Two Improved Data Hiding Schemes

Yuh-Ming Huang
Dept. of Computer Science and Information Engineering
National Chi Nan University
Puli, Nantou, Taiwan, ROC
ymhuang@csie.ncnu.edu.tw

Pei-Wun Jhan
Dept. of Computer Science and Information Engineering
National Chi Nan University
Puli, Nantou, Taiwan, ROC
s98321535@ncnu.edu.tw

***Abstract*—The meaning of data (information) hiding is to embed the secret information into a cover host, such as an image. Usually, the naked eye of the people cannot perceive any change when the image is modified slightly. The evaluation of data hiding schemes should be measured by the distortion (or called Mean Square Error; MSE) and the embedding rate (the average number of bits embedded in a cover pixel). In this paper, we propose two improved data hiding schemes. One is to improve the EMD (Exploiting Modification Direction)-based data hiding algorithm to have higher stego-image quality. The other is to improve the Matrix encoding-based data hiding algorithm by using the idea of Hamming+1 to further improve the stego-image quality. Both proposed improved schemes are verified to be correct through the theoretical analysis and the experiment.**

Keywords- EMD; matrix encoding; magic matrix; (5,3) linear code; linear block code; data hiding; information hiding; Hamming+1; distortion; embedding rate

I. INTRODUCTION

With the rapid growth in digital technology, it is convenient for us to communicate with each other via a variety of communications products. The multimedia data flows, such as image and video, have been around us everywhere. Since that it is not easy for the naked eye to detect any anomaly in a picture with a little distortion. Data hiding technology [1] can be used to hide the secret information in a picture.

In 2006, Zhang *et al.* proposed a data hiding scheme by exploiting modification direction (EMD) [2], in which each secret digit in a $(2n+1)$-ary notational system is carried by n cover pixels and, at most, only one pixel is increased or decreased by 1. Hence, this scheme can provide high stego-image quality, but limited hiding capacity. Theoretically, this scheme is optimal in terms of the stego-image quality for a fixed embedding rate. In 2007, Chang *et al.* [3] showed that the embedding rate is bounded by $\log_2 5/2$. Besides, by using a technology of two-stage embedding, an improved EMD hiding method was proposed to increase the embedding rate. Basically, a trade off exists between the embedding rate and the hiding capacity [3-7]. In 2010, Chang *et al.* [7] extended the work of [2] and proposed a flexible EMD-based data hiding scheme.

Alternative way of data hiding, called matrix encoding, was early proposed by Crandall [8] in 1998. Matrix encoding was used in the well-known steganographic algorithm F5 [9] and applied to large payload applications [10]. Relations between the linear codes and the steganography were studied in [11]. In 2007, Zhang *et al.* [12] proposed the idea of Hamming+1 to increase the embedding rate for matrix encoding. Further improvement in embedding rate and stego-image quality was respectively proposed in [13] and [14].

In this paper, we propose two data hiding schemes respectively for improving the methods of [7] and [14] with lower distortion. The paper is organized as follows. Section II introduces our proposed schemes. The experimental results are shown in section III. Finally, some remarks on future work are made in Section IV.

II. TWO IMPROVED DATA HIDING SCHEMES

A. EMD-Based

Based on the method of Zhang and Wang [2], Chen *et al.* [7] proposed a data hiding scheme with a different extraction function, as shown in Eq. (1), to increase the hiding capacity.

$$f(g_1,g_2)=(g_1\times n^0+g_2\times n^1) \bmod n^2,$$
$$\text{where } 0\le g_i\le 255 \text{ and } 1\le i\le 2. \quad (1)$$

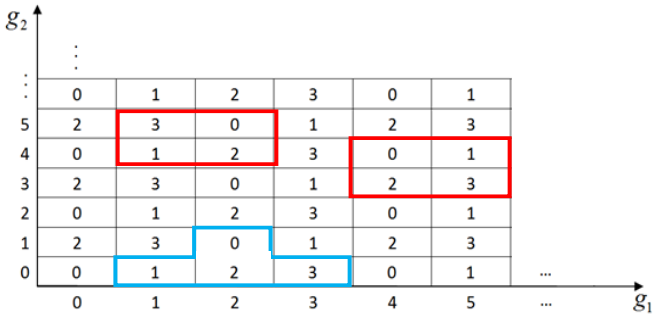

Figure. 1. The f values generated with $n = 2$.

$$f(g_1,g_2)=(g_1\times n^0+g_2\times n^1) \bmod 2^n \quad (2)$$

According to the Eq. (1), the function f values are depicted, as shown in Fig. 1. Those f values, enclosed in a square, are mutually different and within $[0,n^2-1]$ in any $n\times n$ sliding window. The advantage of this method is that it is flexible to choose different n which depends on the requirement of large hiding capacity or high stego-image quality. But, we observe that there exist two flaws for this method. One is that the value of $\log_2 n^2$ is not necessary an integer. Another is that, for the right red box in Fig. 1, the f value 1 nearest to the f value 2 is on the left, not on the upper or lower right corner. Here, we propose another different extraction function, as shown in Eq. (2), and adopt an irregular

sliding window, instead of a square sliding window, to enhance the stego-image quality. In Fig. 1, the red box is the square sliding window and the blue box is the irregular sliding window. Notice that although the f value 1 on the left side of the f value 2 will be included if we choose the sliding window of left red box in Fig. 1, it is up to four square sliding windows and the best data hiding in terms of the stego-image quality can be achieved if the user alternatively choose the sliding window according to the hiding information. We believe it is time-consuming.

1) Embedding

Step 1: The original gray-level image is partitioned into several blocks and each block contains two pixels of which the pixel values are respectively equal to g_1 and g_2.

Step 2: The hiding data represented in a binary bitstream $b_1b_2...b_N$ is converted into a sequence of digits, $s_1s_2...s_{N/n}$, where each digit is represented in a 2^n-ary notational system.

Step 3: For i = 1 to N/n { Calculate the f value f_i associated with the block (g_1, g_2), and then draw the box and f_i is regarded as the center. In the irregular-shaped area, finding the location with the f value equal to s_i, that is $s_i = f(\hat{g}_1, \hat{g}_2)$. Next, g_1 and g_2 is modified as $\hat{g}_1$ and $\hat{g}_2$ to complete the hiding of the secret digit s_i. }

2) Extracting

Step1: The stego-image is partitioned into several blocks and each block contains two pixels of which the pixel values are respectively equal to $\hat{g}_1$ and $\hat{g}_2$.

Step2: According to the Eq. (2), the f value $\hat{f}_i$ associated with each block $(\hat{g}_1, \hat{g}_2)$ is calculated and the value $\hat{f}_i$ is the secret digit s_i.

Example 1.

- *Embedding*

Suppose the secret bitstream is $(01)_2$ with its 4^{th}-ary representation $(1)_4$ and the pixel values of g_1 and g_2 are respectively equal to 2 and 2. Hence, we have f (2, 2) = 2 and g_1 *and* g_2 are respectively modified as 1 and 2, i.e. $\hat{g}_1 = g_1 - 1$, $\hat{g}_2 = g_2$, and f (1, 2) = 1.

- *Extracting*

The secret digit $(1)_4$ can be obtained easily just by directly calculating the f value, i.e. $1 = f(1,2)$.

3) Theoretical analysis of the distortion

Usually, the following two metrics are used to evaluate the efficiency of data hiding schemes.

Embedding rate (R): the number of secret bits embedded in each cover pixel.

Peak Signal to Noise Ratios (PSNR):

$$\text{PSNR} = 10\times\log_{10}(255^2/\textit{distortion}), \quad (3)$$

where $\textit{distortion} = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\alpha_{i,j} - \beta_{i,j})^2$, $\alpha_{i,j}$ and $\beta_{i,j}$ respective denote the $(i,j)^{\text{-th}}$ pixel values of the original image and the setgo-image, and M and N respective denote the width and height of the image.

For n=2, the embedding rate R is equal to 1 (= 2/2) (bit per pixel). Here, we assume that the value of each of the secret digits is uniformly distributed within [0, n^2-1]. Hence, the average distortion for a stego-image is equal to:

$\frac{1}{2}\left[(0^2+0^2)\frac{1}{4}+(1^2+0^2)\frac{3}{4}\right]=\frac{3}{8}\cong 0.375$, and PSNR = 52.39. Whereas, for the method of [7], the average distortion for a stego-image is equal to:

$\frac{1}{2}\left[(0^2+0^2)\frac{1}{4}+(1^2+0^2)\frac{2}{4}+(1^2+1^2)\frac{1}{4}\right]=\frac{4}{8}\cong 0.5$ and PSNR = 51.14.

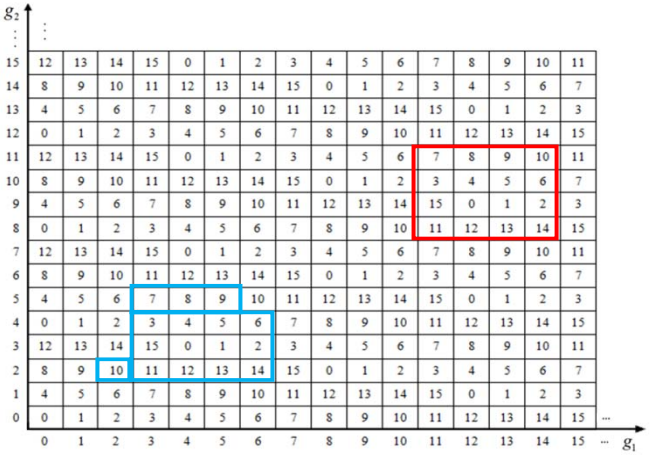

Figure. 2. The f values generated with n = 4.

For n=4, the function f values are depicted, as shown in Fig. 2. The embedding rate R is equal to 2 (= 4/2). The average distortion for a stego-image is equal to:

$$\frac{1}{2}\left[(0^2+0^2)\frac{1}{16}+(1^2+0^2)\frac{4}{16}+(2^2+0^2)\frac{2}{16}+(1^2+1^2)\frac{4}{16}+(1^2+2^2)\frac{5}{16}\right]$$

$$=\frac{45}{32}\cong 1.40 \text{ and PSNR} = 46.67.$$

Whereas, for the method of [10], the average distortion for a stego-image is equal to:

$$\frac{1}{2}\left[(0^2+0^2)\frac{1}{16}+(1^2+0^2)\frac{4}{16}+(0^2+2^2)\frac{2}{16}+(1^2+1^2)\frac{4}{16}+(1^2+2^2)\frac{4}{16}+(2^2+2^2)\frac{1}{16}\right]$$

$$=\frac{48}{32}\cong 1.5 \text{ and PSNR} = 46.37.$$

B. Matrix Encoding-Based

Based on the (7, 4) Hamming code, Chang *et al.* [13] proposed a data hiding scheme with the embedding rate R = 1.

Recently, based on the (5, 3) linear block code and the idea of Hamming+1 [12], Lin *et al.* [14] proposed another data hiding scheme with the same embedding rate and the better stego-image quality. In the proposed scheme, we modify the exclusive-or equations (4-1) ~ (4-3) adopted in [14] by double using the idea of Hamming+1 to further improve the stego-image quality. The modified exclusive-or equations are shown in equations (5-1) ~ (5-3).

Table I. Standard array for a (5, 3) code

data	000 (d_0)	001 (d_1)	010 (d_2)	011 (d_3)	100 (d_4)	101 (d_5)	110 (d_6)	111 (d_7)
Coset Leader	00000 (e_0)(c_0)	00111 (c_1)	01001 (c_2)	01110 (c_3)	10010 (c_4)	10101 (c_5)	11011 (c_6)	11100 (c_7)
	00001 (e_1)	00110	01000	01111	10011	10100	11010	11101
	00010 (e_2)	00101	01011	01100	10000	10111	11001	11110
	00011 (e_3)	00100	01010	01101	10001	10110	11000	11111

1) (5, 3) linear block code

For a (5, 3) linear code with the generator matrix $G = \begin{bmatrix} 10010 \\ 01001 \\ 00111 \end{bmatrix}_{3\times 5}$. The standard array for this code is shown in Table I. Any vector $\boldsymbol{v} \in GF(2)^5$, there must exist a codeword $\boldsymbol{c_i}$ and a coset leader $\boldsymbol{e_j}$, such that

$\boldsymbol{v} = \boldsymbol{c_i} + \boldsymbol{e_j} = \boldsymbol{d_i}\times G + \boldsymbol{e_j} = \boldsymbol{d_i}\times G + (0, 0, 0, \boldsymbol{e_{j0}}, \boldsymbol{e_{j1}})$. That is, the vector $\boldsymbol{v}$ can be obtained once we know $\boldsymbol{d_i}$, $\boldsymbol{e_{j0}}$, and $\boldsymbol{e_{j1}}$. For simplifying the notation, we denote each $\boldsymbol{d_i}$ as (d_1, d_2, d_3).

2) Embedding

Step 1: The original gray-level image is partitioned into several blocks and each block contains six pixels of which the pixel values are respectively equal to $p_1, p_2, \ldots, p_6$ and the binary representation of p_i is $p_{i7}p_{i6}p_{i5}p_{i4}p_{i3}p_{i2}p_{i1}p_{i0}$, where $1 \le i \le 6$.

Step 2: The hiding data represented in a binary bitstream $b_1b_2\ldots b_N$ is partitioned into several blocks and each block contains six bits $s_1s_2s_3s_4s_5s_6$.

Step 3: For $i = 1$ to $N/6$ {For each block of secret bits $s_1s_2s_3s_4s_5s_6$, let $\boldsymbol{v} = (s_1, s_2, s_3, s_4, s_5)$ and find the $\boldsymbol{d_i}$ and the $\boldsymbol{e_j}$ by the equation $\boldsymbol{v} = \boldsymbol{d_i}\times G + \boldsymbol{e_j}$. Then, some of the pixel values, $p_1, p_2, \ldots, p_6$, will be increased or decreased by 1 to make the equations (5-0) ~ (5-3) hold.
}

$$p_{11} \oplus p_{40} = e_{j0} \quad (4\text{-}1)$$
$$p_{21} \oplus p_{50} = e_{j1} \quad (4\text{-}2)$$
$$p_{31} \oplus p_{60} = s_6 \quad (4\text{-}3)$$

$$d(p_{10}p_{20}p_{30}, d_1d_2d_3) = 0 \quad (5\text{-}0)$$
$$p_{11} \oplus p_{21} \oplus p_{40} = e_{j0} \quad (5\text{-}1)$$
$$p_{21} \oplus p_{31} \oplus p_{50} = e_{j1} \quad (5\text{-}2)$$
$$p_{31} \oplus p_{41} \oplus p_{60} = s_6 \quad (5\text{-}3)$$

In Eq. (5-0), $\boldsymbol{d}(p_{10}p_{20}p_{30}, d_1d_2d_3)$ denotes the Hamming distance between $p_{10}\,p_{20}\,p_{30}$ and $d_1\,d_2\,d_3$.

3) Extracting

Step1 : The stego-image is partitioned into several blocks and each block contains six pixels of which the pixel values are respectively equal to $p_1, p_2, \ldots, p_6$ and the binary representation of p_i is $p_{i7}p_{i6}p_{i5}p_{i4}p_{i3}p_{i2}p_{i1}p_{i0}$, where $1 \le i \le 6$.

Step2: For $i = 1$ to $N/6$ {First, we get d_1, d_2, d_3, $\boldsymbol{e_{j0}}$, and $\boldsymbol{e_{j1}}$ by setting $d_1 = p_{10}$, $d_2 = p_{20}$, $d_3 = p_{30}$, $\boldsymbol{e_{j0}} = p_{11}\oplus p_{21}\oplus p_{40}$, and $\boldsymbol{e_{j1}} = p_{21}\oplus p_{31}\oplus p_{50}$. Then, we get the secret bits $(s_1, s_2, s_3, s_4, s_5) = (d_1, d_2, d_3)\times G + (0, 0, 0, \boldsymbol{e_{j0}}, \boldsymbol{e_{j1}})$. Finally, $s_6 = p_{31}\oplus p_{41}\oplus p_{60}$. }

Example 2.

- Embedding

Suppose $(p_1, p_2, p_3, p_4, p_5, p_6) = ((100)_{10}, (100)_{10}, (220)_{10}, (151)_{10}, (95)_{10}, (90)_{10}) = ((01100100)_2, (01100100)_2, (11011100)_2, (10010111)_2, (01011111)_2, (01011010)_2)$ and $(s_1s_2s_3s_4s_5s_6) = (101110)$. First, we find $\boldsymbol{d_i} = (d_1d_2d_3) = (101)$ and $\boldsymbol{e_j} = (0, 0, 0, \boldsymbol{e_{j0}}, \boldsymbol{e_{j1}}) = (0,0,0,1,0)$ by letting $\boldsymbol{v} = (1,0,1,1,1)$. Hence, $p_{10} \ne d_1$, $p_{20} = d_2$, $p_{30} \ne d_1$, $p_{11}\oplus p_{21}\oplus p_{40} = 1$, $p_{21} \oplus p_{31} \oplus p_{50} = 1$, and $p_{31} \oplus p_{41} \oplus p_{60} = 1$. Then, we complement the least significant bit of p_1 and decrease the pixel value of p_3 by 1 to make the equations (5-0) ~ (5-3) hold. That is, $(p_1, p_2, p_3, p_4, p_5, p_6)$ is now modified as $((101)_{10}, (100)_{10}, (219)_{10}, (151)_{10}, (95)_{10}, (90)_{10})$ (= $((01100101)_2, (01100100)_2, (11011011)_2, (10010111)_2, (01011111)_2, (01011010)_2)$) to complete the hiding of the secret bits 101110. Remark: It needs to modify three pixels (complement p_{10}, p_{30}, and p_{50}) in the original method of [14].

- Extracting

According to the six pixel values (01100101, 01100100, 11011011, 10010111, 01011111, 01011010) and the equations (5-1)~(5-3), we first obtain $(d_1, d_2, d_3) = (p_{10}, p_{20}, p_{30}) = (1,0,1)$, $\boldsymbol{e_{j0}} = p_{11}\oplus p_{21}\oplus p_{40} = 0\oplus 0\oplus 1 = 1$, and $\boldsymbol{e_{j1}} = p_{21}\oplus p_{31}\oplus p_{50} = 0\oplus 1\oplus 1 = 0$, then we get the secret bits $(s_1, s_2, s_3, s_4, s_5) = (1, 0, 1)\times G + (0, 0, 0, 1, 0) = (1, 0, 1, 1, 1)$. Finally, $s_6 = p_{31}\oplus p_{41}\oplus p_{60} = 1\oplus 1\oplus 0 = 0$.

4) Theoretical analysis of the distortion

The embedding rate R is equal to 1 (= 6/6). The derivation of the average distortion is similar to that of [14], here, we omit the detailed derivation.

The average distortion for a stego-image is equal to:

$$\frac{1}{6}\left[0\times\frac{1}{64}+1\times\frac{9}{64}+(1^2+1^2)\frac{30}{64}+(1^2+1^2+1^2)\frac{24}{64}\right]=\frac{141}{384}\cong 0.3671875$$

and the PSNR = 52.45.

Whereas, for the method of [14], the average distortion for a stego-image is equal to:

$$\frac{1}{6}[0\times\frac{1}{64}+1\times\frac{9}{64}+2\times\frac{27}{64}+3\times\frac{27}{64}]=\frac{144}{384}\cong 0.375$$

and the PSNR = 52.39.

III. Experimental Results

In our experiment, we randomly choose six gray-level images as the cover images which were used in the past related work. There are two kinds of size 256×256 and 512×512. The secret images are the Barbara images of size 90×91, 181×181, 128×128, and 256×256.

Table II and Table III are the PSNR comparison for the methods of [7] and our proposed scheme 1 respectively under the same embedding rate R = 1 and R = 2. Table IV is the PSNR comparison for the methods of [14] and our proposed scheme 2 under the same embedding rate R = 1 Experimental results show that the proposed scheme 2 outperforms both of the previous work, [7] and [14], in stego-image quality.

Table II. The PSNRs of the stego-images respectively carry the secret image Barbara of size 90×91 and the secret image Barbara of size 181×181 (n = 2 and R = 1)

	Chang *et al.* [7]		Scheme 1	
Image size	256×256	512×512	256×256	512×512
Cameraman	51.1020	51.1481	52.3539	52.3961
Baboon	51.1526	51.1383	52.4125	52.3930
Goldhill	51.1509	51.1318	52.3988	52.3789
Lena	51.1330	51.1375	52.3866	52.3877
Peppers	51.1610	51.1578	52.4054	52.4015
Barbara	51.1441	51.1428	52.3884	52.3867

Table III. The PSNRs of the stego-images respectively carry the secret image Barbara of size 128×128 and the secret image Barbara of size 256×256 (n = 4 and R = 2)

	Chang *et al.* [7]		Scheme 1	
Image size	256×256	512×512	256×256	512×512
Cameraman	46.3402	46.3914	46.6226	46.6494
Baboon	46.3393	46.3841	46.6244	46.6510
Goldhill	46.3688	46.3508	46.6560	46.6473
Lena	46.3897	46.3239	46.6703	46.6254
Peppers	46.3486	46.3651	46.6381	46.6468
Barbara	46.3823	46.371	46.6661	46.6516

Table IV. The PSNRs of the stego-images respectively carry the secret image Barbara of size 90×91 and the secret image Barbara of size 181×181 (R = 1)

	Lin *et al.* [14]		Scheme 2	
Image size	256×256	512×512	256×256	512×512
Cameraman	52.67	52.67	53.16	53.15
Baboon	52.39	52.35	52.92	52.88
Goldhill	52.57	52.69	53.12	53.25
Lena	52.57	52.56	53.10	53.07
Peppers	52.99	53.01	53.47	53.49
Barbara	52.71	52.69	53.26	53.25

IV. Conclusions

From Table II ~Table IV, for each fixed image size, we observe that the variance of the PSNR values is small. That is, the stego-image quality is stable. It is almost independent of the cover image. In the future, instead of directly embedding the secret data into the cover image in the time-domain, we will study on the embedding in the transform-domain and evaluate the robustness of this kind of data hiding scheme while against the JPEG compression.

Acknowledgment

This work was supported in part by the National Science Council, Taiwan R. O. C., under the Contract NSC 99-2628-E-260-010.

References

[1] D. G.L Simmons, "The prisoners problem and subliminal channels," in *Proc. Annu. Int. Cryptology Conf.*, *Santa Barbara*, CA, 1984, pp. 51-67.

[2] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Communications Letters*, vol. 10, no. 11, pp.781-783, Nov. 2006.

[3] C. C. Chang, W. L. Tai, and K. N. Chen, "Improvements of EMD embedding for Large payloads," in *Proc. Intelligent Information Hiding and Multimedia Signal ,* vol.1, 2007.

[4] C. F. Lee, Y. R. Wang, and C. C. Chang, "A steganographic method with high embedding capacity by improving exploiting modification direction," in *Proc. third international conference on IIHMSP* vol. 1, 2007, pp.497-500.

[5] J. Y. Byun, K. H. Jung, and K. Y. Yoo, "Improved data hiding method by exploiting modification direction," in *Proc. International Symposium on Ubiquitous Multimedia Computing,* 2008.

[6] K. H. Jung, K. Y. Yoo, "Improved exploiting modification direction method by modulus operation," *International Journal of Signal Processing, Image Processing and Pattern,* vol. 2, no. 1, pp. 79-88, Mar. 2009.

[7] K. N. Chen, C. C. Chang, and H. C. Lin, "A large payload EMD embedding scheme with high stego-image quality," In *proc. international conference on computational aspects of social networks*, 2010, pp.126-130.

[8] R. Crandall, "Some notes on steganograph," *http://of.inf.tu-duresden.de/~westfeld/crandall.pdf*, 1998.

[9] A. Westfeld, "F5-a steganographic algorithm," *in Proc. of the 4th International Workshop on Information Hiding,* Springer-Verkag, 2001, pp.289-302.

[10] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Inf. Security Forensics,* vol. 1, no. 3, pp. 390-394, Sept. 2006.

[11] M. Khatirinejad and P. Lisonek, "Linear codes for high payload steganography," *Discrete Applied Mathematics,* vol. 157, pp.971-981, Mar. 2009.

[12] W. Zhang, S. Wang, and Z. Xinpeng, "Improving embedding efficiency of covering codes for applications in steganography," *IEEE Communications Letters,* vol.11, pp.680-682, 2007.

[13] C. C. Chang, T. D. Kieu, and Y. C. Chou, "A high payload steganographic scheme based on (7, 4) hamming code for digital images" *in* Proc. *of international symposium on electronic commerce and security*, pp.16-21, 2008.

[14] Hsiu-Feng Lin, Jui-Hsiu Chang, Chiou-Yueh Gun and Chih-Ying Chen, "A Low Distortion Information Hiding Method Based on (5, 3) Code," *Inter. journal of innovative computing, and information control* , pp. 1-10 , Jul. 2011.