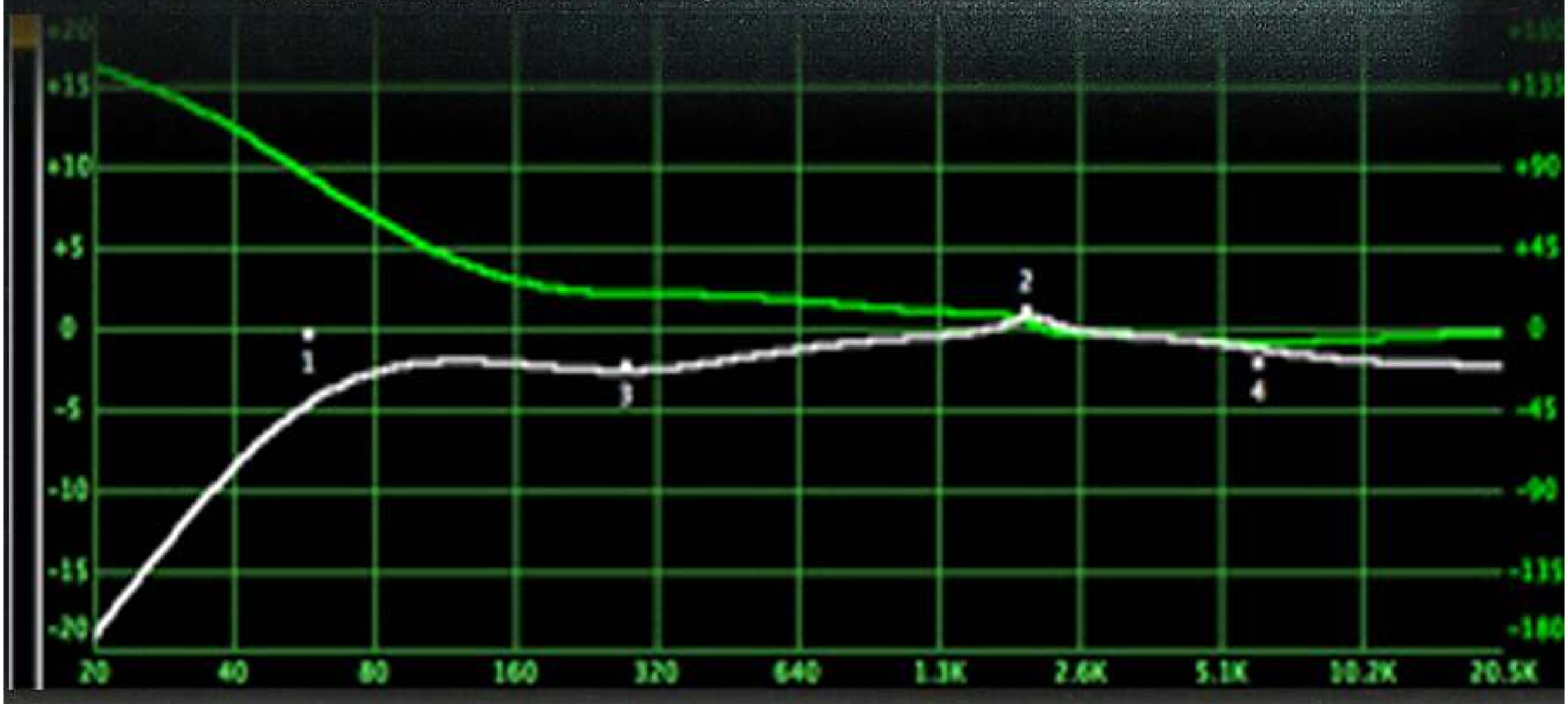
Signal Processing:
An International Journal (SPIJ)

ISSN : 1985-2339

VOLUME 1, ISSUE 1

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR



Signal Processing: An International Journal (SPIJ)

Book: 2007 Volume 1 Issue1
Publishing Date: 31-06-2007
Proceedings
ISSN (Online): 1985-2339

SPIJ Journal is a part of CSC Publishers
http://www.cscjournals.org

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 1, Issue 1, May/June 2007.

Pages

Customized and Secure Image Steganography Through Random Numbers Logic

Sanjeev Manchanda* smanchanda@tiet.ac.in
School of Mathematics and Computer Applications,
Thapar University, Patiala-147004 (INDIA)
**Corresponding author*

Mayank Dave mdave67@yahoo.com
Department of Computer Engg.,
National Instt. Of Technology, Kurukshetra, India

S. B. Singh sbsingh69@yahoo.com
Department of Mathematics,
Punjabi University, Patiala, India

Abstract

Steganography is the science of hiding information in media based data. We present random numbers logic based steganographic methods and layout management schemes for hiding data/image into image(s). These methods and schemes can be customized according to the requirements of the users and the characteristics of data/images. These methods are secure enough to meet the requirements of the users and user can play significant role in selection and development of these methods. Methods can be chosen randomly and implemented dynamically based on inputs, user choices as well as outputs. Experimental results are given to demonstrate the performance of the proposed methods.

Keywords: Steganography, Cryptography, Random numbers logic etc.

1. Introduction

Steganography is the art and science of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "Covered writing." It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications. Steganography and cryptography are cousins in the spy craft family. Cryptography scrambles a message, so that it cannot be understood. Steganography hides the message, so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an "invisible" message created with steganographic methods will not. Modern steganography's goal is to keep hidden message's mere presence undetectable, but steganographic systems because of their invasive nature, leave behind detectable traces in the cover medium. Even if secret content is not revealed, the existence of it can be guessed because modifying the cover medium changes its statistical properties, so eavesdroppers can detect the distortions in the resulting stego medium's statistical properties. The process of finding these distortions is called statistical steganalysis.

In this paper we focus on the developing the techniques that can help hiding messages on the basis of random numbers logic. Present work concentrates upon using Least Significant Bit conversion but is not limited to it. It can involve other methods for steganography discussed in paper. Random numbers based steganographic study is implemented at small scale, but power of random numbers to hide the data/image is not fully exploited and unexplored yet. This paper is an effort to explore the real power of random numbers to hide the messages in secure and customized way.

This paper is organized as follows: Section 2 includes the study of related work. In section 3 we introduce the concept of steganography in more detail, followed by detailed discussion about image processing. In section 4 we discuss about different methods used for steganography. In section 5 includes the proposed methods for steganography based on random numbers logic. In section 6 we discuss few layout management schemes for using methods discussed in section 5. In section 7 we include experimental results to check the strength of proposed methods. Section 8 includes conclusion and future directions for the related research work. Last, but not the least section 9 includes bibliography.

2. Related Work

Steganography is the art of secret communication. Its purpose is to hide the very presence of communication as opposed to cryptography, which aims to make communication unintelligible to those who do not posses the right keys Andersen et al.[1]. The traditional approach to image encoding consists in the source coding, encryption and channel coding Gonzalez et al.[2]. The source coding is used to compress data and match it with the band-width of communication channel. However, the obtained data are sensitive to the communication noise and not protected against unauthorized use. To overcome these disadvantages the next two stages are to be used. To protect data against unauthorized access the encryption is accomplished. The encryption stage is performed separately from source coding. To reduce nuisance of the communication channel noise the channel coding is used which is based on the specialized error correction codes able to detect and correct errors directly during data transmission. Both encryption and channel coding require the introduction of the redundant information in initial data that leads to the increase of data size and corresponded time of transmission. Now a day, we can use digital images, videos, sound files, and other computer files that contain perceptually irrelevant or redundant information as "covers" or carriers to hide secret messages. After embedding a secret message into the cover-image, we obtain a so-called stego-image. It is important that the stego-image does not contain any detectable artifacts due to message embedding. A third party could use such artifacts as an indication that a secret message is present. Once a third party can reliably identify which images contain secret messages, the steganographic tool becomes useless. Obviously, the less information we embed into the cover-image, the smaller the probability of introducing detectable artifacts by the embedding process. Another important factor is the choice of the cover-image. The selection is at the discretion of the person who sends the message. Images with a low number of colors, computer art, images with a unique semantic content, such as fonts, should be avoided as cover images. Some steganographic experts e.g. Aura [3] recommend grayscale images as the best cover images. They recommend uncompressed scans of photographs or images obtained with a digital camera containing a high number of colors and consider them safe for steganography. Pfitzmann and Westfeld [4] introduced a method based on statistical analysis of Pairs of Values (PoVs) that are exchanged during message embedding. Pairs of Values that differ in the LSB only, for example, could form these PoVs. This method provides very reliable results when we know the message placement (such as sequential). However, we can only detect randomly scattered messages with this method when the message length becomes comparable with the number of pixels in the image. Johnson et al. [5, 6] pointed out that steganographic methods for palette images that preprocess the palette before embedding are very vulnerable. Existing cryptographic and steganographic mediums suffer from a myriad of attacks. Johnson [7] has studied such attacks on image steganography, whereas Pal et al. [8] has studied similar attacks in the context of audio steganography. Even though cryptography and steganography are exposed to so many probable attacks, very few people have given a thought to find alternate ways to transmit information. The goal of steganalysis is to defeat steganography methods by identifying the presence of hidden information. This may be done using detection-theoretic methods if the distributions of the cover-image and stego-image are known to the steganalyzer as defined by Cachin [9], and various creative techniques as given by Fridrich et al. [10] etc. Fisk et al. [11] point out the weaknesses of TCP/IP protocol suite and discuss how those weaknesses could be used as covert channels for secret communication, whereas Bao et al. [12] focus on using communication accessories like email headers etc for secret communication. . Avcibas et al. [13] proposed a stegoanalysis technique based on image quality metrics. Fard et al.[18] proposed a novel (Genetic Algorithm) GA evolutionary process to make a secure steganographic encoding on JPEG images. Martín et al. [19] experimentally investigated if stego-images, bearing a secret message were statistically “natural”. Koval et al. [20] discussed the problem of performance improvement of non-blind statistical steganalysis of additive steganography in real images. Luo et al. [21] presented a secure LSB steganography system against sample pair analysis, such as RS, SPA and DIH method by adopting chaotic technique and dynamic compensation skill.

3. A Closer look of Steganography and Image Processing

3.1 Steganography: There are many forms of steganography including audio, video and image media. These forms of steganography often are used in conjunction with cryptography, so that the information is doubly protected; first it is encrypted and then hidden so that an adversary has to first find the information (an often difficult task in and of itself) and *then* decrypt it. The following formula provides a very generic description of the pieces of the steganographic process:

Cover Medium + Hidden Data + Stego Key = Stego Medium

In this context, the *cover medium* is the file in which we will hide the *hidden data*, which may also be encrypted using the *stego key*. The resultant file is the *stego medium* (which will, of course be the same type of file as the cover medium). The cover medium (and thus, the stego medium) are typically image or audio files. In this article, we have focused on image files and therefore, refer to the *cover image* and *stego image*.

3.2 Image Processing: An image file is merely a binary file containing a binary representation of the color or light intensity of each picture element (pixel) comprising the image.

The simplest approach to hiding data within an image file is called *Least Significant Bit (LSB) insertion*. In this method, we can take the binary representation of the hidden data and overwrite the LSB of each byte within the cover image. If we are using 24-bit color, the amount of change will be minimal and indiscernible to the human eye. As an example, suppose that we have three adjacent pixels (nine bytes) with the following RGB encoding:

10010101	00001101	11001001
10010110	00001111	11001010
10011111	00010000	11001011

Now suppose we want to "hide" the following 9 bits of data (the hidden data is usually compressed prior to being hidden): 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in **bold** have been changed):

10010101	0000110**0**	11001001
1001011**1**	0000111**0**	1100101**1**
10011111	00010000	11001011

Note that we have hidden 9 bits successfully, but at a cost of only changing 4, or roughly 50%, of the LSBs.

This description is meant only as a high-level overview. Similar methods can be applied to 8-bit color but the changes, as the reader might imagine, are more dramatic. Gray-scale images, too, are very useful for steganographic purposes. One potential problem with any of these methods is that an adversary who is looking can find them. In addition, there are other methods besides LSB insertion with which to insert hidden information. These methods based on random numbers logic are the subject matter for present paper.

4. Image Steganography Methods

Image steganography has been widely studied by researchers. There are a variety of methods used in which information can be hidden in images. Some of them are described here given by Lee et al. [14], Chan et al. [15], Chang et al. [16], and Hsu et al. [17].

4.1 Replacing Least Significant Bit: In image steganography almost all data hiding techniques try to alter insignificant information in the cover image. For instance, a simple scheme proposed by Lee et al. [14], is to place the embedding data at the least significant bit (LSB) of each pixel in the cover image. The altered image is called stego-image. Altering LSB doesn't change the quality of image to human perception but this scheme is sensitive a variety of image processing attacks like compression, cropping etc.

4.2 Replacing Moderate Significant Bit: Chan et al. [15] showed how to use the moderate significant bits of each pixel in the cover image to embed the secret message. This method improves sensitivity to modification, but it degrades the quality of stego-image.

4.3 Transformation Domain Techniques: Other familiar data hiding techniques use the transformation domain of digital media to hide information discussed by Chang et al. [16] and Hsu et al. [17]. Functions such as the discrete cosine transform (DCT) and the discrete wavelet transform (DWT) are widely applied by Chang et al. [16], and Hsu et al. [17]. These methods hide the messages in the significant areas of the cover image, which makes them robust against compression, cropping and other image processing attacks.

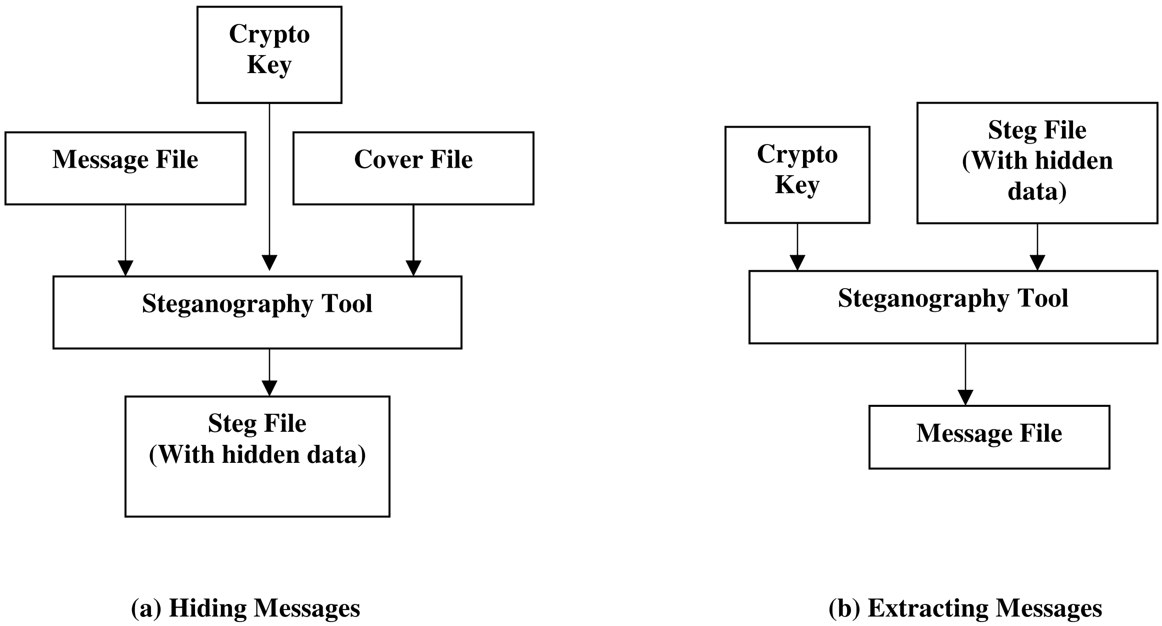
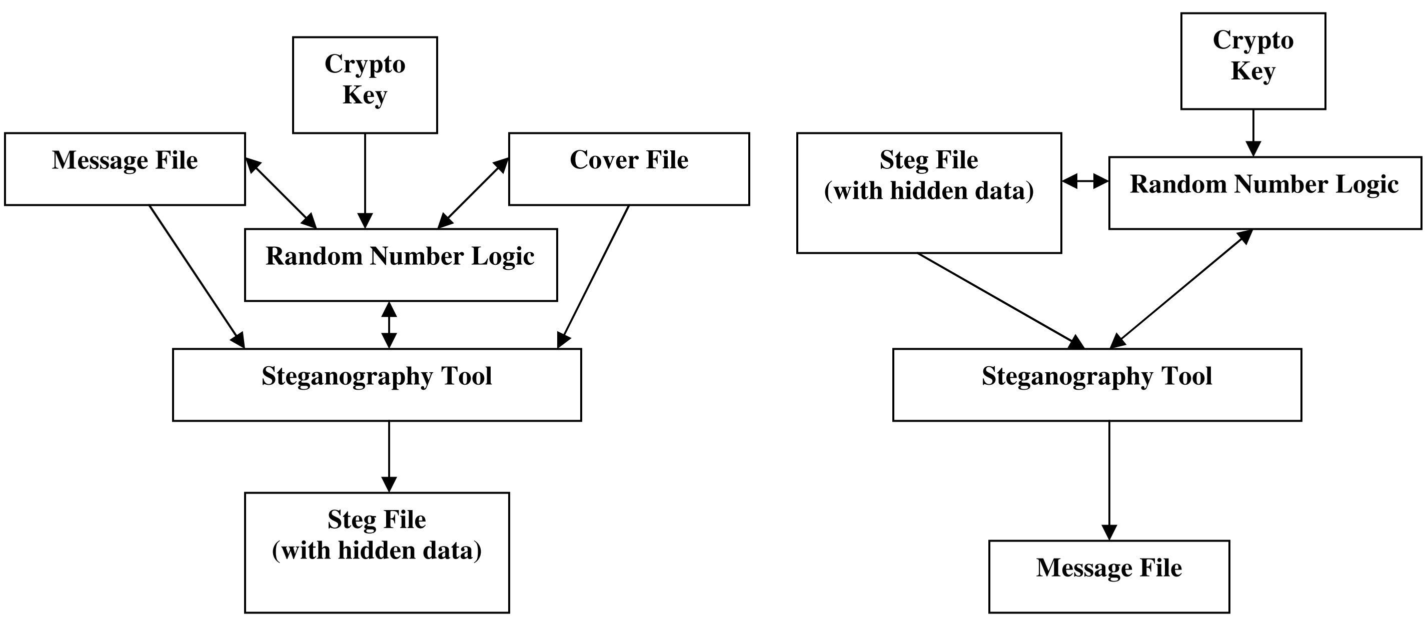


Figure 1: General Image Steganography System



(a) Hiding Messages **(b) Extracting Messages**

Figure 2: Proposed System Based on Random Numbers Logic

5. Proposed Methods

In proposed system, random numbers logic plays an important role in Customization and secure implementation of staganography (figure 2). As type of different methods progress more complexity is added to the implementation.

5.1 Type I Methods (Simple Methods): An obvious method that can be constructed from combination of random numbers generator and least significant bit method (Other methods may also be employed on same tracks, but current paper include only Least Significant Bit based method).

Following general congruential method may be used to generate the location of next bit to be replaced i.e. after x_{i+1} bytes from current byte may be used as target byte to replace the least significant bit.

$$\mathrm{x}_{i+1} = f(x_i, x_{i-1}, ..., x_{i-n+1}) (\bmod m) = (\mathrm{a}_1 \mathrm{x}_\mathrm{i} + \mathrm{a}_2 \mathrm{x}_{\mathrm{i-1}} + ... + \mathrm{a}_\mathrm{n} \mathrm{x}_{\mathrm{i-n+1}} + \mathrm{c}) (\bmod \mathrm{m})$$

In above Formula all m, n, a, c and x are non-negative integers. Given that the previous random number was x_i, the next random number x_{i+1} can be generated. The numbers generated by a congruential method are between 0 and m-1. e.g. suppose that we have three adjacent pixels (nine bytes) with the following RGB encoding:

10010101	00001101	11001001
10010110	00001111	11001010
10011111	00010000	11001011

Now suppose we want to "hide" the following 4 bits of data (the hidden data is usually compressed prior to being hidden): 1011.Random number generator used for hiding data is

$$x_{i+1} = ax_i + c \,(\bmod m)$$

e. g. x_0=a=c=7 and m=3 have list of random numbers (called psuedo random numbers) as 2, 0, 1, 2, 0, 1… bits stored for hiding in above data is marked by bold bits in following data, whereas underlined bit represents modified bits.

10010101	00001101	1100100**1**
1001011**0**	00001111	1100101**<u>1</u>**
10011111	00010000	1100101**1**

The number of successively generated pseudo-random numbers after which the sequence starts repeating itself is called the *period*. Setting up the values of the constants may be divided into two categories of full period or partial period. If the period is equal to m, then the generator is said to have a full period, otherwise partial period. Kelton et al.[22] may be referred for the detailed discussion over setting up the values of these constants. Here we can say that full period methods are observed to be lesser complex and secure as compared to partial period methods.

This method will include the hidden data at random places of different byte in the image. This method will inversely affect the density of bits to be stored in cover media by a factor m/2 (divisor/2). So there is a need to keep a fit between the divisor m and number of bits in hidden data. Increasing the size of cover media may create suspicion to steganalyst about hidden data in image. So there is a need to check the characteristics of cover image and hidden image to formulate the random number generator variables. Second issue about this method is its security aspect. Different combinations of least significant bits may be analyzed very easily and repetition of same sequence will make the steganalysts' job much easier. So this method is not secure enough to meet the requirement of steganography, this method is very easy and may be customized without much effort. Thousands of combinations of functions and values of variables are available to implement it. So, this method can be customized easily and give initial level of security to data.

5.2 Type II methods (Low complexity methods): These methods are based on the combinations of two or more type I methods to hide the data. There is a control procedure attached to these methods that controls the sequence of these methods randomly. This randomness is based on the output generated by earlier sequence. Figure 3 will illustrate the procedure of implementing the randomness.

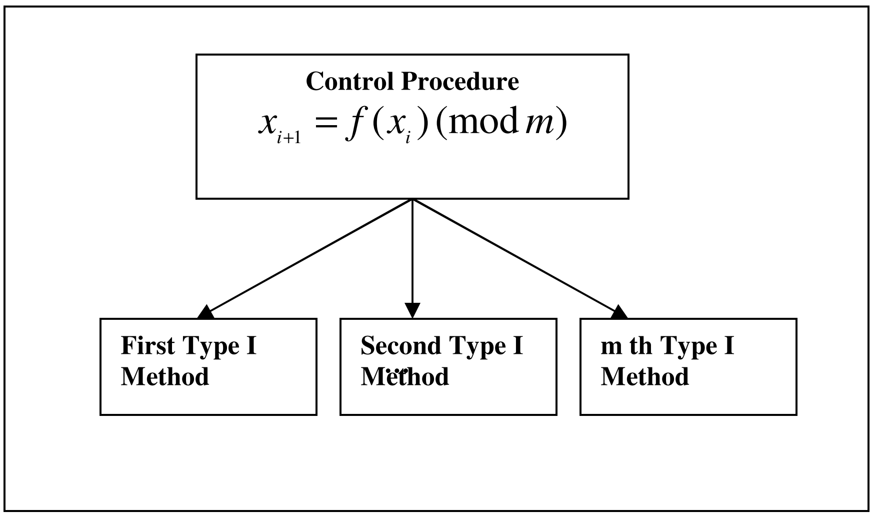


Figure 3: Type II Low complexity methods

Control procedure decides the method to be used for steganography. The programmer initializes initial values of the variables, whereas later values of variables are generated from the output received from earlier implementation, e.g. Sequence of 100 bytes is used to hide a sequence of 20 bits in it. First 10 bits are stored in first 50 bytes of cover media o the basis of initial values supplied to control procedure. A sequence of methods without repetition is involved for the purpose. Output generated is taken as the source of selecting values for variables. If we need only one value for function f, last few bytes of output may be used for the purpose and remaining data is processed on the basis of altered sequence of Type I methods. Implementing the procedure in reverse direction, predefined initial sequence of methods is guided by input data for further iterations, as value of variables is stored in input file. In this way data hiding may be performed in a customized and secure way as functions are decided by users whereas altering sequence of storage can provide security in a well manner.

5.3 Type III method (Moderately complex methods): In this type of methods different encryption algorithms are attached to type II methods. Encrypted hidden image is stored in the cover image through type II methods. Key for encryption is generated through random procedures or is decided by the user and prefixed to hidden data/image.

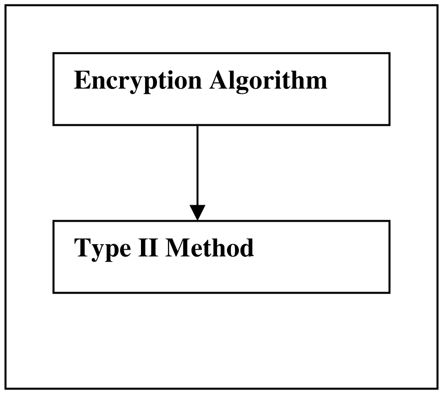


Figure 4: Type III moderate complexity methods

5.4 Type IV method (Highly complex methods): In this type of methods one set of type III methods is used as one block of stego tool. Two or more such blocks (different from each other) may be combined together in parallel, in series or a combination of both as follows:

5.4.1 Parallel combination: In this method different block of type II or type III method may be combined together to hide the data. This kind of scheme is suitable for large images. Cover image and hidden image are divided into small parts (blocks) and each block is applied to hide the data. Distribution of data for different blocks is implemented on random numbers logic.

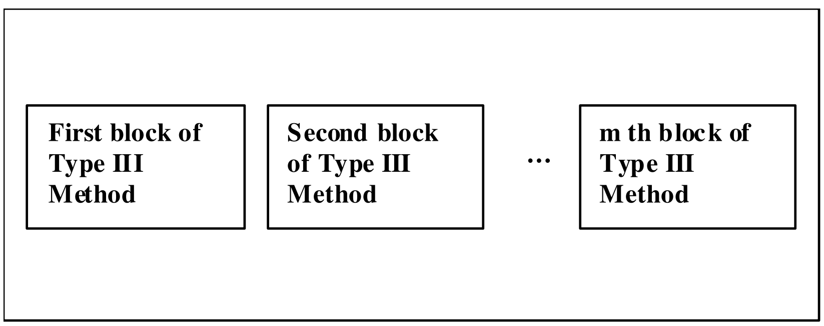


Figure 5: Parallel combination of Type III methods

5.4.2 Serial Combination: In this method different block of type II or type III method may be combined together to hide the data. This kind of scheme is suitable for relatively small hidden image/data, but security is the major concern. In this method cover image, intermediate cover images and hidden image are used for the purpose as shown in figure 6. Initial hidden image is hidden in intermediate cover image, which is further hide into cover image and to generate stego image to be communicated. Such a combination is secure enough to be analyzed, but density of data that one can hide will be very less in quantity.

5.4.3 Hybrid combination: In this method parallel and serial combinations are combined together to hide the data. Different layers of parallel combinations are combined together to hide the data. One parallel combination is called a layer, two or more such layers may be combined together to generate the stego image using intermediate cover images.

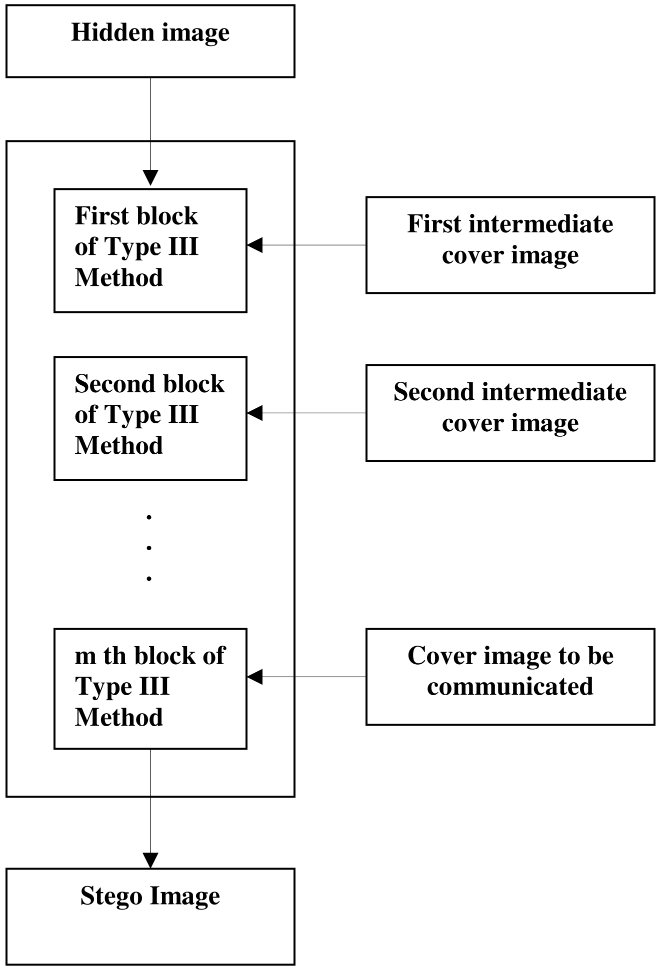


Figure 6: Serial combination of type III methods

6. Layout Management Schemes

Methods discussed in previous section can contribute a lot in hiding information in images. These methods can be made more sophisticated with the help of layout planning. Here layout planning comprises of three basic components i.e. starting point, flow of data and termination point. Starting point is the initial point from where to start hiding information with the help of methods discussed in previous section. Flow of data comprises of the pixel by pixel flow of data hiding in cover image. Termination point is somewhat obvious from the layout i.e. the pixel at which the process of hiding data will terminate. One other component may be the visual sense. By the help of visual senses of eyes, we can decide the part of images, where data can be hidden without being noticed by the stegoanalysis evaluators. Using human eyes may contribute in deciding the part of image, but affects the artificial intelligence of the system, so becomes discretion of user to use it or not. Different Layout Management Schemes are divided under different levels as follows:

Level 0 (Simple Layout Scheme): At this level different types of methods are selected from section 5 and are implemented from the starting point of the cover image and continued pixel by pixel from first pixel to last pixel of the image.

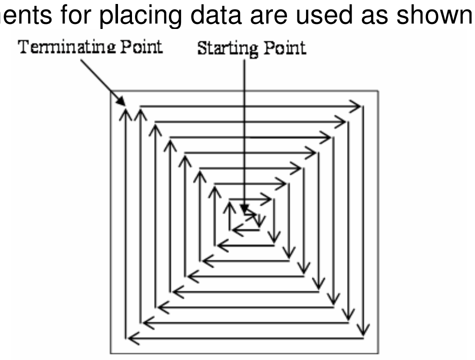
Level 1 (Plain/Garbage Data Insertion): In this scheme before starting point original(from cover image) plain text or garbage data is placed into cover image followed by the hidden data according to the selection of methods from section 5.

Figure 7: Two variations of Level 1 schemes.

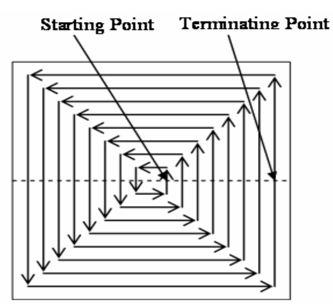
In figure 7 first diagram indicate to leave few pixels before starting and after terminating point, whereas other diagram includes original cover image pixels all around the hidden data. Second diagram is only possible, when hidden data is small as compared to the space available for hidden data in cover image.

Level 2 (Image Processing Operations Based Scheme): Before placing hidden data in cover image, cover image pixels may be inverted, rotated or flipped and then hidden data may be placed according to the combinations of methods selected from section 5 and low level layout schemes.

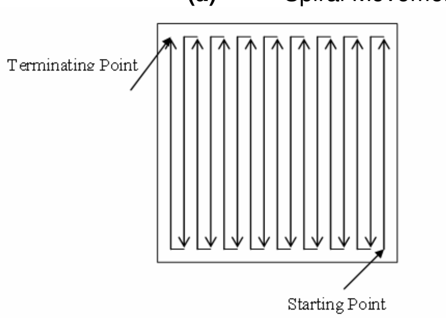
Level 3 (Spiral, Squared, Snake or Ray Movement Based Scheme): In this scheme different movements for placing data are used as shown in figure 8.



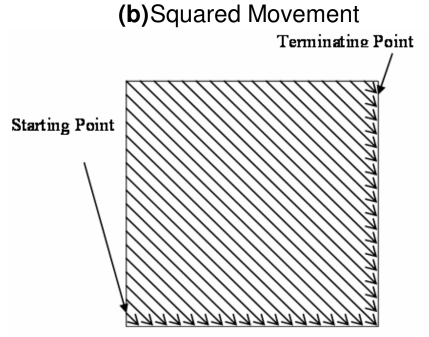
(a) Spiral Movement



(b) Squared Movement



(c) Snake Movement



(d) Ray Movement

Figure 8: Spiral, Squared, Snake or Ray Movement Based Scheme

Level 4 (Marginally Expandable Scheme): This scheme is based on Intersecting Rectangles. In this scheme corners and intersecting points are excluded, resulting into minor expansion of space requirement for hiding data. Starting as well as terminating points can be selected as per the discretion of users for all schemes, so current and following schemes may have alternative choices for them and are not displayed exclusively now onwards.

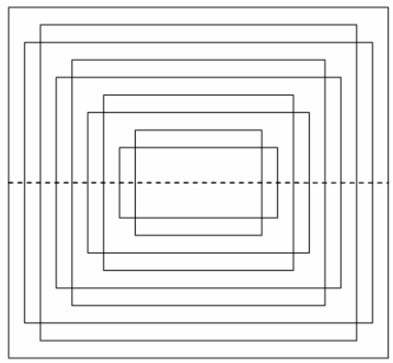


Figure 9: Intersecting Rectangles Based Scheme

Level 5 (Significantly Expandable Scheme): In this scheme squares of pixels are partially used and data is placed only on remaining parts of these squares.

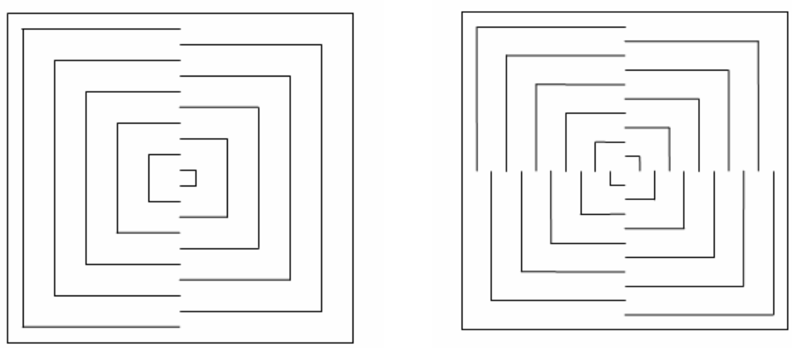


Figure10: Partial Squared Implementations

Level 6 (Very Low Density Hidden Data Scheme): Low Density of Hidden data as lines and dots (pixels) are the places used for hiding data

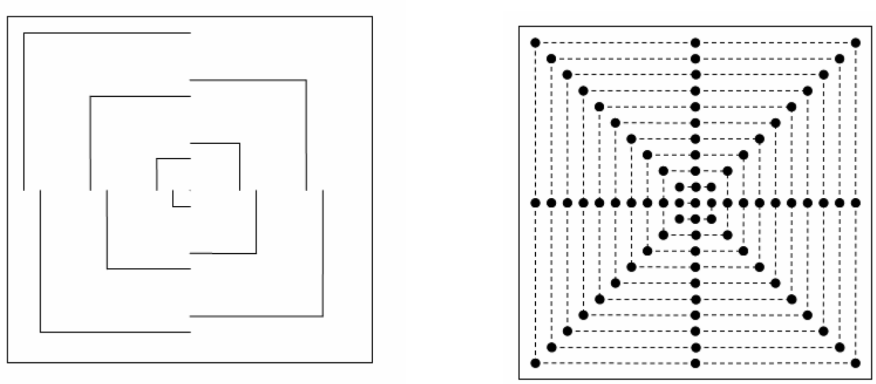


Figure 11: Very Low Density Hidden Data Scheme

7. Experimental Results and Comparison with Other Tools

Experiments are performed to check the performance of different methods using variety of random number generators with varying values of variables and constants. Least Significant Bit transformation was the major way to implement the methods but other techniques like moderate significant bit transformation and transformation domain technique were also tried on similar lines, but here we produce the results based on least significant bit transformation based technique. Four hundred such combinations are tried on all four types of methods and results are combined together as follows:

7.1 Density of hidden data: First of all one of the most important issue is that how much data can be stored in cover image and selection of suitable divisor. A simple Least Significant Bit method requires n bytes for hiding n bits. This density is decreased m times of total storage capacity for type I method and it is further degraded for other methods as shown in the figure 12.

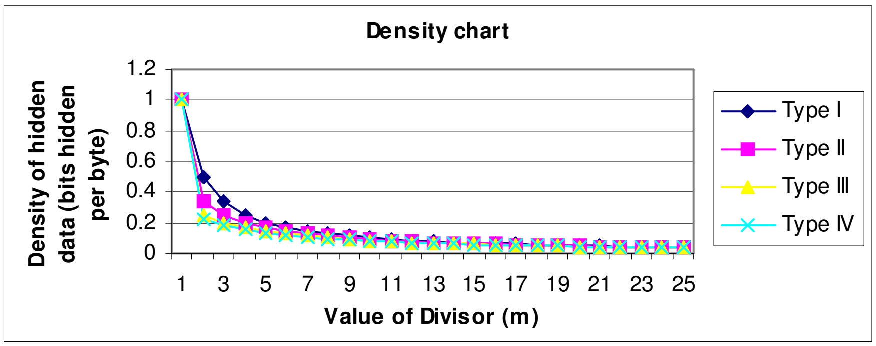


Figure 12: Density chart- Bits hidden per byte vs. value of divisor

It is obvious to receive more densely placed hidden data for smaller value of divisor 'm' as compared to a larger one and it is being indicated in figure 12 as well.

7.2 Quality of Cover image: There are many metrics to quantify the quality of two or more images. To compare these methods objectively, we adopt Mean Square Error (MSE) as our quality metrics. The MSE between the original image $I_o(x,y)$ and the stego image $I_r(x,y)$ for each color channel is defined as :

$$MSE = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n}[I_o(x,y) - I_r(x,y)]^2}{m \cdot n}$$

where m x n is the size of the image

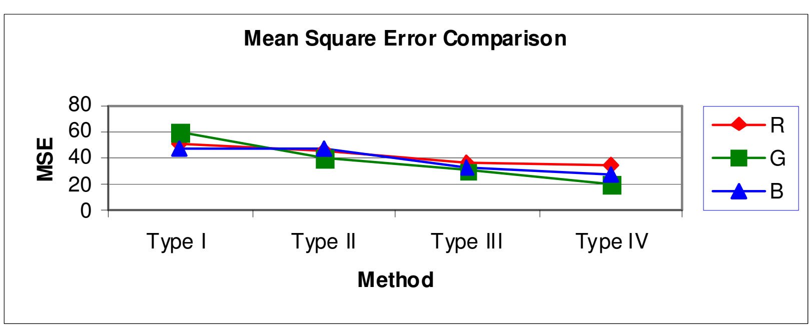


Figure 13: Mean Square Error comparison- Mean Square Error vs. Type(s) of method.

Results indicate that type I methods hides more data so these affect the color combination of different image it, but less for other methods.

7.3 Security of hidden image: Different tools available in market like s-tools, EzStego, Stegdetect, Stego suite etc. are tried on different methods and it is being observed that these tools are well conversant with type I methods, but are unable to detect the presence of hidden data of other complex combinations.

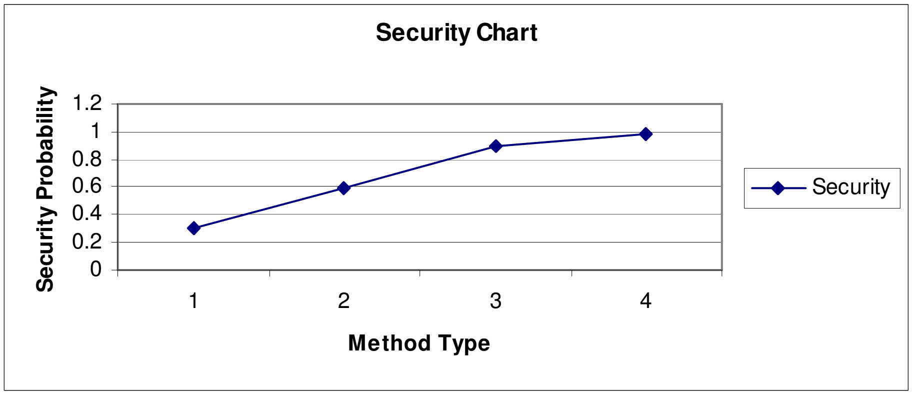


Figure 14: Security chart- Security Probability of different methods.

Experiments performed on all four types of methods through different tools available for stegoanalysis indicate that complex combination of methods are very secure as compared to the simpler one and figure 14 confirms it as well.

7.4 Complexity of algorithms for different methods: Numbers of lines used executed for hiding data increases rapidly as we use more complex methods. As type I is simple method, it is less complex. Complexity increases as we use further much more complex methods.

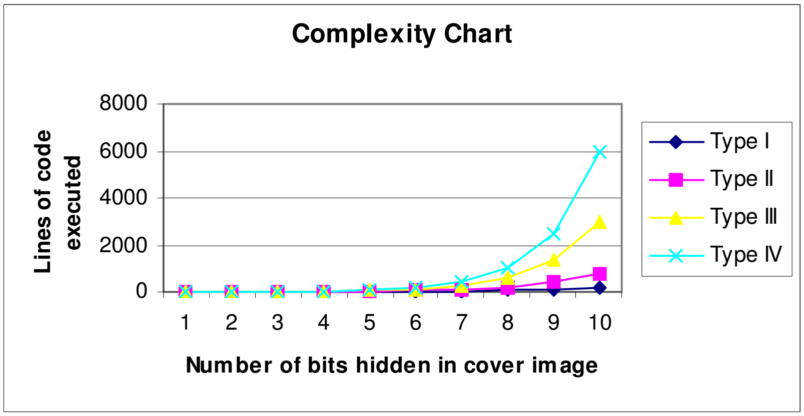
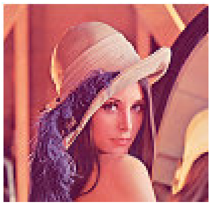


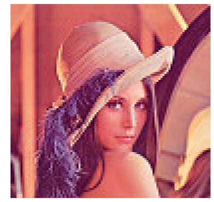
Figure 15: Complexity chart- Lines of code executed for hiding numbers of bits in cover image.

Figure 15 indicates that complexity of algorithms increases as we move from simple to complex methods. Following images are included to show that image ‘a’ and ‘c’ can’t be distinguished by human eyes, but these are significantly different from each other.



(a) Cover image (Lena.tiff)

(b) Hidden image (missile.bmp)

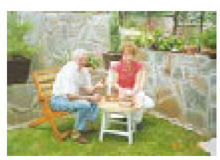


(c) Stego image (lenaNew.tiff)

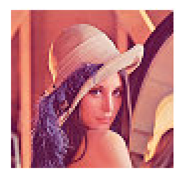
Figure 16: Steganography images: Differentiating images 'a' and 'c' is impossible through eyes.

7.5 Comparison of Proposed Methods with Other Tools: We applied combinations of proposed methods and layout management schemes over many images and compared outcome with the results of many tools available for image staganography. Fridrich et al.[23] proposed RS Steganalysis method to detect the stego images, when applied to other tools and our methods, our methods performed far better than tools like Steganos, S-Tools and Hide4PGP. Table 1 includes the comparison of results for proposed methods with other tools. Even original cover-images may indicate a small non-zero message length due to random variations, which iss termed as initial bias of an image. So first row of the table includes initial bias values for different images. Three images used for study are 'Siesta.bmp' (24-bit color scan, 422×296, message=20% capacity, 100% = 3bpp), 'Cat.bmp' (24-bit color image, 1024×744, message= 5%) and 'Leena.tiff' (24 bit color image, 1024x768, message=12%). According to Fridrich et al.[23] estimated number of flipped pixels in an image must be the sum of half of the hidden message and initial bias (10 + 2.5 = 12.5 for Red color of 'Cat.bmp' image, 2.5 + 0.0 = 2.5 for Red color of 'Siesta.bmp' image and 6 + 1.5 for Red color of 'Leena.bmp' image). Results indicate that the other tools used for comparison are inefficient against RS Steganalysis as actual values are very close to expected values indicating the presence of stego messages, whereas proposed methods are efficient enough to face RS Steganalysis, as actual values are significantly lesser than expected values. Results conclude that proposed methods are far from detection. These results support the claim for proposed methods to be secure enough.

'Cat.bmp'



'Siesta.bmp'



'Lena.tiff'

Figure 17: Images used for testing safety of proposed methods.

Image / Tools	Cat.bmp			Siesta.bmp			Lena.tiff		
	R	G	B	R	G	B	R	G	B
Initial Bias	2.5	2.4	2.6	0.0	0.17	0.33	1.5	1.2	1.8
Steganos	10.6	13.3	12.4	2.41	2.7	2.78	7.2	8.6	8.6
S-Tools	13.4	11.4	10.3	2.45	2.62	2.75	8.4	7.8	7.5
Hide4PGP	12.9	13.8	13.0	2.44	2.62	2.85	7.8	8.9	8.8
Proposed Methods	5.4	4.8	5.7	1.44	1.57	1.68	3.5	3.2	4.1

Table 1: Initial bias and estimated number of pixels with flipped LSBs for the test images.

8. Conclusion and Future Directions

In current paper we have applied random numbers based methods and layout management schemes on least significant bit transformation for steganography. We formulated thousands of variants based on users' choices and calculated the results of around four hundred such combinations. Results cover the security, complexity, density and quality aspects of these methods. Authors are working upon steganalysis for the proposed methods and layout management schemes. In future, research work will be continued on similar lines for implementing more such combinations and for creating steganalysis tools for such combinations.

9. Bibliography

1. Andersen R.J. and Petitcolas F.A.P. *"On the limits of steganography".* IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection 16 No.4, 474-481, 1998

2. Gonzalez R.C. and Woods R.E. *"Digital Image Processing".* Addison Wesley, Reading, 1992

3. Aura T. *"Practical invisibility in digital communication".* Lecture Notes in Computer Science, vol.1174, Springer-Verlag, pp. 265-278, 1996

4. Westfeld A. and Pfitzmann A. *"Attacks on Steganographic Systems".* Lecture Notes in Computer Science, vol. 1768, Springer-Verlag, Berlin, pp. 61-75, 2000

5. Johnson N. F. and Jajodia S. *"Steganography: Seeing the Unseen".* IEEE Computer, pp.26-34, February 1998

6. Johnson N. F. and Jajodia S. *"Steganalysis of Images Created Using Current Steganography Software".* Lecture Notes in Computer Science, vol.1525, Springer-Verlag, Berlin, pp. 273-289, 1998

7. Johnson N. F. *"Steganalysis of images created using current steganographic software".* Second Information Hiding Workshop, 1998.

8. Pal S.K., Saxena P. K. and Mutto S.K. *"The Future of Audio Steganography"*. Pacific Rim Workshop on Digital Steganography, Japan, 2002

9. Cachin C. *"An information-theoretic model for steganography"*. D. Aucsmith (Ed.): *Information Hiding, 2nd International Workshop*, vol. 1525 of *Lectures Notes in Computer Science*, pp. 306-318. Springer, 1998

10. Fridrich J. and Goljan M. *"Practical steganalysis of digital images-state of the art"* Proc. SPIE Photonics West, Vol. 4675,pp. 1-13, San Jose, California, Jan. 2002

11. Fisk G., Fisk M., Papadopoulos C. and Joshua N. *"Eliminating Steganography in Internet Traffic with Active Wardens"*, Available: http://citeseer.nj.nec.com/fisk02eliminating.html. Nov 1, 2002

12. Bao F. and Wang X. *"Steganography of short messages through accessories"* Pacific Rim Workshop on Digital Steganography, Japan, 2002

13. Avcibas I., Memon N. and Sankur B. *"Steganalysis using image quality metrics".* IEEE Transactions on Image Processing, vol. 12, pp. 221–229, Feb. 2003

14. Lee, Y. K. and Chen L. H. *"High Capacity Image Steganographic Model". IEE Proceedings Vision, Image and Signal Processing, pp. 288-294, 2000*

15. Chan Y. K. and Chang C. C. *"Concealing a Secret Image Using the Breadth First Traversal Linear Quad tree Structure". IEEE Proceedings of Third International Symposium on Cooperative Database Systems for Advanced Applications*, pp. 194-199, 2001

16. Chang C. C., Chen T. S. and Chung L. Z. *"A Steganographic Method Based upon JPEG and Quantization Table Modification"* Information Sciences, vol. 141, pp. 123-138, 2002

17. Hsu C. T., Wu J. L. *"Hidden Digital Watermarks in Images"* IEEE Transactions on Image Processing, pp. 58-68, 1999

18. Fard Amin Milani, Mohammad R., Akbarzadeh T. and Varasteh A. Farshad. *"A New Genetic Algorithm Approach for Secure JPEG Steganography"*, Proceedings of the IEEE International Conference on Engineering of Intelligent Systems ICEIS'2006, pp. 216-219, ISBN: 1-4244-0456-8, Islamabad, Pakistan, April 22-23 2006

19. Martín Alvaro, Sapiro Guillermo and Seroussi Gadiel, *"Is Image Steganography Natural?".* IEEE Transactions On Image Processing, Vol. 14, No. 12, December, 2005

20. Koval Oleksiy, Voloshynovskiy Svyatoslav, Holotyak Taras and Pun Thierry. *"Information-theoretic Analysis of Steganalysis in Real Images".* International Multimedia Conference**,** Proceeding of the 8th workshop on Multimedia and security, Geneva, Switzerland, Pages: 11 – 16, 2006

21. Luo Xiangyang, Hu Zongyun, Yang Can, Gao Shanqing. *"A Secure LSB Steganography System Defeating Sample Pair Analysis Based on Chaos System and Dynamic Compensation"*, Feb. 20-22, 2006 ICACT 2006

22. Kelton W. D. and Law A. *"Simulation Modeling and Analysis".* Mcgraw-Hill Science, USA, 2006

23. Fridrich J., Goljan Miroslav and Du Rui. *"Reliable detection of LSB steganography in color and grayscale images".* International Multimedia Conference Proceedings of the workshop on Multimedia and security: new challenges, Ottawa, Ontario, Canada, Publisher ACM Press New York, NY, USA, 2001

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA