Palette-Based Image Steganography by Mapping Pixels to Letters

Khin Kyu Kyu

khinkyu28@gmail.com
Computer University, Pathein, Myanmar

Abstract

This paper is represented for information security by using image steganography method on palette-based image. The palette image based steganography is used LSB encoding procedure. Steganograpyh is the art of hiding information in other information media when communication is take place. Many different file formats can be use, but digital images are more popular usages. We propose the encoding the message to an image and the decoding this message by mapping pixels to letters. The result image is not different as in original image. So, the intruders can't know the messages embedded in the image until they get that image.

Keywords: Steganograhy, LSB encoding, Hiding data behind image

1. Introduction

The growth of the internet usage and the overall development of digital technologies in the past years have sharply increased the availability of digital multimedia. Some of work needs to be done in order to develop security systems to protect the information contained in digital data.

Computer security has become sufficiently important that it was inevitable that governments would decide they needed to do something about it. And when governments want to know something about security, they turn to the experts the military. And they develop standards and measurement about by which security can be measured that are unbiased so as not to favor any one organization.

The security threads in different environments are very different, as the best ways to counter them. The military has traditionally focused on keeping their data secret. They are less concerned about data getting corrupted or forged. In the paper world, forgeries are so difficult and so likely to expose the spies placing them that this threat takes a back seat. In the computerized environment, modification or corruption of data is a more likely threat [2].

Information is an asset that has a value like any other asset. As an asset, information needs to be secured form the attacks. To be secured, information needs to be hidden from unauthorized access, protected form unauthorized change, and available to an unauthorized entity when it is needed.

The confidentiality of the files was achieved by restricting the access to a few authorized and trusted people in the organization. In the same way, only a few authorized people were allowed to change the contents of the files. Availability was achieved by designating at least one person who would have access to the files at all times.

With the advent of computers, information storage became electronic. Instead of being stored on physical media, it was stored in computers. The three security requirements, however, did not change. The files stored in computers require confidentiality, integrity and availability [1].

During the last two decades, computer networks created a revolution in the use of information. Information is now distributed. Authorized people can send and retrieve information from a distance using computer networks. Not only should information be confidential when it is stored in a computer; there should also be a way to maintain its confidentially when it is transmitted form one computer to another.

Steganography refers to the science of "invisible" communication [3]. Unlike cryptography, where the goal is to secure communications from an eavesdropper, steganotraphic techniques strive to hide the very presence of the message itself from an observer. The general idea of hiding some information in digital content has a wider class of applications that go beyond steganography. The techniques involved in such applications are collectively referred to as information hiding.

Compressed image formats have been popular domain of research for steganographic applications [5, 4]. However, limited success has been achieved for palette-based image formats (JPEG/PNG). EZ Stego, one of the most popular data hiding schemes for palette-based images proposed by Machado [8], is similar to the commonly used LSB method for 24 bit colour images (or 8 bit grayscale images). After the palette colours are sorted by luminance, it embeds the message into the LSB of indices pointing to the palette colours. Message recovery is simply achieved by selecting the same pixels and collecting the LSBs of all indices to the ordered palette.

2. Steganography

The word steganography comes from the Greek origins Steganos, which mean covered or secret and graphy mean writing or drawing. Therefore, steganography means, literally, covered writing. Steganography is the art and science of hiding information such that its presence cannot be detected and a communication is happening. Secret information is encoding in a manner such that they the very existence of the information is concealed. Paired with existing communication methods, steganography can be used to carry out hidden exchanges.

The main goal of steganography is to communicate securely in a completely undetectable manner and to avoid drawing suspicious to the transmission of a hidden data. It is not to keep others from knowing the hidden information, but it is to keep others from thinking that the information even exists.

The basic model of steganography consists of Carrier, Message and Password. Carrier is also known as cover-object, which the message is embedded and serves to hide the presence of message [4].

There are many types of steganography such as digital steganography, network steganography, printed steganography, and text steganography. Text steganography is considered to be the most difficult kind of steganography due to lack of redundancy in text as compared to image or audio but still has smaller memory occupation and simpler communication.

A special case of information hiding is digital watermarking. Digital watermarking is the process of embedding information into digital multimedia content such that the information can later be extracted or detected for a variety of purposes including copy prevention and control. Digital watermarking has become an active and important area of research, and development and commercialization of watermarking techniques is being deemed essential to help address some of the challenges faced by the rapid proliferation of digital content.

The key difference between information hiding and watermarking is the absence of and active adversary. In watermarking applications like copyright protection and authentication, there is an active adversary that would attempt to remove, invalidate or forge watermarks. In information hiding there is no such active adversary as there is no value associated with the act of removing the information hiding in the content.

2.1. Uses of Steganography (Audio/Image)

The simplest method of hiding information within a file is to replace all the least significant bits (LSB) within each bit plane of a file. This change can barely been seen by the naked eye even when up to 4 of the LSB's are changed in each plane. This method however is not successful in audio Steganography as changes to the LSB adds 'noise' that can be audible during quiet periods of the sound [15]. Steganalysis tools can also easily detect this method, increased success can be achieved by removing some of the randomness introduced by the bit changes, e.g. a change of every LSB by one would probably not be detected as there is no random element present. The bits would be assumed to form part of the original image.

An increasingly complex method of image Steganography is known as the patchwork algorithm [15]. This algorithm randomly selects pairs of pixels on a given image. The brighter of the two pixels is made brighter, and the darker one darker. This change is so subtle that it is undetectable to the human eye; even at high zoom levels the changes simply are not sufficient to make the image appear altered. The contrast change between these two pixels now forms part of the bit pattern for the hidden file. In order to go undetected by a filtering a limit to a few hundred changes can take place. A similar technique can be used in audio files, increasing the amplitude contrast of pairs of randomly chosen sound samples within the overall audio file. A filter is then applied to remove any high frequency noise created as a result of the increases.

2.2. EzStego

EzStego sorts the color palette so that the colors flow smoothly into each other. The hope is that changing the least significant bit doesn't drastically change the color. Ordering colors in one dimension is easy. Ordering a three dimensional color space is not. EzStego treats the colors as cities in RGB space and and tries to find the shortest path through all of the stops.

The following are the process of EzStego (Figure.1);

1. Sort the palette so closest colors fall next to each other.
2. Encode the encrypted message by twiddling the least significant bit.
3. Unsort the palette by renumbering all of the colors with their original values.
4. Ship the image.
5. Receiver resorts palette using same algorithm and extracts bits by using the sorted palette.

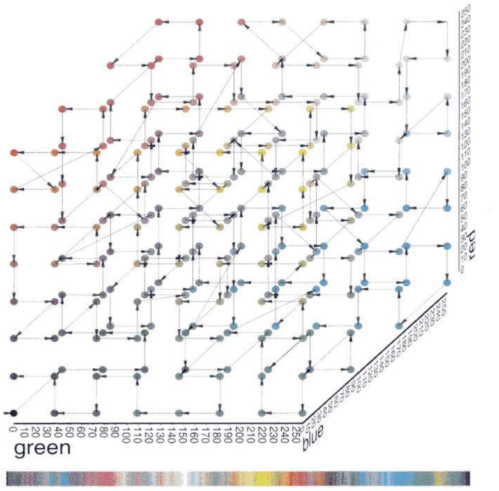


Figure. 1. EzStego Encoding

2.3. Hiding in palette-based image

The palette-based image encoding maintains a table of up to 256 colors that best represent the image. The color of each pixel is described by indexing into this color table. Changing the least significant bit may significantly change the image.

2.4. LSB endcoding

LSB (Least Significant Bit) technique replaces the last bit of any one of the color channel (R, G or B) with a bit of secret data. For all pixels the color channel is going to be same. This is a very simple technique but the probability of finding data is very high[7].

3. Hiding Schemes

First techniques included invisible ink, secret writing using chemicals, templates laid over text messages, microdots, changing letter /word /line/paragraph spacing, changing fonts.

The need of the data hiding is:

- Covert communication using images (secret message is hidden in a carrier image)
- Ownership of digital images, authentication, copyright
- Data integrity, fraud detection, self-correcting images
- Traitor-tracing (fingerprinting video-tapes)
- Adding captions to images, additional information, such as subtitles, to video, embedding subtitles or audio tracks to video (video-in-video)

3.1. Properties of Hiding Schemes

There are four properties of hiding schemes:

3.1.1. Robustness

The ability to extract hidden information after common image processing operations: linear and nonlinear filters, lossy compression, contrast adjustment, recoloring, resampling, scaling, rotation, noise adding, cropping, printing / copying / scanning, D/A and A/D conversion, pixel permutation in small neighborhood, color quantization (as in palette images), skipping rows / columns, adding rows / columns, frame swapping, frame averaging (temporal averaging), etc.

3.1.2. Undetectability

It is impossibility to prove the presence of a hidden message. This concept is inherently tied to the statistical model of the carrier image. The ability to detect the presence does not automatically imply the ability to read the hidden message. Undetectability should not be mistaken for invisibility – a concept related to human perception.

3.1.3. Invisibility

Perceptual transparency is based on the properties of the human visual system or the human audio system.

3.1.4. Security

The embedded information cannot be removed beyond reliable detection by targeted attacks based on a full knowledge of the embedding algorithm and the detector (except a secret key), and the knowledge of at least one carrier with hidden message.

4. Palette-Based Image

The best example of the palette based image is the JPEG standard format. JPEG is named after its origin, the Joint Photographers Experts Group.

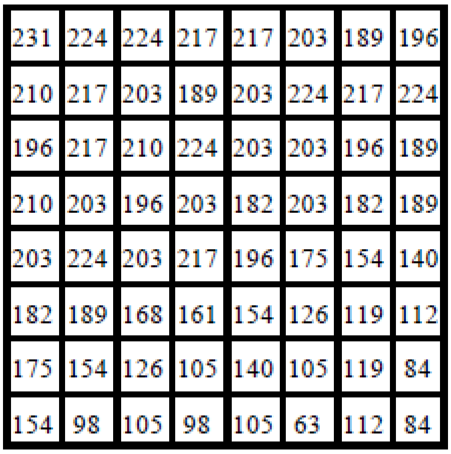
Transform encoding is based on a simple premise: when the signal is passed through the Fourier transform, the resulting data values will no longer be equal in their information carrying roles. In particular, the low frequency components of a signal are more important than the high frequency components. Removing 50% of the bits from the high frequency components might remove, say, only 5% of the encoded information.

JPEG encoding starts by breaking the image into 8×8 pixel groups. The full JPEG algorithm can accept a wide range of bits per pixel, including the use of color information. In this example, each pixel is a single byte, a grayscale value between 0 and 255. These 8×8 pixel groups are treated independently during encoding. That is, each group is initially represented by 64 bytes. After transforming and removing data, each group is represented by, say, 2 to 20 bytes.

During encoding, the inverse transform is taken of the 2 to 20 bytes to create an approximation of the original 8×8 group. These approximated groups are then fitted together to form the uncompressed image. Why use 8×8 pixel groups instead of, for instance, 16×16? The 8×8 grouping was based on the maximum size that integrated circuit technology could handle at the time the standard was developed. In any event, the 8×8 size works well, and it may or may not be changed in the future.

Before encoding, the color tables are constructed (Figure.2):

- Images are composed of dots called pixels
- Each pixel gets its own color by combining percentages of red, green and blue (RGB)
- Each of these colors has value from 0 to 255
- Zero designates that the color is present
- 255 designates complete saturation of that color
- RGB color model has 16,777,216 possible colors
- Total of 255x255x255



231	224	224	217	217	203	189	196
210	217	203	189	203	224	217	224
196	217	210	224	203	203	196	189
210	203	196	203	182	203	182	189
203	224	203	217	196	175	154	140
182	189	168	161	154	126	119	112
175	154	126	105	140	105	119	84
154	98	105	98	105	63	112	84

Figure. 2. Color Table Construction

5. Detail Implementation

The system is to secure message by embedding in a palette base image. So, we construct bit table of image and text characters.

An English message text is written by using the alphabetic characters of the English language (which are 26 letters ('a'…'z')). Some other special characters are useful to use in writing messages which are giving the reader a good understanding of the message. Some of these characters that are adopted in this study: ('space character', '.', ',', '(' , ')' , '''). Therefore, the total numbers of characters that are used to write a message become 32-characters. This means that we need at least 5-bits to represent these 23-characters in any digital system. a gray scale image is using 256 gray scales for each pixel in it. This means that we need (1- byte ≡ 8-bits) per pixel to produce (2$_{(8\text{-bits})}$ ≡ 256) grayscales.

The proposed method is akin to the idea of palette-based image format by color quantization. However, unlike the quantization in generating the color palette, which requires cumbersome computation of centroids in the color space, the quantization is done only by grouping two similar color entries in the palette into the same color. For example, if we select two colors **a** and **b** with a small distance by the color map (palette), we will then assign a new color to both of these entries so that they represent one identical color. We can then assign for instance binary choice 0 to **a** and 1 to **b** to represent the stego message. It is apparent that the distortion on the cover image is independent on the embedded data stream, since the distortion is introduced when assigning entry **a** and **b** with the quantized color. Since they are then identical, they can be used interchangeably, without affecting the outlook of the image.

5.1. Encoding Procedure

For a palette-based image X with the color map, a set of RGB color triplets P = f(r1; g1; b1); :::; (rn; gn; bn)g and the embedding binary secret message M, such that │M│ · │X│, (│¢│ denotes the length) equal to the length of the image, the embedding algorithm works in 5 steps:

1. Construct the color table 256 RGB pattern of the image.
2. Construct 8-bit array of the text messages.
3. Embed the text message bits to the least significant bit to the image bit arrays (pixels).
4. Calculate the new quantized color by.

$$\hat{C}_{ij} = \frac{P(i)\xi(X,i)+P(j)\xi(X,j)}{\xi(X,i)+\xi(X,j)}$$

which is basically a weighted interpolation between P(i) and P(j), where $\xi(X,i)$ denotes the number of occurrences of color c on image X. We will then set

$$P(i) = P(j) = \hat{C}_{ij} .$$

5. For every embeddable position on X, the color entry number is denoted as x. If the message bit to be embedded is 0, then replace x with i, or j if otherwise.

5.1. Decoding Procedure

The procedure is to extract the data from image, as long as we identify the color group of configuration. This is straightforward because palette entries in the same group are quantized into one color. The procedure for extracting the message is as follows:

1. Initialize the color table B with all 256 color entry numbers.
2. Find two colors i, j € B (i < j), which P(i) =P(j), and add the ordered-pair (i; j) into the configuration set A and remove I, j from B.
3. Iteratively repeat step 2 until no such pair can be found.
4. For every pair (i, j) € A, we replace i on X with bit 0, and j with bit 1.
5. Read these bits out as the message M.

6. Experimental Result

The image is a collection of dots, such as pixels. So, we construct the color pixels array table firstly. And, we generate the bit pattern of the message. And then, we embed the message bit to the bit array of the color image. In the retrieving process, we extract the bit pattern from the image, and then map to the color pixels to letter.

Firstly, we get a color image (Figure.3) to embed the message. The system will construct the RGB color arrays to bind the massage bits. Then the required message converts to the bit patterns and combine with the above two bit pairs. These message bit map to the pixels (Figure.4) and embed with image (Figure.5).

Figure. 3. Original Image

Figure.4. Embedded message (mapping to pixels)

Figure.5. Encoded Image

The resulting image is not destroying and it will still as in original image. But, this image is bind together with the message. So, it is secure because it can not know the message in image if the image will get the other third party. The following table is the steganalysis of the tested image-message pairs. The process of embedding message in the image is analyzed the overlapping and non-overlapping groups by dividing with color pixels.

Non-overlapping groups	
Percentage in red:	4.04295
Approximate length (in bytes) from red:	139.72418
Percentage in green:	11.56377
Approximate length (in bytes) from green:	399.64397
Percentage in blue:	21.16609
Approximate length (in bytes) from blue:	731.50007
Overlapping groups	
Percentage in red:	6.27098
Approximate length (in bytes) from red:	216.72507
Percentage in green:	8.90845
Approximate length (in bytes) from green:	307.8759
Percentage in blue:	21.59493
Approximate length (in bytes) from blue:	746.32085
Average across all groups/colors:	12.25786
Average approximate length across all groups/colors:	423.63167

Table.1. Steganalysis of Encoding Process

The sender can send the encoded image to the receiver. If the receiver will get that image, he must extract the embedded message pixels and then translate to message letters.

7. Conclusion

The system is hiding data in an image to secure data between sender and receiver in communication process. Because of the LSB encoding is used, the image may not different the original image. The system uses the color image by

EzStago and LSB encoding. So, the system can be extended in other embedding algorithm. The paper contributes the security system with steagnography by mapping characters to pixels and embeds these pixels to the image without changing the sense of the human.

References

[1] B.A. Forouzan, *Cryptography and Network Security*, McGraw Hill International Edition, Singapore, 2008.

[2] C. Kufman, R. Perlman, M. Speciner, *Network Security Private Communication in a Public World,* Prentice Hall PTR, New Jersay, 2002.

[3] J. Ankit, *A solution for data hiding*, Department of Computer Science and Engineering, Guru Nanak Dev Engineering College, Ludhiana.

[4] M. Mohamed, M.R. Katmin, *Information Hiding Using Steganography,* Department of Computer System & Communication Faculty of Computer Science and Information System, University Technology, Malayisa, 2009.

[5] N.F. Johnson and S. Jajodia. Steganalysis of images created using current steganography software. In Proc. The Second Inform. Hiding Workshop LNCS, volume 1525, pages 273–289. Springer-Verlag, 1998.

[6] D. Kahn. The history of steganography. In R. Anderson, editor, *1st Information Hiding Workshop, Lecture Notes in Computer Science*, volume 1174, pages 1–5, Springer-Verlag, 1996.

[7] R.J. Anderson, M.G. Kuhn. University of Cambridge. (April 1998).
Attacks on Copyright Marking Systems

[8] R. Machado. Ez stego. http://www.stego.com.

[9] T. Morkel, J.H.P. Eloff, M.S. Oliver, *An Overview of Image Steganography*, Information and Computer Security Architecture Research Group, Department of Computer Science, University Pretoria, South Africa.

[10] P. Fred and M. Sean, *Cryptography: A Very Short Introduction*, Oxford University Press, 2002.

[11] N. Johnson, S. Jajodia, *Exploring Steganography: Seeing the Unseen"*, IEEE Computer, pp.26-34, 1998.

[12] N. Johnson and S. Jajodia, *Stegaanalysis of Images Created Using Current Steganography Software*, Leacture Notes in Computer Science, pp.273-289, 1998.

[13] C. Cachin, *Digital Steganograph*, IBM Research, Zuich Research Laboratory, Switzerland, 2005.

[14] D.Bloisi, L. Iocchi, *Image Based Steganography and Cryptography*, Department of Information Science, Sapienza University of Rome, Italy.

[15] R.J. Anderson, M.G. Kuhn, *Attacks on Copyright Marking Systems*, University of Cambridge, 1998.