# Implementation of Image Steganography and Steganalysis

Anjana K N, Shreedhar Bhat

*Abstract*— **Steganography is the art of hiding information in plain sight. The basic structure of Steganography is made up of three components: the "carrier", the message and the key. The carrier can be a painting, a digital image, an mp3, even a TCP/IP packet among other things. A key is used to decode the original message. This can be anything, from a password, a pattern or a black-light. Advantage of steganography over cryptography is that the intended secret message does not attract attention to itself as an object of scrutiny. Coding secret messages in digital images is by far the most widely used of all methods in the digital world of today. This is because it can take advantage of the limited power of the human visual system (HVS) [10]. Almost any plain text, cipher text, image and any other media that can be encoded into a bit stream can be hidden in a digital image. With the continued growth of strong graphics power in computers and the research being put into image based Steganography, this field will continue to grow at a very rapid pace. As other image processing technologies, Steganography can also be applied on both spatial domain and frequency domain.**

**One of the most common methods of implementation is Least Significant Bit Insertion, in which the least significant bit of every bytes is altered to form the bit-string representing the embedded file. Altering the LSB will only cause minor changes in color, and thus is usually not noticeable to the human eye [3].**

*Index Terms*—**Steganography, LSB technique, Steganalysis, Map modification**

## I. Introduction

STEGANOGRAPHY is the art of covered or hidden writing. The purpose of steganography is covert communication to hide a message from a third party. This differs from cryptography, the art of secret writing, which is intended to make a message unreadable by a third party but does not hide the existence of the secret communication. Two other technologies that are closely related to steganography are watermarking and fingerprinting [19]. These technologies are mainly concerned with the protection of intellectual property, thus the algorithms have different requirements than steganography. These requirements of a good steganography algorithm will be discussed below. In watermarking all of the instances of an object are "marked" in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection [21]. With fingerprinting on the other hand, different, unique marks are embedded in distinct copies of the carrier object that are supplied to different customers. This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties [19]. In watermarking and fingerprinting the fact that information is hidden inside the files may be public knowledge – sometimes it may even be visible – while in steganography the imperceptibility of the information is crucial [20]. A successful attack on a steganographic system consists of an adversary observing that there is information hidden inside a file, while a successful attack on a watermarking or fingerprinting system would not be to detect the mark, but to remove it [19].

There are many ways in which messages can be hidden in digital media. Several characteristics of sound can be altered in ways that are indiscernible to human senses, and these slight alterations, such as tiny shifts in phase angle, speech cadence, and frequency, can transport hidden information. Another digital carrier can be the network protocols. Covert Transmission Control Protocol by Craig Rowland, for example, forms covert communications channels using the Identification field in Internet Protocol packets or the sequence number field in Transmission Control Protocol segments.

Nevertheless, image and audio files remain the easiest and most common carrier media on the Internet because of the plethora of potential carrier files already in existence, the ability to create an infinite number of new carrier files, and the easy access to steganography software that will operate on these carriers. In general, steganography can be treated as a double-edged sword depending on who uses it and how. However, the ethical issues related to the utilization of information hiding techniques require consideration in a broader steganography context, which is beyond the scope of this work.

A digital image is described using a 2-D matrix of the color intensities at each grid point (i.e. pixel). Typically gray images use 8 bits, whereas colored utilizes 24 bits to describe the color model, such as RGB model. In Steganography system which uses an image as the cover, there are several techniques to conceal information. The spatial domain techniques manipulate the cover-image pixel bit values to embed the secret information. The secret bits are written directly to the cover image pixel bytes. Consequently, the spatial domain techniques are simple and easy to implement. The Least Significant Bit (LSB) is one of the main techniques in spatial domain image Steganography. In this work, along with LSB insertion method, a new technique of LSB steganography has been proposed which is an improvised version of one bit LSB technique.

## II. HISTORY

Although its roots lay in ancient Greece, steganography has continually been used with great success throughout history. Today steganography is being incorporated into digital technology. The techniques have been used to create the watermarks that are in our nation's currency, as well as encode music information in the ever-popular mp3 music file [3]. The aim of steganographic communication back then and now, in modern applications, is the same: to hide secret data (a steganogram) in an innocently looking cover and send it to the proper recipient who is aware of the information hiding procedure. What distinguishes historical steganographic methods from the modern ones is, in fact, only the form of the cover (carrier) for secret data. Historical methods relied on physical steganography – the employed media were: human skin, game, etc. Further advances in hiding communication based on the use of more complex covers, e.g. with the aid of ordinary objects, whose orientation was assigned meaning. This is how semagrams were introduced. The popularization of the written word and the increasing literacy among people had brought about methods which utilized text as carrier. The World Wars had accelerated the development of steganography by introducing a new carrier – the electromagnetic waves. Presently, the most popular carriers include digital images, audio and video files and communication protocols. The latter may apply to network protocols as well as any other communication protocol (e.g. cryptographic) [26].

Although steganography is an ancient subject, the modern formulation of it is often given in terms of the prisoner's problem proposed by Simmons [17], where two inmates wish to communicate in secret to hatch an escape plan. All of their communication passes through a warden who will throw them in solitary confinement should she suspect any covert communication [18].

The warden, who is free to examine all communication exchanged between the inmates, can either be passive or active. A passive warden simply examines the communication to try and determine if it potentially contains secret information. If she suspects a communication to contain hidden information, a passive warden takes note of the detected covert communication, reports this to some outside party and lets the message through without blocking it. An active warden, on the other hand, will try to alter the communication with the suspected hidden information deliberately, in order to remove the information [19].

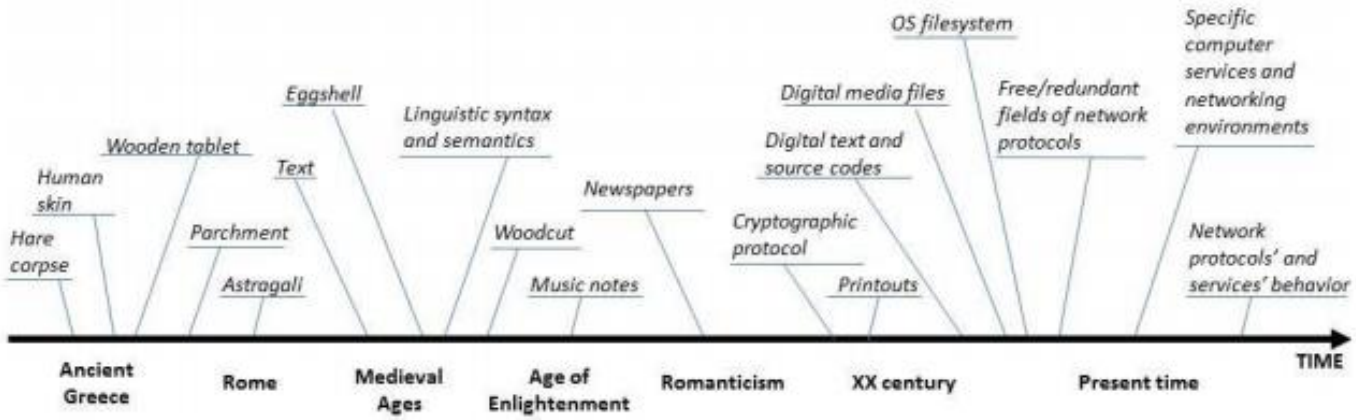


Fig. 1 Timeline of the evolution of hidden data carrier

## III. Architecture

steganography_medium = hidden_message + carrier + steganography_key

### *A. Carrier*

The carrier is the signal, stream, or data file into which the hidden data is hidden by making subtle modifications. Examples include audio files, image files, documents, and executable files. In practice, the carrier should look and work the same as the original unmodified carrier, and should appear benign to anyone inspecting it.
Certain properties can raise suspicion that a file is carrying hidden data:
If the hidden data is large relative to the carrier content, as in an empty document that is a megabyte in size.
The use of obsolete formats or poorly-supported extensions which break commonly used tools.

### *B. Chain*

Hidden data may be split among a set of files, producing a carrier chain, which has the property that all the carriers must be available, unmodified, and processed in the correct order in order to retrieve the hidden data.

### *C. Robustness*

Steganography tools aim to ensure robustness against modern forensic methods, such as statistical Steganalysis. Such robustness may be achieved by a balanced mix of:

- A stream-based cryptography process;
- A data whitening process;
- An encoding process.

## IV. Least Significant Bit Insertion

### *A. Introduction*

The technique works by replacing some of the information in a given pixel with information from the data in the image. While it is possible to embed data into an image on any bit-plane, LSB embedding is performed on the least significant bit(s). This minimizes the variation in colors that the embedding creates. For example, embedding into the least significant bit changes the color value by one. Embedding into the second bit-plane can change the color value by 2. In LSB approach, generally, the bits of the permuted secret binary sequence are distributed among the LSBs of consecutive pixels, starting from the Red plane (R-plane) of the top-left pixel moving towards the right modifying consecutive pixels on the row, and then move to the next row on the same plane and so on until modifying the 1st bit of the R-plane for all pixels. If the R-plane is not enough to host all secret bits, then the process continues in the same way on the Green and Blue planes (G-plane and B plane). If it is still not enough, repeat the procedure above replacing the 2nd LSBs of the RGB planes of all pixels, and continue replacing the 3rd LSBs if the 2nd LSBs of the RGB planes are enough to host the permuted sequence. The process terminated when all secret bits are hidden inside the cover Bytes of the cover image, subject to the constraint that the number of modified LSBs should be not more than 3.

### *B. Illustration*

In LSB encoding method, the least significant bits of the cover-image are altered so that they form the embedded information. Since the 8-bit letter A only requires eight bytes to hide it in, the ninth byte of the three pixels can be used to begin hiding the next character of the hidden message. The following example shows how the letter A can be hidden in the first eight bytes of three pixels in a 24-bit image.

Pixels: (00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
A: 01000001
Result: (00100110 11101001 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

The three underlined bits are the only three bits that were actually altered. LSB insertion requires on average that only half of the bits in an image be changed. Since the 8-bit letter A only requires eight bytes to hide it in, the ninth byte of the three pixels can be used to begin hiding the next character of the hidden message.
A slight variation of this technique allows for embedding the message in two or more of the least significant bits per byte. This increases the hidden information capacity of the cover-object, but the cover-object is degraded more, and therefore it is more detectable. Other variations on this technique include ensuring that statistical changes in the image do not occur. Some intelligent software also checks for areas that are made up of one solid color. Changes in these pixels are then avoided because slight changes would cause noticeable variations in the area [7,8].

### *C. Design and Implementation*

#### *1) Conversion of image to Matrix*

In the conversion process of image to matrix we convert the input cover image into its matrix values (24 bits per pixel)

#### *2) Conversion of text to bits*

The secret text message is converted to bits using its ASCII values.

#### *3) Embedding process*

The message is embedded into the intensity values of image obtained during image to matrix conversion with LSB insertion technique.

#### *4) Creating the image back from pixel matrix*

In this stage intensity values are converted back to image. The image obtained has message embedded into it. The cover image and the image obtained here have to be identical. Hence the objective of Steganography is satisfied.

#### *5) Decoding process*

In this process we extract the message which was embedded during embedding process.

### *D. Simulation and Results*

The image given below shows the original and LSB encoded image of a 300X300 sized image.

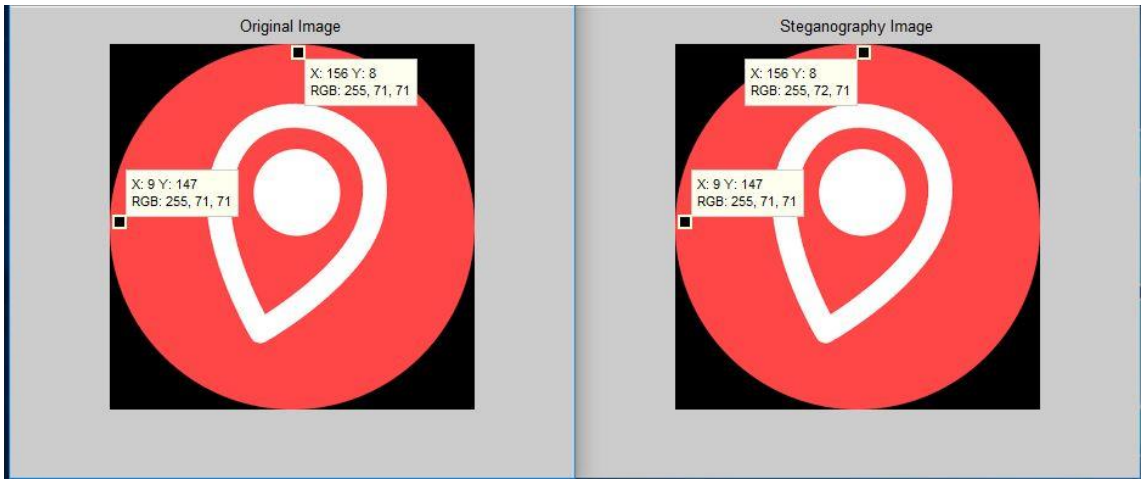


Fig 2.LSB encoding

### *E. Advantages and disadvantages*

With LSB Substitution it is quite easy to tell if an image has been Steganographed with an Enhanced LSB Attack. A complex image yields a much better Steganographed image—in that is harder to visually detect Steganography. Chi-Square Analysis can detect Steganography much better than Enhanced LSB's; however, one can still construct an image to account for statistical irregularities so that when applying Steganography to an image, they can make sure to preserve (as best as possible) the statistical frequencies so that a chi-square analysis fails to produce a qualified result.

Encryption of the message helps to improve the security of the message. Because of the encryption's poly-alphabetic nature it makes it a lot harder to crack.

## V. Steganalysis

Steganalysis is the study of detecting messages hidden using steganography. The goal of Steganalysis is to identify suspected packages, determine whether or not they have a payload encoded into them, and, if possible, recover that payload. Unlike cryptanalysis, where it is obvious that intercepted data contains a message (though that message is encrypted), Steganalysis generally starts with a pile of suspect data files, but little information about which of the files, if any, contain a payload.

The problem is generally handled with statistical analysis. A set of unmodified files of the same type, and ideally from the same source (for example, the same model of digital camera, or if possible, the same digital camera; digital audio from a CD MP3 files have been "ripped" from; etc.) as the set being inspected, are analyzed for various statistics. Some of these are as simple as spectrum analysis, but since most image and audio files these days are compressed with lossy compression algorithms, such as JPEG and MP3, they also attempt to look for inconsistencies in the way this data has been compressed. For example, a common artifact in JPEG compression is "edge ringing", where high-frequency components (such as the high-contrast edges of black text on a white background) distort neighboring pixels. This distortion is predictable, and simple steganography encoding algorithms will produce artifacts that are detectably unlikely.
One case where detection of suspect files is straightforward is when the original, unmodified carrier is available for comparison. Comparing the package against the original file will yield the differences caused by encoding the payload—and, thus, the payload can be extracted.

## VI. Map Modification Technique

### *A. Introduction*

The need for map modification arises due to the advancement in Steganalysis. In an image it is usually expected that the pixels with same shade of a color have same RGB values. But in the above mentioned technique, i.e. LSB encoding method the RGB values on a same shade of a color varies ,though not significantly, this can raise a suspicion that the image maybe steganography image. The sole purpose of steganography is hence lost. To overcome this problem we came up with a solution, though time consuming and tedious, it is less suspicion arousing.

### *B. Procedure*

The map modification method takes an image as an input and converts the RGB image to map-index image using a certain 'key' method. Later the map is encoded with 'message'. And then the image is converted back to RGB. Thus whole of the shade is changed and not just one pixel, while the other is kept is untouched as in the case of LSB encoding method.
That is, a 24-bit bitmap image would be converted to an 8-bit bitmap image while simultaneously encoding the desired hidden information. An algorithm would be created to select representative colors out of the 24-bit image to create the palette for the 8-bit image. This palette would then be optimized to an 8-bit color map that could be applied with minimal changes to the quality of the original image.

In our algorithm, each time a pixel is selected from the image, it is compared to every other color in the color map, and the minimum error between any two colors is calculated. If this error is lower than a certain error level (currently set at 20), then the new color is discarded and another color is selected from the image, otherwise it gets appended to the color map. The drawback of this algorithm is that the number of text bytes that can be encoded gets limited depending on the threshold. To increase the capacity, we developed another algorithm in which every shade of color present in the image is given unique position in the color map. However, the time consumption for encoding and decoding process is higher than the previous algorithm
.

### *C. Illustration*

Step 1: Take an image input and obtain its RGB values

| | | |
|---|---|---|
| 255 | 200 | 255 |
| 20 | 0 | 30 |
| 255 | 30 | 255 |

R

| | | |
|---|---|---|
| 50 | 255 | 50 |
| 70 | 0 | 30 |
| 50 | 30 | 50 |

G

| | | |
|---|---|---|
| 100 | 50 | 100 |
| 80 | 0 | 90 |
| 100 | 90 | 100 |

B

Step 2: Obtain the map and index form of the image

| | | |
|---|---|---|
| 255 | 50 | 100 |
| 200 | 255 | 50 |
| 20 | 70 | 80 |
| 0 | 0 | 0 |
| 30 | 30 | 90 |

MAP

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 3 | 4 | 5 |
| 1 | 5 | 1 |

INDEX

Step 3: Encode the data in ‘MAP’. [Here Data: 011111100100010]

| | | |
|---|---|---|
| 254 | 51 | 101 |
| 201 | 255 | 51 |
| 21 | 70 | 80 |
| 1 | 0 | 0 |
| 30 | 31 | 90 |

Encoded MAP

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 3 | 4 | 5 |
| 1 | 5 | 1 |

INDEX

| | | |
|---|---|---|
| 254 | 201 | 254 |
| 21 | 1 | 30 |
| 254 | 30 | 254 |

Encoded R

| | | |
|---|---|---|
| 51 | 255 | 51 |
| 70 | 0 | 31 |
| 51 | 31 | 51 |

Encoded G

| | | |
|---|---|---|
| 101 | 51 | 101 |
| 80 | 0 | 90 |
| 101 | 90 | 101 |

Encoded B

*D. Simulation and Results*


Fig 3. Map modification

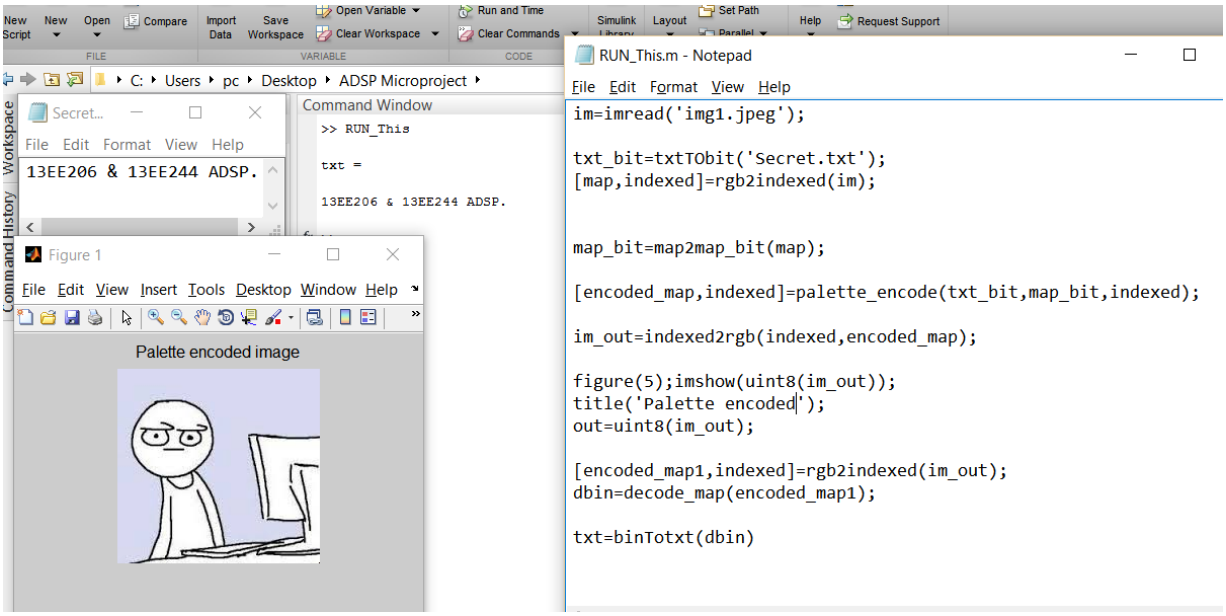


Fig 4. Screenshot of MATLAB simulation along with the secret message and the script file

*E. Advantages and Disadvantages*

The proposed technique provides four advantages as follows.

1) Information is embedded into compression code. The message will not be destroyed by compression processes.
2) The capacity of information hiding is higher than other schemes significantly.
3) The palette image is reversible and deals with the image distortion problem.
4) The stego-image could be compressed again by lossless compression method to save the storage spaces and speed up the transmission rate.

The obvious disadvantage is that the capacity does not depend on the image and is limited by the palette size.

## VII. Steganography Capacity Computation

The maximum image capacity (Cmax) in Byte of the BPIS algorithm can be determined as [9]:

$$Cmax = (s \cdot p \cdot W \cdot H) / b \quad (1)$$

Where, s represents the number of stego bits (limited to 3-bit in the BPIS algorithm), p represents the number of color planes (3 for RGB color image), W and H represent the width and height of the cover image in pixels, b represents the number of bits per Byte (bpb). For s=3, p=3, b=8, Cmax can be calculated as:

$$Cmax = 1.125\ W \cdot H \quad (2)$$

Another parameter was introduced in [11], namely, the occupation time (R), which is calculated by dividing the sizes of the hidden message (S) (secret message plus the stego header) by the maximum allowable image capacity (Cmax). It is expressed mathematically as:

$$R=(S/Cmax)\ X\ 100 \quad (3)$$

Where R is the occupation ratio, and S is the actual size of the hidden message (S=M+D, where M is the size of the secret message and D is the size of the stego header).

## VIII. Performance Evaluation

All steganographic algorithms have to comply with a few basic requirements. The most important requirement is that a steganographic algorithm has to be imperceptible [16].

*1) Invisibility*

The invisibility of a steganographic algorithm is the first and foremost requirement, since the strength of steganography lies in its ability to be unnoticed by the human eye. The moment that one can see that an image has been tampered with, the algorithm is compromised

*2) Payload capacity*

Unlike watermarking, which needs to embed only a small amount of copyright information, steganography aims at hidden communication and therefore requires sufficient embedding capacity.

*3) Robustness against statistical attacks*

Statistical Steganalysis is the practice of detecting hidden information through applying statistical tests on image data. Many steganographic algorithms leave a ‘signature’ when embedding information that can be easily detected through statistical analysis. To be able to pass by a warden without being detected, a steganographic algorithm must not leave such a mark in the image as be statistically significant.

*4) Robustness against image manipulation*

In the communication of a stego image by trusted systems, the image may undergo changes by an active warden in an attempt to remove hidden information. Image manipulation, such as cropping or rotating, can be performed on the image before it reaches its destination. Depending on the manner in which the message is embedded, these manipulations may destroy the hidden message. It is preferable for steganographic algorithms to be robust against either malicious or unintentional changes to the image.

*5) Independent of file format*

With many different image file formats used on the Internet, it might seem suspicious that only one type of file format is continuously communicated between two parties. The most powerful steganographic algorithms thus possess the ability to embed information in any type of file. This also solves the problem of not always being able to find a suitable image at the right moment, in the right format to use as a cover image.

*6) Unsuspicious files*

This requirement includes all characteristics of a steganographic algorithm that may result in images that are not used normally and may cause suspicion.

## IX. Other Techniques

### *A. Overview*

Steganography has received a significant attention from many researchers throughout the world, especially, after the tremendous development in computer and Internet technologies, and the growing concern about information security. Subsequently, many steganography approaches have been proposed and used to develop a huge number of steganography algorithms. In particular, there are four basic broad approaches that can be used to accomplish steganography; these are: Lease-Significant-Bit (LSB), injection, substitution and generation approaches [4, 5].

By using random factors and secret keys the security of steganography can be increased, but by considering the statistical characteristics of these images, most of these techniques will be fractured. However the least significant bits of the pixels looks random, practically they don't have random properties and represent some characteristics of the image .Evaluation on the properties of the image before and after steganography process, can indicate the changes in these least significant bits. As a result the application of steganography technique in spatial domain is not safe enough against the recent developing attacks.

Not all steganographic techniques rely on embedding directly into a pixel. Image formats, especially those that use compression, use mathematical functions to reduce the size of an image. The functions may also be manipulated slightly to allow for the embedding of data. Jsteg is one such algorithm which exploits the format of the JPEG image format to avoid embedding data directly into pixels [6].

### *B. JSTEG*

JPEG, which is a lossy image compression technique, successfully compresses image data by dividing an image into 8 by 8 blocks of pixels. The color assigned to these pixels is determined by a discrete cosine transform (DCT) table [4]. Instead of storing the color values that are assigned to each pixel, the JPEG format stores the coefficients of the cosines; representing the frequency of a value. Jsteg embeds its data into the coefficients which describes the values of the pixels. Because Jsteg embeds its data into the DCT table, the effect of embedding is spread among as many as 256 pixels [5]. Jsteg serves as an exemplary embedding strategy that avoids embedding the data directly into pixels.

### *C. Blocking*

Blocking works by breaking up an image into "blocks" and using Discrete Cosine Transforms (DCT). Each block is broken into 64 DCT coefficients that approximate luminance and color—the values of which are modified for hiding messages. Palette Modification replaces the unused colors within an image's color palette with colors that represent the hidden message. [11]

### *D. Spread Spectrum*

In spread spectrum techniques, hidden data is spread throughout the cover-image making it harder to detect [20]. A system proposed by Marvel et al. combines spread spectrum communication, error control coding and image processing to hide information in images [21]. Spread spectrum communication can be defined as the process of spreading the bandwidth of a narrowband signal across a wide band of frequencies [21]. This can be accomplished by adjusting the narrowband waveform with a wideband waveform, such as white noise. After spreading, the energy of the narrowband signal in any one frequency band is low and therefore difficult to detect [21]. In spread spectrum image steganography the message is embedded in noise and then combined with the cover image to produce the stego image. Since the power of the embedded signal is much lower than the power of the cover image, the embedded image is not perceptible to the human eye or by computer analysis without access to the original image [21].

### *E. Patchwork*

Patchwork is a statistical technique that uses redundant pattern encoding to embed a message in an image [22]. The algorithm adds redundancy to the hidden information and then scatters it throughout the image [25]. A pseudorandom generator is used to select two areas of the image (or patches), patch A and patch B [24]. All the pixels in patch A is lightened while the pixels in patch B is darkened [24]. In other words the intensities of the pixels in the one patch are increased by a constant value, while the pixels of the other patch are decreased with the same constant value [21]. The contrast changes in this patch subset encodes one bit and the changes are typically small and imperceptible, while not changing the average luminosity [25]. A disadvantage of the patchwork approach is that only one bit is embedded. One can embed more bits by first dividing the image

into sub-images and applying the embedding to each of them [23]. The advantage of using this technique is that the secret message is distributed over the entire image, so should one patch be destroyed, the others may still survive [25]. This however, depends on the message size, since the message can only be repeated throughout the image if it is small enough. If the message is too big, it can only be embedded once [22]. The patchwork approach is used independent of the host image and proves to be quite robust as the hidden message can survive conversion between lossy and lossless compression [23].

*F. A novel Genetic Algorithm (GA) evolutionary process was developed in [15] to secure steganography encoding on JPEG images.*

*G. A steganography algorithm for hiding text message using the LSB approach along with the concept of chaos theory was described in [12]. The algorithm provides security and maintains secrecy of the secret and provides more randomness*

*H. In order to maximize the storage of data inside the image, a steganography algorithm was proposed in [13] that utilized a pre-compression step. Further literature review can be found in [14].*

## X. Conclusion

Different approaches were reached in testing the application and several cover images with different text documents were sampled. Concentration was centred on encryption on text documents and embedding same in cover image to produce a stego-image. Normally after embedding the data into the cover image, the stego-image may loss its resolution. In the proposed approach, the image remains unchanged in its resolution as well as size as could be seen in the cover image and the stego-image above. Also it is invariant to lossless compressions, i.e, the embedded text message is not lost with lossless compression of the Stego image as opposed to the LSB encoding. Currently, the length of the message file has some limitations for the Audio Steganography, so for the same, we can have support for a wider size of files. This module can be further extended to use Video and audio files as carriers, and also to embed large pdf or word documents.

Choosing to modify values that have a small effect on the cover medium limits the ability to detect the embedding. Embedding strategies may be easily derived and implemented to complicate detection and inhibit the retrieval of the message by a third party, while still allowing easy retrieval by the intended recipient. LSB Embedding may be detected simply through visual inspection of an image and its bit-planes, or more reliably through methods which use statistical metrics to identify the likelihood an image contains hidden data. While an embedding may be detected, it may not be easily decoded, nor may a stego object be discovered due to the sheer number of images available. Steganography proves to be a significant technique for evading detection when communicating. The detection issues with steganography create challenges for security systems in attempting to prevent the transmission of steganography content. As the need to communicate in secret will always exist, steganography will likely continue to play an important role enabling covert communication.

## XI. References


[1] A Cheddad, J.Condell, K. Curranand P. Mc Kevitt, “Digital Image Steganography: Survey and Analyses of Current Methods”. Signal Processing, Volume 90, Issue 3, March 2010, Pages 727-752.

[2] K Kesslet, Gary C. An Overview of Steganography for the Computer Forensics Examiner, Burlington, 2004

[3] Mamta Juneja, Parvinder S. Sandhu, and Ekta Walia, “Application of LSB Based Steganographic Technique for 8-bit Color Images”, World Academy of Science, Engineering and Technology 26 2009.

[4] Gonzalez, Rafael C., and Paul A. Wintz. "Image Compression Standards." Digital Image Processing. 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002. 492-510. Print.

[5] Westfeld, A., & Pfitzmann, A. (n.d.). Attacks on Steganographic Systems: Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools — and Some Lessons Learned, 1-16.

[6] Aaron Miller,Thesis,“Least Significant Bit Embeddings: Implementation and Detection”, May 2012

[7] Hiding secrets in computer files: steganography is the new invisible ink, as codes stow away on images-An article from: The Futurist by Patrick Tucker.

[8] Ismail Avcıbas¸, Member, IEEE, Nasir Memon, Member, IEEE, and Bülent Sankur, Member, "Steganalysis Using Image Quality Metrics," IEEE Transactions on Image Processing, Vol 12, No. 2,February 2003.
[9] Hussein Al-Bahadili, "A Secure Block Permutation Image Steganography Algorithm", the International Journal on Cryptography and Information Security (IJCIS), 30th July 2013.
[10] Sans Institute, InfoSec reading room, "A detailed look at Steganographic Techniques and their use in an Open-Systems Environment"
[11] Adnan M. Shihab, Raghad K. Mohammed, and Woud M. Abed ,University of Baghdad, Baghdad, Iraq, "EVALUATING THE PERFORMANCE OF THE SECURE BLOCK PERMUTATION IMAGE STEGANOGRAPHY ALGORITHM ", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.5, September 2013.
[12] S. Bhavana and K. L. Sudha. Text Steganography Using LSB Insertion Method Along with Chaos Theory. International Journal of Computer Science, Engineering and Applications (IJCSEA), Vol.2, No.2, pp. 145-149, April 2012.
[13] Rosziati Ibrahim and Teoh Suk Kuan. Steganography Algorithm to Hide Secret Message inside an Image. Journal of Computer Technology and Application, Vol. 2, pp. 102-108, 2011. Ajdkaaaa
[14] Hussein Al-Bahadili. A Secure Block Permutation Image Teganography Algorithm. Submitted to the International Journal on Cryptography and Information Security (IJCIS) on 30th July 2013.
[15] A. M. Fard, M. M. R. Akbarzadeh-T, and F. Varasteh-A. A New Genetic Algorithm Approach for Secure JPEG Steganography. Proceedings of the IEEE International Conference on Engineering of Intelligent Systems, pp. 1-6, Islamabad, Pakistan, 2006.
[16] T. Morkel , J.H.P. Eloff , M.S. Olivier, Information and Computer Security Architecture (ICSA) Research Group Department of Computer Science University of Pretoria, 0002, Pretoria, South Africa, "AN OVERVIEW OF IMAGE STEGANOGRAPHY"
[17] Simmons, G., "The prisoners problem and the subliminal channel", CRYPTO, 1983.
[18] Chandramouli, R., Kharrazi, M. & Memon, N., "Image steganography and steganalysis: Concepts and Practice", Proceedings of the 2nd International Workshop on Digital Watermarking, October 2003.
[19] Anderson, R.J. & Petitcolas, F.A.P., "On the limits of steganography", IEEE Journal of selected Areas in Communications, May 1998.
[20] Wang, H & Wang, S, "Cyber warfare: Steganography vs. Steganalysis", Communications of the ACM, 47:10, October 2004.
[21] Marvel, L.M., Boncelet Jr., C.G. & Retter, C., "Spread Spectrum Steganography", IEEE Transactions on image processing, 8:08, 1999
[22] Johnson, N.F. & Jajodia, S., "Exploring Steganography: Seeing the Unseen", Computer Journal, February 1998.
[23] Petitcolas, F.A.P., Anderson, R.J. & Kuhn, M.G., "Information Hiding – A survey", Proceedings of the IEEE, 87:07, July 1999.
[24] Bender, W., Gruhl, D., Morimoto, N. & Lu, A., "Techniques for data hiding", IBM Systems Journal, Vol 35, 1996.
[25] Johnson, N.F. & Jajodia, S., "Steganalysis of Images Created Using Current Steganography Software", Proceedings of the 2nd Information Hiding Workshop, April 1998.
[26] E. Zielinska, W. Mazurczyk, K. Szczypiorski, **The Advent of Steganography in Computing Environments** - In: Computing Research Repository (CoRR), abs/1202.5289, arXiv.org E-print Archive, Cornell University, Ithaca, NY (USA), published on 23 February 2012.