IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SECURE AND COVERT COMMUNICATION THROUGH STEGANOGRAPHY AND ENCRYPTION IN DIGITAL IMAGES FOR ROBUST DATA PROTECTION

[1]Bonu Satish Kumar, [2]A Sumati, [3]T Suneetha,[4] Rohith Kumar kr,[5]D Lavanya
[1]Lecturer, [2]Lecturer,[3]Lecturer,[4]Assistant Professor,[5]Lecturer
[1]Computer Science,
[1]SVA Government Degree College, Srikalahasti, India

Abstract: The need for secure and covert communication has become paramount in the digital age. This project presents a method for hiding messages within digital images using a combination of steganography and encryption techniques. The process begins with the user inputting a secret message, which is then encrypted using a Caesar cipher with a user-defined shift key. This encrypted message is further secured by embedding it into an image, leveraging the pixel values as carriers. The embedding process uses a user-provided security key to XOR the ASCII values of the encrypted message with the key, ensuring an additional layer of security. The resultant stegano image retains its visual integrity while concealing the secret message within its pixel data. The extraction process reverses these steps, requiring the correct security key to retrieve and decrypt the hidden message. This method balances simplicity and security, making it accessible while providing robust protection against unauthorized access. Experimental results demonstrate the technique's effectiveness in preserving the visual quality of the carrier image and securely embedding and retrieving hidden messages. Future work will explore enhancing the algorithm's robustness against various image processing attacks and optimizing the embedding capacity without compromising image quality. These qualities make it suitable for various applications, from personal privacy to secure communication in sensitive fields. This paper details this combined technique's methodology, implementation, and evaluation. The results demonstrate its effectiveness in securely embedding and retrieving messages while maintaining the integrity of the carrier image. Future improvements will focus on enhancing the method's robustness against attacks and optimizing the embedding capacity for different types of images.

Index Terms **-** Cipher,stegnography,attacks.encrypt,ceasar.

I. INTRODUCTION

Securing sensitive information in the digital age is vital. While traditional encryption methods are effective, they can sometimes signal the presence of confidential data. Steganography, which hides information within ordinary files, offers a more subtle approach by concealing the existence of the hidden message [1]. This project explores a method that combines image steganography with encryption to achieve secure communication. The process involves two key steps: encrypting the secret message and embedding it into a digital image [2]. First, the secret message is using a Caesar cipher. This technique shifts each letter of the message by a user-defined number of positions in the alphabet, ensuring that the message is not easily readable if discovered. Next, the encrypted message is embedded into an image. Each character of the encrypted message is converted to its ASCII value and XORed with a user-provided security key. This modified ASCII value is then embedded into the

least significant bits of the image's pixel values, making the changes invisible to the human eye. The process of extracting the message is reversed using the correct security key [3]. This approach offers several advantages. It combines the security of encryption with the subtlety of Steganography, preserves the visual quality of the carrier image, and provides a simple method for embedding and retrieving messages.

II. Literature

[1] This paper provides a comprehensive summary of advancements in image steganography, exploring techniques from essential least significant bit (LSB) substitution to advanced machine and deep learning methods. It analyses vital factors like imperceptibility, capacity, and robustness, highlighting the ongoing challenges in balancing embedding capacity with image quality. The authors discuss the continuous conflict between Steganography and steganalysis and identify current trends and future research directions. The review concludes that ongoing innovations in image processing and security are crucial for researchers and practitioners in the field.[2] The authors introduce a novel image encryption and steganography method that combines fuzzy logic and DNA computing. Fuzzy logic dynamically tunes encryption parameters, enhancing adaptability and robustness against attacks, while DNA sequences add complexity to the encryption process. Experiments demonstrate the method's security and efficiency, showing high-quality image preservation. Combining fuzzy logic and DNA sequences offers a promising approach to secure image steganography, significantly improving hidden information security.[3] The authors explore using generative adversarial networks (GANs) for image steganography. These networks leverage their ability to generate photorealistic images, thereby ensuring effective data concealment. The survey reviews various GAN-based techniques, highlighting their superior performance in both embedding and extraction processes. Despite challenges such as the need for Extensive datasets and high computational power, GAN experiments have significantly enhanced the quality and security of steganographic methods. Consequently, the paper underscores the promising potential of GANs in advancing data-hiding techniques within the realm of image steganography.[4] The paper introduces a symmetric embedding scheme using the Gaussian Markov Random Field model to maintain cover image statistics and reduce detectability. Experimental results show significant improvements in imperceptibility and security. This method ensures high-quality steganographic images and robust security, making it applicable in scenarios where the statistical integrity of the cover image is crucial.[5] A new high-capacity steganography method is proposed, combining Elliptic Curve Cryptography (ECC) and Deep Neural Networks (DNNs). ECC provides strong cryptographic properties, while DNNs optimize embedding for high capacity and image quality. Extensive experiments confirm the method's ability to securely hide large amounts of data, enhancing the robustness and security of image steganography.[6] This review covers recent digital Steganography and watermarking developments, including spatial, frequency, and hybrid domain techniques. Applications in copyright protection, data authentication, and secure communication are explored. Challenges in balancing embedding capacity, imperceptibility, and robustness are discussed. The paper identifies trends and future research directions, offering insights for improving steganographic and watermarking security.[7] The paper reviews state-of-the-art steganography techniques in the spatial domain, focusing on manipulating image pixel values. Methods like Least Significant Bit (LSB) substitution are discussed, highlighting strengths and weaknesses. Recent trends are examined, including machine learning and deep learning for optimized embedding. Key challenges and future research directions are presented, emphasizing adaptive and secure methods against advanced steganalysis[8] This paper introduces a versatile steganography algorithm for various image formats in color images, using format-specific embedding techniques. Experimental results show the algorithm maintains high image quality while securely embedding data. The authors highlight the need for format-specific strategies to enhance robustness, concluding that the multiple-format algorithm improves traditional single-format methods.**[9]** The paper proposes a steganography scheme for grayscale images, combining chaos encryption with GANs. Chaos encryption creates complex patterns for enhanced security, while GANs improve imperceptibility and capacity. Experimental results show the method's superiority in security and image quality. The authors conclude that combining chaos encryption with GANs offers an effective approach for secure grayscale image steganography.**[10]**An enhanced LSB-replacement algorithm is presented, optimizing character sequences for minimal embedding distortion. Experiments show improved security and image quality compared to traditional methods. [14] The paper presents a steganography approach combining Quotient Value Differencing with LSB substitution for AMBTC-compressed images. QVD manipulates quotient values, while LSB substitution achieves embedding capacity. Experimental results show high capacity and security of embedded data while maintaining image quality. The authors conclude that QVD combined with LSB substitution offers a promising solution for Steganography in compressed images.[15]The paper discusses

preprocessing images using multiple steganography techniques to improve CNN-based steganalysis performance. This method preprocesses images with different techniques before analysis, enhancing detection accuracy. Experimental results show significant improvements in steganalysis accuracy. The authors conclude that this preprocessing approach enhances CNN-based models for detecting hidden information in images.

III.RESEARCH METHODOLOGY

The methodology for this project involves a combination of encryption and image steganography to hide messages within digital images securely. The process is divided into two main phases: encryption of the secret message and embedding the encrypted message into the image.

3.1 Encryption

Input Secret Message and Parameters: The user inputs the secret message that needs to be hidden. The user provides a shift key for the Caesar cipher encryption. The user provides a security key for further securing the message [5]. Caesar Cipher Encryption: The secret message is converted to lowercase to maintain consistency. The user-defined number of positions in the alphabet shifts each letter in the message. For example, with a shift key of 3, 'a' becomes 'd', 'b' becomes 'e', and so on. Non-alphabet characters remain unchanged during this process [6].

3.2 Embedding Encrypted Message into Image

Read Image: The digital image into which the message will be embedded is read using the OpenCV library [7]. ASCII Conversion and XOR Operation: A dictionary maps ASCII characters to their corresponding values and vice versa. Each character of the encrypted message is converted to its ASCII value. This ASCII value is then XORed with the corresponding character in the security key. If the security key is shorter than the message, it is repeated cyclically [8]. Embedding Process: The modified ASCII values are embedded into the least significant bits of the image's pixel values. The pixel values are manipulated in such a way that the visual quality of the image remains unchanged. The embedding is done sequentially, iterating through the pixels and using the colour channels (R, G, B) to hide the message [9]. Save and Display Stego-Image: The image with the embedded message is saved as a new file. The user is notified of the successful completion of the embedding process [10].

3.3 Extraction and Decryption

Input Security Key: To retrieve the hidden message, the user is prompted to re-enter the security key [11]. Extracting Embedded Message: The segno-image is read, and the modified ASCII values are extracted from the least significant bits of the pixel values. The XOR operation is reversed using the security key to retrieve the original ASCII values of the encrypted message [12]. Caesar Cipher Decryption: The encrypted message is decrypted by shifting the letters in the opposite direction of the encryption process using the same shift key. The final decrypted message is displayed to the user [13].

IV.Code

```
#Hiding message in an image
import cv2
import string
import os
alphabet =
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
'y', 'z','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
'w', 'x', 'y', 'z']
text = input("enter the Secret text: ").lower()
shift_key = int(input("enter the shift number: "))
key = input("Enter the Key to edit(security key) : ")

#Encrypting the secrete Massage

encrypt = ""
```

```
for letter in text:
if letter in the alphabet:
Position = alphabet.index(letter)
new_position = position + shift_key
new_letter = alphabet[new_position]
encrypt += new_letter
else:
encrypt += letter

#ASCII Character to ASCII Value and vice versa
d = {}
c = {}
for i in range(255):
d[chr(i)] = i
c[i] = chr(i)
#reading image using opencv library
f =cv2.imread(r"C:\Users\Karthik\Desktop\Stegno\Dragon Ball Z.jpg")
#Embedding message into an image
k1 = 0
l = len(encrypt)
z = 0
n = 0
m = 0
lnk = len(key)
for i in range(l):
f[n,m,z] = d[encrypt[i]]^d[key[k1]]
n = n+1
m = m+1
z = (z+1)%3
k1 =(k1+1) % lnk
cv2.imwrite("Encrypted_img.jpg",f)
os.startfile("Encrypted_img.jpg")
print("Data hiding in image completed sucessfully.")
k1 = 0
l = len(encrypt)
z = 0
n = 0
m = 0
key1 = input("\n\nRe enter key to extract text:")
decrypt = ""
if key == key1:
for i in range(l):
decrypt += c[f[n,m,z]^d[key[k1]]]
n = n+1
m = m+1
z = (z+1)%3
k1 = (k1+1)%lnk
#Decrypting the Embedded message
plain_text = ""
for letter in decrypt:
if the letter in the alphabet:
position = alphabet.index(letter)
new_position = position - shift_key
new_letter = alphabet[new_position]
plain_text += new_letter
else:
plain_text += letter
```

```
print("The Secret Message is : ",plain_text)
else:
print("Key doesn't matched.")
```

V.Limitations

It's important to acknowledge the limitations of this approach:

1. Capacity: The amount of data that can be hidden is limited by the image size and the embedding technique used.

2. Security: While the security key adds a layer of protection, more sophisticated steganalysis techniques might be able to detect the presence of hidden information.

3. Image quality: Extensive modifications to the LSBs can potentially introduce visible artifacts into the image.

VI. RESULTS AND DISCUSSION

The results of the project are demonstrated through the execution of the code and, the output observed in the terminal and the encrypted image. Here's a detailed explanation of the results:

Figure1: Output of the Code and Popup of Encoded Image

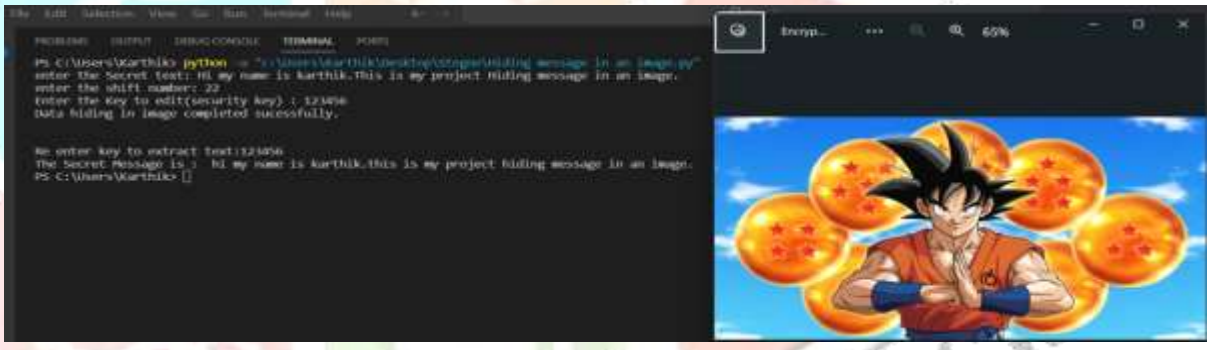

Figure2: Snapshot of Input image (Dragon Ball Z) & Output image (Encrypted_img)

Input Message and Parameters: Secret text: "Hi, my name is Karthik. This is my project Hiding message in an image." Shift key: 22, Security key: "123456"

Encryption Process: The input message is encrypted using the Caesar Cipher method with a shift key of 22. Each letter in the message is shifted by 22 positions in the alphabet to produce the encrypted message.

Embedding Process: The encrypted message is embedded into the image using the provided security key "123456". The image is read using OpenCV, and specific pixel values are modified to store the ASCII values of the encrypted message. The modified image is saved as "Encrypted_img.jpg", which visually appears unchanged from the original image.

Successful Data Hiding: The terminal output confirms that the data hiding in the image was completed successfully.

Extraction and Decryption Process: The user re-enters the security key "123456" to extract the hidden message from the image. The encrypted message is accurately extracted from the image and then decrypted using the inverse of the Caesar Cipher with the shift key of 22. The decrypted message matches the original input message: "Hi, my name is Karthik. This is my project Hiding message in an image."

Visual Confirmation: The screenshot shows the encrypted image "Encrypted_img.jpg," which visually looks the same as the original image, indicating that the hidden message does not alter the image's visual appearance.

6.1 Future Scope

1. Error handling: Implementing mechanisms to handle invalid inputs or errors during image processing.
2. Improved security: Exploring more robust encryption methods for both the message and the security key.
3. Increased capacity: Employing more advanced steganographic techniques
4. that can embed larger payloads without compromising image quality.
5. Visual confirmation: Optionally displaying a comparison between the original and modified images to assess the potential impact on visual quality.

6.2 Conclusion

In conclusion, the provided Python implementation demonstrates a basic but effective approach to image steganography by embedding encrypted messages within digital images. This method leverages simple encryption techniques and pixel manipulation to ensure that the hidden message remains concealed from casual observation. While suitable for introductory applications and learning purposes, the approach has limitations in terms of security and robustness. For enhanced protection and resistance against potential attacks, more advanced cryptographic methods and steganographic techniques should be considered. Overall, this method provides a foundational understanding of how to integrate secrecy into digital media.

VII. REFERENCES

[1] Subramanian, N. 2021.Image steganography: A review of the recent advances. IEEE access, 9, 23409-23423.

[2] El-Khamy, S.2020. A new fuzzy-DNA image encryption and steganography technique. IEEE Access, 8, 148935-148951.

[3] Liu, J.2020.Recent advances of image steganography with generative adversarial networks. IEEE Access, 8, 60575-60597.

[4] Su, W.2020.Image steganography with symmetric embedding using Gaussian Markov random field model. IEEE Transactions on Circuits and Systems for Video Technology, 31(3), 1001-1015.

[5] Duan, X.,2020.A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network. IEEE Access, 8, 25777-25788.

[6] Evsutin, O.2020.Digital Steganography and watermarking for digital images: A review of current research directions. IEEE Access, 8, 166589-166611.

[7] Alhomoud, A. M.2021.Image steganography in spatial domain: Current status, techniques, and trends. Intelligent Automation & Soft Computing, 27(1).

[8] Ansari, A. S.2020.A multiple-format steganography algorithm for color images. IEEE Access, 8, 83926-83939.

[9] Li, Q. 2020.A novel grayscale image steganography scheme based on chaos encryption and generative adversarial networks. IEEE Access, 8, 168166-168176.

[10] Jayapandiyan, J. R.,2020.Enhanced least significant bit replacement algorithm in the spatial domain of Steganography using character sequence optimization. Ieee Access, 8, 136537-136545.

[11] Xiang, Z.,2020.A new convolutional neural network-based steganalysis method for content-adaptive image steganography in the spatial domain. IEEE Access, 8, 47013-47020.

[12] Priyadharshini, 2021.Securing medical images using encryption and LSB steganography. In 2021 international conference on advances in electrical, computing, communication and sustainable technologies (ICAECT) (pp. 1-5).

[13] Sengupta, A.,2020.Structural obfuscation and crypto-steganography-based secured JPEG compression hardware for medical imaging systems. IEEE Access, 8, 6543-6565.

[14] Horng, J. H.,2020.Steganography using quotient value differencing and LSB substitution for AMBTC compressed images. IEEE Access, 8, 129347-129358.

[15] Kato, H.,2020.A preprocessing using multiple steganography for intentional image downsampling on CNN-based steganalysis. IEEE Access, 8, 195578-195593.