Message Guided Adaptive Random Steganography using Counting-out Embedding

Ravuru Rakesh
Dept. of ECE,
SASTRA University,
India

Shantan Devathi
Dept. of ECE
SASTRA University,
India

Prashanth Sekhar Chandra Sekaran
Dept. of Mech
SASTRA University,
India

Bala Surendra Kumar Tatikonda
Dept. of ECE
SASTRA University,
India

ABSTRACT

Digital data protection turns out to be the most essential constituent in the cyber world where digital crime is mounting to its worst. To establish secured data communication a technique called Steganography, one of the fruitful attempts for data hiding, has evolved. The rationale of this scheme is to blot out the message in to an innocuous cover media. For good stego system imperceptibility, randomization and capacity are the important considerations. In this paper we propose a novel keyless random algorithm in image steganography which induces an enhanced security by incorporating "counting-out" embedding. It uses message bits embedded in the current pixel which acts as a key for the next pixel to which data is to be embedded. This method provides adaptive randomization without affecting the imperceptibility and capacity.

Keywords

Information hiding, Image Steganography, Counting-out embedding, Adaptive randomization

1. INTRODUCTION

Communication takes a cardinal role in the human evolution which made apes rise to modern day Homo sapiens and it is necessary to carry out the thoughts and visions to fellow beings. To assure security there is a necessity to protect the communication from snoopers. CRYPTOGRAPHY [1-3], a technique evolved to scramble the message with a key so making it difficult to an eavesdropper to interpret. But this system cannot forestall the unintended users to identify the transmission of message. STEGANOGRAPHY [4-19], a sib of the previous technique evolved later which hides the message in unsuspicious cover and provides better security by making the communication invisible. In fact the term steganography in linguistic form is in use centuries ago which cites the usage of Invisible inks, wax tablets etc [17]. In late 1980's with the conceptualization of prisoners' problem, Simons made the initial impact on technical steganography. With the advancement in technology, steganography made its strong presence in digital domain. In the recent times many data hiding techniques have been evolved to hide data in digital media like images, audio, video etc. An effective steganographic strategy has to exhibit imperceptibility, capacity and security. Imperceptibility refers to the quality of the resultant stego image i.e. it should be in well resemblance with cover image. The amount of information that can be concealed in the cover image is called capacity whereas security lies in the degree of randomization provided by the embedding algorithm. Digital watermarking [11, 13, 15, 17], another pal of steganography is also known as fingerprinting and is widely used for copyright protection [1, 11]. In image steganography the simple way to hide the data is to replace the Least Significant Bits (LSB) [6] of each pixel in cover image with data bits. This scheme is simple and can maintain good amount of quality.

In this proposed method to provide a keyless randomization [14] we use the fact that "data is random in nature". Data which is to be embedded can act as a key to embed i.e. next embedding pixel position is decided upon the data embedded in the present pixel. Different embedding paths can be achieved for each secret data.

2. RELATED WORKS

In recent proximity several approaches have been made in building up of steganographic algorithms in the discipline of image processing. These works are classified in to spatial and frequency domain. The proposal of Chan and Chen's 'Hiding data in images by simple LSB substitution' [6] marks the basis of spatial domain techniques with the principle of replacing the least significant bits of the pixels with secret data.

Frequency domain and spread spectrum techniques [8] has been evolved with the usage of Discrete wavelet transform, Discrete cosine transform etc. Here the data is embedded by modifying the high frequency coefficients in the transformed domain. The capacities of these methods are low but have high robustness and are mainly used in copyright protection.

LSB substitution methods are more widely used because of less complexity and time required for the embedding. The data can be embedded as long as the image quality is not degraded. Security concerns always remains as a key factor. Many new approaches using randomization were proposed to enhance the security to higher level. These approaches can be implemented with or without a key. The approaches like using raster and random scan, space filling schemes [15] like Moore, Hilbert, pixel indicator [14, 18]and MSN randomization [19] etc. fall under randomization techniques.

3. COUNTING-OUT

This is a repetitive process of picking an object from a set and omitting it for the next iteration. A number q is chosen in each iterative loop and the set is counted for q times in a specified closed path. This has to be repeated until all the objects in the set are picked.

The illustration of the counting-out is demonstrated in the fig 1. Let a, b, c, d, e, f be the six objects in the set. In this only two iterations are demonstrated to reduce the complexity. The q values are taken as 7 and 8.

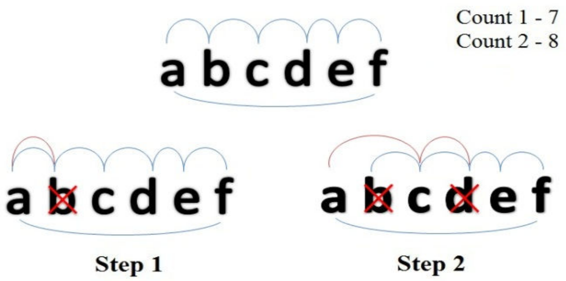

Fig 1 – Counting-out

In figure 1, the step 1 is the outcome after the first counting-out and similarly step 2 represents the outcome after second counting-out.

3. PROPOSED METHODS

In the current approaches we use counting-out technique to randomly place data in image. Image is a vast array of pixels. Adopting counting-out for big numbers is a sophisticated process. To simplify we segment image in to number of 4×4 pixel blocks and the proposed methods 1and 2 are applied on each pixel block. For employing counting-out in the pixel block we need to define a closed path which is shown in the fig 2.

3.1. Method 1

In this process bit stream of data is made in to bursts of 4 bits. First burst of data is embedded in to first pixel of the pixel block and the position of second embedding pixel is determined by counting-out the pixel array with the decimal number represented by first data burst. The process continues till all the pixels in the pixel block are embedded. Then the same is done for all the pixel blocks in the image until entire data is embedded.

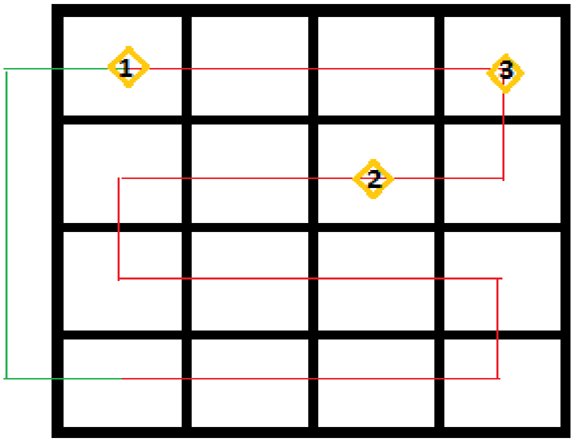

Fig 2 – Embedding path

The figure 2 represents 4×4 pixel block where red path specifies the embedding path in the block and green line denotes the closing circular loop. The yellow rhombus with the number represents the location of the embedded pixel and its serial order. To illustrate the embedding algorithm let us consider the first two data bursts as $(0101)_2$, $(1101)_2$. The embedding starts in the first pixel of the block and is embedded with first burst. Counting-out operation uses the first burst and embeds the second burst in the pixel labeled 2 in the figure 2 (traversing five positions in the loop). Similarly it moves 13 steps for the next embedding. The embedding and extraction algorithm for the method 1are followed in the sections 3.1.1 and 3.1.2.

3.1.1. Embedding algorithm 1

Input : Cover image, data

Output : Stego image

```
1.  Read the input cover image and data
2.  Convert cover image pixel intensities and data in to binary
3.  Group the data in to bursts of 4 bits
4.  Segment the image into $4 \times 4$ sub blocks
5.  Initialize four pointing variables i, j, k, l where i, j are used to point out particular block and k, l acts as local X and Y coordinates within the block.
6.  Initialize i, j to zero
7.  Initialize k, l to zero
8.  Embed the first burst of data in the first pixel of the block using 4 bit embedding
9.  Apply counting out algorithm using the decimal value of data burst to update values of k, l
10. Embed the data in the pixel pointed by k, l
11. Repeat steps 9, 10 until the entire pixel block is embedded
12. Update the i, j values to the next pixel block
13. Go to step 7 until the data is embedded completely
```

3.1.2. Extraction algorithm 1

Input : Stego image

Output : Data

```
1.  Read the stego image.
2.  Convert the image pixel intensities in to binary
3.  Segment the image into $4 \times 4$ blocks
4.  Initialize four pointing variables i, j, k, l where i, j are used to point out particular block and k, l acts as local X and Y coordinates within the block .
5.  Initialize i, j to zero
6.  Initialize k, l to zero
7.  Extract the 4 LSBs of the first pixel
8.  Evaluate the values of k, l using the decimal value of the retrieved last data burst
9.  Extract the data from the pixel pointed by k, l
10. Repeat steps 8, 9 until the data in the entire pixel block is retrieved
11. Update i, j values to the next pixel block
12. Go to step 6 until entire data is retrieved
13. Convert the data in to required format
```

3.2. Method 2

This is an improved version of the previous method. Here instead of having a data burst of 4 bits it is reduced to 3 bits per pixel. After embedding a hybrid burst is utilized for counting-out which is a concatenation of data burst and 4th LSB of the embedded pixel in MSB position. Fig 3 illustrates the embedding procedure of method 2. The binary value of pixel to be embedded is $(10011011)_2$ and the data embedded in the pixel is $(101)_2$ and the hybrid burst formed is $(1101)_2$

1	0	0	1	1	0	1	1

Pixel before embedding

1	0	0	1	1	1	0	1

Pixel after embedding

Fig 3

In the figure 3 bits indicated by bold red color represent embedded data bits, Bit indicated by bold blue is the pixel bit used for hybrid burst formation. The embedding and extraction algorithms of method 2 are followed by sections 3.2.1 and 3.2.2.

3.2.1. Embedding algorithm 2

Input : Cover image, data

Output : Stego image

1. Follow steps 1,2 of embedding algorithm 1
2. Group the data in to bursts of 3 bits
3. Follow steps 4 to 7 of embedding algorithm 1
4. Embed the first burst of data in the first pixel of the block using 3 bit embedding
5. Evaluate the values of k, l using the decimal value of the retrieved last **hybrid** burst
6. Follow steps 10 to 13 of embedding algorithm 1

3.1.2. Extraction algorithm 2

Input : Stego image

Output : Data

1. Follow steps 1 to 6 of extraction algorithm 1
2. Extract the 3 LSBs of the first pixel
3. Evaluate the values of k, l using the decimal value of the hybrid burst of last extracted pixel
4. Follow steps 9 to 13 of extraction algorithm 1

Note: The embedding and extraction algorithms of method 2 indicate only the changes from method 1.

4. SIMULATION AND RESULTS

The embedding and extraction algorithm is implemented using MATLAB 7.0 for three 256×256 8-bit gray scale TIF compression format images Lena, Baboon and Rose.

4.1 Error metrics

The quality of the proposed stego approach can be evaluated by calculating the parameters Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

These are defined as follows:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1} \sum\nolimits_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Where *m, n* are dimensions of the image and I denotes cover image. K denotes stego image.

$$PSNR = 10 \times \log_{10}[\frac{(2^8-1)^2}{MSE}]$$

To have a comparative analysis of above method 1 and method 2 along with 4 bit LSB substitution method their MSE and PSNR values are calculated and tabulated in table 1.

Images		**Lena**	**Baboon**	**Rose**
Method1	MSE	34.8478	34.4716	35.0193
	PSNR (in dB)	32.7090	32.7562	32.6877
Method2	MSE	10.0996	10.1003	10.2164
	PSNR (in dB)	38.0878	38.0875	38.0378
4 bit LSB substitution	MSE	34.5734	34.1640	35.2901
	PSNR (in dB)	32.7434	32.7951	32.6543

Table 1

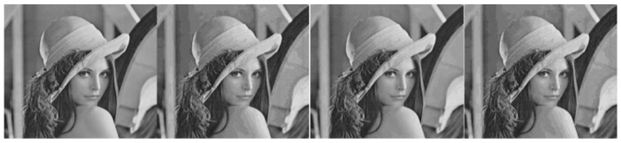

Fig 4- Lena Method 1 Method 2 4 bit LSB

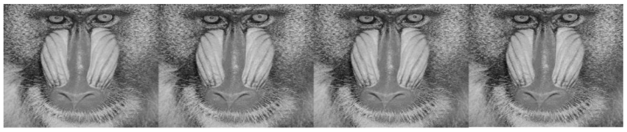

Fig 5- Baboon Method 1 Method 2 4 bit LSB

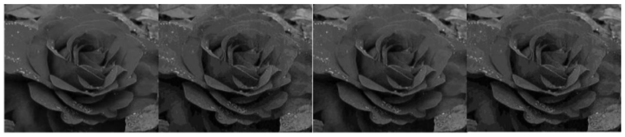

Fig 6- Rose Method 1 Method 2 4 bit LSB

Figures 4, 5, 6 represent the cover and stego images for Method 1, Method 2 and 4 bit LSB substitution method of images lena, baboon and rose.

5. DISCUSSION

By examination of the results in table 1, the proposed method 1 has a similar resemblance with 4 bit LSB substitution method in MSE and PSNR values and they are well above the human visual perception (30 dB). This infers that method 1 improves the security by using keyless randomization without further degrading the image quality.

MSE and PSNR of method 2 are better than that of remaining two methods which implies better image quality. Though it has a reduced capacity it provides better security by employing a hybrid technique which provides keyless randomization by using both data bits and pixel bits.

6. CONCLUSION

A novel successful approach to use data to be embedded as a key is projected in this paper. The proposed methods improve the security of the stego system and maintain a good quality of image. At the reception module the secret message is extracted without any key. In first method embedding data decides the path of embedding. As the data is completely random the embedding paths differ for each message. Where as in the second method embedding path changes even one among the cover image or secret message is varied. The results show that the randomization is achieved without increasing the quality degradation of image.

7. ACKNOWLEDGEMENTS

We are thankful to Prof. Juan Antonio Chávez Domínguez, GSS, UPC, Barcelona, Prof. M. Sridharan and SujithPerla, EEE student of SASTRA University for their support.

8. REFERENCES

[1] BruiceSchneier, Applied Cryptography Protocols, Algorithm and Source Code in C. second edition. Wiley India edition 2007

[2] 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.

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

[4] Moerland, T., "Steganography and Steganalysis", Leiden Institute of Advanced Computing Science,

[5] Silman, J., "Steganography and Steganalysis: An Overview", SANS Institute, 2001

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

[7] Jamil, T., "Steganography: The art of hiding information is plain sight", IEEE Potentials, 18:01, 1999

[8] Marvel, L.M., Boncelet Jr., C.G. &Retter, C., "Spread Spectrum Steganography", IEEE Transactions on image processing, 8:08, 1999

[9] Johnson, N.F. &Jajodia, S., "Exploring Steganography: Seeing the Unseen", Computer Journal, February 1998

[10] 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.

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

[12] Krenn, R., "Steganography and Steganalysis"

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

[14] Adnan Gutub, "Pixel Indicator Technique for RGB Image Steganography", Journal of Emerging Technologies in Web Intelligence (JETWI) (2010)2(1) 56-64

[15] 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

[16] 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

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

[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] RavuruRakesh, ShantanDevathi, PrashanthSekhar Chandra Sekaran, SiramSanath Kumar, Adaptive Randomization in Image Steganography Pertaining to Most Significant Nibble, International Journal of Computer Applications (0975 – 8887), Volume 22 – No. 3, May 2011.