Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **6**, November **2015**

A Formula Diamond Encoding Data Hiding Scheme

Wen-Chung Kuo[a], Po-Yu Lai[a], Chun-Cheng Wang[b] and Lih-Chyau Wuu[a]

[a]Department of Computer Science and Information Engineering
National Yunlin University of Science & Technology
{simonkuo@yuntech.edu.tw}
[b]Graduate School of Engineering Science and Technology Doctoral Program,
National Yunlin University of Science & Technology
Yunlin, R.O.C. Taiwan.

Received December, 2014; revised August, 2015

Abstract. *Recently, many data hiding schemes based on the Exploiting Modification Directions (EMD) method have been proposed. In particular, the EMD-type with diamond encoding (DEMD) method improves upon the original EMD method from* 1.16*bpp (bits per pixel) to* $\log_2(2k^2+2k+1)/2$ *bpp per pixel pair for embedding secret data. However, the DEMD-scheme needs additional space to store and calculate the diamond characteristic value (DCV) matrix. To overcome this disadvantage, a formula diamond data hiding scheme will be proposed in this paper. According to the simulation results, we show our proposed scheme embeds secret data directly without calculating the DCV matrix by using a formula and also maintains good embedding capacity and stego image quality.*

Keywords: Data hiding, Exploiting modification direction, diamond encoding, Steganography, Steganalysis.

1. **Introduction.** Due to communication technology improvements, remarkable advances have been made in telecommunications. A very large amount of digital multimedia is transmitted over the Internet. However, there are many attacks such as illegal duplication or distribution that may occur when digital multimedia is transmitted. Generally, there are two approaches to protect digital data, cryptography and steganography. Cryptography uses encryption technologies such as DES [4], AES [1], or RSA [16] to transform plaintext into ciphertext. However, the attacker can discriminate ciphertext from other data easily when it is transmitted from sender to receiver, because encoded secret messages are always unintelligible. The steganography approach embeds secret information into other digital media that is not readily apparent to the human vision system. Therefore, data thieves will not be alerted when secret digital media is transmitted.

There are many methods proposed for steganographical data hiding schemes such as Least Significant Bit (LSB) [19], LSB Matching [15], EMD-type [3, 9, 17, 21] and other schemes such as [6, 7, 12, 14] etc. The most common data hiding scheme is the LSB method proposed by Turner in 1989 [15]. LSB replaces the k-th bits ($1 \leq k \leq 8$) of each pixel with secret information. In general, the quality of the stego image will be acceptable when the number of replaced bits is low ($k \leq 3$), because humans cannot detect the difference between original images and images embedded with additional data. However, LSB generates some special characteristics during embedding, e.g., the characteristics of the cover image pixel are destroyed by secret data during encryption and the histogram of the stego image pixels will display a Pair of Values (PoVs) [18]. Therefore, the LSB

method is detected easily via steganalysis using methods such as Visual Attack [18], RS Attack [5], etc.

To reduce the number of modified pixels and avoid direct replacement, the LSB matching method [19] was proposed by Mielikainen in 2006. The LSB matching method only modifies each pixel by one bit while two bits of secret information are embedded into one pixel pair. This method reduces the number of modified pixels, so the stego image quality is better than the traditional LSB method for same embedding capacity. At the same time, the LSB matching method affords protection against the Chi-square attack [18] because it only modifies one pixel of two adjacent pixels. In 2006, Zhang and Wang [21] proposed another data hiding scheme based on Exploiting Modification Direction (EMD). This method uses the relationship of n adjacent pixels to embed secret data. In other words, the EMD method is an enhanced LSB matching method by using n pixels to embed $(2n+1)$-ary secret information. The EMD method has more embedding capacity and higher image quality. However, the largest payload is 1.16bpp when $n = 2$, and the embedding capacity deteriorates rapidly when n increases. Afterwards, many similar EMD methods have been proposed [8, 11, 13]. However, these methods require a reference matrix to embed secret data. To overcome this shortcoming, Kuo *et al.* proposed the formula fully exploiting modification directions method (FFEMD) [10] in 2013 to allow the FEMD [8] method to remove the need for a reference matrix [2] during embedding.

In 2009, Chao *et al.* proposed diamond encoding based on the EMD method (DEMD) [3]. It can embed $(2k^2+2k+1)$-ary secret information into a pair of pixels and the embedding capacity can enhanced by increasing parameter k. Although the DEMD method improves the embedding capacity of the EMD method, there is more stego image distortion when parameter k is larger. As with FEMD, the DEMD method needs to calculate the diamond characteristic value (DCV) matrix [3] and look up the modification of pixels in the DCV matrix.

To overcome the above disadvantages and simplify the embedding procedures, a formula diamond encoding data hiding scheme is proposed in this paper. This will allow us to embed secret information directly without any reference matrix. According to the simulation results, the proposed method maintains a high payload, provides acceptable image quality and, most importantly, does not require a reference matrix.

The remainder of this paper is organized as follows: we review the EMD and DEMD data hiding schemes briefly in Section 2. Then, we discuss the proposed method and show the simulation in Section 3. Sections 4 and 5 give a security analysis and concluding remarks, respectively.

2. **Data hiding scheme review.**

In this section, we will review the two data hiding schemes based on the EMD and DEMD methods. In particular, the DEMD method uses a reference matrix to embed secret information in special modulus spaces into adjacent pixels. There are three steps in the DEMD method. First, we must define the extraction function and calculate the reference matrix using a parameterized function. In the embedding phase, we will calculate the value of the function using cover pixels and look up the value in the reference matrix. Then, we use this cover pixel pair as the center, look up the value of the secret data in the reference matrix and determine modification of pixels. Finally, we convert the cover pixels into stego-pixels.

2.1. **EMD method.**

In 2006, Zhang and Wang proposed a data hiding scheme based on EMD [21]. The major idea of this scheme was to embed a $(2n+1)$-ary secret information stream into non-overlapping n-pixels (Fig.1) with only one pixel in the block being increased or decreased by one. In addition, Zhang and Wang defined a new extraction

function $f_e(x_1, x_2, \cdots, x_n)$:

$$f_e(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} x_i \times i \ mod \ (2n+1), \tag{1}$$

where x_i is the i-th pixel value and n is the number of pixels. The EMD method determines the reference matrix (Fig.2) which represents the situation of 5-ary when $n = 2$ by the extraction function.

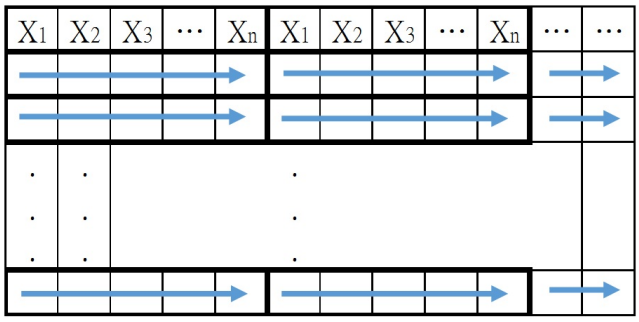


FIGURE 1. The embedding data sequence for EMD

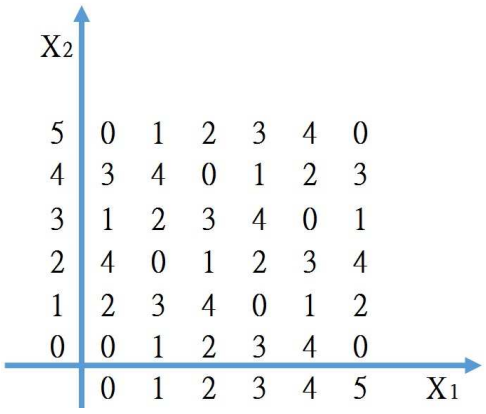


FIGURE 2. The extraction function $f_e(x_1, x_2) = (x_1 + 2x_2) \ mod \ 5$

The 5-ary data is embedded into two adjacent pixels using the reference matrix. That is to say, value of the 5-ary data can only be modified in four directions (top, bottom, left and right). The embedding algorithm is shown as follows:

Algorithm 1: The embedding process of EMD

Input: Cover image I_C and binary secret stream m

Output: Stego image I_S

EMD-1: Obtain an n-pixel block from I_C and secret digit s from m.

EMD-2: Calculate $t = f_e(x_1, x_2, \cdots, x_n)$ by Eq.(1).

EMD-3: Calculate the difference $d = (s - t) \ mod \ (2n + 1)$.

EMD-4: Get stego-pixels according to d.

$$\begin{cases} (y_1, y_2, \cdots, y_n) = (x_1, x_2, \cdots, x_n) & ,if \ d = 0 \\ (y_1, y_2, \cdots, y_d, \cdots, y_n) = (x_1, x_2, \cdots, x_d + 1, \cdots, x_n) & ,if \ d < n \\ (y_1, y_2, \cdots, y_{(2n+1)-d}, \cdots, y_n) = (x_1, x_2, \cdots, x_{(2n+1)-d} - 1, \cdots, x_n) & ,if \ d \geq n \end{cases}$$

EMD-5: Convert $(y_1, y_2, \cdots, y_n)$ into I_C to create I_S.

Example 1: If there are two cover pixels $(x_1, x_2) = (150, 151)$ and the 5-ary data s is $3_{(5)}$, stego-pixels $(151, 151)$ are obtained using Algorithm 1.

Step 1: Calculate the value of the extraction function with (x_1, x_2) which equals 2.
Step 2: Compute the difference $d = 3 - 2 = 1$.
Step 3: Find stego-pixel position $(x_1', x_2') = (151, 151)$.

According to Algorithm 1, this embedding procedure is very easy to implement. Unfortunately, the embedding payload of EMD is only $log_2(2n+1)/n$ bpp and the largest hidden bit rate is 1.16bpp when $n = 2$. Moreover, the embedding capacity decreases when n increases.

2.2. **DEMD method.** To enhance the embedding capacity, the diamond encoding (DEMD) data hiding scheme was proposed by Chao *et al.* in 2009. The main advantage of the DEMD method is the change of the extraction function to Eq.(2). This allows DEMD to embed $(2k^2 + 2k + 1)$-ary data into two cover pixels by modifying at most one pixel, where k is the embedding parameter. Nevertheless, the DEMD method [3] needs a DCV matrix (Fig. 3) for embedding much like EMD. The DEMD extraction function is given as

$$f_d(x_1, x_2) = ((2k+1)x_1 + x_2) \ mod \ (2k^2 + 2k + 1), \tag{2}$$

where x_1 and x_2 are the values of the two pixels in each block and k is the parameter of DEMD method.

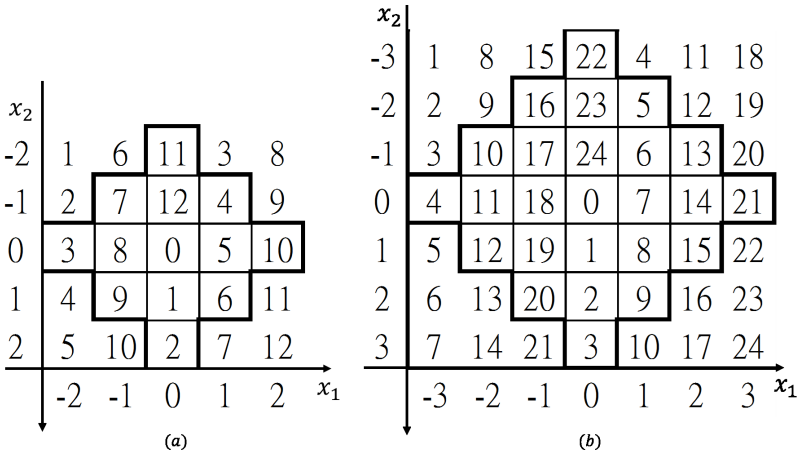


FIGURE 3. Diamond encoding patterns D_k with $k = 2$ (a) and $k = 3$ (b).

Before the data embedding procedure, the DEMD method determines parameter k and produces the DCV matrix with parameter k. Then, it calculates the value of the extraction function with two cover pixels. Finally, a $(2k^2 + 2k + 1)$-ary digit is embedded using the DCV matrix until all secret data is embedded.

Algorithm 2: The embedding process of DEMD
Input: Cover image I_C, parameter k and binary secret stream m
Output: Stego image I_S

DEMD-1: Select parameter k and transform secret data m into $(2k^2 + 2k + 1)$-ary.
DEMD-2: Obtain a block of two non-overlapping pixels from I_C.
DEMD-3: Compute the DCV matrix of the two pixel values by Eq.(2).
DEMD-4: Find stego-pixel values in the DCV matrix with s.
DEMD-5: Convert the cover pixels in I_C into stego-pixels to create I_S.

Example 2: For pixel pair $(2, 3)$, secret data $s = 9_{(25)}$ and parameter $k = 3$, the stego-pixel pair $(1, 4)$ is obtained by Algorithm 2.

Step 1: Compute $f_d(2,3) = (7 \times 2 + 3) \bmod 25 = 17$ and the DCV matrix of two pixel values by Eq.(2) with parameter $k = 3$.
Step 2: The secret data can be found at position $(1, 4)$ in the DCV matrix.
Step 3: The value of pixel pair $(2, 3)$ is replaced with $(1, 4)$.

In the DEMD method, the parameter k decides the embedding capacity and the stego image quality for each image. Embedding capacity can be computed by $\log_2(2k^2 + 2k + 1)/2$bpp. The embedding capacity of DEMD is larger than EMD. However, DEMD needs to compute and search the DCV matrix for each embedding procedure [3].

3. **The proposed formula DEMD scheme.** The DEMD scheme gives improved embedding capacity over EMD from 1.16bpp to $\log_2(2k^2 + 2k + 1)/2$bpp for a pixel pair. However, it needs more storage space to setup the reference matrix to record the relationships between pixels [3]. In order to find the stego-pixel pair without a reference matrix, we will propose an new embedding formula for DEMD scheme in this section.

3.1. **The formula embedding method.** Our proposed process produces at most four vectors for embedding embed. We calculate two vectors when $D \geq 0$. Then we calculate the other two vectors when $D < 0$. In Algorithm 3, i represents vectors 1 to 4 in turn.

Algorithm 3: The embedding process for our scheme
Input: A pixel pair (x_1, x_2), parameter k and secret data s
Output: Stego pixel pair $x_1^{'}$ and $x_1^{'}$

FDEMD-1: Set $f = ((2k+1)x_1 + x_2) \ mod \ (2k^2 + 2k + 1)$
FDEMD-2: Set $D = s - f$
FDEMD-3: If $D < 0$ then $D = D + (2k^2 + 2k + 1)$.
FDEMD-4: Set $next\_t_2 = |D| mod (2k + 1)$
FDEMD-5: While $i = 1$ to 4 do
Set $t_2 = next\_t_2$
Set $t_1 = (D - t_2)/(2k + 1)$
If $|t_1| + |t_2| \leq k$ Then
Set $x_1^{'} = x_1 + t_1$
Set $x_2^{'} = x_2 + t_2$
Return $(x_1^{'}, x_2^{'})$
Else
Switch (i)
Case 1:
Set $next\_t_2 = t_2 - (2k + 1)$
Case 2:
Set $D = D - (2k^2 + 2k + 1)$
Set $next\_t_2 = -(|D| \ mod \ (2k + 1))$
Case 3:
Set $next\_t_2 = t_2 + (2k + 1)$
Case 4:
Print 'Error'
End Switch
End If
End While

The embedding process overview is shown in Fig.4.

Example 3: For cover pixels pair $(10, 12)$, secret data $s = 20_{(25)}$ and $k = 3$, the stego-image pixels pair $(x_1^{'}, x_2^{'}) = (12, 11)$ is obtained by using Algorithm 3.

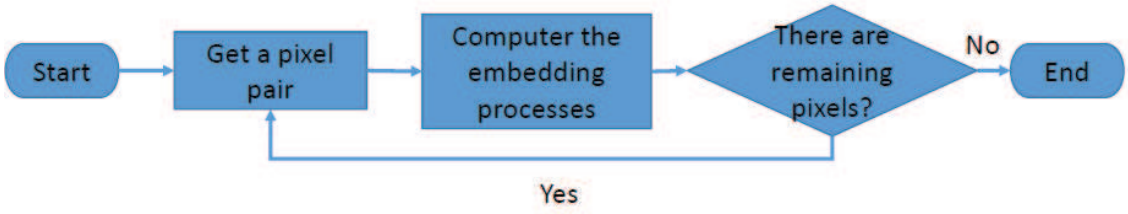


FIGURE 4. The flowchart of the proposed scheme

Step 1: Compute $f = (7 \times 10 + 12)\ mod\ 25 = 7$.
Step 2: Calculate $D = s - f = 13$.
Step 3: Compute $next\_t_2 = |D|\ mod\ (2k+1) = 6$ since $D > 0$.
1. Round 1: $t_2 = 6$ and $t_1 = 1$.
2. $|t_1| + |t_2| > k$ Then $next\_t_2 = t_2 - (2k+1) = -1$.
3. Round 2: $t_2 = -1$ and $t_1 = 2$.
4. $|t_1| + |t_2| \leq k$ then return $(12, 11)$.

3.2. **The extraction method.** As all EMD-type methods, we can easily extract the secret data by the extraction function. The secret data extraction algorithm is:
Algorithm 4: The extraction process for our scheme
Input: Stego image I_S
Output: Secret data

EDEMD-1: Obtain all 2-pixel blocks $(x_1^{'}, x_2^{'})$ from the stego image.
EDEMD-2: Calculate $s_i = f(x_1', x_2') = ((2k+1)x_1^{'} + x_2^{'})\ mod\ (2k^2 + 2k + 1)$ where i represents each block from EDEMD-1.
EDEMD-3: Concatenate all s_i become form binary stream m.

3.3. **Overflow problem and solution.** The original DEMD method will overflow with cover image pixels at minimum or maximum value. From our algorithm, we can get four vectors and choose suitable vectors for replacement. If overflow occurs we can choose the other vector to avoid overflow, i.e., the overflow problem is resolved in our approach.

4. Simulation and secure analysis.

4.1. **Simulation.** This simulation was performed using Matlab 2010a and was tested with eight 512×512 gray scale images as shown in Fig.5. The stego images when $k = 2$ and $k = 3$ are shown in Figs.6 and 7, respectively. The experimental results show no perceivable difference in appearance.

4.2. **Analysis.** The experimental results comparison between the DEMD scheme and our scheme is shown in Table 1. Our scheme has four parameters like the DEMD scheme and the results are the same as DEMD scheme. In particular, our approach uses a mathematical method to embed secret data. Therefore, we do not need any space to store a matrix. However, it cannot increase payload size or quality because we only proposed a formula for embedding. The relationship between the embedding payload and image quality for our method is shown in Fig.8.

4.3. **Security.** The proposed scheme is secure against the steganalysis approaches like Chi-square attack [20] and RS attack [5]. The Chi-square attack was proposed by Westfeld *et al.* in 1999 and the RS attack was proposed by Fridrich *et al.* in 2001.

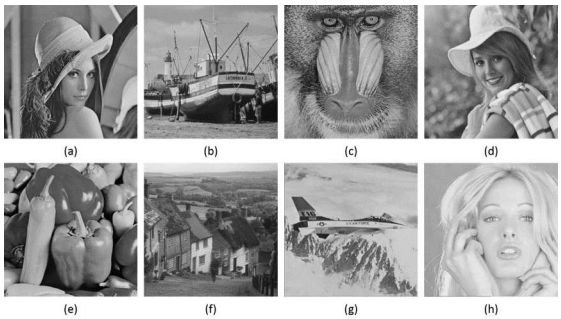


FIGURE 5. The eight gray cover images (a)Lena (b)Boat (c)Baboon (d)Elaine (e)Pepper (f)Goldhill (g)Airplane (h)Tiffany

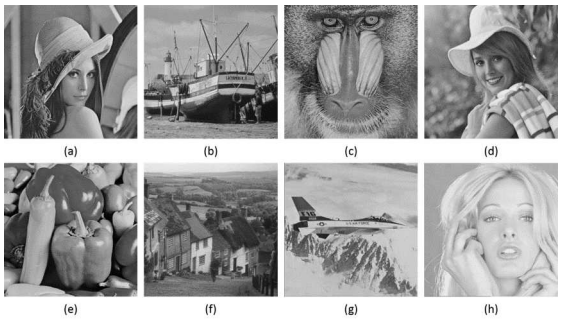


FIGURE 6. The eight gray stego images ($k = 2$; PSNR$\approx$ 47.8dB)

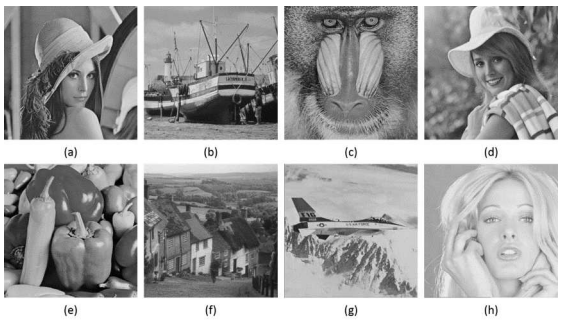


FIGURE 7. The eight gray stego images ($k = 3$; PSNR$\approx$ 45dB)

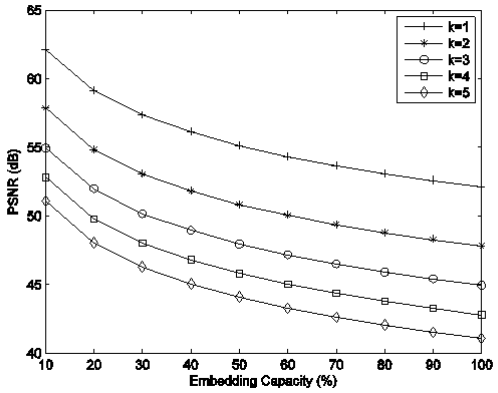


FIGURE 8. The relationships between embedding payload and image quality for our schemes.

TABLE 1. Comparisons between DEMD scheme and Ours

Items	DEMD scheme [3]	Our scheme
Embedding method propose the exact solution	Matrix and Search	Mathematic method
Parameter k	1 - 4	1 - 4
Payload (bpp)	1.16 - 2.68	1.16 - 2.68
PSNR (dB)	52.1 - 42.9	52.1 - 42.9
Need the storage space	Yes	No

4.3.1. *Chi-square attack.* Our proposed embedding function overwrites the least significant bit transforms values into each other like the LSB and LSB matching methods, etc. If the distribution of the secret data used for overwriting the least significant bit is equal to the random distribution of data encryption. That is to say, the LSB will tend to be closer to a 50% distribution in adjacent pixels. For these pixels, half of the LSBs will be 0 and half will be 1, are called a Pair of Values (PoVs) as Fig.9. And the frequencies of both values of each PoVs will become equal.

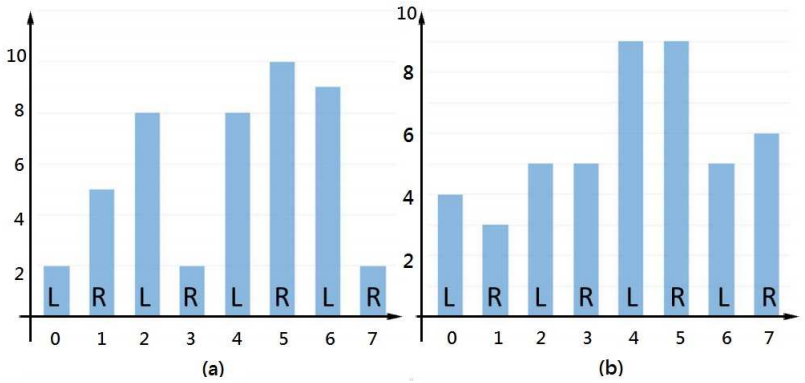
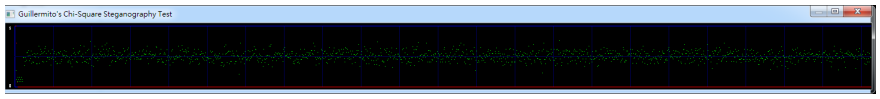
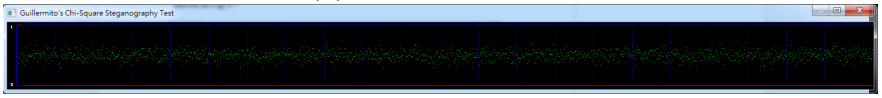


FIGURE 9. Histogram of luminance values before and after embedding a secret data with SB; (a) before embedding; (b) after embedding.

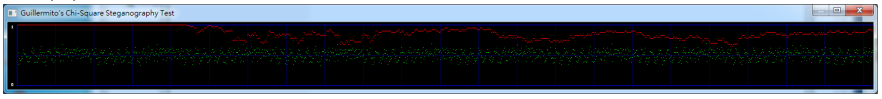
Therefore, the idea of the chi-square attack is used statistics to compare the theoretically expected frequency distribution in the stego image with some sample distribution observed in the possibly changed carrier image. In this experiment, we used the program from "A few tools to discover hidden data" toolbox [18] and the results are as shown in Fig.10.



(a)The cover image



(b)The stego image by our scheme when parameter $k = 2$.



(c)The stego image by LSB method.

FIGURE 10. The experimental results of chi-square attack

In Fig.10, the red curves are the result of the chi-square attack. The probability for a random embedded message is high if the curve is close to one. The green points are the average value of the LSBs on the current block of 128 bytes. They will form a pattern around 0.5 if there is a message embedded. In the results of Fig.10, we can see the red curves of our scheme are close to zero like the cover image. This shows that payload information is not detected by chi-square attack when encrypted using our proposed method.

4.3.2. *RS attack.* The RS attack method proposed by Fridrich *et al.* can directly detect secret data in the stego image. The main idea is to use statistical methods to classify each pixel block into three groups (the regular group R_m, the singular group S_m, and the unusable group U_m) by a flipping function and mask M. Similarly, this method also can classify each block into three groups (R_{-m}, S_{-m}, and U_{-m}) with inverse mask $-M$. The RS attack results are shown in Fig.11 where the x-axis represents the embedding rate and the y-axis represents the percentage of groups R_m, R_{-m}, S_m and S_{-m} with $M = [0\ 1\ 1\ 0]$ and $-M = [0\ -1\ -1\ 0]$. Note our stego image will pass the RS attack when $R_m \simeq R_{-m}$ and $S_m \simeq S_{-m}$ which shows our scheme is resistant to the RS attack.

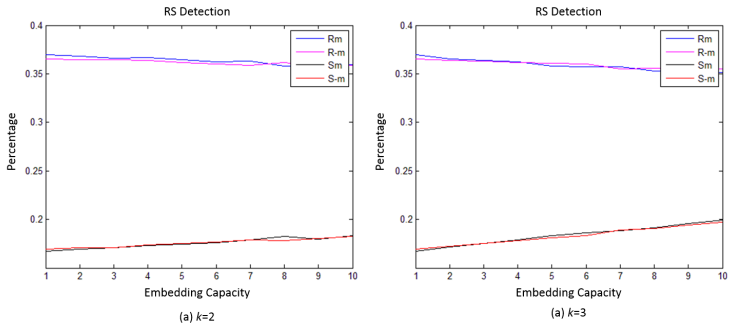


FIGURE 11. The simulation result with difference parameter k.

5. **Conclusion.** A formula diamond encoding data hiding scheme was proposed in this work to improve upon the DEMD method by removing the need for a reference matrix during embedding. In our proposed method, the secret data can be embedded by our formula with difference parameter k. According to the simulation results, our proposed scheme maintains good embedding capacity and stego image quality compared to the DEMD method.

6. **Acknowledgements.** The authors extend appreciation to the anonymous reviewers for providing invaluable comments during the revision process. This work was supported by MOST grant number MOST 104-2221-E-492 -014 -MY2.

REFERENCES

[1] Advanced Encryption Standard, NIST FIPS PUB 197, 2001.
[2] J. Chen, A PVD-based data hiding method with histogram preserving using pixel pair matching, *Signal Processing: Image Communication*, vol.29, no.3, pp. 375-384, 2014.
[3] R. M. Chao, H. C. Wu, C. C. Lee, and Y. P. Chu, A novel image data hiding scheme with diamond encoding, *EURASIP Journal on Information Security*, pp.1-9, 2009.
[4] Data Encryption Standard, NIST FIPS PUB 46-2, 1993.
[5] J. Fridrich, M. Goljan and R. Du, Detecting LSB steganography in color, and gray-scale images, *MultiMedia, IEEE*, vol.8, no.4, pp.22-28, Oct. 2001.

[6] H. C. Huang and F. C. Chang, Hierarchy-based reversible data hiding, *Expert Systems with Applications*, vol. 40, no. 1, pp. 34-43, Jan. 2013.
[7] H. C. Huang and W. C. Fang, Authenticity preservation with histogram-based reversible data hiding and quadtree concepts, *Sensors*, vol. 11, no. 10, pp. 9717-9731, Oct. 2011.
[8] T. D. Kieu and C. C. Chang, A steganographic scheme by fully exploiting modification directions, *Expert Systems with Applications*, vol.38, no.8, pp. 10648 - 10657, 2011.
[9] H. J. Kim, C. Kim, Y. Choi, S. Wang and X. Zhang, Improved modification direction methods, *Computers and Mathematics with Applications*, vol.60, no.2, pp.319 - 325, 2010.
[10] W. C. Kuo and M. C. Kao, A steganographic scheme based on formula fully exploiting modification directions, *IEICE Transactions on Fundamentals of Electronics*, vol.E96-A, no.11, pp. 2235-2243, Nov. 2013.
[11] W. C. Kuo, L. C. Wuu, C. N. Shyi and S. H. Kuo, A data hiding scheme with high embedding capacity based on general improving exploiting modification direction method, *Proceedings of the 2009 Ninth International Conference on Hybrid Intelligent Systems - Volume 03*, pp.69-72, 2009.
[12] C. F. Lee, C. C. Chang, P. Y. Pai, and C. M. Liu, An adjustable and reversible data hiding method based on multiple-base notational system without location map, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 1-28, Jan. 2015.
[13] C. F. Lee, Y. R. Wang and C. C. Chang, A steganographic method with high embedding capacity by improving exploiting modification direction, *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007*, pp. 497-500, Nov, 2007.
[14] J. Marin and F.Y. Shih, Reversible data hiding techniques using multiple scanning difference value histogram modification, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 3, pp. 451-460, Jul. 2014.
[15] J. Mielikainen, LSB matching revisited, *Signal Processing Letters, IEEE*, vol.13, no.5, pp.285-287, May 2006.
[16] R. L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM*, vol.21, no.2, pp. 120-126, Feb. 1978.
[17] H. M. Sun, C. Y. Weng, S. J. Wang and C. H. Yang, Data embedding in image-media using weight-function on modulo operations, *ACM Trans. Embed. Comput. Syst.*, vol.12, no.2, pp.21:1–21:12, Feb. 2013.
[18] Guillermito, Steganalysis Tool: A few tools to discover hidden data, http://www.guillermito2.net/stegano/tools/index.html, Sep. 2004.
[19] L.F. Turner, Digital data security system, Patent IPN, 1989.
[20] A. Westfeld and A. Pfitzmann, Attacks on Steganographic Systems, *Information Hiding 2000*, LNCS-1768, pp.61-76,Springer Berlin Heidelberg, 2000.
[21] X. P. Zhang and S. H. Wang, Efficient steganographic embedding by exploiting modification direction, *Communications Letters, IEEE*, vol.10, no.11, pp. 781-783, Nov. 2006.