STEGANOGRAPHY MESSAGING TOOL

To embed and conceal secret data within an image transmission medium through the use of steganographic embedding.

INSTITUTE of TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

Institute of Technology Carlow
C00241234 – Luke Byrne

Table of Contents

Abstract:

Steganography is the habit of obscuring the presence of sender-receiver communication from a external third-party organisation by the insertion and concealing of secretive data within a particular transmission medium. A transmission medium is any file format such as an image, video, audio, PDF, or text file which is used to essentially transmit and contain the embedded steganographic information. Steganographic embedding algorithms are used in order to embed data within carrier files in a secretive manner. In this research document I will outline the several types of steganographic embedding algorithms and techniques in which are utilized in modern cryptography. This research document will also identify the complexities which have arisen over the years from past research of the implementation of steganographic embedding algorithms along with multiple types of carrier images.

Introduction:

Steganography is a technique used to conceal secretive information or messages inside a particular transmission medium which is not kept secret [1]. The main objective of the use of steganography is to ensure that the transmission of data or information from sender to receiver is not intercepted by third party through the use of a covert channel allowing the receiving end to extract the message in secrecy. The difference between steganography and cryptography is that concept of steganography is to conceal the existence of a message being sent over a covert communication channel [2]. Cryptography differs in comparison to steganography as the purpose of cryptography is to provide a secure communication end to end channel to protect the transmission of data from an adversary through the use of encryption algorithms [7].

There are many different forms of steganography such as image, video, audio, and text-based steganography [2]. Steganography relies on the use of a file container more commonly known as transmission medium. Each form of steganography relies on embedding algorithms in order to compute a successful steganographic message.

The main objective of this research paper is to obtain a high-level understanding of the different forms of steganography and their relevant embedding algorithms and to be able to choose transmission medium and embedding technique for my final fourth year project.

The Earliest known history of Steganography

The ancient Greeks used the earliest forms of steganography. In 440 B.C the Greek ruler Histaeus developed and utilized the earliest method of steganography [3]. The first method involved the melting of the wax from wax tablets then the engravement of the message on the surface of the melted wax tablet. This particular methodology of steganography was used by Demerstus (The king of Spartans) to inform the Spartans of the conquering by the Xerxes [3]. A secret message was etched into the piece of wood then the melted wax was then applied back onto the wood. By reapplying the wax onto the wax tablet, this made the message unnoticeable, making it look like the wax tablet was blank. The wax tablet with the hidden message was then securely delivered to the receiver then the person on the receiving end would melt the wax back off the tablet in order to retrieve the secret message beneath the layer of wax.

Another methodology of steganography which was performed by the ancient Greeks was by scalping a slave's head and tattooing a message on the slave's head. When the slave's hair had grown back, the message could no longer be visible. The slave was then sent to the receiver of the message and then their head was then shaved again in order to reveal the secret message [4]. The clear objective of steganography from a historical perspective was to transmit a message in secret to the receiver to hide the message from the unwanted middleman.

Historical evidence has shown that there were multiple other forms of steganography recorded before the digital age. An example of another form of steganography used before the digital age was the use of invisible inks. The ancient romans began using invisible inks to write messages in the middle of lines on an existing piece of writing on paper which was not kept secret. The invisible ink in which the message was written consisted of substances such as milk, urine, and fruit juice such as lemon [5]. After the message was written, the person on the receiving end would heat the paper to turn the invisible ink brown, transforming the message into a readable format or they would coat the page with dust to reveal the invisible writing.

Figure 1 : Wax tablet containing steganographic message

What is a transmission medium from a computing perspective?

A transmission medium is the specific channel in which carries hidden steganographic information which is transmitted from the sender to the receiver. Cover medias can also be known as a transmission medium or carrier files, which can include any digital file format such as audio files, exe files, PDF documents, text files, video files, and image file formats such as (PNG, GIF, JPEG, BMP) [6].

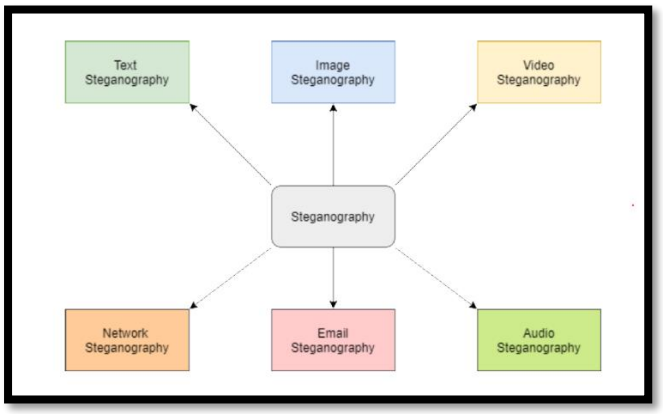

Figure 2: Different digital transmission mediums

The different formations of steganography

1. Audio Steganography

Audio steganography is the technique used to obscure the presence of embedded secretive data within an audio transmission medium (MP4, MP3, WAV, FLAC) [8]. Audio steganography is considered to be one of the hardest methods of steganography to implement, therefore it is a lot harder to achieve a robust steganographic embedding system depending on which embedding methodology is used [11]. There are three different embedding techniques in which can used to embed steganographic data into an audio file such as LSB substitution, echo hiding, and parity coding [9].

1.1 Least Significant Bit

Least significant bit substitution (LSB) is a technique used for the embedding of steganographic data into an image transmission medium. LSB embedding requires the cover file to be read in and separated bit by bit in a binary representation. After the bit planes are separated the least significant bits of the file will need to be determined. The least significant bits of the file are considered to be the far most right bit of a byte (every 8^{th} bit) [10]. After the least significant bits have been determined, the secret data being embedded will be substituted for them becoming embedded using steganography within the audio file meaning that every least significant bit will contain the embedded message. In order to retrieve the embedded message, the least significant bits of the file will need to be extracted and represented into a human readable string format.

1.2 Echo Hiding Coding

Echo hiding is an audio steganographic embedding technique whereby an echo is initiated within the discrete signal. Before the echo is implemented the original signal is segmented into separate blocks. An echo is then produced and then the encoding phase begins. During the encoding phase, only one bit of data can be encoded per echo [10]. After the echo is encoded the separated blocks of the original signal and the encoded data are chained back together to form what is called the final signal. The main advantages of the implementation of echo hiding are that this method provides a high data transmission rate when embedding and it is more robustness than other embedding methods [9].

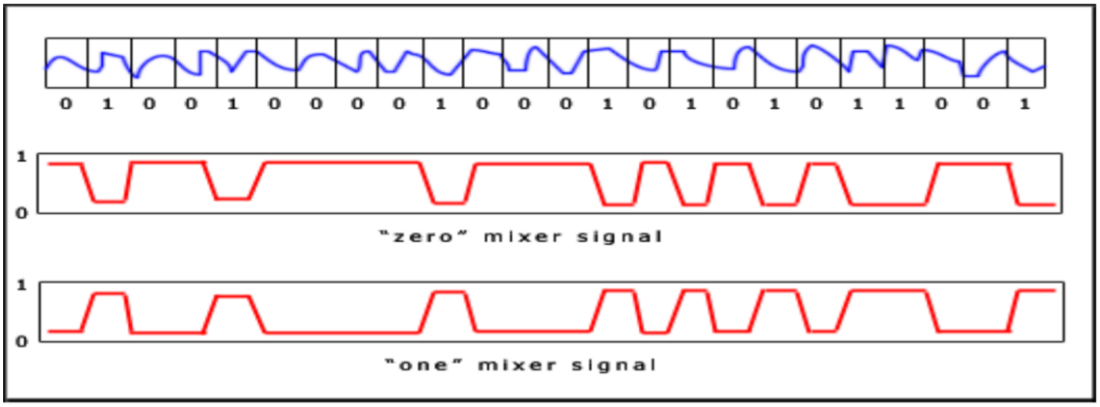

Figure 3: Echo Hiding Coding

1.3 Parity Coding

Parity Coding involves the breaking down of a signal into multiple sample regions. Each parity bit of each sample region is substituted for the secret message in which is being embedded into the audio file. In order to embed data into the audio file using parity coding, the parity bit of the sample region must match the parity bit of the secret data which is being embedded. If the parity bit of the sample regions does not match the individual bit of secret embedding data, the least significant bit of the sample region will need to be inverted in order for the embedding to be successful [9].

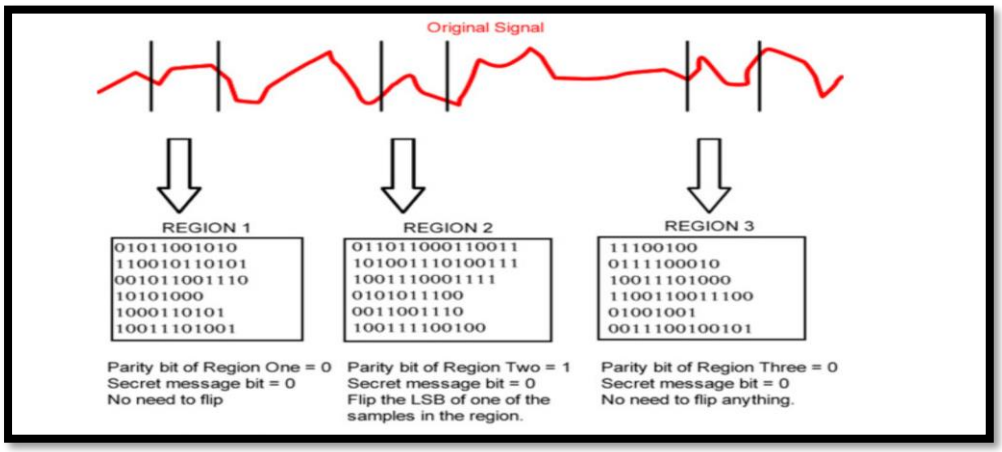

Figure 4: Parity Coding

2. Text Steganography

Text steganography is the art of hiding a secret message or text within an existing piece of text. This can be done by hiding the secret message in every nth character position of every word of the text file. When you read every nth character of each word, you should be able to see a pattern to retrieve the hidden secret. Text steganography is very rarely used as text files contain an extremely small proportion of redundant data [12]. Text steganography can also involve changing the format of a text, the creation of random character sequences and the utilization of context free grammars to construct a piece of text. Text steganography is categorised into the following three different methodologies, format based, random statistical generation and linguistic methods.

Here is an example of text steganography where every nth character contains the embedded message:

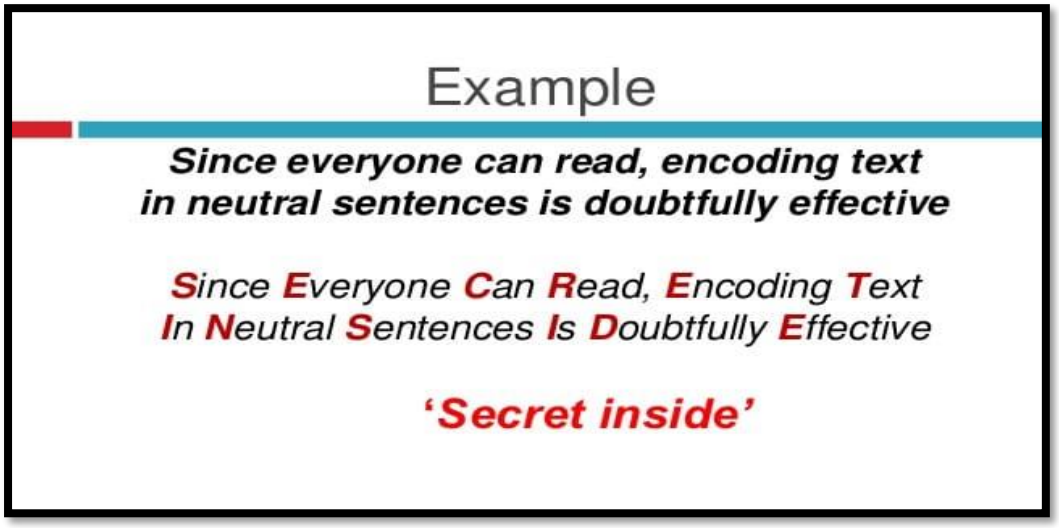

Figure 5: Text Steganography

2.1 Format Based Method

Format based steganography is the physical adjusting of a texts format in order to conceal data using steganography. This method has a large number of weaknesses as the physical adjustments made to the original text in order to hide the secret message can be easily noticed when the text is analysed using a word editor such as MSWord or LibreOffice. Spelling mistakes, extra white spaces and the use of varied sizes and types of fonts can be easily noticed in text editor [12].

2.2 Random and Statistical Method

Random and statistical generation is a method used to initiate cover text by concealing and embedding secret data within certain character sequences or word sequences in order to embed the data in the most random way possible to prevent a third party from observing the secret message [12]. A second technique used to create cover text is to obtain and evaluate the statistical properties of letter frequency and word length to construct words that have the same statistical properties having lexical value.

2.3 Linguistic Method

Linguistic method steganography involves the hiding of information through the utilization of natural language text [13]. By utilizing the natural language text, some words of the secret text are changed to their relevant synonym through synonym substitution and the rearrangement of words while the meaning of the text still remains consistent. Since natural language processing technologies are not able to generate purposeful natural text easily, most steganographic systems which follow the linguistic approach tend to use pre-existing texts as the cover text then the grammatical properties are used to embed the secret message within the cover text using an embedding algorithm and an extraction algorithm would be used to retrieve the embedded message. The two categories in which linguistic method steganography can be subdivided into is syntactic transformation and semantic transformation. Syntactic transformation is the transforming of a sentence into its equivalent meanings through the use of transformation techniques like passivation, clefting, and topicalization while the meaning remains the same [13]. Semantic transformation is a more advanced method of transformation as this requires a more in-depth analysis of the natural language. The semantic method is used to manipulate words in a sentence to their corresponding synonym without altering its meaning.

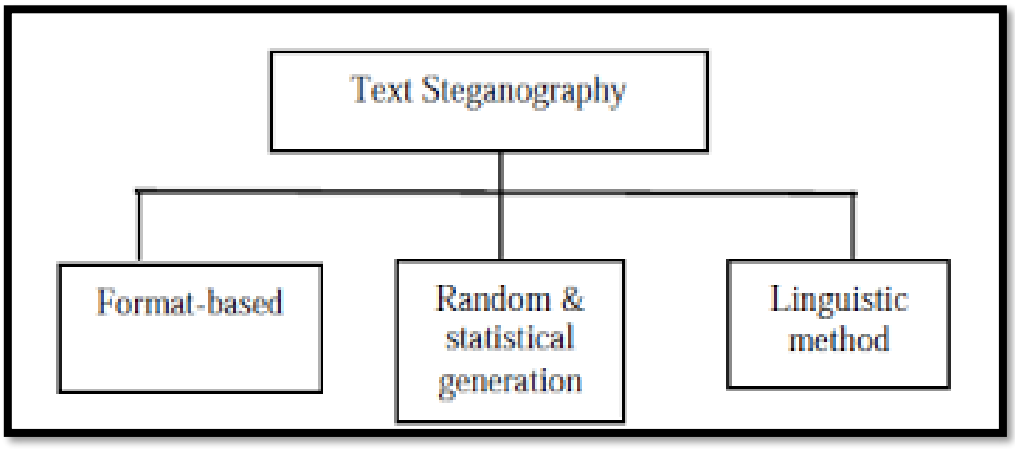

Figure 6: Methods of Text Steganography

3. Video Steganography

Video steganography is the technique used to embed a secret message or text within the contents of a video file without a third party being able to distinguish that the secret message is being transmitted. Steganography within a video or sound file is a much more challenging transmission medium to use to embed and hide data rather than using steganography in an image file as the steganography performance can be impacted based on the embedding efficiency and the embedding payload.

The embedding efficiency is the quantity of data in which can be embedded into the transmission medium (video or image file) whilst creating the least distortion making it more difficult for a third party to assume there is secret data embedded in the file therefore the higher the embedding efficiency, the less distortion there will be to the transmission medium [15]. The embedding payload is the capacity of data which will be embedded into the transmission medium. The higher the embedding payload is, the quality of the transmission medium (video file) will be decreased. Both of these factors oppositely affect each other. Usually when the embedding efficiency is maximised, the embedding payload will be smaller and when the embedding payload is larger the embedding efficiency is reduced.

Video files consist of both audio and video frames, meaning that the secret data which is been embedded is being embedded using audio steganography and video steganography, that way the embedding capacity is increased [14]. The embedding data can be an image, audio, or text. A benefit of this method is that the embedding capacity is increased and there is less reliance on compression, filtering, cropping and rotation [15].

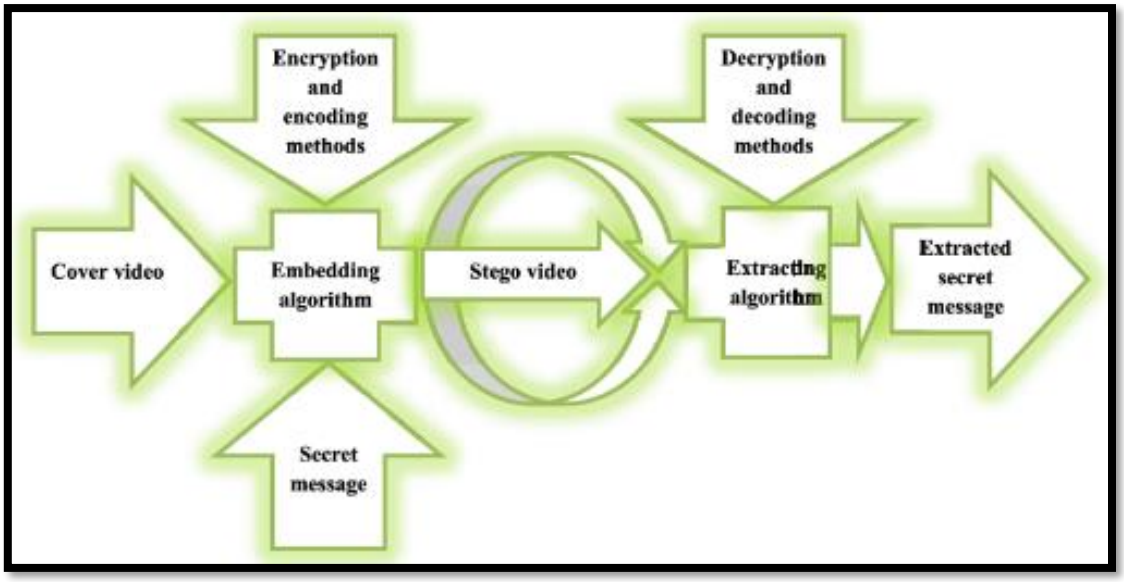

Figure 7: Video steganography embedding and extraction algorithm process

4. Image Steganography

The most common transmission medium to use for steganography is an image. Since there are multiple types of image file formats there are many different steganographic algorithms used to support them. Image steganography consists of concealing a message within an existing image file. When the secret data is embedded within the image file decoding will need to take place for the receiving end to retrieve the file [16].

Computers recognize images as a collection of integers which compose multiple light intensities in different sectors of the image [8]. A grid is formed from this numeric collection and the single coordinates denotes the pixels. Images on computers are made up of a grid of pixels typically represented in bits which are the squares of color displayed horizontally. The bit depth is the number of bits in a color scheme used for each pixel. The minimum bit depth for color arrangements is 8, therefore each pixel consists of 8 bits which is the equivalent of 1 byte which is used to represent the color of each pixel. Monochrome and greyscale images can contain 256 different variations of the shade of grey and each pixel consists of 8 bits.

Digital color images are contained in 24-bit files, and they utilize the RGB model which is also identified as true color. The color variations of the pixels in a 24-bit file image originate from the three primary colors red, green, and blue (RGB). Each individual primary color consists of 8 bits and there are 256 different variants of each primary color (Red, Green, Blue) meaning that there is a combined sum of 16 million different colors. This means that one single pixel could contain more than 16 million different color arrangements [8].

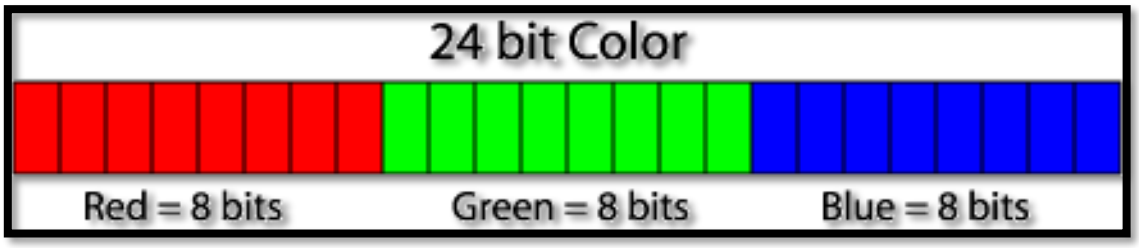

Figure 8: 24-bit image pixel

4.1 Image compression

Image compression is the process of reducing the size of an image files contents into smaller file size by making minimal noticeable changes to the quality and the attributes of the file. Compression is a technique generally used to condense larger images which obtain a bigger bit depth to store and transmit the image data in a more increased efficient manner [8].

Lossy and lossless are the two categoric types of image compression. These two compression techniques have the same objectives to reduce the size of the file being stored, but their formulas vary with each other. Lossy compression minimally alters the attributes of the image file by permanently eliminating any redundant data from the image to create a smaller file. The alterations made to the original file are minute, therefore the changes are too miniscule to be noticed by the naked eye [17]. The image file format JPEG (Joint Photographic Experts Group) utilizes this compression methodology.

Lossless compression varies greatly from lossy compression as this compression method does not involve the elimination of any data from the image file while still reducing the size of the file [17]. Essentially the integrity and the quality of the image remains consistent which was compressed using lossless compression as the lossless compressed image can be rebuilt to its original state when it is decompressed as all bits of the image remain the same [8]. An example of an image file format which uses the lossless compression technique is GIF (Graphical Interchange Format) and BMP (MS Windows Bitmap file).

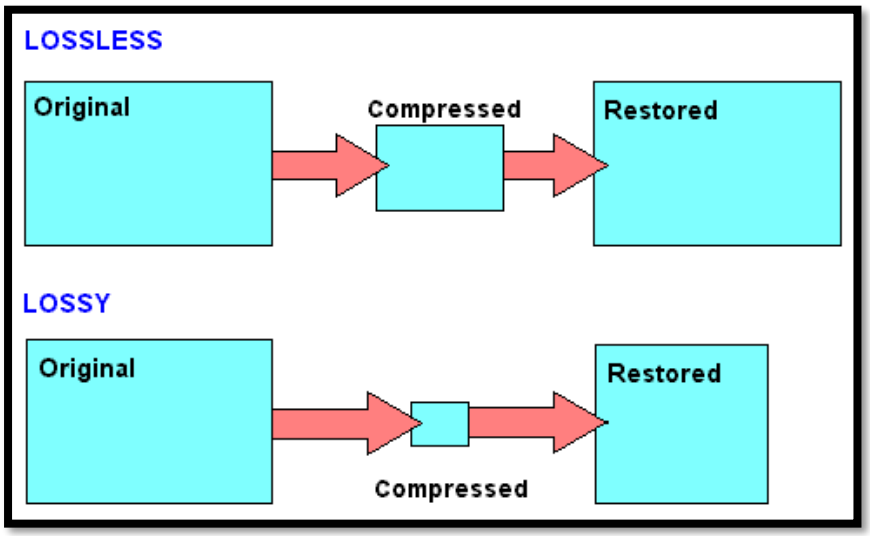

Figure 9: Lossy and lossless compression techniques

4.2 Methods of image steganography

There are multiple steganography embedding techniques in which are widely available used for image steganography such as spatial domain methods and transform domain methods.

4.2.1 Spatial Domain Method

Spatial domain, also known as image domain methodologies, involve the manipulation and embedding of data in digital image pixels through filtering in order to modify or amplify an image in a specific manner [18]. Spatial domain methodologies are heavily reliant on the use of lossless compression techniques as spatial domain methods involve the insertion of data into bits into the image. Lossless compression plays a vital role when using spatial domain techniques as this compression technique allows the embedded data to remain in image whereas lossy compression would result in the embedded data being identified as redundant data leading to it being permanently removed [20].

4.2.2 Transform Domain Method

Primarily, transform domain methods involve the transforming of the image prior to the embedding of secret data into the image occurs. The insertion and embedding of the secret data into the image transmission medium require has a more complexed procedure than spatial domain methods as it requires the use of algorithms, and image transforms in order to embed secret data into areas of the cover image with the most significance. Since data is being embedded within the most significant bits of the cover image through the use of transform domain methodologies, the embedded data is more robust and protected from image processing, compression and cropping in comparison to spatial domain methods [19]. If the secret data was embedded into the least significant bits of the cover image it has more exposure to compression, therefore the data could be seen as redundant data and become permanently eliminated from the cover image.

Comparison of Spatial Domain and Transform Domain

Methodology	Spatial Domain	Transform Domain
Cost of operation	*Less*	*More*
Level of complexity	*Less Complex*	*More Complex*
Robustness	*Less Robust*	*More Robust*
Speed	*Faster*	*Slower*
Computational Time	*Less Time*	*More Time*
Vulnerable To Compression	*Yes*	*No*

Table 1: Comparison of Spatial Domain and Transform Domain Methodologies

Spatial Domain Steganography Techniques

5. (LSB) – Least Significant Bit

The least significant bit is the most common spatial method in which is used to embed secret data into a cover image. Least Significant bit embedding is the most simple but effective approach taken to conceal a message within an image transmission medium. This method allows you to conceal a message in the least significant bits of the image file without making any noticeable changes to the image and its quality as the changes are too minute to be seen by the naked eye [18]. Although the changes in the image and its quality are minute the LSB method is considered to be less robust than other transform domain methods as LSB embedded data can be exposed to lossy compression, image processing and compression as it is a spatial method.

The least significant bit method works by substituting the least significant bits of the cover image with the embedding payload or secret data which you are going to conceal through steganographic means within the cover image being used as the transmission medium. 24-bit images use the RGB model, which is the three primary colors red, green, and blue. Pixels obtain their colors variations from the three primary colors of the RGB model. Each primary color of a pixel contains 8 bits, therefore there are 24 bits in one single pixel. The least significant bit within a byte is generally the 8th bit. When using the least significant bit method to embed secret data into the pixels of the cover image there are three least significant bits. This means that in every pixel will have three least significant bits and each of these bits will be substituted for the data in which is being embedded into the cover image. To embed more data into the cover image, sometimes the second least significant bit can also be substituted for the payload. This is not recommended but it can be achievable depending on the specific image being used.

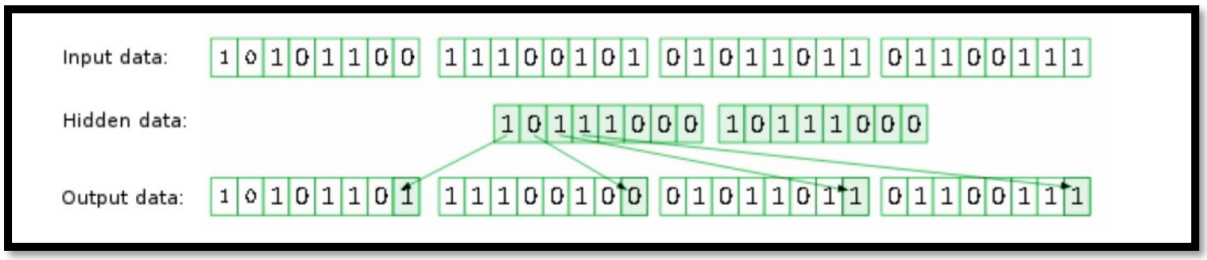

Figure 10: LSB Embedding

5.1 Example of Least Significant Bit Embedding:

Here below is a grid of 3 pixels of a 24-bit image represented in binary format. In order to embed the number 242 into the pixels (cover image) the number 242 will need to be converted into its relevant binary representation format then each bit of the binary output will need to be inserted into the least significant bits of the 3 pixels in order to conceal the secret data, while also creating the least noise as possible in order to prevent the detection of the hidden alterations to the cover image. The number 242 in binary equates to the following in binary → 11110010. The secret data will be embedded into the last bit of every byte as this is the bit which is considered to be the least significant [20].

Original Output:

00101101	**00011100**	**11011100**
10100110	**11000100**	**00001100**
11010010	**10101101**	**01100011**

When the number 242 represented binary data is embedded into the least significant bits in the grid of pixels, the result should be as below.

Steganographic Output:

00101101	**00011101**	**11011101**
10100111	**11000100**	**00001100**
11010011	**10101100**	**01100011**

5.2 LSB Embedding and Retrieving Algorithms

Firstly, before performing the LSB embedding and retrieving algorithm an appropriate image format must be chosen to be used as the cover image. In my opinion the most appropriate image file format to use when embedding data using the Least Significant Bit algorithm would be a Portable Network Graphics (PNG) image which is found to be the most popular form of image on the internet [20]. Since LSB is a spatial domain technique, this methodology is reliant on lossless compression techniques. This file format is most suited to the LSB algorithm because PNG's use lossless compression techniques which means that no data is eliminated from the image file and the quality of the image is not distorted after compression is used. Lossless compression is ideal for Least Significant Bit embedding as the data embedded will never be omitted from the file as data is never lost as a result of lossless compression.

5.2.1 Embedding Algorithm - To embed data into an image using the Least Significant Bit embedding algorithm firstly the image file should be read in bit by bit to determine in which the least significant bits of the image file are by separating the binary output into its necessary the bit planes. When the least significant bits of the pixels have been determined, the secret data or message will be substituted for the least significant bits becoming embedded into the cover image [21].

5.2.2 Extraction Algorithm – To retrieve/extract the secret data which was embedded into the cover image using the least significant bit embedding algorithm the image containing the embedded steganography will need to be read in bit by bit again and separated into its relevant bit planes. After the binary data has been read in you will need to iterate through all the least significant bits of the pixels in which the message was initially embedded into. The least significant bit is the last bit (8^{th} bit) of every byte in each pixel. After iterating through the least significant bits of the cover images pixels the message should be extracted [21].

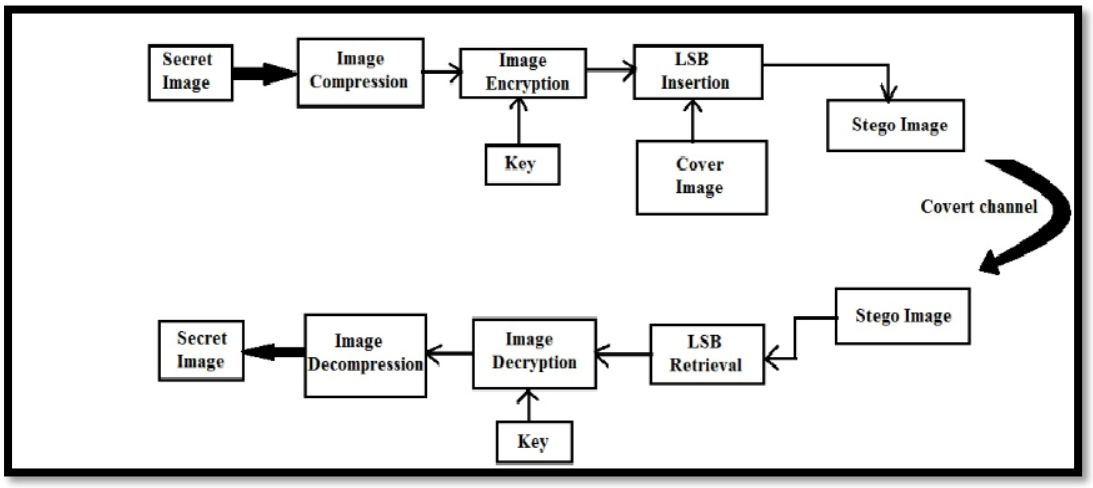

Figure 11: LSB Embedding & Extraction algorithm

5.3 LSB Quality & Performance Measurement Formulas

There are multiple metrics in which are used to calculate the efficiency, robustness and the quality of the steganography performed within a particular cover image. The robustness of a steganographic image is determined by resistance of the image against different vulnerabilities or attacks such as steganalysis which is the technology used for the detection of steganography within an image through the extraction and manipulation of embedded data. To measure the quality of a steganographic image the two equations used are the mean square error (MES) equation and the peak signal to noise ratio (PSNR) equation.

5.3.1 Mean Square Error (MSE) – This formula is used to obtain the mean square error which is the result of the amount of error between the original cover image and the resulting imaging containing the embedded steganographic data. The mean square error formula illustrates the average errors between the original image and the steganographic image by comparing the values of the pixels of the original image to the resulting image containing the steganography [21]. The closer the resulting answer of the mean square error (MSE) of the two images is to 0 "zero" the better the MSE result is.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{m} \sum_{j=1}^{n} [C(i,j) - C'(i,j)]^2$$

Figure 12: MSE Equation

5.3.2 Peak Signal to Noise Ratio (PSNR) – In order to calculate the PSNR, you will need to obtain the Mean Square Error (MSE) of the images. The PSNR formula is used to compute the ratio of the maximum pixel value on a decibel scale. The maximum pixel value is the pixel in which has the most effect on the quality of the rest of the pixels [21]. Generally, a low quality steganographic image would result in the PSNR value being less than 30 decibels and a high quality steganographic image would result in a PSNR value of equal to or greater than 40 dB (decibels) [19]. The resulting PSNR is expressed in dB which is the unit of decibels. The higher the resulting PSNR value, the better the quality of the steganography within the image as there is less distortion cause to the original image. Here below is the formula used to calculate the PSNR of a steganographic image.

$$PSNR = 10 \times \log_{10}\left(\frac{m^2}{MSE}\right)$$

Figure 13: PSNR Equation

5.4 Least Significant Bit Steganography with Bitmap Images (BMP)

Bitmap (BMP) files are images simply represented as an array of binary data which correlates to the values of the pixels within the image. The binary data stored within the rows and columns is then represented as in a graphical representation as pixels. Bitmap file formats utilize lossless compression techniques which means the file format ideal for using the least significant bit embedding algorithm to embed secretive data into the image. Bitmap image formats have an extremely high embedding capacity as they have a greater bit depth than most other image file formats, meaning that more bits can be substituted using the least significant bit algorithm. The disadvantage of being able to a high capacity of data into a BMP file is that the changes are a lot more obvious and can be seen very easily by the naked eye. In this day and age bitmap images are very seldom used on the internet, therefore these image file formats are known to raise more suspicions to least significant bit embedding. Since BMP file formats are known to raise suspicions, there are more file format alternatives which can be used to embed steganographic data using the least significant bit algorithm.

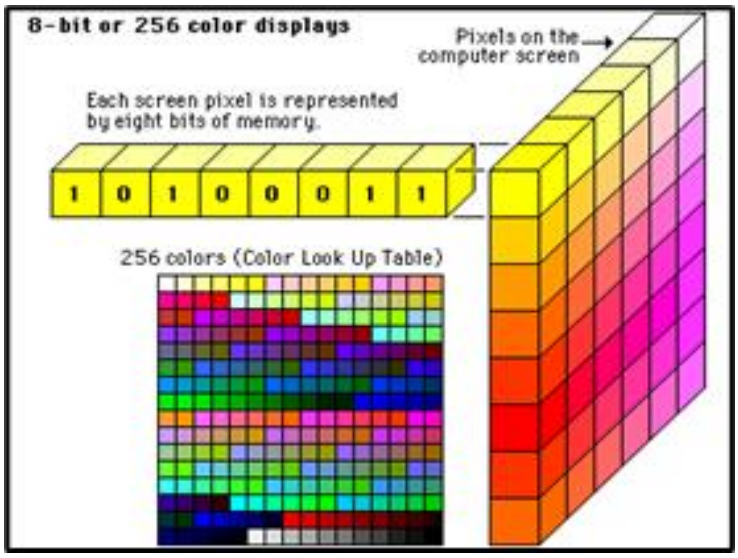

Figure 14: BMP Bit Depth

5.5 Least Significant Bit Steganography with Palette Based images

A palette-based image is common utilized image format on the web. Palette based images are extremely limited to the number of color variants in which can be stored in them because there are only a maximum of 256 colors of choice available. By reducing the number of colors available, this reduces the overall required storage needed for the image. An example of a palette-based images file format is Graphics Interchange Format (GIF). The GIF image format's bit depth is capped at 8-bits meaning that each pixel is only 8 bits, and it can only store a maximum of 256 colors. Colors in which are used in a GIF image are stored in a palette called a color lookup table [8]. Each pixel in a GIF takes up one byte of memory and each pixel an index for the color palette. To reduce the overall lookup time, this color palette is ordered from the most frequently used color to the least frequently used color [22].

Embedding data using the least significant bit algorithm into the least significant bits of a GIF can have many disadvantages. Since a GIF's bit depth is only 8 bits in length the embedded data can totally alter the colors of the pixels in an image meaning that the steganography can be very easily detected within the image essentially because the indexes referring to the RGB color in the palette are being substituted to the data being embedded [8].

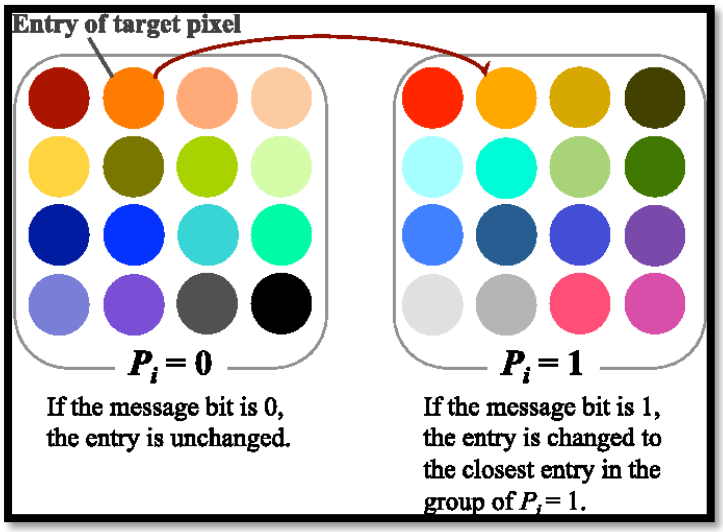

Figure 15: GIF Pallet Lookup table

5.5.1 Remdediations to detection within palette-based images

A remediation to the detection of steganography within palette-based images can be achieved using palette modification. This involves the arrangement of the the color palette into a specific order so that the color variations amongst neighboring colors within the lookup table are minimized [22]. When the color palette has been modified the secret data being embedded into the least significant bits of the palette-based cover image (GIF) will be embedded using an index referring to a color in the new sorted palette. Another method to obscure any changes made to the GIF image would be to add more colors to the palette that are closer shades to existing colors in the palette. Either of these methods can be used to minimize the detection of embedding and minimize the amount of distortion caused by the embedding of data into a palette-based image. The use of a greyscale images to embed data using LSB is also another way to make detection harder within images because the bit depth for a greyscale image is 8 bits and there are 256 different variations of grey in which can be used. The use of 256 different shades of grey makes steganography tremendously hard to be detected within a greyscale/monochrome image [8].

Figure 16: Pallet modification

Transform Domain Steganography Techniques

6. Transform Domain methodologies

Transform domain methods are much more complexed methods used for embedding steganography into cover medias rather than spatial domain embedding methods. While transform domain methods are more complex, they also have more advantages than spatial domain methods too. Transform domain techniques are more resistant to compression techniques, more robust against statistical attacks and image manipulation meaning that the steganographic data being embedded is less distinguishable even using steganalysis techniques and tools. Transform domain methodologies are more mathematical based hence why these methods are a lot more complex and long winded. There are a variety of different transform domain methodologies such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Discrete Fourier Transform.

6.1 Discrete Cosine Transform (DCT)

Frequency domain embedding is used for embedding secret data into the significant frequency values while discarding the higher frequency values [16]. Transformations are then performed, and the transform coefficients are substituted accordingly before the data is embedded. To ensure that the compression efficiency remains stable, each transform coefficient is encoded. The secret data is then embedded within the least significant bits of of the coefficients of the cover image. Pixels are then sorted into 8X8 segments called blocks and these blocks are transformed into 64-DCT coefficient's [8]. Each DCT coefficient is altered to embed the secret data within it but any block containing a coefficient value of 0 is not altered as this could distort the entire image exposing the embedded steganographic data as the image will become distorted [16].

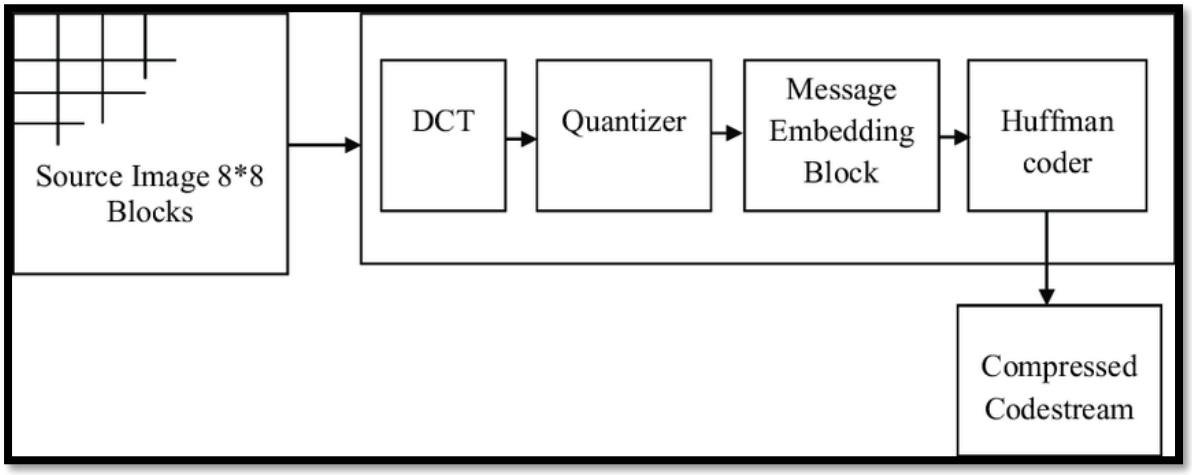

Figure 17: DCT Formula

6.2 Discrete Wavelet Transform (DWT)

Wavelets are known as small wave frequencies in which provide frequency and spatial description of and image. Wavelets are generated by the translations and the dilations of the mother wavelet, which is a fixed function. The DWT method decomposes the secret data and the cover image and segments the pixels into 4X4 blocks of data. The 4X4 blocks of the secret data and the cover image is then compared with each other and error blocks are produced and embedded within the coefficients of the most appropriate boxes.

6.3 Spread Spectrum

This embedding method involves the distribution of the steganographic data spread throughout the image being used. The purpose of spreading the embedding data throughout the cover image is to improve the robustness of the steganography within the image. By improving the quality of the steganography in the image the overall robustness of the image is greatly increased against statistical analysis attacks and image manipulation. A narrowband signal is modulated with white noise and is harmonized over multiple frequencies of a cover image [8]. When spreading occurs, the narrowband signal blended with any one of the frequencies is low which means that it is not easily detectable. To embed steganographic data into a carrier image the noise created contains the steganographic embedding data and it is then modulated with the carrier image to construct the steganographic image. The power of an embedded signal is drastically lower than the carrier images power therefore the presence of steganography within the embedded image produced is less detectable by the naked eye or through the means of statistical steganalysis techniques such as Benford's law or machine learning techniques.

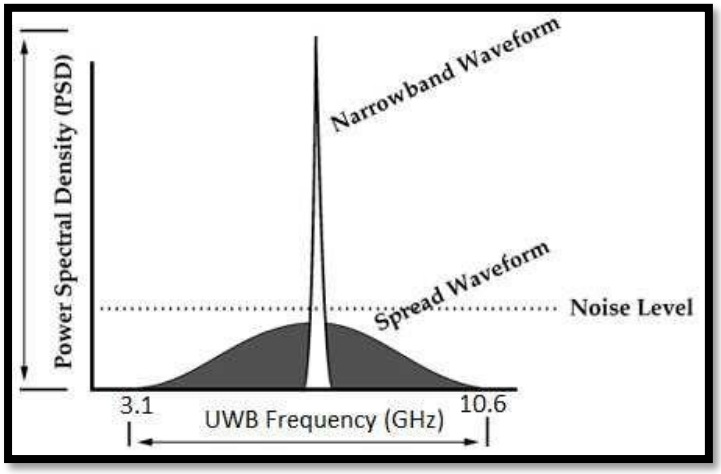

Figure 18: Spread Spectrum Embedding

6.4 JPEG Compression

JPEG files are the most usual format for image files to be found on the internet as they are small in size. JPEG files utilize lossy compression. Lossy compression is used to omit redundant least significant bits of images in order to reduce the overall size of the file [20]. Therefore, previously it was believed that steganography could not be performed successfully through the use of JPEG cover images as steganographic data embedded into the least significant bits of the image file could be eliminated through lossy compression. To overcome the issue of losing embedded data from compression, transform domain methods came into practical use to ensure data remains after compression.

In order to compress a Joint Photographic Experts Group (JPEG) image, the RGB color data needs to be transformed into a YUV representation. YUV formation is a representation where the Y components are referred to the luminance and the U and V components correspond to the color (Chrominance). Recent studies have proven that the human eye is more capable to determine changes in brightness rather than colors within an image. The color components are then down sampled into order to reduce the overall size of the file as well as the U and V are horizontal and vertically halved in order to decrease file size further. After this has been done the JPEG image gets transformed using the discrete cosine transform (DCT). The DCT method transforms a signal from an image format into a frequency formation by using a blocking method which transforms and groups pixels into 8X8 blocks into 64 discrete cosine coefficients. Quantization compression is conducted where all values in the blocks are divided by a quantization coefficient in order to minimize the strength of higher frequency brightness within the image [8]. After the quantization compression is complete the results are rounded up to its nearest integer value and Huffman coding is used to encode the coefficients further reducing the image size through compression.

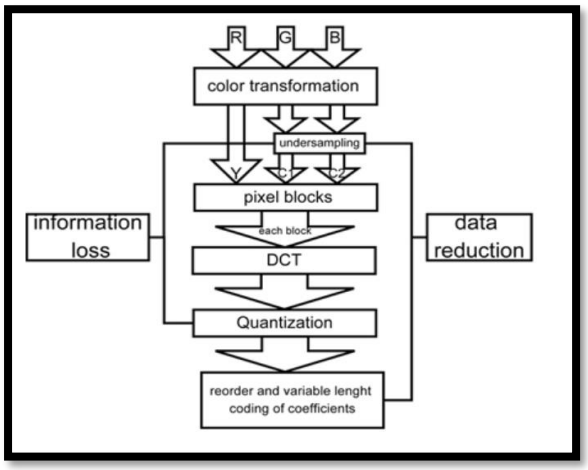

Figure 19: JPEG Compression

6.5 Patchwork

Patchwork can be considered as a transform domain methodology or a spatial domain methodology. Patchwork uses redundant pattern encoding, meaning that the steganographic embedding data is scattered around in the patches generated in the carrier image. Before the data is embedded into the carrier image redundancy is added to the embedding message. Two patches (Patch A & Patch B) are generated by a pseudorandom generator. The light intensity of the pixels in patch A are lightened with a constant value and the light intensity of the pixels in patch B are darkened with the same constant value. Within the patch subsets only one bit is encoded, meaning that changes are minute and imperceptible while also not effecting the overall average luminosity. An advantage of patchwork embedding is that the embedded message is robust against lossy and lossless compression, as the embedded message is present across the entirety of the carrier image [8].

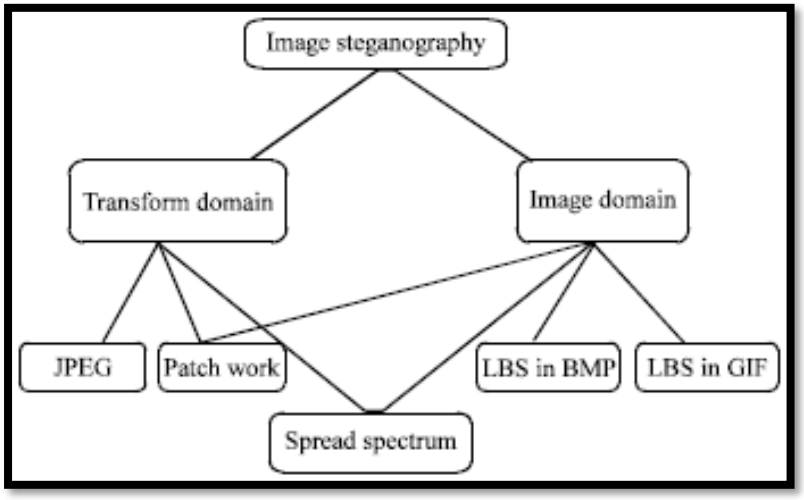

Figure 20: Transform domain & Spatial domain embedding techniques

Factors which influence a strong unobtrusive embedding technique.

From the evaluation of all the above spatial and transform embedding techniques it is clear that many factors can influence the strengths and weaknesses of these particular embedding algorithms mentioned above. Each embedding technique discussed in this research document takes different approaches to embed data, therefore the strengths and weaknesses of each technique will vary. To evaluate how protected an embedding technique is against detection, visual attacks, statistical attacks, and image manipulation the technique must obtain the following requirements mentioned below.

Imperceptibility:

The imperceptibility of an image containing steganography is determined by the amount of distortion caused by the embedding algorithm to the overall visibility and quality of the image. The imperceptibility of an embedding algorithm can be compromised when a user notices that the image has been potentially altered.

Robustness Against Statistical Analysis Attacks:

The main purpose of using steganography is to obscure the presence of embedded steganographic data within a specific transmission medium ensuring that the communication of the data is hidden from a third-party during transit from sender to the receiver. However, statistical analysis techniques can easily detect the existence of steganography within a cover image as steganography embedding algorithms leave a signature behind [8]. For a steganographic embedding algorithms to be robust against statistical analysis they must not leave signatures in the image.

Embedding Payload Capacity:

The embedding payload relates to how much data can be embedded into the carrier image. This is particularly important when embedding a message into the cover medium because if there is not adequate embedding capacity, the message will be too big to be embedded.

Robustness Against Image Manipulation:

Some embedding techniques substitute the least significant redundant bits of the cover image for the embedding data. Since some embedding algorithms rely on the use of redundant bits in the cover image this could lead the hidden message to being tampered or destroyed as the steganography data embedded may be susceptible to alterations such as cropping or rotating.

Independent of file format:

A strong and secure steganographic embedding algorithm can be determined whether it has the capability to embed and securely conceal data within any form of transmission medium as it does not rely on any particular file format.

Unsuspicious files:

When using an embedding algorithm, it is important that the algorithm you choose does not increase the size of the file drastically as this may raise suspicions leading to the embedded steganographic data being exposed if the steganography is discovered by steganalysis or human analysis.

	LBS in BMP	LSB in GIF	JPEG Compression	Patchwork	Spread Spectrum
Imperceptibility	***High***	***Medium***	***High***	***High***	***High***
Payload Capacity	***High***	***Medium***	***Medium***	***Low***	***Medium***
Robustness against statistical attacks	***Low***	***Low***	***Medium***	***High***	***High***
Robustness against Image manipulation	***Low***	***Low***	***Medium***	***High***	***Medium***
Independent of file format	***Low***	***Low***	***Low***	***High***	***High***
Unsuspicious files	***Low***	***Low***	***High***	***High***	***High***

Table 2 : The comparison between different image steganography embedding tools

Steganography tools:

Xioa Steganography

Xiao Steganography is the most renowned free open-source software used to implement steganography within BMP image file formats and WAV files which was developed by Nakasoft [23]. Xiao steganography not only allows the implementation of steganography within BMP files and WAV files as it also supports the extraction of the embedded data/message from the file. This tool is very simple to operate from a user perspective as it requires the user to simple select a cover media being used such as a BMP or WAV file, enter the message to be concealed within the file then choose an encryption algorithm such as (RC4, 3DES, RC2, Triple DES 112) or a hashing algorithm such as (SHA, MD5, MD4, MD2) in order to password protect the cover media being used. In order to extract the message in which was embedded using steganography, the file must be read in by the Xiao Steganography tool [26]. When the file is read in by Xiao steganography it will be decrypted, and the hidden information hidden within the file will be decoded allowing the user to retrieve the original message [25].

Steghide

Steghide is a popular open-source tool licensed by GNU used to embed data into cover medias such as JPEG, BMP, AU files, and WAV. This tool is considered to be a bit more complex towards other tools available as the tool has a command line interface (CLI). Since this tool is CLI based users will need to follow the command line instructions online at http://steghide.sourceforge.net/development.php in order to embed data into the cover medias mentions above and to extract the embedded data [24]. All embedding data is encrypted securely using AES 128 symmetric encryption with CBC mode in order

for the integrity of the message to remain. As well as encryption this tool is compression resistant as it uses transform domain embedding techniques. Steghide is available across all Linux distributions and Windows.

OpenStego

OpenStego is a free open-source steganography tool developed in Java which obtains multiple functionalities such as data embedding and watermarking. This tool provides the function to conceal data through steganographic techniques within file formats such as PNG, BMP, JPG, JPEG, GIF, and WBMP. When embedding data into any of the mentioned file formats the cover media is password protected using AES 128- or 256-bit encryption. Randomized LSB embedding is used to embed and conceal the message within the necessary chosen cover media. This tool is supported across all platforms as it is developed in Java [26].

Tool	Cover medias supported	Encryption used	Embedding algorithm type
Xiao Steganography	BMP or WAV file	(RC4, 3DES, RC2, Triple DES 112) & (SHA, MD5, MD4, MD2)	LSB Embedding – Spatial Domain
Steghide	JPEG, BMP, AU files, WAV	AES 128 with CBC mode	JPEG compression – Transform Domain
OpenStego	PNG, BMP, JPG, JPEG, GIF, WBMP	AES 128 & 256 bit	Randomized LSB – Spatial domain

Table 3: The comparison between different steganography tools

Conclusion:

To conclude, all necessary formations of steganography were discussed in depth within this research paper. A high-level description of each steganographic embedding and retrieving algorithm was briefly outlined in the sections where the formations of steganography were mentioned. After describing the different embedding and retrieving algorithms along with the type of steganography in which the algorithm was compatible with, I highlighted the various file types in which these algorithms support in order for the steganographic technique to function correctly and conceal hidden data within a carrier file in the safest manner possible whilst also not degrading the quality of the carrier file. Different transform mediums use embedding algorithms which have different strengths and weaknesses, therefore I discussed other alternative transmission mediums in which could be used to implement steganography with. From my evaluation of different types of steganography image steganography was considered to be the most popular form of steganography used as image transmission mediums have a wider variety of embedding techniques to choose from and it is generally easier to embed steganographic data within an image rather than an audio, video, or text file. Not all embedding algorithms are safe to use as multiple factors can have a positive or negative affect on the quality of the steganographic output. These factors can have positive or negative effects on the quality of the steganography, the capacity of the embedding payload and the robustness of the steganographic file against detection, image manipulation and compression attacks.

Since this research document was heavily based on image steganography, I carried out a lot of analysis between the different image embedding algorithms in order evaluate and determine a conclusion to which algorithm would be most suited for my final year project.

Bibliography:

[1] Semilof, Margie, and Casey Clark. “What Is Steganography? - Definition from Searchsecurity.” SearchSecurity, TechTarget, 6 July 2021, https://searchsecurity.techtarget.com/definition/steganography.

[2] Stanger, James. “The Ancient Practice of Steganography: What Is It, How Is It Used and Why Do Cybersecurity Pros Need to Understand It.” Default, 2 Nov. 2021, https://www.comptia.org/blog/what-is-steganography.

[3] Siper, A., Farley, R. and Lombardo, C., 2005. The rise of steganography. Proceedings of student/faculty research day, CSIS, Pace University.

[4] Sellars, Duncan. "An introduction to steganography." (2007).

[5] The National Archives. “How to Make Invisible Ink.” The National Archives, The National Archives, 8 Mar. 2021, https://www.nationalarchives.gov.uk/education/families/celebrating-british-science-week/how-to-make-invisible-ink/.

[6] Codr, J., 2009. Unseen: An Overview of Steganography and Presentation of Associated Java Application C-Hide. Retrieved January, 8, p.2010.

[7] Rout, H. and Mishra, B.K., 2014. Pros and cons of cryptography, steganography and perturbation techniques. IOSR Journal of Electronics and Communication Engineering, pp.76-81.

[8] Morkel, T., Eloff, J.H. and Olivier, M.S., 2005, June. An overview of image steganography. In ISSA (Vol. 1, No. 2, pp. 1-11).

[9] Malviya, S., Saxena, M. and Khare, D.A., 2012. Audio steganography by different methods. Int. J. Emerg. Technol. Adv. Eng, 2(7), pp.93-98.

[10] Jayaram, P., Ranganatha, H.R. and Anupama, H.S., 2011. Information hiding using audio steganography–a survey. The International Journal of Multimedia & Its Applications (IJMA) Vol, 3, pp.86-96.

[11] Tekeli, K. and Asliyan, R., 2017. A comparison of echo hiding methods. The Eurasia Proceedings of Science Technology Engineering and Mathematics, (1), pp.397-403.

[12] Agarwal, M., 2013. Text steganographic approaches: a comparison. arXiv preprint arXiv:1302.2718.

[13] Chang, C.Y. and Clark, S., 2014. Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. Computational linguistics, 40(2), pp.403-448.

[14] Prof. Dr. P. R. Deshmukh, and Bhagyashri Rahangdale. “Data Hiding Using Video Steganography.” International Journal of Engineering Research & Technology, IJERT-International Journal of Engineering Research & Technology, 19 Apr. 2014, https://www.ijert.org/data-hiding-using-video-steganography.

[15] Bandyopadhyay, Prof.Samir Kumar. “(PDF) Various Methods of Video Steganography.” ResearchGate, https://www.researchgate.net/publication/328610598_VARIOUS_METHODS_OF_VIDEO_STEGANOGRAPHY.

[16] Bhallamudi, S., 2015. Image Steganography Final project–Report. In Tech. Rep.. Wright State University.

[17] About Adam HarknessIn his role at NetMotion Software. “How Does Data Compression Work?” NetMotion Software, 18 Jan. 2021, https://www.netmotionsoftware.com/blog/connectivity/how-does-data-compression-work.

[18] Hariri, M., Karimi, R. and Nosrati, M., 2011. An introduction to steganography methods. World Applied Programming, 1(3), pp.191-195.

[19] Hamid, N., Yahya, A., Ahmad, R.B. and Al-Qershi, O.M., 2012. Image steganography techniques: an overview. International Journal of Computer Science and Security (IJCSS), 6(3), pp.168-187.

[20] Sinha, Bharat. “Comparison of PNG & JPEG Format for LSB Steganography - IJSR.” Comparison of PNG & JPEG Format for LSB Steganography, International Journal of Science and Research (IJSR), https://www.ijsr.net/archive/v4i4/29031501.pdf.

[21] Viraktamath, S.V., Kinagi, B., Kannur, K., Pavan, M.S. and Hunagund, V., 2018, July. Performance analysis of steganography for hiding miscellaneous data using Daubechies wavelet. In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 846-850). IEEE.

[22] Chandramouli, R., Kharrazi, M. and Memon, N., 2003, October. Image steganography and steganalysis: Concepts and practice. In International Workshop on Digital Watermarking (pp. 35-49). Springer, Berlin, Heidelberg.

[23] Zeki, A.M., Ibrahim, A.A. and Manaf, A.A., 2012. Steganographic software: analysis and implementation. International Journal of Computers and Communications, 6(1), pp.35-42.

[24] Kundu, D. and Upreti, A., 2018, October. Study of various steganography tools. In 2018 International Conference on Automation and Computational Engineering (ICACE) (pp. 117-120). IEEE.

[25] Bhatia, N. and Kaur, G., Xiao Steganography

[26] Pilania, U., Tanwar, R. and Gupta, P., 2021. Performance Analysis of Open Source Image Steganography Tools. Journal of Engineering and Applied Sciences, 8(1), pp.50-67.

Table of figures:

List of Tables: