Secure Modulus Data Hiding Scheme

Wen-Chung Kuo

Department of Computer Science and Information Engineering,
National Yunlin University of Science & Technology
Yunlin, Taiwan, R.O.C.
[e-mail: simonkuo@yuntech.edu.tw]

Received October 11, 2012; revised December 19, 2012; revised January 18, 2013; revised February 19, 2013; accepted February 23, 2013; published March 29, 2013

Abstract

In 2006, Zhang and Wang proposed a data hiding scheme based on Exploiting Modification Direction (EMD) to increase data hiding capacity. The major benefit of EMD is providing embedding capacity greater than 1 bit per pixel. Since then, many EMD-type data hiding schemes have been proposed. However, a serious disadvantage common to these approaches is that the embedded data is compromised when the embedding function is disclosed. Our proposed secure data hiding scheme remedies this disclosure shortcoming by employing an additional modulus function. The provided security analysis of our scheme demonstrates that attackers cannot get the secret information from the stegoimage even if the embedding function is made public. Furthermore, our proposed scheme also gives a simple solution to the overflow/underflow problem and maintains high embedding capacity and good stegoimage quality.

Keywords: Data hiding, Modulus method, Cryptography, Steganography, Least significant bit replace method.

http://dx.doi.org/10.3837/tiis.2013.03.011
1. Introduction

Due to the Internet’s technological advances, digital multimedia transmission speeds have continued to increase. At the same time, digital multimedia is often transmitted through insecure public channels where there exist many attacks such as illegal copying, forgery and cheating. Therefore, how to protect digital data becomes a very important issue. In general, two common methodologies are used to protect data, i.e., cryptography and steganography. Using cryptography, we can secure data by using encryption methods such as DES[15] or RSA[13]. In the transmission process, the message can be secured though the encrypted message is transformed into a length of illegible cryptotext and may attract unwanted attention. If this encrypted message is somehow decrypted, then there is no security provided to the original data. Another method, steganography, is used to protect digital multimedia security and intellectual property rights. Steganography hides secret data within meaningful host data to avoid the attention of would-be observers. Presently, there are many researchers [3-12, 14, 16, 17, 19] which propose to hide secret data within a meaningful image so that casual observance would not reveal the existence of hidden data. Therefore, the major goal of a data hiding scheme is not only to enhance the embedding capacity but also to maintain the quality of the stegoimage.

The least significant bit replacement method (LSB-R) is a common and easy data hiding technology proposed by Turner [16] in 1989. The general approach is that the secret data is embedded into the kth bit (where 1 ≤ k ≤ 8) of each pixel of the cover image. The stegoimage quality for LSB-R is acceptable since it has been determined that human perception cannot detect secret data embedded in the cover image when k ≤ 3 (i.e. the 3 least significant bits). However, it has been shown that LSB-R is very easily detected because there is an asymmetric characteristic in this method [2].

In order to improve LSB-R’s drawbacks, LSB Matching method (LSB-M)[14] was proposed by Sharp. In LSB-M, the pixel value of the cover image is incremented by 1 or decremented by 1 randomly when the secret data is not equal to the LSB of the cover image. Hence, the embedding capacity of LSB-M and LSB-R are equal but the LSB-M method is more complex than LSB-R. In 2006, Mielikainen proposed the LSB Matching Revisited method [11] (LSB-M-R) to enhance the embedding capacity of LSB-M. The major contribution of this scheme is the Expected Number of Modifications per Pixel (ENMPP) is smaller than LSB-M. Note that stegoimage quality is inversely proportional to ENMPP. In other words, stegoimage quality is better when ENMPP is smaller. Specifically, the ENMPP value of LSB-M-R is 0.375 and the ENMPP of LSB-M is 0.5. So, the stegoimage quality of LSB-M-R is better than LSB-M. However, the embedding rate for both LSB-M and LSB-M-R is on average at most 1 bit(s) per pixel (bpp) which is very poor in terms of embedding capacity. Recently, Chan et al. [1] proposed an image hiding scheme based on circular use of the exclusive OR (XOR) operator. In their scheme, the authors guarantee that only one pixel at most is required to be modified by adding/subtracting its value to/from one, and three secret bits can be embedded in three pixels. However, in some cases when three secret bits are embedded, two pixels will be modified. These cases include 0, 255 or if the number of different values between the calculated XOR values and the secret bits is greater than 2 in a group.

In 2006, Zhang and Wang proposed an exploiting modification direction (EMD) method [19] to increase data hiding capacity. The EMD scheme uses the relationship of n adjacent pixels to embed the secret data. That is to say, the secret binary data stream will be organized
into blocks and transformed into a \((2n + 1)\)-ary. Therefore, the secret will be embedded into \(n\) adjacent pixels where \(n>1\). For example, secret data can be embedded in two adjacent pixels, i.e., only one of two adjacent pixels is modified in the EMD scheme – by adding one, subtracting one, or staying the same. From a spatial point of view, two adjacent pixels can only have five orientations – moving upward, downward, left, right, or not moving at all. From their experimental results, Zhang and Wang claimed that the EMD scheme can enhance the capacity of the secret message and maintain good stegoimage quality. In 2007, Lee et al. [8] proposed an improved EMD scheme (HC-EMD) which enhanced the embedding ratio. The major idea of HC-EMD is to use both adjacent pixels at the same time and increase the resulting possibilities from five to eight. According to their experimental results, HC-EMD shows an improvement of 1.5\(x\) more capacity than EMD. Although HC-EMD’s embedding capacity is better than the EMD scheme, they do not account for when the extraction function becomes public. In other words, the secret data is disclosed when the embedding function is known. In addition, to avoid the overflow/underflow problems, Lee et al. make all pixels conform to the range of \([0, 255]\) but they do not propose any approach to do so.

Since the benefit of EMD is providing embedding capacity greater than 1 bit per pixel, many EMD-type data hiding schemes [4-6, 9, 10, 17] have been proposed previously. However, there is a serious disadvantage common to these approaches as the embedded data is compromised when the embedding function, with fixed weighting parameters and modulus, is disclosed. Note the previous work in [5] allowed for preshared weighting parameters, but still did not solve the problem of total disclosure of said parameters. In order to maintain embedded data security and improve the overflow/underflow problem, we propose a secure data hiding scheme by employing an additional modulus function. In the proposed scheme, the attacker cannot get secret information from the stegoimage even if the extraction function is made public. According to our experimental results, we can guarantee that our proposed scheme not only maintains the embedded data security but also solves the overflow/underflow problem while providing high embedding capacity and good stegoimage quality.

This paper is organized as follows: Section 2 will introduce the EMD and HC-EMD schemes. Then, we propose our secure data hiding scheme in Section 3 and give experimental results in Section 4. Finally, conclusions will be provided in Section 5.

### 2. Review Exploiting Modification Direction Techniques

#### 2.1 EMD Data Hiding Scheme

In 2006, an efficient data hiding scheme based on the exploiting modification direction method was introduced by Zhang and Wang [19]. The main characteristic of the EMD scheme uses the relationship of \(n\) adjacent pixels to embed \((n - 1)\)-ary secret data stream. For instance, a \(5\)-ary secret data stream will be embedded into two adjacent pixels, i.e., it modifies each of two adjacent pixels in the EMD scheme by adding one, subtracting one, or allowing it to stay the same. For the conversion and pixel group, Zhang and Wang defined the extract function \(f_e(\cdot)\) as the following:

\[
f_e(p_1, p_2, \ldots, p_n) = [\sum_{i=1}^{n} (p_i \times i)] \mod (2n + 1),
\]

where \(p_i\) is the \(i^{th}\) pixel value, and \(n\) as the number of pixels. For EMD, we have made a 2D hyper-cube (Fig.1) that reflects the situation of \(5\)-ary when \(n=2\). From Fig.1, we can visualize the physical meaning of EMD extract function: in all four directions (top, bottom, left, and right) of any numbers from 1 to 4, you can find 3 other different numbers in the chart. For
example, hiding data 3 is to be embedded at position \((p_1, p_2) = (2, 4)\). However, the original data at position \((2, 4)\) is 0, so we replace the original position \((p_1, p_2) = (2, 4)\) by \((p_1, p_2) = (2, 3)\) to complete the data hiding.

Some notations are defined for introducing the EMD scheme.

**I\(_C\):** The grayscale cover image.

\(O_{EMD}()\): Obtain all \(n\)-tuples \((p_1, p_2, \ldots, p_n)\) from partitioning the image \(I_C\) into the non-overlapping \(n\)-pixel blocks by scanning each row from left to right and top-down, as shown in Fig. 2.

**Algorithm EMD (Embedding Algorithm for EMD Scheme):**

Input: the cover image \(I_C\) and \((2n+1)\)-ary secret data.

Output: the stegoimage \(I_S\).

(EMD-1): Obtain all \(n\)-pixel blocks \((p_1, p_2, \ldots, p_n)\) from \(I_C\) and \(O_{EMD}(I_C)\).

(EMD-2): For each block \((p_1, p_2, \ldots, p_n)\) do the following:

- \{(Calculate \(t = f_c(p_1, p_2, \ldots, p_n)\),
  
  Compute \(d = (s - t) \text{ mod } (2n+1)\) when embed the secret data is \(s\),
  
  If \((d = 0)\), then \((y_1, y_2, \ldots, y_n) = (p_1, p_2, \ldots, p_n)\)
  
  Else \{if \((n \geq d)\), then \((y_1, y_2, \ldots, y_d, \ldots, y_n) = (p_1, p_2, \ldots, p_d+1, \ldots, p_n)\),
  
  Else \((y_1, y_2, \ldots, y_{(2n+1)-d'}, \ldots, y_n) = (p_1, p_2, \ldots, p_{(2n+1)-d'}, \ldots, p_n)\)\}

(EMD-3): Modify the \((p_1, p_2, \ldots, p_n)\) in \(I_C\) by \((y_1, y_2, \ldots, y_n)\) to create \(I_S\).

Example 1. Let adjacent four pixels \((p_1, p_2, p_3, p_4) = (131, 128, 130, 129)\) and secret data \(s =
(101)_2. Using the EMD scheme and following steps, we find the four steego pixels \((y_1, y_2, y_3, y_4) = (130, 128, 130, 129)\).

**Step 1.** Convert secret data \(s = (101)_2 = (5)_{10}.

**Step 2.** Compute \(f_s(131, 128, 130, 129) = 6 \mod 9.

**Step 3.** Compute the difference value \(d = (5-6) \mod 9 = 8 \mod 9.

**Step 4.** Get \((y_1, y_2, y_3, y_4) = (130, 128, 130, 129).

From the theoretical estimation, the embedding capacity of EMD is \((\log_2(2n+1))/n\) bpp. However, the best data hiding bit rate (1 bpp) exists when it is 5-ary, i.e., \(n=2\). When \(n\) increases, the number of pixels in a group increases, and the hiding bit rate is decreased \([19]\).

### 2.2. High Embedding Capacity by Improving EMD scheme

In 2007, Lee et al. provided high embedding capacity by the improving EMD scheme\([8]\]. Their main contribution is that the embedding capacity of HC-EMD (1.5 bpp) is better than the EMD scheme (1 bpp). There are two major differences between the EMD and HC-EMD schemes. The first difference is their extraction functions, i.e., the extraction function of the EMD scheme for 2-tuples \((p_1, p_2)\) is \(f_s(p_1, p_2) = [1 \times p_1 + 2 \times p_2] \mod 5\) and the extraction function for the HC-EMD scheme for 2-tuples \((p_1', p_2')\) is \(f_s(p_1', p_2') = [1 \times p_1' + 3 \times p_2'] \mod 8\). The other is the embedding data of these two schemes are 5-ary and the octal number system, respectively.

The following notations are defined before we introduce the HC-EMD scheme.

- **\(I_c\):** The grayscale cover image.
- **\(O_{HC-EMD}(\cdot)\):** Obtain all 2-tuples \((p_1, p_2)\) from partitioning the image \(I_c\) into non-overlapping 2-pixel blocks by scanning each row from left to right and top-down, as shown in Fig.3.

![Fig. 3. The embedding data sequence](image)

**Algorithm HC-EMD (Embedding Algorithm for HC-EMD Scheme):**

**Input:** the cover image \(I_c\) and secret data \(s\)

**Output:** the stegoimage \(I_s\)

**HC-EMD-1:** Obtain all 2-pixel blocks \((p_1', p_2)\) from \(I_c\) and \(O_{HC-EMD}(I_c)\).

**HC-EMD-2:** For each block \((p_1', p_2)\) do the following:

\{Calculate \(t = f_s(p_1', p_2) = 1 \times p_1 + 3 \times p_2 \mod 8,\)

If \((t = s)\), then \((y_1, y_2) = (p_1', p_2),\)

Else if \(s = t + 1 = f_s(p_1 + 1, p_2)\) then \(y_1 = p_1 + 1, \ y_2 = p_2,\)
Else if \( s = t + 2 = f_h(p_1 - 1, p_2 + 1) \), then \( y_1 = p_1 - 1, \ y_2 = p_2 + 1 \).

Else if \( s = t + 3 = f_h(p_1, p_2 + 1) \), then \( y_1 = p_1, \ y_2 = p_2 + 1 \).

Else if \( s = t + 4 = f_h(p_1 + 1, p_2 + 1) \), then \( y_1 = p_1 + 1, \ y_2 = p_2 + 1 \).

Else if \( s = t + 5 = f_h(p_1, p_2 - 1) \), then \( y_1 = p_1, \ y_2 = p_2 - 1 \).

Else if \( s = t + 6 = f_h(p_1 + 1, p_2 - 1) \), then \( y_1 = p_1 + 1, \ y_2 = p_2 - 1 \).

Else if \( s = t + 7 = f_h(p_1 - 1, p_2) \) then \( y_1 = p_1 - 1, \ y_2 = p_2 \).

(HC-EMD-3): Modify \((p_1, p_2)\) in \( I_c \) by \((y_1, y_2)\) to create \( I_s \).

The HC-EMD extract function value \( f_h(p_i, p_{i+1}) \) can be represented by the value of the \( p_i \)th row and the \( p_{i+1} \)th column in the matrix \( R \). Therefore, we let \( R[p_i][p_{i+1}] \) be the center of a \( 3 \times 3 \) block where there are eight different surrounding values, \( R[p_i+1][p_{i+1}], \ R[p_i-1][p_{i+1}], \ R[p_i][p_{i+1}+1], \ R[p_i][p_{i+1}-1], \ R[p_i+1][p_{i+1}+1], \ R[p_i+1][p_{i+1}-1], \ R[p_i-1][p_{i+1}+1] \) and \( R[p_i-1][p_{i+1}-1] \). For example, let \( p_1 = 158 \) and \( p_{i+1} = 73 \), i.e., \( R[158][73] = f_h(158, 73) = 1 \). Then, eight different values around \( R[158][73] \) are shown as Fig. 4. However, there are two major problems in the HC-EMD scheme. One is the security of the HC-EMD scheme is dependent on the extraction function. In other words, the secure data embedded in the HC-EMD scheme will be disclosed when the extraction function is made public. The other problem is that the solution to the overflow/underflow problem is not discussed in detail[8]. In other words, Lee et al. just assumes all pixels conform to the range of \([0, 255]\) to avoid overflow/underflow problems but they do not propose any practical approaches.

![Fig. 4. Matrix R](image-url)

3. The Proposed Secure Steganographic Method

In order to maintain the same embedding capacity and improve the overflow/underflow problem, we propose a secure modulus data hiding scheme(M-EMD scheme) in this section. First, a new modified extract function \( f_m \) is defined as Eq.(2):

\[
f_m(g'_1, g'_2) = \lfloor 1 \times (g'_1) + 3 \times (g'_2) \rfloor \mod 8
\]

where \( g_1 \) and \( g_2 \) are two adjacent pixels. If \( g_1 \mod 3 = 2 \) then \( g'_1 = -1 \), else \( g'_1 = 2 \) for \( i \in \{1, 2\} \)[18]. There are three phases (blocking and encoding phase, embedding phase, and extraction phase) included in our proposed scheme.

3.1 Blocking and Encoding Phase

The secret data stream will be divided into three binary bits per block and embedded into two
neighbor pixels of the cover image. Then, we convert these three bits into the decimal number \( s \) (i.e. \( s \in [0, 7] \)) and encode \( s \) by using Table 1 before the embedding phase.

**Table 1.** The secret number \( s \) encoding table

<table>
<thead>
<tr>
<th>( s )</th>
<th>( (g'_1, g'_2) )</th>
<th>( s )</th>
<th>( (g'_1, g'_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0, 0)</td>
<td>4</td>
<td>(1, 1)</td>
</tr>
<tr>
<td>1</td>
<td>(1, 0)</td>
<td>5</td>
<td>(0, 2) ( \equiv (0, -1) )</td>
</tr>
<tr>
<td>2</td>
<td>(2, 1) ( \equiv (-1, 1) )</td>
<td>6</td>
<td>(1, 2) ( \equiv (1, -1) )</td>
</tr>
<tr>
<td>3</td>
<td>(0, 1)</td>
<td>7</td>
<td>(2, 0) ( \equiv (-1, 0) )</td>
</tr>
</tbody>
</table>

### 3.2 Embedding Phase

The following notations are defined before the M-EMD scheme is proposed.

- \( I_C \): The grayscale cover image.
- \( O_{M-EMD}() \): Obtain all 2-tuples \((p_1, p_2)\) from partitioning the image \( I_C \) into non-overlapping 2-pixel blocks by scanning each row from left to right and top-down, as shown in Fig. 3.
- \( O_{M-E-EMD}() \): Obtain all 2-tuples \((y_1, y_2)\) from partitioning the stegoimage \( I_s \) into non-overlapping 2-pixel blocks by scanning each row from left to right and top-down.

**Algorithm M-EMD (Embedding Algorithm for M-EMD Scheme):**

**Input:** the cover image \( I_C \) and secret data \( S \)

**Output:** the stegoimage \( I_s \)

- (M-EMD-1): Obtain all 2-pixel blocks \((p_1, p_2)\) from \( I_C \) and \( O_{M-EMD}(I_C) \).
- (M-EMD-2): For each block \((p_1, p_2)\) do the following:
  
  1. Encode the secret data \( S \) and get \((g'_1, g'_2)\) by using Table 1,
  2. Compute \((p_1 \mod 3, \ p_2 \mod 3) = (b_1, b_2)\) and the difference value \((d_1, d_2) = (g'_1 - b_1, g'_2 - b_2)\).
  3. If \((d_1, d_2) = (0, 0)\), then \((y_1, y_2) = (p_1, p_2)\), else \((y_1, y_2) = (p_1 + d_1, p_2 + d_2)\).

- (M-EMD-3): Modify the \((p_1, p_2)\) in \( I_C \) by \((y_1, y_2)\) to create \( I_s \).

**Example 2.** Let adjacent two pixels \((p_1, p_2) = (131, 124)\) and secret data \( s = (101)_2 = (5)_{10} \).

Using the M-EMD scheme and the following steps, we find the stego pixel pair \((y_1, y_2) = (132, 125)\).

**Step 1.** Convert secret data \( s = (101)_2 = (5)_{10} \) and encode it using Table 1 to get \((g'_1, g'_2) = (0, 2)\).

**Step 2.** Compute \((131 \mod 3, 124 \mod 3) = (2, 1) \neq (0, 2)\).

**Step 3.** Compute the difference value \((d_1, d_2) = (-2, 1) = (1, 1)\).

**Step 4.** Get \((y_1, y_2) = (132, 125)\).

### 3.3 Extraction Phase

**Algorithm M-E-EMD (Extract Algorithm for M-EMD Scheme):**
Input: the stegoimage $I_s$
Output: the secret data $S$

(M-E-EMD-1): Obtain all 2-pixel blocks $(y_1, y_2)$ from $I_s$ and $O_{M=E-EMD}(I_s)$.

(M-E-EMD-2): For each block $(y_1, y_2)$ do the following:
- Compute $(y_1 \mod 3, y_2 \mod 3) = (g'_1, g'_2)$.
- Recover the secret data $(d)_{10}$ from $f_m(g'_1, g'_2)$.
- Convert decimal number $(d)_{10}$ to binary $(s)_{2}$.

(M-E-EMD-3): Concatenate $(s)_{2}$ from each block to recover the original secret data $S$.

**Example 3.** Let two adjacent stego pixels $(y_1, y_2) = (132, 125)$. We recover the secret data $(101)_2$ from $(y_1, y_2)$ using following steps:

*Step 1.* Compute $(132 \mod 3, 125 \mod 3) = (0, 2)$.

*Step 2.* Using the Table 1, we calculate the secret data $d$ from $f_m(g'_1, g'_2) = 5$.

*Step 3.* Convert decimal number 5 to binary $(101)_2$.

### 3.4 The overflow/underflow Problem

A big problem in the data hiding scheme is the overflow/underflow problem. The overflow/underflow problem occurs when the calculated pixel value is not defined. For example, if the pixel’s value is bigger than 255 or smaller than 0 then no color is represented. Therefore, the overflow problem will occur in $p_i$ when 1 is added to 255 and underflow problem will occur when 1 is subtracted from 0. However, we can use the characteristics of $1 \equiv -2 \mod 3$ and $2 \equiv -1 \mod 3$ based on the modulus operation to avoid the overflow/underflow problem.

**Example 4.** Let two adjacent pixels $(p_1, p_2) = (254, 255)$ and the secret data $s = (010)_2 = (2)_{10}$.

Using the proposed data hiding method and following steps, we calculate the stego pixel pair $(y_1, y_2) = (254, 253)$.

*Step 1.* Convert the secret data $s = (010)_2 = (2)_{10}$ and encode it using Table 1 to get $(g'_1, g'_2) = (2, 1)$.

*Step 2.* Compute $(254 \mod 3, 255 \mod 3) = (2, 0) \neq (2, 1)$.

*Step 3.* Compute the difference value $(d_1, d_2) = (0, 1)$.

*Step 4.* The resulting stego pixel pair $(y_1, y_2) = (254, 253)$.

### 4. Experimental Results and Analysis

To compare the performance of HC-EMD and our method, the following experiment and analysis were implemented in Matlab 7.8 on an Intel®Core™ Duo 2.53GHz CPU PC with 1.93GB of memory running Windows XP Professional. The proposed scheme and the HC-EMD scheme were tested on five 512×512 grayscale images (Lena, Baboon, Airplane, Barbara and Goldhill) shown in Fig.5. The corresponding stegoimages are shown in Fig.6 and Fig.7. There is no perceivable difference in stegoimage appearance between the HC-EMD scheme and our proposed scheme.
The embedding capacity is 1.5 bpp for both the HC-EMD and our proposed scheme. In other words, there are 3 bits of secret data embedded into two pixels.

### 4.1 Experimental Results

In the HC-EMD scheme, for each block, the probability of \((y_1, y_2) = (p_1, p_2)\) is 0.125. In other words, the probability of \((y_1, y_2) \neq (p_1, p_2)\) is 0.875. Therefore, the probability for any pixel value being changed is \(\text{ENMPP}_{\text{HC-EMD}} = \frac{10}{16} = 0.625\). In our proposed scheme, for each block, the probability of \((y_1, y_2) = (p_1, p_2)\) is \(\frac{1}{8} = 0.125\) and the probability of \((y_1, y_2) \neq (p_1, p_2)\) is \(\frac{7}{8} = 0.875\). There are eight possible cases in \((g_1', g_2')\) and each case contains two variables resulting in 16 possible cases in total. The corresponding pixel value will be adjusted when \(d_1\) or \(d_2\) is not zero. According to our analysis, there are 12 cases when \(d_1\) or \(d_2\) is not zero. In other words, the probability that \(d_1\) or \(d_2\) is not zero is 12/16. Therefore, the probability for each pixel value to be modified in our proposed scheme is \(\text{ENMPP}_{\text{M-EMD}} = (\frac{7}{8}) \times (\frac{12}{16}) = 0.6563\). Hence, the stegoimage quality of the proposed scheme is slightly less than the HC-EMD scheme.
4.2 Security Analysis

In this subsection, we focus our discussion on the enhanced security features of the proposed scheme.

4.2.1 Embedding Algorithm

For the HC-EMD scheme, fixed coefficients are used in the extraction function. Therefore, it does not provide any security mechanism to resist reversal because the embedding procedure is described publicly. However, there are 5 different codes (shown as Table 2) used in our proposed scheme.
Table 2. The different coding tables

<table>
<thead>
<tr>
<th>d</th>
<th>((g'_1, g'_2))</th>
<th>Code1</th>
<th>Code2</th>
<th>Code3</th>
<th>Code4</th>
<th>Code5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0,0) or (2,2)</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(2,2)</td>
</tr>
<tr>
<td>1</td>
<td>(1,0) or (-1,-2)</td>
<td>(1,0)</td>
<td>(1,0)</td>
<td>(1,0)</td>
<td>(2,1)≡(-1,-2)</td>
<td>(1,0)</td>
</tr>
<tr>
<td>2</td>
<td>(2,0) or (-1,1) or (0,-2)</td>
<td>(2,0)</td>
<td>(2,1)≡(-1,1)</td>
<td>(0,1)≡(0,-2)</td>
<td>(0,1)≡(0,-2)</td>
<td>(2,0)</td>
</tr>
<tr>
<td>3</td>
<td>(0,1) or (1,-2)</td>
<td>(0,1)</td>
<td>(0,1)</td>
<td>(1,1)≡(1,-2)</td>
<td>(1,1)≡(1,-2)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>4</td>
<td>(1,1) or (-1,-1)</td>
<td>(1,1)</td>
<td>(1,1)</td>
<td>(2,2)≡(-1,-1)</td>
<td>(2,2)≡(-1,-1)</td>
<td>(1,1)</td>
</tr>
<tr>
<td>5</td>
<td>(2,1) or (0,-1)</td>
<td>(2,1)</td>
<td>(0,2)≡(0,-1)</td>
<td>(0,2)≡(0,-1)</td>
<td>(0,2)≡(0,-1)</td>
<td>(2,1)</td>
</tr>
<tr>
<td>6</td>
<td>(0,2) or (1,-1)</td>
<td>(0,2)</td>
<td>(1,2)≡(1,-1)</td>
<td>(1,2)≡(1,-1)</td>
<td>(1,2)≡(1,-1)</td>
<td>(0,2)</td>
</tr>
<tr>
<td>7</td>
<td>(1,2) or (-1,0)</td>
<td>(1,2)</td>
<td>(2,0)≡(-1,0)</td>
<td>(2,0)≡(-1,0)</td>
<td>(2,0)≡(-1,0)</td>
<td>(1,2)</td>
</tr>
</tbody>
</table>

Thus, even if the embedding algorithm is made public, the secret data still cannot be recovered from the stego image because the coding table is unknown.

Example 5. Let adjacent stego pixels \((y'_1, y'_2)\) be (132, 125). We can recover the different secret data 5 and 6 from Code2 and Code5, respectively.

4.2.2 Simple Solution to Overflow/underflow Problems

In order to prevent the overflow/underflow problem, Lee et al. makes all pixels conform to the range of \([0, 255]\), i.e., they may have to change the value of cover pixel before embedding secret data [8]. However, they do not propose any solution to explain this method. In this paper, we use the modulus characteristic to solve these problems in section 3.3.

Below, the functionality of the proposed scheme is compared with HC-EMD scheme and Kuo-Wang scheme. The HC-EMD scheme is a subcase of the Kuo-Wang scheme[4]. Here, we summarize the comparisons in Table 3.

Table 3. Functionality comparison table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding Algorithm can be public</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Solve the overflow/underflow problem</td>
<td>Pixel values are normalized to ([0, 255])</td>
<td>No</td>
<td>Based on modulus characteristic</td>
</tr>
<tr>
<td>Embedding method</td>
<td>Extraction function</td>
<td>Extraction function</td>
<td>Lookup table</td>
</tr>
<tr>
<td>Maintain the embedding message security</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Embedding rate</td>
<td>1.5 bpp</td>
<td>1.5 bpp</td>
<td>1.5 bpp</td>
</tr>
<tr>
<td>Average PSNR</td>
<td>50.2 dB</td>
<td>50.2 dB</td>
<td>49.9 dB</td>
</tr>
</tbody>
</table>

5. Conclusion

Previous EMD-type data hiding methods provided high secret message capacity and good stego image quality. However, a serious drawback of these existing schemes is that the
embedded data is revealed when details of the scheme are made public. Specifically, disclosing the embedding function and the parameters along with the modulus compromises the security of the scheme. In this paper, a secure modulus data hiding scheme is proposed. The major advantages of our scheme are preserving embedded data security when the embedding function is made public and also providing a simple solution to the overflow/underflow problem while maintaining higher embedding capacity and good stegoimage quality.

Acknowledgements

This work was supported by NSC 101-2221-E-150-076.

References


Wen-Chung Kuo received his B.S. degree in Electrical Engineering from National Cheng Kung University in 1990 and his M.S. degree in Electrical Engineering from National Sun Yat-Sen University in 1992. He received his Ph.D. degree from National Cheng Kung University in 1996. Currently, he is an associate professor in the Department of Computer Science and Information Engineering at National Yunlin University of Science & Technology. His research interests include steganography, cryptography, network security and signal processing.