http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (7): 1285-1294, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1285.1294

Adaptive Image Steganography Based on Optimal Embedding and Robust Against Chi-square Attack

Omid Zanganeh and Subariah Ibrahim
Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Malaysia

Abstract: A real-life requirement motivated this case study of secure covert communication. Steganography is a technique used to transfer hidden information in an imperceptible manner. We proposed a novel approach of substitution technique of image steganography. The proposed method is flexible on size of secret message and allows us to embed a large amount of secret messages as well as maintaining good visual quality of stego-image. Using this method, message bits are embedded into uncertain and higher LSB layers, resulting in increased imperceptible and robustness of stego-image. Results show that the proposed algorithm provides large embedding capacity without losing the imperceptibility of the stego-image. The algorithm is also robust against Chi-square attack.

Key words: Information security, data hiding, image steganography, substitution technique, steganalysis

INTRODUCTION

Popularity of the Internet provides a great opportunity for transferring large amount of data in networks. However, it also increases the risk of illegal and unauthorized access to data during transmission. Mechanisms should be provided to protect against these attacks.

Steganography is a way for secret communication by using digital media to convey essential messages. The word steganography derived from Greek and it means cover writing. It is all about creating a form of secret communication between two parties and it is a complement to cryptography whose goal is to conceal the content of a message (Al-Jaber and Aloqily, 2003). Steganography uses a media like an image, video, audio or text file to hide information in such a way that it does not attract any attention and looks like an innocent medium (Hmood *et al*., 2010a).

The first use of steganography was in military and government for secret communication but nowadays, steganography methods become widely used for many purposes. Researchers propose new methods in this area as well as enhancing the previous approaches to improve the steganography applications (Al-Frajat *et al*., 2010).

There are lots of methods used in image steganography. However, they have their own weaknesses and strengths. Since Least Significant Bit (LSB) insertion method is one of the simplest data hiding techniques, it has long been a focus for researchers to propose attacking methods and they are called either steganalytic or steganalysis attacks. It is proved that sometimes simple LSB method is not secure at all because some harmful statistics can be exploited to reveal the existence of the secret data (Lee *et al*., 2009). In this study most of the effort is done to get a better imperceptibility of the stego-image without losing the embedding capacity.

RELATED WORK

Simple LSB substitution: The word LSB stands for Least Significant Bit. This method is simple and easy to implement in Steganography area. This method actually substitutes the LSBs of cover image with secret bits sequentially. In order to hide messages by this approach at least one bit is stored in each pixel of cover image. For example by using 8-bit gray scale image format with the size of 512×512 can embed 262, 144 bits (32,768 bytes or pixels) (Hmood *et al*., 2010b). By embedding this amount of data both stego-image and its respective cover image look the same since human eye cannot distinguish little changes in the value of pixels. Embedding more than one bit in each pixel by using pixels in the edge area will make more change in high frequency areas so it can still be undetectable by human eye as well (Lee and Chen, 1999). By considering the size of cover image pixels there is no limitation on embedding rate in

Corresponding Author: Omid Zanganeh, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Malaysia

this method but the more secret bits we embed, the less imperceptibility of stego-image is obtained. So researchers proposed several methods to improve the weakness of this method.

The optimal LSB method: The simple LSB method can be modified to improve the quality of stego-image. The algorithms of such improved schemes are still based on simple LSB method. In this section one of the improved methods, called Optimal LSB which applies Optimal Pixel Adjustment Process (OPA) to improve the stego-image quality is presented. Three candidates are picked out for the pixels value and compared to see which one has the closest value to the original pixel value with the secret data embedded in. The best candidate is then called the optimal pixel and used to conceal the secret data (Chan and Cheng, 2004).

The embedding process is described by Wu and Hwang (2007) as follows:

- Let p_i be the original pixel value and k bit (s) be secret data to be embedded
- Embed k bit (s) of secret data into p_i by using the LSBs method. The stego-image p_i can then be obtained
- Generate another two pixel values by adjusting the (k + 1)th bit of p_i. Therefore, p_- and p_+ can be calculated as follows:

$$(p'_+, p'_-) = \begin{cases} p'_+ = p'_i + 2^k \\ p'_- = p'_i - 2^k \end{cases}$$

obviously, the hidden data in p_- and p_+ are identical to p_i because the last k bits of them are the same.

The best approximation to the original pixel value, p_i, (i.e., the optimal candidate) is found by the following formula:

$$p''_i = \begin{cases} p'_i & \text{if} \left|p_i - p'_i\right| \le \left|p_i - p'_-\right| \le \left|p_i - p'_+\right| \\ p'_+ & \text{if} \left|p_i - p'_+\right| \le \left|p_i - p'_i\right| \le \left|p_i - p'_-\right| \\ p'_- & \text{if} \left|p_i - p'_-\right| \le \left|p_i - p'_i\right| \le \left|p_i - p'_+\right| \end{cases}$$

Finally, all the optimal candidates for p''_i replace the original pixel values p_i and the embedding algorithm come to its end.

In order to make the algorithm faster and using low complicated calculation, the pseudocode of the algorithm is given below (Tseng *et al*., 2008). The inputs of the algorithm are secret image S and cover image C and the output is stego-image C`. M and N are the height and the width and i and j are the coordinate of the cover image C.

```
Inputs: Transformed Secret ImageS`, Cover Image C
Output: Stego-image C`
{
For j←0 to M-1
Do for i←0 to N-1
Do D [j] [i] = C [j] [i] mod $2^k$-S [j] [i]
     If (D [j] [i] >$2^{k-1}$
          Then if (C [ j] [i] <255-$2^{k-1}$
               Then C` [j] [i] = C [j] [i]-D [j] []
          Else if (D [j] [i]<-$2^{k-1}$
Else if (D [j] [i]<-$2^{k-1}$
     Then if (C [ j] i] <-$2^{k-1}$
          Then C` [j] [i] = C [j] i]-D [j] [i]-$2^k$
     Else C` [j] [i] = C [j] [i]-D [j] [i]
Else C` [j] [i] = C [j] [i]-D [j] [i]
}
```

The secret image S is rearranged to become S′by the same size as the cover image C but with the smaller bit plane than the k bit plane of cover image (k is embedded bit plane). D [j] [i] is the value that subtracts the value of secret image S′ (S′ [j] [i]) from the same coordinate of cover image C (C [j] [i]). For instance, if C [j] [i] = 50_{10}= 110010_2, S′ ([j] [I]) = 7_{10} = 111_2 k = 3, then D ([j] [i]) = 50 mod 2^3-7 = -5 is smaller than -2^2. Due to that, the value of C′ ([j] [i]). So, the difference will be smaller than embedding like the simple LSB substitution. The difference between cover pixel value and stego-image value by OPA algorithm is 3 (50-47 = 110010_2-101111_2 = 3_{10})which is smaller than the difference after embedding by the simple LSB which is 5 (50-45 = 110010_2-110111_2 = 5_{10}). This smaller difference will enhance the imperceptibility of the stego-image by reducing the value of mean square error.

Wang *et al.* (2001) optimal LSB substitution method: Wang *et al.* (2001) approach is based on simple LSB substitution and Genetic Algorithm (GA). The process of Wang *et al.* (2001) method is shown in Fig. 1. There are two differences between simple LSB and Wang *et al.* (2001) method. The first one is that in the LSB substitution method interceptors can extract the secret data (secret image) from stego-image easily, because the hidden secret image is regularly distributed in stego-image. To eliminate this disadvantage, Wang *et al.* (2001) method uses a transformation function which is described below, to modify each location of decomposed image SI to a new location in the meaningless image ESI. Assume pixel locations in ESI are numbered sequentially from 0 to p-1. The transform function has been used before replacing C′ by ESI*.

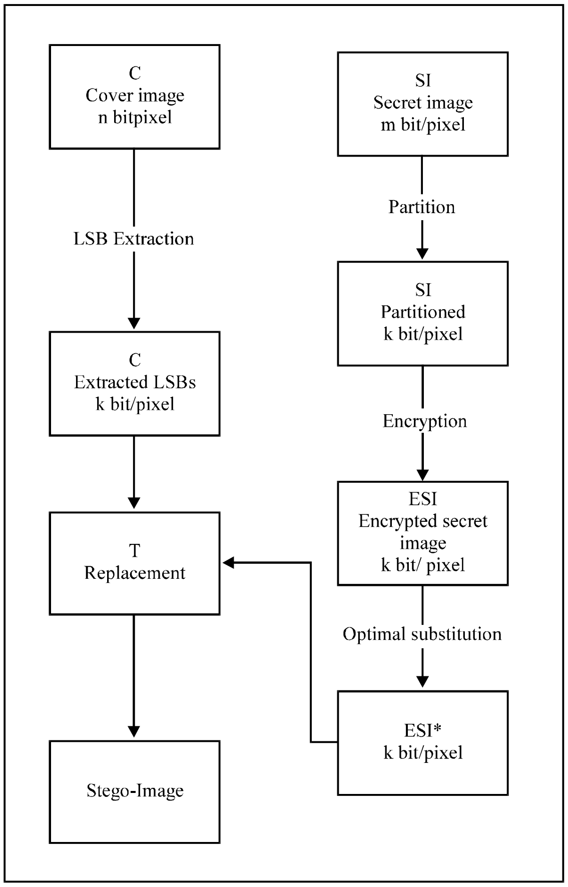


Fig. 1: Block diagram of Wang *et al.* (2001) optimal substitution method

The transform function used is f (x) = (k0 + k1 ×x) mod p and gcd (k1, p) = 1 where k0 and k1 are the key constants dcd (,) means greatest common divisor. k0 and k1 are needed for recovering the hidden secret image from ESI′. Illegal interceptors will not be able to gain the secret data without knowing these two keys and the cipher process.

Second significant difference between Wang *et al.* (2001) method and simple LSB replacement is that Wang's replacement is optimal substitution instead of simple substitution. To achieve this goal in Wang's method, a substitution matrix S = $\{S_{ij}\}$ is used to convert each pixel value i of ESI′ to another value j in ESI^* where $0 \le i < 2^{k\text{-}i}$ and $0 \le i < 2^{k\text{-}i}$. In the matrix S there is only one element in each row and column that has the value 1. Thus, there are $(2^k)!$ possible substitution matrices and only the best one will be chosen which makes the least distortion between secret image ESI′ and host image C. A simple example of matrix S for a 4×4 secret image with k = 2 is presented in Fig. 2.

0	1	3	2
2	3	0	0
2	0	1	3
3	1	2	2

ESI'

$$S = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

3	1	2	0
0	2	3	3
0	3	1	2
3	1	0	0

ESI*

Fig. 2: An Example of optimal substitution process from ESI′ to ESI^* by Matrix S

According to optimal substitution matrix S, the pixel value 0 of ESI′ would be replaced by value 2 in ESI^* and the pixel value of 3 in ESI^* should be replaced by 0 in $ESI^*$and so on. The substitution matrices must be saved for sending to the receiver. The receiver needs the matrices for extracting the embedded information from the stego-image. Without these information receiver will not be able to be aware of the hidden information.

Finding the best substitution matrix takes a lot of time if K is a large number. For example if K = 3 there are $(2^3)!$ = 40, 320 different matrices and to find the best one we should check them all and it is too time consuming. So Wang *et al.* (2001) method used GA to find the best matrix to reduce searching time.

Wu *et al.* (2004) global and local optimal LSB substitution: In Wang *et al.* (2001) method, optimal substitution matrix is derived for the whole image but if we use the same mechanism for each block of the image, the PSNR value will be increased. That is the significant difference between Wang *et al.* (2001) and Wu *et al.* (2004) method. In fact Wang *et al.* (2001) global transformation idea is not beneficial for the whole image (Wu *et al.*, 2004). In Wu *et al.* (2004) method, the transformation matrix is calculated according to block characteristics of cover image. So the quality of stego-image would be better.

The block diagram of Wu *et al.* (2004) method is presented in Fig. 3. In their method the decomposed image S′ is divided into n blocks, $\{ES'_0, ES'_1, ...ES'_{n-1}\}$. C′also will be divided into $\{C'_0, C'_1, ... C'_{n-1}\}$. The next step is to search for the best match between ES'_i and C_j. Most similar ones between ES'_i and C_j will be selected as a match pair. GA is used to perform this step.

Furthermore, Wu *et al.* (2004) proposed two different strategies. The first strategy is the global optimal substitution strategy and the second one is the local optimal substitution strategy. The first one uses the same optimal substitution matrix for all blocks and also one substitution matrix for blocks mapping. Similar to the Wang *et al.* (2001) approach, the matrices must be saved to be sent to the receiver for extraction process. So the global optimal strategy needs fewer data to be recorded in

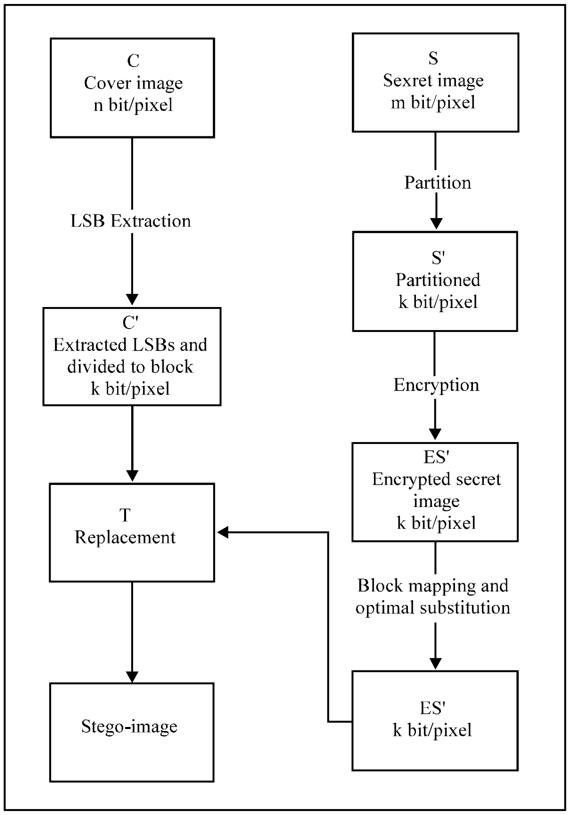


Fig. 3: Block diagram of Wu *et al.* (2004) method

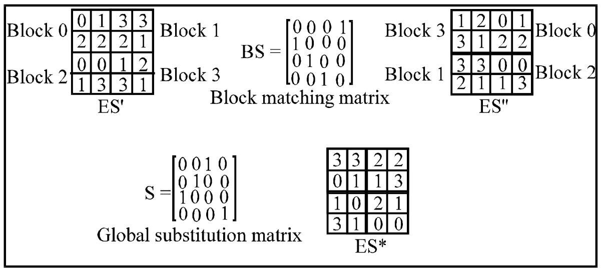


Fig. 4: An example of Global Optimal Substitution method

comparison with their second strategy. The second one uses different substitution matrices for blocks and more matrices are required so more data need to be recorded although a better stego-image quality is provided. The first strategy which is based on global optimal substitution is called global method and the second strategy which is based on local optimal substitution matrix is called local method.

A simple example of the global method is presented in Fig. 4. As shown, after finding the best match between ES^*_i and C_j, only one substitution matrix has been used for the whole image. The same strategy is used for each block of cover image in local method.

Figure 5 shows an example of local method and we can see that each block has its own substitution matrix and also a matrix for block mapping.

THE PROPOSED METHOD

As mentioned earlier many algorithms are proposed by researchers to solve the problems of simple LSB and increase the imperceptibility of stego-image (Chan and Cheng, 2004; Wang *et al.*, 2001; Wu *et al.*, 2004, 2005; Zhang and Wang, 2005). But the proposed method (Adaptive Optimal Embedding) has a higher imperceptibility of the stego-image by using more characteristics of cover image. The proposed approach searches a pixel to find a match between original pixel bits and secret bits. In conventional LSB method, the hidden information is embedded sequentially and starts to embed from the first LSBs of each pixel. On the other hand, the embedding position in our method depends on pixel value. If there is a match between secret bits and original cover pixel's bits, there is no need to embed and we just have to identify the starting bits of found match. The starting bits where the secret bits are embedded will be combined together to form a stream of bits. This stream of bits is used as a stego-key that needs to be communicated to the receiver for extracting the secret message.

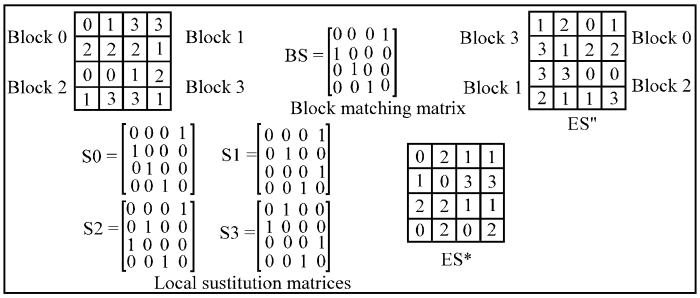


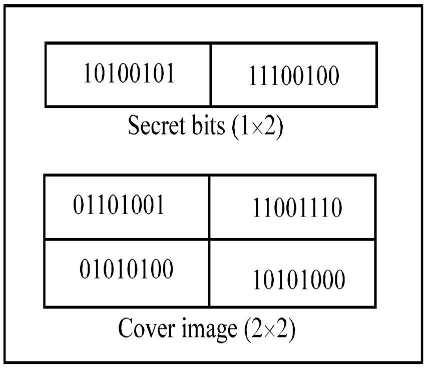
Fig. 5: An example of Local Optimal Substitution method

Experiment shows that the method produces no image distortion and the imperceptibility increases significantly.

For embedding process, first we have to know the embedding rate (number of bits we embed per pixel). Since the 8-bit gray-scale image is selected as cover image, the embedding rate is simply obtained by dividing number of secret bits to number of pixels. For example, we have 1, 048, 576 secret bits and size of cover image is 512×512. So by dividing 1, 048, 576 to 262, 144 the embedding rate would be four which means four bits should be embedded in each pixel of cover image. Using sample pixels of cover image and sample secret bits is useful to explain the process of embedding in detail which is given in Fig. 6. Also the respective flowchart of embedding process is presented in Fig. 7.

The embedding process starts from the leftmost pixel of the first row and moving to the right of the same row before continuing to the subsequent rows of the image. For example, the first secret bits to embed are 1010 and the cover pixel value is 01101001. We can notice that the four bits starting from third LSB are the same as the secret bits. Therefore no embedment is required in this case. Hence, we do not cause any image distortion because we did not change any original bits. We need to identify that the four bits of this pixel are secret bits and inform the receiver for extraction process. To embed the next secret bits (0101) in respective cover pixel, again we need to search for a match in the corresponding pixel. Since there is no match between secret bits and cover bits, we used the four LSBs of cover pixel to embed the secret bits by using OPA algorithm. The stego-key is generated by determining the first bit position where the leftmost bit of secret bits is embedded. As shown in Fig. 8, the stego-key for the sample secret bits is Fig. 8 depicts where the secret bits are embedded and shows how the stego-key is derived.

The receiver extracts the secret bits by using the stego-key and the stego-image. According to Figure 9, first bits of stego-image's pixel are and the first two bits of stego-key are . It means that our secret bits are the four bits after second LSB of stego-pixel. So for extracting the secret data we take four bits of the modified pixel, starting from third LSB which are . The second two bits of stego-key are and this means extract the bits from the second pixel by starting first LSB which are . Then repeat the extracting process for the rest of the pixels. The extraction flowchart is given in Fig. 9.



10100101	11100100

Secret bits (1×2)

01101001	11001110
01010100	10101000

Cover image (2×2)

Fig. 6: An example of secret bits and cover image

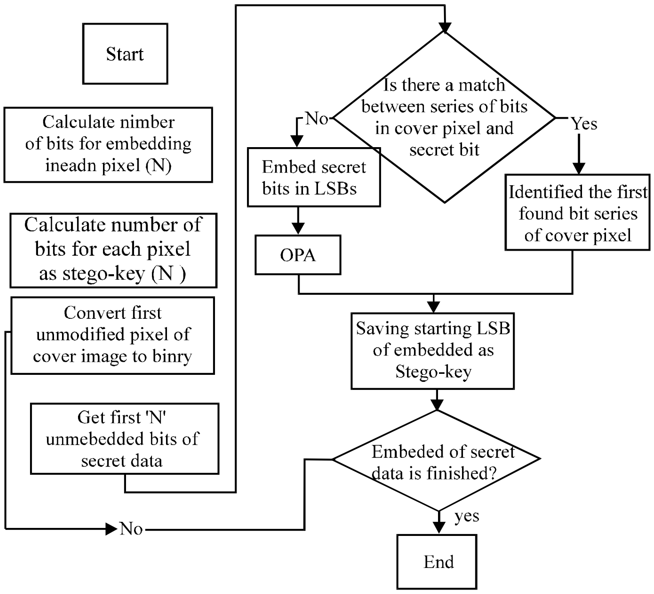

Start
Calculate nimber of bits for embedding ineadn pixel (N)
Calculate number of bits for each pixel as stego-key (N)
Convert first unmodified pixel of cover image to binry
Get first 'N' unmebedded bits of secret data
Is there a match between series of bits in cover pixel and secret bit
No
Yes
Embed secret bits in LSBs
OPA
Identified the first found bit series of cover pixel
Saving starting LSB of embedded as Stego-key
Embeded of secret data is finished?
No
yes
End

Fig. 7: The desired embedding flowchart of Adaptive Optimal Embedding

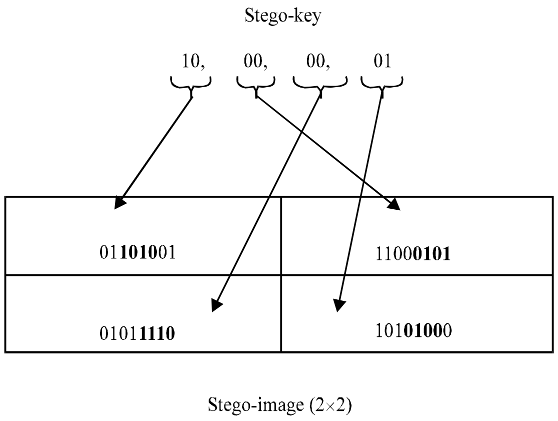


Fig. 8: The stego-image and stego-key after embedding phase

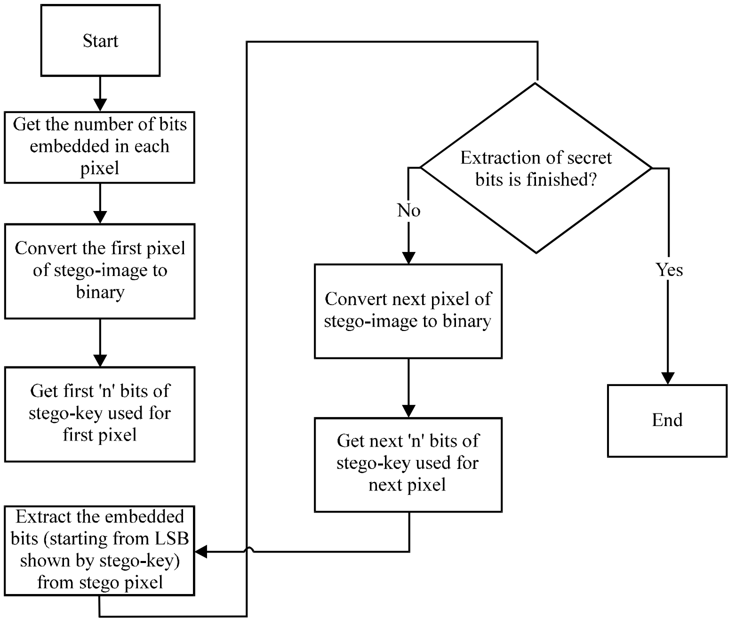

Start
Get the number of bits embedded in each pixel
Convert the first pixel of stego-image to binary
Get first 'n' bits of stego-key used for first pixel
Extract the embedded bits (starting from LSB shown by stego-key) from stego pixel
Extraction of secret bits is finished?
No
Yes
Convert next pixel of stego-image to binary
Get next 'n' bits of stego-key used for next pixel
End

Fig. 9: The desired extraction flowchart of Adaptive Optimal Embedding

EXPERIMENTAL RESULTS

Four methods that provide high embedding capacity were described in Section two. These four methods are Simple LSB, Optimal LSB, Wang *et al.*'s method and Wu *et al.*'s method. This section presents the comparison of these methods and the proposed method. The images used as cover and secret images are 8 bit grayscale. The standard image Lena with the size of which is shown in Fig. 10 is used as cover image. Figure 11 shows three secret images Barbara, Peppers and Airplane-F16 which are used as secret images. The size of each secret image is. All standard images are collected from (Gonzalez and Woods, 2008).

Fig. 10: The cover image Lena with size of 512×512

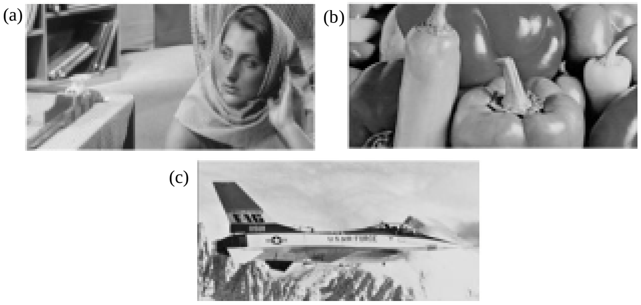


Fig. 11: Secret Images with the size of 512×256. (a): Barbara (b): Peppers (c): Airplane F16

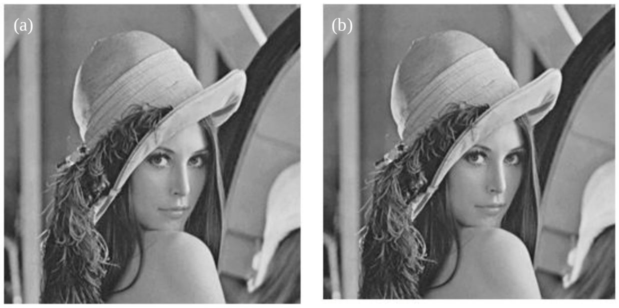


Fig. 12: The Stego-image Lena, embedded with secret image Barbara. In (a) n = 1 and (b) n = 2

Imperceptibility of proposed method: To evaluate the imperceptibility of the stego-image after embedding and also to compare with previous works, PSNR (Peak Signal-to-Noise Ratio) metric is used. As we know the higher stego-image quality, the more imperceptibility of the hidden message. The PSNR is the very first metric which can judge imperceptibility very well. The formula is as follows:

$$\text{PSNR} = 10\log_{10}\frac{255^2}{\text{MSE}}\,\text{dB}$$

Whereby:

$$\text{MSE} = \left(\frac{1}{M\times N}\right)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(P(x,y) - P'(x,y))^2$$

where, M and N represent the image size. In the formula, P (x ,y) stands for the original pixel value and P′ (x ,y) represents the pixel value in position (x ,y) with the secret data already hidden in. A greater PSNR value means a lower degree of image distortion after embedding process of the secret data. For example, given a gray scale image as the cover image to hide secret data in, it is hard for any human being to perceive any difference between the cover image and the stego-image if the PNSR value of the stego-image goes beyond 36 dB (Wu and Hwang, 2007).

Table 1 shows the result of embedding 1,048,576 bits (four bits per pixel) in cover image Lena by Simple LSB, Optimal Pixel Adjustment, Wang *et al.* (2001) method, Wu *et al.* (2004) global method, Wu *et al.* (2004) local method and Adaptive Optimal Embedding. The results show that the value of PSNR in Wu *et al.* (2004) local method is significantly better than Wang *et al.* (2001) method because more attributes of block characteristics were explored. However, the best PSNR was obtained by Adaptive Optimal Embedding and then by OPA. In Wang *et al.* (2001) and Wu *et al.* (2004) method the receiver needs block matching matrix and optimal substitution matrices for extraction. In Adaptive Optimal Embedding algorithm n is the number of bits used as stego-key for each pixel. We assume 8 bits of cover pixel's are presented as $C_1C_2C_3C_4C_5C_6C_7C_8$. If n = 1 the algorithm checks the first two layers of cover pixel ($C_5C_6C_7$ and $C_4C_5C_6C_7$) to find the match for secret bits which needs one bit to identify these two layers. When n = 2 the first four layers are checked ($C_5C_6C_7C_8$, $C_4C_5C_6C_7$, $C_2C_3C_4C_5$) Fig. 12.

Table 1: The result of embedding secret images Barbara, Peppers and Airplane F-16 into cover image Lena

Secret Image	Barbara	Peppers	Airplane F-16
Simple LSB	32.2974	32.3681	31.8779
OPA	34.7876	34.8045	34.8114
Wang *et al.* (2001) method	32.8824	32.5453	33.0296
Wu *et al.* (2004)			
Global method	32.9873	32.6748	33.1669
Local method	34.3948	34.3637	34.5349
Adaptive Optimal Embedding			
n=1	35.1540	35.1636	35.0538
n=2	35.7823	35.6984	35.7227

Table 2: The result of embedding different amount of secret data with different ‘n’ in cover image Lena

			Adaptive Optimal Embedding			Difference of PSNR value between OPA and Adaptive Optimal Embedding		
	Simple LSB	OPA	n = 1	n = 2	n = 3	n = 1	n = 2	n = 3
524288 bits(2 bit/pixel)	43.8019	46.3827	47.4458	50.1369	53.8557	1.0631	3.7542	7.473
786432 bits(3 bit/pixel)	38.0825	40.7139	41.2131	42.6614	44.0779	0.4992	1.9475	3.364
1048576 bits(4 bit/pixel)	31.8779	34.8114	35.0538	35.7227	36.0743	0.2424	0.9113	1.2629
Storage Overhead (bit)	0	0	262144	524288	786432			

As shown in Table 1, there is a significant improvement of PSNR value by Adaptive Optimal Embedding method. When n = 2 the probability of finding the desired match of secret bits in cover bits is two times higher than when n = 1, so the quality of stego-image is significantly better by applying n = 2. However, by using one bit as stego-key for each pixel (n = 1) the results obtained by Adaptive Optimal Embedding is still better than other methods. It is noteworthy that in Wang *et al.* (2001) and Wu *et al.* (2004) methods if embedding rate is more than two, it takes a lot of time to find the best, respective optimal substitution matrices so they use genetic algorithm to increase the efficiency of finding these matrices but in Adaptive Optimal Embedding since the embedding process is simple and even sometimes there is no need to embed any bits (in situations that match is found) the time of applying our algorithm is nearly the same as simple LSB and OPA.

Experiment was also conducted to evaluate the performance of different payloads by embedding two, three and four bits per pixel. As shown in Table 2, the results obtained by Adaptive Optimal Embedding method are still better than the OPA algorithm. By embedding two bits per pixel, the difference between Adaptive Optimal Embedding and OPA is more than when we embed three bits per pixel with the same n. The reason is when the embedding rate is two, the probability of finding the match is higher than when the embedding rate is three, because we are looking for a specific series of two bits (as secret bit) in population of four different values represented by two bits. So the probability of finding is 1/4. But when we embed three bits, we look for a series of three bits in eight values (represented by three bits) and the probability of finding this series is 1/8. The reason of smaller difference in PSNR value between Adaptive Optimal Embedding and other methods by embedding four bits and three bits is the same.

As discussed earlier, the best PSNR is obtained by our method and then by OPA. Although in our method the size of stego-key is large, the experiments showed most of the values of stego-key in this method are the same and they are zero. The probability of finding a 4-bit match between a bit stream is 1/16 because we are looking for four special bits in a population of four bits which can show sixteen different values. For instance we look for 1101 in four bits and these four bits can show a value of 0000 to 1111. In the proposed algorithm, the more bits we use for searching in cover pixel (n), the more chance to find a match. But since the chance of finding the match is not much, most of the bits are embedded in LSB. Therefore, by using a compression algorithm like Huffman which is so suitable here, the size of stego-key can be decreased significantly.

Security of proposed method: As the opponent side to steganography, the goal of steganalysis is to detect whether hidden message exists in a media or not and afterward, detect the hidden data if there is. In the past decades, many steganalysis methods have been proposed by researchers to achieve this goal (Xia *et al.*, 2009). One of the steganalysis methods is called Chi-square attack which is used to prove the security of proposed method.

The idea of Chi-square attack is simple and it is all based on the probability of how zeros and ones are distributed all over the medium (stego-image). If the bits in secret messages are generated randomly, then the probability that half of the number of bits is either zero or one is fifty percent (Westfeld and Pfitzman, 1999). This fact makes the idea of $\chi 2$ test (Chi-square) clearer.

Let $X^{128\times1}$ and $Y^{128\times1}$ be two vectors such that x_k = frequentlcy (2k) and y_k = frequentlcy (2k + 1). Initially, every entry in $0 \le k \le 127$ x and y is set to 0. Then the algorithm counts the gray-scale values in the test image and increments the corresponding entry in X or Y. The erotical expected frequency of gray-scale values of 2k and 2k + 1 is:

$$z_k = \frac{x_k + y_k}{2}$$

Now suppose that there are m categories. In the case of 8-bit grayscale images, there are 128 categories (256 grayvalues/2) which means also 128 PoVs. So to check all the possible POVs, 128 categories should be taking into consideration by the given Equation below. The Chi-square statistic, with m-1 degrees of freedom, is calculated by:

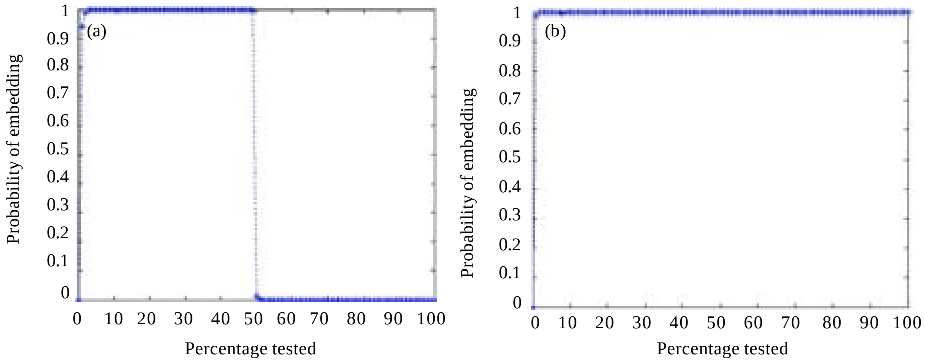


Fig. 13: Results for Chi-Square attack on stego-image Lena, embedded by simple LSB and tested on (a): fifty percent (131, 072 bits) and (b): 100% (524, 288 bits) of the image

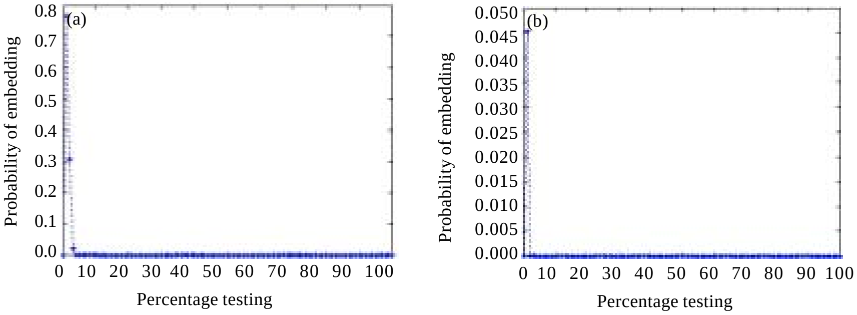


Fig. 14: Results for Chi-square attack on stego-image Lena, embedded by Adaptive Optimal Embedding and tested on 1, 048, 576 bits, (four random bits embedded in per pixel (a): n =1 (b): n =2)

$$\chi^2_{m-1} = \sum_{i=0}^{127} \frac{(x_i - z_i)^2}{z_i} \text{where} z_i = \frac{x_i + y_i}{2}$$

The next step of Chi-square method is calculating the probability of embedded information by integrating the density function with x^2_{k-1} as its upper limit:

$$p = 1 - \frac{1}{2^{\frac{m-1}{2}} \Gamma(\frac{m-1}{2})} \int_0^{\chi^2_{m-1}} e^{-\frac{u}{2}} u^{\frac{k-1}{2}-1} du$$

This probability of embedding is the probability of x^2_{m-1}under the condition that $x_i = z_i$ for all i in Chi-square statistic Equation (Stanley, 2005).

As described in (Westfeld and Pfitzman, 1999), once the embedding is done the more bits embedded, the probability is stronger and more reliable. This means that when 100% of LSBs are embedded by random bits, it is expected to see the best result of Chi-square attack saying the probability of embedding is one for the whole 100% of the stego-image.

Figure 13a shows the result of embedded information in fifty percent of the cover image when in Fig. 13b, more than one bit is embedded in all pixels of the cover image. In both of them, Simple LSB algorithm is applied for embedding process. By taking a simple look at the Fig. 13, it is noticed that the test works out and attacks the very existence of secret data for these specific payloads. The payload of 1231, 072 bits is exactly using fifty percent of the stego-image by embedding one bit per pixel and it has been detected that the same probability has been embedded with some secret bits. The payload of 104,576 embeds more than one bit in each respective pixel of the cover image and the attack is still reliable since Chi-square detects 100% of the zstego-image has the probability of embedding equal to one.

As the same attack used for the stego-images embedded by Adaptive Optimal Embedding, the shapes of diagrams changed. The Chi-square attack is tested on different amount of data by embedding two, three and four bits per pixel. The test is done by applying two different n (n = 1 and n = 2). So finally, we show the result of embedding by proposed approach is strongly robust against Chi-square attack. As results show in Fig. 14, by embedding more than one bit in 100% of the cover image, this method is still reliable against Chi-square attack since there is no change in Chi-square curve that means there is no way to know the percentage of the embed data in cover image.

There is one point left which worth describing and that is the detected probability is more reliable when a kind of straight line can be seen which is steady or fixed in some area and it drops in some particular points and then goes on the same steady way. But, there is only one change between zero percent and less than five percent of the curves, in Fig. 14. Although there is one change in the Chi-square curve in Fig. 14 but the detected probability in this area for first curve is near to one and for second curve is almost zero (0.05). Also 4 bits is embedded in all pixel of the cover image, so if the Chi-square attack can detect the embedded information, the value of probability of embedding must be one for the whole stego-image.

CONCLUSION

A new approach is presented to resolve the most important problem of image steganography which is imperceptibility of the stego-image without losing the embedding capacity. To solve this problem, the proposed method identifies a certain bit stream of the cover image as secret bits. The image distortion is decreased by using OPA technique when bits in the cover image's pixel are not same as secret bits. To show the robustness of this approach, Chi-square attack was tested on different stego-images which have been embedded with different payloads. In order to enhance the effectiveness of the proposed method, it is recommended to apply some algorithms either to decrease the size of stego-key such as Huffman or embed the stego-key in another cover image.

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.

Al-Jaber, A. and I. Aloqily, 2003. High quality steganography model with attacks detection. Inform. Technol. J., 2: 116-127.

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

Gonzalez, R.C. and R.E. Woods, 2008. Digital Image Processing. 3rd Edn., Prentice Hall, New Jersey, USA., ISBN-13: 9780131687288, pp: 954.

Hmood, A.K., H.A. Jalab, Z.M. Kasirun, B.B. Zaidan and A.A. Zaidan, 2010a. On the Capacity and security of steganography approaches: An overview. J. Applied Sci., 10: 1825-1833.

Hmood, A.K., B.B. Zaidan, A.A. Zaidan and H.A. Jalab, 2010b. An overview on hiding information technique in images. J. Applied Sci., 10: 2094-2100.

Lee, Y.K. and L.H. Chen, 1999. An adaptive image steganographic model based on minimum-error LSB replacement. Proceedings of the 9th National Conference on Information Security, May 14-15, Taichung, Taiwan, pp: 8-15.

Lee, Y.K., G. Bell, S.Y. Huang, R.Z. Wang and S.J. Shyu, 2009. An advanced least-significant-bit embedding scheme for steganographic encoding. Adv. Image ideo Technol., 5414: 349-360.

Stanley, C.A., 2005. Pairs of values and the chi-squared attack. Master's Thesis, Department of Mathematics, Iowa State University.

Tseng, L.Y., Y.K. Chan, Y.A. Ho and Y.P. Chu, 2008. Image hiding with an improved genetic algorithm and an optimal pixel adjustment process. Proceedings of the 8th International Conference on Intelligent Systems Design and Applications, Nov. 26-28, Kaohsiung City, Taiwan, pp: 320-325.

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

Westfeld, A. and A. Pfitzman, 1999. Attacks on steganographic systems breaking the steganographic utilities ezstego, jsteg, steganos and s-tools-and some lessons learned. LNCS., 1768: 61-67.

Wu, H.C., N.I. Wu, C.S. Tsai and M.S. Hwang, 2005. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEE Proc. Vision Image Signal, 152: 611-615.

Wu, N.I. and M.S. Hwang, 2007. Data hiding: Current status and key issues. Int. J. Network Sec., 4: 1-9.

Wu, M.N., M.H. Lin and C.C. Chang, 2004. A LSB substitution oriented image hiding strategy using genetic algorithms. Content Comput., 3309: 219-229.

Xia, Z., X. Sun, J. Qin and C. Niu, 2009. Feature selection for image steganalysis using hybrid genetic algorithm. Inform. Technol. J., 8: 811-820.

Zhang, X. and S. Wang, 2005. Steganography using multiple-base notational system and human vision sensitivity. Signal Process. Lett., 12: 67-70.