A Comparative Analysis of Image Steganography

R.Amirtharajan
Assistant Professor
ECE/SEEE
SASTRA University

R. Akila
ECE / SEEE
SASTRA University
Tamil nadu

P.Deepikachowdavarapu
ECE / SEEE
SASTRA University
Tamil nadu

ABSTRACT
Digital communication has become an essential part of infrastructure now-a-days, a lot of applications are Internet-based and in some cases it is desired that the communication be made secret. Two techniques are available to achieve this goal: cryptography and steganography. In this paper, various digital steganographic techniques are implemented which are capable of producing a secret-embedded image that is indistinguishable from the original image to the human eye. A comparative analysis is made to demonstrate the effectiveness of the proposed methods. The effectiveness of the proposed methods has been estimated by computing Mean square error (MSE) and Peak Signal to Noise Ratio (PSNR).

Categories and Subject Descriptors
D.2.11 Information hiding
D.4.6 Security and Protection

General Terms
Security

Keywords
LSB Steganography, Information hiding, Inverted Pattern Approach, Pixel value differencing, Steganography.

1. INTRODUCTION
Internet has becomes essential the most effective and fastest media for communication. Albeit, it is susceptible to face many problems such copyright, hacking, eavesdropping etc. Hence the need for secret communication is required. Cryptography [4, 6, 11] and Steganography [1-3, 5, 7 - 19] are the two fields available for data security. Cryptography is a technique in which the data is scrambled in an unintelligent gibberish fashion so that it becomes difficult for any malicious user to extract the original message. Only the desired recipient will be having the code for decryption and will be able to extract messages. Cryptography has helped a great deal in data security but it has some disadvantages. The encrypted data will arouse suspicion to malicious users and there is a possibility of it being decrypted or being suppressed. Hence the intended information might not reach its destination effectively. The disadvantages of Cryptography have lead to the development of Steganography.

In recent years, enormous research efforts have been invested in the development of digital image steganographic techniques[]. The major goal of steganography is to enhance communication security by inserting secret message into the digital image, modifying the nonessential pixels of the image [ ]. The image after the embedding of the secret message, so-called stego-image, is then sent to the receiver through a public channel. In the transmission process, the public channel may be intentionally monitored by some opponent who tries to prevent the message from being successfully sent and received. The opponent may randomly attack the stego-image if he/she doubts the stego-image carries any secret message because the appearance of the stego-image shows obvious artifacts of hiding effect. For this reason, an ideal steganography scheme, to keep the stego-image from drawing attention from the opponent, should maintain an imperceptible stego-image quality. That is to say, if there are more similarities between the cover image and the stego-image, it will be harder for an attacker to find out that the stego-image has important secret data hidden inside it . This way, the secret data is more likely to travel from the sender to the receiver safe and sound. For the past decade, many steganographic techniques for still images have been presented [1-3, 5, 7 - 19] both in spatial and frequency domains. A simple and well known approach is directly hiding secret data into the least-significant bit (LSB) of each pixel in an image.

2. Review on literature
2.1 LSB Substitution Method [5]
The most well-known steganographic technique in the data hiding field is least-significant-bits (LSBs) substitution. This method embeds the fixed-length secret bits in the same fixed-length LSBs of pixels. Although this technique is simple, it generally causes noticeable distortion when the number of embedded bits for each pixel exceeds three. Several adaptive methods for steganography have been proposed to reduce the distortion caused by LSBs substitution. For example, adaptive methods vary the number of embedded bits in each pixel, and they possess better image quality than other methods using only simple LSBs substitution. However, this is achieved at the cost of a reduction in the embedding capacity.

2.2 Optimum Pixel Adjustment Procedure [5]
The proposed Optimal Pixel adjustment Procedure (OPAP) reduces the distortion caused by the LSB substitution method. In OPAP method the pixel value is adjusted after the hiding of the secret data is done to improve the quality of the stego image without disturbing the data hidden.

2.2.1 Procedure for hiding:
- First a few least significant bits are substituted with the data to be hidden.
- Then in the pixel, the bits before the hidden bits are adjusted suitably if necessary to give less error.
- Let n LSBs be substituted in each pixel.
- Let d= decimal value of the pixel after the substitution.
- d1 = decimal value of last n bits of the pixel.
- d2 = decimal value of n bits hidden in that pixel.

- If(d1~d2)<=(2^n)/2
  then no adjustment is made in
  that pixel.
- Else
  If(d1<d2)
  d = d – 2^n .
  If(d1>d2)
  d = d + 2^n .

This d is converted to binary and written back to pixel.

2.2.2 Retrieval:

The retrieval follows the extraction of the least significant bits(LSB) as hiding is done using simple LSB substitution.

2.2.3 Advantages:

1. Simple methodology.
2. Easy retrieval.
3. Improved stego-image quality than LSB substitution.

2.3 Inverted Pattern Approach (IP)[17]

This inverted pattern (IP) LSB substitution approach uses the idea of processing secret messages prior to embedding. In this method each section of secret images is determined to be inverted or not inverted before it is embedded. In addition, the bits which are used to record the transformation are treated as secret keys or extra data to be re-embedded.

2.3.1 The embedding procedure is:

- The embedded string is ***S***, the replaced string is ***R,*** and the embedded bit string to divided to ***P*** parts.
- Let us consider n-bit LSB substitution to be made. Then S and R are of n-bits length.
- For P part in i = 1 to P
  If MSE(Si,Ri) ≤ MSE(S'i,Ri)
  Choose Si for embedding
  Mark key(i) as logic '0'
  If MSE(Si,Ri) ≥ MSE(S'i,Ri)
  Choose S' i for embedding
  Mark key(i) as logic '1'
  MSE – Mean Square Error.
- End

where,

S is the data to be hidden

S' is the data to be hidden in inverted form.

2.3.2 Procedure for retrieval is:

The stego-image and the key file are required at the retrieval side.

- First corresponding numbers of LSB bits are retrieved from the stego-image.
- If the key is '0', then the retrieved bits are kept as such.
- Else if the key is '1', then the bits are inverted.
- The bits retrieved in this manner from every pixel of the stego-image gives the data hidden.

2.4 **IP Method Using Relative Entropy [17]**

Relative entropy [7] measures the information discrepancy between two different sources with an optimal threshold obtained by minimizing relative entropy. In this method instead of finding the mean square error for inverted pattern approach, the relative entropy is calculated to decide whether S or S' suites the pixel. In probability theory and information theory, the Kullback–Leibler divergence (also information divergence, information gain, or relative entropy) is a non-symmetric measure of the difference between two probability distributions *P* and *Q*.

It is given by,

$$D(p \parallel q) = \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)} = E_p \log \frac{p(X)}{q(X)}$$

2.4.1 ***Embedding procedure is:***

- Divide the cover image into ***P*** blocks of same size, the embedding string is ***S***, and the replaced string is ***R.***
- For P part in i =1 to P
  If rel.entropy(Si,Ri) ≤ rel.entropy (S'i,Ri)
  Choose Si for embedding
  Mark key(i) as logic '0'
  If rel.entropy (Si,Ri) ≥ rel.entropy (S'i,Ri)
  Choose S'i for embedding
  Mark key(i) as logic '1'

End

where,

S is the data to be hidden

S' is the data to be hidden in inverted form.

2.4.2 Procedure for retrieval is:

The stego-image and the key file are required at the retrieval side.

- First corresponding numbers of LSB bits are retrieved from the stego-image.
- If the key is '0', then the retrieved bits are kept as such.
- Else if the key is '1', then the bits are inverted.
- The bits retrieved in this manner from every pixel of the stego-image gives the data hidden

2.5 The proposed Hiding Streams of 1s and 0s

The usual steganographic methods fetch few bits from the secret data to be embedded. But this method fetches the 1s or 0s present consecutively for hiding. This is an innovative steganographic method where the data to be hidden is converted to binary. The number of 1s and 0s are counted and stored in the pixels of the cover image in this method. The number of 1s is stored in the odd columns of the pixel and the number of 0s is stored in the even columns.

2.5.1 The steps for hiding are:

- Let the data to be hidden be in y.

- Find the consecutive number of 1's present in y until a '0' is encountered.
- Hide the cnt in the pixel using 'mod10' method.
- Let w(i,j) be the gray value of the pixel,
- d=mod(w(i,j),10)~cnt, d1=mod(w(i,j),10)~(10-cnt).
- if(d<d1)
  w(i,j)=w(i,j)-mod(w(i,j),10)+cnt
  key='0'.
- else
  w(i,j)=w(i,j)-mod(w(i,j),10)+(10-cnt),
  key='1'.
- If the cnt>10, then hide it in the pixel by 'mod100' method, and instead of '0' and '1', use '!' and ' '.
- The above steps are repeated for stream of 0s.

Steps for retrieval:

- Let the pixel values of stego image are in a variable named w.
- We move along the image row wise.
- The 1s are obtained from odd columns and 0s are obtained from even columns.
- Mod 10 operation is performed in each pixel. This gives the number of bits hidden in that pixel.
- Grouping the data from all the pixels in a similar way gives the data hidden.

Advantages

- Simple to implement.
- More embedding capacity.
- Smaller key size.

2.6 **Pixel Value Differencing (PVD) [12, 15, 18]**

Pixel Value Differencing is able to provide a high quality stego image in spite of the high capacity of the concealed information. That is, the number of insertion bits is dependent on whether the pixel is an edge area or smooth area. In edge area the difference between the adjacent pixels is more, whereas in smooth area it is less. While human perception is less sensitive to subtle changes in edge areas of a pixel, it is more sensitive to changes in the smooth areas. Two techniques of PVD are explained here.

2.6.1 PVD method [18]

This method hides the data in the target pixel by finding the characteristics of four pixels surrounding it, indicated in the table below:

g(x-1,y-1) top left pixel	g(x-1,y) top pixel	g(x-1,y+1) top right pixel
g(x,y-1) left pixel	g(x,y) target pixel	

g(x-1,y-1) , g(x-1,y) , g(x-1,y+1) , g(x,y-1) are the gray values of the pixels surrounding the target pixel g(x,y).

2.6.1.1 Embedding procedure:

- Select the maximum and the minimum values among the four pixel values that have already finished the embedding process. Calculate the difference value *d* between the maximum pixel value and the minimum pixel value using the following formula

" d = gmax − gmin "

where,

g_{max} = max(g(x−1,y−1), g(x−1,y) ,g(x−1,y+1), g(x,y−1)) and

g_{min} = min(g(x−1,y−1), g(x−1,y) ,g(x−1,y+1), g(x,y−1))

- Using above equations,we judge whether the target pixel is included in an edge area or a smooth area.the number of bit *n* , inserted into the target pixel is determined by value d.
- Calculate $n = \log_2 d - 1$, if d > 3.
  = 1 otherwise.
- Calculate a temporary value $t_{x,y} = b - (g_{x,y} \bmod 2^n)$
  where b is the data to be hidden.
- Calculate t1
  =t(x,y)
  if $(-(2^n-1)/2) \le t(x,y) \le (2^n-1)/2^n$
  $=t(x,y)+2^n$
  if $(-2^n+1) \le t(x,y) < (-2^n-1)/2^n$
  $=t(x,y)-2^n$
  if $(2^n-1)/2 < t(x,y) < 2^n$
- g1(x,y)=g(x,y)+t1(x,y).
  g1(x,y) is the new pixel value.

2.6.1.2Retrieval:

- n is calculated in the same way as in the sender side.
- The target pixel value is present in g(x,y).
- The data hidden is $b=g1(x,y) \bmod 2^n$.

2.6.2 The proposed PVD method

This method also uses the concept of hiding the data using the difference between the pixel values. Unlike the previous method, this method hides the data in the difference between two adjacent pixel values. This method gives the stego images of better quality than the traditional method while maintaining a high embedding capacity.

2.6.2.1 ***Steps for embedding:***

1. Read the cover image and save the pixel values in a variable, say 'w'.

2. Let the data to be hidden be in binary format in another variable, say 'y'.

3. Leave the first row and the first column of the image.

4. Let the pixel in which we are going to hide the data be the current pixel.

5. d1= difference(in binary) between the current pixel and its left pixel.

6. d2= difference(in binary) between the current pixel and its top pixel.

7. Also let l1 and l2 be the lengths of d1 and d2.

8. Extract the data from y of lengths l1-1, l1, l1+1, l2-1, l2, l2+1 and save them in the same order in dif(i), where i=1 to 6.

9. Now find if the d1 or d2 is nearer to any one of the dif(i).

Whichever dif(i) that is nearer, let us name it as min.

10. if (min nearer to d1)&&((min~d1)< 8)

(If the difference is more than 8, considerable distortion would take place in the image which is avoided.)

Then adjust the current pixel value so that the difference between the current pixel and the left pixel is the data that is fetched from y, which is the data to be hidden.

Save the key as '0'.

11. if (min nearer to d2)&&((min~d2)< 8)

Then adjust the current pixel value so that the difference between the current pixel and the top pixel is the data that is fetched from y, which is the data to be hidden.

Save the key as '1'.

12. if (min~d1)>8 && (min~d2)>8

then skip the current pixel without embedding.

Save the key as ' '.

13. The number of bits embedded in the pixels which is immediately below the current pixel as follows:

If (the number of bits hidden<=10)

then use mod10* method to hide the number of bits in the pixel below.

Else if(the number of bits hidden>10)

then use the mod100* method to hide the number of bits in the pixel below.

Our system refers to four pixels adjacent to a target pixel in the embedding process:

14. Repeat the above steps until all the bits in the y are hidden successfully.

15. Then with the help of the

	upper pixel
left pixel	current pixel
	lower pixel

stego image and the key file the data hidden can be extracted.

2.6.2.2 Steps for the retrieval of the data

1. Read the stego image and save the gray values of the pixels in a variable, say 'w'.

2. Read the key file and save the binary data in a variable say 'y'.

3. Skip the first row and first column.

4. if (y = ='0')

Then find the difference between the current pixel and the left pixel and save it in 'd'.

Also find the number of bits stored by mod* method from the below pixel and save it in 'n'.

Convert 'd' into binary of 'n' bits.

5. Else if (y = ='1')

Then find the difference between the current pixel and the top pixel and save it in 'd'.

Also find the number of bits stored by mod method from the below pixel and save it in 'n'.

Convert 'd' into binary of 'n' bits.

6. Else if(d = =' ')

Skip the current pixel.

2.7 The proposed mod method:

In this method embedding is done by subtracting any remainder obtained by dividing with 10 and adding the data to be hidden. This is demonstrated by let the pixel be 'p' and data be less than 10 say 'd' then new pixel formed is

p1=p- remainder (p/10) +d.

Similarly mod100 method divides the pixel by 100, remainder obtained is subtracted and the data is added to it to get the new pixel. The data hidden will simply be equal to the remainder obtained by dividing the new pixel by 10 or 100 accordingly.

Advantages:

1. This method has high embedding capacity.

2. It retains the quality of the image.

3. This is a method where the data is hidden in the difference between the adjacent pixels, so that mere extraction of few lsb bits will never give the data hidden.

4. Also there is the key file without which extraction of data becomes impossible.

2.8 The proposed MOD10 based method

This method hides the data in the remainder obtained by dividing the gray value of the pixel by 10. This method has a key which determines whether the data is same as the remainder or 10-remainder. The key improves the quality of the stego image.

2.8.1 Procedure for hiding:

- Let the data to be hidden is in the variable y in binary.
- Fetch 3 bits at a time from y and convert into decimal and store it in a variable, say x.
- Let w(i,j) be the gray value of the pixel.
- d1= mod(w(i,j),10)~x ,

d2= mod(w(i,j),10)~(10-x).

- If(d1<d2)

w(i,j) = w(i,j)-mod(w(i,j),10)+x,

key='0'.

else

w(i,j)= w(i,j)- mod(w(i,j),10)+(10-x),

key='1'.

2.8.2 Retrieval:

If the key of a pixel is '0', the data

d= mod(w(i,j),10) otherwise

d=10- mod(w(i,j),10).

now d is converted into binary data of width 3bits.

2.9 DCT

In this method, a transform domain technique, DCT is used to hide messages in significant areas of the cover image. Here pixels are split into 8×8 blocks. Then, all blocks are DCT transformed each block encodes exactly one secret message bit.

2.9.1 ***Procedure for hiding:***

- The embedding process starts with selecting a block bi which will be used to code the 'i'th message bit.
- Let Bi = D{bi} be the DCT-transformed image block.
- Before the communication starts, both sender and receiver have to agree on the location of two DCT coefficients, which will be used in the embedding process. Let us denote these two indices by (u1, v1) and (u2, v2).
- Let m(i) be the 'i'th message bit.
- If m(i)=0 ,
- if Bi(u1, v1) > Bi(u2, v2) then
- swap Bi(u1, v1) and Bi(u2, v2).
- else if m(i)=1,
- if Bi(u1, v1) < Bi(u2, v2) then
- swap Bi(u1, v1) and Bi(u2, v2).
- The last step is to take inverse dct of the blocks to obtain the stego image.
- During the retrieval, again the stego image is split as 8X8 pixel blocks and are dct transformed.
- Now, the predetermined set of two DCT coefficients are compared for all the blocks.
- if Bi(u1, v1) > Bi(u2, v2) then the message bit=1,
- else 0.

2.9.2 Procedure for Retrieval:

- A block bi is selected in the stego-image.
- Then the dct is performed on the block,Bi=D{bi}.
- Then the two indices (u1,v1) and (u2,v2) which are chosen by both sender receiver are compared.
- If Bi(u1, v1) > Bi(u2, v2)
  Then data hidden='1'
- Else if Bi(u1, v1) > Bi(u2, v2)
  Then data hidden='0'
- This procedure is repeated for all the blocks in the image.

2.9.3***Advantages:***

- This method is more robust to attacks, such as compression, cropping etc.
- Though the embedding capacity is low, the quality of image is good.

3. Result and discussion

In this present implementation Lena and baboon 256 × 256 digital Lena and Baboon images has been taken as cover images and the results are presented in figures 1 and 2. The estimating parameters of the two stego covers have been performed using indigenous matlab code in Intel Core2 Duo CPU processor @ 1.60 GHz, 1GB RAM. Our proposed methodology has been compared with available methods and the results are tabulated in Table 1.

Table 1 Comparative performance of all the methods

Method	Cover Image	size of secret data	MSE	PSNR	Time (s)
OPAP	Lena	25KB	0.1503	56.71	7.56
	Baboon	25.8KB	0.1620	55.96	7.66
IP	Lena	24.5KB	0.2635	51.86	7.96
	Baboon	24.5KB	0.2039	52.06	7.98
Stream of 1s and 0s	Lena	21KB	3.9013	42.21	8.776
	Baboon	21KB	3.8091	42.33	8.456
PVD 1	Lena	22KB	12.4715	37.87	8.6
	Baboon	25KB	21.456	35.40	8.91
PVD 2	Lena	10.2KB	8.4798	38.84	8.156
	Baboon	12KB	7.4712	39.57	8.047
DCT	Lena	100B	0.0255	64.07	7.46
	Baboon	100B	0.0244	65.15	7.38
Mod 10	Lena	18.1KB	1.63	50.23	8.63
	Baboon	18.7KB	1.58	51.61	8.52

4. Conclusion

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. It is therefore a book on magic. It is emerging in its peak because it does not attract anyone by itself. In this paper a comparative analysis of several methods has been successfully implemented and results are delivered. The MSE and PSNR of all the methods are also compared and also this paper presented a background discussion and implementation on the major algorithms of steganography deployed in digital imaging. The emerging techniques such as LSB based, OPAP, Inverted pattern based LSB using MSE, Inverted pattern based LSB using Relative entropy,

String of 1 and 0 based, mod based and Mod 10 Generally few of these methods tend to have a lower payload compared to spatial domain algorithms. There are different ways to reduce the bits needed to encode a hidden message

5. Acknowledgement

The author's wants to heartily thank their supervisor, Dr.R. John Bosco Balaguru, Professor and Dr. K.Thenmozhi Professor ECE / SEEE, SASTRA University, for their critique discussion

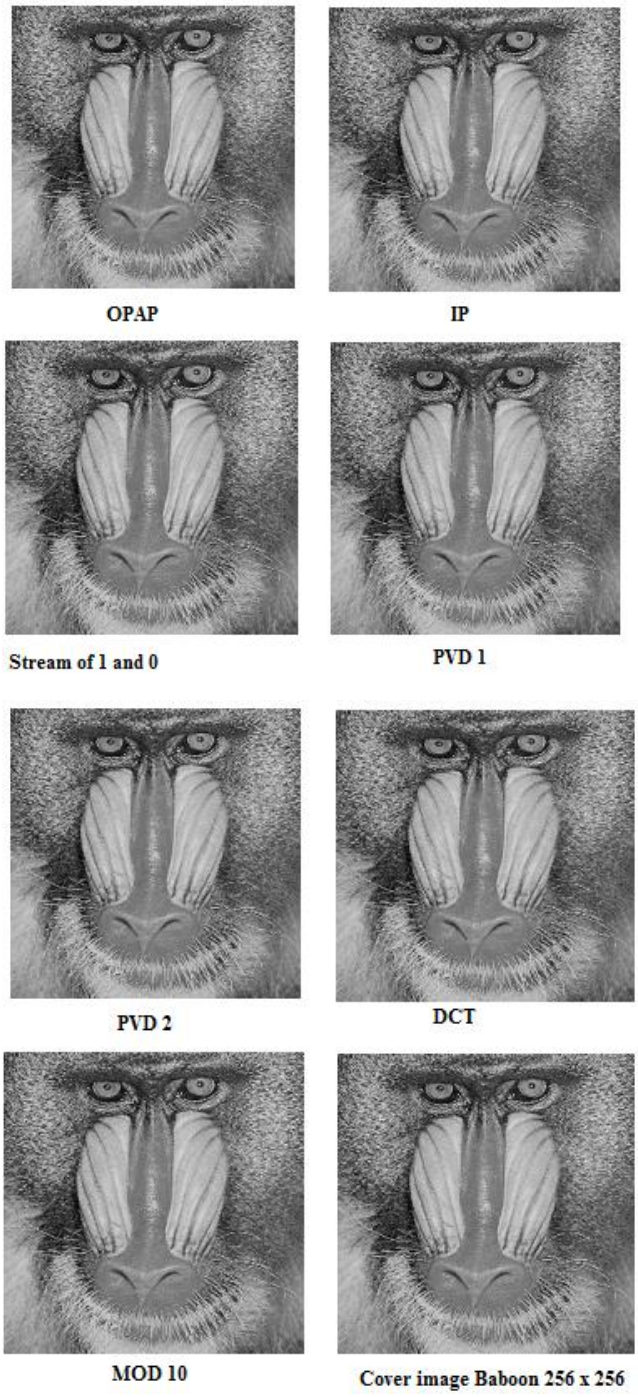

Figure 1 & 2. Lena and Baboon 256 × 256 and its outputs

References

[1]. Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt, Digital image steganography: Survey and analysis of current methods Signal Processing 90 (2010) 727–752

[2]. R.Amirtharajan , R. John Bosco Balaguru, Constructive Role of SFC & RGB Fusion versus Destructive Intrusion International Journal of Computer Applications. **Volume 1 – No. 20.** ISSN : 0**975 – 8887 pp 34-40**

[3]. W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Syst. J. 35 (3&4) (1996) 313–336.

[4]. Bruice Schneier, Applied Cryptography Protocols, Algorithm and Source Code in C. Second edition. Wiley India edition 2007

[5]. C.K. Chan, L.M. Chen, Hiding data in images by simple LSB substitution, Pattern Recognition 37 (3) (2004) 469–474.

[6]. W. Diffie and M. E. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," IEEE Computer, Vol. 10, 1977, pp. 74-84.

[7]. S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.

[8]. L.M. Marvel, C.G. Boncelet Jr., C.T. Retter, Spread spectrum image steganography, IEEE Trans. Image Process. 8 (8) (1999) 1075-1083.

[9]. F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proc. IEEE 87 (7) (1999) 1062–1078.

[10]. Provos, N., Honeyman, P, Hide and seek: An introduction to steganography, IEEE Security & Privacy Magazine 1 (2003) 32-44.

[11]. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, (1978) 120–126.

[12]. C.C. Thien, J.C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recognition 36 (11) (2003) 2875–2881.

[13]. Tuomas Aura, Practical invisibility in digital communication, in proceedings of the Workshop on Information Hiding, LNCS 1174 (1996) 265-278.

[14]. R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34 (3) (2000) 671–683.

[15]. C.M. Wang, N.I. Wu, C.S. Tsai, M.S. Hwang, A high quality steganography method with pixel-value differencing and modulus function, J. Syst. Software 81 (1) (2008) 150–158.

[16]. Westfeld Space filling curves in steganalysis in E.J Delp III & P.W. Wong(Eds), Security, steganography and watermarking of multimedia contents VII SPIE 5681, (2005) 28-37

[17]. C.H. Yang, Inverted pattern approach to improve image quality of information hiding by LSB substitution **Pattern Recognition** 41 (2008) 2674–2683

[18]. Young-Ran Park, Hyun-Ho Kang, Sang-Uk Shin, and Ki-Ryong Kwon, An Image Steganography Using Pixel Characteristics Y. Hao et al. (Eds.): CIS 2005, Part II, Springer-Verlag Berlin Heidelberg LNAI 3802, (2005) 581– 588.

[19]. Yuan-Hui Yu , Chin-Chen Chang, Iuon-Chang Lin, A new steganographic method for color and grayscale image hiding Computer Vision and Image Understanding 107 (2007) 183–194