Research Inventy: International Journal of Engineering And Science
Vol.4, Issue 2(February 2014), PP 40-46
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Data Hiding Scheme in Digital Image using Steganographic Techniques: International Journal of Engineering and Science

[1]Narsimha Banothu, [2]K Srinivas, [3]K Krishna Reddy
[1,2,3]*(Holymary Institute of Technology and Science)*
(narsimha532@gmail.com)

ABSTRACT *- Steganography is hiding the secret data into a cover object to protect it from unauthorized access. It is a technique of invisible communication which hides the existence of the message. If the cover object used is an image, the steganography is known as image steganography. It has many applications like Online transactions, military communication etc. In this paper we are discussing Existing important image steganography techniques like Least Significant Bit (LSB), Pixel Value Differencing (PVD), and Modified Kekre Algorithm (MKA) etc.*

Keywords*: Steganography, digital image, LSB, cover- image, stego-image, Peak Signal to Noise ratio, Mean Square Error.*

.

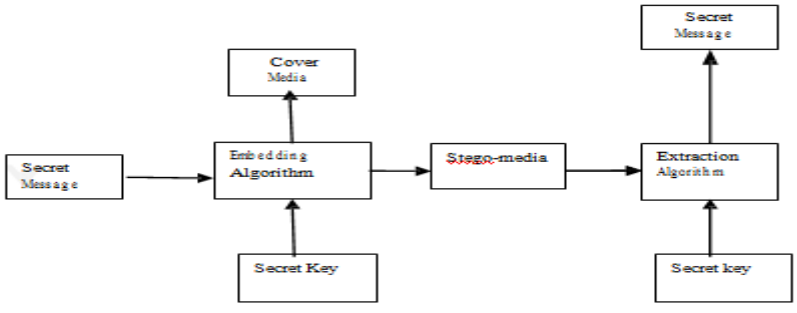
I. INTRODUCTION

The word steganography comes from Greek word steganos which means covered or secret and the graphy means writing or drawing. So, literal meaning of steganography is "covered writing" [12]. Generally steganography is known as invisible communication. Cryptography provides confidentiality, steganography on the other hand hide the message and there is no knowledge of the existence of the message. In simple words, it is hiding the information into other information. Steganography do not alter the message structure but hides inside a cover object. Steganography and cryptography are different techniques. Steganography hides the information and cryptography protects the information. Due to hidden or invisible factor it is difficult to recover hide information. Procedure to know the steganography technique is known as steganalysis.

A good steganographic method should possess following three main properties [9]:

- **Capacity**- the amount of the secret data to be hidden without significantly change in the cover image.
- **Robustness**- resistance for possible modification and
- **Invisibility**- detection of the existence of the secret data can't be notified by anybody except receiver.

Some of the basic term used in steganography: **Message**: Actual information which is used to hide into images. Message could be a plain text or other image.
Cover-object: It refers to the object used as a carrier to embed message into.
Stego-object: Object which carrying a hidden message. **Stego-key**: A key refers to a password used to hide and later retrieval of message.
Embedding algorithm: An algorithm used to hide the message.
Extracting algorithm: An algorithm used to unhide/uncover the message
Basic steganography diagram shown in fig 1.



A. Fig 1 Basic steganographic process

In the basic steganographic process, the secret message is hidden into a cover object. The cover object can be any of text, image, audio, video etc. A secret key is also used and the secret message is embedded into the cover object using the secret key. This new message obtained is called stego message. The stego message is transmitted over the public channel. The receiver gets the message and retrieves the message using the stego key which is same as used by the sender. In this way security is achieved by hiding the existence of the message.

image steganography is a process that hides the message into cover-image and generate a stego-image. That stego-image then sent to the receiver without anyone else knowing that it contain the hidden message. The receiver can extract the message with or without stego- key that depends on the hidden scheme [4].

II. TECHNIQUES OF IMAGE STEGANOGRAPHY

There have been a large embedding techniques proposed number of steganography in the literature. These techniques modify the cover-image with different approaches as well as constrains. But all embedding techniques share the important goal of maximizing the capacity of the stego channel. In other words their aim is to embed at highest possible rate while remaining undetectable to steganalysis attack. All the popular data hiding methods can be divided into two major classes: spatial domain embedding and transform domain embedding. Next we will review them.

A. *Spatial Domain*

Spatial domain techniques embed information in the intensity of the original image pixels directly. Basically least significant bit (LSB) method is used where it replaces the least significant bit of original pixel with the message bit.

B. *Transform Domain*

Transform domain also known as frequency domain where images are first transformed then the message is embedded in the image. Discrete cosine transformation (DCT) technique is used in JPEG images to achieve compression. DCT is a lossy compression transform where the cosine values cannot be generated as original, because DCT alter values to hide the information

2.1. MLSB Technique

The most basic and important image Steganographic Technique is Least Significant Bit [2, 7] embedding technique. In this technique data can be hidden in the least significant bits of the cover image and the human eye would be unable to notice the hidden image in the cover file. This technique can be used for hiding images in 24-bit, 8-bit or gray scale format. In this technique, least significant bit of each pixel is replaced with secret message bit until message end. When using a 24 bit image one can store 3 bit in each pixel by changing a bit of each if the red, green and blue color components. An 800 x 600 pixel image can store 1,4400,00 bits or 180,000 bytes of embedded data. For example a 24 bit can be as follows:

(10110101 01101100 10101101)
(10110110 11001101 00111110)
(10110101 01100011 10001110)

The number 150 which binary representation is 10010110 is embedded into the least significant bits of this part of the image, the resulting grid as follows: (1011010**1** 0110110**0** 1010110**0**)
(1011011**1** 1100110**0** 0011111**1**)
(1011010**1** 0110000**1** 1000111**0**)

Although the number is embedded into the first 8 bytes of the grid, only the 3 underlined bits need to be changed according to the embedded message. On an average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. There are 256 possible intensities of each primary color, so, changing the LSB of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye, thus the message is successfully hidden. If the message is hidden even in the second to least significant as well as in least significant bit then too no difference is seen in the image. In LSB Technique, consecutive bytes of the image data from the first byte to the end of the message are used to embed the information. But this approach is very easy to detect. A more secure system can be in which the sender and receiver share a secret key that specifies only certain pixels to be changed. Even if the intruder suspects that LSB steganography has been used, there is no way of knowing which pixels to target without the secret key. In its simplest form, LSB makes use of BMP images, since they use lossless compression. To hide a secret message inside a BMP file, one would require a very large cover image. For this reason, LSB method has also been developed for use with other image file formats. This type of information hiding algorithm could be a major risk because eavesdropper can apply sequential scanning based techniques [4] to recover the secret message

2.2. Pixel Value Differencing Scheme

Wu & Tsai [12] discuss Pixel Value Differencing (PVD) scheme. This technique takes advantage of the characteristics of human visual system. In this technique, the original cover image is divided into non overlapping blocks of two pixels. A range table with a number of contiguous ranges is fabricated. The width of each range in the table is in power of 2. Now, difference is calculated between two consecutive pixels of a block. The block with large difference value is considered in edge area and with small difference value is considered in smooth area where the small or large values are taken depending upon some pre-specified threshold value. The human eyes are more sensitive to noise in smooth area than in the edge area. This method embeds more bits in edge areas in contrast to smooth areas. This technique doesn't have sufficient embedding capacity. Another technique [13] discussed by Wu, Tsai and Hwang that also exploits the characteristics of the human visual system. In this method, the image is also divided into non-overlapping blocks of two consecutive pixels and then the difference value is calculated for each block in similar way as in [13]. On the basis of the difference value, each block is identified either as a part of smooth region or edge region. This method embeds the secret data bits into the smooth regions by simple LSB substitution method and for edge area the Wu & Tsai's scheme [12] is used. Thus, it increases the data hiding capacity to a great extent without disturbing the image quality much. The method discussed in [12, 13] identify the horizontal edges only.

Range (R)	**er bound (LB)**	**er bound (UB)**
R_1	0	15
R_2	16	31
R_3	32	63
R_4	64	127
R_5	128	255

A. *Table-1 Range Table*

2.3. Modified Kekre's Algorithm (MKA)

Modified Kekre's Algorithm (MKA) [5] is based on Least Significant Bit (LSB) method. MKA can be applied on 8 bit gray scale images or 24 bit Read Green Blue (RGB) color image. It uses up to five LSB's of a pixel to embed the data. The number of secret data bits that can be embedded in the pixels depends upon the pixel intensity of the pixels of the cover image. To achieve more security MKA uses 8 bit secret key to perform XOR operation to all the bytes of the secret message. While extracting the message XOR operation is also performed using the same key. The embedding algorithm maintains a matrix of pixels where up to 5 bits of message are used to embed, and this matrix is required while extracting the secret hidden message from stego-image. In Table-2.1 'x' shows don't care bit whose value can be either '0' or '1'. "Pixel intensity" shows the value of pixel. "Data Bit to Embed" shows current message bit used to embed into the cover- image. "Matrix Entry" maintains a matrix. "Utilize Bits" shows the total number of bits embedded into a pixel of the cover image.

Suppose pixel intensity is 245 which exist in the row number 1 and 2 of the Table-2.1. For embedding the secret data bits its 5 or 4 LSBs can be utilized, depending on current data bit to embed. If current data bit is 0 then 4 LSBs otherwise 5 LSBs are used for embedding data bits. Mark a 1 bit entry into maintained matrix pixel position to identify that if this pixel contains 5 bits of data. Same procedure is applicable for other pixels of the image according to Table-1.

S.No	**Pixel Intensity**	**Data bit to Embed**	**Matrix Entry**	**Utilize Bit/bits**
1	**240-255**	**1**	**1**	**5**
2	**240-255**	**0**	**-**	**4**
3	**224-239**	**0**	**1**	**5**
4	**224-239**	**1**	**-**	**3**
5	**192-223**	**X**	**-**	**2**
6	**0-191**	**X**	**-**	**1**

TABLE 2 MODIFIED KEKRE ALGORITHM SCHEME [5]

Where x: Don't care bit.

Same procedure will be run to extract the data bits of message using maintained matrix because it keeps the track of the pixel position where 5 LSBs are utilized. At the end, 8 bit secret key with XOR operations applied on the extracted message to regenerate original message which was embedded.

Hussain [6] discusses a method that is an improvement of MKA [5]. It also applies 8 bit secret key with XOR operation on all bytes of message to change the originality of message. It maintains a matrix for those pixels which will embed 5, 3 and 2 LSBs of data. In Table-2.2 "Pixel intensity" shows the value of pixel. "Data Bit to Embed" shows current message bit used to embed into the cover-image. "Matrix Entry" maintains a matrix which denotes the 5 LSB are embedded. "Utilize Bits" shows the total number of bits embedded into a pixel.

If pixel intensity is 33 which exist in row number 7 and 8 of Table-2.2. For data embedding, 2 or 1 bits of pixel can be utilized depending on current data bit to embed. If current message (want to embed data) bit is 0 then 2 bits otherwise 1 LSB are used for embedding data bits. If this pixel contains 2 bits of data, mark a 1 bit entry into maintained matrix pixel position for identification of extra bits. Same procedure is applicable for other pixels of the image according to Table-2.2.

TABLE 3 Hussain's Modified MKA[6]

S.No	Pix el Intensi	Data Bit to Emb	Matrix Entry	Utilize Bit/Bits
1	240-255	1	1	5
2	240-255	0	-	4
3	224-239	0	1	5
4	224-239	1	-	3
5	192-223	0	1	3
6	192-223	1	-	2
7	32-191	0	1	2
8	32-191	1	-	1
9	16-31	0	1	3
10	16-31	1	-	2
11	0-15	0	1	5
12	0-15	1	-	4

5	192-223	0	1	3
6	192-223	1	-	2
7	32-191	0	1	2
8	32-191	1	-	1
9	16-31	0	1	3
10	16-31	1	-	2
11	0-15	0	1	5
12	0-15	1	-	4

III. IMAGE QUALITY METRICS

The image quality metrics are figures of merit used for the evaluation purpose of the image quality. These metrics provide some measures of the closeness between two digital images by exploiting the differences in the statistical distribution of pixel values. The most commonly used quality metrics are:

- Mean Square Error (MSE)
- Root Mean Square Error (RMSE)
- Structural Similarity (SSIM)
- Peak Signal to Noise Ratio (PSNR)

3.1. Mean Square Error (MSE)

The mean square error is defined as the square of the difference between the pixel values of the original image and the stego image and then dividing it by size of the image. The mathematical formula for computing mean square error between x and y images of sizes M*N is given below

MSE =

$$\frac{1}{M*N}\sum_{x=1}^{M}\sum_{y=1}^{N}[x(m,n)-y(m,n)]^2$$

The lower value of Mean Square Error (MSE) signifies lesser error in the stego image in other words better quality.

3.2. Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is calculated by getting the square root of the mean square error (MSE). The RMSE can be calculated as follows.

RMSE = $\sqrt{MSE}$

3.3. Structural Similarity (SSIM)

The SSIM metric was given by Wang el. This method is used to measure the similarity between two images [4]. It is designed by modeling any image distortion as a combination of three factors that are loss of correlation, luminance distortion and contrast distortion. Mathematically, the SSIM is calculated as follows: Where,

$$SSIM(x,y) = l(x,y)\cdot c(x,y)\cdot s(x,y)$$

$$l(\mathbf{x},\mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

$l(x, y)$, luminance comparison function is used to measure the closeness of two images and μ_x and μ_y are the mean luminance used to calculate $l(x,y)$. $C(x, y)$ contrast comparison function which measures the closeness of the contrast of the two images and σ_x , σ_y are standard deviation. $s(x, y)$ structure comparison function which measures the correlation coefficient between the two images x and y and σ_{xy} is the covariance between x and y. C_1, C_2, C_3 are the positive constants.

3.4. Peak Signal to Noise Ratio (PSNR)

The Peak Signal to Noise Ratio (PSNR) measures the estimates of the quality of stego image compared with an original image and is a very commonly used metric way to measure image reliability or conformity. The mathematical formula to calculate the PSNR value is as follows:

PSNR = $20log_{10}[\frac{\mathbf{MAXPIX}}{\mathbf{MSE}}]$

Where MAXPIX is the maximum pixel value and MSE is the Mean Square Error.

In PSNR, 'signal' is the original image and 'noise' is the error in the stego image resulting due to encoding and decoding. PSNR is a number that reflects the quality of the stego image and is measured in decibel (dB). Mathematically, PSNR is inversely proportional to the MSE, which implies the lower the value of MSE higher is its PSNR. Thus higher the Peak Signal to Noise Ratio (PSNR) is better.

IV. IMAGE STEGANOGRAPHY CLASSIFICATIONS

Generally image steganography is categorized in following aspects and Table-4 shows the best steganographic measures.

High Capacity: Maximum size of information can be embedded into image.

Perceptual Transparency: After hiding process into cover image, perceptual quality will be degraded into stego-image as compare to cover-image.

***Robustness*:** After embedding, data should stay intact if stego-image goes into some transformation such as cropping, scaling, filtering and addition of noise.

Temper Resistance: It should be difficult to alter the message once it has been embedded into stego-image.

***Computation Complexity*:** How much expensive it is computationally for embedding and extracting a hidden message?

Table-4: IMAGE STEGANOGRAPHY ALGORITHM MEASURES

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

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

V. CONCLUSION

The methods discussed above have trade off in between capacity versus quality. In other words, when the number of bits of the secret data is low, the stego image quality is high and vice versa. MKA improves the capacity as well as PSNR value of the image by a great value and is much better than the LSB technique. MKA is applied to color images. PVD is also a good steganography technique and improves the capacity and quality in case of grey scale images. Limitation of the PVD is that it cannot be applied to color images. Before embedding the secret bits we can preprocess them in such a way that their numbers are reduced. The reduced bits of the secret data can give the actual secret data by performing reverse operation of the preprocessing. By doing this the data hiding capacity is increased without degrading the quality of the stego image.

REFERENCE

[1] Chandramouli R. and Memon N. (2001), "Analysis of LSB based Image Steganography Techniques", Proceedings of ICIP 2001, Thessaloniki, Greece, October 7−10.

[2] Deshpande N, Snehal K., "Implementation of LSB Steganography and Its Evaluation for Various Bits" K.K.Wagh Institute of Engineering Education & Research, Nashik India

[3] http://www.appliedtrust.com/resources/security/every-company-needs-tohave-a-security- program

[4] Johnson, N.F. & Jajodia, S. (1998), "Exploring Steganography: Seeing the Unseen", *Computer Journal*

[5] Kekre H.B, Athawale A,Halarnkar P.N(2009), "Performance Evaluation of Pixel Value Differencing and Kekre's Modified Algorithm for Information Hiding in Images", International Conference on Advances in Computing, Communication and Control, pp 342-346

[6] M. Hussain, M. Hussain., (2010) "Pixel Intensity Based High Capacity Data Embedding Method", Information and Emerging Technologies, International conference 978-1-4244-8003

[7] Morkel, T., Eloff, J.H.P & Olivier, M.S., (2005) "An overview of Image Steganography", Proceedings of Information Security South Africa (ISSA) Conference

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

[9] N. Tiwari and M. Shandilya, (2010) "Secure RGB Image Steganography from Pixel Indicator to Triple Algorithm-An Incremental Growth", International Journal of Security and Its Applications Vol. 4(4)

[10] Swain G, Lenka S.K (2011),"Steganography Using the Twelve Square Substitution Cipher and an Index Variable".

[11] Swain G, Lenka S.K (2010), "A Hybrid Approach to Steganography Embedding at Darkest and Brightest Pixels", International Conference on Communication and Computational Intelligence

[12] Wu D. C and Tsai W. H. (2003), "A steganographic method for images by pixel- value differencing", Pattern Recognition Letters, vol. 24, no. 9-10, pp. 1613-1626.

[13] Wu H.C., et al. (2005), "Image Steganographic scheme based on pixel-value differencing and LSB replacement methods", VISP(152), No.

[14] Babloo Saha, Shuchi Sharma (2012), "Steganographic Techniques of Data Hiding Using Digital Image", vol. 62, pp. 11-18