

A Revisit to LSB Substitution Based Data Hiding for Embedding More Information

Yanjun Liu^{1,*}, Chin-Chen Chang¹, and Tzu-Yi Chien²

¹Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan, R.O.C
yjliu104@gmail.com, alan3c@gmail.com

²Department of Information Engineering and Computer Science,
National Chung Cheng University, Chiayi 62102, Taiwan, R.O.C.
m80429@yahoo.com.tw

Abstract. Steganography is a widely used approach to embed tremendous amount of secret message while maintaining satisfactory visual quality. Least-significant-bit (LSB) substitution is one of the famous techniques applied in steganography, which makes modifications to the cover image by simply substituting secret bits for the LSBs of the cover pixel. This paper presents a novel data hiding scheme based on LSB substitution in which binary secret data can directly be concealed into the cover image. To enhance the embedding capacity, as much as 2 secret bits can be embedded into each cover pixel by modifying 3 LSBs with the guidance of a reference table. Experimental results confirm that the proposed scheme outperforms the related schemes in terms of embedding capacity and visual quality.

Keywords: Steganography, least-significant-bit (LSB), embedding capacity, visual quality

1 Introduction

Nowadays, steganography (also called data hiding) has played an important role in information security due to the advantage that it not only embeds a massive amount of secret message into cover media, but also maintains satisfactory image quality without visual perception. Therefore, more and more researchers concentrate on data hiding in which digital images, videos and audios are commonly used as cover media. Different data hiding methods such as LSBs [1–8], pixel value difference (PVD) [9, 10] and Gray code [11–14] have been developed.

Data hiding technique can be traced back to the method proposed by Petitcolas et al. [1] in 1999 in which only the communicational entities know that the secret message was embedded into the multimedia. To increase the security, EL-Emam [2] proposed a method that embedded different numbers of secret bits using LSB in colorful images by three channels, i.e. red channel, green channel and blue channel. Although this method can strengthen the security during the transmission, the embedding capacity is much lower by some restrictions. Thus,

to enhance the embedding capacity, Ioannidou et al. [3] proposed a scheme based on the method of EL-Emam [2] to embed one extra secret bit in the edge area of the image.

Least-significant-bit (LSB) substitution [1–8] is one of the famous techniques applied in data hiding, which makes modifications to the cover image by simply substituting secret bits for the LSBs of the cover pixel. LSBs can achieve high embedding capacity while keeping good image quality. However, it cannot resist statistical analysis on the stego-image. In order to overcome the disadvantage of LSBs, Chen and Chang [11] proposed a method which followed the rule of Gray codes and divided the Gray codes into odd Gray codes and even Gray codes. As a result, the secret data can be embedded in the cover image by odd Gray codes or even Gray codes using 4-LSB. However, this method can neither accommodate a massive amount of secret data and nor maintain great visual quality of stego-image due to the fact that it modified four bits to only embed one secret bit.

On the other hand, Mielikainen [8] developed a method which exploited pairs of LSB matching to improve the embedding efficiency. Shen and Huang [9] utilized this advantage to propose a method that adopted Hilbert curve to segment the cover image into several pixel pairs. The difference of the two pixels in each pair is calculated and simple function (i.e., modular function) is used to embed secret data in each pixel pair. However, this method may cause overflow or underflow problems. In 2015, Jung and Yoo [10] proposed another method based on pixel value difference (PVD). Unfortunately, the visual quality of stego-image decreased rapidly when the cover image accommodated a massive amount of secret data.

In this paper, we propose a novel data hiding scheme based on LSB substitution in which binary secret data can directly be concealed into the cover image. According to a constructed reference table, 2 secret bits can be embedded into each cover pixel by modifying the 3 LSBs at most by 2. The proposed scheme can achieve high embedding capacity while maintaining great visual quality of the stego-image.

The rest of the paper is organized as follows. In Section 2, we briefly review Chen and Chang’s method [11]. Our proposed scheme is described in detail in Section 3. Finally, our experimental results and conclusions are presented in Sections 4 and 5, respectively.

2 Review of Chen and Chang’s scheme

Chen and Chang’s data hiding scheme followed the rule of Gray code [12] to embed secret message. The Gray code is an approach to encode an integer in the 2^n -ary numeral system as an n -bit binary code in such a manner that two consecutive integers have only one bit different. The method of reflected Gray code is used to create n -bit Gray codes in Chen and Chang’s scheme. Let G_n be

an n -bit Gray code sequence, then G_1, G_2 and G_3 are shown as follows:

$$G_1 = \begin{cases} 0 = 0 \\ 1 = 1 \end{cases}, G_2 = \begin{cases} 00 = 0 \\ 01 = 1 \\ 11 = 2 \\ 10 = 3 \end{cases} \text{ and } G_3 = \begin{cases} 000 = 0 \\ 001 = 1 \\ 011 = 2 \\ 010 = 3 \\ 110 = 4 \\ 111 = 5 \\ 101 = 6 \\ 100 = 7 \end{cases}.$$

Denote $g = g_n g_{n-1} \dots g_1$ as a Gray code in G_n , where g_i is one binary bit. Then the Gray code function $\text{Gray}(g)$ is denoted as the corresponding 2^n -ary value of g . In Chen and Chang's scheme, g is regarded as the n -LSB string of a cover pixel. One secret bit is embedded into a cover pixel in such a manner that just the right-most bit (i.e., g_1) of g in each cover pixel is flipped according to the characteristic of Gray code. Therefore, this scheme significantly reduces the distortion by modifying each pixel at most by 1. The embedding algorithm is illustrated below:

Embedding algorithm

Input: Cover image CI , binary secret data stream S , n -bit Gray code sequence G_n

Output: Stego image SI

Step 1. Obtain a pixel p from CI .

Step 2. Extract the n LSBs of p and denote the n -LSB string as g .

Step 3. Read one bit m from S . If $m = 1$, then go to Step 4; otherwise, go to Step 5.

Step 4. If $\text{Gray}(g)$ is odd, then remain p unchanged; otherwise, flip the right-most bit of p . Go to Step 6.

Step 5. If $\text{Gray}(g)$ is even, then remain p unchanged; otherwise, flip the right-most bit of p .

Step 6. Go to Step 1 until all secret bits have been embedded.

To clearly understand Chen and Chang's scheme, we give an example to demonstrate the embedding algorithm under $n = 4$ as follows. Assume that a cover pixel is $214_{10}(= 11010110_2)$ and the secret bit to be embedded is 1. Obviously, the last four bits of the cover pixel is $g = 0110$. We obtain $\text{Gray}(g) = \text{Gray}(0110) = 4$, which is an even value. Consequently, the right-most bit of the cover pixel is flipped and the cover pixel is modified as $215_{10}(= 11010111_2)$.

3 Proposed scheme

Chen and Chang's scheme can significantly reduce the distortion by modifying each pixel at most by 1 to the characteristic of Gray code. However, their scheme has a disadvantage that the embedding capacity is very low because each pixel

can accommodate only one secret bit in spite of the value of n . Consequently, in order to enhance the capacity, we propose a novel data hiding scheme based on LSB substitution such that 2 secret bits can be embedded into each cover pixel.

In our proposed scheme, the binary secret message is divided into a sequence of 2-bit segments and the value of each segment is represented as a 4-ary digit. To conceal a digit in to a cover pixel, four candidate reference tables are constructed to guide the embedding process. As shown in Tables 1 (a)-(d), each element in the first row of each table represents 3 LSBs of a cover pixel, which corresponds to a 4-ary digit in the second row. It should be noticed that each table can imply the embedding of only three digits that are included in it. For example, Table *A* is used to embed the digits 0, 1 and 2 but not the digit 3. Similarly, Table *B*, *C* and *D* cannot be used to embed 0, 1 and 2, respectively. To solve this problem, we can simply employ an indicator to identify the digit that does not occur in a specified reference table, and then transform this digit to one of the digits included in this table to perform embedding. In order to minimize the number of indicator bits so as to further increase the embedding capacity, we count the frequency of the secret digits 0, 1, 2 and 3, respectively, and select one reference table in which the digit with the lowest frequency does not occur. This digit is then transformed to the digit with the second lowest frequency and the embedding is conducted according to the selected reference table. For example, if the digit 3 has the lowest frequency in the secret message, we select reference table *A* to for embedding. If the digit to be embedded is 0, 1 or 2, it can be easily embedded according to table *A*. In particular, we set the indicator as 0 for the embedded digit 0 that has the second lowest frequency. If the digit to be embedded is 3, we set the indicator as 1 meanwhile transform the digit to 0 and embed it just as the way that 0 does.

Table 1. Candidate reference tables

(a) Reference table A in which digit 3 does not occur								
3 LSBs of a cover pixel	000	001	010	011	100	101	110	111
Corresponding digit	0	1	2	0	1	2	0	1

(b) Reference table B in which digit 0 does not occur								
3 LSBs of a cover pixel	000	001	010	011	100	101	110	111
Corresponding digit	1	2	3	1	2	3	1	2

(c) Reference table C in which digit 1 does not occur								
3 LSBs of a cover pixel	000	001	010	011	100	101	110	111
Corresponding digit	2	3	0	2	3	0	2	3

(d) Reference table D in which digit 2 does not occur								
3 LSBs of a cover pixel	000	001	010	011	100	101	110	111
Corresponding digit	3	0	1	3	0	1	3	0

The embedding algorithm of our proposed scheme is described as follows:

Embedding algorithm

Input: Cover image CI , secret image S

Output: Stego-image SI

- Step 1.** Covert S to a binary secret data stream S' and divide S' into a sequence of 2-bit segments. Represent the value of each segment as a 4-ary digit from 0-3.
- Step 2.** Count the frequency of the secret digits 0, 1, 2 and 3, respectively. Denote s_1 and s_2 as the digits with the lowest and second lowest frequency, respectively.
- Step 3.** Select the reference table in which s_1 does not occur.
- Step 4.** Obtain a pixel p from CI . Extract 3 LSBs of p and denote the 3-LSB string as p^* .
- Step 5.** Read a 4-ary digit s from S' . If $s = s_1$, set the indicator $flag = 1$ and let $s = s_2$; if $s = s_2$, set the indicator $flag = 0$.
- Step 6.** Find the corresponding digit d of p^* in the reference table.
- Step 7.** Choose the digit d' that is equal to s and has the shortest distance with d in the selected reference table.
- Step 8.** Find the first-row element p^{**} in reference table which corresponds to the digit d' .
- Step 9.** Modify p^* to p^{**} .
- Step 10.** Go to Step 4 until all secret data have been embedded. Output stego-image SI .

The above embedding algorithm indicates that the cover pixels are modified by +1 or -1 except for those of which the 3 LSBs are 000 and 111. Thus, the distortion of our proposed scheme can be very small with a high embedding capacity.

After the receiver obtains the stego-image, he/she can extract the secret image from the stego-image by inverting the data embedding process. The extraction algorithm of our proposed scheme is described as follows.

Extraction algorithm

Input: Stego-image SI

Output: Secret image S

- Step 1.** Obtain a pixel p' from SI . Extract 3 LSBs of p' and denote the 3-LSB string as p'^* .
- Step 2.** Find the corresponding digit d' of p'^* in the reference table.
- Step 3.** If $d' = s_2$ and $flag = 1$, s_1 is the hidden data; if $d' = s_2$ and $flag = 0$, s_2s_1 is the hidden data; otherwise, d' is the hidden data.
- Step 4.** Go to Step 1 until all secret data have been extracted. Output the secret image S .

For clearer explanation, we give an instance to demonstrate our scheme. Assume that a cover pixel p is 155 and the secret bits to be embedded are $s = 01_2(1_4)$. Suppose 01_2 and 10_2 have the lowest and second lowest frequency in the

binary secret stream S' , respectively. Therefore, $s_1 = 01_2(1_4)$ and $s_2 = 10_2(2_4)$ and Table C is selected as the reference table. Because the extracted 3 LSBs of p are 011_2 and $s = s_1$, we set $flag = 1$ and let $s = s_2 = 10_2(2_4)$. Then, we find the corresponding digit $d = 2_4$ of 011_2 in the reference table C . Thus, 3 LSBs of p remain unchanged because the corresponding digit $d' = s = 2_4$ of 011_2 (3-LSB of p itself) has the shortest distance with d . Finally, the stego pixel is 155. In the extraction, the extracted 3 LSBs of stego pixel 155 is 011_2 and we find the corresponding digit $d' = 2_4$ of 011_2 in the reference table C . Since $d' = s_2$ and $flag = 1$, the hidden data is $s_1 = 1_4 = 01_2$.

4 Experimental results

Peak signal to noise ratio (PSNR) is used in our experiments to measure the similarity between the stego-image and the cover image. PSNR is defined as follows:

$$\text{PSNR} = 10\log_{10} \left(\frac{255^2}{\text{MSE}} \right), \quad (1)$$

where the mean square error (MSE) for a $W \times H$ grayscale image is defined as follows:

$$\text{MSE} = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H (I_{i,j} - S_{i,j})^2, \quad (2)$$

where $I_{i,j}$ and $S_{i,j}$ are the original pixel value and the stego pixel value, respectively. In the following, we employ four 512×512 standard grayscale images, Lena, Baboon, GoldHill and Peppers as test images in our experiments. As shown in Figure 1 (a), a 512×512 binary image Peppers is adopted as the secret image. Table C is selected as the reference table due to the fact that the bit string 01 has the lowest frequency occurring in the secret image Peppers. The experimental results are shown in Table 2 which indicates that our proposed scheme outperforms Chen and Chang's scheme in terms of the embedding capacity (EC) while great PSNR is still achieved. That is, PSNR in our scheme can reach 49 dB even if we embed 517,388 bits in the cover images. In particular, Figure 1 shows the stego-image for Lena (see Figure 1 (b)) and the extracted binary image Peppers (see Figure 1 (c)). It can be observed that it is impossible to distinguish between the cover image and stego-image by human eyes since the PSNR value is achieved up to 49.04 dB.

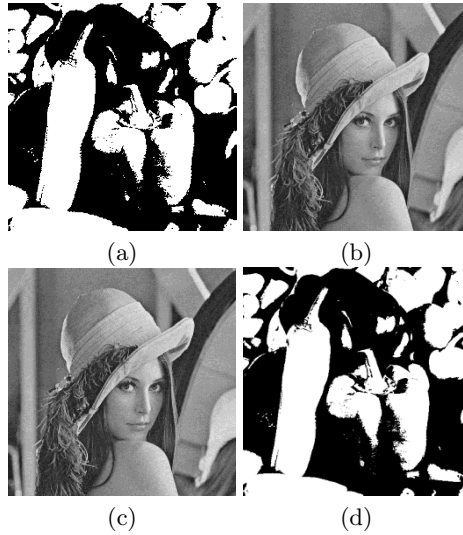


Fig. 1. Results of the proposed scheme (a) original secret image (b) original cover image (c) stego-image (PSNR = 49.04 dB) (d) extracted secret image

Table 2. Performance comparisons between Chen and Chang's scheme and our scheme

	Chen and Chang [11]		Our scheme	
Cover image	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)
Lena	262,144	51.15	517,388	49.04
Baboon	262,144	51.13	517,388	49.13
GoldHill	262,144	51.13	517,388	49.13
Peppers	262,144	51.14	517,388	49.11

To further evaluate the performance of our proposed scheme, Table 3 give a comparison of PSNR values among different data hiding schemes under the same embedding capacity (262,144 bits). The results are tested on 512×512 binary secret image Peppers and eight 512×512 grayscale cover images Lena, Baboon, Airplane, Boat, Barbara, Peppers, Tiffany and Man. From Table 3, we can see that the average PSNR of our proposed scheme is greater than 52 dB, which is better than that of other schemes.

Table 3. PSNR values of different schemes

		Maleki et al. [15]	Shen et al. [9]	Jung and Yoo [10]	Our scheme
Cover image	EC (bits)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
Lena	262,144	49.73	44.8	37.19	52.03
Baboon	262,144	49.26	42.56	30.66	52.16
Airplane	262,144	49.87	44.63	35.66	52.17
Tiffany	262,144	49.72	44.71	36.56	52.13
Barbara	262,144	49.46	43.28	32.54	52.12
Man	262,144	49.52	44.34	35.56	52.18
Peppers	262,144	49.70	44.67	36.85	52.12
Boat	262,144	49.77	44.39	35.76	52.12
Average	262,144	49.63	44.17	35.10	52.13

5 Conclusions

In this paper, we proposed a novel data hiding scheme based on LSB substitution. With the guidance of the constructed reference table, each cover pixel can embed two secret bits. The distortion of the cover pixel is at most 2, which leads to a great image quality. The experimental results show that for a 512×512 image, our proposed scheme can embed at most 517,388 bits while the image quality can maintain above 49 dB. Comparisons demonstrate that the proposed scheme outperforms the related schemes in terms of embedding capacity and visual quality.

References

1. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn.: Information hiding – a survey. *Proceedings of the IEEE*, Vol. 87, No. 7, pp. 1062–1078 (1999)
2. N. N. EL-Emam.: Hiding a large amount of data with high security using steganography algorithm. *Journal of Computer Science*, Vol. 3, No. 4, pp. 223–232 (2007)
3. A. Ioannidoua, S. T. Halkidisb and G. Stephanidesb.: A novel technique for image steganography based on a high payload method and edge detection. *Expert Systems with Application*, Vol. 39, No. 14, pp. 11517–11524 (2012)
4. H. R. Kanan and B. Nazeri.: A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. *Expert Systems with Application*, Vol. 39, No. 14, pp. 11517–11524 (2014)
5. K. H. Jung and K. Y. Yoo.: Steganographic method based on interpolation and LSB substitution of digital images. *Multimedia Tools and Applications*, Vol. 74, No. 6, pp. 2143–2155 (2015)
6. C. H. Yang.: Inverted pattern approach to improve image quality of information hiding by LSB substitution. *Pattern Recognition*, Vol. 41, No. 8, pp. 2674–2683 (2008)

7. X. Liao, Q. Y. Wen and J. Zhang.: A steganographic method for digital images with four-pixel differencing and modified LSB substitution. *Journal of Visual Communication and Image Representation*, Vol. 22, No. 1, pp. 1–8 (2011)
8. J. Mielikainen.: LSB matching revisited. *IEEE Signal Processing Letters*, Vol. 13, No. 5, pp. 285–287 (2006)
9. S. Y. Shen and L. H. Huang.: A data hiding scheme using pixel value differencing and improving exploiting modification directions. *Computer and Security*, Vol. 48, pp. 131–141 (2015)
10. K. H. Jung and K. Y. Yoo.: High-capacity index based data hiding method. *Multimedia Tools and Applications*, Vol. 74, No. 6, pp. 2179–2193 (2015)
11. C. C. Chen and C. C. Chang.: LSB-based steganography using reflected gray code. *IEICE Transactions on Information and Systems*, Vol. E91-D, No. 4, pp. 1110–1116 (2008)
12. M. Schwartz and T. Etzion.: The structure of single-track gray codes. *IEEE Transactions on Information theory*, Vol. 45, No. 7, pp. 2383–2396 (1999)
13. C. C. Chang, C. C. Lin and Y. H. Chen.: A secure data embedding scheme using gray-code computation and SMVQ encoding. *Information Hiding and Applications*, Vol. 227, pp. 63–74 (2009)
14. X. Y. Luo, F. L. Liu, C. F. Yang, S. G. Lian and Y. Zeng.: Steganalysis of adaptive image steganography in multiple gray code bit-planes. *Multimedia Tools and Applications*, Vol. 57, No. 3, pp. 651–667 (2012)
15. N. Maleki, M. Jalali and M. V. Jahan.: Adaptive and non-adaptive data hiding methods for grayscale images based on modulus function. *Egyptian Informatics Journal*, Vol. 15, No. 2, pp. 115–127 (2014)

Advances in Intelligent Information Hiding and
Multimedia Signal Processing
Proceeding of the Twelfth International Conference on
Intelligent Information Hiding and Multimedia Signal
Processing, Nov., 21-23, 2016, Kaohsiung, Taiwan,
Volume 1

Pan, J.-S.; Tsai, P.-W.; Huang, H.-C. (Eds.)

2017, XIV, 336 p. 170 illus., Hardcover

ISBN: 978-3-319-50208-3