PAPER

Secure Bit-Plane Based Steganography for Secret Communication

Cong-Nguyen BUI[†a)], ***Nonmember***, **Hae-Yeoun LEE**[††], ***Member***, **Jeong-Chun JOO**[†], ***and*** **Heung-Kyu LEE**[†], ***Nonmembers***

SUMMARY A secure method for steganography is proposed. Pixel-value differencing (PVD) steganography and bit-plane complexity segmentation (BPCS) steganography have the weakness of generating blocky effects and noise in smooth areas and being detectable with steganalysis. To overcome these weaknesses, a secure bit-plane based steganography method on the spatial domain is presented, which uses a robust measure to select noisy blocks for embedding messages. A matrix embedding technique is also applied to reduce the change of cover images. Given that the statistical property of cover images is well preserved in stego-images, the proposed method is undetectable by steganalysis that uses RS analysis or histogram-based analysis. The proposed method is compared with the PVD and BPCS steganography methods. Experimental results confirm that the proposed method is secure against potential attacks.

key words: *steganography, steganalysis, matrix embedding, pixel-value differencing, bit-plane complexity segmentation*

1. Introduction

Secret communication is required for business transactions and military purposes. Steganography hides the existence of a secret message in cover works. As a result of the growing popularity of the Internet, many steganography methods that hide messages in slightly modified images have been studied, because human eyes are insensitive to these modifications.

Least Significant Bit (LSB) steganography, in which the LSBs of pixels are replaced with message bits, is simple to implement. However, it is vulnerable to many steganalysis methods. χ^2-statistical steganalysis methods use the pair of value distribution [1], [2]. RS steganalysis not only detects the existence of the relatively small message bits, but also reliably estimates the length of the message [3]. Although pixel-value differencing (PVD) [4] and bit-plane complexity segmentation (BPCS) [5] steganography are famous in the spatial domain, these schemes have weak points and are vulnerable to steganalysis; see [6] for PVD and [7], [8] for BPCS. These steganography and steganalysis schemes are reviewed in Sect. 2.

In this paper, a secure bit-plane based steganography method is presented that avoids the weaknesses of the BPCS and PVD methods, i.e., blocky effects and noise in smooth areas. A robust measure to select noisy blocks for message embedding is designed and a matrix embedding technique is applied to reduce the changes that are made to the cover images. The proposed method is compared with the BPCS and PVD steganography methods. Since the statistical property of cover images is well preserved, the proposed method achieves the security against steganalysis using RS analysis or histogram-based analysis including potential attacks.

The remainder of the paper is organized as follows. In Sect. 2, the use of steganography in the spatial domain and the means by which steganalysis might defeat it are explained. In Sect. 3, a secure steganography method is described. In Sect. 4, experimental results are presented. Section 5 concludes.

2. Steganography and Steganalysis

Whenever a steganography algorithm appears, steganalysts try to identify its weaknesses and defeat it. This section reviews two recent steganography methods, PVD and BPCS steganography, and their weakness against steganalysis.

2.1 PVD Steganography and Its Weakness

PVD steganography embeds message bits by changing the difference between the values of two pixels. The basic process is as follows [4]. The cover image is divided into non-overlapping blocks of two consecutive pixels, which is performed through all rows in a zigzag manner. The difference between the values, p_i and p_{i+1}, of the two pixels in the block is calculated as follows: $d = p_{i+1} - p_i$, whose range is $[-255, 255]$. $|d|$ is classified into K contiguous ranges R_k where $k = 0, 1, \ldots, K-1$ and the width R_k is a power of 2. The practical set of R_k is [0 7], [8 15], [16 31], [32 63], [64 127], [128 255] with lower bound l_k, upper bound u_k, and width w_k. Finally, if $|d|$ is in R_k, a message word with $\log_2(w_k)$ bits is embedded into the corresponding two-pixel block.

A $\log_2(w_k)$-bit message word has a decimal value b. The new difference d', which has the same range R_k with $|d|$, is calculated as follows:

$$d' = \begin{cases} l_k + b, \text{if d} \geq 0 \\ -(l_k + b), \text{if d} < 0 \end{cases} \tag{1}$$

Manuscript received April 30, 2009.
Manuscript revised September 7, 2009.

[†] The authors are with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea.

[††] The author is with the School of Computer and Software Engineering, Kumoh National Institute of Technology, Gumi, Gyeongbuk, Republic of Korea.

a) E-mail: nguyenbc@mmc.kaist.ac.kr

DOI: 10.1587/transinf.E93.D.79

Message bits are embedded by adjusting the difference between pixel values as follows:

$$(p_i', p_{i+1}') = \begin{cases} (p_i - r_c, p_{i+1} + r_f), & \text{if d is odd} \\ (p_i - r_f, p_{i+1} + r_c), & \text{if d is even} \end{cases} \tag{2}$$

where

$$r_c = \left\lceil \frac{d' - d}{2} \right\rceil, \; r_f = \left\lfloor \frac{d' - d}{2} \right\rfloor \tag{3}$$

When it is possible that a block will cause an overflow due to message embedding, *i.e.*, if the result of embedding will be that the value of p_i' or p_{i+1}' lies outside the range [0 255], that block is labeled as *unusable* and excluded. Obviously, if the portion of *unusable* blocks is large, the embedding capacity is decreased significantly.

Weakness: PVD steganography method avoids randomly changing pixel values; hence, RS steganalysis fails to defeat it. However, PVD histogram-based steganalysis can detect it [6]. Although enhanced PVD steganography methods are robust against PVD histogram-based steganalysis, the fact that the difference of pixel values is used and the property of neighboring pixels in smooth areas is erased by making noise in all bit planes can be used by steganalysis researchers to defeat it. In addition, using small block sizes e.g. 2 × 1 to find embeddable blocks for message embedding cannot be adaptive with image property.

2.2 BPCS Steganography and Its Weakness

BPCS steganography embeds message bits by replacing the bit planes of noisy binary blocks with message bits. To select the noisy binary blocks, the complexity α of the $m \times m$ size block is measured as follows [5]:

$$\alpha = \frac{k}{2 \times m \times (m-1)}, \tag{4}$$

where k is the total number of the change between 0 and 1 along all rows and columns. When the complexity α exceeds α_T, the block is regarded as a noisy binary block, where α_T is a noise threshold ($\alpha_T = 0.3$ or $\alpha_T = 0.5$).

The basic embedding process is as follows [5]. The cover image is converted into canonical gray code. After the image is decomposed into N bit planes, each bit plane is divided into non-overlapping blocks of size $m \times m$. In addition, the message bits are split into a series of blocks of size $m \times m$. If a message block is not noisy, its conjugation is generated to make a noisy binary block. Finally, the bit planes of noisy binary blocks in the cover image are replaced by the series of message blocks.

Weakness: Differently from PVD steganography, BPCS steganography does not make non-noisy blocks in bit planes into noisy blocks. However, BPCS steganography methods use large block sizes (4 × 4 or 8 × 8 pixels) as a unit to determine whether a block is noisy or not. As a result, blocky effects occur in bit planes during the replacing of the blocks of the cover image by message blocks. In addition, using a checkerboard pattern to convert message blocks into cover blocks is not always successful [12]. Steganalysis methods [7], [8] are able to exploit the blocky effects to defeat BPCS steganography. Although using small block sizes can avoid blocky effects, BPCS steganography requires more header information to keep the conjugation map.

3. Proposed Steganography Method

Several factors need to be considered if a secure steganography method is to be designed: avoiding non-noisy blocks to embed message bits (thereby avoiding the weakness of PVD steganography), measuring the complexity based on a small block size (thereby avoiding the weakness of BPCS and PVD steganography), measuring the complexity using high bit planes of pixels, and embedding into multiple bit planes of pixels simultaneously (thereby avoiding the weakness of BPCS steganography).

Taking these factors into account, we design a measure to calculate the complexity using a small block size and high bit planes of pixels. In addition, the matrix embedding technique to scatter message bits is adapted, such that noisier blocks are embedded before less noisy blocks. These techniques for message embedding and extraction are explained below.

3.1 Selecting Embeddable Blocks

Each pixel of an image I is composed of $N + 1$ bits (gray: $\underline{N = 7}$, RGB: $N = 7$ for each channel). A binary sequence $b_0..b_{N_{x,y}}$ represents a pixel value $p_{x,y}$ at position (x, y), where b_N is the least significant bit. The set of binary numbers b_i of all pixels in one block represents the bit plane B_i of that block.

Given that the proposed method embeds message bits into noisy blocks and avoids embedding them into non-noisy blocks of bit planes, a rule is formulated to identify the noisy blocks of bit plane i. We define a block whose size is 2 × 2 pixels: $\left[p_{x,y}, p_{x+1,y}; p_{x,y+1}, p_{x+1,y+1}\right]$. The block is i-bit-plane smooth when the following four conditions are satisfied at the same time:

$$\begin{cases} |\overline{b_0..b_{i-1}}_{x,y} - \overline{b_0..b_{i-1}}_{x+1,y}| \leq t \\ |\overline{b_0..b_{i-1}}_{x,y} - \overline{b_0..b_{i-1}}_{x,y+1}| \leq t \\ |\overline{b_0..b_{i-1}}_{x+1,y+1} - \overline{b_0..b_{i-1}}_{x+1,y}| \leq t \\ |\overline{b_0..b_{i-1}}_{x+1,y+1} - \overline{b_0..b_{i-1}}_{x,y+1}| \leq t \end{cases} \tag{5}$$

where the threshold t is set at 1 throughout the experiment. For example, when a 2 × 2 block has the pixel values 191, 168, 190, 155 ($\underline{101}11111$, $\underline{101}01000$,$\underline{101}11110$, $\underline{100}11011$ in binary), the four conditions above are satisfied when $i = 3$ and hence this block is 3-bit-plane smooth. All pixels of the cover image are defined as a set R. All overlapping blocks that are i-bit-plane smooth are defined as a set RS_i. Then, the set of all i-bit-plane noisy blocks is $RN_i = R - RS_i$ with

$i_S \leq i \leq 7$. The i-bit-plane noisy blocks RN_i are embeddable blocks, where message bits are embedded into low bit planes from B_i to B_7. The i-bit-plane noisy blocks will be sharp edge areas if i is small.

In our experiment, we set at $i_S = 2$. Problems of overlapping will occur during the embedding of message bits into low bit planes of a block that is i-bit-plane smooth and j-bit-plane smooth, where $j < i$. To avoid the overlapping problem, we determine, for each i, the set of embeddable blocks such that message bits are embedded only into these blocks. The set of the embeddable blocks corresponding to the set of i-bit-plane noisy blocks will include the set of all i-bit-plane noisy blocks and exclude the set of j-bit-plane noisy blocks with $j < i$, as follows:

$$RE_i = \begin{cases} RN_i & \text{if} \quad i = i_s \\ RN_i - \bigcup_{l=i_s}^{i-1} RN_l & \text{if} \quad i > i_s \end{cases} \tag{6}$$

where $i_S \leq i \leq 7$. For each i, message bits are embedded into bit-plane by bit-plane from B_i to B_N of RE_i.

3.2 Scattering Message Bits

i-bit-plane noisy blocks with small i values are noisy parts, such as edge areas. i-bit-plane noisy blocks with high i values include flat areas. Therefore, when i has high values, the embedding priority of that block should be low. To determine the embedding priority, we apply a matrix embedding technique called matrix encoding [9]–[11].

Assume that we have an n-bit codeword a for embedding a k-bit message x. A hash function f extracts k bits from a modified codeword a'. The matrix embedding technique finds a suitable modified codeword a' for every a and x with $x = f(a')$, such that the Hamming distance $d(a, a') \leq d_{max}$ ($d_{max} = 1$ [9]). We define (d_{max}, n, k) as the matrix embedding parameter, where a codeword with n places will be changed in not more than d_{max} places to embed a k-bit message. For $(1, n, k)$, the codeword length is $n = 2^k - 1$. The hash function $f(a)$ is defined as follows:

$$f(a) = \bigoplus_{j=1}^{n} a_j.j \tag{7}$$

where $\oplus$ is a bitwise XOR operation that yields 1 as result if and only if an odd number of the variables are 1. The bit position that should be changed in the codeword a is selected by calculating $s = x \oplus f(a)$. The changed codeword results in

$$a' = \begin{cases} a, \quad \text{if} \quad s = 0 \quad (x = f(a)) \\ (a_1, a_2, \ldots, \neg a_s, \ldots a_n) \quad \text{otherwise} \end{cases} \tag{8}$$

For example, a message x that has $k = 3$ bits: $x = 001_2$ is embedded into a codeword a that has $n = 7$ bits: $a = 1011111_2$. Then, the matrix embedding technique is applied with parameters $(d_{max}, n, k) = (1, 7, 3)$. So, we modify one bit in the codeword a to embed three bits of the message x as follows. First, we calculate a hash value $f(a)$.

$$\begin{aligned} &f(a) \\ &= \bigoplus_{i=1}^{n} a_i.i \\ &= 001_2 \oplus 000_2 \oplus 011_2 \oplus 100_2 \oplus 101_2 \oplus 110_2 \oplus 111_2 \\ &= 010_2 \end{aligned} \tag{9}$$

Then, we find the bit position s to change: $s = x \oplus f(a) = 001_2 \oplus 010_2 = 011_2 = 3_{10}$. As a result, the modified codeword will be $a' = 1001111_2$. To extract x from a', we have

$$x = f(a') = \bigoplus_{i=1}^{n} a'_i.i = 001_2 \tag{10}$$

The proposed method modifies fewer bits in the noisy blocks of lower bit planes (RN_i having high i values). With high k values, the length of codeword a is long with $n = 2^k - 1$ bits and only one bit is modified in $2^k - 1$ bits. Therefore, k is set to increase as i increases. In our experiment, we embedded message bits into i-bit-plane noisy blocks with $i_S = 2 \leq i \leq 7$ and $k = i - 1$.

In addition, a pseudo-random sequence is used to scatter message bits more widely. This sequence indexes the embeddable bits in bit-planes. Hence, a sender and a receiver who have the same steganography key can generate the same pseudo-random sequence. In our experiment, an ISAAC (Indirection, Shift, Accumulate, Add, and Count) pseudo-random generator [13] is adapted to generate a random sequence.

3.3 Embedding Process

The secret message is compressed and encrypted with a cryptography scheme to increase the security. Then, it is embedded into the bit planes of cover images. The embedding process, which uses the complexity measure and matrix embedding technique, is depicted in Fig. 1.

(E1) A cover image I is decomposed into $(N + 1)$ bit planes $B_0, .., B_N - 1, B_N$ and i is set at $i_S = 2$. **(E2)** A secret message M is compressed and encrypted with a cryptographic key K_C to stream message bits M'. **(E3)** The ISAAC pseudo-random generator is initiated with a steganographic key K_S to generate a pseudo-random sequence that indexes the bits that can be embedded in bit planes. **(E4)** Message bits are embedded into bit planes by repeating i from i_S to $N = 7$. **(E4.1)** After the set of i-bit-plane smooth blocks RS_i has been selected by identifying those blocks that meet the four conditions for all overlapping 2×2 blocks in the image I, the set of embeddable i-bit-plane noisy blocks RE_i is acquired. **(E4.2)** The matrix embedding parameters k, n are calculated and the embedding loop index e is initiated as N. **(E4.3)** The message bits M' are embedded into the bit plane B_e of RE_i. This embedding step is repeated with $e = e - 1$ until $e = i$ or until the end of the message bit M'. **(E4.4)** When the end of the message bits M' is reached, i.e., once all of the message bits M' have been embedded, the embedding process is complete and the stego-image I' is returned. **(E4.5)** If $e = i$ is not satisfied, the embedding process is performed on the bit plane B_e of RE_i. Remaining message bits

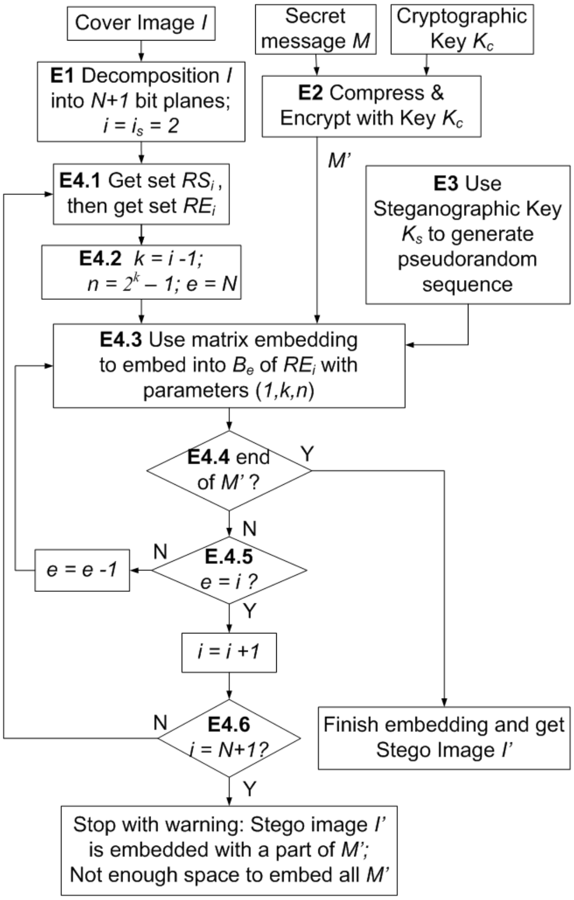
Cover Image I
Secret message M
Cryptographic Key Kc
E1 Decomposition I into N+1 bit planes; i = is = 2
E2 Compress & Encrypt with Key Kc
M'
E3 Use Steganographic Key Ks to generate pseudorandom sequence
E4.1 Get set RSi, then get set REi
E4.2 k = i -1; n = 2^k – 1; e = N
E4.3 Use matrix embedding to embed into Be of REi with parameters (1,k,n)
E4.4 end of M' ?
Y
N
e = e -1
E.4.5 e = i ?
Y
i = i +1
N
E4.6 i = N+1?
Y
Finish embedding and get Stego Image I'
Stop with warning: Stego image I' is embedded with a part of M'; Not enough space to embed all M'

Fig. 1 Embedding process.

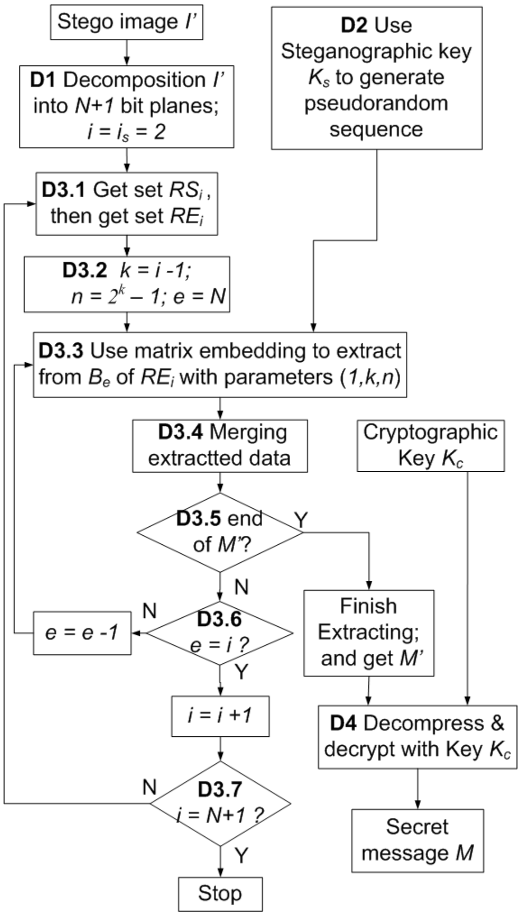
Stego image I'
D2 Use Steganographic key Ks to generate pseudorandom sequence
D1 Decomposition I' into N+1 bit planes; i = is = 2
D3.1 Get set RSi, then get set REi
D3.2 k = i -1; n = 2^k – 1; e = N
D3.3 Use matrix embedding to extract from Be of REi with parameters (1,k,n)
D3.4 Merging extractted data
Cryptographic Key Kc
D3.5 end of M'?
Y
N
N
e = e -1
D3.6 e = i ?
Y
Finish Extracting; and get M'
i = i +1
D4 Decompress & decrypt with Key Kc
N
D3.7 i = N+1 ?
Y
Secret message M
Stop

Fig. 2 Extracting process.

are embedded continuously into the next bit plane B_{e-1} of RE_i until $e = i$. When $e = i$, step E4.1 is performed with $i = i + 1$ until the condition $i = N + 1$ is satisfied. **(E4.6)** If $i = N + 1$ and message bits M' remain, the proposed method is not able to embed all the message bits M' into the image that was chosen for embedding.

3.4 Extracting Process

Due to the fact that the secret message was embedded into the bit planes from B_i to B_N of i-bit-plane noisy blocks, the noisy blocks should be identified using only stego-images. In other words, the proposed method does not use cover images during extraction. The extraction process is depicted in Fig. 2.

(D1) A stego-image I' is decomposed into $(N + 1)$ bit planes $B_0, .., B_N - 1, B_N$. **(D2)** The ISAAC pseudo-random generator is initiated with a steganographic key K_S and generates a pseudo-random sequence, which is the same as that used during the embedding process. **(D3)** Embedded message bits are extracted from bit planes by repeating i from i_S to $N = 7$. **(D3.1)** The set of i-bit-plane noisy blocks RE_i are identified using a process similar to that which was used during the embedding process. **(D3.2)** Matrix embedding parameters k, n, are calculated and the loop index e is initiated as N. **(D3.3)** Message bits are extracted from the bit plane B_e of RE_i by using the pseudo-random sequence. **(D3.4)** The extracted message bits are merged to check whether the extraction process has been completed. **(D3.5)** This extraction step is repeated with $e = e - 1$ until $e = i$ or the end of the message bits. **(D3.6)** When the end of the message bits is reached, the extraction process is complete, i.e., all the message bits M' have been extracted. **(D3.7)** If $e = i$ is not satisfied, the extraction step is performed on the bit plane B_e of RE_i. Remaining message bits are continuously extracted from the next bit plane B_{e-1} of RE_i until $e = i$. When $e = i$, step D3.1 is performed with $i = i + 1$ until the condition that $i = N + 1$ is satisfied. If $i = N + 1$, the extraction process stops because there is no bit plane that is lower than $N + 1$ bit-plane. **(D4)** After the extraction process has been completed, the message bits M' are decompressed and decrypted with the cryptographic key K_C to get the secret message M.

4. Experimental Results

The proposed method was tested on a set of classical images and Kodak photographs (http://r0k.us/graphics/kodak) and the results compared with those yielded by the BPCS and PVD steganography methods. The images were num-

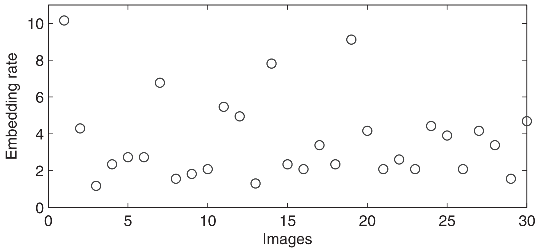

Fig. 3 Embedding capacity of the presented method (%).

bered in the following order: Baboon, House, Crowd, Lena, Peppers, F16 (512 × 512 pixels) and 24 Kodak images (512 × 768). The BPCS steganography was implemented with threshold $\alpha_T = 0.5$ and 4×4 block size, where header information about the conjugation map was not considered when calculating the capacity. The PVD method was implemented with a set of R_k as follows: [0 7], [8 15], [16 31], [32 63], [64 127], and [128 255].

Figure 3 depicts the maximum embedding capacity of the proposed method, which depends on the images that are chosen for embedding and has values of around 1.2% to 10.2%. For example, a 26 KB message was embedded into the 258 KB Baboon image (the embedding rate was 26/258 ≈ 10.2%). The maximum embedding capacity of the proposed method was lower than that of the BPCS and PVD steganography methods. However, the BPCS and PVD steganography methods reveal their weakness at high embedding rates, as is described in Sect. 4.1. The only way to increase the security of the BPCS and PVD steganography methods is by using low embedding rates.

In order to compare steganography methods under RS steganalysis and histogram based steganalysis (Sects. 4.2 and 4.3), the same low embedding rate was applied for each method. First, we applied the proposed steganography method with maximum embedding rates to obtain the embedded image and the embedded message bits for each image. Then, the same message bits were embedded using the BPCS and PVD steganography methods. Hence, the embedding rate is the same for each of the proposed, BPCS, and PVD steganography methods.

For each image, Fig. 4 depicts Peak Signal-to-Noise Ratio (PSNR) values with the embedding rates stated in Fig. 3 under the three steganography methods. The PSNR values of the proposed method were lower than those of other steganography methods. However, at the low embedding rate, the PSNR is not a good measure of security because the PSNR values of stego-images are always high. When the PSNR values are high, the naked eye cannot discriminate between cover images and stego-images.

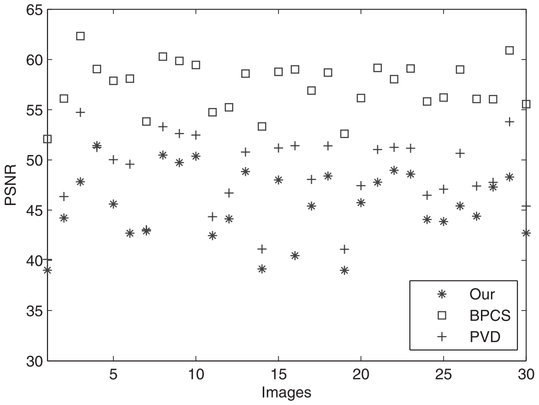

Fig. 4 PSRN values of stego-images by the presented, BPCS and PVD method at the same embedding rate.

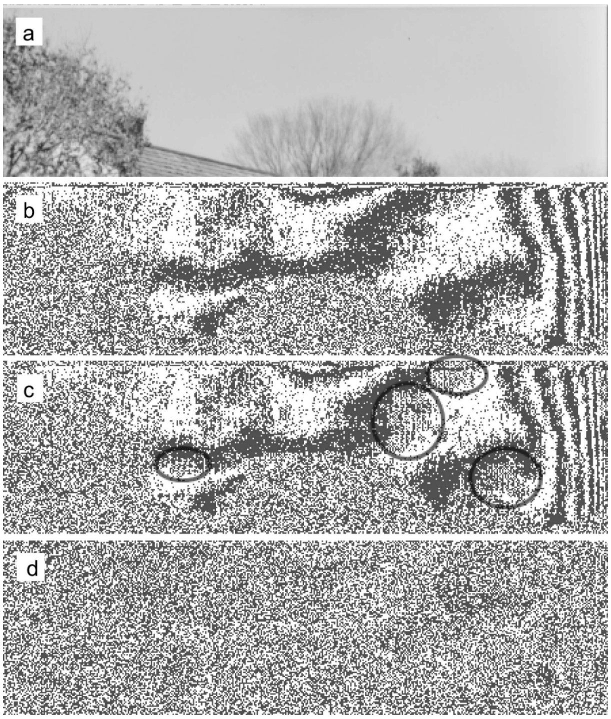

Fig. 5 Bit-plane view of each steganography method. (a) Cover image, (b) bit plane B_6 of the cover image, (c) bit plane B_6 of the stego-image produced by BPCS steganography, and (d) bit plane B_6 of the stego-image produced by PVD steganography.

4.1 Analysis with Bit-plane View

Although a high rate of embedding can be achieved when using BPCS and PVD steganography, their weaknesses can be seen easily at high embedding rates. Figure 5 depicts the bit plane B_6 of a stego-image at an embedding rate 20%, using BPCS and PVD steganography. With BPCS steganography, bit plane B_6 has visible noise in cycle areas. As mentioned in Sect. 2, PVD steganography makes noise in flat areas of bit planes. Hence, with PVD steganography, bit-plane B_6 did not have visible shapes like the bit plane B_6 of the cover image in any areas. Given that these artifacts are easily noticed with the naked eye, BPCS and PVD steganography can be defeated by visual analysis of the bit plane.

4.2 Analysis with RS Steganalysis

RS steganalysis was designed to defeat LSB steganography and is able to estimate the message length [3]. RS steganalysis begins by dividing a cover image into disjoint groups of n adjacent pixels $G = (x_1, .., x_n)$. In practice, $n = 4$, which represents consecutive pixels in a row. The discrimination function f measures the difference in value between n pixels in the group as follows:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{11}$$

The invertible function F on a pixel value x is defined by three cases. The first case is an identity permutation function F_0: $F_0(x) = x$. The second case is a LSB flipping function F_1, which flips the LSB of each gray level: $0 \leftrightarrow 1, 2 \leftrightarrow 3, 4 \leftrightarrow 5, \ldots, 254 \leftrightarrow 255$. The third case is a shifted LSB flipping function F_{-1}: $F_{-1}(x) = F_1(x+1) - 1$: $-1 \leftrightarrow 0, 1 \leftrightarrow 2, 3 \leftrightarrow 4, \ldots, 253 \leftrightarrow 254, 255 \leftrightarrow 256$. After the function F is applied to a group G, three groups can be generated: regular group R, singular group S, and unusable group U.

$$\begin{array}{l} \text{R: } f(F(G)) > f(G) \\ \text{S: } f(F(G)) < f(G) \\ \text{U: } f(F(G)) = f(G) \end{array} \tag{12}$$

where $F(G) = F(x_1), \ldots, F(x_n)$. The various kinds of flipping are applied to different pixels in the group G decided by a mask M, which is a n-tuple with values -1, 0 and 1 (in [3], $M = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}$). Let the number of the regular R and singular S group for the mask M be R_M and S_M, respectively, and let the corresponding values for the negative mask $-M$ be R_{-M} and S_{-M}, respectively (as percentages of all groups). In [3], $R_M \approx R_{-M}$ and $S_M \approx S_{-M}$ are satisfied in typical images. When an image breaks these rules, it is concluded on the basis of the RS steganalysis that the image includes secret message bits.

In our experiment, we defined a RS difference rate RS_{dif} as follows to check whether an image breaks the rules and thereby to detect the existence of the secret message:

$$RS_{dif} = \frac{|R_M - R_{-M}|}{R_M} + \frac{|S_M - S_{-M}|}{S_M} \tag{13}$$

If RS_{dif} is high, RS steganalysis can detect a secret message. Figure 6 depicts the RS_{dif} for test images. Although the BPCS steganography method was performed with low embedding rates, the LSB bit plane was modified; hence, RS steganalysis sensitive to LSB modification could defeat it. Table 1 summarizes the mean and standard derivation of the RS_{dif} on 30 test images. The the presence of a message embedded using BPCS steganography could be detected because the statistical difference between stego-images and cover images was large. However, the proposed and PVD steganography methods overcame RS steganalysis at the same embedding rate.

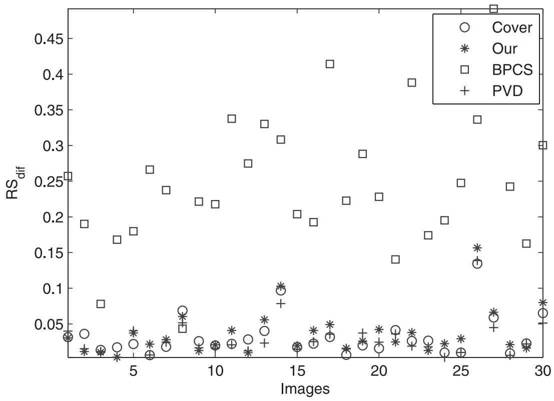

Fig. 6 RS steganalysis on the BPCS, PVD, and proposed steganography methods.

Table 1 RS steganalysis on the BPCS, PVD, and proposed steganography methods: the mean and standard deviation for 30 test images.

Method	Cover	Our	BPCS	PVD
Mean	0.032	0.037	0.245	0.029
Std.	0.028	0.032	0.094	0.027

4.3 Analysis with PVD Histogram Steganalysis

The PVD histogram-based steganalysis method was applied to stego-images produced by the proposed, BPCS, and PVD steganography methods. In all images, the PVD histogram of the BPCS and PVD steganography methods was not well preserved. The PVD steganography method caused fluctuation and step effects in the histogram. In addition, the BPCS steganography method caused fluctuation and step effects, although these were less severe in this case than in the case of the PVD steganography method. Figure 7 depicts the PVD histogram from the three steganography methods at the same embedding rate in the Baboon image. The use of the BPCS and PVD steganography methods were detectable by PVD histogram-based steganalysis. However, the PVD histogram was almost unchanged when the proposed method was used. Therefore, the proposed method is robust against this steganalysis.

5. Conclusion

Recent BPCS and PVD steganography methods on the spatial domain have been studied to enhance their security against their steganalysis. However, they still have weaknesses that can be exploited, such as histogram fluctuation and blocky effects.

In this paper, a secure steganography method based on embedding message bits into multi bit-planes of cover images was presented. We designed a complexity measure to locate the noisy blocks for message embedding and adopted the matrix embedding technique to reduce the change of cover images. The proposed method was tested against RS steganalysis and PVD histogram-based steganalysis and the results compared with the results of using BPCS and PVD steganography methods at the same embedding rates. The experiment results confirmed that the proposed method is more robust against steganalysis than other steganography

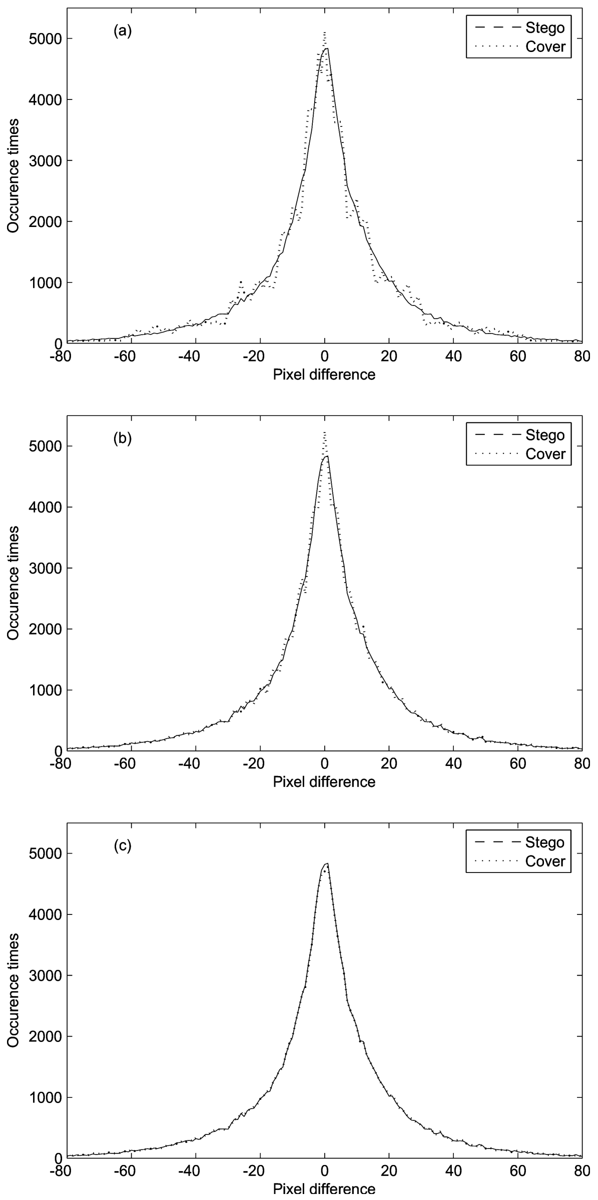

Fig. 7 PVD histogram of the Baboon image, before and after message embedding by (a) the PVD method, (b) the BPCS method, and (c) the proposed method at the embedding rate = 10.2%.

methods, although the maximum embedding capacity was low. In future work, the proposed method will be applied to images on the wavelet domain. In addition, its robustness against various steganalysis methods will be analyzed.

Acknowledgement

This research was supported by NRL (National Research Lab) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (No. R0A-2007-000-20023-0), and by the Ministry of Culture, Sports and Tourism (MCST) and Korea Culture Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2009.

References

[1] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," Lect. Notes Comput. Sci., vol.1768, pp.61–76, 2000.

[2] N. Provos, "Defending against statistical steganalysis," Proc. 10 USENIX Security Symposium, vol.10, pp.323–335, 2001.

[3] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," Proc. ACM Workshop on Multimedia and Security, pp.27–30, Ottawa, CA, 2001.

[4] D.C. Wu and W.H. Tsai, "A steganographic method for images by pixel-value differencing," Pattern Recognit., vol.24, no.9–10, pp.1613–1626, 2003.

[5] E. Kawaguchi and R.O. Eason, "Principle and applications of BPCS-Steganography," Proc. SPIE: Multimedia Systems and Applications, vol.3528, pp.464–472, 1998.

[6] X. Zang and S. Wang, "Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security," Pattern Recognit., vol.25, no.3, pp.331–339, 2004.

[7] M. Niimi, T. Ei, and H. Noda, "An attack to BPCS-steganography using complexity histogram and countermeasure," Proc. IEEE Int. Conf. on Image Processing, vol.5, pp.733–736, 2004.

[8] S. Tan, J. Huang, and Y.Q. Shi, "Steganalysis of enhanced BPCS steganography with the hilbert-huang transform based sequential analysis," Proc. 6th Int. Workshop on Digital Watermarking, pp.112–126, 2008.

[9] A. Westfeld, "High capacity despite better steganalysis (F5. A Steganographic Algorithm)," Lect. Notes Comput. Sci., vol.2137, pp.289–302, 2001.

[10] R. Crandall, Some notes on steganography, 1998, http://os.inf.tu-dresden.de/˜westfeld/crandall.pdf

[11] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," Proc. SPIE, vol.6072, pp.727–738, 2006.

[12] H. Hirohia, "A Data embedding method using BPCS principle with new complexity measures," Proc. Pacific Rim Workshop on Digital Steganography, pp.30–47, 2002.

[13] B. Jenkins, ISAAC: a fast cryptographic random number generator, 1996, http://www.burtleburtle.net/bob/rand/isaacafa.html

Cong-Nguyen Bui received his M.S. degree in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2005. He is now pursuing the Ph.D. degree at KAIST, Korea. His major interests are steganography, steganalysis, and digital watermarking.

Hae-Yeoun Lee received his M.S. and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1997 and 2006, respectively. From 2006 to 2007, he was a Post-doctoral Researcher at Weill Medical College, Cornell University, USA. He is now with Kumoh National Institute of Technology, Korea. His major interests are digital watermarking, image processing, remote sensing, and digital rights management.

Jeong-Chun Joo received a B.S. degree in Computer Science from Korea Military Academy (KMA), Korea, in 1996 and M.S. degrees in Computer Science from Korea National Defense University (KNDU), Korea, in 2002. He is currently a Ph.D candidate in Computer Science at Korea Advanced Institute of Science and Technology (KAIST). His major interests are steganography and steganalysis.

Heung-Kyu Lee received a B.S. degree in Electronic Engineering from Seoul National University, Seoul, Korea, in 1978, and M.S. and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1981 and 1984, respectively. Since 1986, he has been a Professor in the Department of Computer Science, KAIST. His major interests are digital watermarking, digital fingerprinting, and digital rights management.