Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090547

New Reversible Data Hiding Scheme for Encrypted Images using Lattices

Young-Sik Kim[1], *Kyungjun Kang*[2] *and Dae-Woon Lim*[2,*]

[1] Department of Information and Communication Engineering, Chosun University, Gwangju 501-759, Korea
[2] Department of Information and Communication Engineering, Dongguk University, Seoul 100-715, Korea

Received: 7 July, 2014, Revised: 18 Nov, 2014, Accepted: 13 Dec. 2014
Published online: 1 Sep. 2015

Abstract: Reversible data hiding is a technique to hide arbitrary data, without influencing the original images. In 2011, Zhang proposed a reversible data hiding scheme for encrypted images, by using a spatial correlation of the decrypted original image. Later, Hong *et al.* proposed an improved scheme by using side-match techniques, and modifying the correlation calculation function. In this paper, we propose an improved reversible data hiding scheme for encrypted images with lower bit error rates with the same PSNR (Peak Signal-to-Noise Ratio), by introducing a lattice pattern to confine pixels to be used for embedding, and modifying the correlation calculation function, which extracts more information from neighbor pixels. In the proposed scheme, it is possible to hide more data, because the error probability becomes zero for smaller block sizes, than with previous schemes.

Keywords: Data hiding, encrypted image, image recovery, reversible data hiding, spatial correlation.

1 Introduction

Reversible data hiding is a technique to hide arbitrary data, without influencing the original audio and image files [1],[2]. The reversibility is important in certain applications, such as military and medical ones. They are required to conserve original images, to make a correct decision that is based on the original images. Today, many reversible data hiding schemes have been proposed. The differences between two consecutive pixels are used to generate a new least significant bit plane that embeds additional hidden information in difference expansion schemes [3]. Celik *et al.*, proposed another method to hide data, using a lossless data compression for the creation of a space to store additional information [4]. Ni *et al.*, proposed a reversible data hiding method using a histogram shift that exploits the zero and peak points of the histogram of original images to embed hidden data in the image [5]. In 2010, Luo *et al.*, proposed a reversible image watermarking, using interpolation techniques that exploit the interpolation-error, the difference between interpolation value and the corresponding pixel value [6].

There is another line of research for reversible data hiding for encrypted images. In 2011, Zhang proposed a reversible data hiding scheme. This is applied for encrypted messages by a homomorphic encryption scheme [7]. In this scheme, for data hiding, the encrypted image pixels are divided into several groups of s^2 pixels where s is the number of pixels on the horizontal or vertical line of a group, and using a data hiding key, the pixels in a group are pseudo-randomly partitioned into two subsets, S_0 and S_1. In Zhang's scheme, the three least significant bits of pixels in S_0 or S_1 are inverted, according to the hidden bit.

In order to reconstruct the hidden data that is embedded in the original image, the data hider who knows the data hiding key should separate s^2 pixel groups into two pixel sets S_0 and S_1, after firstly decrypting the encrypted image. Then, the data hider generates two hypothesis groups H_0 and H_1. Depending on hidden data, one hypothesis is the same as the corresponding block in the original image, and the other is false, as every pixel in the block is distorted. Thus, it is decided which group has the lower deviation, in terms of the spatial correlation value. Because every pixel in the false hypothesis has its three least significant bits inverted, the difference between the spatial correlation values that is used to reconstruct the original data is not high enough. In turn, the error probability is also not small.

* Corresponding author e-mail: daewoonlim@gmail.com

Later, Zhang *et al.* [8] modified the original Zhang's scheme, by using pseudo-random sequence modulation to hide embedded data. Hong *et al.* [9] proposed an improved scheme by modifying the correlation calculation function, and using the side math techniques. Hong *et al.* [10] modified the scheme in [9], in order to be able to apply cartoon images, which have more flat area. Recently, Ma *et al.* [11] proposed another approach for reversible data hiding, which reserves room for data hiding before encryption. Karim and Wong [12] proposed a reversible data hiding scheme in the encryption domain, which applies entropy encoding to cipher-text. Zhang *et al.* [13] suggested another method to estimate some information, before applying encryption, which embeds and extracts hidden data without inducing any errors in the original images. In addition, Zhang also proposed a separable reversible data hiding for encrypted images, which allows to extract hidden data without decryption [18] and Zhang *et al.* also proposed a new efficient reversible data hiding based on lossless compression of encrypted images [19].

Here, we will focus on Zhang's original scheme [7] and its improvement by Hong *et al.* [9]. In the reversible data hiding schemes by Zhang and Hong *et al.*, depending on block sizes, the probability of occurrence of errors is not zero, where the error is defined as the difference between the hidden information and reconstructed information. Since error in the reconstruction of hidden data will induce distortion of the original image, this means that in a strict sense it is not reversible, for those sizes. However, by increasing the size of blocks, eventually we can obtain error-free size, real reversible data hiding. Increasing the block size means reducing the size of hidden data. Therefore, it is an important issue to reduce the size of error-free data, for a given image.

In this paper, we propose a reversible data hiding scheme for encrypted images, by improving Zhang's and Hong *et al.*'s schemes. We improve the previous schemes in two aspects. First, only pixels over the lattice pattern are able to change their values and they are divided into two subsets, as in Zhang's scheme for extracting hidden bits, and reconstructing the original image. Half of the pixels that are only over the lattice are selected, by using a pseudo random number generator, and can be modified to hide information. Among the five pixels (left/right/up/down/center), only the center pixel can be changed for hiding data. Using this, it is also possible to reduce the error probability in the reconstruction stage, which utilizes the deviation between the spatial correlation values. Second, we try to modify the fluctuation function to extract more information, in order to reduce error probability.

The remainder of this paper is organized as follows: In Section 2, the previous schemes are reviewed. Then, Section 3 explains the proposed schemes. In Section 4, the proposed scheme is analyzed, and finally the paper is concluded in Section 5.

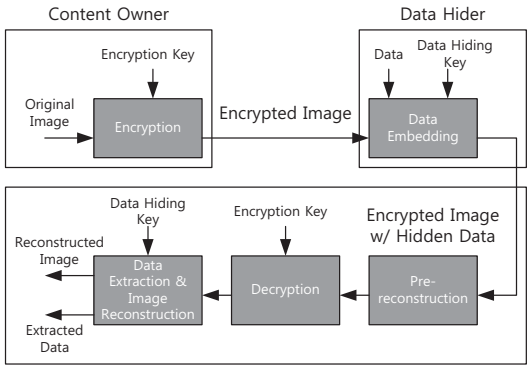

Fig. 1: Proposed reversible data hiding scheme.

2 Background Knowledge

In this section, the previous data hiding schemes for encrypted images proposed by Zhang and Hong *et al.* are reviewed [7], [9]. There are two users at data hiding stage in previous schemes, the data owner and data hider, as in Fig. 1. They can be distinct, and do not need to share any information.

2.1 Reversible Data Hiding for Encrypted Images by Zhang

At the first step, it is assumed that a user who owns the original image encrypts the image, using a secret key. Note that the encryption scheme should satisfy the homomorphic property, that is, the result of an operation in a cipher-text should be the same as the cipher-text of the result of the same operation in the corresponding plain-text. The well-known homomorphic encryption schemes are synchronous stream ciphers. Even though there exist other homomorphic encryption schemes, many of them are impractical [14,15,16,17]. The synchronous stream ciphers exploit the XOR operation between a given message, and the same length of the secret key. This secret key is generated by a pseudo random number generator.

After an encrypted image is given, the data hider (another user) is able to hide the data into the encrypted image, due to the homomorphic property. For data hiding, firstly the data hider divides the image pixels into blocks of s^2 pixels for an integer s, as presented in Fig. 2. For every block of s^2 pixels in Fig. 2, only one bit of information can be hidden. To hide a secure bit in the block, a block with s^2 pixels is partitioned into two subsets S_0 and S_1, which can be constructed by random selection of half of pixels in the block, by using a pseudo-random number generator with the data hiding key, as presented in the center and right parts of Fig. 3. If the secret bit is i ($i = 0$ or 1), then the three least

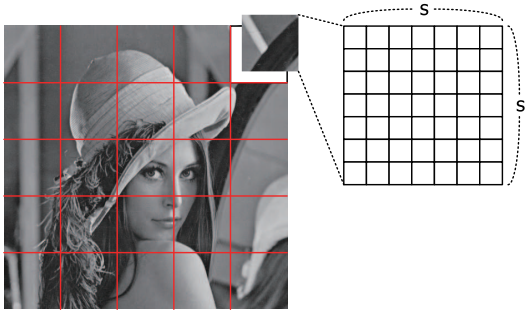

Fig. 2: An image which is possibly encrypted is partitioned into $s \times s$ blocks. Each block will be used to contain one bit of hidden information.

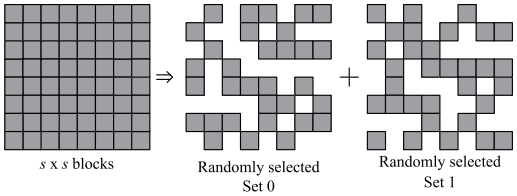

Fig. 3: Data embedding for encrypted images in Zhang and Hong *et al.*'s schemes.

significant bits of every pixel in the subset S_i are inverted, while the pixels in the subset S_{1-i} remain unchanged.

In Zhang's scheme, the encryption can be carried out both before and after data embedding, due to the homomorphic property. However, the decryption should be carried out before the reconstruction of the embedded data, as it is unable to compute the correct spatial correlation between adjacent pixels, due to the randomized pixels after the encryption.

In order to reconstruct the hidden data, the data reconstructor should use the same data hiding key, to obtain exactly the same subsets S_0 and S_1, after the image is decrypted. However, the data reconstructor cannot directly know the hidden information, even though he or she knows the data hiding key. Instead, he or she has to estimate the hidden information. For the estimation of the hidden data, he or she constructs two hypothesis groups H_0 and H_1, in which the three least significant bits of every pixel in S_i ($i = 0, 1$) are inverted. If a hidden bit is i, then no pixel in H_i is inverted (i.e., true hypothesis), while every pixel in H_{1-i} is inverted (i.e., false hypothesis). According to the hidden bit, one of H_0 or H_1 is considered as correct. In order to decide a true hypothesis, the following fluctuation function is used to estimate the spatial correlation values for pixels in each hypothesis.

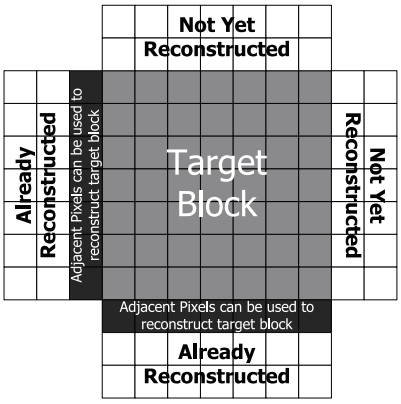

Fig. 4: Example of the side-match technique. The side pixels of the left and bottom blocks are used to reconstruct the target block (center).

$$f_Z = \sum_{u=2}^{s-1}\sum_{v=2}^{s-1}\left|p_{u,v} - \frac{p_{u-1,v} + p_{u,v-1} + p_{u+1,v} + p_{u,v+1}}{4}\right|. \tag{1}$$

The fluctuation function in (1) accumulates the differences between the center pixel and the average of the four neighboring bits (up/down/left/right), except for the outermost pixels of the image. The fluctuation values for H_0 and H_1 are denoted as f_0 and f_1. Then, the hypothesis with a smaller value is considered as a correct hypothesis. That is, if $f_i < f_{1-i}$, then i is considered as a hidden value. Finally, the original image can be recovered, by restoring the inverted bits.

2.2 Hong et al.'s Improvement: Side-Match

While the four borders of each block are not inverted, and do not join the calculation of block smoothness in Zhang's scheme, Hong *et al.* proposed a data hiding scheme which can exploit the four border blocks of each block [9]. Hong *et al.* changes the fluctuation function as

$$f_H = \sum_{u=1}^{s_2}\sum_{v=1}^{s_1-1}|p_{u,v} - p_{u,v+1}| + \sum_{u=1}^{s_2-1}\sum_{v=1}^{s_1}|p_{u,v} - p_{u+1,v}|. \tag{2}$$

This fluctuation function accumulates differences between the center pixel and horizontal or vertical pixels. As in Zhang's scheme, three LSB of pixels in the sets S_0 and S_1 of each blocks are inverted, according to hidden bits in the data embedding phase. For reconstruction, two hypothesis group H_0 and H_1 are generated, in order to extract the hidden bit, and reconstruct the original images. For each hypothesis groups H_0 and H_1, an initial

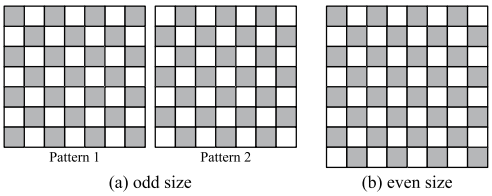

Fig. 5: Proposed lattice patterns.

calculation to evaluate the fluctuation values denoted by f_{H_0} and f_{H_1} is carried out using (2). Then, the difference $A_{m,n} = |f_{H_0} - f_{H_1}|$ for m,n is calculated, and sorted in descending order. Then, the reconstruction process is started, according to the order of the sorted values $A_{m,n}$. For reconstructing each block, if there are already reconstructed blocks among four surrounding blocks, side pixels in the other blocks are used to reconstruct the given block. This is called the side-match technique [9]. Before reconstruction, pixels in a block can have slightly different information from the original image. Once reconstruction is completed, if the block is correctly reconstructed, we have the exact information to be used for reconstructing adjacent blocks.

3 Improved Reversible Data Hiding Method for Encrypted Images

In this section, we propose an improved reversible data hiding scheme for encrypted images. Improvements take place according to two points. Firstly, we confine pixels that can be changed during the data embedding process in a specified location, called a lattice. Secondly, we modify the fluctuation function, to extract more information between the two hypothesis sets.

3.1 Introduction of the Lattice

In the previous schemes by Zhang and Hong *et al.*, at the reconstruction stage, one is a true hypothesis, whose pixels are the same as the original image, and the other is a false hypothesis, whose pixels have the three least significant bits (LSB) inverted. That is, all neighboring pixels in the false hypothesis are changed, and therefore, for calculating fluctuation functions in (1) or (2), the comparative information with the center pixels does not reflect the original image, but the fluctuated image from the embedding process.

The proposed lattice patterns are demonstrated in Fig. 5. As presented in Fig. 5, depending on the sizes of blocks, the pattern can be divided into two types, for odd and even patterns. Considering Fig. 5, it is easy to see that the proposed lattices can be represented as a matrix,

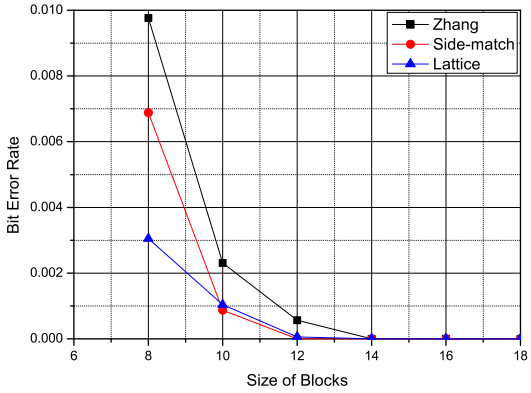

Fig. 6: Performance comparison between previous schemes and lattice with the fluctuation function by Zhang.

$L = [l_{i,j}]$ $(0 \le i,j \le s-1)$, where, for even s and Pattern 1 of odd s,

$$l_{i,j} = \begin{cases} 1, & \text{if } i \equiv 0 \bmod 2 \text{ and } j \equiv 0 \bmod 2 \text{ or} \\ & \quad i \equiv 1 \bmod 2 \text{ and } j \equiv 1 \bmod 2 \\ 0, & \text{otherwise} \end{cases}$$

and, for Pattern 2 of odd s,

$$l_{i,j} = \begin{cases} 1, & \text{if } i \equiv 0 \bmod 2 \text{ and } j \equiv 1 \bmod 2 \text{ or} \\ & \quad i \equiv 1 \bmod 2 \text{ and } j \equiv 0 \bmod 2 \\ 0, & \text{otherwise.} \end{cases}$$

Then, let P be a matrix of pixels in a $s \times s$ block. Then, the pixels over the lattices can be represented by

$$B = [b_{i,j}] = P \circ L = [p_{i,j} \times l_{i,j}] \tag{3}$$

where, $0 \le i,j \le s-1$ and $\circ$ means the Hadamard product, element-wise product of matrices. Also, define the complementary matrix of B as $C = [c_{i,j}]$ $(0 \le i,j \le s-1)$, that is,

$$B + C = P \text{ and } B \circ C = \mathbf{0} \tag{4}$$

where, $\mathbf{0}$ means the all-zero matrix. That is, if $b_{i,j}$ is equal to the pixel located in coordinate (i,j) in the given block, $c_{i,j} = 0$ and $c_{i-1,j}, c_{i+1,j}, c_{i,j-1}$, and $c_{i,j+1}$ are equal to the corresponding pixels in the block.

Before explaining the modified fluctuation function, it is worthwhile checking the effect of the lattice. Figs. 6 and 7 show the effect of directly introducing the lattice when we use the fluctuation function proposed by Zhang for the original Zhang's scheme and side-match by Hong *et al.*, respectively. By introducing the lattice, it is

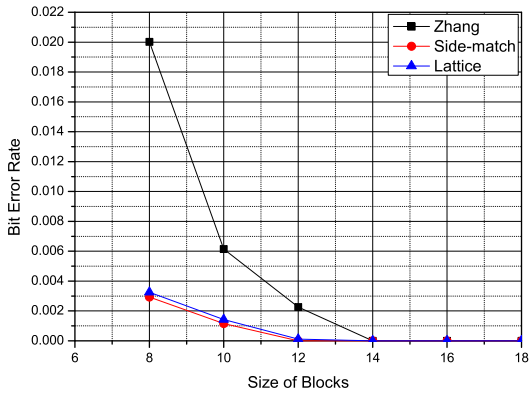

Fig. 7: Performance comparison between previous schemes and lattice with the fluctuation function by Hong *et al.*

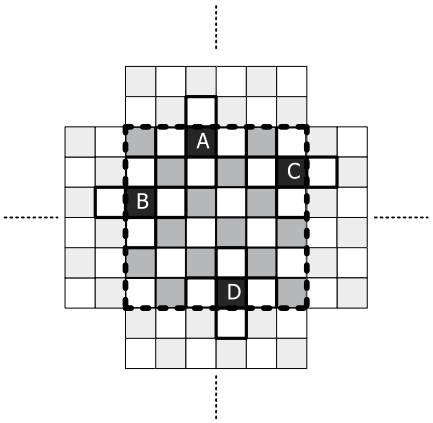

Fig. 8: Example of lattice with the side-match technique.

possible to obtain better BER performance than in Zhang's scheme, and comparable BER performance to Hong *et al.*'s side-match technique, even though we do not modify the fluctuation function as in [7] and [9] to optimize the BER performance. Note that the PSNR (Peak Signal-to-Noise Ratio) of the case of using the lattice is 40.9 dB, while that of the previous schemes is 37.9 dB, since only half of the pixels participate in the embedding process, due to the constraint of the lattice. That is, the lattice shows almost the same BER performance with better PSNR characteristics than that of Hong *et al.*'s scheme.

In Fig. 8, one reason for better performance with the lattice pattern is demonstrated. For a given block (in this example, 6×6 blocks), every side pixel is unchanged, even for pixels on edges of the block. For example, for 'A', 'B', 'C', and 'D' pixels, four neighbor pixels can be unchanged, even if the pixels are not located in the same block. Therefore, it is always possible to compare the center pixels with the pixels from original images. Note that one of the advantages of the proposed scheme is that it will show clear difference between true and false hypotheses in a cartoon image with more flat color area because every neighboring pixel is never changed, and thus cartoon images also can be applied by the proposed scheme [10].

In contrast to this, not only in Zhang's scheme, but also in Hong *et al.*'s scheme, all neighboring and center pixels in false hypothesis have their lower three bits inverted during the embedding process. This change can randomly cancel each other out, when the fluctuation functions in (1) or (2) accumulate differences between a center and its neighbors.

The remaining question is which fluctuation function should be used to reconstruct the embedded information, Zhang's in (1), Hong *et al.*'s in (2), or another one.

3.2 Modifying Fluctuation Functions

For a given pixel, we can use at least four adjacent pixels (up, down, left, and right), to evaluate the spatial correlation. Therefore, we can categorize the possible information from the four neighboring pixels, as follows

1. Difference between a pixel and average of four neighbors, $|p_{i,j} - \frac{p_{i-1,j}+p_{i,j-1}+p_{i+1,j}+p_{i,j+1}}{4}|$
2. Difference between a pixel and up, down, left, or right pixel, $|p_{i,j} - p_{i-1,j}|$, $|p_{i,j} - p_{i+1,j}|$, $|p_{i,j} - p_{i,j-1}|$, or $|p_{i,j} - p_{i,j+1}|$
3. Difference between a pixel and average of up-down neighbors or left-right neighbors, $|p_{i,j} - \frac{p_{i-1,j}+p_{i+1,j}}{2}|$ or $|p_{i,j} - \frac{p_{i,j-1}+p_{i,j+1}}{2}|$

Zhang used the information 1) [7] as in (1), and Hong *et al.* proposed the sum of two differences in 2), down and right pixels to fit their side-match scheme as in (2), since the left and right differences (also, the up and down differences) will contribute the same values. For the proposed lattice patterns, we can use every item of information in 1), 2), and 3) at the same time.

Let us briefly analyze the characteristics of the three types of differences, as in the above list. Firstly, note that the first and third differences use the average values around center pixels (possibly inverted for the three least significant bits). However, the second one directly uses the difference between the center pixel and its neighbor. It is easy to see that if at a gradually varied area of the original image, the average values contain proper information for the reconstruction, while at a drastically varied area, the average values can show noisy information for the reconstruction.

The main idea of improvement from the previous schemes starts from the effort to maximize the utilization of the neighbor pixels. In the stage of extracting hidden bits at the previous schemes, the difference between the spatial correlation values of H_0 and H_1 can be calculated from the fluctuation function in (1). Since this difference is not so high, the probability of error to extract hidden bits from the original image is not small. To overcome it, the following fluctuation function is used for the proposed scheme:

$$f = \sum_{i=0}^{s-1}\sum_{j=0}^{s-1}\Bigg[\Big|b_{i,j}-c_{i,j-1}\Big| + \Big|b_{i,j}-c_{i,j+1}\Big| + \Big|b_{i,j}-c_{i+1,j}\Big| + \Big|b_{i,j}-c_{i-1,j}\Big| + \Big|b_{i,j}-\frac{c_{i-1,j}+c_{i+1,j}+c_{i,j-1}+c_{i,j+1}}{4}\Big| + \min\left\{\Big|b_{i,j}-\frac{c_{i-1,j}+c_{i+1,j}}{2}\Big|, \Big|b_{i,j}-\frac{c_{i,j-1}+c_{i,j+1}}{2}\Big|\right\}\Bigg] \tag{5}$$

where B and C are defined in (3) and (4). Note that for $i = 0$ (or $i = s-1$), $c_{-1,j}$ $(0 \le j \le s-1)$ (or $c_{s,j}$) are pixels in the neighboring, left (or right) blocks. Similarly, for $j = 0$ (or $j = s-1$), $c_{i,-1}$ $(0 \le i \le s-1)$ (or $c_{i,s}$) are pixels in the neighboring, up (or down) blocks. Therefore, while in the same block, left and right differences (the first and second terms in (5)) have the same values, in the edges of the blocks, the values can be different, and for this reason, we will use both left and right differences. This is similar to up and down differences.

In addition, note that the pixels in the other blocks are never in the lattice, as depicted in Fig. 8. Therefore, we do not need to carry out the initial calculation, for the difference $A_{u,v}$ between fluctuation values and sorting results of the initial calculation as in Hong *et al.*'s scheme [9].

3.3 Refined Data Hiding Method

The proposed reversible data hiding for encrypted images is based on the lattice pattern as shown in Fig. 5. The block size can be either even or odd. For odd block size s, there are two distinct patterns, which should be alternately used for each consecutive block, so as to ensure that surrounding pixels of the center pixels are always unchanged. In contrast to this, for even size, only one pattern is enough to ensure the surrounding pixels of the center pixels are always unchanged.

The proposed scheme allows only pixels on the specified lattice to be possibly changed, when the data embedding process is carried out. The number of pixels on the lattice can be determined for a given block size s. For odd Pattern 1 or Pattern 2 in Fig. 5, the number of pixels on the lattices is given as $\frac{s^2+1}{2}$, which is an odd number, or $\frac{s^2-1}{2}$, which is an even number, respectively. For an even block size s in Fig. 5, the number of pixels on the lattice is clearly $\frac{s^2}{2}$, which is an even number.

For the data hiding, using the data hiding key, random numbers are generated using a pseudo-random number generator; and based on the random numbers, (almost) half of the pixels on the lattice are selected as Set 0, and the remaining pixels are considered as Set 1. In particular, since the number of pixels on the lattice for odd pattern 1 is odd, we have to select $\frac{s^2-1}{4}$ pixels as Set 0, and the remaining $\frac{s^2+3}{4}$ pixels are considered as Set 1. Similar to Zhang's scheme, depending on the hidden bit, some least significant bits of every pixel in Set 0 or Set 1 are modified.

Here, in contrast to previous schemes, we can modify more than three bits in a small portion of pixels over the lattice, while maintaining the same PSNR, compared to previous schemes. Since a lesser number of pixels is changed, and thus the proposed scheme has a larger PSNR than that of the previous schemes due to the lattice, we have room to insert more information into the encrypted images.

Generally, in order to hide a bit of data, the center pixel is bitwise exclusive-ORed by a pre-specified intensity value I. Otherwise, the center pixel is added by the pre-specified intensity value I, as follows:

$$p'_{i,j} = p_{i,j} \oplus I \tag{6}$$

where, $\oplus$ is the bitwise XOR. Note that if $I = 7$, then it is the same as Zhang's proposition, except that only pixels in the lattice can be changed. Increasing the intensity I will lead to reducing BER, at the cost of reducing PSNR for a given image. In the proposed scheme, we will consider only $I = 7$ or $I = 15$.

3.4 Reconstruction of Hidden Data

In order to reconstruct hidden data, a user should use the same data hiding key to generate the same subsets S_0 and S_1. Then, two hypothesis sets H_0 and H_1 are generated, where for H_i, every pixel in S_i is changed, and the other pixels do not change. Then, using the fluctuation function defined in (5), the fluctuation values f_0 and f_1, which correspond to H_0 and H_1 are calculated, respectively.

If $f_i < f_{1-i}$, then i is considered as the hidden data. For pixels in the four edges of blocks, the side-match technique is also used to evaluate the corresponding fluctuation function. However, note that because of the lattice structure presented in Fig. 8, the initial calculation of fluctuation function and sorting in descending order, as in Hong *et al.*'s scheme is no longer necessary. We can directly use adjacent pixels to evaluate the corresponding fluctuation values, without worrying about the correctness of values.

Fig. 9: Samples images: Lena, Baboon, Sailboat, and Peppers from left.

4 Performance Analysis

4.1 PSNR of the Proposed Scheme

The PSNRs of the proposed scheme are calculated in this subsection. As the pixels to be changed will be influenced by the amount of intensity value I, the energy of average error can be calculated as [7]

$$E = \begin{cases} 21, & \text{if } I = 7 \\ 85, & \text{if } I = 15. \end{cases}$$

In the scheme using the proposed lattice, for odd s, at least $(s^2+3)/4$ or $(s^2-1)/4$ pixels are changed among s^2 pixels. Thus, the PSNR can be calculated, as follows:

$$\begin{aligned} \text{PSNR} &= 10\log_{10}\frac{4\times 255^2 s^2}{\tilde{E}(s^2+3)} \\ &= 54.15\text{dB} + 10\log_{10}\left(\frac{s^2}{s^2+3}\right) - 10\log_{10}\tilde{E} \quad (7) \end{aligned}$$

or

$$\begin{aligned} \text{PSNR} &= 10\log_{10}\frac{4\times 255^2 s^2}{\tilde{E}(s^2-1)} \\ &= 54.15\text{dB} + 10\log_{10}\left(\frac{s^2}{s^2-1}\right) - 10\log_{10}\tilde{E} \quad (8) \end{aligned}$$

where, $\tilde{E}$ is the energy of average error when we use both $I = 7$ and $I = 15$ at the same time, which is given as

$$\tilde{E} = 21\times(1-x) + 85\times x \quad (9)$$

where, x is the portion of pixels over the lattices that use $I = 15$ for data embedding. For even s, $s^2/4$ pixels are changed among s^2 pixels. Thus, the PSNR can be given as

$$\begin{aligned} \text{PSNR} &= 10\log_{10}\frac{4\times 255^2 s^2}{\tilde{E}s^2} \\ &= 54.15\text{dB} - 10\log_{10}\tilde{E}. \quad (10) \end{aligned}$$

To meet the same PSNR 37.9 dB of the previous schemes, we can determine the portion x, the ratio of the number of blocks using $I = 15$ over the total number of blocks in Set 0 or Set 1. For even s, from (10), we have

$$\text{PSNR} = 54.15 - 10\log_{10}\tilde{E} = 37.9$$

Table 1: Portion of pixels to use $I = 15$ to meet overall PSNR 37.9 dB for odd block size s.

s	Pattern 1		Pattern 2	
	x	# of pixels	x	# of pixels
7	0.293	3	0.345	4
9	0.307	6	0.339	6
11	0.315	9	0.336	10
13	0.319	13	0.335	14
15	0.322	18	0.334	18
17	0.324	23	0.333	23
19	0.325	29	0.333	29
21	0.326	35	0.332	36

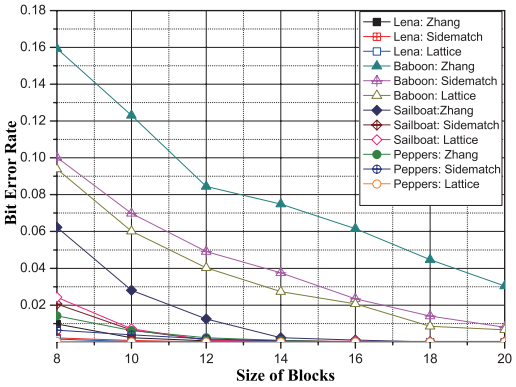

Fig. 10: Comparison of BER performance for all images.

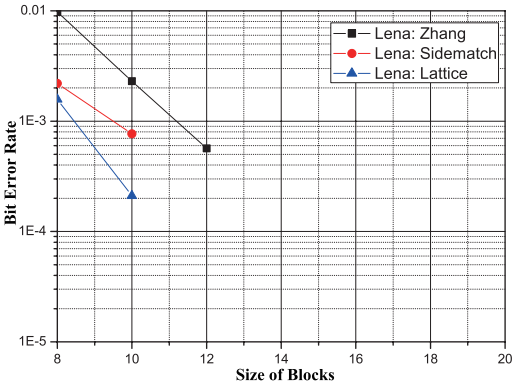

Fig. 11: Comparison of BER performance in log scale for Lena.

and in turn, $\tilde{E} = 42.17 = 64x + 21$. Thus, the portion is given as $x = 0.33$. For odd s, from (7) and (8), depending on s, for Pattern 1, x ranges over 0.293 to 0.326; and for Pattern 2, x ranges over 0.345 to 0.332 from $s = 7$ to $s = 21$, as presented in Table 1. The selection of pixels for $I = 15$ instead of $I = 7$ can be carried out by using the same pseudo-random number generator for separation of Set 0 and Set 1 in the embedding process.

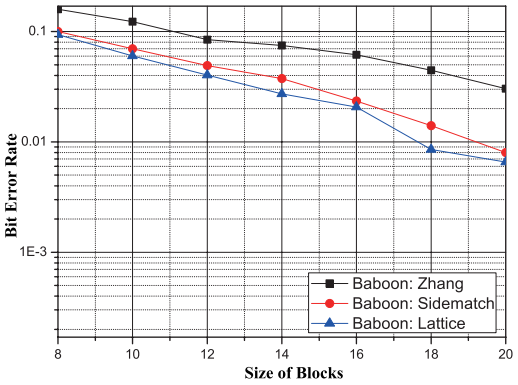

Fig. 12: Comparison of BER performance in log scale for Baboon.

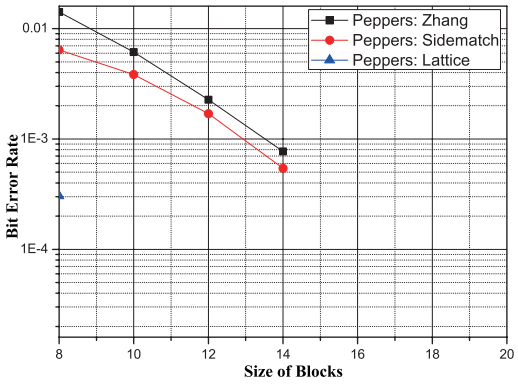

Fig. 14: Comparison of BER performance in log scale for Peppers.

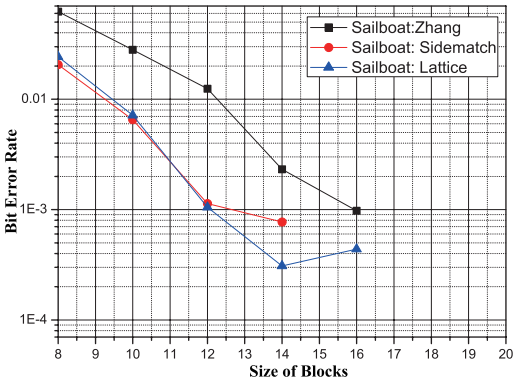

Fig. 13: Comparison of BER performance in log scale for Sailboat.

4.2 Comparison of BER Performance

In order to compare the BER performance of the proposed scheme and previous schemes, we use 4 sample images, as in Fig. 9: Lena, Baboon, Sailboat, and Peppers, whose sizes are 512×512. For fair comparison, we try to keep the same PSNR, 37.9 dB for every scheme. That is, for the proposed schemes, we use both $I = 7$ and $I = 15$. The number of pixels that use $I = 15$ is 33% of $s^2/4$ pixels, and they are randomly selected. The overall results are presented in Fig. 10. Except for Baboon, the BER performance for all the other images converges to zero with a certain length of block size. In order to distinguish their performance, we present four results of BER performance in log scale, as in Figs. 11–14. The results in Figs. 11–14 are obtained by averaging the BERs from 20 distinct data hiding keys. Non-zero BER means that it is not reversible, because error in the reconstruction will destroy the original image, when the embedded data is extracted. Conversely, fast convergence to zero BER means that the block size can be smaller than the others, for the use of reversible data hiding.

Except for Sailboat, the proposed scheme shows better BER performance than the previous schemes by Zhang and Hong *et al.*. For the case of Sailboat, the BER performance is almost matched. However, considering the simplified reconstruction in the proposed scheme, the proposed scheme still has an advantage over the previous scheme.

5 Conclusions

In this paper, we proposed a reversible data hiding scheme, with lower error probability than the previous scheme. In the proposed scheme, the encrypted image can be obtained without any distortion after the hidden data is successfully extracted.

We proposed a new reversible data hiding scheme for encrypted images, which is improved over Zhang's and Hong *et al.*'s schemes in two points: introducing the lattice, and modifying the fluctuation function. Because of the lattice structure, we can simplify the reconstruction of hidden information. That is, in the proposed scheme, the initial calculation and sorting process are not necessary contrary to Hong *et al.*'s scheme. In the results, it is also possible to reduce the error probability in the reconstruction stage, which utilizes the deviation between the spatial correlation values.

Acknowledgements

This research was supported by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2014-H0301-14-1044) supervised by the NIPA(National ICT Industry Promotion Agency) and

supported by the Power Generation and Electricity Delivery of the KETEP grant funded by the Korea government Ministry of Trade, Industry and Energy (20131020400760).

References

[1] Arindam Dey, Rangaballav Pradhan, Anita Pal, Tandra Pal, The Fuzzy Robust Graph Coloring Problem, Advances in Intelligent Systems and Computing, **327**, 805-813 (2015).
[2] T. Filler, J. Judas, J. Fridrich, Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes, Information Forensics and Security, IEEE Transactions on, **6**, 920-935 (2011).
[3] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
[4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.
[5] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst., Video Technol.*, vol. 16, no. 3, pp. 354–362, 2006.
[6] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 1, pp. 187–193, 2010.
[7] X. Zhang, "Reversible data hiding in encrypted image," *IEEE. Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.
[8] X. Zhang, C. Qin, and G. Sun, "Reversible data hiding in encrypted images using pseudorandom sequence modulation," in *Proc. IWDW 2012, LNCS*, vol. 7809, pp. 358–367, 2013.
[9] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Process. Lett.* vol. 19, no. 4, pp. 199–202, Apr. 2012.
[10] W. Hong, T.-S. Chen, J. Chen, Y.-H. Kao, H.-Y. Wu, and M.-C. Wu, "Reversible data embedment for encrypted cartoon images using unbalanced bit flipping," in *Proc. ICSI 2013, LNCS*, vol. 7929, pp. 208–214, 2013.
[11] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reverving room before encryption," *IEEE Trans. Inf. Forensics and Security*, vol. 8, no. 3, pp. 553–562, Mar. 2013
[12] M. S. A. Karim and K. Wong, "Universal data embedding in encrypted domain," *Signal Processing*, vol. 94, pp. 174–182, 2014.
[13] W. Zhang, K. Ma, N. Yu, "Reversibility improved data hiding in encryted images," *Signal Processing*, vol. 94, pp. 118–127, 2014.
[14] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC 2009, Proceeding of the 41st annual ACM symposium on Theory of computing*, 2009, pp. 169–178.
[15] M. van Dijk, G. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proc. EUROCRYPT 2010, LNCS*, vol. 6110, pp. 24–43, 2010.
[16] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Proc. PKC 2010, LNCS*, vol. 6056, pp. 420–443, 2010.
[17] R. Gennaro, G. Gentry, and Bryan Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. CRYPTO 2010, LNCS*, vol. 6223, pp. 465–482, 2010.
[18] X. Zhang, "Separable Reversible Data Hiding in Encrypted Image," *IEEE Trans. Inf. Forensics & Security*, vol. 7, no. 2, pp. 826–832, Nov. 2011.
[19] X. Zhang, Z. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," *J. Visual Commun. Image Representation*, vol. 25, no. 2, pp. 322–328, Feb. 2014.

Young-Sik Kim received B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Seoul National University in 2001, 2003, and 2007, respectively. He joined Semiconductor Division, Samsung Electronics and carried out research and development for secure hardware IPs for various embedded systems, especially for smart-cards until the end of August in 2010. He is an assistant professor at Chosun University, Gwangju, Korea. He is an Editor of the Journal OF Communications and Networks (JCN) from 2013. His research interests include cryptographic engineering and information theory including hardware security, embedded security, physical layer security, data hiding, channel coding, and signal design.

Kyungjun Kang received the B.S and M.S degrees in Information Security from Dongguk University in 2014. He is currently a research engineer at Nicstech, Seoul, Korea. His research interests are in the area of Mobile Security including VMI(Virtual Mobile Architecture), MAM, BYOD.

Dae-Woon Lim received the B.S. and M.S. degrees in department of electrical engineering from KAIST, Daejeon, Korea, in 1994 and 1997, respectively. In 2006, he received the Ph.D. degree in electrical engineering and computer science from Seoul National University. From 1997 to 2002 he was with LG Industrial Systems as a senior research engineer, where he developed recognition algorithm, real-time tracking algorithm, and electric toll collection system. He is currently an associate professor in department of information and communication engineering at Dongguk University, Seoul, Korea. His research interests are in the area of signal processing, wireless communications, cryptography, and security.