Image Steganography using AES Encryption

Haneen Noushad (2020A7PS218U), Hirash Joshy (2020A7PS0163U)

Abstract — **Steganography is the practice of encoding hidden information into an audio, video, image, or text file. It is one of the strategies used to safeguard sensitive or secret information against malicious intrusions. We concentrate on image steganography in this report, which is the practice of concealing information within an image file.**

Steganalysis is the art and science of identifying the use of steganography. Steganalysis often entails a number of processing steps, including noise removal, cropping, blurring, image scaling, and compression. To find hidden information in a stego image, there are many steganalysis tools accessible.

No one is supposed to understand that there exists a secret message in the image other than the intended receiver. So there shouldn't be any distortions, color shifts, or noise in the image as a result of the hidden message. The concealed message should also not be easily accessible to the intruders. We attempt to solve this problem through this paper.

We conclude this by giving a solutions to the above problem. We do it by encrypting the information before embedding it in the image, testing against various attacks

I. Introduction

The Greek words "stegos" which means "to cover" and "grayfia" which means "writing" are the source of the word "steganography," which can be translated as "covered writing" or "hidden writing".

Different Types of Steganography include Text Steganography, Image Steganography, Audio Steganography, Video Steganography, and Network or Protocol Steganography. Compared to traditional cryptographic techniques, steganography has a significant benefit. When someone uses cryptography, they are subtly drawing attention to the fact that there is hidden information present in the relevant medium. Hence, the mere existence of encrypted data, signals to intruders that there exists some top-secret information. Steganography conceals the private data, hence potential hackers are unaware that there is anything intriguing and secret in the first place.

The most commonly used algorithm is Least Significant Bit (LSB). In memory, a picture is represented as an N*M (for grayscale images) or N*M*3 (for color images) matrix, with each entry denoting the pixel's intensity value. By changing the values of a few pixels that are selected by an encryption method, a message can be hidden within an image using image steganography. To know which pixels to choose in order to extract the message, the recipient of the image must be aware of the same process.

II. Literature Survey

The study begins by explaining that digital image steganography involves hiding secret information in carrier images, while steganography analysis aims to detect the presence of hidden information. Then, some conventional methods of digital picture steganography, such as frequency-domain steganography and LSB steganography, are discussed. The paper discusses the latest developments in digital image steganography, including the use of Generative Adversarial Networks (GANs) and adversarial examples. Convolutional Neural Networks (CNNs), in particular, are used in adversarial instances to identify steganography. On the other hand, GANs are used in both carrier modification steganography and steganographic carrier generation. A generator and discriminator are used by GANs to embed secret messages. In summary, the paper provides an overview of the development of traditional digital image steganography, discusses the new directions involving adversarial examples and GANs, and highlights the importance of steganography analysis in information security. [1]

The study focuses on categorizing steganographic research according to its goals. By doing this, the authors hope to give readers a deeper knowledge of the many steganographic approaches and methods.In the paper, assessment tools are carefully analyzed because they are essential for determining the effectiveness and quality of steganographic approaches. The paper also explores different steganographic methods and their evolution and the upgrades over time. It discusses the use of adaptive methods, machine learning, artificial intelligence, generative adversarial networks (GANs), convolutional neural networks (CNNs), coverless techniques, and other approaches employed in steganography. In order to achieve greater security and imperceptibility, the development of effective steganographic techniques is highlighted as a crucial component. The authors review well-known datasets used in steganography research as evidence for their analysis. These datasets are very useful tools for testing and evaluating steganographic methods since they let researchers compare various approaches and measure how well they work. Overall, this paper provides useful data for both inexperienced and professional researchers. [2]

The paper discusses the advancements in image steganography, which involves hiding secret information within digital images for secret communication. The emphasis is on non-additive distortion steganography systems that try to increase the security of additive steganographic distortion functions, such as Synch, CMD, and Dejoin-J. To improve the security of both spatial and JPEG picture steganography, the authors suggest a general technique termed Intra-block Modification Optimization (IbMO). By lowering the absolute difference in noise residuals between cover and stego pictures, IbMO reduces the spatial embedding impact. A recently proposed BBC-based Intra-block Joint Modification (IbJM) approach for JPEG picture steganography includes the IbMO strategy. According to experimental findings, IbMO outperforms Synch and rivals CMD on some metrics when it comes to the security performance of additive distortion steganography techniques. On particular benchmarks, the IbJM technique outperforms DeJoin-J with less computational overhead, and when combined with IbMO, its security performance is further improved. The paper comes to the conclusion that reducing the spatial embedding influence is a good way to improve the security of image steganography. [3]

The area of digital image steganography, which includes securely and secretly concealing information within photographs, is covered in this paper. The three primary techniques are neural network-based, transform-based, and spatial steganography. In order to maximize concealment between cover and steganographic images, this research presents a modified CNN structure with a gain function based on image similarity criteria. The research focuses on using deep convolutional neural networks (CNNs) for steganography. SSIM, MSE, and PSNR are a few of the image metrics that are used to assess how well the suggested algorithm performs. The results demonstrate that the steganographic images produced by the suggested methods are readable while being invisible to the human eye. In terms of SSIM, the proposed method performs better than the original methodology and compares favorably with the currently used steganography techniques. The paper makes several recommendations for improvements, such as updating the code to use TensorFlow 2.0, utilizing a larger and more varied dataset, performing steganalysis on the images that are produced, making fair comparisons with other steganography methods, and highlighting the significance of network structure and hyperparameters. Overall, the research points out the importance of network design decisions in generating better outcomes and shows the potential of using CNNs for image steganography. [4]

The paper uses steganography, as method of data concealment within files, to address the issue of a need for secure transmission of electronic files over the internet. The authors suggest using a two-fold strategy to accomplish this. In order to produce an integer polynomial sequence in coefficient form for non-overlapping groupings of pixels in the cover image, they first introduce the ballot transform (BaT). The data is transformed to make it ready for embedding. Secondly, they use an Index Value Mapping (IVM) and Genetic Algorithm (GA) combination to generate a password of k digits. The positions of the secret data bits that can be inserted into the changed coefficients are determined by the IVM approach. The application of GA tries to balance embedding capability and image fidelity in order to optimize the output quality of the stego image. The proposed technique delivers acceptable Peak Signal-to-Noise Ratio (PSNR) values while preserving a sizable payload capacity, according to experimental results. Additionally, the strategy guarantees two-way security for the data conveyed, strengthening the security and integrity of the information transferred across digital channels. [5]

The research provides an image steganography method that enhances the visual appeal of stego images by using improved matrix encoding. Utilizing more pixel bit-planes also increases the embedding capacity of the approach. The size of the hidden message and the cover image's textural characteristics determine how many image layers are used. Based on the variation between the middle pixel and its neighbors, pixel blocks are classified according to their complexity. In order to accurately identify the embedded regions during extraction without the need for additional information, the suggested approach ensures that the complexity remains intact throughout the data hiding process.

Experimental findings show that the suggested strategy produces stego pictures that are more resistant to statistical and visual steganalysis methods than those produced by previous techniques. The Ensemble Classifier significantly enhances security against detection and is four times more secure than the top-performing previous techniques. [6]

The focus of the research is on image steganography, specifically the least significant bit (LSB) technique, which frequently used to insert secret information. The study suggests a modified method termed circular shift LSB to improve the performance of LSB steganography. The study gives evaluation findings for LSB text steganography utilizing several cover image formats, such as .png, .bmp, and .jpeg, and provides a thorough explanation of the LSB embedding procedure. To determine whether the suggested strategy is effective, a comparison study is done. Peak signal-to-noise ratio (PSNR) and mean square error (MSE) are used to evaluate the method's effectiveness. The study shows that the LSB techniques enhance PSNR and produce low MSE values, indicating high-quality cover pictures that are undetectable to potential intruders or human eyes. [7]

We encounter digital photo images everywhere these days, especially on the internet. These photographs can be easily changed thanks to advances in digital cameras. One method is picture-based steganography, in which secret information is masked by an image. This strategy aims to hide a message within an image without arousing suspicion. This method uses an encoder to conceal the message inside the image. This image doesn't draw unwanted attention from unauthorized people regarding the existence of the secret message when we transfer it to someone else across a network. However, the owner of the decoder can quickly and simply decipher the image's concealed message. Only a little, unnoticeable section of the image is altered by the tool employed for this alteration. The secret message can only be decoded by those who are aware of it. In this manner, even on an insecure network, the message can be sent to the intended recipient without notifying other users. The least significant bits (LSBs) of the picture pixels are often swapped out for hidden bits in this operation. This makes sure that the visual alterations are subtle and difficult to see. The pixel indicator technique (PIT), on the other hand, combines the benefits of a number of earlier steganography techniques. It provides a performance comparison of the PIT, coded LSB substitution, and LSB sequential substitution approaches. Image quality, capacity (the amount of hidden data that can be stored), and security concerns are the main topics of this comparison. [8]

In particular, a compound sinusoidal discrete memristor map and a bit counting strategy are used in this research to present a new technique for concealing sensitive information within color images. A unique class of map known as a 2D compound sinusoidal discrete memristor map is first created by the authors. This map produces chaotic sequences with strong unpredictability features. The secret information will be inserted into the image using these chaotic sequences. The authors then put forth a steganography technique that combines the multiple-base system with the bit counting method. The noticeable alterations or distortions in the final stego image are minimized by this technique. The influence on the visual appearance of the image is minimized by using a multiple-base approach and selectively choosing the bits to alter. The authors create a framework based on the HSV (Hue, Saturation, Value) color model in order to increase the overall security of the steganography system. This approach makes it easier to manipulate the image's color components, making it more difficult to decipher the hidden information. The experimental findings show that the suggested algorithm achieves good imperceptibility, which means that the visual alterations are not immediately apparent to the human eye. This algorithm also has the capacity to incorporate a significant amount of sensitive information into the image. [9]

The necessity to securely share medical information between healthcare practitioners and organizations has elevated to a top concern due to the quick development of telemedicine services. Future e-health systems must succeed by safeguarding patient data. Different from other types of information, medical information, especially medical photographs, requires specific protective procedures. Because healthcare workers rely on medical images to provide precise diagnoses and save lives, maintaining the integrity of these images is of the utmost significance. The quality of patient care may be compromised if the medical pictures are manipulated or tampered with. This research article examines information security measures designed especially for medical imaging. Finding security objectives that can be used to successfully protect medical information is the goal. Encrypting the message or data before hiding it behind medical photographs is one strategy covered in the paper. Sensitive patient information is protected throughout the transfer of medical data thanks to this extra security measure. A particular type of picture steganography called medical image steganography includes concealing data within photographs. The Digital Imaging and Communications in Medicine (DICOM) standard is widely used for storing and transferring medical pictures in the context of imaging. It acts as the foundation for numerous medical imaging departments. This research's main goal is to give an overview of the developments and uses of a particular method termed Least Significant Bit-based (LSB) steganography in relation to medical imaging. [10]

This research focuses on a methodology for using lossy imagine steganography to conceal colored message images within colored cover images. To guarantee the confidentiality and integrity of the secret message, a number of actions must be taken. First, a transformation known as Discrete Cosine Transformation (DCT) is applied to the secret images (message images). This alteration makes the image data more suited for future processing by helping to represent it in a new way. The next step is quantization, which lowers the precision of the image data to minimize storage requirements and boost productivity. The Advanced Encryption Standard (AES) technique is then used to encrypt the message images, adding another level of security to safeguard the message's content. Using a method known as Least Significant Bit (LSB) embedding, the message images are added to the cover images after the encryption stage. In order to create room for the concealed message, the LSB method modifies the cover image's pixel values' least significant bits. The cover image's visual quality shouldn't be greatly impacted by this minor change. Several measures are employed to evaluate the steganography process's effectiveness and quality. A measurement of the visual distortion brought on by the embedding process, the Peak signal-to-noise ratio (PSNR) compares the cover and stego images. Brightness, contrast, and structural information are taken into account when determining how similar the cover and stego images are via the Structural Similarity Index measurement (SSIM). In order to determine how accurately the concealed message was retrieved, the zero-normalized cross-correlation (ZNCC) method compares the extracted message images to the original ones. [11]

The paper focuses on addressing the challenge of data security and privacy in cloud computingWith cloud computing, data is stored and processed on a network of connected computers. Nevertheless, it might be challenging to guarantee the security and privacy of data in the cloud. The authors of the paper used image steganography—the practice of concealing data within images—to find a better approach to get around this problem. To increase the security of the data, they proposed the Multi Level Encryption Algorithm (MLEA) and the Two-Level Encryption Algorithm (TLEA) encryption methods. The study's main goal was to develop a technique for image steganography using the Least Significant Bit (LSB) technique. The least important sections of the image pixels, or the pixels that have the least bearing on the image's overall appearance, are where the data is embedded in this technique. To improve security, the authors processed the image in stages. The data was concealed within the image using the LSB approach, and it was further protected using the MLEA and TLEA encryption algorithms. The authors established that their research offered superior security by comparing it with existing methods and evaluating their method against them. Their suggested approach performed, on average, 19.7814% better than other current approaches. [12]

III. Problem Statement

The key to successful image steganography is to effectively conceal sensitive information inside the image while making sure that the recipient is the only one who can see it. What we need for this is the secret message needs to be resistant to a variety of techniques, including cropping, filtering, and compression and the concealed message must be protected from illegal access and extraction.

IV. Proposed Solution

A. AES Encryption

An extremely popular and secure encryption algorithm is AES (Advanced Encryption Standard). It offers powerful encryption and has undergone thorough testing and professional evaluation. You add an additional layer of security and make it more difficult for unauthorized parties to access the hidden information by encrypting the data before embedding it into an image. AES encryption has other benefits beyond security, such as confidentiality, data integrity, compatibility, and strong encryption.

The secrecy of the embedded data is ensured by AES encryption. The original data cannot be decrypted or retrieved without the encryption key, which is only available to authorized receivers. This prevents unauthorized people from accessing or understanding sensitive data.

AES encryption offers integrity protection in addition to data encryption. The encrypted data is kept intact during transmission and image embedding by using cryptographic algorithms. By restricting any unwanted adjustments, this helps protect the confidentiality of the buried information.

Many different software libraries and systems support AES encryption. By using AES, compatibility with already-in-use encryption technologies is ensured, making picture steganography systems implementation and integration simpler.

AES offers powerful encryption capabilities with key lengths of 128, 192, or 256 bits. The encryption is stronger the larger the key size. By doing this, it is made sure that the embedded data is safe against sophisticated assaults and brute-force attempts.

It's important to keep in mind that while utilizing AES encryption in image steganography increases security, the system's overall security also depends on other elements including key management, embedding methods, and how reliable prospective attackers' steganalysis techniques are.

B. Flowchart for Algorithms

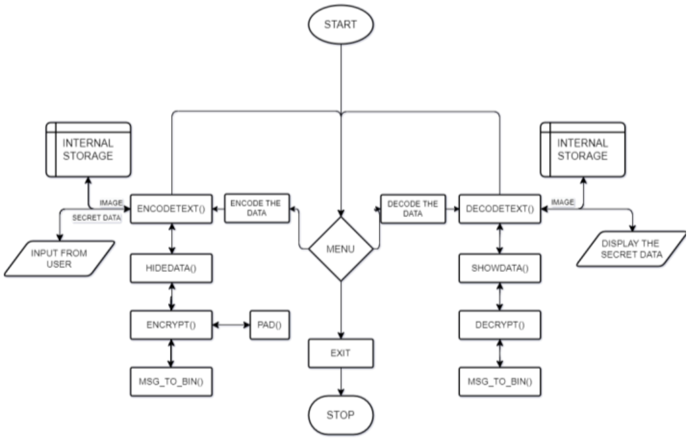


Fig. 1 Flowchart for Algorithms

The above Flowchart shows us the flow of our program and how it goes from one algorithm to the other.

C. Algorithms

There are many different functions involved in our project. ‘//’ denotes the comments in our algorithm.

//AES key was generated using a random bytes generating function

```
AES_key=b'\xa4\x8f\xbd\xdd\x83+\xeb\xf2\x9c\x1d0\x87\xb9\xb9P\xbe\x0b8\xfc\xf4\xa0\t~#\xbb\xaf\xa2\xb1\xd25\xdc\x8d'
```

```
//AES Encryption takes in data of 16-bit blocks so we need to add extra bits to the last block to make it 16 bits
  Algorithm pad(s):
      return s + ((16 - (len(s) % 16)) * '$')

//AES Encyption Algorithm which uses the pad function
Algorithm encrypt(plaintext):
      AES_msg=encrypt(pad(plaintext).toBytes())
      return string(AES_msg)

//AES Decryption function
Algorithm decrypt(ciphertext):
      dec = decrypt(ciphertext).toString
      l = dec.count('$')  //count number of pad bits
     //remove pad bits
      return dec

// Converting types to binary
Algorithm msg_to_bin(msg):
      if type(msg) == str:
         return 8- bit binary number of ascii value for each character in msg
      if type(msg) == bytes or type(msg) == array :
         return 8-bit binary
      if type(msg) == int:
         return 8-bit binary of integer
      else:
         Print("Input type not supported")

// Function to encode the secret message into the image
Algorithm hide_data(img, secret_msg):
     // calculating the maximum bytes for encoding
      nBytes = img.shape[0] * img.shape[1] * 3 / 8
      print("Maximum Bytes for encoding:", nBytes)
      // checking whether the number of bytes for encoding is less than the maximum bytes in the image
      secret_msg=encrypt(secret_msg)
      if len(secret_msg) > nBytes:
         Print("Error encountered insufficient bytes, need bigger image or less data!!")
      secret_msg += "~~~"       //delimiter
      dataIndex = 0
      // converting the input data to binary format using the msg_to_bin() function
      bin_secret_msg = msg_to_bin(secret_msg)

      // finding the length of data that requires to be hidden
      dataLen = len(bin_secret_msg)
      for values in img:
         for pixels in values:
            // converting RGB values to binary format
            r, g, b = msg_to_bin(pixels)
            // modifying the LSB only if there is data remaining to store
            if dataIndex < dataLen:
               // hiding the data into LSB of Red pixel
               dataIndex += 1
            if dataIndex < dataLen:
               // hiding the data into LSB of Green pixel
               dataIndex += 1
            if dataIndex < dataLen:
               // hiding the data into LSB of Blue pixel
               dataIndex += 1
            // if data is encoded, break out the loop
            if dataIndex >= dataLen:
               break
      return img

   // Function to decode the secret message into the image
   Algorithm show_data(img):
      bin_data = ""
      for values in img:
         for pixels in values:
            // converting the Red, Green, Blue values into binary format
            r, g, b = msg_to_bin(pixels)
            // data extraction from the LSB of Red pixel
            bin_data +=   lsb(r)
            // data extraction from the LSB of Green pixel
            bin_data +=   lsb(g)
            // data extraction from the LSB of Blue pixel
            bin_data +=   lsb(b)
      // split by 8-Bits
      allBytes = bin_data /8
      // converting from bits to characters
      decodedData = ""
      for bytes in allBytes:
        // checking if we have reached the delimiter which is "~~~"
        // if true break
         decodedData += chr(int(bytes, 2))
   return decodedData

   // Function to encode data into Image
   Algorithm encodeText():
      // reading the input image
      img = read(img_filename)

      data = input("Enter data to be encoded: ")
      if (len(data) == 0):
         print('Data is Empty')

      // calling the hide_data() function to hide the secret message into the selected image
      encodedImage = hide_data(img, data)
   cv2.imwrite(Encodedimg_filename,encodedImage)
```

```
//  Function to decode the data in the image
    Algorithm decodeText():
        img = read(Encodedimg_filename)  // reading the image
        text = show_data(img)  //calling show_data() function
        A=text.toBytes() //converting string to bytes
    return decrypt(A) //return decrypted text


//  Image steganography  Menu
    while True:
        n = int(input("\nImage Steganography \n1. Encode the data \n2. Decode the data \n3. Exit\n Select the option: "))
        if (n == 1):
            print("\nEncoding...")
            encodeText()
        else if (n == 2):
            print("\nDecoding...")
            print("Decoded message is :" + decodeText())
        else if (n == 3):
            print("\nExiting...")
            break
        else:
        print("Inserted value is incorrect!")
```

V. Simulation and Results

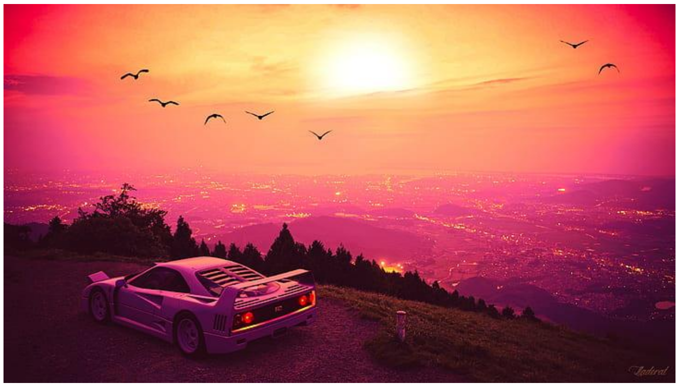


Fig. 1 Image before encoding

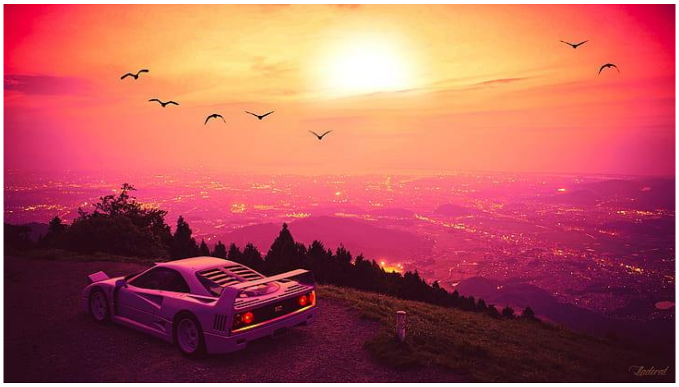
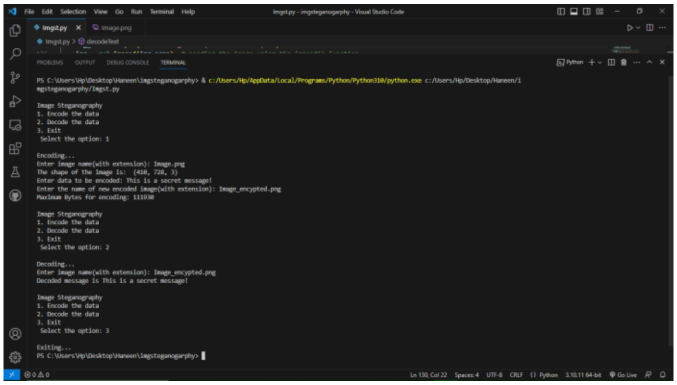


Fig. 2 Image after encoding

We can see that there is no difference in Figure 2 and Figure 3 to our human eye, but some data is encoded into the Figure 3. To our eyes we cannot distinguish between the two and hence no one will come to know that there exists a secret data in that image.

In case some attacker gets to know that there exists some data in the image and try to encode the image the data will be useless to the attacker as it is encrypted using a very strong AES Encryption. Only a person with the appropriate key will be able to decrypt that data.



File Edit Selection View Go Run Terminal Help
Imgst.py - imgsteganogarphy - Visual Studio Code
Imgst.py
Image.png
Imgst.py > decodeText
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Ln 130, Col 22 Spaces: 4 UTF-8 CRLF {} Python 3.10.11 64-bit Go Live

```
PS C:\Users\Hp\Desktop\Haneen\imgsteganogarphy> & c:/Users/Hp/AppData/Local/Programs/Python/Python310/python.exe c:/Users/Hp/Desktop/Haneen/i
mgsteganogarphy/Imgst.py

Image Steganography
1. Encode the data
2. Decode the data
3. Exit
 Select the option: 1

Encoding...
Enter image name(with extension): Image.png
The shape of the image is:  (410, 728, 3)
Enter data to be encoded: This is a secret message!
Enter the name of new encoded image(with extension): Image_encypted.png
Maximum Bytes for encoding: 111930

Image Steganography
1. Encode the data
2. Decode the data
3. Exit
 Select the option: 2

Decoding...
Enter image name(with extension): Image_encypted.png
Decoded message is This is a secret message!

Image Steganography
1. Encode the data
2. Decode the data
3. Exit
 Select the option: 3

Exiting...
PS C:\Users\Hp\Desktop\Haneen\imgsteganogarphy> ▌
```

Fig. 4 Running the code

Figure 4 shows us how the code is run and how we can encode and decode the data using our program. First from the menu we have to choose whether to encode image, decode image, or to exit. If we choose to encode, we have to provide the image file name with extension and also the secret data to be encoded into the image. We will also have to provide the filename for the encoded image to save the image after encoding. If we choose to decode, we just have to provide the encoded image file name with extension and the program will retrieve the encoded data in the image.

VI. Conclusion

Image steganography is the art of successfully hiding sensitive information inside an image while ensuring that only the receiver may see it. Image steganography may be made more efficient and safe by utilizing a variety of approaches, testing against various threats, and embedding in the frequency domain.

It's worth noting that while using AES encryption in image steganography provides added security, the overall security of the system also depends on other factors like key management, embedding techniques, and the robustness of the steganalysis techniques employed by potential attackers. AES encryption also provides Confidentiality, Data Integrity, Compatibility, and Strong encryption.

REFERENCES

[1] "The Development and New Direction of Digital Image Stenography" *He Huang, Yinghui Xue , Linna Fan, Mo Li*

[2] "Digital image steganography survey and investigation (goal, assessment, method, development, and dataset)" *De Rosal Ignatius Moses Setiadi, Supriadi Rustad, Guruh Fajar Shidik 2022*

[3] "Towards improving the security of image steganography via minimizing the spatial embedding impact" *Wenkang Su, Jiangqun Ni, Fangjun Huang 2022*

[4] "Image Steganography Using Deep Learning Techniques" *Anthony Rene Guzman 2022*

[5] "A secured image steganography method based on ballot transform and genetic algorithm" *Sabbir Hossain, Souradeep Mukhopadhyay, Biswarup Ray 2022*

[6] "A secure image steganography based on modified matrix encoding using the adaptive region selection technique" *Tuan Duc Nguyen, Hai Quoc Le 2022*

[7] "Analytical Study on LSB-Based Image Steganography Approach" *Oluwakemi Christiana Abikoye 2022*

[8] "Performance Evaluation of LSB Sequential and Pixel Indicator Algorithms in Image Steganography" *Jabed Al Faysal, Khalid Mahbub Jahan 2022*

[9] "A novel color image steganography algorithm based on bit counting and multiple-base system" *Hong-wei Xie, Ya-jun Gao, Jing-yu Sun 2022*

[10] "A Comprehensive Review on Medical Image Steganography Based on LSB Technique and Potential Challenges" *Bushra Abdullah Shtayt, Nur Haryani Zakaria, Nor Hazlyna Harun 2021*

[11] "A Novel Image Steganography Technique Using AES Encryption in DCT Domain" *Aman Sahu, Chittaranjan Pradhan 2023*

[12] "An Improved Approach of Image Steganography Based on Least Significant Bit Technique for Secure Communication in Cloud" *Md. Khorshed Alam, Samia Nushrat 2023*

Haneen Noushad
2020A7PS0218U
Bits Pilani Dubai Campus
f20200218@dubai.bits-pilani.ac.in

Hirash Joshy
2020A7PS0163U
Bits Pilani Dubai Campus
f20200163@dubai.bits-pilani.ac.in