Improvements of image sharing with steganography and authentication

Ching-Nung Yang *, Tse-Shih Chen, Kun Hsuan Yu, Chung-Chun Wang

Department of Computer Science and Information Engineering, National Dong Hwa University, #1, Sec. 2, Da Hsueh Rd., Hualien, Taiwan

Received 22 October 2005; received in revised form 18 November 2006; accepted 26 November 2006

Abstract

Recently, Lin and Tsai proposed an image secret sharing scheme with steganography and authentication to prevent participants from the incidental or intentional provision of a false stego-image (an image containing the hidden secret image). However, dishonest participants can easily manipulate the stego-image for successful authentication but cannot recover the secret image, i.e., compromise the steganography. In this paper, we present a scheme to improve authentication ability that prevents dishonest participants from cheating. The proposed scheme also defines the arrangement of embedded bits to improve the quality of stego-image. Furthermore, by means of the Galois Field GF(2^8), we improve the scheme to a lossless version without additional pixels.

Keywords: Image secret sharing scheme; Steganography; Authentication

1. Introduction

Steganography is a method of embedding the secret message into a camouflage media to ensure that an unintended party will not be aware of the existence of the embedded secret in stego-images (still images containing the hidden secret image). As proposed in Thien and Lin (2002, 2003), Chang and Hwang (1998), Feng et al. (2005), Lin and Tsai (2004), Chang and Lin (2003), these image secret sharing schemes can produce both meaningful and meaningless stego-images based on the polynomial-based \((k, n)\) secret sharing scheme (Shamir, 1979), depending on the application constraints and user’s requirements. The so-called \((k, n)\) secret sharing scheme, where \(k \leq n\), the content of a secret message is divided into \(n\) shadows in the way that requires at least \(k\) shadows for the message reconstruction. Considering that the secret is an image, Shamir’s polynomial-based secret sharing scheme performs well. Additionally, other image secret sharing schemes like visual secret sharing (VSS) scheme (Naor and Shamir, 1995) show a completely different approach to share the secret pixel and its decoding method only needs human visual sight. However, the quality of recovered secret image for VSS scheme is extremely poor and not suitable for applications. In Lukac and Plataniotis (2005), the authors introduced simple calculations to VSS scheme for recovering the secret image that the quality is the same to polynomial-based image secret sharing scheme.

Polynomial-based image sharing schemes are briefly reviewed in the sequel. It is evident that we can hide the secret as the constant term in \((k – 1)\)-degree polynomials to construct a \((k, n)\) image secret sharing scheme. For further reducing the size of noise-like shadow images, Thien and Lin (2002) embedded the secret pixels in all the coefficients of the \((k – 1)\)-degree polynomials to encrypt the secret image into shadow images with size \(1/k\) times that of the original image. Due to the small shadow images, it is more suitable for fast transmission in the distributed storage environment. The same authors proposed in Thien and Lin (2003) to share user-friendly shadow images which are a shrunken version of the secret image and have insufficient quality to recover the input image for practical use in high-end applications by expanding the shadow image.
directly. However, using $k$ or more shadow images gets the high-quality reconstructed image. The shrunken version of the original image on shadows provides an easy-to-manage environment. Chang and Hwang (1998) used another approach that shares the vector quantization (VQ) codebook of the secret image instead of the secret image itself to improve the efficiency of the sharing process.

However, noise-like shadow images are suspected to censors and thus it would be beneficial to design an image secret sharing scheme with the steganography ability, i.e., shadow images look like the camouflage image and called as stego-images. Some schemes with the steganography ability were introduced in Feng et al. (2005), Chang and Lin (2003), Lin and Tsai (2004). Feng et al. (2005) proposed a general access structure solution for sharing multiple secret images based on the so-called sharing circle. Chang–Lin steganography scheme (Chang and Lin, 2003) provided the authentication ability for sharing color secret image. The dealer signs the serial number of camouflage image, the pixel value and the dealer's public key to generate signature. Twelve bits (every four LSB bits in red, green, and blue planes of a camouflage image) in a color pixel are reserved, nine bits for embedding the secret data and three signature bits for authenticating check. Although every four LSB bits are modified for each color plane, the quality of stego-image is also good enough for color camouflage images according to experimental results. However, changing four LSBs of a pixel in a gray camouflage image will seriously distort the quality of stego-images.

For gray stego-images, Lin and Tsai (2004) embedded the shared bits and authentication bit in a four-pixel block to slightly distort the quality of stego-image. Besides, Lin–Tsai scheme combined steganography and authentication features to prevent from incidentally bringing an erroneous stego-image or intentionally providing a false stego-image by using the parity check bit. The incidental provision was achieved but the intentional provision just addressed a direct and easy manipulation, i.e., bringing a false stego-image. Generally, verified secret sharing schemes (Qiong et al., 2005) always treat the participants as dishonest. There is no doubt that avoiding the participants from cheating is more practical in the real environment. Unfortunately, the property of the parity checking leaks the information of verification and the dishonest participants can easily manipulate fake stego-images.

Following the above discussion, this paper shows that Lin–Tsai scheme cannot avoid the dishonest participants from cheating. Also, we introduce an improved image secret sharing framework. The proposed framework uses hash function instead of parity checking and rearranges the positions of shared bits and authentication bit to improve the authentication ability and the quality of stego-image obtained using Lin–Tsai scheme. Additionally, all the previous polynomial-based image secret sharing schemes are not lossless version (i.e. no distortion in the secret image) due to using mod 251 in the calculation of polynomial. The possible values of an 8-bit gray pixel are from 0 to 255, so the gray scale values greater than 250 need to be modified to 250. This modification will slightly distort the secret image and is discussed and solved in Thien and Lin (2002) using two pixels to represent 250–255. By choosing the proper Galois Field in the calculation of polynomial, we improve the proposed scheme to a lossless version without additional pixels. The remainder of this paper is organized as follows. In Section 2 we briefly the weaknesses of Lin–Tsai scheme. In Section 3 we design the proposed scheme. Experimental results are given in Section 4, and we draw our conclusions in Section 5.

2. Lin–Tsai image secret sharing scheme and its weaknesses

To embed one secret pixel into a four-pixel square block in the camouflage images for Lin–Tsai image secret sharing scheme (Lin and Tsai, 2004), the grayscale value of pixel is hidden as the constant term of a $(k - 1)$-degree polynomial. The input of polynomial is chosen from the upper left pixel in this square block. The nine bits (8-bit output of polynomial and one parity check bit) are put into other three pixels of the block, i.e., every last three last bits in a pixel, to form a stego-image.

Since Lin–Tsai scheme provides the parity bit for authenticating the fidelity of stego-images and offers the ability to prevent the participant incidentally or intentionally from providing fake stego-images. However, it is evident that dishonest participants can manipulate fake stego-images to compromise the secret recovery and pass the authentication because they know all the parity information from their own stego-images. The problem of dishonest participant is described in Weakness 1. Other two weaknesses, Weakness 2 and Weakness 3, affect the qualities of stego-image and the recovered secret image, respectively.

2.1. Weakness 1 (dishonest participants)

For Lin–Tsai scheme, the parity bit in the upper right pixel is chosen to make this pixel even or odd parity as a binary parity sequence generated by a secret key. Because a participant can find out the parity information from his own stego-image, he can maliciously make a false stego-image. For example, a dishonest participant can modify an upper right pixel $(110111011)$ to $(110111110)$ with the same odd parity but change the 8-bit input of $(k - 1)$-degree polynomial. So, the parity authentication is successful but the $(k - 1)$-degree polynomial cannot be obtained. Also, the participant can modify other three pixels to change the input and output of $(k - 1)$-degree polynomial and does not influence the upper right pixel for passing authentication but compromising steganography.

2.2. Weakness 2 (positions of shared bits and check bit in a four-pixel block)

There are total 17 bits (8-bit input and 8-bit output of $(k - 1)$-degree polynomial, and 1 parity bit) in the four-
pixel block used for embedding and recovering process. In Lin–Tsai scheme the upper left pixel is used as the 8-bit input and other nine bits are put into the last three LSBs in other three pixels. As we know, the modification of least significant bits will result in less distortion rather than modification of most significant bits. By arranging these altered bits to approach the LSBs, we may deteriorate the quality of stego-images as small as possible. It is observed that the arrangement of 17 bits is not an optimal solution. How to arrange the positions of these 17 bits for improving the quality of stego-image is discussed in this paper.

2.3. Weakness 3 (non-lossless secret image)

The prime number 251 was recommended in the calculation of \((k - 1)\)-degree polynomial to process the most range \((0–250)\) in \((0–255)\) for an 8-bit pixel. Although this results in the slight distortion of grayscale value of secret pixel is constrained between 0 and 250. Although this results in the slight distortion of recovered image, we cannot actually get a lossless version and 250. Although this results in the slight distortion of grayscale value of secret pixel is constrained between 0 and 255. However, the stego-image of proposed scheme.

Fig. 1. One four-pixel square block. (a) The camouflage image. (b) The stego-image of proposed scheme.

\[ q(x) = \left( a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1} \right) \mod(2^8) \]

\[ i^{(j)} = j \text{th stego-image of } 2m \times 2m \text{ pixels. There are } m^2 \text{ four-pixel blocks } B^{(i)} \text{ containing four pixels } X_i, W_i, V_i, \text{ and } \bar{U}_i \text{ with grayscale values } x_i, w_i, v_i, \text{ and } \bar{u}_i \text{ shown in Fig. 1b, respectively, where } (q_{11}, q_{12}, \ldots, q_{16}) \text{ is the binary representation of } q_i \text{ and } p_i \text{ is the authenticated bit based on hash function described in Section 3.1.}

3.1. Authentication ability

To prevent participants from manipulating stego-images, we use a hash function with the secret key \(K\), \(H_k(\cdot)\), such as a NIST standard FIPS198 HMAC (Stallings, 2003). First, calculate the output, 160 bits, of HMAC (e.g., use SHA-1 as the embedded hash function in HMAC) in (1)

\[ H_k \left( \left( B^{(i)} - p_i \right) || i \right) \]

where \(B^{(i)} - p_i\) are 31 bits exclusive the check bit \(p_i\) from the block \(B^{(i)}\) and the index of block \(i \in [1,m^2]\) used as the block identification (block ID). Then, XOR these 160 bits to get the hash bit \(p_i\) denoted as

\[ p_i = \text{XOR} \left( H_k \left( \left( B^{(i)} - p_i \right) || i \right) \right) \]

for this block. Replace the parity bit with the hash bit \(p_i\). Therefore, any modification of block \(B^{(i)}\) can be detected by the hash bit with probability 1/2. The block ID used in HMAC is to prevent the attacker from using other blocks in the stego-image to counterfeit like the so-called vector quantization (VQ) attack. Considering that one could use blocks of same ID in other stego-images to manipulate stego-image, the stego-image identification, \(I^{(1)}_{ID}\), is added into HMAC to resist the attack. Then, (1) is modified as

\[ H_k \left( \left( B^{(i)} - p_i \right) || i || I^{(1)}_{ID} \right) \]

The random binary sequences \((p_1, p_2, \ldots, p_m)\) are lurked in the entire stego-image. Without the secret key \(K\), a participant “\(j\)” cannot modify \(X_i, \bar{W}_i, \bar{V}_i\), and \(\bar{U}_i\) even though he knows \(p_i\). He can guess each hash bit \(p_i\) with 1/2 probability and only has \(1/2^m\) probability to successfully make a fake stego-image that passes the authentication. The successful authentication probability \((1/2^m)^2\) will significantly reduce when the size of stego-image increases, e.g., the successful authentication probability is only \(3 \times 10^{-5} \approx (1/2^2)^5\) for a small stego-image of size 10 x 10 pixels. So we can provide the real authentication ability that prevents dishonest participants from malicious modifications.

3.2. Improvement of the stego-image quality

In this section, we show how to arrange 17 bits (16 bits \((x_i, q_i)\) for the input/output of \((k - 1)\)-degree polynomial and one hash bit \(p_i\) to further improve the quality of
Table 1

<table>
<thead>
<tr>
<th>$L_X - L_1 - L_2 - L_U$</th>
<th>$L_C$</th>
<th>Estimated PSNR</th>
<th>$L_X - L_1 - L_2 - L_U$</th>
<th>$L_C$</th>
<th>Estimated PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0–0–8–1</td>
<td>1</td>
<td>10.783</td>
<td>0–2–4–3</td>
<td>1</td>
<td>34.195</td>
</tr>
<tr>
<td>0–0–7–2</td>
<td>1</td>
<td>16.826</td>
<td>0–3–3–3</td>
<td>1</td>
<td>36.909</td>
</tr>
<tr>
<td>0–0–6–3</td>
<td>1</td>
<td>22.843</td>
<td>1–1–1–6</td>
<td>2</td>
<td>22.893</td>
</tr>
<tr>
<td>0–0–5–4</td>
<td>1</td>
<td>28.092</td>
<td>1–2–5</td>
<td>2</td>
<td>28.953</td>
</tr>
<tr>
<td>0–1–7–1</td>
<td>1</td>
<td>16.828</td>
<td>1–1–3–4</td>
<td>2</td>
<td>34.264</td>
</tr>
<tr>
<td>0–1–6–2</td>
<td>1</td>
<td>22.877</td>
<td>1–2–4–2</td>
<td>2</td>
<td>34.806</td>
</tr>
<tr>
<td>0–1–5–3</td>
<td>1</td>
<td>29.013</td>
<td>1–2–3–3</td>
<td>2</td>
<td>38.131</td>
</tr>
<tr>
<td>0–1–4–4</td>
<td>1</td>
<td>32.220</td>
<td>2–3–2–2</td>
<td>4</td>
<td>39.380</td>
</tr>
<tr>
<td>0–2–5–2</td>
<td>1</td>
<td>28.927</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>Fig. 2. The four-pixel square blocks for different ($L_X - L_1 - L_2 - L_U$).</th>
</tr>
</thead>
<tbody>
<tr>
<td>($L_X-4, L_C-2$)</td>
<td>(0–1–0–8);</td>
<td>Lin–Tsai scheme</td>
<td>(2–3–2–2);</td>
<td>the worst arrangement</td>
</tr>
<tr>
<td>($L_X-C, L_1-3$)</td>
<td>(0–3–3–3);</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>($L_X-U, L_C-2$)</td>
<td>(2–3–2–2);</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.3. A lossless image secret sharing scheme

The possible grayscale values of an 8-bit pixel are from 0 to 255. To hide the secrets in the $a_0$ coefficient of $(a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1})$ mod 251, the pixel values greater than 250 are modified to 250. This modification will slightly distort the secret image and is solved in Thien–Lin scheme using two pixels to represent 250–255 (Thien and Lin, 2002). For example, the pixel value 255 is represented as two pixels: 250 and 5. When decoding, if the recovered secret pixel is 250, we need to check the next pixel to obtain the real value. For constructing a lossless secret image, we use $g(x) = (a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1})$ mod $g(x)$, where $g(x) = (x^8 + x^4 + x^3 + x + 1)$, i.e., using the Galois Field $GF(2^8)$ instead of $GF(251)$, such that we can process all the grayscale values for an 8-bit pixel. Finally our scheme is improved to a lossless version without additional pixels.

3.4. Embedding algorithm for the proposed scheme

Using the (2–3–2–2) arrangement of block (Fig. 2d) and the hash-based authentication with the block ID and the stego-image ID, the formal encoding procedure for our improved scheme is shown in the following algorithm.

**Embedding algorithm.**

**Input.** A secret image $S$, $n$ camouflage images $I^0$, with identification, $j \in [1,n]$, and a secret key $K$.

**Output.** $n$ stego-images $I^j$, $j \in [1,n]$.

For $i = 1$ to $m^2$ do

{Calculate $p_i = \text{XOR}(H_K(\tilde{B}^i - p_i))$;}

Choose a $(k - 1)$-degree polynomial $q(x)$ with $a_0 = x_i$;

/* hide the secret pixel $s_i$ as the constant term in $q(x)$ */

$q(x) = (a_0 + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1})$ mod $g(x)$,
where $g(x) = (x^8 + x^4 + x^3 + x + 1)$;

/* Calculate $q(x_i) = q_i$ and find its binary representation */

/* $(q_1, q_2, \ldots, q_8)$ */

/* The same block in other stego-images occurs, the bit $x_{i6}$ should be modified */

/* $X_i = (x_{i1}, x_{i2}, \ldots, x_{i8})$ where 8 bits are located in $X_i$ and $V_i$ shown in Fig. 1b, if the duplicate of $x_i$ should be modified */

/* $X_i = (x_{i1}, x_{i2}, \ldots, x_{i8})$; */

/* replace the last two LSBs in $X_i$ with $q_{i1}$ and $q_{i2}$ to form $X_i$ */

/* $W_i = (w_{i1}, \ldots, w_{i5}, p_i, q_{i2}, q_{i4})$ */

/* replace the last three LSBs in $W_i$ with $p_i, q_{i3}, q_{i4}$ to form */

/* $V_i = (v_{i1}, \ldots, v_{i4}, x_{i7}, x_{i8}, q_{i5}, q_{i6})$ */

/* replace the last two LSBs in $V_i$ with $q_{i5}$ and $q_{i6}$ to form */

/* $U_i = (u_{i1}, u_{i2}, \ldots, u_{i6}, q_{i5}, q_{i6})$ */

/* replace the last two LSBs in $U_i$ with $q_{i7}$ and $q_{i8}$ to form */

Put $\tilde{B}^i$ in $I^i$;}

4. Experimental results

4.1. Improvements of the qualities for the stego-images and secret image

In this section, we take a (2,3)-threshold scheme to test Lin–Tsai scheme and the proposed scheme. Test images are
in stego-images. Fig. 4a shows the cropped area in the original Lena image which has the smooth grayscale level near the shoulder. However, for the cropped areas, Fig. 4b and c, obtained from Fig. 3b and c, the smooth is distorted and our scheme (Fig. 4c) has the better result than Lin–Tsai scheme (Fig. 4b).

Suppose the secret image is general test pattern of 256 × 256 pixels (see Fig. 5a) chosen from the USC-SIPI image database. White areas in Fig. 5b show the pixels of grayscale values more than 249. These 9084 pixels should be replaced by two pixels to obtain a lossless version. To keep the aspect ration of secret image invariant, we add additional 456 pixels and finally the modified secret image is 274 × 274(=65,536 + 9084 + 456) pixels. Finally, the size is 1.15 times that of the original secret image. Our scheme is a lossless version without the extension of secret image.

### 4.2. Evaluation of the authentication ability

Parity checking bits in Lin–Tsai scheme leaks the information of verification and dishonest participants can easily counterfeit a fake stego-image that pass the parity checking authentication. According to the modifications of stego-images, we categorize the manipulations into three types. The first type is the unobvious modification of stego-image. Dishonest participants modify one of four LSB bits: $x_{i8}$, $q_{i2}$, $q_{i5}$, $q_{i8}$ for Lin–Tsai scheme such that the recovery of polynomial fails and the distortion of stego-image is slight. Nevertheless, the dishonest participant cannot modify the LSBs $q_{i2}$, $q_{i5}$, $q_{i6}$ and $q_{i8}$ in $\tilde{X}^i$, $\tilde{W}^i$, $\tilde{V}^i$ and $\tilde{U}^i$ for our scheme (see Fig. 1b) to pass the check of hash bit because he does not know the secret key $K$ and HMAC is a one-way function. Fig. 6a shows a modified region, an upper left corner of Pepper stego-image, shown as red dotted line. The fake stego-image is similar to the original stego-image and one cannot observe the modification. Fig. 6b and c show that the parity authentication fails but the hash value authentication performs well, respectively. The modified region is localized when using our scheme (see Fig. 6c).

Same to the above manipulation, but the dishonest participant adds an extra image satisfying the parity check into the stego-image. For example, an “Apple” is intentionally added on the upper left corner in the Pepper stego-image (Fig. 7a). For this manipulation type, the dishonest participant choose the grayscale values $\tilde{x}_{i8}$, $\tilde{v}_{i8}$, $\tilde{w}_{i8}$ and $\tilde{u}_{i8}$ according to the image “Apple”. but the bit $q_{i2}$ is modified to make the parity of $\tilde{W}^i$ unchanged for compromising Lin–Tsai scheme’s steganography. Fig. 7b and c are the authentication checking results for Lin–Tsai scheme and the proposed scheme, respectively. The localization region in Fig. 7c shows the shape of “Apple”. Fig. 7d–f show another example that a duplicated boat in the modified Lake stego-image.

The third manipulation type is to replace a stego-image with another image satisfying the parity authentication. A dishonest participant having all the parity information of the original Pepper stego-image may manipulate another
stego-image Jet by modifying the bit $q_{2}$ to make the parity of $\hat{W}$ unchanged. In Fig. 8, our authenticated result shows
that the modified regions spread out in the whole fake stego-image.

5. Conclusions

In this paper, we present an improved image secret sharing scheme with steganography and authentication. The main objective is to prevent the participants from cheating. More in detail, the paper makes the following contributions. First, it shows that the authentication can be greatly improved by hashing the four-pixel block, block ID and image ID. At this time, even the dishonest participant could not manipulate the fake stego-image from his own stego-image. Second, we take into account the improvement of both qualities for stego-images and secret image. By defining a new arrangement of 17 bits (16 bits for the input and output of \((k-1)\)-degree polynomial, and 1 hashed bit) in the for-pixel square block, we improve the quality of stego-images. In order to improve the scheme to a lossless version with no additional pixels, we use the Galois Field \(GF(2^8)\) in \(q(x)\).

Acknowledgment

The authors thank the anonymous reviewers for their helpful comments.

References


