Journal of the ACS, Vol. 7, May 2013

Hide and Seek: A White Paper in Steganography

Dr. Emad S. Othman
Senior Member IEEE - Region 8,
High Institute for Computers and Information Systems,
AL-Shorouk Academy, Cairo – Egypt,
PH- 0020-01025830256, E-mail: emad91@hotmail.com

Abstract:

Steganography, derived from the Greek, literally means, "Covered writing". It is the science of hiding information in ways that prevent the detection of hidden messages. Digital technology gives us new ways to apply steganographic techniques, including one of the most trickery that of hiding information in multimedia likes image, audio and video. It includes a huge array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications. Steganography and Cryptography are cousins in the spycraft family. Cryptography scrambles a message so it cannot be understood while Steganography hides the message so it cannot be seen. A message in ciphertext, for instance, might arouse suspicion on the part of the recipient while an "invisible" message created with steganographic methods will not.

The purpose of the presented white paper is to provide momentary information in the area of Steganography with illustrations of some methods for Steganography. Providing and evaluating steganographic methods with experiments. In this article image files are discussed and how to hide information in them

Keywords: Cover image, Embedding, Extraction, Steganography, Secret message.

1. Evolved History of Steganography

Steganography is an art, which dates back to ancient times [1]. It has been used by ordinary people, spies, rulers, governments, armies, …. etc down through the ages. It is the original method of information concealment. Information has been hidden in drawings, paintings, books, newspapers, in speech, in written word, even in postage stamps.

The Greeks, from the histories of Herodotus, wrote text on wax-covered tablets. In one story, Demeratus wanted to notify Sparta that Xerxes intended to invade Greece. To avoid capture, he scraped the wax off the tablets and wrote a message on the underlying wood. He then covered the

tablets with wax again. The tablets appeared to be blank and unused so they passed inspection by guards without question.

The Chinese, would often shave the head of a messenger and tattoo a message or image on the messengers head. After allowing his hair to grow, the message would be undetected until the head was shaved again.

The Egyptians*,* used illustrations to conceal messages. The idea being that one person could send the illustration to the other in reasonable confidence that if the messenger was questioned then the illustration would not arouse any interest from his enemies. This is a form of Steganography.

Another common form of invisible writing is using Invisible inks. Such inks were used with much success as recently as WWII. An innocent letter may contain a very different message written between the lines. Early in WWII steganographic technology consisted almost exclusively of invisible inks. Common sources for invisible inks are milk, vinegar, fruit juices and urine. All of these are darken when heated.

With the improvement of technology and the ease as to the decoding of these invisible inks, inks that are more sophisticated were developed which react to various chemicals. Some messages had to be "developed" much as photographs are developed with a number of chemicals in processing labs [2]. Null ciphers (unencrypted messages) were also used. The real message is "camouflaged" in an innocent sounding message.

An example of a message containing such a null cipher is:

> Fi**s**hing fr**e**shwater be**n**ds an**d** sa**l**twater co**a**sts re**w**ards an**y**one fe**e**ling st**r**essed. Re**s**ourceful an**g**lers us**u**ally fi**n**d ma**s**terful le**a**pers fu**n** and ad**m**it sw**o**rdfish ra**n**k ov**e**rwhelming an**y**day.

By taking the third letter in each word, the following message emerges:

> **Send Lawyers, Guns, and Money.**

Another example of a message containing such a null cipher actually sent by a German Spy in WWII is:

> A**p**parently n**e**utral's p**r**otest i**s** t**h**oroughly d**i**scounted a**n**d i**g**nored. I**s**man h**a**rd h**i**t. B**l**ockade i**s**sue a**f**fects p**r**etext f**o**r e**m**bargo o**n** b**y** products, e**j**ecting s**u**ets a**n**d v**e**getable o**i**ls.

Taking the second letter in each word the following message emerges:

> **Pershing sails from NY June 1.**

Usually the secret information is concealed by the use of an innocuous cover as to not arouse suspicion if hostile agents discover the cover. As an example, the cover text:

> I'**m** f**e**eling r**e**ally s**t**uffy. E**m**ily's m**e**dicine w**a**sn't s**t**rong e**n**ough w**i**thout a**n**other f**e**brifuge.

If the reader retains the second letter of each word in sequence, it hides the sentence:

> **Meet me at nine.**

With many methods being discovered and intercepted and with every discovery of a message hidden using an existing application, a new steganographic application is being devised. There are even new twists to old methods. Drawings have often been used to conceal or reveal information. It is simple to encode a message by varying lines, colors or other elements in pictures. Computers take such a method to new dimensions. Techniques for concealing meta-information about a message, such as its existence, duration, sender and receivers are collectively known as traffic security. Steganography is often considered to be a proper subset of this discipline rather than being co-extensive with it and the classification of hiding techniques are shown in the Figure 1.

The study of this subject in the scientist literature may be traced to Simmons, who in 1983 formulated it as the Prisoners' Problem" [3]. In this scenario, Alice and Bob are in jail, and wish to hatch an escape plan; all their communications pass through the warden, Willie; and if Willie detects any encrypted messages, he will frustrate their plan by throwing them into

solitary confinement. So they must find some way of hiding their ciphertext in an innocuous looking covertext.

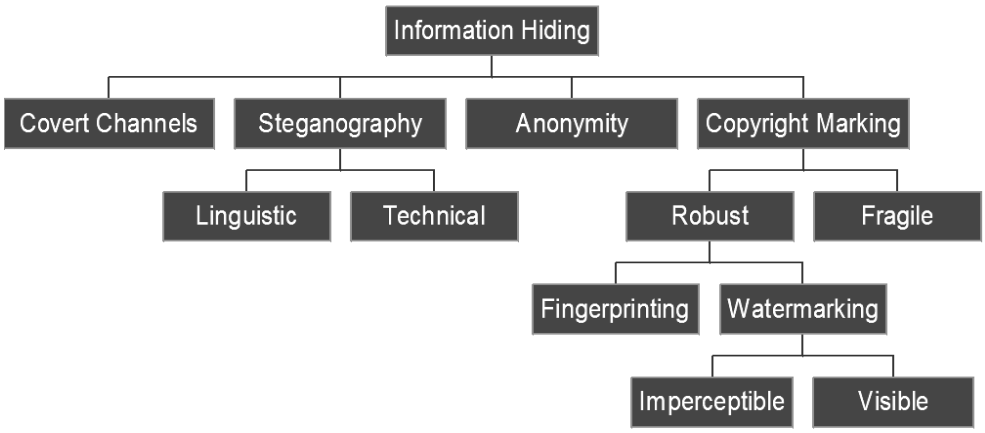


Figure 1 : classification of hiding techniques

As in the related field of cryptography, we assume that the mechanism in use is known to the warden, and so the security must depend solely on a secret key that Alice and Bob have somehow managed to share. Here why steganography works:

- **Human Visual System (HVS) - characteristics include:**
  - » Insensitivity to gradual changes in shade
  - » Insensitive to high frequencies and blue region of visible spectrum
- **Human Audio System (HAS) - characteristics include:**
  - » Sensitive to additive random noise
  - » Inability to perceive absolute phase

But, poor human *perceptibility≠undectability*

Common communication systems have a huge number of characteristics and only a small fraction of what looks like noise can actually be replaced by the statistically very clean noise of a cryptographic ciphertext. Noise in communication systems is often created by modulation, quantization and signal crossover and is heavily influenced by these mechanisms and in addition, by all kinds of filters, echo cancellation units, data format converters, … etc. Many steganographic systems have to work in noisy environments and consequently require synchronization and forward error correction mechanisms that have to be undetectable as long as the secret key is unknown. Nevertheless, with respect to computer communications, image files are good cover to hide secret information exchanging in an insecure communications channels as described in the following section.

2. Steganography in Communication Systems

It may be useful at this point to recall the book cipher. The sender and receiver share a book and encipher a message as a series of pointers to words. So the cipher group “78216” might mean page 78, paragraph 2 and the 16th word. Book codes can be secure provided that the attacker does not know which book is in use, and care is taken not to reuse a word (or a word close enough to it) [4]. The book cipher is a kind of selection channel. Steganography is closely related to the problem of "hidden channels" in secure operating system design, a term that refers to all communication paths that cannot easily be restricted by access control mechanisms (e.g. two processes that communicate by modulating and measuring the CPU load).

Steganography is also closely related to spread spectrum radio transmission, a technique that allows receiving radio signals that are over 100 times weaker than the atmospheric background noise, as well as TEMPEST, techniques, which analyze RF transmissions of computer and communication equipment in order to get access to secret information handled by these systems [5]. Most communication channels like telephone lines and radio broadcasts transmit signals, which are always accompanied by some kind of noise.

This noise can be replaced by a secret signal that has been transformed into a form that is indistinguishable from noise without knowledge of a secret key and this way, the secret signal can be transmitted undetectable. This basic design principle of steganographic systems, i.e. replacing high entropy noise with a high entropy secret transmission, is obvious. The noise on analog systems has a large number of properties very characteristic to the channel and the equipment used in the communication system. A good steganographic system has to observe the channel, has to build a model of the type of noise which is present and has then to adapt the parameters of its own encoding algorithms so that the noise replacement fits the model parameters of the noise on the channel as well as possible.

3. Hiding Information in Digital Medium

Information can be hidden by many different ways in images. To hide information, straight message insertion may encode every bit of information in the image or selectively embed the message in "noisy" areas that draw less attention those areas where there is a great deal of natural color variation. The message may also be scattered randomly throughout the image. Redundant pattern encoding "wallpapers" the cover image with the message. The following formula provides a very generic description of the pieces of the steganographic process:

cover_medium + hidden_data + stego_key = stego_medium

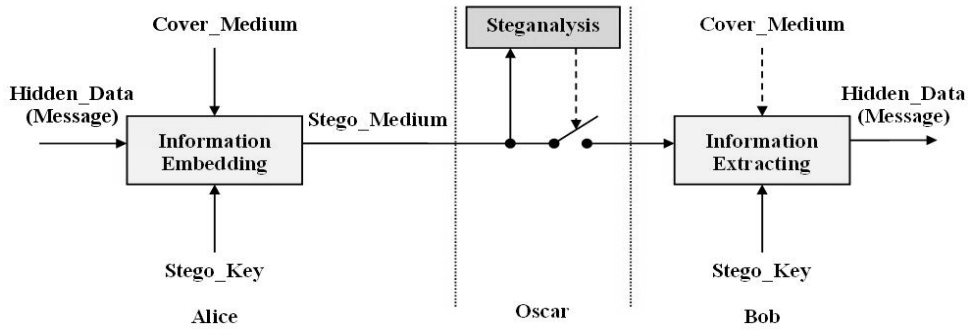


Figure 2. Dataflow for information hiding with an adversary (steganography).

Classical steganography concerns itself with ways of embedding a secret message (which might be a copyright mark, or a covert communication, or a serial number) in a cover message (such as a video lm, an audio recording, or computer code) as shown in Figure 2. The embedding is typically parameterized by a key; without knowledge of this key (or a related one) it is difficult for a third party to detect or remove the embedded material. Once the cover object has material embedded in it, it is called a stego object. Thus, for example, we might embed a mark in a covertext giving a stegotext; or embed a text in a cover image giving a stego-image; and so on.

(This terminology was agreed at the First International Workshop on Information Hiding [6]). Below is a Table 5.1, summarizing common techniques in the art of Steganography.

Table 1. Common techniques in different medium.

Media	Techniques Used
Text	Line Shift, Word Shift, or Feature Coding
Images	Least Significant Bit, Masking and Filtering, or Algorithms and Transformations
Audio	Least Significant Bit, Phase Coding, Spread Spectrum Coding or Echo Hiding
video	Least Significant Bit, Pixel Mapping, Discrete Cosine and Wavelet Transform

PowerPoint file steganography is a novel approach of hiding data which gained importance recently. In this approach, the data is stored in various animation-timing effects. This approach uses animations for applying on text, graphs, and images. Even by the addition of animation effects the content of the file is not changed. The timing effects and sound effects used for the animations are used to control the flow of data.

Military organizations also use unobtrusive communications. Their preferred mechanisms include spread spectrum and meteor scatter radio [7], which can give various combinations of resistance to detection, direction finding and jamming; they are vital for battlefield communications, where radio operators who are located are at risk of being attacked. On the Internet, anonymous remailers can be used to hide the origin of an email message, and analogous services are being developed for other protocols such as ftp and http.

4. Image Files

To a computer, an image is an array of numbers that represent light intensities at various points (pixels). These pixels make up the image's raster data. The size of an image file, then, is directly related to the number of pixels and the granularity of the color definition [8]. A typical 640 x 480 pixels image using a palette of 256 colors (8 bit per pixel), as shown in Figure 3, would require a file about 300 Kbytes in size. Such an image could contain about 300 Kbits of data. Usually, digital images are typically stored in either 24-bit or 8-bit files.

A 24-bit image provides the most space for hiding information; however, it can be quite large (with the exception of JPEG images). All color variations for the pixels are derived from three primary colors: Red, Green, and Blue (RGB). Each primary color is represented by 1 byte; 24-bit images use 3 bytes per pixel to represent a color value. These 3 bytes can be represented as hexadecimal, decimal, and/or binary values.

Figure 3. The Renoir Cover Image File as an example.

The Hyper Text Markup Language (HTML) format for indicating colors in a Web page often uses a 24-bit format employing six hexadecimal digits, each pair representing the amount of Red, Blue, And Green, respectively. The color Orange, for example, would be displayed with Red set to 100% (decimal 255, hex FF), Green set to 50% (decimal 127, hex 7F), and no Blue (0), so we would use "#FF7F00" in the HTML code. In many Web pages, the background color is represented by a six-digit hexadecimal number - actually three pairs representing Red, Green, and Blue. A white background would have the value FFFFFF: 100 percent Red (FF), 100 percent Green (FF), and 100 percent Blue (FF). Its decimal value is 255, 255, 255, and its binary value is 11111111, 11111111, 11111111, which are the three bytes making up white.

This definition of a white background is analogous to the color definition of a single pixel in an image. Pixel representation contributes to file size. For example, suppose having a 24-bit image 1024 pixels wide by 768 pixels high - a common resolution for high-resolution graphics. Such an image has more than 2 million pixels, each having such a definition, which would produce a file exceeding 2 Mbytes. Because such 24-bit images are still relatively uncommon on the Internet, their size would attract attention during transmission [9]. Due to the potential large size of such files, compression algorithms are used to reduce the image to a suitable size for sending across the Internet. File compression would thus be beneficial, if not necessary, to transmit such a file.

5. Image Compression

Two kinds of compression are used lossless and lossy. Both methods save storage space but have different results, interfering with the hidden information, when the information is uncompressed .

5.1 Lossless Compression lets us reconstruct the original message exactly; therefore, it is preferred when the original information must remain intact (as with steganographic images). Lossless compression is typical of images saved as GIF (Graphic Interchange Format) and 8-bit BMP (a Microsoft Windows and OS/2 bitmap file).

5.2 Lossy Compression, on the other hand, saves space but may not maintain the original image's integrity. This method typifies images saved as JPEG (Joint Photographic Experts Group) and yields very good compression. Due to the lossy compression algorithm, the JPEG formats provide close approximations to high-quality digital photographs but not an exact duplicate. Hence the term "lossy" compression.

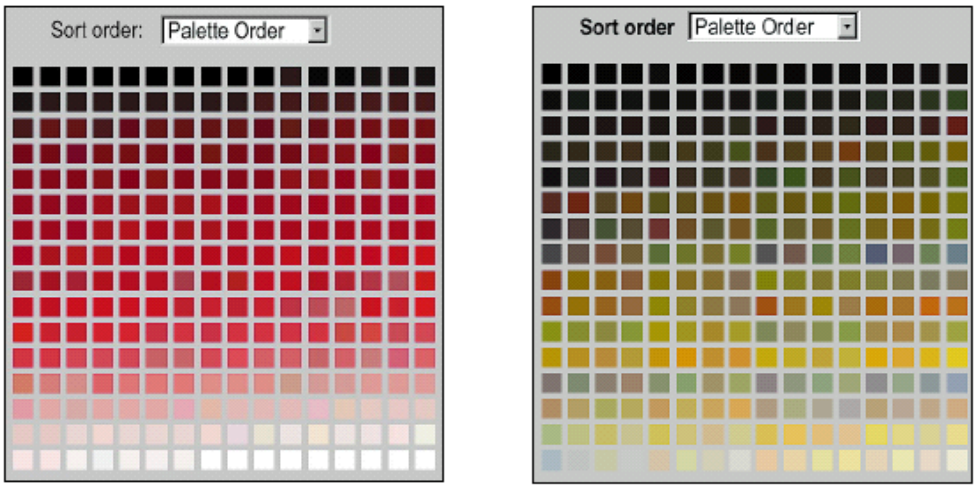
6. Embedding data in images

Information can be hidden many different ways in images. To hide information, straight message insertion may encode every bit of information in the image or selectively embed the message in "noisy" areas that draw less attention -those areas where there is a great deal of natural color variation. The message may also be scattered randomly throughout the image. Redundant pattern encoding "wallpapers" the cover image with the message. Embedding data, which is to be hidden, into an image requires two files. The first is the innocent-looking image that will hold the hidden information, called the cover image. The second file is the message - the information to be hidden. A message may be plain text, ciphertext, other images, or anything that can be embedded in a bit stream.

When combined, the cover image and the embedded message make a stego-image. A stego-key (a type of password) may also be used to hide, then later decode, the message. Most Steganography techniques neither supports nor recommends using JPEG images, but recommends instead the use of lossless 24-bit images such as BMP. The next-best alternative to 24-bit images is 256-color or gray-scale images. The most common of these found on the Internet are GIF files.

In 8-bit color images such as GIF files, each pixel is represented as a single byte, and each pixel merely points to a color index table (a palette) with 256 possible colors. The pixel's value, then, is between 0 and 255. The software simply paints the indicated color on the screen at the selected pixel position [10]. Figure 4 (a), a red palette, illustrates slight changes in color variations: visually differentiating between many of these colors is difficult.

Figure 4 (b) shows slight color changes as well as those that seem extreme, but this detail is lost in black and white copy.



(a) A 256-color red palette (b) A 256- color Renoir palette

Figure 4. Representative color palettes.

The Renoir palette is so named because it comes from a 256- color version, as shown in Figure 3, of Pierre-Auguste Renoir's "Le Moulin de la Galette". Many Steganography experts recommend using images featuring 256 shades of gray. Gray-scale images are preferred because the shades change very gradually from byte to byte, and the less the value changes between palette entries, the better they can hide information. Figure 5 shows a gray-scale palette of 256 shades. Some images are 4-bit, created with 16 shades of gray; obviously, these images offer many fewer variations. While gray-scale images may render the best results for Steganography, images with slight color variations are also highly effective, as Figure 3 showed. When considering an image in which to hide information, one must consider the image as well as the palette. Obviously, an image with large areas of solid colors is a poor choice, as variances created from the embedded message will be noticeable in the solid areas.

It is easy to see that Figure 3 (b), the palette for the Renoir cover image, makes a very good cover for holding data.

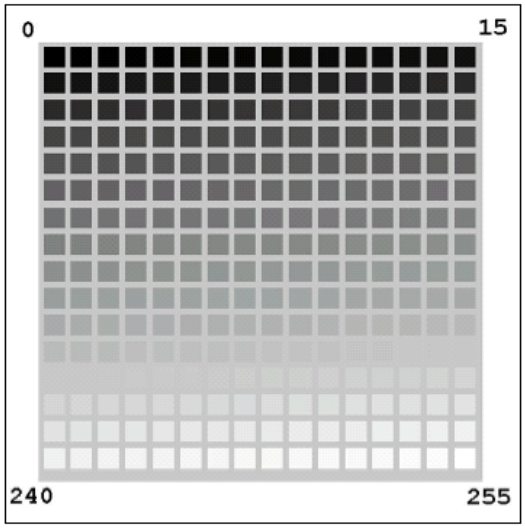


Figure 5. Representative gray-scale palette of 256 shades.

Once selecting a cover image, one must decide on a technique to hide the information you want to embed and that what will be shown in the next section.

7. Common Approaches for Hiding Information in Image

Before discussing how information is hidden in an image file, it is worth to know a number of ways to hide information in digital images include:

1. Least significant bit insertion,
2. Algorithms and transformations, and
3. Masking and filtering.

Each of these techniques can be applied, with varying degrees of success, to different image files. In the next sections, presenting these techniques in more elaboration with examples and discussing the results obtained from the experiments. First, starting with the Least Significant Bit modification [11].

7.1 Least Significant Bit Insertion (LSB)

Sometimes abbreviated as LSB, the least significant bit is the lowest bit in a series of numbers in binary, the LSB is located at the far right (also referred to as the noise), or rightmost of a string. For example in the number **01001001,** the least significant bit is the 1 at the end of the line. Least significant bit (LSB) insertion is an approach for embedding information in a cover file. Unfortunately, it is vulnerable to even a slight image manipulation. Converting an image from a format like GIF or BMP, which reconstructs the original message exactly (lossless compression) to a JPEG, which does not (lossy compression), and then back could destroy the information hidden in the LSBs. LSB manipulation is a quick and easy way to hide information.

- **24-bit images.** To hide an image in the LSBs of each byte of a 24-bit image, you can store 3 bits in each pixel. A 1024 x 768 image has the potential to hide a total of 2,359,296 bits (294,912 bytes) of information. If one compresses the message to be hidden before you embed it, you can hide a large amount of information. To the human eye, the resulting stego-image will look identical to the cover image.

In this method, we can take the binary representation of the hidden_data and overwrite the LSB of each byte within the cover_image. If using 24-bit color, the amount of change will be minimal and unapparent to the human eye. As an example, suppose that we have three adjacent pixels (nine bytes) with the following RGB encoding (the highlighted bits are the least significant bit in each byte):

10010101 00001101 11001001
10010110 00001111 11001010
10011111 00010000 11001011

Now suppose we want to "hide" the following 9 bits binary value of data (the hidden data is usually compressed prior to being hidden): 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we would get the following result (where bits in **bold** and underlined have been changed):

10010101 00001100 11001001
10010111 00001110 11001011
10011111 00010000 11001011

Note that we have successfully hidden 9 bits but at a cost of only changing 4 bits, or on average, LSB requires that only half the bits in an image be changed.(50%, of the LSBs). Also, hiding data in the least and second least significant bits is promising and still the human eye would not be able to discern it.

- **8-bit images.** 8-bit images (256 gray color) are not as tolerant to LSB manipulation because of color limitations. Steganography software authors have devised several approaches -some more successful than others- to hide information in 8-bit images. First, the cover image must be more carefully selected so that the stego-image will not broadcast the existence of an embedded message.

When information is inserted into the LSBs of the raster data, the pointers to the color entries in the palette are changed. In an abbreviated example, a simple four-color palette of white, red, blue, and green has corresponding palette position entries of 0 (00), 1 (01), 2 (10), and 3 (11), respectively. The raster values of four adjacent pixels of white, white, blue, and blue are 00 00 10 10. Hiding the binary value 1010 for the number 10 changes the raster data to 01 00 11 10, which is Red, white, Green, Blue. These gross changes in the image are visible and clearly highlight the

weakness of using 8-bit images. On the other hand, there is little visible difference noticed between adjacent gray values, as Figure 6 shows.

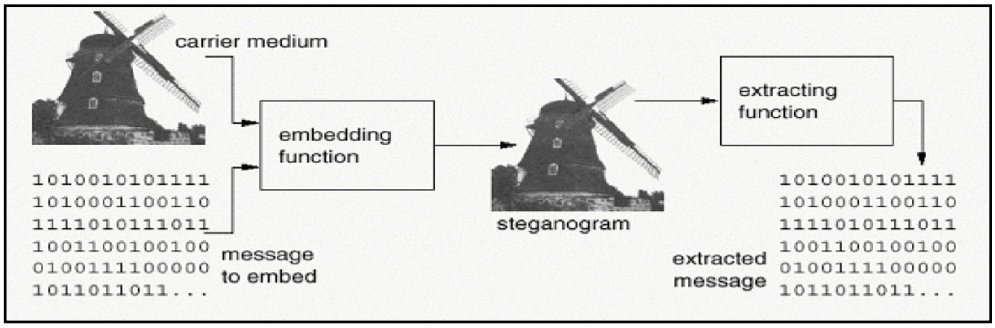


Figure 6. LSB for Image-Steganography

Steganography software processes LSB injection [12] to make the hidden information less detectable. For example, the EzStego tool arranges the palette to reduce the occurrence of adjacent index colors that contrast too much - before it inserts the message. This approach works quite well in grayscale images and may work well in images with related colors. S-Tools, another steganography tool, takes a different approach by closely approximating the cover image, which may mean radical palette changes. As with 24- bit images, changing the pixels' LSB may create new colors. (New colors may not be added to an 8-bit image due to the palette limit.) Instead, S-Tools reduces the number of colors while maintaining the image quality, so that the LSB changes do not severely change color values.

For example, eight color values are required for each color if values 000 through 111 are to be stored. Reducing the number of unique colors to 32 ensures that these values can be used and that the number of colors will not exceed 256 (256/8 = 32). Each of the 32 unique colors in the palette may be expanded to eight colors having LSB values of the red, green, blue (RGB) triples ranging from 000 to 111. This results in multiple colors in the palette that look the same visually but that may vary by one bit. These tools take a similar approach with gray-scale images. However, the resulting stego-images as applied with S-Tools are no longer gray-scale. Instead of simply going with

adjacent colors as EzStego does, S-Tools manipulates the palette to produce colors that have a difference of one bit.

For example, in a normal gray-scale image, white will move to black with the following RGB triples (255 255 255), (254 254 254), ...,

(1 1 1), (0 0 0)

After processing with S-Tools, the value for white will be spread over a range of up to eight colors such as (255 255 255), (255 255 254), and (255 254 255) .

Visually, the stego-image may look the same as the gray-scale cover image, but it has actually become an 8-bit color image.

7.2 Algorithms and Transformations

Lossy compression is a key advantage that JPEG images have over other formats. High color quality images can be stored in relatively small files using JPEG compression methods; thus, JPEG images are becoming richer on the Internet. JPEG images use the DCT "discrete cosine transform" to achieve compression. DCT is a lossy compression transform because the cosine values cannot be calculated exactly, and repeated calculations using limited precision numbers introduce rounding errors into the final result. Variances between original data values and restored data values depend on the method used to calculate DCT.

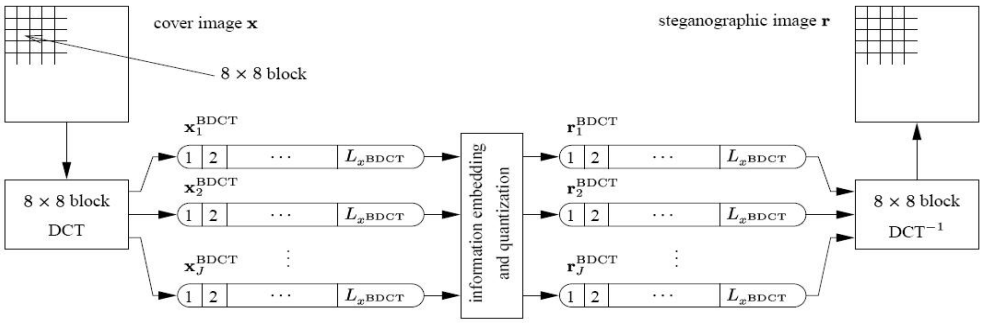


Figure 7. Image Steganography using the 8 x 8 Block DCT (BDCT).

A stochastic model of the cover data based on a two-dimensional Discrete Cosine Transform (DCT) of non-overlapping 8 x 8 blocks of the image pixels is presented. Figure 7 illustrates the 8 x 8 block DCT, which is denoted as BDCT subsequently. The *i*th 8 x 8 blocks in row-scan is transformed into 64 DCT coefficients $\{x_{i,1}^{BDCT}, x_{i,2}^{BDCT}, ..., x_{i,j}^{BDCT}, ..., x_{i,64}^{BDCT}\}$. Next, the coefficients with identical frequency index *j* from all 8 x 8 blocks compose the signal x_j^{BDCT}, which can be considered as a sub channel. Thus, there are 64 sub channels, all having the same length $L_{x^{BDCT}}$ which is identical to the number of 8 x 8 blocks in the given image x.

The common zig-zag scan is used for labeling the 64 signals x_j^{BDCT}. The JSteg algorithm. As it runs, the algorithm sequentially replaces the least-significant bit of discrete cosine transform (DCT) coefficients with message data. It does not require a shared secret.

Input: message, cover image

Output: stego image

while data left to embed **do**

get next DCT coefficient from cover image

if DCT ≠ , 0 and DCT ≠ , 1 **then**

get next LSB from message

replace DCT LSB with message LSB

end if

insert DCT into stego image

end while

In addition to DCT, images can be processed with FFT "Fast Fourier Transformation" and Wavelet Transformation or use redundant pattern encoding or spread spectrum methods to scatter hidden information throughout the cover images.

7.3 Masking and Filtering

Masking and filtering techniques, usually restricted to 24-bit and gray-scale images, hide information by marking an image, in a manner similar to paper watermarks. Watermarking techniques may be applied without fear of image destruction due to lossy compression because they are more integrated into the image. Visible watermarks are not Steganography by definition. The difference is primarily one of intent [13].

Traditional Steganography conceals information; watermarks extend information and become an attribute of the cover image. Digital watermarks may include such information as copyright, ownership, or license. In Steganography, the object of communication is the hidden message. In digital watermarks, the object of communication is the cover. One can use the watermarked image to hide plaintext or ciphertext information. Masking is robust with respect to compression, cropping, and some image processing. Masking techniques embed information in significant areas so that the hidden message is more integral to the cover image than just hiding it in the "noise" level.

8. Entropy

Entropy arguments are used in conventional information theory; how far will they get us in steganography? Assuming that the material to be embedded is indistinguishable from random data (as would be the case were it competently encrypted), then entropy will be strictly additive: the entropy of the stegotext S will equal the entropy of the covertext C plus the entropy of the embedded material E:

$$H(S) = H(C) + H(E)$$

Thus in order to make our embedding process secure against an opponent who merely has to detect the presence or absence of embedded material; it appears that we have two alternatives:

1. Keep H(E) much less than the uncertainty in the opponent's measurement of H(C)

2. Find some way of processing C to reduce its entropy by an amount that can then be made up by adding E. For example, one might use a

noise reduction or lossy compression algorithm to remove some unnecessary information from C before embedding E.

The problem is that we do not know how competent our opponent is at measuring the entropy of the covertext we are using, or, equivalently, at discriminating signal from noise. We will often be up against an opponent of unpredictable power (a pirate attacking our system a generation from now); and these are precisely the circumstances where we may want a security proof.

But the more stegotext we give the warden, the better he may be able to estimate the statistics of the underlying covertext, and so the smaller the rate at which Alice will be able to tweak bits safely. The rate might even tend to zero, as was noted in the context of covert channels in operating systems [14]. However, as a matter of empirical fact, there do exist channels in which ciphertext can be inserted at a positive rate, and people have investigated correlations in various types of content such as digital video. So measuring entropy may be useful in a number of applications. But is there any prospect of developing steganographic techniques which we can prove will resist an opponent of arbitrary ability?

9. The Dead Drop

The ‘dead drop’ is a term used by security agencies around the globe to describe a process of anonymous communication between two parties. The integral advantage of the dead drop is the anonymity of all individuals involved in a message passing. This is why Steganography has possibly become a tool for the dead drop over the Internet. Information can be communicated over the Internet reasonably anonymously; the using of pseudonym emails and usernames aids this. Encryption can also hinder any attempts to intercept. Eventually though even if the content of the actual transmissions are never discovered their sources and perhaps even individuals can be traced from a point of origin. Remember a sender needs to know who to send to. Steganography can provide the extra piece of secrecy that these transactions require, complete anonymity between sender and receiver.

The Internet would be ideal for a dead drop. Images could be posted on bulletin boards that appear perfectly normal to everyone else but to the right person the image can be downloaded and hidden information extracted. Secret correspondence could also be requested. An individual could post a request on a board asking for a specific type of image/audio file. The other person in the chain, which would be looking for posts like this, would then see their contact would like to communicate. They would then post the required image (hidden information inside).

10. Characterizing Data Hiding Techniques

Steganographic techniques embed a message inside a cover [15]; various features characterize the strengths and weaknesses of the methods. The relative importance of each feature depends on the application as shown in the "Magic" Triangle Figure 8.

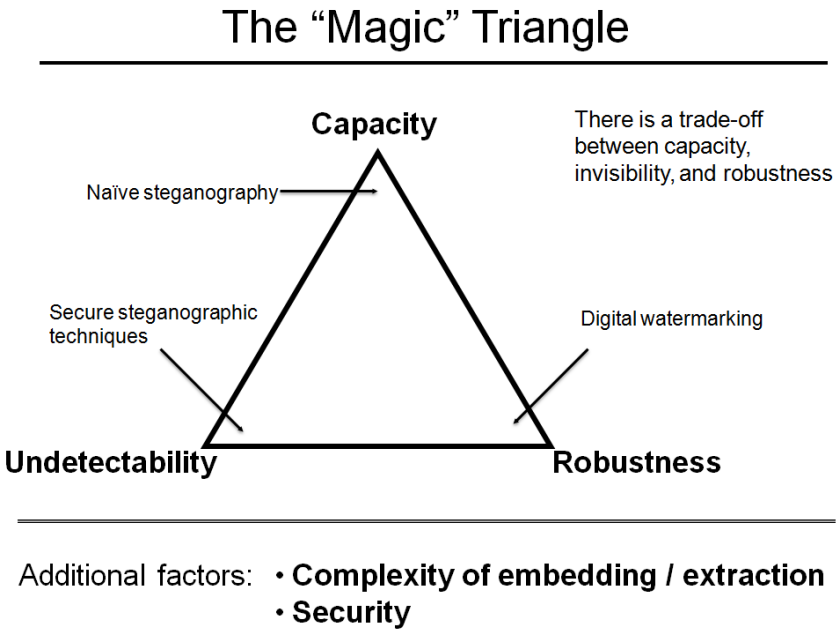


Figure 8. The Magic Triangle

(a) Hiding Capacity: Hiding capacity is the size of information that can be hidden relative to the size of the cover. A larger hiding capacity allows the use of a smaller cover for a message of fixed size, and thus decreases the bandwidth required to transmit the stego-image.

(b) Perceptual Transparency: The act of hiding the message in the cover necessitates some noise modulation or distortion of the cover image. It is important that the embedding occur without significant degradation of perceptual quality of the cover.

(c) Robustness: Robustness refers to the ability of embedded data to remain intact if the stego-image undergoes transformations, such as filtering, scaling, rotations and cropping or lossy compression.

(d) Tamper Resistance: Beyond robustness to destruction, tamper-resistance refers to the difficulty for an attacker to alter or forge a message once it has been embedded in a stego-image, such as a pirate replacing a copyright mark with one claiming legal ownership. Applications that demand high robustness usually also demand a strong degree of tamper resistance.

(e) Computational Complexity: Computational complexity of encoding and decoding is another consideration and individual applications may have additional requirements. For example, for a copyright protection application, a watermark should be resistant to collusion attacks where many pirates work together to identify and destroy the mark.

Table 2. Image Steganography Algorithm Measures

Measures	Advantage	Disadvantage
High Capacity	High	Low
Perceptual Transparency	High	Low
Robustness	High	Low
Temper Resistance	High	Low
Computation Complexity	Low	High

In the next section, a number of experimental results and analyze the undetectability are presented in the two cases of images (8 bit and 24 bit).

11. Experimental Results and Performance Analysis

Firstly, image steganography terminologies are as follows:-

- **Cover-Image:** Original image which is used as a carrier for hidden information.
- **Message:** Actual information which is used to hide into images. Message could be a plain text or some other image.
- **Stego-Image:** After embedding message into cover image is known as stego-image.
- **Stego-Key:** A key is used for embedding or extracting the messages from cover-images and stego-images.

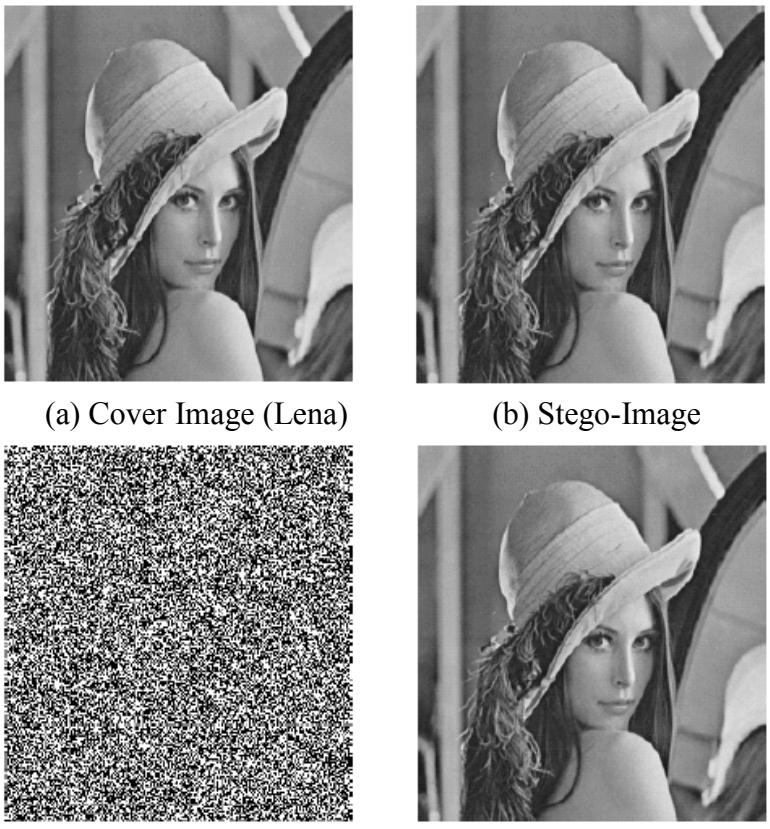
A great deal of 8 bit (gray-scale) and 24 bit images have been investigated with the some of the embedding scheme. All cover images are exported from USC-SIPI Image Database. First, hiding text message into an 8 bit cover-image experiment then hiding an image into four 8 bit cover-images. Finally, hiding an encrypted-image into a 24 bit cover-image. In addition to presenting, their corresponding created stego-images and their histogram.

11.1 Experiment 1: Hiding message into a 8 bit image

This steganographic scheme embeds the bits of the message directly into the least-significant bit of the cover image in a deterministic sequence. Modulating the least-significant bit does not result in a human-perceptible difference because the amplitude of the change is small. Also, anther technique “processes” the message with a pseudorandom noise sequence (e.g., BBS) before or during insertion into the cover image. LSB embedding also allows high perceptual transparency. Below is an illustration of the LSB embedding. The message that we want to hide is a text file containing a single paragraph as shown in Figure 9 while, the cover Lena image is shown on Figure 10 (a), which is Lena 200 x 200 pixel (8-bit) grayscale image.

```
Steganography serves to hide secret messages in
other messages, such that the secret is very
existence is concealed. Generally, the sender
writes an innocuous message and then conceals a
secret message on the same piece of paper.
Historical tricks include invisible inks, tiny
pin punctures on selected characters, minute
differences between handwritten characters,
```

Figure 9. The plaintext, which want to be hidden.



(a) Cover Image (Lena) (b) Stego-Image

(c) Difference Image (d) Modified Stego-Image

Figure 10. Hiding Plaintext Message into Lena Image (Steganography).

Using LSB, the stego-image shown on Figure 10 (b) is produced. (The encryption phase was ignored in this example). The difference image is shown on Figure 10 (c), where "white" pixels indicate the spatial locations where the images differ.

The cover image can hold up to 5000 bytes of information but the plaintext message is 484 bytes only which can be embedded straightforwardly. LSB is able to recover the message when the stego-image is available for decoding. However to evaluate the fragile nature of the embedding, Gaussian additive noise (with zero-mean and unit variance) was added to each pixel intensity value in the stego-image to produce the altered stego-image shown on Figure 10 (d).

Unfortunately, LSB was not able to extract the message, which mistakenly believed that the modified stego-image contained some encrypted data and asked for a decryption key.

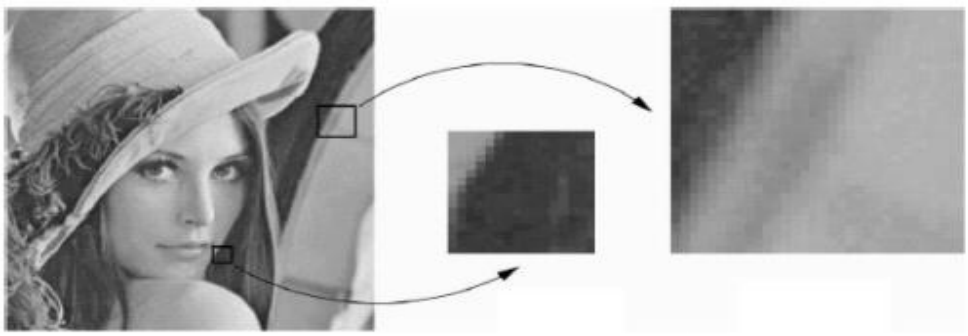


Figure 11. Lena stego-image analysis.

By the human eyes, both images looks same but as an expert analysis and using the matlab utilities to detect the modified pixels as shown in Figure 11. The advantage of LSB embedding is its simplicity and speed. LSB encoding is extremely sensitive to any kind of filtering or manipulation of the stego-image.

11.2 Experiment 2: Hiding image into four 8 bit images

This experiment will hide an original image (tank) as shown in Figure 12 into different four gray-scale cover-images, entitled Chemical plant, F16, Fishing boat, Peppers, that are in Figure 13(a), (b), (c) and (d) respectively.

The size of the original tank image is 100 x 25 pixels (2.5 Kbytes) while the size of these cover-images is 512 x 512 (256 Kbytes), thus there are 20,000 message-bits can be embedded in each image easily.

In case of embedding the original image into the cover images Figures 13(a) and 13(b) , embedding in the least significant bit only are done, while embedding in Figures 13(c) and 13(d) , embedding in the least and second least significant bits are achieved. The resulted stego-images are shown in Figure 14.

Figure 12. Tank (original-image).

From Figures 13 and 14, it is seen that cover-images and stego-images are nearly indistinguishable by the naked eyes. More recently, people are hiding secret messages in graphic images. Replace the least significant bit of each byte of the image with the bits of the message. The graphical image will not change appreciably - most graphics standards specify more gradations of color than the human eye can notice - and the message can be stripped out at the receiving end. The success of Steganography is dependent upon selecting the proper cover image.

(a) Chemical plant

(b) F16 aircraft

(c) Fishing boat

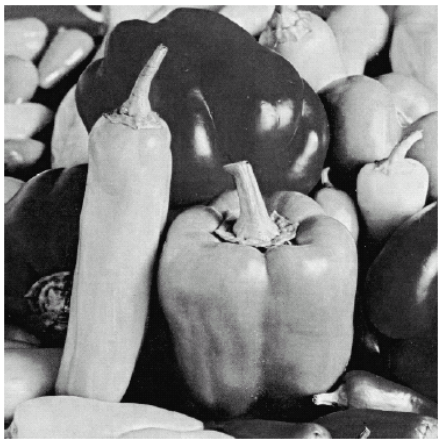
(d) Peppers

Figure 13. Four original cover-images. (a) Chemical plant. (b) F16. (c) Fishing boat. (d) Peppers.

(a) Stego-Chemical plant

(b) Stego-F16 aircraft

(c) Stego- Fishing boat



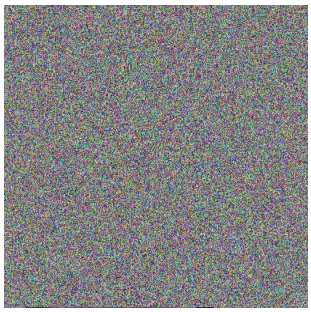
(d) Stego-Peppers

Figure 14. Four stego-images resulted from applying the proposed Hybrid method on Figure 13. (a) Chemical plant stego-image. (b) F16 stego-image. (c) Fishing boat stego-image. (d) Peppers stego-image.

11.3 Experiment 3: Hiding encrypted image into a 24 bit image

Steganography goes well beyond simply embedding text in an image. It also pertains to other media, including voice, image, binary files, and communication channels. In this experiment, hiding an encrypted image into a 24 bits image is illustrated. Here, the original image is F16 with 24 bit with dimension of 50 x 50 pixels as shown in Figure 15 (a) while after applying a Cryptography method, the corresponding encrypted image is shown in Figure 15 (b). The size of the original F16 image is 50 x 50 (2.5 Kbytes) and consequently the same size for the encrypted image.

(a) Original image “F16”



(b) Encrypted “F16”

Figure 15. Original F16 and its corresponding encrypted one.

(a) Cover baboon Image

(b) Stego- baboon Image

Figure 16. Baboon cover image and its corresponding stego image.

The cover image as shown in Figure 16 is a 24 bit baboon with size 1024 x 768 pixels (2.25 Gbytes). A 1024 x 768 image has the potential to hide a total of 2,359,296 bits (294,912 bytes = 288 Kbytes) of information while the encrypted image that will be hiding is only 2.5Kbytes therefore, this operation will be done with no trouble.

As presented, LSB Embedding has the advantage that it is simple to implement. This is especially true in the 24-bit bitmap case. It also allows for a relatively high payload, carrying one bit of the secret message per byte of pixel data. In addition, it is also seemingly undetectable by the average human if done right. However, the assumption has been that the stego-image is indistinguishable from the original cover image by the human eye.

11.4 Undetectability

The larger the cover message is (in data content terms - number of bits) relative to the hidden message, the easier it is to hide the latter. For this reason, digital pictures (which contain large amounts of data) are used to hide messages on the Internet and on other communication media. For example: a 24 bit bitmap will have 8 bits representing each of the three color values (red, green, and blue) at each pixel. If we consider just the blue there will be 2^8 different values of blue. The difference between say 11111111 and 11111110 in the value for blue intensity is likely to be undetectable by the human eye. There are some guidelines and restrictions when embedding data, that what will be discussed next.

11.5 Guidelines and Restrictions when Embedding Data

As mentioned before, the goal of Steganography is to conceal data. There are a few features and restrictions to successfully hide data. “The goal is for the data to remain “hidden.” The word “hidden” has two meanings here, (1) the data can be “hidden” and not visible to the human eye (2) the data can be visible and still not visible to the human eye . If the focus is deterred from the data, the data will not be seen, which means that it is “hidden”. The following guidelines represent a few features and restrictions when embedding data[16].

- The cover image should not be degraded by the embedded data. It should appear that the cover image does not look distorted. It should not have a noticeable change in color composition, or that of the luminance. Frequently a cover image's size becomes enlarged; this can look very suspicious.
- Embedded data should be directly embedded into the cover image not in the header or wrapper. The embedded data needs to remain intact across different file formats.
- The embedded data should be immune to any manipulation. This ranges from any intentional modification or any modification through transmission.
- Error correcting codes should be inserted in the cover image to ensure integrity of data when or if the cover image is modified or tampered with.
- The embedded data should be recoverable and intact if only fragments of the cover image remain.

12. Commercial Steganography Tools Examples:

EzStego	online.securityfocus.com/tools/586/scoreit/
F5	wwwrn.inf.tu-dresden.de/~westfeld/f5.html
Hide and Seek v4.1	ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/
Hide and Seek for Win95	ftp.hacktic.nl/pub/crypto/incoming/
Hide4PGP	www.heinz-repp.onlinehome.de/Hide4PGP.htm
Jpeg-Jsteg	ftp://ftp.funet.fi/pub/crypt/steganography/
Mandelsteg	ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/
MP3Stego	www.cl.cam.ac.uk/~fapp2/steganography/mp3stego/
OutGuess	www.outguess.org/download.php
Steganos	www.steganos.com/en/
S-Tools v4	members.tripod.com/steganography/stego/s-tools4.html
White Noice Storm	ftp://ftp.esua.berkeley.edu/pub/cypherpunks/steganography/

For a steganography tool table, see
www.jjtc.com/Steganography/toolmatrix.htm

13. Advantages and Disadvantages of Steganography

While Steganography tools would seem more useful for cheating purposes than for good (as in military applications), that does not always have to be the case. One of the main areas of concern in the security world today is preventing other possible criminals' activities. It seems that criminals and other unlawful individuals and organizations are using this form of information hiding (Steganography) as means to transmit data or communicate with each other in a covert manner. They can post instructions for their group on sports chat rooms, pornographic bulletin boards (it is such a popular medium of carrying hidden messages) and other Web sites.

Steganography allows these types of people to place their hidden communications in a image or message then place that message in a central location (drop point), thereby allowing others to access the information without ever having to make face-to-face or verbal contact with the one who posted the message. What better means of communication could a criminal use in such a stealthy manner? It is still hard to tell if an image may be contain hidden secrets and even harder to track where and who put them there. There are millions of people surfing the Internet daily, uploading and downloading information and images, and posting messages; so think about this, that next graphic you download could contain the next criminal plot, but would you know it?

Since ancient times, man has found a desire in the ability to communicate covertly. In this chapter, a full discussion about Steganography and its place in security is presented. An overview of Steganography was presented along with real application examples that can benefit from this technology [17,18]. Steganography is not intended to replace Cryptography but supplement it.

14. Evaluation of Image Quality

For comparing stego image with cover results requires a measure of image quality, commonly used measures are Mean-Squared Error, Peak Signal-to-Noise Ratio and capacity.

14.1 Mean-Squared Error: The mean-squared error (MSE) between two images I1(m,n) and I2(m,n) is: M and N are the number of rows and columns in the input images, respectively.

$$MSE = \frac{\sum_{M,N}[I1(m,n) - I2(M,N)]^2}{M * N}$$

14.2 Peak Signal-to-Noise Ratio: Peak Signal-to-Noise Ratio (PSNR) avoids this problem by scaling the MSE according to the image range: PSNR is measured in decibels (dB). PSNR is a good measure for comparing restoration results for the same image.

$$PSNR = 10 \log_{10} \frac{256^2}{MSE}$$

14.3 Capacity: It is the size of the data in a cover image that can be modified without deteriorating **the** integrity of the cover image. The steganography embedding operation needs to preserve the statistical properties of the cover image in addition to its perceptual quality. Therefore capacity depends on total number of bits per pixel & number of bits embedded in each pixel. Capacity is represented by bits per pixel (bpp) and the Maximum Hiding Capacity (MHC) in terms of percentage.

14.4 Domain Type (DOM): DOM is either Spatial(S) or Transform (T). The techniques that use transform domain hide information in significant areas of the cover images and may be more complex for attackers.

14.5 Normalized Coefficient (NC): Correlation is one of the best methods to evaluate the degree of closeness between the two functions. This measure can be used to determine the extent to which the original image and stego image remain close to each other, even after embedding the data.

15. Conclusion and Remarks

Steganography has its place in security. It is not intended to replace cryptography but supplement it. Hiding a message with steganography methods reduces the chance of a message being detected. From the experimental results, we conclude that the Steganography serves to hide secret messages in other media, such that the secret is very existence is concealed. After embedding the message into cover-image, the output image still undetectable by the human eyes. Steganographic techniques are best suited as good carrier techniques for confidential data transfers.

Using the digital images as an envelope to embed secret data imperceptible to any human eyes without changing the visual quality of this dummy image. It does not change even the file size. If one wants to see the secret data, one can easily restore it from the stego-image. So, one can store/send it very safely through insecure communication channels. Steganography helps protect against hidden message extraction but not against message destruction through image processing. Steganography, like Cryptography, will play an increasing role in the future of secure communication in the "digital world".

Acknowledgements: Some of the presented survey here were claried by discussion with Prof. Dr. Shilong Ma in Beijing University of Aeronautics and Astronautics (BUAA), China.

References

1. Vijay Kumar Sharma, Vishalshrivastava, "A Steganography Algorithm for Hiding Images by improved LSB substitution by minize detection."Journal of Theoretical and Applied Information Technology, Vol. 36 No.1, ISSN: 1992-8645, 15th February 2012.
2. Stuti Goel, Arun Rana, Manpreet Kaur, Comparison of Image Steganography Techniques, ISSN: 2278-5183, International Journal of Computers and Distributed Systems, www.ijcdsonline.com, Vol. No.3, Issue I, April-May 2013

3. Bret Dunbar. “Steganographic Techniques and their usage in an Open-System Environment,”ACMSANS Institute Info.Secur., 2002.
4. J.RKrenn.“Steganography and Steganalysis,” IEEE Info.Assu.Secu., Jan 2004.
5. Mathkour Hassan, Sadoon A1 Batool and Touir Ameur, “A New Image Steganography Technique” IEEE Wireless Com.,Ntwrk. And Mob. Cmptn.,Oct 2008.
6. Pooyan M and Delforouzi A. “LSB-based Audio Steganography Method Based on Lifting Wavelet Transform,” IEEE International Symposium Sgnl. Proc and Info.Tech.., Dec 2007.
7. Neeta D, Snehal K and Jacobs D. “Implementation of LSB Steganography and its Evaluation for Various Bits,” IEEE Digi. Info. Mang. pp 173-178, Dec 2006.
8. Shirali-Shahreza M and Shirali-Shahreza S. “Steganography in TeX Documents,” IEEE Intel’s. Sys. Knwldg. Engg. Vol 1, 2008
9. De Carvalho DF, Chies R and Freire AP. “Video Steganography for Confidential Documents: Integrity, Privacy and Version Control,” ACM 26th International,2008.
10. Wen-Chao Yang and Ling-Hwei Chen. “A Novel Steganography Method via Various Animation Effects in PowerPoint Files,” ACM Library Proceeding to Seventh International Conference on Machine Learning Cybernetics, July 2008.
11. Min-Qun Jing,Wen-Chao Yang and Ling-Hwei Chen. “A New Steganography Method via Various Animation Timing Effects in PowerPoint Files,”ACM Library Proceeding to Eighth International Conference on Machine Learning Cybernetics,July2009.
12. Andreas Westfeld and Gritta Wolf. “Steganography in a Video Conferencing System,” Institute for Theoretical Computer Science, pp 32-47,Jan 1998.
13. Mehdi Kharrazi, Husrev T, Sencar, and Nasir Memon. “Image Steganography: Concepts and Practice,” Stevens Institute of Technology pp 204-211, Feb 2004.
14. J Dong, W Wang and T Tan. “Multi-class Blind Steganalysis Based on 35 Image Run-Length Analysis,” IEEE Assu.Secu.2009

15. Lingyun Xiang, Xingming Sun, Gang Luo and Can Gan. "Research on Steganalysis for Text Steganography Based on Font Format," IEEE Info.Assu. and Sec., Third International Symposium, Aug 2007.
16. Pritish Bhautmage, Prof. Amutha Jeyakumar, Advanced Video Steganography Algorithm, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, January -February 2013, pp.1641-1644.
17. T. Narasimmalou, Allen Joseph .R, "Optimized Discrete Wavelet Transform based Steganography", IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT),2012.
18. Neda Raftari and Amir Masoud Eftekhari Moghadam, "Digital Image Steganography Based on Assignment Algorithm and Combination of DCT-IWT", Fourth International Conference on Computational Intelligence, Communication Systems and Networks, 2012.