A module-based LSB substitution method with lossless secret data compression

Shang-Kuan Chen*
Department of Computer Science and Information Engineering, Yuanpei University, Hsinchu, Taiwan, ROC

ARTICLE INFO

Article history:
Received 26 May 2009
Received in revised form 16 July 2010
Accepted 27 November 2010
Available online 19 January 2011

Keywords:
Module-based LSB substitution
Data hiding
Lossless compression

ABSTRACT

All the various data hiding methods can be simply divided into two types: (1) the extracted important data are lossy, (2) the extracted important data are lossless. The proposed method belongs to the second type. In this paper, a module-based substitution method with lossless secret data compression function is used for concealing smoother area of secret image by modifying fewer pixels in the generated stego-image. Compared with the previous data hiding methods that extract lossless data, the generated stego-image by the proposed method is always with better quality, unless the hidden image is with very strong randomness.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Data hiding [1–10] is a commonly used technique that embeds important data into the host images by modifying the pixels to protect the data from illegal peeking or damaging. In general, a stego-image is generated by hiding the important data into a host image. The data are transmitted much safely by using the stego-image than by transmitting the data themselves directly. We can categorize data hiding schemes into two types. In the first type, the data extracted from the generated stego-image are allowed with some distortions. In the second type, the extracted data cannot be allowed with any distortion. In [1,2], the extracted data were distorted, but these two methods are with extremely-high hiding capacity, for they could conceal data of very large size. For example, Chung et al. [1] hid the important image in a host image of the same size. Wang and Tsai [2] even hid a larger important image in a smaller host image. In both methods, the quality of the extracted image is acceptable, although not lossless. The LSB (Least-Significant Bit) substitution approach is a lossless approach. The simplest one is the so-called simple LSB substitution method which directly replaces the least-significant bit planes of the host image with hidden data. The related works [3–5] of LSB substitution method are stated in the next section.

In this paper, the repetition of the secret data is considered for reducing possible replacement of pixels in the stego-image. Hiding the secret data of heavy repetition will replace fewer pixels than that of light repetition by using the proposed method. Therefore, if the secret data is an image, in almost all the cases, the quality of the stego-images generated by the proposed method will be better than the current LSB-substitution hiding methods [3–5].

This paper is organized as follows. The related works are introduced in Section 2. The proposed method is stated in Section 3. Experimental results and analysis are shown in Sections 4 and 5, respectively. Finally, the conclusion is made in Section 6.

2. The related works

The related works [3–5] of simple LSB substitution method enable the stego-image to obtain better quality. Wang et al. [3] defined an optimal re-naming problem for the hidden data, and then used a genetic algorithm for searching the problem’s nearly optimal solution. Chang et al. [4] gracefully developed a dynamic programming strategy to replace Wang et al.’s [3] genetic algorithm to get a faster algorithm to obtain the aforementioned theoretical solution. Rather than finding the optimal re-naming solution, Thien and Lin [5] proposed a digit-by-digit data hiding method based on modulus function. The quality (PSNR values) of their stego-image outperformed that of Ref. [3,4], without damaging the hiding capacity or lossless recovery of the secret data. The method is also faster than the two methods [3,4].

Thien and Lin’s method [5] first partitions the secret data into a sequence of non-overlapping segments, and each segment \( m_i \) is in the range \( [0, 1, 2, \ldots, q - 1] \) where \( q \) is a given positive integer called the base for module function. A segment \( m_i \) can be hidden in the \( i \)th pixel (whose original gray value is \( p_i \)) of the host image. In the simple LSB substitution method, the \( i \)th pixel value of the generated stego-image is \( p_i′ \). The \( i \)th pixel value of the generated stego-image is

\[
p_i′ = p_i − (p_i \mod q) + m_i \quad \text{true} \quad \text{for each} \quad i
\]

In Thien and Lin’s method, \( p_i′ \) is adjusted further to a new number \( p_i″ \) that is closer to \( p_i \) and yet still congruent to \( p_i′ \) on the module base \( q \), i.e. \( m_i = p_i″ \mod q \). Therefore, the decoding (to get \( m_i \) from \( p_i″ \)) is simple and fast, and the impact to the host image is smaller.

To improve Thien and Lin’s method, the proposed method uses repetition property of secret data for decreasing the number of modified pixels without extra bit for recording the repetition. That is a
3. The proposed method

This section is divided into two subsections including encoding phase and decoding phase.

3.1. Encoding phase

The encoding phase is divided into two stages. In Stage 1, each datum is hidden in the pixel of host image to be a stego-pixel, while Stage 2 provides the pixel adjustment for obtaining nearest stego-pixel under the same datum extraction.

3.1.1. Stage 1: The hiding process

Let $s_i$ denote the $i$th secret datum ranging from 0 to $m - 1$, and each $s_i$ be hidden in a pixel of the host image. The stego-image is generated in a pixel-by-pixel manner. Each time when hiding a not-yet-hidden datum $s_i$, we also take next coming data simultaneously if their values are all identical to $s_i$, i.e., if $r$ contiguous data $s_i, s_{i+1},...$, and $s_{i+r−1}$ are all with the same value $z$ for some positive integer $r$. Originally, these $r$ data are supposed to be hidden in $r$ pixels $p_i, p_{i+1},..., p_{i+r−1}$ of the host image, respectively. To generate temporary stego-pixels, in the slight-repetition occurring, if $r = 1, 2, 3$, then for each $i = l, i + 1, ..., i + r − 1$, we assign the temporary stego-pixel $t_i$ to $p_i − \lfloor p_i \mod (m + 1) \rfloor + s$. In a numerical system with base $b = m + 1$, the available digits are $\{0, 1, 2, ..., m\}$. Because each datum ranges from 0 to $m - 1$, the only unused digit is $m$. This special digit $m$ (the so-called flag-digit) is especially reserved to indicate the special case when the repetition length is over 3, i.e., $r > 3$.

When a special heavy-repetition case occurs, besides using a flag digit (value $m$) for identifying the case, the repeated value $z$ and the repetition length $r (r > 3)$ also need to be recorded. A three-digit vector $\{f_i, g_i, r_i\}$ is adopted for describing easily. The third component is $r_i$ for identifying the case, the repeated value $z$, and the difference in order to save coding-length. Notably, in this operation, the maximal ratio of the number of unchanged pixels to total number of pixels is $(m + 1)/(m + 4)$. Of course, with slight modification, these unchanged pixels can also be used to record some other data. This slightly-modified version can either increase PSNR or increase the amount-of-hidden-data (the so-called hiding-capacity).

Let the ordered n-tuples $<h_1, h_2, ..., h_n>$ be the hidden data, the ordered n-tuples $(img_1, img_2, ..., img_n)$ be the pixels of an n-pixels image, and the ordered n-tuples $(d_1, d_2, ..., d_n)$ be the differences between the pixels of the host image and the stego-image. Assume that the data $<2, 4, 3, 3, 3, 3, 5, 1>$ are to be hidden in the ten pixels $(70, 67, 66, 61, 68, 64, 63, 60, 61, 58)$ of the host image. Since the data range from 0 to 5, $m$ is assigned to 6 to reduce the distortion. Thus, the base system is $m + 1 = 7$. The first two data are with no repetition; thus, the first two stego pixels are $63/7 = 2$ and $67/7 = 4$, where the operator $/$ is integer division. From the third datum to the seventh datum, the values are the same and the number of these data is larger than 3; the lossless compression of these data can be used. First, the number 6 should be hidden in the third pixel of the host image; then the datum 3 should be hidden in the fourth pixel; finally, the repetition length 5 is hidden in the fifth pixel by hiding $(5 − 4 = 1)$. Therefore, using the proposed method to encode, the generated temporary stego-pixels will be $[65, 67, 69, 59, 64, 64, 61, 57, 57]$. The differences between the host image and the temporary stego-image are thus $[5, 0, 3, 2, 4, 0, 0, 1, 4, 1]$. Notably, both the sixth and the seventh pixels of the host image and the temporary stego-image are the same, because the third, the fourth, and the fifth pixels of the temporary stego-image have already recorded the information of $<3, 3, 3, 3, 3>$ of the hidden data, by hiding a three-tuple $(6, z = 3, r = 4) = 1$ in the third, fourth, and fifth pixels of the host image. Here, 6 is the heavy-repetition-flag, $z = 3$ is the heavy-repeated-value, and $r + 4 = 1 + 4 = 5$ is the repetition length that $z$ is repeated.

3.1.2. Stage 2: The adjusting process

After the proposed hiding process, the difference between the pixel value $t_i$ of the temporary stego-image and the corresponding $p_i$ of the host image ranges from $−m$ to $m$. To reduce the distortion, this stage adjusts the stego-pixel value if there is another value that has the same extraction datum also closest to the pixel of the host image. The proposed method divides the possible situations into the following five cases:

Case 1: $(\lfloor (m + 1)/2 \rfloor − t_i − p_i) ≤ m$ and $t_i ≥ m + 1$.

$$g_i ← t_i − m − 1$$

Case 2: $− m ≤ t_i − p_i < − \lfloor (m + 1)/2 \rfloor$ and $t_i ≤ 254 − m$.

$$g_i ← t_i + m + 1$$

Case 3: $− \lfloor (m + 1)/2 \rfloor ≤ t_i − p_i ≤ \lfloor (m + 1)/2 \rfloor$.

$$g_i ← t_i$$

Case 4: $(\lfloor (m + 1)/2 \rfloor − t_i − p_i) ≤ m$ and $t_i ≤ m$. (Boundary)

$$g_i ← t_i$$

Case 5: $− m ≤ t_i − p_i < − \lfloor (m + 1)/2 \rfloor$ and $t_i ≥ 255 − m$. (Boundary)

$$g_i ← t_i$$

In the following, it is shown that the $|g_i − t_i| ≤ m/2$ in Cases 1 and 2.

Proposition 1. In Cases 1 and 2 of the proposed adjusting process, the difference of $g_i$ and $p_i$ is $|g_i − p_i| ≤ m/2$.

Proof. Recall that in Case 1, $g_i = t_i − m − 1$, and in Case 2, $g_i = t_i + m + 1$.

In Case 1:

$$g_i − p_i = t_i − m − 1 − p_i = (t_i − p_i) − m − m − 1 = − 1$$

$$g_i − p_i = t_i − m − 1 − p_i = (t_i − p_i) − m − 1 = − m/2 − 1$$

Because $p_i$ is integer, $g_i − p_i ≥ − m/2 − 1$. When $m$ is even, $g_i − p_i < \lfloor (m + 1)/2 \rfloor − m − 1 = m/2 − 1 = − m/2 − 1$. Therefore, $g_i − p_i$ is integer, $g_i − p_i ≥ − m/2 − 1$. When $m$ is odd, $g_i − p_i < \lfloor (m + 1)/2 \rfloor − m − 1 = (m + 1)/2 − m − 1 = − m/2$.

Combined (1) and (2), $|g_i − p_i| ≤ m/2$.

In Case 2:

$$g_i − p_i = t_i + m + 1 − p_i = (t_i − p_i) + m + 1 ≥ −m + m + 1 = 1.$$
In this subsection, the overhead of the proposed method is analyzed. When we hide a not-yet-hidden datum, next coming data are taken simultaneously if their values are all identical to that of the datum. If the repetition is less than 4, the data are hidden in the host image by using larger module \((m + 1)\) instead of \(m\). Using \(m + 1\) for the module, the average difference between the pixel of the host image and that of the stego image will be lightly increased. In detail, when \(m\) is even, in Thien and Lin’s method, omitting boundary condition, the maximum difference is \(lm/2!\) that is the same as in the proposed method; when \(m\) is odd, in Thien and Lin’s method, omitting boundary condition, the maximum difference is still \(lm/2!\), but the maximum difference is \((m + 1)/2!\) in the proposed method. However, if the repetition length is larger than 3 and no less than \(m + 4\) (if the repetition length is larger than \(m + 4\), we divide the data into several parts with each size \(m + 4\) or less), the number of pixels that should be modified is at most 3. Therefore, using the proposed method, the extra flag for indicating the repetition can reduce the possibility of pixel replacement. Besides the above overhead analysis, extra memory

![Fig. 1](image_url)
requirement is free and the whole processing power is still \(O(\text{size of secret data})\).

5.2. Improvement analysis

In this subsection, the improvement of the proposed method is analyzed. Suppose an image has \(R\) repetitions of \(L\) average length, each repetition length must be also known as \(l_i\), \(i = 1, 2, \ldots, R\). We have \(\sum_{i=1}^{R} l_i = L\). In the proposed method, when the repetition length \(l_i\) is at most 3, there is no improvement; when the repetition length \(l_i\) is between 4 and 19, the improvement from the proposed method is expressed by unchanging \(l_i - 3\) pixels; when the repetition length \(l_i\) is greater than 19, the repetition will be divided into more than one part and the improvement is counted part by part. Therefore, the expected improvement \(E(\text{Improv})\) is

\[
E(\text{Improv}) = \frac{1}{R} \sum_{i=1}^{R} \left( \left\lfloor \frac{l_i}{19} \right\rfloor \times 16 + \max((l_i \mod 19) - 3, 0) \right),
\]

where \(\sum_{i=1}^{R} l_i = RL\).

6. Conclusion

The traditional LSB-substitution methods can be viewed as methods that use the power of 2 as the base for modulus function. Module-base \(m + 1\) (\(m\) is not necessarily the power of 2) is adopted in this paper, in which the \(m\) digits \([0, 1, 2, \ldots, m-1]\) are used as ordinary numbers, and the special digit \(m\) is a flag for identifying the heavy repetition case. The quality of the stego-images generated by the proposed method is often better than that generated by Thien and Lin’s method [5], which in turn is better than that generated by [3,4]. On the other hand, when the data are with very slight repetition, Thien and Lin’s method [5] outperform the proposed method. (In Table 1, Ref. [5] leads in PSNR by a 0.48 dB difference when the data are formed of noise only; and a 0.05 dB difference when the secret data are formed of the image Baboon.) The proposed method can perform well in terms of hiding heavy-repetition data.

In decoding, the proposed method has the same very-low computation complexity \(O(\text{image size})\) as the reported LSB substitution methods [3,4] or module-based method [5]. In encoding, its computation complexity is the same as [5], and hence, much faster than the methods [3,4] which search for best renaming using genetic algorithm or dynamic programming.

The limitation of the proposed algorithm is that the proposed method only uses 4 bits to indicate the repetition length, so that, the repetition length can be only ranged from 4 to 19 and if consecutive data are repeated more than 19 times, the data need be recorded in additional three-digit vectors. However, if we use more than 4 bits for indicating the repetition length, then when the repetition length ranges from 4 to 19, more than 4 bits is still necessary for indicating the repetition length. The future research can focus on finding a balance way for indicating the repetition length to reach minimal distortion. Another future study can use dynamic programming to find the optimal bijection mapping and use the proposed method to

Fig. 2. The host images are of size 512 × 512.

Fig. 3. (a): The stego-image of hiding Fig. 1(a) in Fig. 2(a). (b): The stego-image of hiding Fig. 1(b) in Fig. 2(a). (c): The stego-image of hiding Fig. 1(c) in Fig. 2(a).
decrease the distortion of the generated stego-image. From this research, people can learn how the being hidden data are also possibly rearranged for minimizing the distortion of the stego-pixels.

**Acknowledgment**

The author is very grateful to anonymous reviewers for their constructive comments. This work was supported by the National Science Council, Republic of China, under grant NSC96-2218-E-264-002.

**References**


Shang-Kuan Chen received his B.S. degree in Applied Mathematics from Fu Jen Catholic University, Taiwan, in 1994, and M.S. degree in Applied Mathematics from NCTU, Taiwan, in 1998. In 2006, he received his Ph.D. degree in Computer Science from NCTU. He joined the Department of Computer Science and Information Engineering at Yuanpei University in 2006 and is currently an assistant professor there. His research interests include data hiding, visual cryptography and image sharing.

**Table 1**
The PSNR of stego-images (with m = 16) is generated using the simple LSB substitution, Chang et al.’s method [4], Thien and Lin’s method [5], and the proposed method. The proposed method outperforms the other three methods, unless the smooth area is too small in the secret image (for example, Baboon image, or another image formed only of random noise.).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>House</td>
<td>32.69</td>
<td>33.10</td>
<td>34.78</td>
<td>36.37</td>
<td></td>
</tr>
<tr>
<td>Milk</td>
<td>32.26</td>
<td>32.53</td>
<td>34.86</td>
<td>35.76</td>
<td></td>
</tr>
<tr>
<td>Jet</td>
<td>31.95</td>
<td>32.89</td>
<td>34.76</td>
<td>35.73</td>
<td></td>
</tr>
<tr>
<td>Tiffany</td>
<td>31.32</td>
<td>32.90</td>
<td>34.80</td>
<td>35.49</td>
<td></td>
</tr>
<tr>
<td>Pepper</td>
<td>32.44</td>
<td>32.57</td>
<td>34.79</td>
<td>35.46</td>
<td></td>
</tr>
<tr>
<td>Baboon*</td>
<td>32.65</td>
<td>32.71</td>
<td>34.78</td>
<td>34.73</td>
<td></td>
</tr>
<tr>
<td>Random-noise*</td>
<td>31.82</td>
<td>31.85</td>
<td>34.81</td>
<td>34.33</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>House</td>
<td>32.70</td>
<td>33.14</td>
<td>34.79</td>
<td>36.37</td>
<td></td>
</tr>
<tr>
<td>Milk</td>
<td>32.26</td>
<td>32.56</td>
<td>34.80</td>
<td>35.80</td>
<td></td>
</tr>
<tr>
<td>Jet</td>
<td>32.04</td>
<td>32.93</td>
<td>34.82</td>
<td>35.72</td>
<td></td>
</tr>
<tr>
<td>Tiffany</td>
<td>31.40</td>
<td>32.92</td>
<td>34.82</td>
<td>35.52</td>
<td></td>
</tr>
<tr>
<td>Pepper</td>
<td>32.46</td>
<td>32.58</td>
<td>34.80</td>
<td>35.44</td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>32.58</td>
<td>32.63</td>
<td>34.82</td>
<td>35.36</td>
<td></td>
</tr>
<tr>
<td>Random-noise*</td>
<td>31.83</td>
<td>31.89</td>
<td>34.80</td>
<td>34.34</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2**
The comparisons among the simple LSB substitution method, Chang et al.’s method [4], Thien and Lin’s method [5], and the proposed method using plain text documents (TXT), binary documents (.DOC), and archive files (.ZIP) as secret messages. The host image, Lena, is of size 257 K bytes and each secret message is 128 K bytes.

<table>
<thead>
<tr>
<th>PSNR</th>
<th>.TXT</th>
<th>.DOC</th>
<th>.ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple LSB substitution</td>
<td>31.96</td>
<td>30.23</td>
<td>31.77</td>
</tr>
<tr>
<td>Chang et al.’s method [4]</td>
<td>34.75</td>
<td>34.75</td>
<td>34.75</td>
</tr>
<tr>
<td>Thien and Lin’s method [5]</td>
<td>34.84</td>
<td>34.88</td>
<td>34.83</td>
</tr>
<tr>
<td>The proposed method</td>
<td>34.34</td>
<td>36.02</td>
<td>34.38</td>
</tr>
</tbody>
</table>