructions and intuitive controls. Provide presets and automated modes for non-technical users, while offering advanced options for experienced users.

E. Compatibility

Device and OS Compatibility: Ensuring the application works across a wide range of Android devices with different hardware specifications and versions of the Android OS. Image Format Support: Supporting various image formats (JPEG, PNG, etc.) and handling differences in how these formats store data.

Solution: Ensure the app supports a wide range of Android versions by using backward-compatible libraries and testing on multiple devices. Employ responsive design principles to adapt the app to various screen sizes and resolutions.

F. Legal and Ethical Considerations

Regulatory Compliance: Navigating the legal landscape regarding data encryption and steganography, as some countries have strict regulations on the use of such technologies. Ethical Use: Ensuring the application is not used for malicious purposes, such as hiding illegal content or communications.

Solution: Stay informed about and comply with local regulations regarding data encryption and steganography. Include features that allow users to anonymize and securely manage their data, ensuring legal use of the app.

G. Testing and Debugging

Varied Environments: Testing the application across different devices, OS versions, and real-world conditions to ensure reliability and robustness. Bug Handling: Identifying and fixing bugs that may arise from the diverse range of hardware and software configurations in the Android ecosystem.

Solution: Conduct extensive testing on a diverse range of devices, OS versions, and real-world conditions. Utilize automated testing tools and frameworks like Espresso and Firebase Test Lab to streamline the testing process.

VII. FUTURE SCOPE

Algorithmic Advancements: Continuously researching and implementing advanced steganographic algorithms to enhance security and robustness while maintaining imperceptibility and resilience to detection. This could involve exploring novel techniques for embedding and extracting data within images.

User Experience Enhancement: Focusing on improving the user interface and experience to make the application more intuitive and user-friendly. This might involve streamlining the process of embedding and extracting hidden information, providing clear instructions, and incorporating user feedback for iterative improvements.

Cross-Platform Compatibility: Expanding compatibility beyond the Android platform to include other operating systems such as iOS or desktop environments. This would require adapting the application to different system architectures and user interfaces while maintaining consistent functionality and security.

REFERENCES

[1] Mohammed Abod Hussein, Saad Al-Momen Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq, A Blind Image Steganography Algorithm Based on Knight Tour Algorithm and QR Codes Article in Academic Journal of Nawroz University · August 2023.

[2] Younis Mohammed Younis, Ramadhan J. Mstafa1,2, Haval I. Hussein, Ahmed L. Alyousify, Department of Computer Science, University of Zakho Kurdistan Region, Iraq Department of Computer Science, Nawroz University, Duhok, KRG – Iraq, Linear Feedback Shift Registers-Based Randomization for Image Steganography Article · August 2023.

[3] Suraj Kumar, Santosh Kumar, Neeraj Kumar Singh, Anandaprova Majumder, Suvamoy Changder, A Novel Approach to Hide Text Data in Colour Image, 978-1-5386-4692-2/18/$31.00 ©2018 IEEE. -This paper proposed a method which uses the technique of selective embedding as the stego-key to feed the data into the image. The noise bed obtained is not so distorted as to arouse suspicion.

[4] Nandhini subramanian 1, (member, IEEE), omar elharrouss1 , somaya al-maadeed 1 , (senior member, IEEE), and ahmed bouridane 2 , (senior member, IEEE), Image Steganography: A Review of the Recent Advances, date of publication January 25, 2021.

[5] Kevin A. Zhang, Alfredo Cuesta-Infante, Lei Xu, Kalyan Veeramachaneni, MIT, Cambridge, MA - 02139, USA, Univ. Rey Juan Carlos, Spain, SteganoGAN: High-Capacity Image Steganography with GANs, 30 Jan 2019.

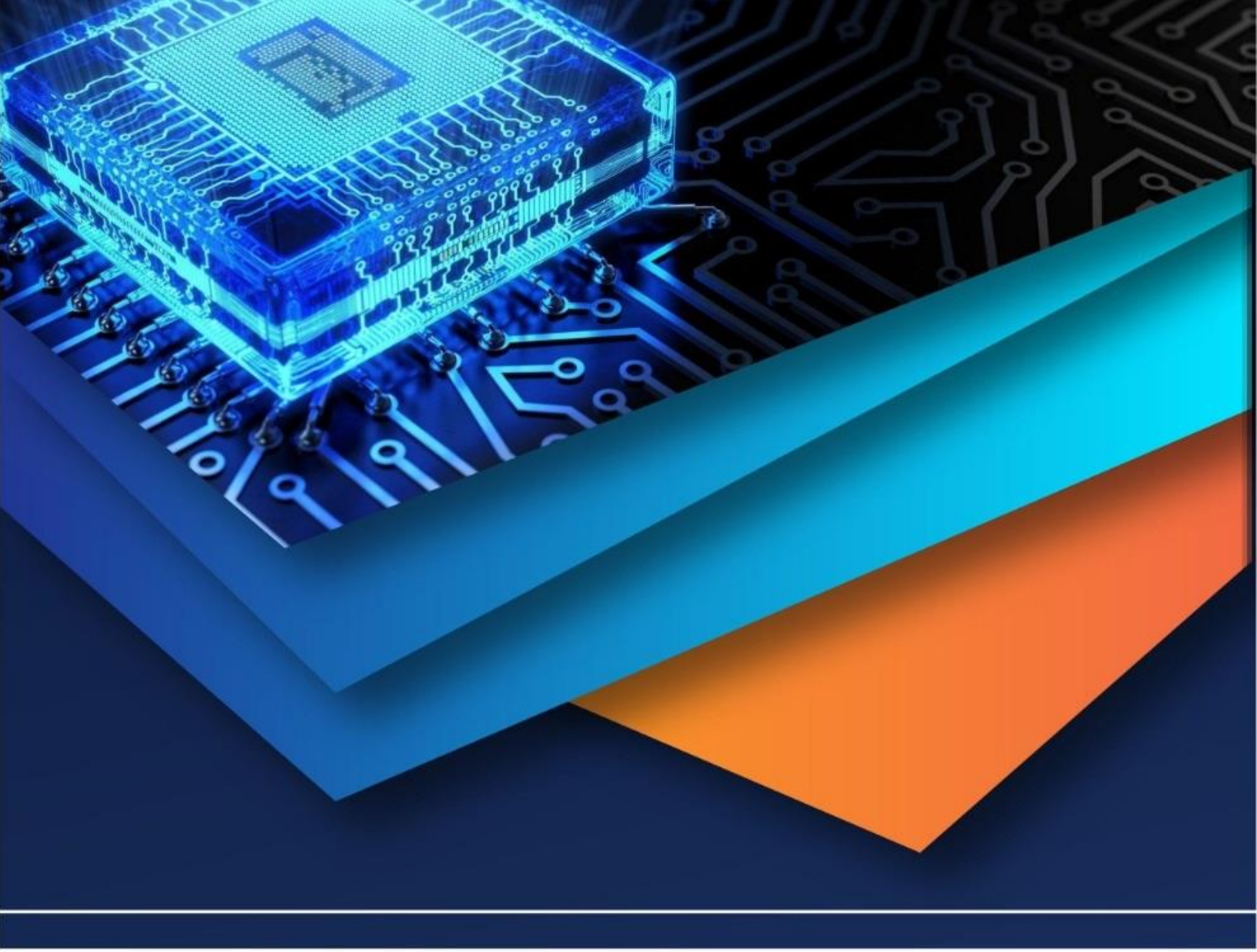
[6] Puteri Awaliatush Shofro, Kiki Widia, Dwi Dian Ayu Puji Astuti, Eko Hari Rachmawanto, De Rosal Ignatius Moses Setiadi, Christy Atika Sari, Computer Science Faculty Dian Nuswantoro University Semarang, Indonesia, Improved Message Payload and Security of Image Steganography using 3-3-2 LSB and Dual Encryption, 2018.

[7] C. Gayathri #1, V. Kalpana, #2 Computer Science & Engineering, School of Computing, SASTRAUNIVERSITY, Tirumalaisamudram, Thanjavur - 613401.Tamilnadu, India, Study on Image Steganography Techniques, ISSN: 0975-4024 Vol 5 No 2 Apr-May 2013.

[8] Savitha Bhallamudi Image Steganography Technical Report · December 2015, EE 7150 – Digital Image Processing Fall 2015.

[9] Parberry, I. (1997). An efficient algorithm for the Knight's tour problem. Discrete Applied Mathematics, 73(3), 251–260.

[10] Setiadi, D. R. I. M. (2021). PSNR vs SSIM: imperceptibility quality assessment for image steganography. Multimedia Tools and Applications, 80(6), 8423–8444. -The paper aims to review, prove and analyze the results of PSNR and SSIM measurements on three spatial domain image steganography methods (LSB, PVD, CRT).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY