A NEW METHOD FOR IMAGE STEGANOGRAPHY AND ITS COMPARISON WITH LSB TECHNIQUE

Ashwin Deshpande[1], Suyash Dhondkar[2], Shreyas Godbole[3], Gaurav Jadhav[4]

[1]*Department of Computer Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India*
[2]*Department of Computer Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India*
[3]*Department of Computer Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India*
[4]*Department of Computer Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, Pune, India*

Abstract

In this paper we have proposed a new method for hiding a 24-bit color image into another 24-bit color image. The new approach introduced here uses the concepts of average value and floating point precision to replace the pixel rather than bit by bit replacement. It exploits the limitations of the human eye to spot minute variations and also provides enough complexity to avoid decryption by brute force. We have also analyzed the popular LSB technique and compared it with our technique on the grounds of ease of decryption and Peak signal-to-noise Ratio (PSNR). Eventually by means of above parameters we have concluded that the proposed method provides much better security than the LSB method with a comparable PSNR.

Keywords- *Steganography; Least Significant Bit; Weighted Mean; Information Security; PSNR*

----------***----------

1. INTRODUCTION

The problem of information security[1] is prevalent in the modern era, to counter these problems various methods for secured transmission of information are being researched. The traditional methods for secured transfer of data are cryptography and steganography. These methods have been used from ages, and evolved over the years to increase the security with advent of computers in the last few decades. Initially these methods were used to secure information by physical means [2], whereas now their implementations are changed to secure information in the digital world. The World Wide Web helps people to communicate information digitally. This has caused an enormous need for information security.

Cryptography [3] is the art of hiding messages so that they are decipherable to the intended recipient only. It is generally about building and analyzing protocols that prevent third parties or intruders from reading the private messages. It is the practice of hiding information by various obfuscatory techniques. It allows us to achieve data confidentiality, integrity and authentication. The word cryptography is derived from a Greek word which means hidden or secret. Prior to the modern age, cryptography was equivalent to encryption. But now encryption is part of cryptography which works on the principle of key(s) [3][4] which transforms the original data into cipher text.

A cipher is an algorithm or method used to perform encryption of data and then decryption of that data. The operation or method of cipher usually depends on both the algorithm and the key. The key is confidential string of characters, which is required to decrypt the cipher text.

Steganography is also a form of cryptography that inserts data into other mediums in undetectable ways, unlike encryption. Steganography [2] is the method of hiding text, images, video or audio into typically human view-able objects such as image, video or audio. The information encrypted using encryption techniques is easily noticeable as compared to steganography. Information concealed by encryption attracts attention due to its nature, whereas in steganography the information is hidden in plain sight. A simple implementation of steganography could be the use of invisible ink to write text in a document. Thus the inability to see correctness in a given element is the foundation for steganography. The earliest recorded uses of steganography can be traced back to 440 BC.

Image steganography is the method to hide text or an image (hidden image) in to another image (cover image) to form a resulting image (stegoimage). This stegoimage then can be sent to the recipient in a way that the actual content (hidden image) is not visible. The aim of this paper is to provide a new technique which provides with better encoding and decreases the difference between the cover image and the stegoimage.

In this paper we first describe the LSB method, then we explain our proposed method. Following which we present an analysis on the basis of complexity of decryption and

PSNR values for both the methods. Then we compare both the methods on the aforementioned basis and conclude which method is better.

2. THE LEAST SIGNIFICANT BIT ALGORITHM (LSB)

2.1 General Idea

The LSB[5][6] algorithm is based on the idea of embedding the pixel of the hidden image in to the LSB (Least Significant Bit) of the cover image bit by bit. The RGB pixel values of the hidden image are stored in bit form in the LSB of the pixel of the cover image. It is the most used method for steganography due to its simplicity. There are various improvisations made in this method to provide more level of security, such as the introduction of a secret key [7], cryptography [8] and so on.

2.2 Illustration

The following example illustrates the working of the LSB algorithm

Consider a 24-bit cover image and hidden image,

The spatial binary pixmap of the cover image

Table 1

R	G	B
10010110	10101001	10101101
10101110	11010101	11010101
01010111	10110101	10111011
11110001	10111101	10101101
10101011	10101001	10100000
11111110	11101010	10101010
10010011	10101010	10001010
10110110	10101010	10100011

Consider a pixel of the hidden image which is to be embedded in the above pixmap

Table 2

R	G	B
10101101	11011010	10101100

The changed pixmap of the cover image after embedding the values

Table 3

R	G	B
1001011**1**	1010100**1**	1010110**1**
1010111**0**	1101010**1**	1101010**0**
0101011**1**	1011010**0**	1011011**1**
1111000**0**	1011110**1**	1010110**0**
1010101**1**	1010100**1**	1010000**1**
1111111**1**	1110101**0**	1010101**1**
1001001**0**	1010101**1**	1000101**0**
1011011**1**	1010101**0**	1010001**0**

3. THE PROPOSED METHOD

The LSB method is based on the idea of bit by bit replacement whereas in our method we have used weighted mean to replace the entire pixel in the cover image and camouflaged it by average values. The idea behind the technique is that a human eye is not capable to detect minor variations of the colors and this capability is reduced even more if high resolution images are used.

3.1 Encoding Process

The choice of cover image in this case is every crucial, a high contrast between the cover and the hidden image should be avoided. Another important factor to consider here is the ratio of the number of pixels between the cover and the hidden image. The ratio (R) of number of pixels in the cover image to that of the hidden image should be at least 9:1.

The first step starts with selecting a pixel from the cover image which is to be replaced with the hidden image pixel. Then average of the neighboring eight pixels is calculated, the following illustration would make it clear.

P_1	P_2	P_3
P_4	P_x	P_5
P_6	P_7	P_8

Consider a part of pixmap of the cover image where,
P_x is the pixel which is to be replaced and RGB values be (R_x, G_x, B_x)
P_i be the neighboring pixel where $i \varepsilon [1,8]$

The average be denoted by $P_{av=}$ (R_{av}, G_{av}, B_{av})

$$R_{av} = \sum(R_i)/8$$

$$G_{av} = \sum(G_i)/8$$

$$B_{av} = \sum(B_i)/8$$

After the values of P_{av} are calculated the next step is to take the weighted average with the pixel of hidden image.

Consider $P_h$as the pixel of the hidden image and P_s the final value of the pixel of the stegoimage.

Then

$$R_s = \alpha R_{av} + (1-\alpha) R_h$$

$$G_s = \alpha G_{av} + (1-\alpha) G_h$$

$$B_s = \alpha B_{av} + (1-\alpha) B_h$$

$$\forall \alpha \varepsilon (0,1)$$

Thus P_s= (Rs, Gs, Bs)

Repeat the entire process to replace the P_{x+R} with P_s for every pixel P_h in the hidden image.

3.2 Decoding Process

The stegoimage is the input of the decoding process, the hidden image is retrieved from the stegoimage by the following process.

The first step of decoding is similar to the process of encoding, the average of the eight pixels surrounding P_s calculated to find P_{av}

P_1	P_2	P_3
P_4	P_s	P_5
P_6	P_7	P_8

Therefore, $P_{av} = (R_{av}, G_{av}, B_{av})$

Similarly, the values of $P_h = (R_h, G_h, B_h)$

$$R_h = (R_s - \alpha R_{av})/(1-\alpha)$$

$$G_h = (G_s - \alpha G_{av})/(1-\alpha)$$

$$B_h = (B_s - \alpha B_{av})/(1-\alpha)$$

Repeat this process for every P_s in the stegoimage until every pixels P_h are completely retrieved.

Analysis on the basis of decryption by brute force method

LSB Method

In the LSB method in order to hide the image well only the last 4 bits of the LSB can be used. As using higher bits would cause a major deviation in colors and would not be well hidden. So it only leaves 4 bits of LSB in the consideration for hiding, the maximum deviation that can be caused by it would be 15 in the RGB of pixel. That is 15 in R, G and B each, accounting for a 5.8% deviation in individual colors and 17.4% overall.

The maximum number of attempts that would be required to retrieve the extract hidden image (by brute force) would be

$$2^4 * 2^4 * 2^4 = 2^{12}$$

For one single pixel of the cover image.

If more than 4 bits are used say 5, then the maximum deviation from the original color would be 12.5% in each and 37.64% overall. Such a huge deviation would result in a very significant difference between the colors of stegoimage and cover image, hence more than 4 bits cannot be used for encoding.

Proposed Method

In the proposed method rather than replacing the bits we replace the entire pixel of the cover image by hidden image and camouflage it by taking average values.

The entire process of hiding the replacement pixel in the cover image is based on the value α. The value of α can be selected so as to increase the complexity of the entire decryption process (by brute force). The LSB method cannot use more than 4 bits of LSB whereas in this case the value of α can be used up to 64 bit floating precision (19 decimal places).

Complexity comparison with respect to decryption by brute force LSB

Table 4

Number of LSB bits used	Complexity of Decryption	Overall deviation (%)
1	2^3	1.17
2	2^6	4.70
3	2^9	8.23
4	2^{12}	17.64

Proposed Method

Table 5

Precision of α (decimal places)	Complexity of Decryption	Overall deviation (%)
1	$2^{3.16}$	30
4	$2^{13.13}$	0.03
8	$2^{26.42}$	0.000003
12	$2^{39.71}$	0.0000000003

1) The calculations in Table 5 are made as follows:
For 1 decimal place of α the value of α is taken as maximum possible i.e. 0.9, successively the values of α are taken as 0.9999, 0.99999999, 0.999999999999. The deviation for one color is thus calculated as

$$D = ((1-\alpha) * 255/255) * 100$$

Hence for all 3 colors total deviation would be 3*D.

The tables show that on increasing the number of bits for hiding the image by LSB method results in significantly high deviation of the color values from the actual values.

On the contrary in the proposed method the deviation becomes negligible with increase in the number of decimal places (precision) of α.

2) Complexity of Decryption is the total number combinations possible of the bits or digits in case of binary or decimal respectively.

For example, consider the following,

The values of α (4-digit precision) would be from 0001 to 9999, which comes out to be $2^{13.13}$ number of values.

By observing the tables, it is evident that on increasing the number of bits in the LSB for encryption leads to an increase in deviation from original values of colors. On the other hand, by increasing the precision of α the deviation is substantially reduced.

Peak Signal to Noise Ratio (PSNR)

For the Proposed method

In order to compare the two methods discussed above in an objective way PSNR is the most commonly used method. In calculating PSNR we need the Mean Square Error (MSE) which is given by

$$MSE = 1/(m*n) * \sum\sum (I_{ij} - K_{ij})^2$$

Now we will calculate the maximum possible mean square error.

The term I_{ij}-K_{ij} is the difference between the corresponding pixels in cover image and stegoimage. To calculate maximum MSE, we assume that we will get the maximum possible difference D for each pixel hidden in the stegoimage. We have concealed a total of m'*n' pixels (size of hidden image) in our stegoimage. Hence we get,

$$MSE = m'*n'/(m*n) * [D]$$

Now while calculating D we observe that it will be a sum of two entities

$$D = d^2+x*9^2$$

Where,
d is the max deviation from the original value given by,

$$d = \alpha*(0) + (1-\alpha) *(255)$$
$$= (1-\alpha)*255$$

The value of 'd' turns out to be so because the RGB values range from 0 to 255 and while calculating the weighted average, if the cover image pixel value is 0 and the hidden image pixel value is 255 then the stegoimage pixel value will contain information only from the hidden image (though it will be reduced to some extent due to multiplying factor (1-α)).

The term $x*9^2$ stemsfrom the following observation:

Consider the following example:

Let α = 0.9789, R_{av} = 124, R_s=225
Then,

$$R_s = \alpha R_{av} + (1-\alpha) R_h$$

$$R_s = 0.9789*124 + 0.0211*225$$

$$R_s = 126.1311$$

Now to recover the R_s correctly we will need to somehow save the 4 digits after the decimal point i.e. 1,3,1,1 since in our image we can only store the part of the number before the decimal point. Not saving these digits will lead to significant errors while recovering R_h. In the above example if we do not store these digits the recovered R_h will be as follows:

$$R_h = (126 - 0.9789*124) / 0.0211$$

$$R_h = 218.78$$

which is deviating from our original R_h = 225

If we save the 4 digits then the recovered R_h will be:

$$R_h = (126.1311 - 0.9789*124) / 0.0211$$

$$R_h = 225$$

which is what we wanted.

Note that we have saved only 4 digits because α = 0.9789 was of 4-digit precision. Generalizing this argument for an α with x digit precision, to retrieve the hidden image we must somehow save at least x digits after decimal point after calculating R_s, G_s, B_s.

These x digits can be saved either externally (as a key) or embedded in the image itself. Note that these x digits obtained after calculating the pixel values (R_s, B_s, G_s) of stegoimage may be different for each pixel which is to be hidden.

If we choose to save the digits in the image itself, we the maximum possible difference between cover image and stegoimage will be 9. Hence assuming the worst case scenario, for each pixel to be hidden we will be modifying x pixels in the stegoimage, introducing an error of $x*9^2$ (per hidden pixel) in the MSE.

Thus, the maximum possible MSE is given by

$$MSE= m'*n'/(m*n) *[d^2 + x*9^2]$$

Where,
d is the max deviation from the original value
x is the precision of α
m*n are the dimensions of the cover image
m'*n' are the dimensions of the hidden image

Now, let R = m'*n'/(m*n)

$$MSE = R*[d^2 + x*9^2]$$

PSNR is given as:

$$PSNR = 10\log [(255)^2/MSE]$$

Therefore, in our case,

$$PSNR = 10\log [(255)^2/ R*[d^2 + x*9^2]]$$

The plots for PSNR vs ratio R for different values of x are given below. Since R is the ratio between size of hidden image and size of cover image, R will be in the range (0,1). Although the graph has been shown for full range of R i.e. from 0 to 1, note that our method has placed restrictions on R. R cannot exceed 1/9 since for each pixel in hidden image we are using 9 pixels in the cover image.

For x=1 and d=25

PSNR = 19.6-10*log(R)

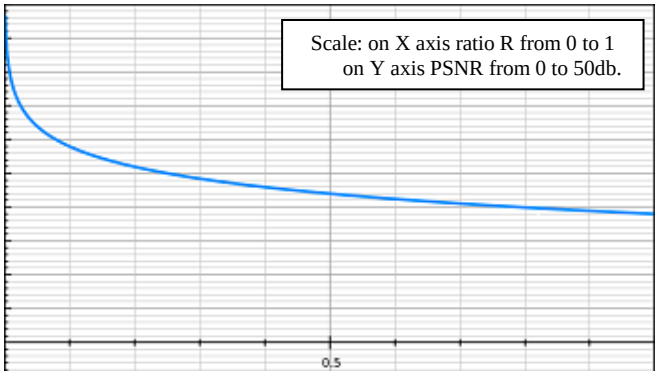

Fig 1

For x=3 and d=0

PSNR = 29.04-10*log(R)

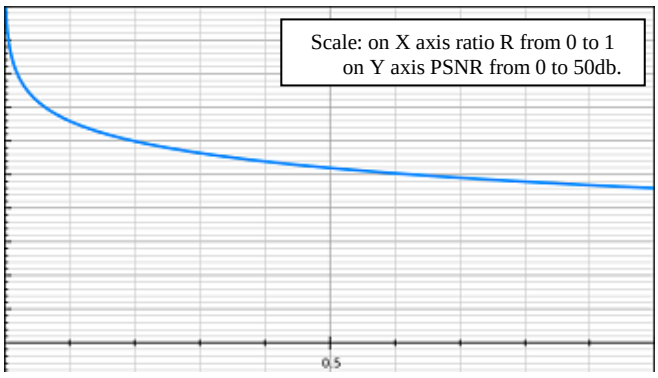

Fig 2

By Fig 1 and Fig 2 the PSNR value comes out to be around 35db for R=0.1, which is still higher than minimum required 28db [8].

For LSB method

PSNR values for LSB method are [4] –

Table 6

Number of LSBs used	PSNR Values (db.)
1	57.16
2	44.64
4	27.61

The above values are calculated considering the worst case possible deviation from the original colors for 24-bit True color images.

Sample Images

The cover image here is a 1920*1080p 24-bit True color image, and the hidden image is 400*400p 24-bit True color.

Fig 3 Cover Image

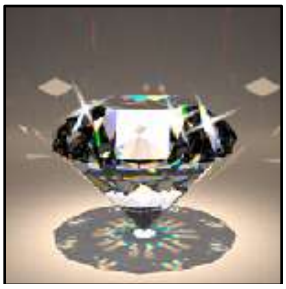

Fig 4 Hide Image

Fig 5 Stegoimage

Fig 6 Recovered Image

4. CONCLUSION

The method demonstrated in this paper is an alternative approach for image steganography. The main advantage of this method is that it works with both binary as well as decimal number system. In terms of complexity of decryption, the proposed method fares better than the LSB method by providing better PSNR for similar complexity. In the LSB method if multiple LSB's are used for encryption the stegoimage changes significantly from the cover image. On the contrary in the proposed method on increasing the precision of α the deviation between the cover image and the stegoimage is reduced significantly. This method is also capable of embedding high quality 24–bit True color image with lossless retrieval. Thus by observing the results obtained by this demonstration it is clear that the proposed method can be implemented as and when required for better security and image quality instead of LSB.

REFERENCES

[1] Stefan Katzenbeisser, Fabien A.P. Petitcolas (a), the new Charles Ng, NIU Xin-xin, etc.(translation) information hiding – steganography and digital watermarking [M] Bejing: People's Posts and Telecommunications Press, 2001,50-62.

[2] Steganography and Digital Watermarking, Jonathan Cummins, Patrick Diskin, Samuel Lau and Robert Parlett, School of Computer Science, The University of Birmingham.

[3] Cryptography and Network Security Principles and Practices, Fourth Edition by William Stallings, Publisher: Prentice Hall Pub Date: November 16, 2005 Print ISBN-10: 0-13-187316-4, pg. no 260.

[4] Huang Jian RSA safety analysis and improvement of public key encryption system [J] computer and network, 2016,01: 70-73.

[5] Li Li. Study [D] based on the LSB information hiding technology, Beijing University of Posts and Telecommunications, 2011.

[6] Yan Xiaomeng Zhang Tao, Xi Ling, PING Xi build one for LSB matching steganography load new positioning algorithm [J] data acquisition and processing, 2016, 01: 145-151.

[7] S. M. Masud Karim, M. S. Rahman and M. I. Hossain, "A new approach for LSB based image steganography using secret key," Computer and Information Technology (ICCIT), 2011 14th International Conference.

[8] Xinyi Zhou, Wei Gong, WenLong Fu, LianJing Jin978-1-5090-0806-3/16/$31.00 copyright 2016 IEEE ICIS 2016, June 26-29, 2016, Okayama, Japan.

[9] Comparative Analysis of Steganography for Coloured Images Shrutika Suri, Himani Joshi, Vishakha Mincoha and Akash Tyagi, International Journal of Computer Sciences and Engineering Vol.-2(4), pp (180-184) April 2014, E-ISSN: 2347-2693.

[10] Jiaohua Qin, Xiaoyu Guo, XuyuXiang Lingyun Xiang and LiliPan, Steganalysis Based on Least Square Method for Multiple Least Significant Bits Steganography, Information Technolgy Journal, 2013 ISSN 1812-5638/DOI: 10.3923/itj.2013.

[11] Pooja Kaushik and Yuvraj Sharma, "Comparison Of Different Image Enhancement Techniques Based Upon PSNR & MSE", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11, 2012.

[12] Gurpreet Kaur and Kamaljeet Kaur, "Digital Watermarking and Other Data Hiding Techniques", International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-2, Issue-5, April 2013.

[13] J. Fridrich and M. Goljan, "On estimation of secret message length in LSB steganography in spatial domain," in Security, Steganography, and Watermarking of Multimedia Contents VI, E. J. Delp III and P. W. Wong, eds., Proc. SPIE 5306, pp. 23–34, 2004.

[14] A Review of Approaches for SteganographyKomal Arora and Geetanjali Gandhi, B.S.Anangpuria Institute of Technology and Management, International Journal of Computer Sciences and Engineering Vol.-2(5), PP(118-122) May 2014, E-ISSN: 2347-2693.