Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 5 (3): 363-372, 2013
ISSN 1815-7432 / DOI: 10.3923/rjit.2013.363.372

A New Steganography Algorithm against Chi Square Attack

[1]V. Thanikaiselvan, [1]K. Santosh, [1]D. Manikanta and [2]Rengarajan Amirtharajan
[1]School of Electronics Engineering, VIT University Vellore, 632014, Tamilnadu, India
[2]School of Electrical and Electronics Engineering SASTRA University India-613 401, India

Corresponding Author: V. Thanikaiselvan, School of Electronics Engineering, VIT University Vellore, 632014, Tamilnadu, India

ABSTRACT

The Internet mostly uses insecure links, where information for communication is put to test through its open exposure to interception. As far as information is concerned, its safeguarding measures are of prime concern nowadays. Some solution to be discussed is how to pass the not to be disclosed message in a fashion that its subsistence is made unidentified to invaders. Stenography is a hiding mechanism where the charisma of the secret is veiled by infixing the same in cover files. Steganalysis is the field associated with steganography detection by all its attributes which has now acknowledged much notice from media, law enforcement etc. Chi square test is the one such statistical technique used for steganalysis. This paper purports an algorithm for embedding data within images such that the chi square test fails to detect the hidden information. Merits and demerits of the algorithm are also discussed. Simple LSB substitution along with proposed modified LSB substitution is discussed with the help of PSNR and Chi square value. Another algorithm called as IPMCS (Increased Probability of matching between Cover and Secret data) is introduced which can be used to improve the image quality in terms of PSNR. It is important to know that IPMCS works well only for those images whose PSNR value obtained after applying OPAP is below a threshold value. Stegnographers should obtain stego outputs which when seen from naked eye, are impossible to tell apart from their corresponding covers.

Key words: LSB, OPAP, PSNR, IPMCS, Chi Square test

INTRODUCTION

The motto behind stegnography is hiding one message in other clean images such that they should not give any clue to the third party about even the presence of a secret, thus, is the form of security through obscurity (Amirtharajan *et al*., 2012; Cheddad *et al*., 2010). A Carrier which carries the surreptitious message is labeled cover (image) whereas one obtained after hiding the message is labeled stego (image). During this process of hiding the data, the eminence of the former gets sullied. In most stegnography techniques the bits that belong to the secret unswervingly substitute the LSBs pixels in covers. High capacity can be achieved by using these LSB substitution techniques (Amirtharajan and Rayappan, 2012a-d).

An LSB substitution data hiding technique along with OPAP would maximize images's SNR drastically (Chan and Cheng, 2004). OPAP reduces Worst Mean Square Error to less than half which is gained through simple LSB routine. Experimental results show that with little computational complexity OPAP can improve the image quality in a remarkable way.

The most commonly known stegnographic algorithms in the field of data hiding are the simple LSBs substitutions. In this method, fixed length of secret gets rooted in fixed length right most LSBs of cover pixels. Though this seems simple, it causes some visual distortion when more LSBs are replaced by the secret data (Amirtharajan and Rayappan, 2012a-c). One of the most important parameters in stegnography is PSNR indicating efficiency of the embedding algorithm. It is a appraisal of image revival quality in lossy compression codecs. Signal, here, is nothing but original data; also, noise-blunder initiated through hiding data in the images mainly in LSB substitution. The higher the PSNR, better the visual quality.

Chi square test: Chi-square test is one of the algorithms to test the detectability of any secret data embedding algorithm (Zanganeh and Ibrahim, 2011). It is a numerical assail intending to determine embedding in pixels unswervingly. Based on frequency with which pixel values appear, the chi square value is calculated. There is no need of a authoritative line linking stochasticity in the stego upshot and non-uniformity that is sufficient enough for clean image. Here, this chi-square run comes to play which is widely accepted as an apt thing to test this dissimilarity, there by providing quantifiable result.

The next section is the proposal of an algorithm to improve PSNR of the stego-image.

IPMCS

As mentioned earlier in the paper OPAP increases the image eminence by increasing PSNR. There be some cases where PSNR can be still increased by applying our algorithm. IPMCS basically stands for increased probability of matching between cover and secret files. Apart from increasing PSNR of stego output, it also converts the data into an unintelligible form so that it makes difficult for the hacker to retrieve the information. The key generated after applying this algorithm is also transmitted so that the receiver at the receiving point can again restore the original information.

In this study, an embedding algorithm is proposed which escapes chi square test to detect the hidden information. LSB substitution is used along with proposed method. IPMCS algorithm is also used to progress the image quality in terms of PSNR. The proposed algorithms are implemented with a step by step procedure and the simulation results are included which can be used to analyze the parameters like PSNR, chi-square values.

RELATED WORK

Steganography grabs its attention world wide, which forces the research community in scheming algorithms to improve robustness (Al-Frajat *et al*., 2010; Hansen, 2007; Provos and Honeyman, 2003; Wang *et al*., 2001; Stefan and Fabin, 2000; Thenmozhi *et al*., 2012; Zhu *et al*., 2011).

Cheddad *et al*. (2010) in their paper discussed on chief practices in steganography which are exploited in digital images additionally associated with their processing. Robustness and payload capacity is discussed for various methods.

Chan and Cheng (2004) explained the data infixing strategy via simple LSB in images with OPAP by which visual presence and PSNR of stego images are deeply enhanced. Extensive experiments show that the effectiveness of this method.

Amirtharajan *et al*. (2010) in his study has given a comparative analysis of different stegnography methods. The paper includes computed values of MSE, PSNR of all methods and

discussed on major emerging stegnographic techniques such as LSB based, Inverted pattern based LSB using MSE, OPAP and relative entropy in substitution and a string of 1 and 0 and mod based (Padmaa *et al*., 2011).

Hansen (2007) considered the embedding of data in images by making use of singular value decomposition. He asserted that chi-square test measures the matching level between the given trial with the theoretically random one. It acts very well in studying the divergence, given quantitative and numerical end result.

Provos and Honeyman (2003) discussed about the existing stegnographic strategies and steganalysis. Their paper includes an introduction to stegnography and explains the basics of embedding.

PROPOSED METHOD

Algorithm for chi square test on stego-images: The Chi-square test detects presence of furtive message in cover using some statistical properties of the images. It is unusual in case where pixel value's frequency 2k is almost same as that of 2k+1 indistinctive image having no embedded data. This property is exploited in detecting secret message in cover image. The Algorithm is given below:

Step 1: Say n_i = total pixels in image sample of gray value 2i. Suppose these values have even parity with no loss of generality

Step 2: Say n_i^* = expected value of pixels in the sample image having gray value ni, pressuming our image is embodied with homogeneously dispersed payload:

$$n_i^* = (|\text{pixels with gray value } 2i| + |\text{pixels with gray value } 2i+1|)/2$$

Step 3: In the first column in table, accumulate gray value 2i, in second, gray value 2i+1. If less than 8 pixels possess values in the range {2i, 2i+1}, then, i_{th} bin is joint with one that is nearer to it, along with the coalesce of even values and odd values all together

Step 4: The chi-square (x^2) run having k-1 DOF is expressed as:

$$X^2_{k-1} = \sum_{i=1}^{k} (n_i - n_i^*)/(n_i^*)$$

Step 5: Say 'p' = probability of this statistic, i.e., odds of our image having message veiled in it, as per the rules that n_i and n_{i*} are alike:

$$P = 1 - \frac{1}{\frac{k-1}{2^2}\Gamma\frac{(k-1)}{2}} \int_0^{X^2} \frac{-x}{e^2}\, x^{\left(\frac{k-1}{2}-1\right)}\, dx$$

Algorithm which escapes from chi square test: This algorithm uses the loop holes in the chi square test to screen secret in cover file. Here rows get embedded in such a way that the chi square value is controlled wherein the probability function which uses chi square value gives nearly zero as the probability of embedding. As the algorithm is mainly image dependent, that is, the embedding capacity varies from image to image. This is the major disadvantage but the advantage

is that the security is increased where the hacker cannot predict the information using chi square test. Algorithm and experimental results are included.

Embedding procedure: Following are the steps involved in embedding procedure:

Step 1: To embed information in the ith row of the cover image, first calculate the chi square value considering the rows from 1 to ith row in the partially embedded stego image and let this value is denoted by g

It infers that for embedding data in the first row we are actually calculating the chi square value for the first row of the clean cover image. Depending upon the chi square value we may or may not embed the data in the first row. Similarly for embedding data in the second row we should consider the rows of the partially embedded (if the message is embedded in the previous rows) stego image. This continues until we traverse all the rows of the cover image.

Step 2: If $0<g = 140$ then

Follow 0 bit LSB substitution for the ith row

Else if $140<g = 3000$ then

Follow 4 bit LSB substitution for the ith row

Else

Follow 0 bit LSB substitution for the ith row

Step 3: When 'i' reach the last row of the image, data embedding is completed. Now a key had to be generated to know which rows contain the 4 bit LSB substitution. Use run length coding technique to reduce the key length

Extraction procedure: Following are the steps involved in embedding procedure:

Step 1: Using the key, find out the rows in which message is embedded

Step 2: After knowing the rows in which secret data is embedded retrieve last four bits of all the pixels from top to bottom

Algorithm of IPMCS: The basic idea behind IPMCS algorithm has been discussed earlier in this paper and step by step procedure is included in this section.

Let M represent the n-bit secret message:

$$M = \{ m_i \mid 0 = i<n, m_i \varepsilon \{0,1\} \}$$

As in simple LSB substitution, assume that n-bit secret message has to be embedded into k-rightmost LSBs of the carrier i.e. Cover image:

Step 1: Total binary message is again discretised into (n/k) packets each of size equal to k bits. The decimal value of each packet ranges from 0 to. 2^{k-1}. Therefore we get 2^k different values named as symbols

Step 2: Find the probability of each such generated symbol in the message M'. The mathematical expression for M' is given by

$$M' = \{ mi' \mid 0 \leq i<n',\ mi' \varepsilon \{0,1,2,\ldots..,2\text{^}k-1\}$$

Where n' is the number of packets formed after discretizing the secret message, m_i' is the variable which holds the packet value

Step 3: Consider the k-rightmost LSBs of each pixel in the cover-image whose value again falls in the range (0, 2^{k-1}) which means that we get 2^k different values called as symbols. Now find the probability of each such symbol in the cover-image

$$m_i = \sum_{j=0}^{k-1} m_{i\text{x}k-j} \text{ x } 2^{k-1-j}$$

Step 4: Map most frequently (high probability) occurred symbol in the secret message M' to the most frequently occurred symbol in the cover image. Similarly map next highest probable symbol in the secret message to next highest probable symbol in the cover image, continue in the same way for other symbols also

By doing so finally we are actually mapping least frequently occurred symbol in the secret message to the least frequently occurred symbol in the cover image

Step 5: Mapping key is transmitted along with the stego image so that the intended receiver at the destination can restore the original information. The information in the key gives how symbols are mapped to one another; the same process can be done in the reverse manner to retrieve the original information

Algorithm for variable length bit embedding: In this section, a data hiding method called alternate variable length bit embedding is discussed.

Let k represent the rightmost LSBs that are substituted by the secret message

Let C represent the 8-bit grayscale cover-image of size $M_c \times N_c$ pixels

$$C = \{x_{ij} \mid 0 = i<Mc,\ 0 = j<Nc,\ x_{ij} \epsilon \{0, 1, \ldots, 255\}\}$$

In a constant bit embedding k is constant for all the pixels that are modified whereas in variable length coding k value is varied alternatively between 2 and 3 from pixel to pixel.

Total binary secret message is divided alternatively as 2 bits and 3bits and are represented as packet1 and packet2.This can be explained clearly in much simpler way as first 2'bits in message as packet1 and next 3'bits in message as packet2 and next 2'bits as packet3 and so on. After performing this operation we get a total of 2*n/5 discrete messages and the value of each such packet range from 0 to 2^{k-1}. We get 2^k different values called as symbols.'n' represents the size of the secret message in bits.

Step 1: Take the alternate pixels (2N-1), where N ranges from 1 to 32768 such that the pixels are (1, 3, 5, 7..., 65535). Embed these pixels with 2'bit LSB substitution in the cover image C

Step 2: Take the next alternate pixels 2N, where N ranges from 1 to 32768 such that the pixel numbers are (2,4,6,8,..., 65536). Embed these pixels with 3'bit LSB substitution in the cover image C

EXPERIMENTAL RESULTS

Here, discussed the analytical results observed for four cover files. It contains four standard gray scale covers, 'Cameraman', 'Lena', 'Penguin' along with 'Peppers', of dimension 256×256, are given in Fig. 1a-e, respectively. There are three sets of covert information. First set of secret data is the erratically formed data of 256×256×k bits; here k represents bits that get replaced. Here, first set of secret message, the value of k = 2 and for second set of secret message, the value of k = 3. For example, suppose that two bits of data is placed in last 2 LSBs of every cover pixel, then, secret information should be of size 256×256×2 = 131072 bits and suppose that three bits of data is placed in the last three LSBs, then the secret data should have the size 256×256×3 = 196608 bits. The number of pixels in an 256×256 gray scale image is 256×256 = 65536 pixels and for a third set of secret message, our proposed method alternate 2 and 3'bit LSB substitution consists of randomly generated message of (((256×256)/2)×2)+(((256×256)/2)×3) bits. In which half of the number of pixels are embedded with 2'b LSB substitution and another half is done so by 3'bit LSB substitution.

The second set consists of gray scale stego-images obtained by 2' bit replacement. For this LSB substitution the value of k is 2, so the length of the message is 131072 bits and the stegoimages of cameraman, Lena, penguin and peppers are given in Fig. 2-d, respectively. Third set consists of stego images is obtained by 3' bit swapping. For this LSB substitution the value of k is 3 and the length of the message is 196608 bits. The fourth set consists of gray scale stego-images for cameraman, Lena, peppers and penguin as shown in Fig. 3-d, respectively, obtained by variable length bit embedding (alternatively 2 and 3'bit) for this the value of k is 2 and 3. In this case the length of the message is 163840 bits and the stegoimages of cameraman, Lena, penguin and peppers are given in Fig. 4-d, respectively.

The stegnographic parameter (PSNR) after embedding the first, second and third secret data sets into first group of covers and are represented as second, third and fourth set of grayscale stego images, respectively.

After OPAP to subsequent second, third and fourth set of gray scale stego-images the resultant images are shown and analysed. Generally by applying OPAP, the PSNR value gets increased. Through applying our advised routine IPMCS before embedding, the PSNR value is further increased to a small extent, but the constraint is that the PSNR value after obtaining OPAP should be less than some threshold value for any particular type of simple LSB substitution. IPMCS algorithm for the stego images Camera man, Circuit Obama, Rice and Peppers are shown in Fig. 5a-e, respectively The results show that IPMCS method outperforms OPAP as in Table 1.

The chi-square value of the full image for 2'bit embedding is greater than 1'bit, 3'bit and 4'bit embedding unless for some exceptional images. Generally, the chi-square value is increased for 2'bit and then decreased for 3'bit and it is further decreased for 4'bit embedding and is given in Table 2.

Table 1: PSNR and capacity values of the proposed method for various images

	Proposed method (Escape chi square test)	
Image 256×256	PSNR	Capacity
Cameraman	37.4680	151553
Obama	34.6024	244737
Peppers	34.8366	258049
Rice	50.4130	7169
Circuit	38.4174	113665

Fig. 1(a-d): Cover images (a) Camera man (b) Lena (c) Penguin and (d) Peppers

Fig. 2(a-d): Stego-images for 2'bit LSB embedding (a) Camera man (b) Lena (c) Penguin and (d) Peppers

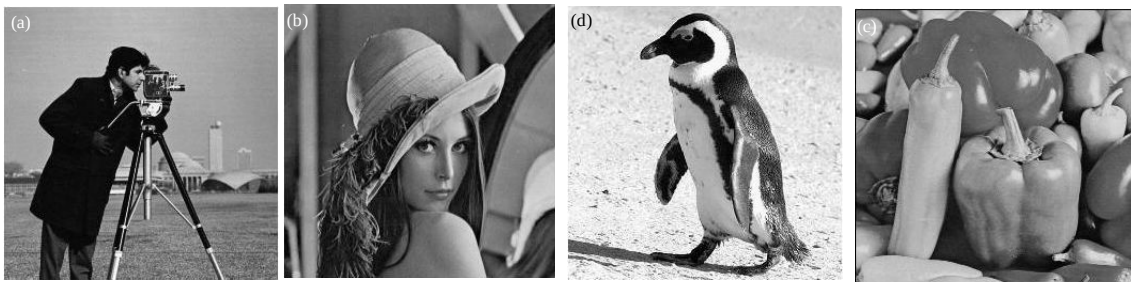

Fig. 3(a-d): Stego-images for 3'bit LSB embedding (a) Camera man (b) Lena (c) Penguin and (d) Peppers

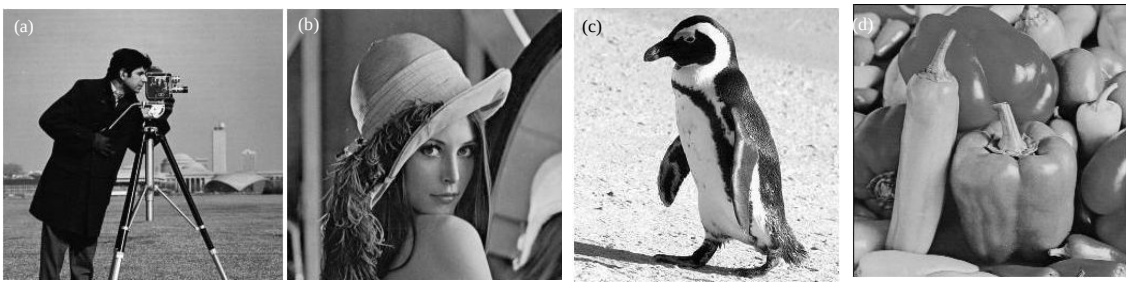

Fig. 4(a-d): Stego-images for variable length embedding (a) Camera man (b) Lena (c) Penguin and (d) Peppers

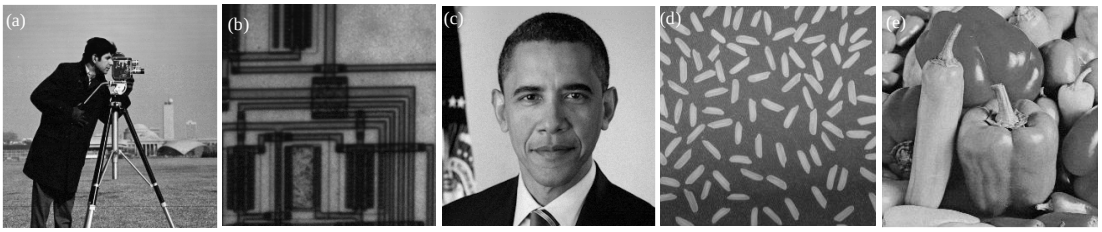

Fig. 5(a-e): Stego-image obtained for proposed algorithm (escape chi square test) (a) Cameraman (b) Circuit (c) Obama (d) Rice and (e) Peppers

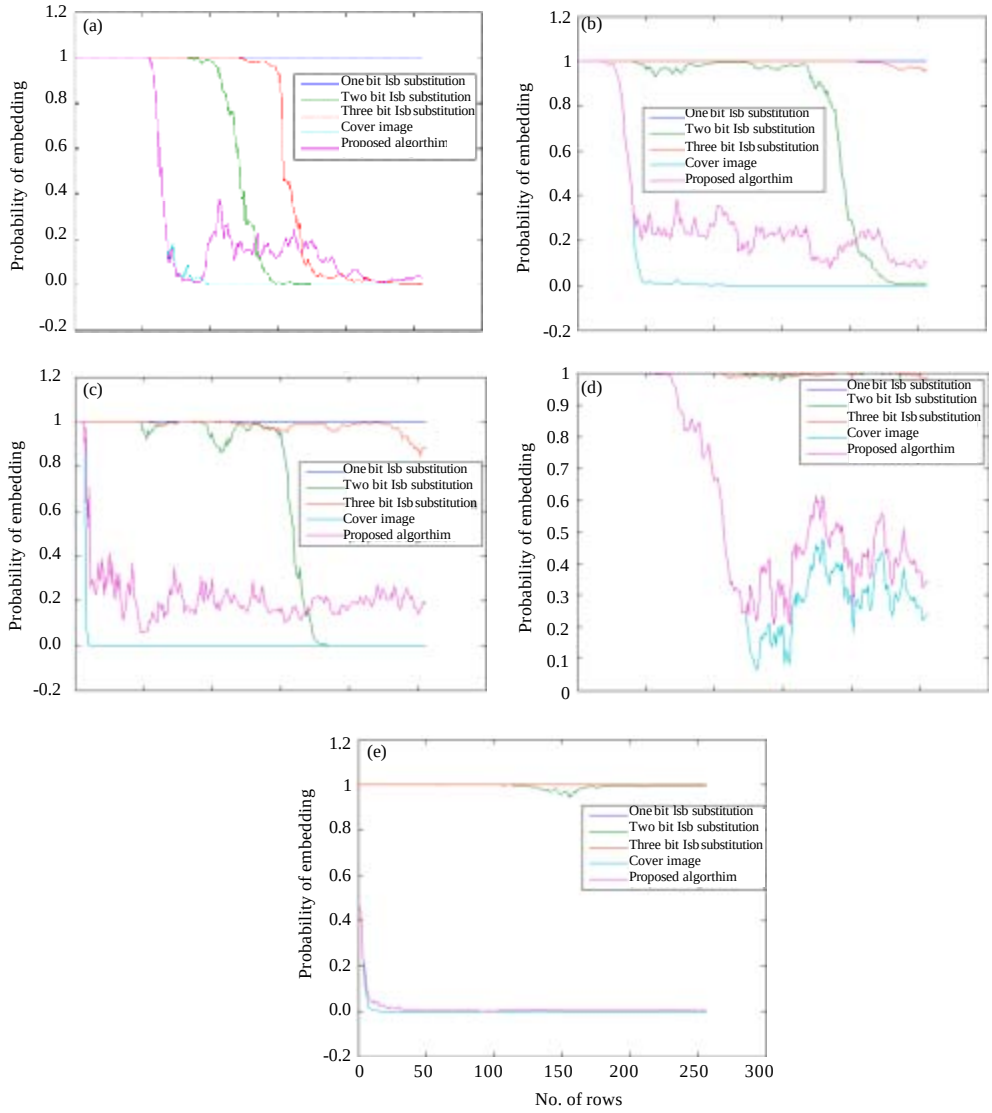

Fig. 6(a-e): Comparison of chisquare test output of the stego image (a) Camera man (b) Circuit (c) obama (d) Rice and (e) Peppers, with the other embedding techniques

Table 2: PSNR analysis for various bit embedding

Images	Parameter	1 bit embedding	2 bit embedding	3 bit embedding	Variable length embedding
Cameraman.tif	PSNR before OPAP	51.1338	44.1582	37.9335	40.0048
	PSNR after OPAP	51.1338	46.3721	40.7521	42.7065
	Chi square	57.4049	281.9506	175.1212	209.8113
Penguin.jpg	PSNR before OPAP	51.014	45.8652	37.4661	39.6037
	PSNR after OPAP	51.014	45.8031	39.7416	41.8035
	Chi square	845.5537	838.7794	590.1248	618.6267
Lenna.jpg	PSNR before OPAP	51.1557	44.1387	37.9121	39.9604
	PSNR after OPAP	51.1557	46.3561	40.7094	42.6839
	Chi square	52.8185	105.8594	92.9596	62.2078
Peppers.png	PSNR before OPAP	51.1578	44.1498	37.9066	39.9977
	PSNR after OPAP	51.1578	46.3237	40.6533	42.6778
	Chi square	43.3688	103.1791	83.5379	66.7095

Table 3: PSNR analysis of IPMCS algorithm

Cover Image	Secret data (Image)	PSNR after OPAP	PSNR after OPAP and IPMCS
Rice.png	Cameraman.tif	46.3469	46.3896
Rice.png	Penguin.jpg	46.37	46.3853
Rice.png	Peppers.png	46.3588	46.3655

Figure 3 shows all the stego-images obtained after applying the algorithm. All the images are 256×256 gray scale images. Fig. 6a-e shows the output of Chi square test corresponds to stegoimages of camera man, circuit, obama, rice and peppers respectively. In almost all the images probability of embedding is approximately one for single bit LSB substitution. For 2'bit and 3'bit LSB substitution chi square test gives a clue of the presence of secret data. For the proposed algorithm it drastically reduces the probability of embedding (nearly equal to zero). Hence this method is superior to other methods that are there in the literature. With the proposed algorithm the hacker finds it difficult to trace the data using chi square test. The only disadvantage of this method is embedding capacity depends on the image. Table 3 gives the information of PSNR and capacity for the stego images obtained after using proposed algorithm.

CONCLUSION

The algorithm which escapes from chi square test gives interesting results wherein the chi square test fails to detect the hidden data. The disadvantage with this algorithm is its dependence on the cover image but this can be overcome by using selected cover images. Extensive experiments show the effectiveness of the proposed method. In this paper, PSNR value is analyzed for constant bit LSB substitutions along with variable bit length LSB substitution (Table 2) and the main conclusion that can be drawn from alternate 2'bit and 3'bit embedding is that the hacker can easily identify the secret data in case of single bit substitution but difficult to trace out in the case of alternate 2'bit and 3'bit embedding. IPMCS can be applied only to those images whose PSNR value obtained after applying OPAP is below some threshold value. PSNR possessed by stego images is slightly superior over normal OPAP in addition to less computational intricacy.

REFERENCES

Al-Frajat, A.K., H.A. Jalab, Z.M. Kasirun, A.A. Zaidan and B.B. Zaidan, 2010. Hiding data in video file: An overview. J. Applied Sci., 10: 1644-1649.

Amirtharajan, R. and J.B.B. Rayappan, 2012a. An intelligent chaotic embedding approach to enhance stego-image quality. Inf. Sci., 193: 115-124.

Amirtharajan, R. and J.B.B. Rayappan, 2012b. Brownian motion of binary and gray-binary and gray bits in image for stego. J. Applied Sci., 12: 428-439.

Amirtharajan, R. and J.B.B. Rayappan, 2012c. Inverted pattern in inverted time domain for icon steganography. Inf. Technol. J., 11: 587-595.

Amirtharajan, R. and J.B.B. Rayappan, 2012d. Pixel authorized by pixel to trace with SFC on image to sabotage data mugger: A comparative study on PI stego. Res. J. Inf. Technol., 4: 124-139.

Amirtharajan, R., J. Qin and J.B.B. Rayappan, 2012. Random image steganography and steganalysis: Present status and future directions. Inf. Technol. J., 11: 566-576.

Amirtharajan, R., R. Akila and P. Deepikachowdavarapu, 2010. A comparitive analysis of image stegnography. Int. J. Comput. Appl., 2: 41-47.

Chan, C.K. and L.M. Cheng, 2004. Hiding data in images by simple LSB substitution. J. Pattern Recogn. Soc., 37: 469-474.

Cheddad, A., J. Condell, K. Curran and P.M. Kevitt, 2010. Digital image steganography: Survey and analysis of current methods. Signal Process., 90: 727-752.

Hansen, E.T., 2007. Analysis of the singular value decomposition in data hiding. M.Sc. Thesis, Iowa State University, Ames, Iowa.

Padmaa, M., Y. Venkataramani and R. Amirtharajan, 2011. Stego on 2^n: 1 Platform for users and embedding. Inf. Technol. J., 10: 1896-1907.

Provos, N. and P. Honeyman, 2003. Hide and seek: An introduction to steganography. IEEE Secur. Privacy, 1: 32-44.

Stefan, K. and A. Fabin, 2000. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, London, UK.

Thenmozhi, K., P. Praveenkumar, R. Amirtharajan, V. Prithiviraj, R. Varadarajan and J.B.B. Rayappan, 2012. OFDM+CDMA+Stego = Secure communication: A review. Res. J. Inf. Technol., 4: 31-46.

Wang, R.Z., C.F. Lin and J.C. Lin, 2001. Image hiding by optimal LSB substitution and genetic algorithm. Pattern Recognit., 34: 671-683.

Zanganeh, O. and S. Ibrahim, 2011. Adaptive image steganography based on optimal embedding and robust against chi-square attack. Inf. Technol. J., 10: 1285-1294.

Zhu, J., R.D. Wang, J. Li and D.Q. Yan, 2011. A huffman coding section-based steganography for AAC audio. Inf. Technol. J., 10: 1983-1988.