https://doi.org/10.31436/ijpcc.v10i1.449

Image Steganography:

Comparative Analysis of their Techniques, Complexity and Enhancements

Ahmad Zulfakar Bin Abd Aziz, Muhammad Fitri Bin Mohd Sultan, Nurul Liyana Binti Mohamad Zulkufli*

Department of Computer Science, Kulliyyah of Information Communication Technology, International Islamic University Malaysia

*Corresponding author: liyanazulkufli@iium.edu.my

(Received: 10th June 2023; Accepted: 27th September 2023; Published on-line: 28th January 2024)

Abstract— Steganography is the practice of hiding secret information within other media, such as images, audio, video, and text. It has become increasingly important in today's society as a way to enable private and secure communication. This research project focuses on image steganography techniques that are used to evade detection of the secret message through statistical steganalysis techniques. The aim of this research is to compare and evaluate different image steganography methods, study their implementation complexity, and propose a framework to improve current approaches. The research will provide a comparison of the efficiency of different steganography techniques in avoiding detection by steganalysis and may lead to the development of better steganography techniques in the future. This paper focuses on the three steganography methods in the spatial domain: Least Significant Bit (LSB), Pixel-Value Difference (PVD) and Edge-based Data Embedding (EBE) methods. A simple experiment has been conducted to encrypt several images using these three methods, and the distortion measure for LSB using mean square error (MSE) and peak-to-noise ratio (PSNR) has been investigated. Although the distortion measure result is considered acceptable for LSB method in the experiment, all methods resulted in significant difference in the file capacity. This shows that further enhancement in the security of the encryption is necessary so that the secret message will not be easily discovered. Thus, in this paper, we propose a conceptualized enhancement in the security using Morse Code, Base 64, SHA-245, and Advanced Encryption Standard (AES) before encrypting using PVD.

Keywords— Steganography, steganalysis, spatial domain, Edge-based Data Embedding.

I. Introduction

The internet and advanced communications have made it easier to transfer information from one end of the world to the other. It has, however, resulted in the development of the idea of information security. The most widely used means of ensuring information security is cryptography, which consists of encrypting the information being transmitted. However, if an attacker steals the data and discovers a way to decode it, he will gain access to private information. Therefore, this is where steganography comes in, which aims to introduce data into undoubtful situations.

Carrier data that contains secret messages is delivered to the recipient suspiciously. This carrier data may be in the form of a protocol, audio, video, text, picture, or any other digital object of any type. If the attacker manages to access the data, he will have no doubts that the data is carrying more information. Several steganography for video, image, audio, text including their techniques and performance have been explored in a number of publications [1][2][3]. In these publications, scholars discussed various steganography types, their techniques, evaluations, and future research prospects.

Considered as the art of writing secretly, steganography is a technique that involves hiding information within media such as images, audio, or video in a way that is difficult for outsiders to detect. It has a number of potential applications, including the safe transmission of secret data by organisations or agencies, the embedding of personal information within photographs on smart identity cards for copyright control, and the embedding of patient details within medical images to protect information and speed up transmission. Overall, steganography can be used to protect and transmit sensitive information in a covert manner. Digital steganography has grown into a method for concealing a file within a piece of media, such as a picture or an audio or video. Steganography's goal is to hide the payload (embedded information) in the cover picture so that the payload's existence in the cover image is undetectable to human beings (see Figure 1).

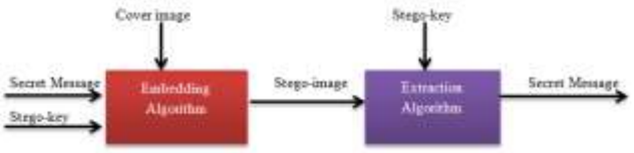


Fig. 1 Basic model of image steganography

As the demand for security and technological advancement grows, there is also a need for innovative image steganography algorithms. Therefore, several researchers have proposed various methods to increase the efficacy and level of security of image steganography in order to avoid discovery of the secret message by well-known statistical steganalysis tools. There are two sorts of domains in which picture steganography may be implemented: the spatial domain method and the transform domain method. In the spatial domain, processing is performed directly on the pixel values of the image, whereas in the frequency domain, pixel values are first transformed, and then processing is performed on the transformed coefficients.

In this research, our objectives are to gain a better understanding of factors causing differences in each technique, to analyse the complexity between every steganography technique, and to suggest a new algorithm or ideas to enhance current image steganography methods.

As for the significance of this research, we hope that more people will have a better understanding of which strategies are most effective and how complex these methods are, so that they may use them in their own projects. Moreover, we hope that our proposed algorithm or concepts of image steganography will aid in improving the security and performance of present image steganography techniques, as well as making them more efficient.

II. Review of Previous Researches

There is a study that presented a specific image-based steganography approach for exchanging information more securely between two sites by combining the notion of a secret key for authentication at both ends in order to reach a high level of security [4]. Prior to the embedding procedure, the cover picture is split into various objects using a normalised cut. Permutation, integer wavelet transformation by lifting technique, and segmentation have all been used to enhance the security of the data, which has been embedded using a modified LSB embedding method on various cuts of the cover picture to create a variety of stego objects as an additional safeguard. In the end, the stego image is created by fusing together several stego objects and then being transmitted to the recipient. Using the Iterative Center Weighted Median (ICWM) algorithm, Gallegos-Funes et al. demonstrated a robust steganographic approach for hiding information in the wavelet domain.

SOM and wavelet contrast are used in a steganographic approach by Zhang Jiajia and their colleagues to give a large capacity for the concealed secret data and to preserve a good visual quality of the hidden secret data [5]. To begin, the wavelet contrast was obtained by first dividing a picture into blocks and then decomposing each block into a single-level wavelet. With SOM Neural Networks, the blocks were further categorised. Finally, steganography based on modulus was used to hide the hidden information.

Another unique approach for image steganography that is more effective and can conceal more data in a single cover image than any other currently known technique has been suggested by [6]. In this paper, the Most Significant Bit (MSB) is presented. The method relies on the cover image's bit difference between two adjacent pixels. Embedding is done with bits 5 and 6 of a pixel in the image. The difference between bits 5 and 6 is determined by the secret information bit that is obtained. If the difference between bit 5 and 6 is equal to the incoming secret bit, no modification is needed in bit 5. If the difference between bit 5 and 6 does not match the incoming bit, then bit 5 is modified such that the difference and the incoming bit are equal.

Existing steganography methods can be divided into three categories: methods that exploit picture format, methods that embed in the spatial domain, and methods that embed in the frequency domain. Image format exploitation methods are the most common type of steganography method. Steganography, in its most basic form, is performed by changing the image's Least Significant Bits (LSBs) in such a way that the carrier image retains its visual integrity. In order to embed images in the spatial domain method, the least significant bits of the image pixel values' bytes must be altered. This method can be carried out in a sequential manner or in a randomised manner.

Algorithms based on this method have a large payload, however the method is unstable and vulnerable to statistical attacks, and visual attacks can occasionally be sufficient. The second type of method, the frequency domain method. This method is based on the embedding in the coefficient in the frequency domain. For instance, Discrete Cosine Transformation (DCT) and Discrete Wavelet Transformation (DWT). This type of technique is more resistant to standard image processing operations as well as lossy compression than other types of techniques.

One additional way is adaptive steganography, which is a technology that adapts the message embedding technique to the actual content and characteristics of the image. These methods can be used to avoid areas of uniform colour and choose pixels with a high local standard deviation. Edge embedding can also be used in conjunction with adaptive Steganography to conceal information [7].

Additional steganography approaches based on spatial domain and transform domain techniques are presented in the following works to demonstrate that the Image steganography may be adapted and improved from time to time. For spatial domain method, Masoud Nosrati et al. [8] utilising Genetic Algorithm in their project. Based on pre-embedded concealing techniques, Genetic Algorithm-based steganography in Image Segments is a useful tool for storing data in a carrier image. Message strings and LSBs are segmented into blocks for the evolutionary algorithm in order to achieve this. In order to extract the message's embedded data, a key file was constructed later once the exact locations of the embedded data were determined. Analyses based on Least Changes in Sample Image and Histogram demonstrate that the suggested technique may be used in the field effectively.

According to Tahir Ali et al.'s proposed method in paper [9], the message bit is stored in the LSB of one of the three colour components, RGB, depending on the parity of the three LSBs of R, G, and B components of a 24-bit colour picture. Secret data may be recovered and hidden using the notion of parity check in this technique The RGB components of a 24-bit colour picture each have 8-bits, and the LSB of three of them create a group of three bits. A series of even or odd numbers of 1s can be generated afterwards by combining these three bits. Odd parity is used when the three bits acquired have an odd number of 1s, whereas even parity is used when the three bits have an even number of 1s. These parity and message bits are used to determine how each colour component's LSB is embedded. With very little alterations to the image's pixel values, the researchers were able to conceal an enormous amount of data within just a single RGB picture.

Next, there is an LSB replacement method for 24-bit colour images mentioned in Vijaya Bhandari and colleagues' [10] paper. In this technique, a 24-bit colour image is used to demonstrate how more data, i.e. secret image message bits, are hidden in the blue plane than in the red and green planes, due to the lower intensity of blue light or objects in human visual perception when compared to the other two colours. This approach is proven in a mat lab, and the results reveal that the PSNR value of the 24-bit colour picture is higher, and the histogram comparison shows that the stego image is more comparable to the original cover image than the 8-bit colour image.

Turning to Transform Domain approaches, the paper by Ahmed ElSayed et. al [11] demonstrates a low frequency curvelet transform method for a highly secure data concealing system under a cover picture. Only the low frequency component of the curvelet transform is used to offer strong security with the four secret keys (the two shuffle keys, the encryption key, and a key for data concealing). The Curvelet transform's low-frequency component provides advantages over other steganography approaches, including the decreased processing time. As a result, the Curvelet transform is able to manage curve discontinuities without affecting the edges since it hides data in the low-frequency components, which improves stego object quality. The results show that while there are no noticeable variations in the Wavelet transform Case between the stego and cover images, there are noticeable variances in the Curvelet transform Case.

Next, according to the research conducted by V. Senthooran et al. [12], it provides a novel data concealing strategy that is based on the values of the modified quantization table and the DCT coefficients. According to the mathematical formula, the embedding strength of each coefficient is obtained by comparing the suitable quantization table value and DCT coefficients in the correct sequence. In a later step, the hidden bits are stored in frequency components of quantized DCT coefficients, which is accomplished by employing the LSB approach. The suggested embedding approach is divided into three parts. Initially, the segment separates the cover-image into blocks of 8 x 8 pixels that do not overlap. In the second segment, the DCT coefficient value in each block is compared to the appropriate quantization table entry in each block. This suggested technique creates increased embedding capacity and acceptable picture quality by making adjustments to the conventional quantization table in the middle portion of the segment, which happens in quantized coefficients in every block throughout the last segment. By doubling the quantization table and using an interpolation approach, it is possible to expand the picture size to 32x32 or 16x16 pixel blocks in the future, and the stego image size is evaluated in each instance.

Jyoti Gaba and colleagues [13] have suggested a technology known as compress encrypt stego (CES) for the sharing of information in a safe way. This approach entails pre-processing of data before masking it over a cover picture. Pre-processing involves the use of the compression factor to lower the size factor, which is then adjusted to make use of the key. Compression decreases the amount of data, allowing for more information to be contained in the cover picture. Because the changed data is received after compression, even if attackers attempt to identify the presence of data, they will be unable to recover the original texts buried without the use of a key. Furthermore, secret data is hidden in DCT coefficient values rather than in the full data set. The results of the investigation show that the approach is more resilient and secure since the data is buried in the blue component DCT coefficients and because it is not sensitive to the human eye.

Other than general images, steganography researches also are conducted on medical data such as brain magnetic resonance (MR) images and electric encephalography (EEG) data. Devi et. al. [30] found that steganography using LSB substitution method doesn't affect the classification rate of “normal” and “abnormal” brain MR images. In another paper [31], they utilized discrete wavelet transformation (DWT) to extract relevant features from both original and stego MR images. This can boost the privacy and security of the medical images while still preserve satisfactory classification rate.

For enhancement of the steganography algorithm, Dicky Nofriansyah et al. [14] discusses a new image encryption technique that combines these three techniques: Hill cipher, Morse code, and LSB steganography. The article describes how these techniques are used together to create a secure image encryption system. Similar to that, Dr. R. Sridevi et al. [15] describes a technique for combining image steganography and cryptography using the Least Significant Bit (LSB) technique and the Advanced Encryption Standard (AES) algorithm. In this technique, the LSB technique is used to hide a secret message within an image, and AES is used to encrypt the resulting "stego" image. The encrypted stego image can then be recovered to reveal the original image and extract the hidden message. The authors of the paper conclude that this technique is effective for secret communication and provides good security. They also suggest that future research could focus on combining image encryption and data hiding with lossy compression.

Sujay Narayana et al. [16] also introduces the concept of combining cryptography and steganography and proposes a new algorithm to improve resistance to steganalysis. The proposed method aims to achieve a higher similarity between the cover and stego images, as well as better imperceptibility. According to the results obtained, using steganography in combination with encryption provides a secure method of secret communication between two parties. The authors suggest that future work could involve using the proposed method to arrange the text obtained from the encryption of an image into a word or meaningful sentence, and exploring new methods to prevent steganalysis beyond the least significant bit (LSB) method.

Some of these steganography algorithms are quite complex owing to the length of time required to conceal secret data, whilst others are straightforward and simple. Based on the study of previous researches in this field, current image steganography methods can be improved in terms of efficiency and security since the steganography attempt is still slightly noticeable on human eyes.

III. Image Steganography Algorithms

There are two broad categories of image steganography algorithms which are spatial domain method and transform domain method:

A. *Spatial Domain Method*

Using this method, the secret data is stored by performing direct manipulation on the pixel values of the cover image. It implies that only a small number of pixel values in the cover image are changed during data concealment and encryption. There are many different Spatial domain approaches, however a few of the most common are listed below:

- Least Significant Bit (LSB)
- Pixel Value Differencing (PVD)
- Edges-based Data Embedding method (EBE)

1) *Least Significant Bit (LSB)*

Least Significant Bit Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message.

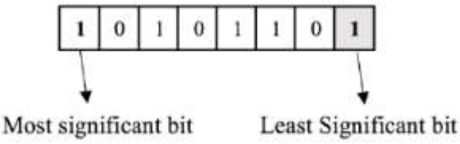


Fig. 2 Least and most significant bit of a 8-bit binary array

When using a 24-bit image, each pixel in the image has 8 bits of data that represent the colours red, green, and blue. Using the final bit of each 8-bit data block, the secret data may be substituted. This alteration will be so tiny that it will go unnoticed by the human eyes. Figure 4 illustrates the process of embedding a 6-bit 1101101 data in 2 pixels. The cover data are the ones that are located on the left side of the figure if it is deemed to be enlarged by two pixels. 6-bits of data are concealed inside the RGB data of two pixels, and the stego-data conversion of the cover data may be seen on the right-side figure. As can be seen quite obviously in figure3, there has been no discernible change to the cover data.

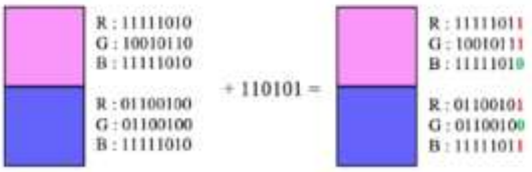


Fig. 3 Least significant bit of a 8-bit binary array

On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary colour, changing the LSB of a pixel results in small changes in the intensity of the colours. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference.

In its simplest form, LSB makes use of BMP images, since they use lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion. For this reason, LSB steganography has also been developed for use with other image file formats.

2) Pixel Value Differencing (PVD)

Based on the fact that our human vision is sensitive to slight changes in the smooth regions, while can tolerate more severe changes in the edge regions, the PVD-based and simple least significant bits scheme have been proposed to enhance the embedding capacity without introducing obvious visual artefacts into stego images. In PVD-based schemes, the number of embedded bits is determined by the difference between the pixel and its neighbour. The larger the difference amount is, the more secret bits can be embedded. Usually, PVD based approaches can achieve more imperceptible results compared with those typical LSB-based approaches with the same embedding capacity. Proposed method also hides large and adaptive k-LSB substitution at the edge area of image and PVD for smooth regions of image. So, in this way the technique provides both larger capacity and high visual quality according to experimental results. This method is complex due to adaptive k generation for substitution of LSB.

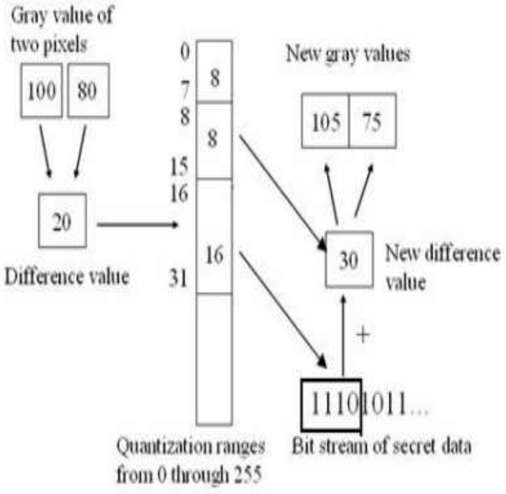


Fig. 4 Data embedding process of Wu-Tsai's PVD method

3) Edges-based Data Embedding (EBE)

Edges-based data embedding (EBE) steganography is in which only the sharper edge regions are used for hiding the message while keeping the other smoother regions as they are. It is more difficult to observe changes at the sharper edges than those in smoother regions. In this method Enhanced Least Significant Bit algorithm is used which can reduce the rate of pixel modification thereby increasing the security both visually and statistically. Edge Detection algorithm hides secret data into the pixels that make up the extracted edges of the carrier image. The secret data can be of any type, not necessarily text, and they are actually concealed into the three LSBs (Least Significant Bits) of the pixels of the carrier image, but not in every pixel, only in the ones that are part of the edges detected by the edge detection algorithm.

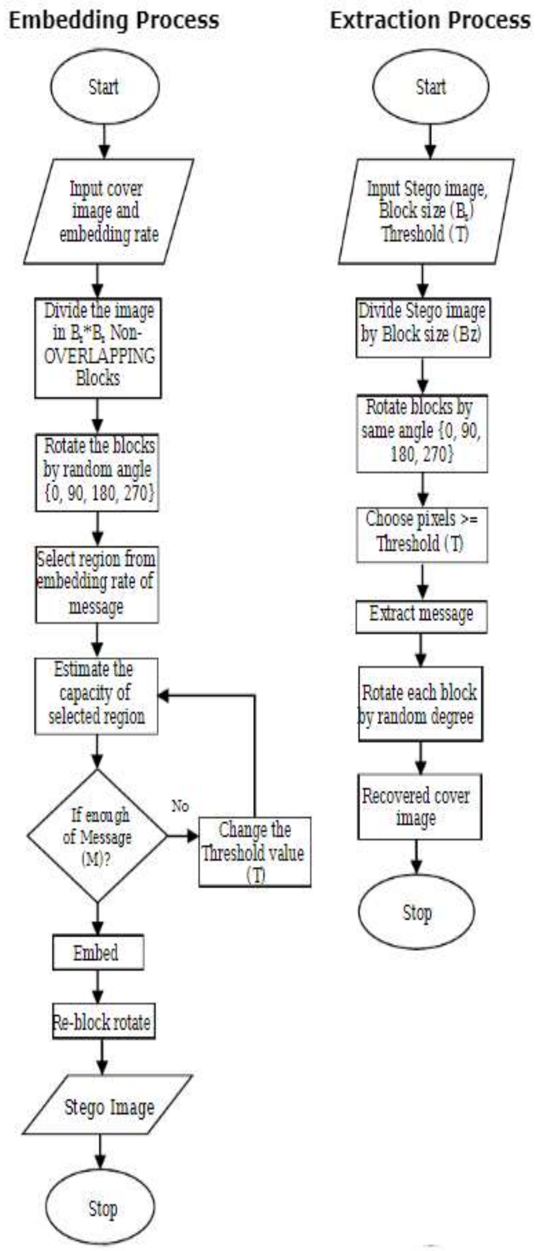

Embedding Process
Extraction Process
Start
Input cover image and embedding rate
Divide the image in B1*B1 Non-OVERLAPPING Blocks
Rotate the blocks by random angle {0, 90, 180, 270}
Select region from embedding rate of message
Estimate the capacity of selected region
If enough of Message (M)?
No
Change the Threshold value (T)
Embed
Re-block rotate
Stego Image
Stop
Start
Input Stego image, Block size (B1) Threshold (T)
Divide Stego image by Block size (Bz)
Rotate blocks by same angle {0, 90, 180, 270}
Choose pixels >= Threshold (T)
Extract message
Rotate each block by random degree
Recovered cover image
Stop

Fig. 5 Flowchart of EBE method

B. *Transform Domain Method*

It is also known as frequency domain techniques because it includes embedding hidden data in the frequency or transform of a cover image, which is known as the frequency domain technique. These strategies are a little more complicated than the previous methods of concealing data in photographs. Various transformations and algorithms are employed to conceal data, but only a few are effective which are:

- Discrete Cosine Transformation (DCT)
- Discrete Wavelet Transformation (DWT)
- Spread Spectrum

However, due to lack of tools and resources as well as the fact that there is a relatively large number of image steganography algorithms available, we chose to carry out an experiment to compare several different approaches that are considered the best in the spatial domain method in the recent years.

IV. Evaluation Parameters

In this paper, we focus on three techniques in the spatial domain methods: LSB, PVD, and EBE.

The three basic criteria for evaluating image steganographic algorithms are hiding capacity, distortion measure, and security. The fourth parameter may alternatively be thought of as the difficulty or complexity of the algorithm. However, no researcher has used algorithm complexity as an evaluation measure in the literature.

A. *Hiding Capacity*

The maximum amount of data that may be hidden in an image is known as the maximum hiding capacity. It is quantified in bits, bytes, or kilobytes. The bit-rate, also known as bits per pixel (bpp) or bits per byte, is the maximum number of bits that can be buried per pixel (bpp). The steganography method is more effective if the hiding capability is greater.

B. *Distortion Measure*

Stego-images should not be visible, which means the distortion should not be obvious. There are numerous criteria that can be used to evaluate the distortion, including; (i) Mean Square Error (MSE), (ii) Root Mean Square Error (RMSE), (iii) Peak signal-to-noise Ratio (PSNR), (iv) Weighted PSNR (WPSNR) (v) Correlation, (vi) Quality Index, (vii) Structural SIMilarity (SSIM) index (viii) Kullback-Leibler Divergence (K-L divergence), (ix) Manhattan Distance, and (x) Euclidean Distance.

For the experiment, we will use some images to identify their MSE and PSNR.

C. *Security Check*

If a steganographic technique is resistant to multiple steganalytic attacks, it is said to be secure. There are numerous steganalysis methods for evaluating the security of a steganographic method. A technique based on LSB substitution can be evaluated using RS analysis, while a technique based on PVD can be evaluated using pixel difference histogram analysis.

V. Comparison Experiment

Tests are carried out using a set of original JPG image files for inserting a message text or message file that will be hidden.

We used six samples of different image files of different sizes and messages to be hidden that have been through the encryption process (see Fig. 6):

Fig. 6 Images used for the experiment
(from top left to right: image01.jpg, image02.jpg, image03.jpg, image04.jpg, image05.jpg, and image06.jpg)

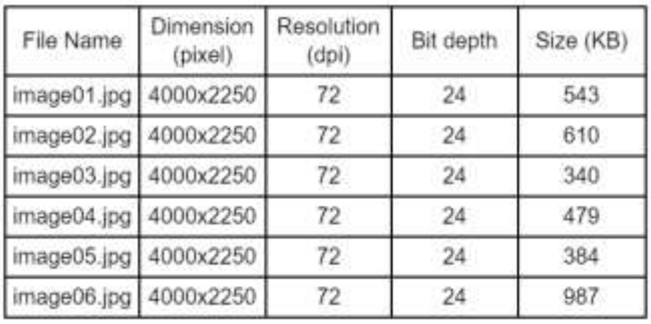
We carried out this experiment to identify the properties of each image in its original form and after alterations that contained hidden messages in them. The properties that are being tested are item type, size (in kB), dimensions (in pixels), resolution (in dpi), and bit depth. All of the images are in jpg format. The hidden text for all images is the same, which is “THIS IS A TEST."

The testing process is done using different techniques of image steganography. We will be focusing on the Spatial Domain Method which is LSB, PVD and EBE.

For the LSB method, we use the QuickStego application. The QuickStego application allows hidden text to hide messages inside images. For PVD and EBE methods, we run the algorithms using Python language.

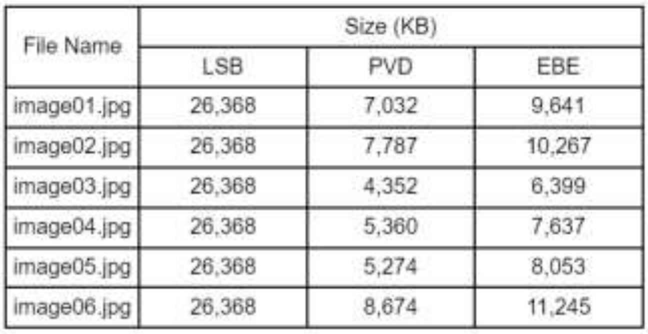
The details and result of the experiment is shown in the table below:

TABLE I
THE PROPERTIES OF IMAGES BEFORE ENCRYPTION



File Name	Dimension (pixel)	Resolution (dpi)	Bit depth	Size (KB)
image01.jpg	4000x2250	72	24	543
image02.jpg	4000x2250	72	24	610
image03.jpg	4000x2250	72	24	340
image04.jpg	4000x2250	72	24	479
image05.jpg	4000x2250	72	24	384
image06.jpg	4000x2250	72	24	987

TABLE II
THE SIZES OF IMAGES AFTER ENCRYPTION USING DIFFERENT TECHNIQUES IN SPATIAL DOMAIN METHOD



File Name	Size (KB)		
	LSB	PVD	EBE
image01.jpg	26,368	7,032	9,641
image02.jpg	26,368	7,787	10,267
image03.jpg	26,368	4,352	6,399
image04.jpg	26,368	5,360	7,637
image05.jpg	26,368	5,274	8,053
image06.jpg	26,368	8,674	11,245

Based on the testing shown in Table I and Table II, it shows that the change in the file capacity size after each method's encryption is very significant. This mean the it is obvious that there is a hidden message in the image, thus implies that further enhancement

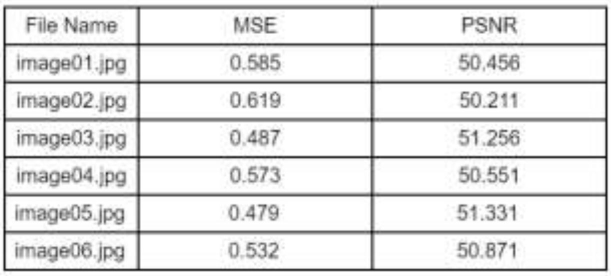
Later in the experiment, the MSE and PSNR are calculated using the formula in (1) for MSE and (2) for PSNR. The MSE should be kept to a minimum. MSE is zero if the original image and the stego-image are identical. A greater PSNR value indicates less distortion. A PSNR greater than 40 decibels (dB) is excellent. A PSNR between 30 and 40 dB may be acceptable, but a PSNR less than 30 dB is unacceptable because of the excessive distortion. For colour images a pixel comprises 3 bytes.

$$\mathrm{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - q_{ij})^2 \qquad (1)$$

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255 \times 255}{\mathrm{MSE}} \qquad (2)$$

The MSE and PSNR results of the experiment are shown in the table below:

TABLE III
RESULT OF MSE AND PSNR CALCULATED BY USING STEGO-IMAGE FROM LSB TECHNIQUE



File Name	MSE	PSNR
image01.jpg	0.585	50.456
image02.jpg	0.619	50.211
image03.jpg	0.487	51.256
image04.jpg	0.573	50.551
image05.jpg	0.479	51.331
image06.jpg	0.532	50.871

Based on the results in Table 2, the average MSE for the six images is 0.545833, while the average PSNR is 50.77933.

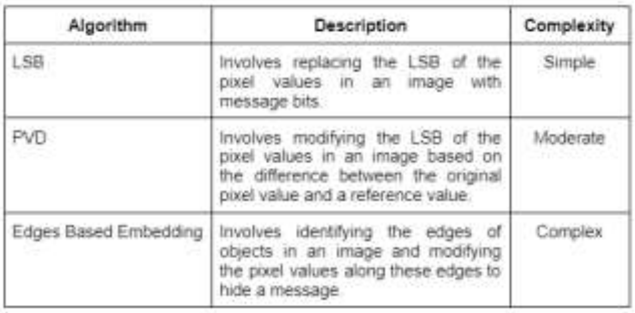
From the experiment, although the distortion rate based on the PSNR values for LSB method is acceptable, there are some changes in the image's properties after the encryption, where the sizes of the images increased significantly. This is true for not only LSB, but also for the PVD and EBE methods.

This implies that the image manipulation is obvious where the size change can easily be detected, thus increasing the chance of the secret message being discovered. Thus, it is important to make sure that the secret message will not be easily discovered by improving the security before the encryption.

VI. IMPLEMENTATION COMPLEXITY

Here are some findings about the implementation complexity from our analysis of these image steganography approaches, based on the experiments we ran.

TABLE IV
IMPLEMENTATION COMPLEXITY OF STEGANOGRAPHY METHODS



Algorithm	Description	Complexity
LSB	Involves replacing the LSB of the pixel values in an image with message bits.	Simple
PVD	Involves modifying the LSB of the pixel values in an image based on the difference between the original pixel value and a reference value.	Moderate
Edges Based Embedding	Involves identifying the edges of objects in an image and modifying the pixel values along these edges to hide a message.	Complex

The implementation and the needs of the application can affect the implementation complexity of these methods. The complexity of these algorithms can vary depending on the specific implementation and the requirements of the application.

LSB algorithms are generally the simplest to implement, but they may have lower capacity and may not provide as much security against steganalysis attacks as other algorithms. PVD is a slightly more complex algorithm to implement, but it has a relatively high capacity and allows for lossless recovery of the original message. Edges based embedding is the most complex of these algorithms to implement, but it can be relatively robust against image processing operations and may provide some security against steganalysis attacks.

VII. Conceptualization of the Research Approach

In this paper, one of our objectives is to improve upon the methods and algorithms currently used in steganography. We offer some suggestions for the development of a strong algorithm for steganography. However, there is no universally accepted definition of the "strongest" steganography algorithm because algorithm strength depends on the specific security requirements and constraints of a given application. But still, we do uncover a few broad notions that can be exploited to improve upon the state-of-the-art of image steganography algorithms and design a new technique that is more secure. These concepts are as follows:

1) Using strong encryption:
To ensure the safety of the hidden message, steganography algorithms should employ rigorous encryption techniques. Those who do not know the password or decryption key will have a hard time understanding what is being said.
2) Use multiple layers of encoding and encryption:
Utilise a combination of encoding and encryption methods to ensure that the concealed message remains secure. The cover media might be encoded with a steganography algorithm after the message was encrypted with AES and Base64, for instance.
3) Use a robust cover media:
It is important to protect the hidden message using cover material that can't be easily altered. An image that has been significantly manipulated or compressed, for instance, might not be as good at concealing the message as one that has not been altered in any way.

Therefore, based on these fundamental principles, we have developed the concept of combining steganography and cryptography in order to produce an image steganography algorithm that is both secure and efficient, is resistant to attacks, and offers a high level of security for the data that is being hidden.

As a means of improving the image steganography algorithms, we suggest a framework in which the Pixel Value Differencing (PVD) algorithm is used in tandem with several other kinds of cryptographic function: Morse Code, Base 64, SHA-256, and Advanced Encryption Standard (AES); as shown in Fig. 7 below:

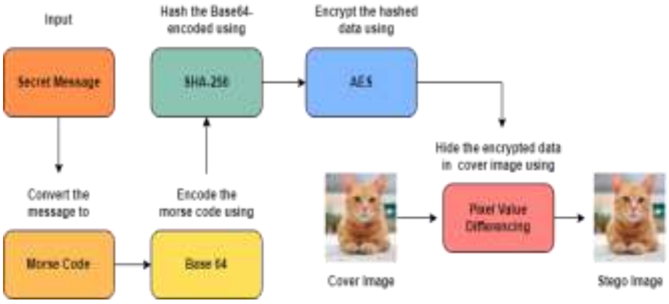

Input
Secret Message
Hash the Base64-encoded using
SHA-256
Encrypt the hashed data using
AES
Convert the message to
Morse Code
Encode the morse code using
Base 64
Hide the encrypted data in cover image using
Pixel Value Differencing
Cover Image
Stego Image

Fig. 7 Flow Diagram of the Proposed Framework

Pixel Value Differencing (PVD) is a method for concealing data within an image by altering the pixel values in a way that is difficult to identify, while at the same time keeping the overall visual quality of the image intact. In order to conceal the data, this method requires the selection of a certain group of pixels within the image and the subsequent modification of the values of these pixels in a specified pattern.

Reason why PVD is used is because it is a relatively simple algorithm to implement, compatible with a wide range of image formats, has a relatively high capacity and allows for lossless recovery of the original message and relatively robust against image processing operations, making it a practical choice for many applications as compared to other steganography algorithms.

The Least Significant Bits (LSB) method is also a relatively simple algorithm to implement. However, they may have lower capacity and may not provide as much security against steganalysis attacks as other algorithms. Moving to Edge's based embedding method, it is a more complex algorithm to implement, as it involves identifying the edges of objects in an image and modifying the pixel values along these edges to hide a message. This type of algorithm may require more sophisticated image processing techniques and may not be suitable for all types of images.

Therefore, we believe that PVD would be the most suitable algorithm to employ for our concept of combining image steganography with cryptography functions.

Next, to construct a secure and durable image steganography algorithm that is resistant to attacks and provides a high level of security for the hidden data, we combine the PVD image steganography algorithm with numerous cryptographic algorithms like AES, base64, SHA-256, and Morse code.

Morse code is a method of encoding characters as a series of dots and dashes that can be transmitted using a simple signalling device, such as a flashlight or a telegraph key. Morse code is not typically considered to be a strong encryption method, as it can be easily decoded by anyone who is familiar with the code. However, using Morse code in combination with other techniques (such as PVD) may make it more difficult to detect the hidden message.

Base 64 is a method of encoding binary data as a series of ASCII characters. It is often used to represent binary data in a text-based format, such as in email attachments or in HTML. Base 64 is not typically considered to be a strong encryption method, as it can be easily decoded by anyone who is familiar with the encoding. However, using Base 64 in combination with other techniques (such as PVD and Morse code) may make it more difficult to detect the hidden message.

SHA-256 is a cryptographic hash function that is used to compute a digital fingerprint (or hash) of a file or message. It is a one-way function, which means that it is not possible to reverse the process and reconstruct the original message from the hash. SHA-256 is considered to be a strong encryption method, but it is not typically used to encrypt messages directly. Instead, it is often used to create a unique digital fingerprint of a message, which can be used to verify the integrity of the message.

AES (Advanced Encryption Standard) is a symmetric key encryption algorithm that is widely used to secure data transmitted over networks or stored in electronic devices. AES is considered to be a strong encryption method, and it is widely used to protect sensitive information, such as credit card numbers and passwords.

VIII. Implementation Procedure for the Proposed Framework

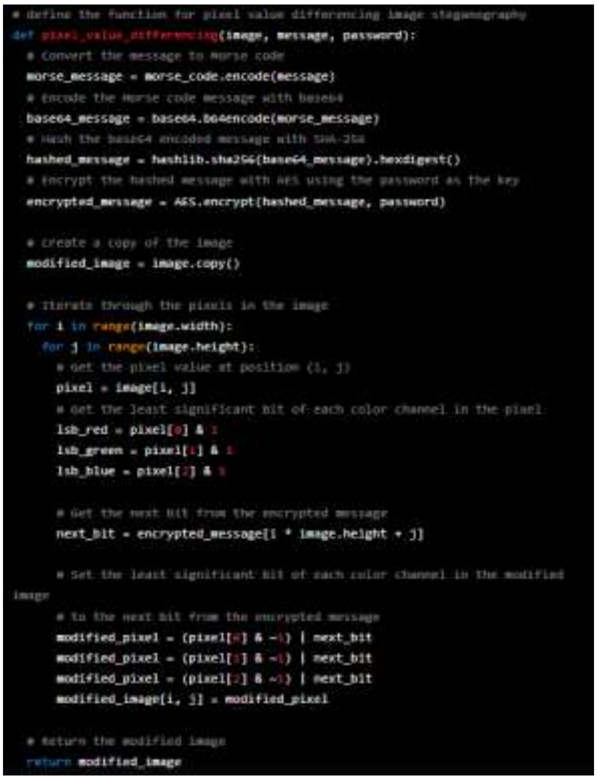
The following is the procedure for the implementation of our suggested method of image steganography, combining Pixel Value Differencing algorithm with Morse code, Base 64, SHA-256 and AES encryption technique (refer to Fig. 8):

1. Convert the data to be hidden to Morse code. This is done by mapping each character in the data to a series of dots and dashes, according to the Morse code alphabet.
2. Encode the Morse code data using base64. Base64 is a widely used encoding scheme that represents binary data in an ASCII string format, allowing it to be easily transmitted over networks or stored as text.
3. Hash the base64-encoded data using SHA-256. SHA-256 (Secure Hash Algorithm 256) is a widely used cryptographic hash function that is considered to be very secure and resistant to attacks.
4. Encrypt the hashed data using AES. AES (Advanced Encryption Standard) is a widely used symmetric encryption algorithm that is considered to be very secure and efficient.
5. Divide the encrypted data into blocks. This can be done by dividing the data into fixed-size blocks or by using a block cipher mode of operation such as CBC (Cipher Block Chaining).
6. Select the pixels in the image in which to hide the data. This can be done by selecting a set of pixels that are evenly spaced throughout the image, or by using a more sophisticated selection algorithm that takes into account the characteristics of the image and the data being hidden.
7. Modify the values of the selected pixels using pixel value differencing, in a way that hides the data blocks within the pixel values. This can be done by adding or subtracting a predetermined value from the pixel values, or by using a more complex pixel value differencing algorithm.
8. Save the modified image with the hidden, encrypted data. The resulting stego image contains the hidden, encrypted data.

This pseudo code (refer to Fig. 8) assumes that the input image is a 2D array representing the pixel values of the image, and that the morse_code and AES libraries have functions encode and encrypt, respectively, that take a message and password as input and return the encoded or encrypted message.

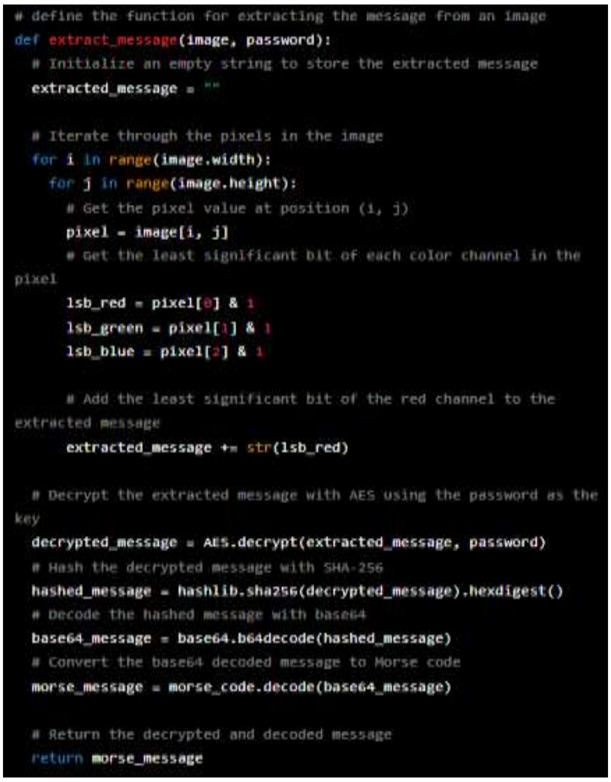
Next, to retrieve the hidden data, the following steps would be performed (refer to Fig. 9):

1. Load the stego image and retrieve the hidden data from the pixel values using pixel value differencing.
2. Merge the data blocks into a single encrypted message.
3. Decrypt the encrypted data using AES.
4. Hash the decrypted data using SHA-256 and compare the resulting hash to the original hash that was included with the hidden data. If the hashes match, it indicates that the decrypted data has not been modified and is authentic.
5. Decode the decrypted data using base64.
6. Convert the base64 data to Morse code.
7. Convert the Morse code data to its original form.
8. This method utilises a number of different levels of encryption, as well as pixel value differencing algorithm, in order to guarantee the confidentiality and credibility of the data that is concealed. It is feasible to develop a robust image steganography algorithm that is resistant to attacks by utilising AES, base64, SHA-256, and Morse code. This type of technique offers a high level of security for the data that is concealed and is capable of hiding images.



```
# define the function for pixel value differencing image steganography
def pixel_value_differencing(image, message, password):
  # Convert the message to Morse code
  morse_message = morse_code.encode(message)
  # Encode the Morse code message with base64
  base64_message = base64.b64encode(morse_message)
  # Hash the base64 encoded message with SHA-256
  hashed_message = hashlib.sha256(base64_message).hexdigest()
  # Encrypt the hashed message with AES using the password as the key
  encrypted_message = AES.encrypt(hashed_message, password)

  # create a copy of the image
  modified_image = image.copy()

  # Iterate through the pixels in the image
  for i in range(image.width):
    for j in range(image.height):
      # Get the pixel value at position (i, j)
      pixel = image[i, j]
      # Get the least significant bit of each color channel in the pixel
      lsb_red = pixel[0] & 1
      lsb_green = pixel[1] & 1
      lsb_blue = pixel[2] & 1

      # Get the next bit from the encrypted message
      next_bit = encrypted_message[i * image.height + j]

      # Set the least significant bit of each color channel in the modified
image
      # to the next bit from the encrypted message
      modified_pixel = (pixel[0] & ~1) | next_bit
      modified_pixel = (pixel[1] & ~1) | next_bit
      modified_pixel = (pixel[2] & ~1) | next_bit
      modified_image[i, j] = modified_pixel

  # Return the modified image
  return modified_image
```

Fig. 8 The proposed framework's pseudo code for the embedding process



```
# define the function for extracting the message from an image
def extract_message(image, password):
  # Initialize an empty string to store the extracted message
  extracted_message = ""

  # Iterate through the pixels in the image
  for i in range(image.width):
    for j in range(image.height):
      # Get the pixel value at position (i, j)
      pixel = image[i, j]
      # Get the least significant bit of each color channel in the
pixel
      lsb_red = pixel[0] & 1
      lsb_green = pixel[1] & 1
      lsb_blue = pixel[2] & 1

      # Add the least significant bit of the red channel to the
extracted message
      extracted_message += str(lsb_red)

  # Decrypt the extracted message with AES using the password as the
key
  decrypted_message = AES.decrypt(extracted_message, password)
  # Hash the decrypted message with SHA-256
  hashed_message = hashlib.sha256(decrypted_message).hexdigest()
  # Decode the hashed message with base64
  base64_message = base64.b64decode(hashed_message)
  # Convert the base64 decoded message to Morse code
  morse_message = morse_code.decode(base64_message)

  # Return the decrypted and decoded message
  return morse_message
```

Fig. 9 The proposed framework's pseudo code for the extraction process

IX. Conclusion and Future Work

Based on the findings of our experiment with several methods of image steganography in spatial domain method, LSB insertion embedded the message in the cover image. Choosing a pixel to embed was critical since LSB insertion affects pixels. Modified pixels in parts of the image with similar pixels were more noticeable. PVD divides images into two-pixel blocks. The difference between two pixels is mapped to a range table to determine how many bits to hide. Information concealment using steganography, especially edge detection filters, is a way to label distinct colours to identify dark areas of image. This method hides text in dark areas, but the data is put in low bits of each eight-bit pixel.

Next, our proposed framework of combining the pixel value differencing image steganography approach with AES encryption, Morse code, base64 encoding, and SHA-256 hashing can provide a variety of benefits in terms of the security and concealment of the secret data within the image. These benefits include the following:

1) Improved Security:
By using multiple techniques to hide and protect the message, the proposed model may be more resistant to detection and decoding by an attacker. For example, using PVD to hide the message within an image, AES to encrypt the message, Morse code to encode the encrypted message, and SHA-256 to create a digital fingerprint of the message may make it more difficult for an attacker to access the hidden message and to verify that the data has not been tampered with.

2) Robustness:
By using multiple techniques to hide and protect the message, the proposed model may be more resistant to degradation or tampering during transmission. For example, using PVD to hide the message within the image and SHA-256 to create a digital fingerprint of the message may allow the recipient to verify the integrity of the message upon receipt.

3) Data capacity:
The different techniques used in the proposed model may allow for a larger amount of information to be hidden within a given cover media, such as an image. For example, using PVD to hide the message within the image and Morse code to encode the message may allow for a larger message to be hidden within the image than would be possible with either technique alone.

4) Increased Versatility
Combining different techniques may allow the message to be transmitted over a wider range of

communication channels and in different formats, which may be useful in different situations.

5) Lossless recovery of the original message:
Some techniques, such as PVD, allow for lossless recovery of the original message, which means that the message can be extracted without any loss of quality. Using multiple techniques may allow for the preservation of this lossless recovery property while still providing additional security.

6) Resistance to steganalysis:
Using multiple techniques may make it more difficult for an attacker to use steganalysis techniques to detect the presence of a hidden message.

As per our proposed framework of image steganography algorithm combining pixel value differencing with several cryptography functions, there are still few potential areas for further work on this concept.

1) Extension to other media types:
We could explore whether this steganography method can be extended to work with other types of media, such as audio, video, medical image data, or EEG brain signal data.

2) Integration with other techniques:
We could investigate the feasibility of integrating this steganography method with other techniques, such as digital watermarking or error correction, to improve its performance or functionality.

3) Real-world application:
We could consider the practical applications of this steganography method and conduct experiments to see how it performs in a real-world setting.

Acknowledgment

The authors would like to express gratitude for the IJPCC reviewers for their contribution in studying the manuscript and find it acceptable for publishing. This paper is supported by International Islamic University Malaysia (IIUM) and partially through IRAGS18-030-0031.

Conflict of Interest

The authors declare that there is no conflict of interest.

References

[1] Macit, Hüseyin & Koyun, Arif & Güngör, Orhan. (2018). A REVIEW AND COMPARISON OF STEGANOGRAPHY TECHNIQUES. https://www.researchgate.net/publication/330162221_A_REVIEW_AND_COMPARISON_OF_STEGANOGRAPHY_TECHNIQUES

[2] Kanzariya, N. and Nimavat. (2013). Comparison of Various Images Steganography Techniques. https://www.academia.edu/10634704/Comparison_of_Various_Images_Steganography_Techniques

[3] Kadhim, I. J., Premaratne, P., Vial, P. J., & Halloran, B. (2018). Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. Neurocomputing, 335, 299–326. https://doi.org/10.1016/j.neucom.2018.06.075

[4] Bhattacharyya, S., Kshitij, A. P., & Sanyal, G. (2010). A novel approach to develop a secure image based steganographic model using integer wavelet transform. 2010 International Conference on Recent Trends in Information, Telecommunication and Computing. https://doi.org/10.1109/itc.2010.68

[5] Jiajia, Z., Jing, L., & Cheng, D. (2009). A steganographic method based on SOM and wavelet contrast. 2009 International Conference on Artificial Intelligence and Computational Intelligence. https://doi.org/10.1109/aici.2009.492

[6] Islam, A. U., Khalid, F., Shah, M., Khan, Z., Mahmood, T., Khan, A., Ali, U., & Naeem, M. (2016). An improved image steganography technique based on MSB using bit differencing. 2016 Sixth International Conference on Innovative Computing Technology (INTECH). https://doi.org/10.1109/intech.2016.7845020

[7] Cheddad, A. (2008, January 1). Enhancing steganography in digital images. 2008 Canadian Conference on Computer and Robot Vision. Retrieved June 25, 2022, from https://www.academia.edu/es/277018/Enhancing_Steganography_In_Digital_Image

[8] Nosrati, M., Hanani, A., & Karimi, R. (2015). Steganography in image segments using genetic algorithm. 2015 Fifth International Conference on Advanced Computing &amp; Communication Technologies. https://doi.org/10.1109/acct.2015.57

[9] Ali, T.S., & Doegar, A. (2015). A Novel Approach of LSB Based Steganography Using Parity Checker.

[10] Rawat, D., & Bhandari, V. (2013). A steganography technique for hiding an image in an image using LSB method for 24 bit color image. International Journal of Computer Applications, 64(20), 15–19. https://doi.org/10.5120/10749-5625

[11] ElSayed, A., Elleithy, A., Thunga, P., & Wu, Z. (2015). Highly secure image steganography algorithm using curvelet transform and DCT encryption. 2015 Long Island Systems, Applications and Technology. https://doi.org/10.1109/lisat.2015.7160204

[12] Senthooran, V., & Ranathunga, L. (2014). DCT coefficient dependent quantization table modification steganographic algorithm. 2014 First International Conference on Networks &amp; Soft Computing (ICNSC2014). https://doi.org/10.1109/cnsc.2014.6906644

[13] Gaba, J., & Kumar, M. (2013). Implementation of steganography using CES technique. 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013). https://doi.org/10.1109/iciip.2013.6707622

[14] Nofriansyah, D., Defit, S., Nurcahyo, G. W., Ganefri, G., Ridwan, R., Ahmar, A. S., & Rahim, R. (2018). A new image encryption technique combining Hill cipher method, Morse code and least significant bit algorithm. Journal of Physics: Conference Series, 954, 012003. https://doi.org/10.1088/1742-6596/954/1/012003

[15] Sridevi, D. R., Paruchuri,, V. L., & Rao, K. S. S. S. (2013). Image steganography combined with cryptography. INTERNATIONAL JOURNAL OF COMPUTERS &amp; TECHNOLOGY, 9(1), 976–984. https://doi.org/10.24297/ijct.v9i1.4160

[16] Narayana, S., & Prasad, G. (2010). Two new approaches for secured image steganography using cryptographic techniques and type conversions. Signal Image Processing : An International Journal, 1(2), 60–73. https://doi.org/10.5121/sipij.2010.1206

[17] Kumar, Shiva & K B, Raja & Chhotaray, R & Pattnaik, Sabyasachi. (2011). Performance Comparison of Robust Steganography Based on Multiple Transformation Techniques. International Journal of Computer Technology and Applications. 02.

[18] AbdelWahab, O. F., Hussein, A. I., Hamed, H. F., Kelash, H. M., Khalaf, A. A., & Ali, H. M. (2019). Hiding data in images using steganography techniques with compression algorithms. TELKOMNIKA (Telecommunication Computing Electronics and Control), 17(3), 1168. https://doi.org/10.12928/telkomnika.v17i3.12230

[19] Mahjabin, T., Hossain, S. M., & Haque, M. S. (2012). A block based data hiding method in images using pixel value differencing and LSB substitution method. 2012 15th International Conference on Computer and Information Technology (ICCIT). https://doi.org/10.1109/iccitechn.2012.6509770

[20] A.J. Umbarkar, R. Kamble, P., & V. Thakre, A. (2016). Comparative study of edge based LSB matching steganography for color images. ICTACT Journal on Image and Video Processing, 06(03), 1185–1191. https://doi.org/10.21917/ijivp.2016.0173

[21] Lin, W.-B., Lai, T.-H., & Chang, K.-C. (2021). Statistical feature–based steganalysis for pixel-value differencing steganography. https://doi.org/10.21203/rs.3.rs-512243/v1

[22] Brundick, F. S., & Marvel, L. M. (2001). Implementation of spread Spectrum Image Steganography. https://doi.org/10.21236/ada392155

[23] Abboud, G., Marean, J., & Yampolskiy, R. V. (2010). Steganography and visual Cryptography in Computer Forensics. 2010 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering. https://doi.org/10.1109/sadfe.2010.14

[24] Doshi, R., Jain, P., Gupta, L. (2012). Steganography and Its Applications in Security, International Journal of Modern Engineering Research (IJMER), (2)6, 4634-4638, ISSN: 2249-6645

[25] Kaur, H., Rani, J. (2016). A Survey on Different Techniques of Steganography, MATEC Web of Conferences 57, 1-6, DOI: 10.1051/matecconf/20165702003

[26] Rakhi, P.G., Gawande, S. (2013). A Review on Steganography Methods, International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, (2)10, 4635-4638, ISSN: 2320 – 3765.

[27] Rabah, K. (2004). Steganography – The Art of Hiding Data, Information Technology Journal, 3(3), 245-269, ISSN:1682-6027.

[28] Chunfang Yang, Fenlin Liu, Xiangyang Luo, &amp; Ying Zeng. (2013). Pixel Group trace model-based quantitative steganalysis for multiple least-significant bits steganography. IEEE Transactions on Information Forensics and Security, 8(1), 216–228. https://doi.org/10.1109/tifs.2012.2229987

[29] Setiadi, D. R., Rustad, S., Andono, P. N., & Shidik, G. F. (2023). Digital Image Steganography Survey and Investigation (goal, assessment, method, development, and dataset). *Signal Processing*, *206*, 108908. https://doi.org/10.1016/j.sigpro.2022.108908

[30] Devi, S., Sahoo, M. N., Muhammad, K., Ding, W., & Bakshi, S. (2019). Hiding medical information in brain MR images without affecting accuracy of classifying pathological brain. *Future Generation Computer Systems*, *99*, 235-246.

[31] Devi, S., Sahoo, M. N., & Bakshi, S. (2020). A novel privacy-supporting 2-class classification technique for brain MRI images. *Biocybernetics and Biomedical Engineering*, *40*(3), 1022-1035.