An Application of Palette Based Steganography

Prof. Samir Kumar Bandyopadhyay
Professor, Dept of Computer Science & Engineering,
University of Calcutta,
92 A.P.C. Road, Kolkata – 700009, India

Indra Kanta Maitra
Research Fellow, Dept of Computer Science & Engg,
University of Calcutta,
92 A.P.C. Road, Kolkata – 700009, India

ABSTRACT
Steganography is the art of writing hidden messages in such a way that no one; apart from the sender and intended recipient even understand there is a hidden message. Steganography includes the concealment of information within computer files. One of the most common methods of implementation is Least Significant Bit Insertion, in which the least significant bit of every byte is altered to form the bit-string representing the embedded file. Altering the LSB will only cause minor changes in color. While this technique works well for 24-bit color image files, steganography has not been as successful when using an 8-bit color image file, due to limitations in color variations and the use of a color table. Color table is organized as- the first three bytes correspond to RGB components and the last byte is reserved or unused. The proposed technique is to generate the image from a 24-bit bitmap to an 8-bit bitmap using color quantization resulted in minor variations in the image, which are barely noticeable to the human eye. However, these slight variations aid in hiding the data.

Keywords
Steganography, Least significant bit (LSB), color Table, Color Quantization Algorithm, Euclidean Distance.

1. Introduction
Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography.

1.1. Steganography
The word steganography is originally derived from Greek words, which mean "Covered Writing". It has been used in various forms for thousands of years. In the 5th century BC Histaiacus shaved a slave's head, tattooed a message on his skull and the slave was dispatched with the message after his hair grew back [4, 5, 10, 9]. In Saudi Arabia at the King Abdulaziz City of science and technology, a project was initiated to translate into English some ancient Arabic manuscripts on secret writing which are believed to have been written 1200 years ago.

Some of these manuscripts were found in Turkey and Germany [12]. Five hundred years ago, the Italian mathematician Jérôme Cardan reinvented a Chinese ancient method of secret writing. The scenario goes as follows: a paper mask with holes is shared among two parties, this mask is placed over a blank paper and the sender writes his secret message through the holes then takes the mask off and fills the blanks so that the message appears as an innocuous text. This method is credited to Cardan and is called Cardan Grille [4].

With the boost in computer power, the internet and with the development of digital signal processing (DSP), information theory and coding theory, steganography has become "digital". In the realm of this digital world steganography has created an atmosphere of corporate vigilance that has spawned various interesting applications, thus its continuing evolution is guaranteed. Contemporary information hiding is due to [13]. One of the earliest methods to discuss digital steganography is credited to Kurak and McHugh [7], who proposed a method, which resembles embedding into the 4 LSBs (least significant bits). They examined image downgrading and contamination, which is known now as image-based steganography.

Information can be hidden inside a multimedia object using many suitable techniques. As a cover object, we can select image, audio or video file. Depending on the type of the cover object, definite and appropriate technique is followed in order to obtain security.

1.2. Least Significant Bit
Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [9]. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue color components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data [6]. For example a grid for 3 pixels of a 24-bit image can be as follows:

(00101101 00011100 11011100)

(10100110 11000100 00001100)

(11010010 10101101 01100011)

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [6]. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. The human eye cannot perceive these changes, thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference [4].

A slight variation of this technique allows for embedding the message in two or more of the least significant bits per byte. This increases the hidden information capacity of the cover-object, but the cover-object is degraded more, and therefore it is more detectable. Other variations on this technique include ensuring that statistical changes in the image do not occur. Some intelligent software also checks for areas that are made up of one solid color. Changes in these pixels are then avoided because slight changes would cause noticeable variations in the area [14, 1].

While LSB insertion is easy to implement, it is also easily attacked. Slight modifications in the color palette and simple image manipulations will destroy the entire hidden message. Some examples of these simple image manipulations include image resizing and cropping [10, 3].

2. Palette Based Image

Palette based or Indexed colors image that enables 8 bits per pixel or less to look almost as good as 24 bits per pixel. The technique determines the 256 most frequently used colors in the image and creates a color lookup table, also called a color map or color palette, which is stored with the image. Rather than each pixel in the image having all three RGB colors (one 8-bit red, one 8-bit green and one 8-bit blue), each pixel contains one 8-bit number that indexes into the 256-color lookup table, which contains the RGB values. This is reducing images to their smallest size and these images are most commonly used on Web pages, as they are small and quick to load. The 256-color palette is mapped for best results on the Internet.

When early computer screens were commonly limited to 256 colors, indexed color methods were essential. Even then, two indexed photos on screen at the same time with vastly different color schemes would overload the hardware's color capacity and display improperly. Today, computer hardware easily renders full 24-bit color, but 8-bit indexed images are still widely used because file size is still of utmost importance and smaller file sizes are optimal in network communication although network speed is increasing and bandwidth problems are decreasing. Thus, the current steganographic use of 24-bit images leads to slower communication and development of an 8-bit image format would be beneficial.

3. LSB and Palette Based Image

Palette based images, for example BMP images, are another popular image file format commonly used on the Internet. An indexed BMP image cannot have a bit depth greater than 8, thus the maximum number of colors that a BMP can store is 256 [11]. BMP images are indexed images where the colors used in the image are stored in a palette, sometimes referred to as a color lookup table [11]. Each pixel is represented as a single byte and the pixel data is an index to the color palette [4]. The colors of the palette are typically ordered from the most used color to the least used colors to reduce lookup time [4].

BMP images can also be used for LSB steganography, although extra care should be taken. The problem with the palette approach used with BMP images is that should one change the least significant bit of a pixel, it can result in a completely different color since the index to the color palette is changed [4]. If adjacent palette entries are similar, there might be little or no noticeable change, but should the adjacent palette entries be very dissimilar, the change would be evident [4]. One possible solution is to sort the palette so that the color differences between consecutive colors are minimized [2]. Another solution is to add new colors, which are visually similar to the existing colors in the palette. This requires the original image to have less unique colors than the maximum number of colors (this value depends on the bit depth used) [8]. Using this approach, one should thus carefully choose the right cover image.

4. Proposed Technique

The proposed technique is to compressing the image from a 24-bit bitmap to an 8-bit bitmap resulted in minor variations in the image, which are barely noticeable to the human eye. However, these slight variations aid in hiding the data. Since there would not be an original 8-bit image to compare with the stego-image, it would be impossible to discern that the slight variations caused by hiding the data are different from the slight variations caused by compression.

This steganographic implementation for 8-bit images enabled smaller file sizes to be utilized in steganographic communications. While also limiting the size of the hidden file, this implementation addressed issues that have been passed by in other applications, and provided a more compact vehicle for those secret communications that do not require a large cover-file.

4.1. Palette Generation

Traditionally, color image quantization is used to reproduce 24-bit images with a limited number of simultaneous colors. This fundamental concept of color image quantization is further extended to include the reproduction of an already quantized color image, typically with several hundred to several hundred thousand colors in a 24-bit RGB format, using a much smaller color palette, typically with 256 or less 24-bit colors display. Quantization inevitably introduces distortion.

Figure 1. 24-Bit Image is divided into 240 Blocks.

Ideally, a quantization algorithm should distribute any visible distortion "evenly" throughout the quantized image so that none stands out to be found particularly objectionable by an average human observer. This means that many factors have to be considered including color space, image context, viewing condition, and viewer's experience and aesthetic judgment [15]. Here we proposed a color quantization method to generate a reduced color space.

In an 8-bit indexed color image has a maximum of 256 colors. Initially maximum 240 colors are created from the original image. Sixteen colors will be added to the palette by the time the final picture is written. In order to select the 240 original colors, the image is divided into fifteen rows and sixteen columns to generate maximum 240 blocks, as seen in figure 1. One color is chosen from each of these blocks by randomly selecting a set of X and Y coordinates within each block. Calculations are then made to determine the index of the pixel in the color table that represents the image data. A color table is a structure containing four bytes, one each for the red, green, and blue intensity and a reserved byte, is depicted in figure 2.

Index	Reserved (1 byte)	Blue (1 byte)	Green (1 byte)	Red (1 byte)
0	0	B_0	G_0	R_0
1	0	B_1	G_1	R_1
2	0	B_2	G_2	R_2
…	…	…	…	…
…	…	…	…	…
254	0	$B_{(n-1)}$	$G_{(n-1)}$	$R_{(n-1)}$
255	0	B_n	G_n	R_n

Figure 2. Color Table Structure.

Each time a color is selected from a block, it is compared to every other color in the color table, and the minimum difference between any two colors is calculated. If this difference is lower than a certain level, then the new color is discarded and another color is selected from the image (lower level is depending on the size of the color space of a particular image). After five attempts to find a color from a certain block that differs enough from all the other colors in the color table, the selected color is appended to the color table else moves to the next block. The method is described in algorithm 1.

Algorithm 1: Palette Generation

```
Input: 24-bit Bitmap Image
Output: 240-Color Color Table
B=Block
T=Terms
P=Pixel Information
Begin
Step1.  B=240
Step2.  Loop I=1, 240
            Read B
            Loop T=1, 5
                P = Random(X), Random(Y)
                If P in Palette OR < Lower Level
                    Skip
                Else
                    Append in Palette
                    Break
                End IF
                T=T+1
            End Loop
            I=I+1
        End Loop
Step3.  End
```

4.2. Optimizing the Palette

The initial color table contains 240 colors that were picked out of the original image. These colors were chosen from the entire image but that does not guarantee that these colors are the most representative of the colors that exist in the image. Therefore, the Color table must then be optimized to provide the best 240 colors for the colors in this particular image.

Now the palette is fixed, a pixel is chosen from the original 24-bit image and its RGB values are measure to the RGB values of every color in the color table to determine the distance between two using Euclidean distance methods. In mathematics, the Euclidean distance is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula.

$$Euclidean\ Distance = Sqrt\ ((r_2-r_1)^2+(g_2-g_1)^2+(b_2-b_1)^2)$$

For each comparison a distance level is calculated using the Euclidean distance method of the red, green, and blue component of the color. The color table color that produces the smallest amount of distance is the color table color that is closest to the pixel's RGB values. The RGB values of the pixel are then added to the RGB values of the color table color. A counter is implemented to keep track of how many pixels are assigned to each color table color and is incremented each time a match is found. Once the algorithm has gone through the whole image, dividing the red, green, and blue values by the counter for that particular container averages the RGB values of each color table container. This process produces a Color Table with slightly new, "better" colors in it. The process is repeated until the new color table is optimized. To determine when the color table is optimized, the distance levels are recorded during each run and when a certain distance level is attained the algorithm is finished.

4.3. Sorting the Palette

Each pixel in an 8-bit color image is an 8-bit pointer to a 24-bit color in the color table. Looking ahead to the LSB insertion, a pixel pointing to a pink color could suddenly point to a green color by a simple flip of the least significant bit. In order to reduce dramatic noise such as that, the color table was sorted so that similar colors are next to each other before the pixels are assigned to Color Table colors. The sorting algorithm works as follows.

The initial color of a color table is 0, 0 and 0 for red, green and blue respectively. For each entry of color table Euclidean distance is calculated and sorting the color of color table is done by using said calculated distance. Using a certain distance of distance value calculated, rest 16 colors are generated by simply averaging two consecutive colors of the color table and insert the color in between the two. So, sorted 256-color color table is ready to handle the LSB insertion method.

4.4. Generating the 8-Bit Image

After generating the sorted 256-color or 8-Bit palette or color table, a pixel is chosen from the original 24-bit image and its RGB values are compared to the RGB values of every color in the color table using the Euclidean distance method. For each

comparison a distance is calculated of the red, green, and blue color components. The color table color that produces the smallest amount of distance is the color table color that is assigned to this pixel.

4.5. Hiding and Unhiding Information

The hiding function takes three parameters and two steps in order to complete. The data string of text, picture data and binary data string are the three parameters for the first step in hiding the text into the image. The first step in the hiding function is to convert the ASCII text into its binary equivalent. In order to do this, each character of the text message is converted to its ordinal number i.e. 'a' = 97. The ordinal number is then converted to binary using the following method called the division-remainder routine. An ordinal number is divided by two using the mod function. This function returns either a one or a zero, which is then placed in a remainder array. This is continued until the dividend is zero. The remainder array, which is binary equivalent of text, is ready to hide.

We will choose the color pair from the color table, which has the minimum Euclidean distance among all the color entries of the color table i.e. $(r_{n+1}, g_{n+1}, b_{n+1})$ and $(r_{n+2}, g_{n+2}, b_{n+2})$ pair has the minimum Euclidean distance. We select the first one for LSB insertion because that will produce the minimum color change in the image in compare to all. The hiding of text will be sequentially altering the least significant bit of the image data as necessary. To detect the stating and ending of hiding information a specific bit pattern may be used for unhiding the text information in receiver end.

Receiver will calculate the Euclidean distance of color table, identify the minimum distance color pair and sequentially read the image data with the index value n+2. When he found the starting bit pattern collect the LSB and continue until the ending bit pattern. The collected LSB received will reconstruct the desired information to be unhiding.

5. Conclusion

What we have proposed is simple but effective method in the field of steganography. The said method describes a technique to successfully hiding and unhiding data in an 8-bit color image. At end, this can be said that the color quantization algorithm may be improved, palette optimization may be more accurate, cryptography may be used before hiding or other future improvement may be incorporated. Lastly it is expected by the authors that any kind of future endeavor in said field will definitely route it a path to design a secure system using the proposed algorithm for both Internet and Mobile Communication Technology.

6. Reference

[1] Avcibas I, "Steganalysis Using Image Quality Metrics", IEEE Transactions on Image Processing, Vol 12, No. 2, February 2003.

[2] Chandramouli R, Kharrazi M & Memon N, "Image steganography and steganalysis: Concepts and Practice", Proceedings of the 2nd International Workshop on Digital Watermarking, October 2003

[3] Chandramouli R and Memon N, "Analysis of LSB Based Image Steganography Techniques", IEEE 2001.

[4] Johnson N.F. and Jajodia S, "Exploring steganography: Seeing the Unseen", IEEE Computer, 31(2) (1998) pp 26-34.

[5] Judge J.C, "Steganography: Past, present, future", SANS Institute publication, http://www.sans.org/ reading_room/whitepapers/stenganography/552.php, 2001.

[6] Krenn, R., "Steganography and Steganalysis", http://www. krenn.nl/ univ/cry/steg/article.pdf

[7] Kurak C. and McHugh J, "A cautionary note on image downgrading", Proceedings of the IEEE 8th Annual Computer Security Applications Conference, 30 Nov-4 Dec 1992, pp 153-159.

[8] Moerland T, "Steganography and Steganalysis", Leiden Institute of Advanced Computing Science, www.liacs.nl/home/ tmoerl/privtech.pdf

[9] Moulin P and Koetter R, "Data-hiding codes", Proceedings of the IEEE, 93 (12) (2005) pp 2083-2126.

[10] Provos N and Honeyman P, "Hide and seek: An introduction to steganography", IEEE Security and Privacy, 01 (3) (2003) pp 32-44.

[11] Reference guide: Graphics Technical Options and Decisions, http://www.devx.com/projectcool/Article/19997

[12] Sadkhan S.B, "Cryptography: Current status and future trends", Proceedings of IEEE International Conference on Information & Communication Technologies: From Theory to Applications, Damascus. Syria, April 19-23, 2004, pp 417-418.

[13] Simmons G.J, "The prisoners' problem and the subliminal channel", Proceedings of International conference on Advances in Cryptology, CRYPTO83, August 22-24, 1984, pp 51-67.

[14] Tucker P, "Hiding secrets in computer files: steganography is the new invisible ink, as codes stow away on images", An article from The Futurist

[15] Xiang Z, "Color Image Quantization by Minimizing the Maximum Intercluster Distance", ACM Transactions on Graphics, Vol 16, No 3, July 1997, pp 260 - 276