Image Steganography Based on Adaptive Optimal Embedding

Omid Zanganeh
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
Johor, Malaysia
o.zanganeh@gmail.com

Subariah Ibrahim
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
Johor, Malaysia
subariah@utm.my

***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 completely flexible on size of secret message bits 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.**

Keywords-Data Hiding; Image Steganography; Substitution Techniques

I. INTRODUCTION

Popularity of the Internet provides a great opportunity to transfer large amounts of data in networks. However, it also increases the risk of illegal and unauthorized access to deal with the content, while the data is transferred. Mechanisms should be prepared to provide protection against attacks and make a secure transfer. In the last ten years, several methods have been proposed by researchers to develop an environment for transferring important information.

Steganography is a way for secret communication by using digital media to convey essential messages. The word “Steganography” derives from Greek and it means "cover writing". Steganography is all about creating a form of secret communication between two parties and it is a complement of cryptography that whose goal is to conceal the content of a message. Steganography uses a media like an image, video, audio or text file to hide some information inside it in such a way that it does not attract any attention and looks like an innocent medium [1].

There are lots of algorithms used in image steganography area. 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 [2] because some harmful statistics are exploited that reveal the existence of the secret data. In this study most of the effort is done to get a better imperceptibility and decreasing image’s distortion and increasing capacity without losing stego-image quality.

The rest of this paper is organized as follows. First, the introduction of the LSB method, the Optimal LSB method, Wang *et al.*’s and Wu *et al.*’s method is given in the second section. Then, a new method is proposed in the third section which is called “Adaptive Optimal Embedding”. The fourth section gives the experimental results and discussions. Finally, the conclusion is given in the fifth section.

II. RELATED WORK

A. The Simple LSB Substitution

The word LSB stands for Least Significant Bit. This method is one of the most simple and easy to implement methods 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 262144 bits (32768 bytes or pixels). By embedding this amount of data both stego-image and its respective cover image look the same since human eye cannot distinguish this little changed value of pixels, but embedding more than one bit in each pixel by using the edge area pixels will make more change in high frequency areas so it can be still undetectable by human eye as well [3]. By considering the size of cover image pixels there is no limitation on embedding rate in this method, but the more secret bits we can embed, the less imperceptibility of stego-image is obtained. So researchers proposed several methods to improve the weakness of this method.

B. The Optimal LSB method

The simple LSB method can be modified so the quality of stego-image gets improved. The algorithms of such improved schemes are still based on simple LSB method. In this section we introduce one of the improved methods, called Optimal LSB which applies Optimal Pixel Adjustment

This paper is supported by Universiti Teknologi Malaysia (UTM) under the VOT: 79309.

Process (OPA) to improve the stego-image quality. Three candidates are picked out for the pixel's 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 [4].

The embedding process is described as follows [5]:

1- Let P_i be the original pixel value and k bit(s) of secret data is to be embedded.

2- Embed k bit(s) of secret data into P_i by using the LSBs method. The stego-image P_i' can then be obtained.

3- 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.

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

$$p_i^{"} = \begin{cases} p_i', & if\, |p_i - p_i'| \le |p_i - p_-'| \le |p_i - p_+'| \\ p_+', & if\, |p_i - p_+'| \le |p_i - p_i'| \le |p_i - p_-'| \\ p_-', & if\, |p_i - p_-'| \le |p_i - p_i'| \le |p_i - p_+'| \end{cases}$$

Finally, all the optimal candidates for $P_i^{"}$ replace the original pixel values p_i and the embedding algorithm come to its end.

C. Wang et al.'s Optimal LSB substitution Method

Wang *et al*.'s [6] approach is based on simple LSB substitution and Genetic Algorithm (GA). The process of Wang *et al.*'s method is shown in Fig. 1. There are two differences between simple LSB and Wang *et al.*'s 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.*'s 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^*.

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 and gcd(,) 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.*'s 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 $0 <= i < 2^{k-1}$ and $0 <= j < 2^{k-1}$. In 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.

According to optimal substitution matrix S, the pixel value 0 of ESI' would be replaced by value 3 in ESI^*, and the pixel value of 2 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.*'s method used GA to find the best matrix to reduce searching time.

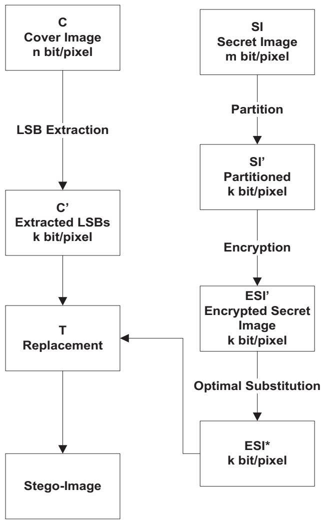
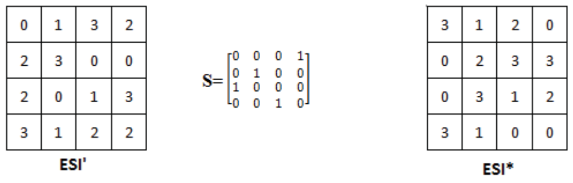


Figure 1. Block diagram of Wang *et al.*'s optimal substitution method



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*

Figure 2. An example of optimal substitution process from ESI' to ESI^* by matrix S

D. Wu et al.'s Global and Local Optimal LSB Substitution

In Wang *et al.*'s 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.*'s method and Wu *et al.*'s method. In fact Wang *et al.*'s global transformation idea is not beneficial for the whole image [7]. In Wu *et al.*'s 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.*'s method is presented in Fig. 3. In their method the decomposed image S' is divided into $\{ES'_0, ES'_1, \ldots, ES'_{n-1}\}$, in total there are n blocks. C' also will be divided into $\{C'_0, C'_1, \ldots, C'_{n-1}\}$. 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.* 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. Same as the Wang *et al.*'s approach, the matrices must be saved to be sent to the receiver for extraction process. So it needs fewer data to be recorded. 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.

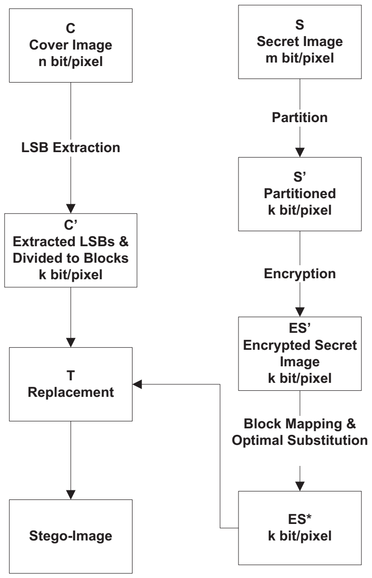
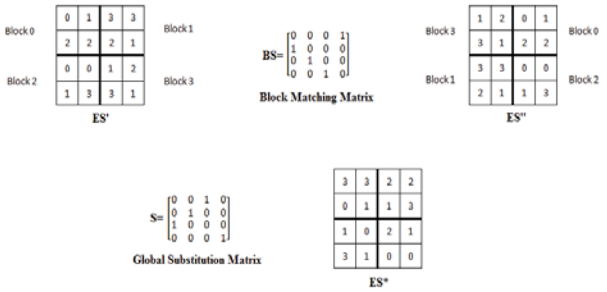


Figure 3. Block diagram of Wu *et al.*'s global and local method



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

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

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

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

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

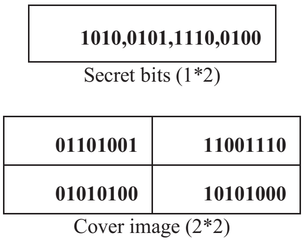
Figure 4. An example of global optimal substitution method

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.

III. 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 [4,6,7,8,9,10]. 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 were embedded sequentially and started to embed from first LSB of each pixel. On the other hand, in our method depending 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 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. 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 1048576 secret bits and size of cover image is $512 * 512$. So by dividing 1048576 to 262144 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. 5. Also the respective flowchart of embedding process is presented in Fig. 6.



1010,0101,1110,0100

Secret bits (1*2)

01101001	11001110
01010100	10101000

Cover image (2*2)

Figure 5. An Example of Secret Bits and Cover Image

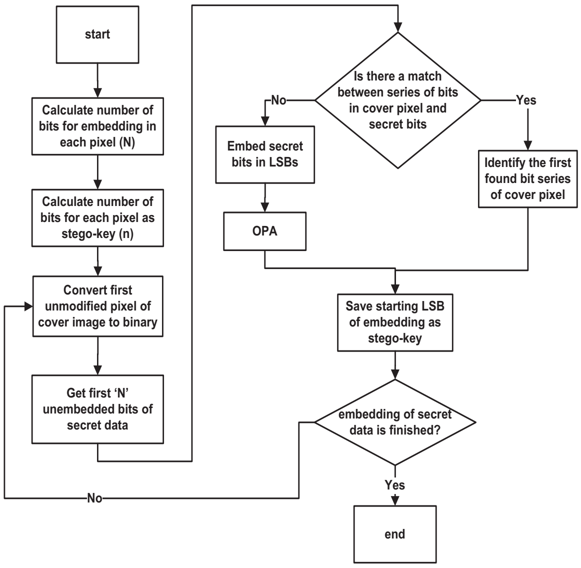

start
Calculate number of bits for embedding in each pixel (N)
Calculate number of bits for each pixel as stego-key (n)
Convert first unmodified pixel of cover image to binary
Get first 'N' unembedded bits of secret data
Is there a match between series of bits in cover pixel and secret bits
No
Yes
Embed secret bits in LSBs
OPA
Identify the first found bit series of cover pixel
Save starting LSB of embedding as stego-key
embedding of secret data is finished?
Yes
No
end

Figure 6. The Desired Embedding Flowchart of Adaptive Optimal Embedding

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. The first secret bits to embed are "1010" and the cover pixel value is "01101001". We can notice that the four bits after 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 of 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. 7, the stego-key for the sample secret bits is "10000001". Fig. 7 depicts where the secret bits are embedded and shows how the stego-key is derived.

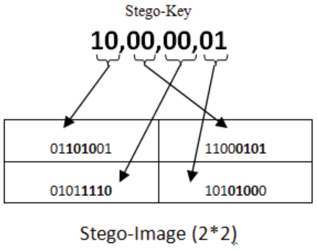


Figure 7. The Stego-Image and Stego-Key after Embedding Phase

The receiver extracts the secret bits by using the stego-key and the stego-image. According to Fig. 7, first bits of stego-image's pixel are "01101001" and the first two bits of stego-key are "10". 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 "1010". The second two bits of stego-key are "00" and this means extract the bits from the second pixel by starting first LSB which are "0101". Then repeat the extracting process for the rest of the pixels. Rest of extracting process will be done like this. The extraction flowchart is given in Fig. 8.

IV. Experimental Results

Four methods that provide high embedding capacity were described in section II. 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 $512 * 512$ which is shown in Fig. 9 is used as cover image. Fig. 10 shows three secret images Barbara, Peppers and Airplane-F16 which are used as secret images. The size of each secret image is $512 * 256$. All standard images are collected from [11].

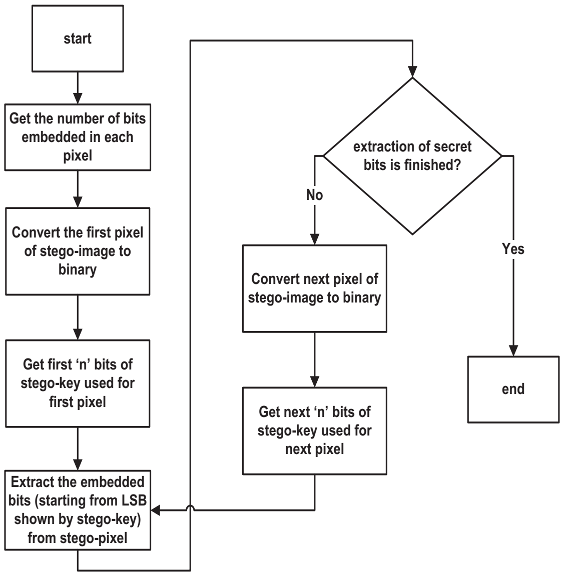

start
Get the number of bits embedded in each pixel
Convert the first pixel of stego-image to binary
Get first 'n' bits of stego-key used for first pixel
Extract the embedded bits (starting from LSB shown by stego-key) from stego-pixel
extraction of secret bits is finished?
No
Yes
Convert next pixel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

Figure 8. The Desired Extraction Flowchart of Adaptive Optimal Embedding

Figure 9. The Cover Image Lena with The Size of $512 * 512$

(a)

(b)

(c)

Figure 10. Secret Images with the Size of $512 * 256$: a: Barbara, b: Peppers, c: AirplaneF-16

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:

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

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

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 [5].

Table 1 shows the result of embedding 1048576 bits (four bits per pixel) in cover image Lena by Simple LSB, Optimal Pixel Adjustment, Wang *et al.*'s method, Wu *et al.*'s global method, Wu *et al.*'s local method and Adaptive Optimal Embedding. The results show that the value of PSNR in Wu *et al.'s* local method is significantly better than Wang *et al.*'s method because more attributes of block characteristics were explored. But the best PSNR was obtained by Adaptive Optimal Embedding and then by OPA. In Wang *et al.*'s and Wu *et al.'*s 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 ($C_5C_6C_7C_8$ 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 ($C_5C_6C_7C_8$, $C_4C_5C_6C_7$, $C_3C_4C_5C_6$, $C_2C_3C_4C_5$).

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.*'s and Wu *et al.*'s 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.

In Table 2 our algorithm is tested by 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". Because 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.

TABLE I. The Result of Embedding Secret Images 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.*'s method		32.8824	32.5453	33.0296
Wu *et al.*'s	Global method	32.9873	32.6748	33.1669
	Local method	34.3948	34.3637	34.5349
Optimal Embedding	n=1	35.1540	35.1636	35.0538
	n=2	35.7823	35.6984	35.7227

(a) (b)

Figure 11. The Stego-Image Lena with Embedded Secret Image Barbara. In (a) n=1 and (b) n=2

As discussed earlier, the best PSNR is obtained by our method and then by OPA. Although in our method size of the stego-key is large, but since the experiments show, 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 for 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 decreases significantly.

V. 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 embeds secret message bits in the next LSBs of some certain pixels of cover image. The image distortion is decreased by using OPA technique. 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 embedding the stego-key in another cover image.

VI. REFERENCES

[1] Morkel T *et al.* AN OVERVIEW OF IMAGE STEGANOGRAPHY[J]. Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005),2005.

[2] Lee Yeuan-kuen *et al.* An Advanced Least-Significant-Bit Embedding Scheme for Steganographic Encoding[J]. 2009, 5414/2009:349-360.

[3] Lee Yeuan-Kuen,Chen Ling-Hwei. An Adaptive Image Steganographic Model Based on Minimum-Error LSB Replacement[J]. 1999,(In Proceedings of the Ninth National Conference on Information Security):1-8.

[4] Chan C. Hiding data in images by simple LSB substitution[D]. . *Pattern Recognition*. 2004. pp. 469-474; http://linkinghub.elsevier.com/retrieve/pii/S003132030300284X.

[5] Wu Nan-i,Hwang Min-shiang. Data Hiding : Current Status and Key Issues[J]. International Journal,2007, 4(1):1-9.

[6] Wang Ran-Zan *et al.* Image hiding by optimal LSB substitution and genetic algorithm[J]. Pattern Recognition,2001, 34(3):671-683; http://linkinghub.elsevier.com/retrieve/pii/S0031320300000157.

[7] Wu Ming-ni *et al.* A LSB Substitution Oriented Image Hiding Strategy Using Genetic Algorithms[J]. 2004,:219-229.

[8] Zamani Mazdak *et al.* Robust audio steganography via genetic algorithm[J]. 2009 International Conference on Information and Communication Technologies,2009,:149-153; http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5267197.

[9] Wu H *et al.* Image steganographic scheme based on pixel-value differencing and LSB replacement methods[J]. October,2003, 152(5):611-615.

[10] Zhang Xinpeng,Wang Shuozhong. Steganography using multiple-base notational system and human vision sensitivity[J]. IEEE Signal Processing Letters,2005, 12(1):67-70; http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1369277.

[11] Gonzalez Rafael C,Woods Richard E. Digital Image Processing[M]. ,Pearson Education, Inc. Pearson Prentice Hall, Upper Saddle River, New Jersey 07458. 2008.

TABLE II. THE RESULT OF EMBEDDING DIFFERENT AMOUNT OF SECRET DATA WITH DIFFERENT 'N' IN COVER IMAGE LENA

Method / Size of Secret Data	Simple LSB	OPA	Adaptive Optimal Embedding			Difference of PSNR value between OPA and Adaptive Optimal Embedding		
			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			