An Advanced Least-Significant-Bit Embedding Scheme for Steganographic Encoding

Yeuan-Kuen Lee[1], Graeme Bell[2], Shih-Yu Huang[1], Ran-Zan Wang[3], and Shyong-Jian Shyu[1]

[1] Computer Science and Information Engineering, Ming Chuan University, Taiwan
[2] International College, Ming Chuan University, Taiwan
[3] Computer Science and Engineering, Yuan Ze University, Taiwan
{yklee,gbb,syhuang,sjshyu}@mail.mcu.edu.tw, rzwang@saturn.yzu.edu.tw

Abstract. The advantages of Least-Significant-Bit (LSB) steganographic data embedding are that it is simple to understand, easy to implement, and it results in stego-images that contain hidden data yet appear to be of high visual fidelity. However, it can be shown that under certain conditions, LSB embedding is not secure at all. The fatal drawback of LSB embedding is the existence of detectable artifacts in the form of pairs of values (PoVs). The goals of this paper are to present a theoretic analysis of PoVs and to propose an advanced LSB embedding scheme that possesses the advantages of LSB embedding suggested above, but which also provides an additional level of communication security. The proposed scheme breaks the regular pattern of PoVs in the histogram domain, increasing the difficulty of steganalysis and thereby raising the level of security. The experimental results show that both the Chi-square index and RS index are less than 0.1, i.e., the hidden message is undetectable by the well-known Chi-square and RS steganalysis attacks.

Keywords: Steganography, steganalysis, LSB embedding.

1 Introduction

Both steganography and cryptography may be used to protect secret messages in order to achieve private communication. Steganography not only hides the meaning but also the existence of the hidden message. Ideally, only the intended receiver can extract the message, as other people viewing the carrier medium are unaware of the existence of the hidden message. Steganographic techniques can therefore protect not only the secret message but also the sender and the receiver. In the field, cryptographic techniques are typically sufficient to protect secret data. However, users such as informers may need steganographic techniques to protect themselves and their whole organization [1].

In recent years, many discreet methods for hiding encrypted messages within digital 'carrier' media have become conveniently available. One such approach is the LSB embedding approach, which simply replaces the least significant bit of

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 349–360, 2009.

each carrier data value with the message-bit. This approach is simple to understand and easy to implement, and the resulting 'stego-media' containing hidden messages appear to be of high visual fidelity. Consequently, the LSB embedding approach has become the basis of many techniques that hide messages within multimedia carrier data. LSB embedding may even be applied in particular data domains - for example, embedding a hidden message into the color values of RGB bitmap data, or into the frequency coefficients of a JPEG image. LSB embedding can also be applied to a variety of data formats and types [2]. Therefore, LSB embedding is one of the most important steganographic techniques in use today.

Since LSB embedding is one of the simplest effective data hiding techniques, it has long been a focus for researchers proposing steganalytic attack methods. The Chi-square attack was the first statistical test that could detect hidden messages automatically [3]. Two values whose binary representations differ only in the LSB are called a pair of values (PoV). For example, $68(01000100)_2$ and $69(01000101)_2$ are a PoV. If the numbers of 1s and 0s are equal and distributed randomly in the secret message that is to be embedded steganographically, the frequency of two values in each PoV will be equal after message embedding. This regular equality pattern, called the PoVs artifact, is an unusual characteristic in the histogram domain. If the PoVs artifact can be found in a digital media, there is a high probability that a hidden message is embedded in the media. The Chi-square attack is a very effective technique against LSB embedding systems. A known counter-technique to avoid exposing hidden messages to this attack involves decreasing the embedding capacity of the carrier medium. If less than 50% of the maximum capacity of the carrier medium is used, the risk of detection drops accordingly.

For detecting messages embedded in 24-bit color images, Fridrich et al. proposed the RPQ (Raw Quick Pairs) steganalysis system in 2000 [4]. However, the technique was shown to be unreliable for digital camera images that are stored in an uncompressed format, where a large number of unique color values may exist. In 2001, Fridrich et al. proposed a more reliable attack on LSB embedding called RS steganalysis [5]. Fridrich et al. estimate that messages hidden within high quality images using an embedding rate of more than 0.005 bits per pixel are detectable by RS steganalysis.

F5 is a steganographic algorithm proposed in 2001 for JPEG images [6]. In the F5 algorithm, statistical properties in the histogram of quantized DCT coefficients are preserved and a matrix encoding [7] is implemented. Matrix encoding decreases the number of changes needed, in order to improve the embedding efficiency. In IHW 2002, Fridrich et al. proposed a steganalytic method for breaking the F5 algorithm [8]. The key element of this attack comes from the estimation of the cover-image histogram from the stego-image. Experimental results have shown that modifications of as few as 10% of the usable capacity of the DCT coefficients, can be reliably detected.

Recently, many steganographic methods based upon LSB embedding have attracted statistical attacks, and experimental results have shown that this

approach is generally not secure at all. T. Sharp proposed an implementation, called '*Hide*', of key-based image steganography in 2001 [9]. *Hide* uses a modified LSB method for embedding messages. The LSBs are not simply replaced; instead the data value is incremented or decremented if the LSB differs from the message-bit. *Hide* uses a pseudorandom sequence generator to determine whether to increment or decrement the data value.

In this paper, we first present a theoretic analysis of the LSB embedding approach and then propose an advanced LSB embedding scheme. The sample value that will be incremented or decremented depends on a series of predefined thresholds that are generated by the user-specified stego-key. The new sample value not only depends on the generated pseudorandom number but also depends on the original sample value. Experimental results show that both of the well-known Chi-square and RS steganalysis attacks are unable to detect the existence of secret messages embedded with the new system. Using the proposed scheme is therefore more secure than using traditional LSB embedding techniques.

The rest of this paper is organized as follows. In Section 2, a theoretic analysis of the weakness of the LSB embedding approach is presented. An advanced LSB embedding scheme is proposed in Section 3. The experimental results are discussed in Section 4. Finally, the paper is concluded in Section 5.

2 Analysis of LSB Embedding

LSB embedding involves replacing the least significant bit of the original data value with the secret message-bit directly. For a grayscale image, the intensity values range from 0 to 255. These can be grouped into 128 PoVs, i.e., $(2k, 2k+1), k = 0, 1, \ldots, 127$. Applying the LSB embedding operation cannot change a value so that it corresponds to another, different PoV. Thus, the operation of LSB embedding on a PoV satisfies the closure property, i.e., no matter whether the embedded message-bit is 1 or 0, the result will continue to belong to the same PoV.

Let I denote an original grayscale cover-image and I' denote the created stego-image in which the secret message is embedded. Let H_I denote the histogram of a grayscale image I. Let $H_I(i)$ denote the frequency of gray value i, and let $HP_I(k)$ denote the frequency of values in the k-th PoV in I. Then,

$$HP_I(k) = H_I(2k) + H_I(2k+1), \tag{1}$$

$$HP_{I'}(k) = H_{I'}(2k) + H_{I'}(2k+1). \tag{2}$$

The closure property ensures that summing the histogram values for each value in the PoV, produces a total that will be unchanged by LSB modification. Thus,

$$HP_I(k) = HP_{I'}(k) \tag{3}$$

Let T denote the embedding rate, that is,

$$T = t/N, \tag{4}$$

$0 \leq T \leq 1$, where t is the length of secret message and N is the total number of pixels in the image I. A total of t pixels are selected randomly for embedding t bits of secret message. In general, the secret message that is being embedded is always compressed and encrypted before embedding. The number of '1' and '0' in the hidden message can therefore reasonably be assumed to be equal. Thus, among the selected pixels, half of the pixels with even values ($2k$) will not change when the embedded message-bit is 0, and half of the pixels with odd values ($2k+1$) will change into $2k$ when the embedded message-bit is 0. Similarly, among the unselected pixels, the number of pixels with value $2k$ is $H_I(2k)(1-T)$. Thus, the number of pixels with value $2k$ can be derived as follows.

$$\begin{aligned} H_{I'}(2k) &= H_I(2k)(1-T) + H_I(2k)(T/2) + H_I(2k+1)(T/2) \\ &= H_I(2k)(1-T) + [(H_I(2k) + H_I(2k+1)](T/2) \\ &= H_I(2k)(1-T) + HP_I(k)(T/2), \end{aligned} \tag{5}$$

Similarly, the frequency of the other value in the same PoV can be derived as follows.

$$\begin{aligned} H_{I'}(2k+1) &= H_I(2k+1)(1-T) + H_I(2k+1)(T/2) + H_I(2k)(T/2) \\ &= H_I(2k+1)(1-T) + [(H_I(2k+1) + H_I(2k)](T/2) \\ &= H_I(2k+1)(1-T) + HP_I(k)(T/2), \end{aligned} \tag{6}$$

Let $DP_I(k)$ denote the difference between the frequencies of values in the k-th PoV in the cover-image I. Thus,

$$DP_I(k) = |H_I(2k) - H_I(2k+1)|, \tag{7}$$

From Eqs. (5) and (6), when the embedding rate is T, the difference between the frequencies of values in the k-th PoV in the stego-image I' can be derived as follows.

$$\begin{aligned} DP_{I'}(k) &= |H_{I'}(2k) - H_{I'}(2k+1)| \\ &= |H_I(2k)(1-T) - H_I(2k+1)(1-T)| \\ &= |[H_I(2k) - H_I(2k+1)]|(1-T) \\ &= DP_I(k)(1-T). \end{aligned} \tag{8}$$

So, the difference between the frequencies of values in the same PoV will become $(1-T)$ times after LSB embedding. When $T = 0$, there is no secret message embedded in the image. Thus,

$$H_{I'}(2k) = H_I(2k), \tag{9}$$

$$H_{I'}(2k+1) = H_I(2k+1), \tag{10}$$

$$\begin{aligned} DP_{I'}(k) &= |H_{I'}(2k) - H_{I'}(2k+1)| \\ &= |H_I(2k) - H_I(2k+1)| \\ &= DP_I(k). \end{aligned} \tag{11}$$

So, the closure property of LSB embedding is obvious. When $T = 1$, all pixels will used to embed the secret message. From Eqs. (5), (6) and (8),

$$H_{I'}(2k) = HP_I(k)/2, \tag{12}$$

$$H_{I'}(2k+1) = HP_I(k)/2, \tag{13}$$

$$DP_{I'}(k) = 0. \tag{14}$$

From Eqs. (12) and (13), when $T = 1$, then

$$H_{I'}(2k) = H_{I'}(2k+1) = HP_I(k)/2. \tag{15}$$

The regular equality pattern of PoVs in the histogram domain has been proven.

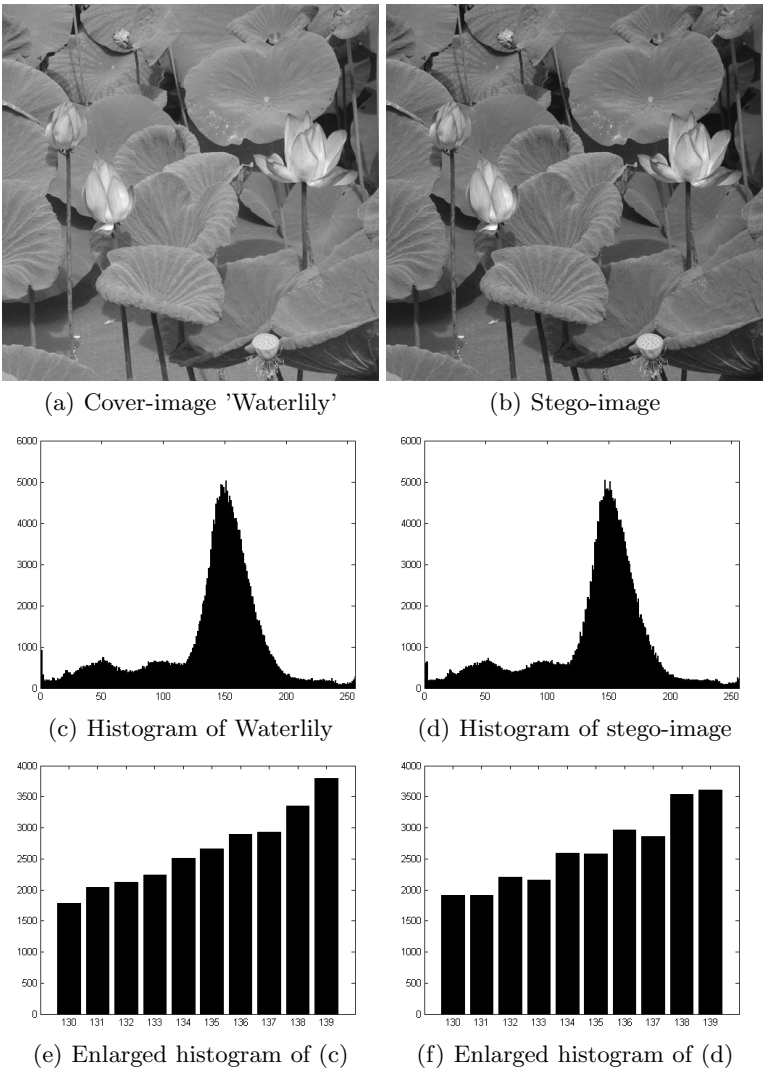
Fig. 1 is an example of the PoVs artifact caused by LSB embedding in the histogram domain. Figs. 1(a) and 1(b) show the original cover-image entitled Waterlily and its corresponding stego-image with full capacity of binary random data, respectively. The peak signal-to-noise ratio (PSNR) of Fig. 1(b) is 51.1409*db*. Figs. 1(c) and 1(d) are the histograms of Figs. 1(a) and 1(b), respectively. Figs. 1(e) and 1(f) show the enlarged histograms between values from 130 to 139 of Figs. 1(c) and 1(d). Note that the PoVs artifact appears in Fig. 1(f), and that the differences decrease within the 5 PoVs from $(130, 131)$ to $(138, 139)$.

3 Proposed Scheme and Discussion

The PoVs artifact exposes the existence of a hidden message. One obvious way to decrease the risk of message exposure resulting from the PoVs artifact is to decrease the embedding capacity. This paper proposes a second way to raise the security level, in which the embedding capacity is not reduced, while the fidelity of the stego-image is maintained.

The idea behind the proposed scheme is direct. PoVs will be disturbed in the embedding process. For any integer q, there are two neighbors with the same LSB, i.e., $q-1, q+1$. The PoVs artifact is caused by having a fixed choice of neighbor value to replace the original value, that is, a 'pair value'. Yet, two possible neighbors of equal difference to the integer q exist. Further, no matter which neighbor is selected, the fidelity of the resulting stego-image will be as good as that created by a traditional LSB embedding approach.

Two data embedding models, a basic model and an advanced model that are both based upon this idea of alternative neighbours, are included in the proposed scheme. Basically, the basic model is similar to the method used in [9]. Note that the message extraction process for both of these new models is almost identical to the traditional LSB embedding method. The embedding process will now be described in detail.



(a) Cover-image 'Waterlily' (b) Stego-image

(c) Histogram of Waterlily (d) Histogram of stego-image

(e) Enlarged histogram of (c) (f) Enlarged histogram of (d)

Fig. 1. PoVs artifact exists in the histogram after applying LSB embedding

3.1 Basic Model

A pseudorandom number generator (PRNG) seeded with a value known to both sender and receiver, is used to randomly select one of two neighbors, i.e., $q-1$ or $q+1$, where q is the original value. Let M denote a binary secret message sequence, $M = \{m_i | m_i \in \{0,1\}, i = 0,1,\ldots,t-1\}$, where t is the message length. Let $f_I(x,y)$ denote the grayscale value at (x,y) in cover-image I, and let $LSB_I(x,y)$ denote the LSB of the grayscale value at (x,y). The embedding algorithm is as follows.

Embedding algorithm of basic model: E_B
Input: cover-image I, binary message sequence M.
Output: stego-image I'.

Step 1:
Set $I' = I$.
Step 2:
Use a PRNG to randomly select t pixels from I'.
Let (x_i, y_i) denote the coordinate of the selected pixel. $i = 0, 1, \ldots, t-1$.
Step 3:
Let $q_i = f_I(x_i, y_i)$ denote the grayscale value of pixel (x_i, y_i).
Let m_i denote the message-bit to be embedded in pixel (x_i, y_i).
For all pixels (x_i, y_i),
if $LSB_{I'}(x_i, y_i) = m_i$,
do nothing;
if $LSB_{I'}(x_i, y_i) \neq m_i$,
use a PRNG to generate a random number γ, $0 \leq \gamma \leq 1$,
if $\gamma > 0.5$,
set $f_{I'}(x_i, y_i) = q_i + 1$;
if $\gamma \leq 0.5$,
set $f_{I'}(x_i, y_i) = q_i - 1$;
Step 4:
Output I'

So, in short, we choose pseudorandomly (but in a manner predictable to both sender and receiver) one of the two possible neighboring values, whenever it is necessary to perturb the pixel value to encode a message bit. The receiver can reconstruct the message by revisiting the pseudorandomly selected pixels and extracting the LSBs of pixel values directly.

Fig. 2 illustrates how the LSB embedding and the basic model perform message embedding under the condition where the value is an even number $2k$. Thus, $LSB_{I'}(x_i, y_i) = 0$. The probability that message bit $m_i = 1$ is $1/2$, hence half of the pixel values will change to $2k+1$ by using the LSB embedding. However, in the embedding process of the basic model only a quarter of the pixel values will change to $2k+1$, and the other quarter of the pixel values will change to $2k-1$. Therefore, there is no fixed pair of values in the histogram of stego-image I'. The frequency of pixel value q can be derived as:

$$H_{I'}(q) = H_I(q-1)(T/4) + H_I(q)(1 - T/2) + H_I(q+1)(T/4). \qquad (16)$$

Since the frequency of pixel value q in the stego-image I' is contributed towards from the frequencies of $q-1, q$ and $q+1$ in the cover-image I, there is no PoVs artifact.

Fig. 3 gives a sample of experimental results examining the behavior of the basic model. Both the cover-image and the secret message are the same as those used in Fig. 1. Figs. 3(a) and 3(b) show the created stego-image and its corresponding histogram, respectively. The PSNR of Fig. 3(b) is $51.1409db$, which is

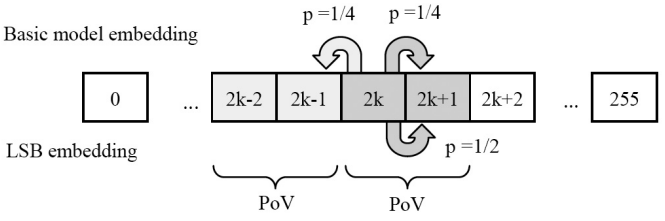


Fig. 2. Comparison between the basic model and LSB embedding

the same as Fig. 1(b). Fig. 3(c) shows a part of the enlarged histogram from values 130 to 139. In Fig. 3(c), we can observe that the PoVs artifact has been removed. Applying the Chi-square attack on Fig. 3(a), the Chi-square index is very close to 0, $p = 3.4417e - 15$. This means the message hidden in Fig. 3(a) is essentially undetectable by the Chi-square attack.

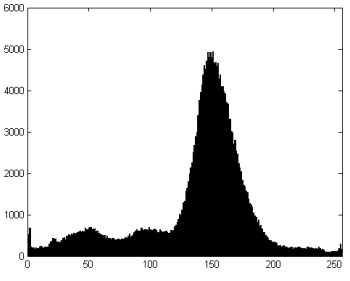
3.2 Advanced Model

The goal of the advanced model is to raise the security level even further. Although the PoVs artifact does not appear in the stego-images created by the basic model, only one unknown variable exists in the right side of Eq. (16). Observing Eq. (16), it may be noticed that the source of the frequency of q in I' is the halved frequencies of selected pixels with values $q-1$ and $q+1$ in I. When the value of $LSB_{I'}(xi, yi) \neq mi$, the threshold used to decide the chance of selecting $q-1$ and $q+1$ is 1/2, that is, a 50% chance. In the advanced model, these thresholds are allowed to vary within a predefined set. Let $\beta(q)$ denote the predefined threshold used on the pixels with value q. $\beta(q)$ can be generated by the same PRNG used in the embedding process. Now, the dependence of histogram value q upon values $q-1$ and $q+1$ is unpredictable. The embedding algorithm of the advanced model is described as follows.

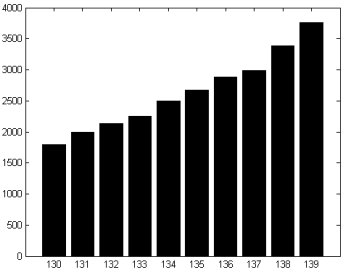
Embedding algorithm of advanced model: E_A
Input: cover-image I, binary message sequence M
Output: stego-image I'

Step 1:
$I' = I$
Step 2:
Use a PRNG to generate 256 random numbers, β, $0 \leq \beta \leq 1$.
Let $\beta(q)$ denote these random numbers, $q = 0, 1, \ldots, 255$.
Step 3:
Use a PRNG to randomly select t pixels from I'.
Let (x_i, y_i) denote the coordinate of selected pixel. $i = 0, 1, \ldots, t-1$.

(a) Stego-image



(b) Histogram



(c) Enlarged histogram

Fig. 3. A sample of experimental results based upon the basic model

Step 4:
Let $q_i = f_I(x_i, y_i)$ denote the gray value of pixel (x_i, y_i) .
Let m_i denote the message-bit embedded in pixel (x_i, y_i).
For all pixels (x_i, y_i),
if $LSB_{I'}(xi, yi) = m_i$,
do nothing;
if $LSB_{I'}(x_i, y_i) \neq m_i$,
use a PRNG to generate a random number γ, $0 \leq \gamma \leq 1$,
if $\gamma > \beta(q)$,
then $f_{I'}(x_i, y_i) = q_i + 1$;
if $\gamma \leq \beta(q)$,
then $f_{I'}(x_i, y_i) = q_i - 1$;

Step 5:
Output I'

Fig. 4 illustrates the difference between the basic model and advanced one. When $LSB_{I'}(x_i, y_i) \neq m_i$, the probability that the value q will change to $q-1$ or $q+1$ is not equal any more, and depends indirectly upon the value q. The frequency of pixel value q, $H_{I'}(q)$, can be expressed as follows.

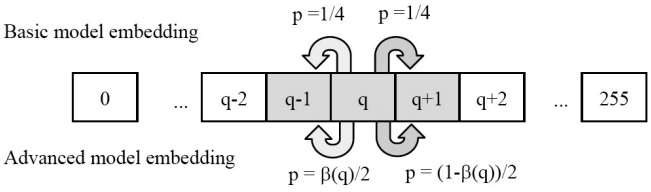


Fig. 4. Comparison between two proposed models

$$H_{I'}(q) = H_I(q-1)(T)(1-\beta(q-1))/2+H_I(q)(1-T/2)+H_I(q+1)(T)(\beta(q+1)/2). \tag{17}$$

Obviously, Eq. (17) is more complex than Eq. (16). In addition to the embedding rate T, another (unpredictable) value $\beta(q)$ has been added to the right side of Eq. (17). Thus, the security level has been elevated further in the advanced model.

4 Experimental Results

To verify the undetectability of the proposed modified LSB embedding scheme, two statistical attacks are used, the Chi-square attack and RS steganalysis. These two attacks generally perform very well at detecting hidden messages embedded by LSB embedding techniques.

The test set contains 150 original images - including 8 standard images downloaded from the USC-SIPI image database [10], 75 images downloaded from the photoSIG [11], and 67 images obtained from a Panasonic Lumix FX7 digital camera. A PRNG was used to generate simulated encrypted secret messages. This is reasonable because encrypted message binary data would be indistinguishable from pseudo random binary data.

For every image, pseudorandom data was embedded using LSB embedding, basic model embedding and advanced model embedding, respectively. Varied embedding rates (from 5% to 100%) were also tested to measure the risk of exposure of the hidden message.

Fig. 5 shows the average experimental result for 150 stego-images with the same embedding rate. The x-axis shows the embedding rate from 5% to 100%. The y-axis is the Chi-square index which estimates the probability of a hidden message existing in the picture. In Fig. 5, using LSB embedding, the average Chi-square index is larger than 0.5 when the embedding rate is over 80%. Using the basic model to embed a message, the average Chi-square index is always below 0.1, no matter what embedding rate is chosen. Using the advanced model, all of the average Chi-square index values are near 0 - that is, lower than the value of original cover-image.

Fig. 6 shows the average experimental result of RS steganalysis. The x-axis shows the actual embedding rate and the y-axis is the estimated embedding rate using RS steganalysis. In Fig. 6, we can observe clearly that the embedding rate of traditional LSB embedding can be estimated precisely with RS steganalysis.

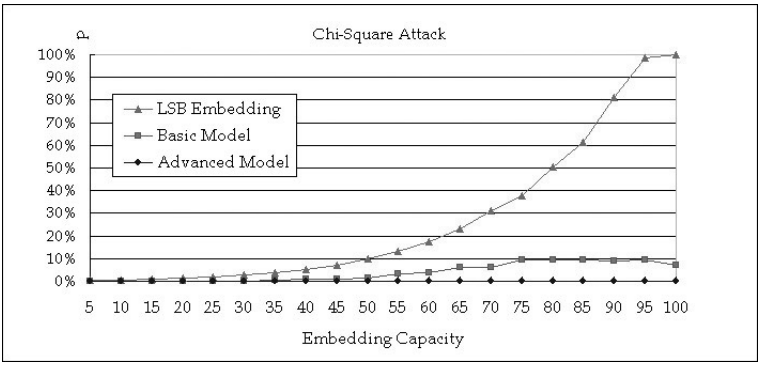


Fig. 5. Average experimental result of Chi-Square attack

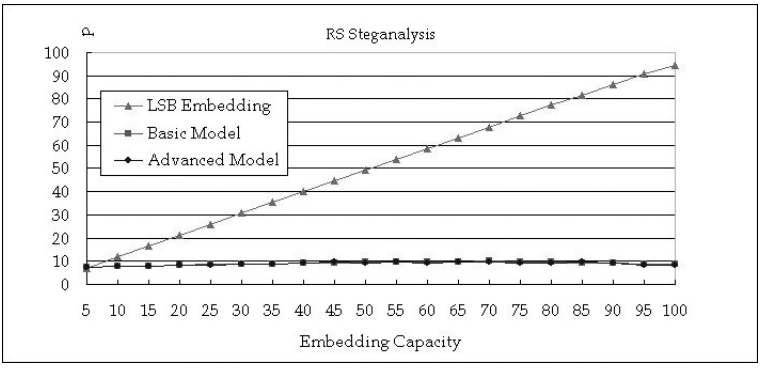


Fig. 6. Average experimental result of RS steganalysis

In contrast, when using either the basic model or the advanced model to embed the secret message, all the embedding rates estimated by RS steganalysis fall in the range 7% to 10%. This experimental result demonstrates that the proposed scheme is essentially undetectable when attacked by RS steganalysis.

5 Conclusion

The PoVs artifact caused by traditional LSB embedding exposes the existence of a hidden message. In order to raise the security level of covert communication, the weakness of the LSB embedding system has been theoretically analyzed here and a two-variant modified LSB embedding scheme has been proposed. There are three important features within the modified scheme. Firstly, the extraction process used in the proposed scheme is almost identical to the one used in traditional LSB embedding. Secondly, from a PSNR point of view, the fidelity of the stego-images resulting from the proposed scheme is as good as

those created by traditional LSB embedding. Finally and most importantly, the PoVs artifact is removed from the stego-images. Experimental results show that both of the well-known Chi-square and RS steganalysis attacks are unable to detect the existence of secret messages embedded with the new system. Using the proposed scheme is therefore more secure than using traditional LSB embedding techniques.

References

[1] Kahn, D.: The Codebreakers - the Comprehensive History of Secret Communication from Ancient Times to the Internet. Scribner, New York (1996)
[2] Johnson, N., Jajodia, S.: Exploring Steganography: Seeing the Unseen. IEEE Computer, 26–34 (February 1998)
[3] Westfeld, A., Pfitzmann, A.: Attacks on Steganographic Systems. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 61–76. Springer, Heidelberg (2000)
[4] Fridrich, J., Du, R., Meng, L.: Steganalysis of LSB Encoding in Color Images. In: IEEE International Conference on Multimedia and Expo., pp. 1279–1282 (2000)
[5] Fridrich, J., Goljan, M., Du, R.: Detecting LSB Steganography in Color and Gray Images. Magazine of IEEE Multimedia (Special Issue on Security), 22–28 (October-November 2001)
[6] Westfeld, A.: F5 - A Steganographic Algorithm High Capacity Despite Better Steganalysis. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
[7] Crandall, R.: Some Notes on Steganography. Posted on Steganography Mailing List (1998), http://os.inf.tu-dresden.de/~westfeld/crandall.pdf
[8] Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of JPEG Images: Breaking the F5 Algorithm. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 310–323. Springer, Heidelberg (2003)
[9] Sharp, T.: An Implementation of Key-Based Digital Signal Steganography. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 13–26. Springer, Heidelberg (2001)
[10] USC-SIPI image database (accessed 12th August 2008), http://sipi.usc.edu/database/
[11] photoSIG (accessed 12th August 2008), http://www.photosig.com