Data Hiding Using Sequential Hamming + k with m Overlapped Pixels

Cheonshik Kim[1*], Dongkyoo Shin[1*], Ching-Nung Yang[2], Yi-Cheng Chen[2], Song-Yu Wu[2]
[1] Dept. of Computer Science and Engineering, Sejong University,
Seoul, Republic of Korea
[e-mail: mipsan@paran.com, shindk@sejong.ac.kr]
[2] Dept. of Computer Science and Information Engineering,
National Dong Hwa University, Hualien, Taiwan
[e-mail: cnyang@gms.ndhu.edu.tw, hippo032318@gmail.com,
610621220@gms.ndhu.edu.tw]
*Corresponding author: D. Shin, C. Kim

*Received November 7, 2018; revised January 22, 2019; accepted June 13, 2019;
published December 31, 2019*

Abstract

Recently, Kim et al. introduced the Hamming + k with m overlapped pixels data hiding (Hk_ mDH) based on matrix encoding. The embedding rate (ER) of this method is 0.54, which is better than Hamming code HC $(n, n-\ k)$ and HC $(n, n-\ k)$ +1 DH (H1DH), but not enough. Hamming code data hiding (HDH) is using a covering function $COV(1, n = 2^k - 1, k)$ and H1DH has a better embedding efficiency, when compared with HDH. The demerit of this method is that they do not exploit their space of pixels enough to increase ER. In this paper, we increase ER using sequential Hk_mDH (SHk_mDH) through fully exploiting every pixel in a cover image. In SHk_mDH, a collision maybe happens when the position of two pixels within overlapped two blocks is the same. To solve the collision problem, in this paper, we have devised that the number of modification does not exceed 2 bits even if a collision occurs by using OPAP and LSB. Theoretical estimations of the average mean square error (AMSE) for these schemes demonstrate the advantage of our SHk_mDH scheme. Experimental results show that the proposed method is superior to previous schemes.

Keywords: Data Hiding, Matrix Encoding, Hamming DH (HDH), Hamming+1 DH (H1DH), Hk_mDH

http://doi.org/10.3837/tiis.2019.12.020 **ISSN : 1976-7277**

1. Introduction

Secure communication is often accomplished through cryptography. However, For secret communication, steganography is more suitable than cryptography.The reason is the purpose of steganography is that the existence of hidden data is not detected by the adversary [1]. This technique uses a communication channel to transmit secretly after concealing messages on various media and it is called data hiding (DH) [2]. DH is to embed secret bits by slightly modifying the cover media. In fact, images contain a lot of redundant bits, making them very suitable for delivering secret messages. Meanwhile, techniques for detecting hidden data in images, i.e., steganalysis [3, 4], have also been actively researched. Therefore, most researches on DH have been dedicated on minimizing the damage of an image while allowing sufficient messages to be concealed in images.

The method based on Least Significant Bit (LSB) is easily implemented by increasing or decreasing the values of pixels by only one to match the pixels of the image with the bits of the data. Meanwhile, in the aspect of image quality based on LSB, the optimal pixel adjustment process (OPAP) [5] is a great help for improving peak signal to noise ratio (PSNR) of stego image. Using only LSB, it is impossible to guarantee the quality of a stego image, because it could provoke distortion. Recently, Yang et al. [6] adopted LSB and OPAP with a cover image generated neighbor mean interpolation to improve visual quality in DH.

Matrix encoding is a method to embed or extract the message by using syndrome. Crandall [7] proposed a DH called matrix encoding, Westfeld implemented the F5 algorithm [8] using Hamming code, and Bierbrauer et al. [9] analyzed the possibility of this method. Willems [10] introduced a binary embedding method and a ternary embedding method using Hamming code and Golay code respectively. The F5 algorithm is a method that conceals k bits into $(2^k - 1)$ pixels by using modifying the only one bit. F5 has higher embedding efficiency (EE) (which is the average number of embedded bits per change) [11] than the simple LSB method. Since then, many kinds of research based on matrix encoding have been developed inspired by the F5 [12].

Zhang [13] and Fridrich [14] proposed ternary Hamming code that each secret digit in a $(2n+1)$-ary notational system is carried by n cover pixels by increasing or decreasing only one bit. Mielikainen [15] devised LSB matching revised method to achieve the same payload by modifying fewer pixels to the cover image than a simple LSB-based method. In Mielikainen's scheme, the embedding procedure performs for two cover pixels, x_i and x_i +1, at a time. The value of the i^{th} message bit m_i is equal to the LSB of the i^{th} pixel y_i. The value of the i + 1^{th} message bit m_i + 1 is a function (LSB $(\lfloor y_i /2 \rfloor + y_i + 1)$) of y_i and y_i + 1. Zhang et al. [11] proposed Hamming + 1 DH (H1DH) embedding one more bit than COV $(1, 2^k - 1, k)$ with using one additional pixel. In H1DH, the performance improves by employing the OPAP. Recently, Kim and Yang [16] proposed a method of overlapping three pixels to improve ER with COV $(1, 2^k - 1, k)$, and the ER was increased to about 0.12 compared to HDH.

In [16], the second covering function COV $(1, 2^k - 1, \text{k})$ uses a second LSB. Therefore, it causes a high distortion. Hamming + k (HkDH) [17] is a method to embed 2k in a fully overlapped virtual block (LSB $\oplus$ 2LSB) by using COV $(1, 2^k - 1, k)$, and OPAP and LSB were used for optimization. Therefore, HkDH has achieved high ER.

Afterward, Kim et al. [18] extended HkDH to m overlapped pixels data hiding (Hk_mDH) to embed 2k bits in $(2^{k+1} - m - 2)$ pixels by modifying at most 3 bits. But, if the values of two COV $(1, 2^k - 1, k)$ are the same, a collision problem may occur, which may inevitably cause a mean square error. Finally, the Hk_mDH achieves better embedding rate compared with

H*k*DH. The problem of H*k*_*m*DH is that the number of pixels that need to be excavated to embed many bits still remains. In this paper, we introduce sequential H*k*_*m*DH (SH*k*_*m*DH) to solve the matter of H*k*_*m*DH. SH*k*_*m*DH has a way of improving the performance of the ER without significantly degrading the quality of the cover image.

The rest of this paper is organized as follows. Section 2 concisely reviews Hamming code, HDH, and Kim and Yang's DH, and H*k*_*m*DH. In Section 3, we present SH*k*_*m*DH. In addition, the AMSE of SH*k*_*m*DH is theoretically derived. In Section 4, the proposed SH*k*_*m*DH is tested using some images. A comparison with the HDH, H1DH, Kim and Yang's DH, and H*k*_*m*DH is also provided. In addition, we apply RS steganlysis to SH*k*_*m*DH. Finally, this paper concludes in Section 5.

2. Related Work

2.1 Hamming Code

HC (n, k) codes are a single error correction linear code with d_{min} = 3 and parity check matrix **H** of $k \times n(= 2^k - 1)$, where k is the number of information bits and $(n - k)$ is the number of parity bits. The parity check matrix **H** for k = 3 is

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{1}$$

The n-bit column vectors x_b and y_b are LSB of pixels $x = (x_{1,} \ldots, x_{n-k})$ and $y = (y_{1,} \ldots, y_n)$, where x_b and $y_b$will be denoted $F_2^{n-k}$and F_2^n, respectively.

Let y_b is obtained from an information bits $x_b$via $(n-k) \times n$ generator matrix G, *i.e.*, $y_b = x_b \cdot G$. For any y_b, the vector $S = Hy_b{}^T (\in F_2^k)$ is called the syndrome of y_b. For each syndrome S, the set $\hat{C}(S) = \{y_b \in F_2^n \mid \mathrm{H}y_b = S\}$ is called a coset. Note that $\hat{C}(0) = \hat{C}$. For any syndrome S, let dec(S) be the integer. For any non-zero syndrome S, the vector e (S) = (0,…,1,…,0) with 1 at the dec(S)-*th* place, i.e., $He(S) = S$. To embed δ ($\in F_2^k$) message bits, replace y with $\hat{y} = y_b \oplus e\ (\delta - \mathrm{H}y_b)$. The receiver extracts the message from $\hat{y}$ by Eq. (2).

$$\begin{cases} \hat{y} \cdot H^T = (y_b \oplus e) \cdot H^T = y_b \cdot H^T + \delta - y_b \cdot H^T \\ \qquad\qquad\qquad\qquad\qquad\ = \delta \end{cases} \tag{2}$$

In order to embed δ message bits in each subset by making at most one embedding change, we divide the cover image into $N/n(= 2^k - 1)$ subsets, each consisting of n pixels. Suppose that there is one error bit occurred in $\hat{y}$ (say the 6-th bit from left), i.e., $e = (e_1, \ldots, e_7)$ = (00000**1**0). The syndrome S = (110) is extracted. It means that the error position is the 6-th from left.

2.2 Hamming coding+1 data hiding (H1DH)

For H1DH, k secret bits δ embed to $y_b = (y_{1,} \ldots, y_n)$ of $n\ (= 2^k - 1)$ pixels using $COV(1, n, k)$ by flipping at most one bit. An extra 1-bit embeds at 2^k by using OPAP. That is, first k was embedded by using syndrome $\hat{y} = y_b \oplus e\ (\delta - \mathrm{H}y_b)$. If $(\hat{y} \cdot H^T = \delta)$, there is no needed any action. For embedding δ, one change is needed with probability$(2^k - 1)/2^k$. To embed $k + 1$ bits, they append one pixel like $y = (y_{1,} \ldots, y_n, y_{n+1})$. By $y_i \pm 1$, its $b(y_i)$

becomes the same binary $b(y_i)\oplus 1$, while $2b(y_i)$ can either be 0 or 1 after $b(y_i)\oplus 1$, where $b(\cdot)$ = (y_i mod 2) and $2b(\cdot)$ = ($\lfloor y_i/2 \rfloor$ mod 2) are LSB and 2LSB, respectively.

$$(\delta_1, \dots, \delta_k)^T = H \cdot y_b \tag{3}$$

$$\delta_{k+1} = \left(2b(y_1) + \dots + 2b\left(y_{2^k-1}\right) + b\left(y_{2^k}\right)\right) \bmod 2 \tag{4}$$

There are four embedding optimal rules for H1DH.
(O1): When Eqs. (3) and (4) are satisfied, no action.
(O2): When only Eq. (3) is fulfilled, process $y_{2^k} \pm 1$.
(O3): When only Eq. (4) is satisfied, process $y_i \pm 1$.
(O4): When both of two are not satisfied, process $y_{2^k} \pm 1$.

2.3 Hamming+*k* scheme with *m* overlapped pixels (H*k*_*m*DH)

The H*k*_*m*DH [19] is the extended method of H*k*DH [18]. Suppose that one block is composed of (2^{k+1}–*m*–2) pixels $\left(y_1, \dots, y_{2^{k+1}-m-2}\right)$ with *m* overlapped pixel.

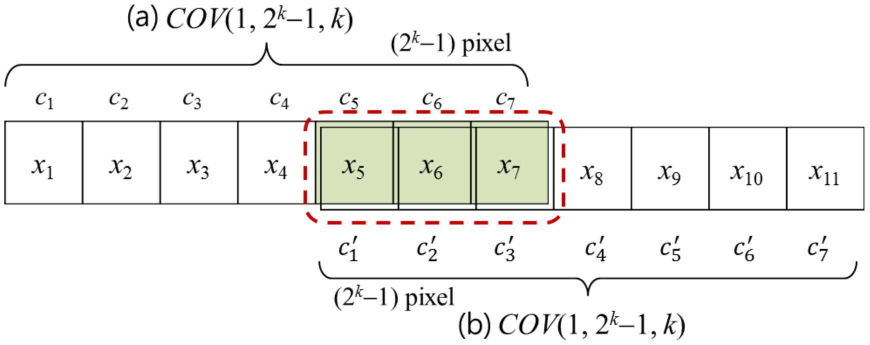


Fig. 1. Block diagram of H*k*_*m*DH with *k*=3 and *m* = 3.

H*k*_*m*DH uses $COV(1, 2^k–1, k)$ to embed *k* secret bits ($\delta_1,\dots,\delta_k$) into the first (2^k–1) pixels, and then embed the other *k* secret bits into the last $(2^k - 1)$ pixels. The first $(2^k - 1)$ pixels are $\left(y_1, \dots, y_{2^k-1}\right)$, and the other pixels are $(y_{2^k-m}, \dots, y_{2^k-m-1})$. As shown in **Fig. 1**, consider the example *k*=3 (i.e., using (7, 4) Hamming code) and *m*=3. There are total 11(=2^{3+1}–3–2) pixels with 3 overlapped pixels. To embed secret bits, we apply Eqs. (5) and (6) to each pixels $(2^k - 1)$.

$$\begin{cases} p = \left(\left(b(y_1), \dots, b\left(y_{2^k-m-1}\right), b\left(y_{2^k-m}\right)\oplus 2b\left(y_{2^k-m}\right), \dots, b(y_{2^k-1})\oplus 2b\left(y_{2^k-1}\right)\right)\right. \\ (\delta_1, \dots, \delta_k)^T = H \cdot p^T \end{cases} \tag{5}$$

$$\begin{cases} q = \left(\left(2b\left(y_{2^k-m}\right), \dots, 2b\left(y_{2^k-1}\right), \left(y_{2^k}\right), \dots, b\left(y_{2^{k+1}-m-2}\right)\right)\right. \\ (\delta_{k+1}, \dots, \delta_{2k})^T = H \cdot q^T \end{cases} \tag{6}$$

We embed *k* secret bits into the first seven bits *p* for Eq. (5), and then embed the other *k* secret bits into the other seven bits *q* of Eq. (6). Except the *m* overlapped pixels, other pixels adopt the LSBs. The overlapped pixels in Eq. (5) (respectively, Eq. (6)), are the XOR-ed results of 1st LSB and 2nd LSB, and 2nd LSB, respectively.

The embedding algorithm is as follows.

The embedding algorithm (H*k*_*m*DH**)**, *m*=3

```
Input: Original image OI
Output: Stego image SI
(Step 1) Read one block from the cover object, and generate a codeword, $p$ by Eq.(5).
(Step 2) $S_1 = H \cdot p^T$ and $S_1' = S_1 \oplus \delta_j^{j+2}, j = 1$.
(Step 3) If $(S_1' \leq (2^k - m)$, $\pm(x_{S_1'})$,
        else if $(S_1' \leq (2^k - 1))$, when $(2b|b(x_{S_1'}) = ([00]\ or\ [10]))$: $y_{S_1'} + 1$ and
                                      when $(b(y_{S_1'}) = ([01]\ or\ [11]))$: $y_{S_1'} - 1$.
(Step 4) Generate another codeword, $q$ by Eq.(6)
(Step 5) $S_2 = H \cdot q^T$ and $S_2' = S_2 \oplus \delta_j^{j+2}$.
(Step 6) If $(S_1' = S_2')$, {$\pm(y_{2^{k+1}-m-2})$, goto (Stegp 5)}
         else if $(S_1' \neq S_2')$, {
              If $S_2' \geq 2^k$, $\pm(x_{S_2'})$,
              else if $(S_2' \geq 2^k - m\ \&\ S_2' \leq 2^k + 1)$,
                  when $(2b|b(x_{S_2'}) = ([00]\ or\ [10]))$: $x_{S_2'} - 1$ and
                  when $(2b|b(x_{S_2'}) = ([01]\ or\ [11]))$: $x_{S_2'} + 1$.
         }
(Step 7) Go to Step 1 until not end of block.
```

3. The Proposed Schemes

3.1 Design concept

All Hamming code based DHs are shown in **Fig. 2**. The ERs for HDH, H1DH, H*k*DH, and H*k*_*m*DH are *k*/*n*, (*k*+1)/*n*, 2*k*/*n*, 2*k*/(2*n*–*m*), respectively (see **Figs. 2**(a)~(d)). As shown in Fig. 2(e), the proposed SH*k*_*m*DH is to sequentially use H*k*_*m*DH to embed secret bits. The ER of the proposed SH*k*_*m*DH is derived as Eq. (7).

$$ER = \frac{\alpha \cdot k}{\alpha \cdot (n-m)+m} = \frac{k}{(n-m)+m/\alpha} = \frac{k}{n-m} \text{ (for } \alpha \rightarrow \infty) \tag{7}$$

The value of *k*/(*n*–*m*) is larger than *k*/(2*n*–*m*) of H*k*_*m*DH. For instance, when α = 3, *n* = 7, and *m* = 3, ER = 9/15 = 0.75. In aspect of PSNR, the SH*k*_*m*DH is the same to H*k*_*m*DH. We can reduce the distortion by using OPAP and LSB.

In our proposed block overlapping approach, an image is divided by blocks of sized (1 × *n*) pixels. At this time, every block has 2*m* overlapped pixels except for the first block including *m* overlapped pixels. The main difference between the SH*k*_*m*DH and the H*k*_*m*DH is that we sequentially embed secret bits with (*n* – 2*m*) ≥ 0 (see **Fig. 2**(e)).

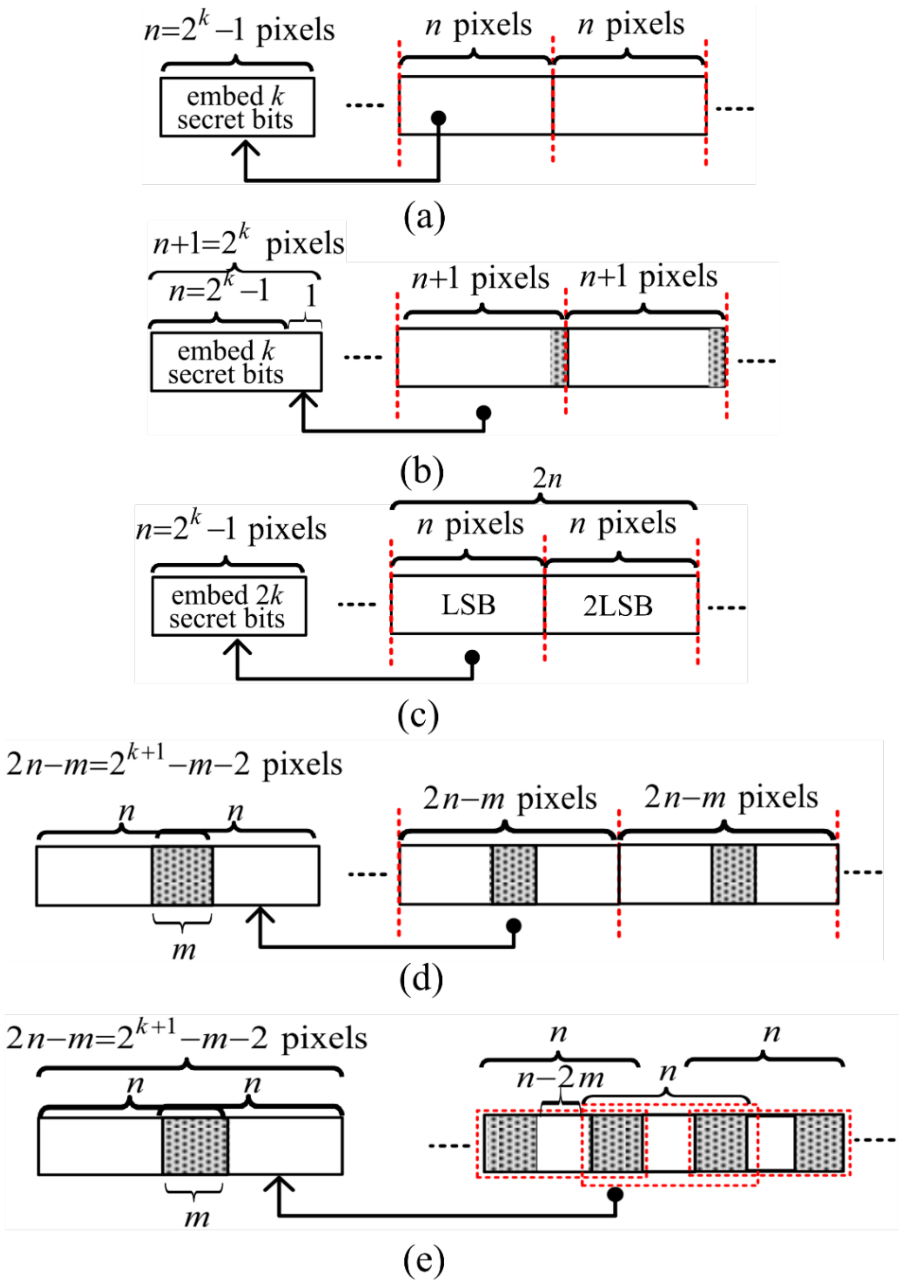


Fig. 2. Diagrammatical representation of Hamming code based DH:
(a) HDH (b) H1DH (c) H*k*DH (d) H*k*_*m*DH (e) the SH*k*_*m*DH.

3.2 Embedding Procedure

For brief explanation, we here assume that $k = 3$ (i.e., by COV (1, 7, 3) covering function) and $m = 3$ for the proposed SH*k*_*m*DH, which can embed 3 secret bits into 7 overlapped cover pixels.

The detailed procedure of the embedding is as follow.

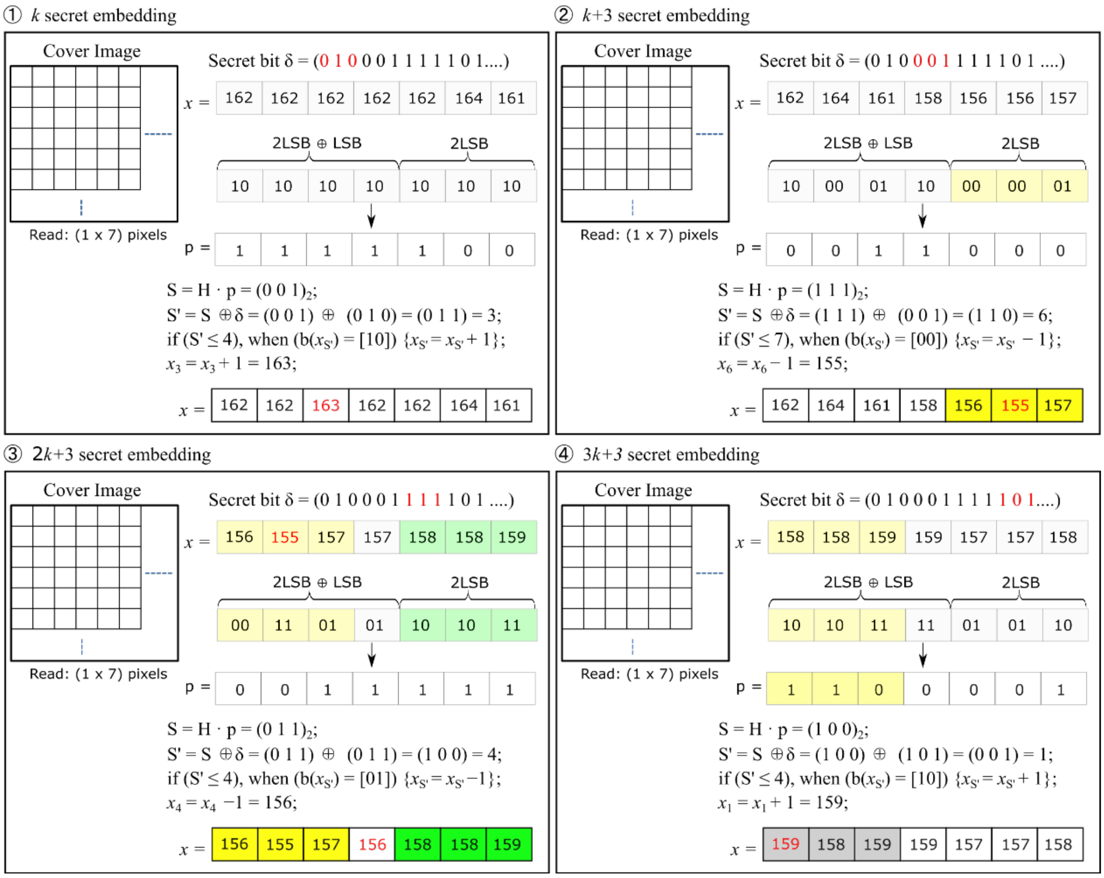
The embedding algorithm (SH*k*_*m*DH)

```
Input: Original image OI
Output: Stego image SI
(Step 1) Read one block, overlapped $(2^k - 1)$ pixels $(x_i^{2^k-1})$, from the cover object and generate a codeword, $p \in F_2^n$, by Eq.(8).
(Step 2) Calculate the syndrome $S = H \bullet p^T$ and $S' = S \oplus (\delta_{j=1}^{j+2})$.
(Step 3) if $S \neq 0$ {if $(S \leq 4)$}{figure out Eq. (9) },
            else if $(S \leq 7)$ { figure out Eq. (10)}.
            Note: 2b|b($\boldsymbol{x}$) means 2LSB and LSB of parameter pixel $x$ in Eqs.(9) and (10).
(Step 4) If not end of the block, $i = i + 4$, $j = j + 3$ and go to Step 1.
```

$$p = \left(b(x_1) \oplus b\left(\left|\frac{x_1}{2}\right|\right), \ldots, b(x_4) \oplus \left(\left|\frac{x_4}{2}\right|\right), b\left(\left|\frac{x_5}{2}\right|\right), \ldots, b\left(\left|\frac{x_7}{2}\right|\right)\right) \tag{8}$$

$$x_{S'} = \begin{cases} x_{S'} + 1 & if\ (2b|b(x_{S'}) = ([00]\ or\ [10])) \\ x_{S'} - 1 & if\ (2b|b(x_{S'}) = ([01]\ or\ [11])) \end{cases} \tag{9}$$

$$x_{S'} = \begin{cases} x_{S'} + 1 & if\ (2b|b(x_{S'}) = ([01]\ or\ [11])) \\ x_{S'} - 1 & if\ (2b|b(x_{S'}) = ([00]\ or\ [10])) \end{cases} \tag{10}$$



① k secret embedding
Cover Image
Secret bit δ = (0 1 0 0 0 1 1 1 1 1 0 1....)
x = 162 162 162 162 162 164 161
2LSB ⊕ LSB
2LSB
10 10 10 10 10 10 10
Read: (1 x 7) pixels
p = 1 1 1 1 1 0 0
S = H · p = (0 0 1)2;
S' = S ⊕δ = (0 0 1) ⊕ (0 1 0) = (0 1 1) = 3;
if (S' ≤ 4), when (b(xS')) = [10]) {xS' = xS' + 1};
x3 = x3 + 1 = 163;
x = 162 162 163 162 162 164 161
② k+3 secret embedding
Cover Image
Secret bit δ = (0 1 0 0 0 1 1 1 1 1 0 1 ....)
x = 162 164 161 158 156 156 157
2LSB ⊕ LSB
2LSB
10 00 01 10 00 00 01
Read: (1 x 7) pixels
p = 0 0 1 1 0 0 0
S = H · p = (1 1 1)2;
S' = S ⊕δ = (1 1 1) ⊕ (0 0 1) = (1 1 0) = 6;
if (S' ≤ 7), when (b(xS')) = [00]) {xS' = xS' − 1};
x6 = x6 − 1 = 155;
x = 162 164 161 158 156 155 157
③ 2k+3 secret embedding
Cover Image
Secret bit δ = (0 1 0 0 0 1 1 1 1 1 0 1 ....)
x = 156 155 157 157 158 158 159
2LSB ⊕ LSB
2LSB
00 11 01 01 10 10 11
Read: (1 x 7) pixels
p = 0 0 1 1 1 1 1
S = H · p = (0 1 1)2;
S' = S ⊕δ = (0 1 1) ⊕ (0 1 1) = (1 0 0) = 4;
if (S' ≤ 4), when (b(xS')) = [01]) {xS' = xS' −1};
x4 = x4 −1 = 156;
x = 156 155 157 156 158 158 159
④ 3k+3 secret embedding
Cover Image
Secret bit δ = (0 1 0 0 0 1 1 1 1 1 0 1....)
x = 158 158 159 159 157 157 158
2LSB ⊕ LSB
2LSB
10 10 11 11 01 01 10
Read: (1 x 7) pixels
p = 1 1 0 0 0 0 1
S = H · p = (1 0 0)2;
S' = S ⊕δ = (1 0 0) ⊕ (1 0 1) = (0 0 1) = 1;
if (S' ≤ 4), when (b(xS')) = [10]) {xS' = xS' + 1};
x1 = x1 + 1 = 159;
x = 159 158 159 159 157 157 158

Fig. 3. Embedding example using SH*k*_*m*DH.

Example 1: We demonstrate the embedding procedure using given x and δ as shown in Fig. 3. For embedding, first, x = (162, 162, 162, 162, 162, 164, 161) and δ = (010) is given.
The codeword p is obtained from a vector x using Eq. (8) and then figure out syndrome S = H·p = (001) using the codeword p. First 4 bits of the codeword is obtained by using 2LSB ⊕ LSB. The other 3-bits are extracted from 2LSB of the x directly. S'(= S ⊕ δ) means the codeword p has new error e = (0010000). If we flip the third bit, then syndrome of the codeword p' is equal to S = (011). However, it is not acceptable to modify a bit in the codeword p for embedding secret bits directly, because the codeword p cannot store any bits. Therefore, we must modify a pixel in the vector x directly according to the rule in Eqs. (9) and (10). In this case, since S' is 3, it is belonging to the rule of Eq. (9). Thus, because $b2|b(x_{S'})$ is equal to "[10]," figure out $x_{S'} = x_{S'} + 1$. That is, x_3 become 163.

For second, x = (162, 164, 161, 158, 156, 156, 157), including three overlapped pixels (162, 164, 161) of former pixel, and secret bits δ = (001) are given.
First, we generate the codeword p from the vector x using Eq. (8) and calculate both S and S' in the same way in the first case. Since S' is less than or equal to 7, it is belong to the rule of Eq. (10). That is, x_6 = 155. For third and fourth cases, we can get the same way like **Fig. 3**(a) and (b).

3.3 Theoretical Estimation of PSNR

The *AMSE* of SHk_mDH is estimated as follows. Consider a codeword C_i of length n ($=2^k-1$), where we could embed k secret bits. Because the proposed SHk_mDH embeds every k secret bits with m overlapped pixels recursively. Therefore, embedding secret bits in the previous codeword C_{i-1} and the next codeword C_{i+1} may modify the pixels in the current codeword C_i.

Originally, when the modification places in overlapped area of are the same, we should modify other two modification places to avoid the collision. There are j modifications, $0 \le j \le 3$, in C_i.
Case (1) j=0:

$$AMSE_0 = \overbrace{\left(\frac{n-m+1}{n+1}\times\frac{1}{n+1}\times\frac{n-m+1}{n+1}\right)\times 0^2 \Big/ n}^{\text{0 modification in } C_{i-1},\, C_i \text{ and } C_{i+1}}.$$

Case (2) j=1:

$$AMSE_1 = \overbrace{\left(\frac{n-m+1}{n+1}\times\frac{n}{n+1}\times\frac{n-m+1}{n+1}\right)\times 1^2 \Big/ n}^{\substack{\text{1 modification in } C_i\text{; 0 modification in ovelapped}\\ \text{area between } C_{i-1} \text{ and } C_i\text{; 0 modification}\\ \text{in ovelapped area between } C_i \text{ and } C_{i+1}}} + \overbrace{\left(\frac{m}{n+1}\times\frac{1}{n+1}\times\frac{n-m+1}{n+1}\right)\times 1^2 \Big/ n}^{\substack{\text{0 modification in } C_i\text{; 1 modification in ovelapped}\\ \text{area between } C_{i-1} \text{ and } C_i\text{; 0 modification in}\\ \text{ovelapped area between } C_i \text{ and } C_{i+1}}} +$$

$$\overbrace{\left(\frac{n-m+1}{n+1}\times\frac{1}{n+1}\times\frac{m}{n+1}\right)\times 1^2 \Big/ n}^{\substack{\text{0 modification in } C_i\text{; 0 modification in ovelapped}\\ \text{area between } C_{i-1} \text{ and } C_i\text{; 1 modification in}\\ \text{ovelapped area between } C_i \text{ and } C_{i+1}}}.$$

Case (3) j=2:

$$AMSE_2 = \overbrace{\left(\frac{m}{n+1}\times\frac{1}{n+1}\times\frac{m}{n+1}\right)\times(1^2+1^2)\Big/n}^{\substack{\text{0 modification in } C_i\text{; 1 modification in ovelapped} \\ \text{area between } C_{i-1} \text{ and } C_i\text{; 1 modification} \\ \text{in ovelapped area between } C_i \text{ and } C_{i+1}}} + \overbrace{\left(\frac{m}{n+1}\times\frac{n}{n+1}\times\frac{n-m+1}{n+1}\right)\times(1^2+1^2)\Big/n}^{\substack{\text{1 modification in } C_i\text{; 1 modification in ovelapped} \\ \text{area between } C_{i-1} \text{ and } C_i\text{; 0 modification in} \\ \text{ovelapped area between } C_i \text{ and } C_{i+1}}}$$

$$+\overbrace{\left(\frac{n-m+1}{n+1}\times\frac{n}{n+1}\times\frac{m}{n+1}\right)\times(1^2+1^2)\Big/n}^{\substack{\text{1 modification in } C_i\text{; 0 modification in ovelapped} \\ \text{area between } C_{i-1} \text{ and } C_i\text{; 1 modification in} \\ \text{ovelapped area between } C_i \text{ and } C_{i+1}}} + \overbrace{2\times\left(\frac{n-m+1}{n+1}\times\frac{1}{n+1}\times\frac{m}{n+1}\right)\times 1^2\Big/n}^{\text{collisions occur in the second term and the third term}}.$$

Case (4) *j*=3:

$$AMSE_3 = \overbrace{\left(\frac{m}{n+1}\times\frac{n}{n+1}\times\frac{m}{n+1}\right)\times(1^2+1^2+1^2)\Big/n}^{\substack{\text{1 modification in } C_i\text{; 1 modification in ovelapped} \\ \text{area between } C_{i-1} \text{ and } C_i\text{; 1 modification} \\ \text{in ovelapped area between } C_i \text{ and } C_{i+1}}} + \overbrace{2\times\left(\frac{m}{n+1}\times\frac{1}{n+1}\times\frac{m}{n+1}\right)\times 1^2\Big/n}^{\substack{\text{collisions occur in the ovelapped area between } C_{i-1} \text{ and } C_i\text{,} \\ \text{and the ovelapped area between } C_i \text{ and } C_{i+1}}} \quad .$$

Finally, the AMSE of SH*k*_*m*DH is derived as follows. For simplicity, we may neglect the cases that the collision in overlapped area to get a simple form of AMSE as $\frac{1}{2^k}+\frac{m/(2^k-1)}{2^{k-1}}$ (see Eq.(11)).

$$\begin{cases} AMSE_{SHk\_m} = \sum_{j=0}^{3} AMSE_j = \overbrace{\frac{2m+n}{n+1}\Big/n}^{\text{no collision}} + \overbrace{\frac{2m\cdot(n-m+2)}{(n+1)^3}\Big/n}^{\text{collisions occur}} \\ \approx \frac{2m+n}{n+1}\Big/n = \frac{2m+2^k-1}{2^k}\Big/(2^k-1) = \frac{1}{2^k}+\frac{m/(2^k-1)}{2^{k-1}}. \end{cases} \tag{11}$$

4. Experiment and Comparison

4.1 Experimental Results

To demonstrate the performance of the proposed method, we compare it with HDH, H1DH, Kim & Yang's DH, and H*k*_*m*DH by using PSNR and EC. In this experiment, we use 512×512 original grayscale images as cover images [19]. The PSNR is an expression for the ratio between the maximum possible value (energy) of a signal and the energy of distorting noise that affects the quality of its representation. The visual quality of digital images is very subjective because of personal bias. This is why we should establish quantitative/empirical measures to compare the effects of image quality. Using the same set of tests images, different DH algorithms can be compared fairly to identify which algorithm has better results. If a stego image can closely resemble the original, then it is a good algorithm. In this aspect, the PSNR may help us to do this.

The value of MSE used to measure average the squared intensity differences between a distorted image and reference image. If an image has a small MSE value, it has a relatively good visual quality. The MSE between two image *x* and *y* is

$$MSE(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i-y_i)^2 \; . \tag{12}$$

The value $e_i = x_i - y_i$ denotes the difference between the original and distorted signal. The MSE is a function for PSNR measure where L is the dynamic range of allowable pixel intensities. For example, for an 8-bit per pixel image, $L = 2^8 - 1 = 255$.

$$PSNR = 10log_{10}\frac{L^2}{MSE} \tag{13}$$

Meanwhile, the EC adopts to compare the performance of the DH through measuring embedding capacity of the cover image. Obviously, most DHs endeavor to increase EC without degrading the quality of stego image. Eq. (14) implies that EC is the ratio of the number of message bits (||δ||)to the total number of pixels.

$$\rho = \frac{||\delta||}{N \times N} \tag{14}$$

Table 1 shows the comparison among HDH, H1DH, Kim and Yang's DH, H*k*_*m*DH, and the proposed SH*k*_*m*DH which are HC based DHs. Each ER of those is 0.43, 0.5, 0.54, 0.54, and 0.75, respectively. The ER of SH*k*_*m*DH is higher than those of other DHs. Moreover, the SH*k*_*m*DH is also better than that of Kim and Yang's DH in the aspect of ER and PSNR. The PSNR of H*k*_*m*DH is slightly higher than that of SH*k*_*m*DH, but ER of the SH*k*_*m*DH is 0.75 and so it is higher than 0.54 of H*k*_*m*DH.

Table 1. Performance comparison of all Hamming-like DHs

Images	HDH		H1DH		Kim and Yang's DH		H*k*_*m*DH		Sequential H*k*_*m*DH	
	PSNR	ρ	PSNR	ρ	PSNR	ρ	PSNR	ρ	PSNR	ρ
Baboon	57.1670	0.43	58.5731	0.5	53.96	0.54	56.0045	0.54	54.2749	0.75
Barbara	57.1598	0.43	58.5851	0.5	53.93	0.54	56.0121	0.54	54.2874	0.75
Boats	57.1644	0.43	58.6066	0.5	53.94	0.54	55.9930	0.54	54.2823	0.75
Goldhill	57.1588	0.43	58.5937	0.5	53.95	0.54	56.0005	0.54	54.2974	0.75
Airplane	57.1497	0.43	58.5755	0.5	53.95	0.54	55.9985	0.54	54.2943	0.75
Lena	57.1620	0.43	58.5772	0.5	53.95	0.54	56.0164	0.54	54.3018	0.75
Peppers	57.1687	0.43	58.5616	0.5	53.95	0.54	56.0063	0.54	54.2832	0.75
Tiffany	57.1780	0.43	58.6239	0.5	53.96	0.54	56.0104	0.54	54.2949	0.75
Zelda	57.1677	0.43	58.5806	0.5	53.99	0.54	56.0112	0.54	54.2902	0.75
Average	57.1640	0.43	58.5863	0.5	53.95	0.54	56.0058	0.54	54.2896	0.75

Table 2. Performance of SH*k*_*m*DH for 1≤*m*≤3.

Images	*m* = 1		*m* = 2		*m* = 3	
	PSNR	ρ	PSNR	ρ	PSNR	ρ
Baboon	56.3327	0.5	55.4245	0.6	54.2749	0.75
Barbara	56.3379	0.5	55.4020	0.6	54.2874	0.75
Boats	56.3391	0.5	55.4134	0.6	54.2823	0.75
Goldhill	56.3486	0.5	55.4075	0.6	54.2974	0.75
Airplane	56.3399	0.5	55.3981	0.6	54.2943	0.75
Lena	56.3422	0.5	55.3820	0.6	54.3018	0.75
Peppers	56.3403	0.5	55.4063	0.6	54.2832	0.75
Tiffany	56.3469	0.5	55.4177	0.6	54.2949	0.75
Zelda	56.3305	0.5	55.3951	0.6	54.2902	0.75
Average	56.3397	0.5	55.4051	0.6	54.2896	0.75

H1DH is superior in quality compared to other DHs, but the proposed SH*k*_*m*DH is the highest among all DHs in the respect of PSNR. Especially, SH*k*_*m*DH and H*k*_*m*DH will be able to determine ER using the parameter *m*. In fact, there is a trade-off between PSNR and ER. We may adjust the value of *m* according to our application.

Table 2 shows the performance of SH*k*_*m*DH based on variable *m*. For 1≤*m*≤3, ERs are 0.5, 0.6, and 0.75, respectively. That is to say, ERs increase as increasing the parameter *m*. Meanwhile, PSNRs decrease. The strength of our SH*k*_*m*DH is that users may embed a large number of bits by increasing the value of *m* with retaining the high PSNR.

Table 3. Performance comparison of proposed scheme and previous schemes (when ER = 0.34)

Images	HDH (*n*=7, *k*=3)	H1DH (*n*=7, *k*=3)	Kim and Yang's DH (*n*=7, *k*=3)	H*k*_*m*DH (*n*=7, *k*=3)	SH*k*_*m*DH (*n*=15, *k*=4)
	PSNR	PSNR	PSNR	PSNR	PSNR
Baboon	58.1147	60.2717	57.8956	57.9809	58.9346
Barbara	58.1150	60.2090	57.8783	57.9931	58.9332
Boats	58.1231	60.2636	57.8873	57.9963	58.9249
Goldhill	58.1094	60.2299	57.8964	58.0172	58.9320
Airplane	58.1130	60.2636	57.8823	58.0117	58.9382
Lena	58.1241	60.2412	57.8967	58.0127	58.9290
Peppers	58.1295	60.2439	57.9062	58.0261	58.9316
Tiffany	58.1819	60.2579	57.9517	58.0209	58.9558
Zelda	58.1186	60.2642	57.8952	58.0008	58.9263
Average	58.1254	60.2494	57.8988	58.0066	58.9339

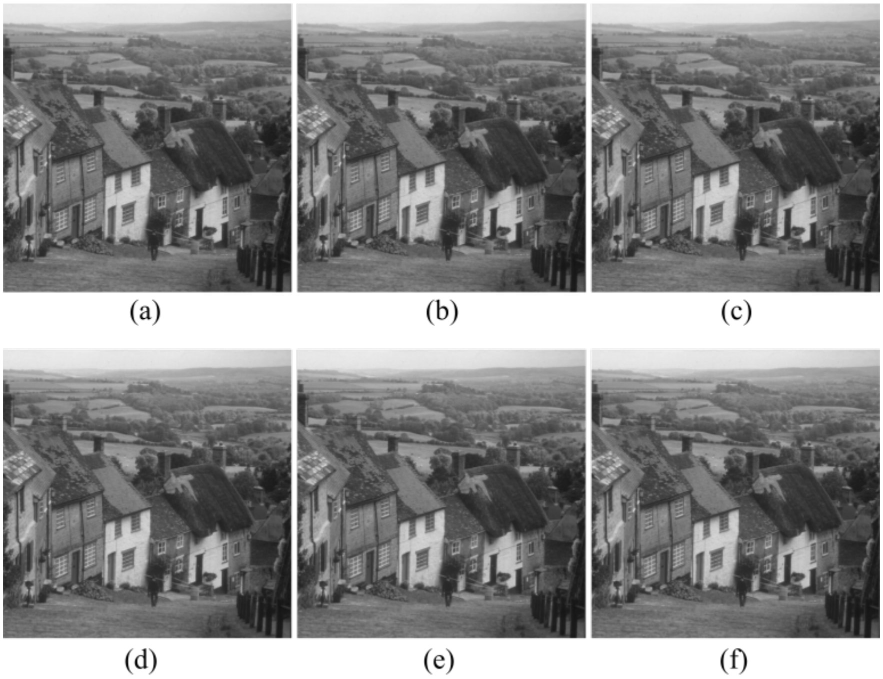


Fig. 4. Comparison of Goldhill images generated from various schemes: (a) original (b) HDH (58.12 dB), (c) H1DH (60.24 dB), (d) Kim and Yang's DH (57.89 dB), (e) H*k*_*m*DH (58.01 dB),and (f) SH*k*_*m*DH (58.92 dB).

Table 3 shows the comparison of PSNR between proposed SH*k*_*m*DH and previous schemes such as HDH, H1DH, Kim & Yang's DH, and H*k*_*m*DH when ER is 0.34. As shown in **Table 3**, even if the same amount of data are hidden in given images, the PSNR of the stego images appears different results depending on the performance of the algorithm. In this experiment, SH*k*_*m*DH (in case of COV (15, 4)) had higher PSNRs than HDH, Kim and Yang's DH, and H*k*_*m*DH, while PSNR of H1DH is 0.13 dB higher than that of SHk_mDH. In fact, SH*k*_*m*DH maximize ER while maintaining PSNR.

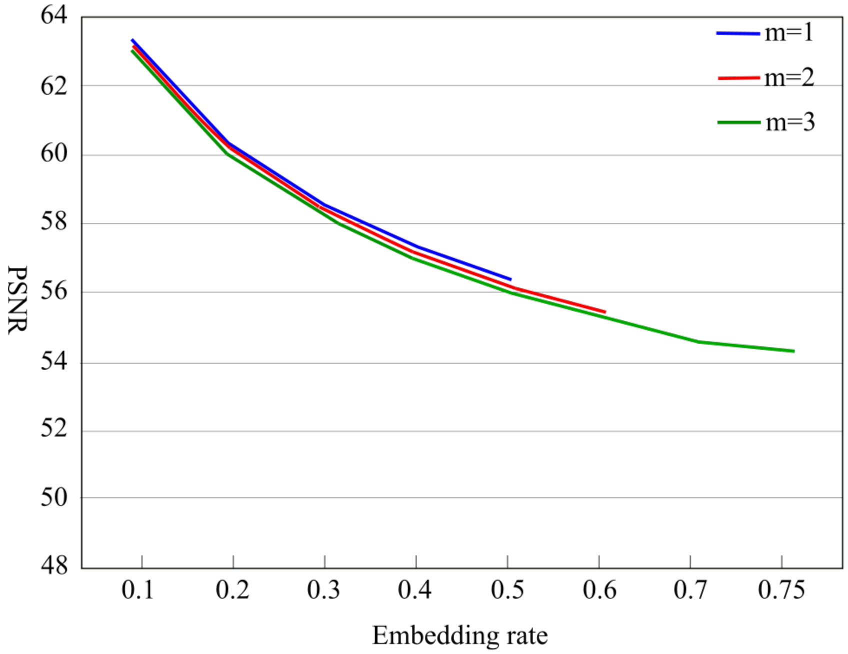


Fig. 5. PSNR trends curves of Lena image (when *m* = (1..3)).

When ER = 0.34, **Fig. 4** represents the comparison among original images and stego images such as (a) original (b) HDH, (c) H1DH, (d) Kim and Yang's DH, (e) H*k*_*m*DH, and (f) SH*k*_*m*DH. The PSNR of H1DH is the highest. The PSNR of SH*k*_*m*DH is higher 0.92 dB than that of H*k*_*m*DH.

As shown in **Fig. 5**, experiments were conducted to verify the performance of the proposed method for 1≤*m*≤3 and the ER was (0.1...0.75). The grayscale Lena image is used for ths experiment. For the same ER, the PSNR of *m*=1 is higher than those of *m*=2 and 3.

Entropy [21] is a statistical measure of randomness that can be used to characterize the texture of the input image. The normalized histogram is an estimate of the underlying probability of pixel intensities, *i.e.*, *N*/*h*(*i*), where *h*(*i*) denotes the histogram entry of intensity value in an image and *N* is the total number of pixel. The entropy of an image is computed as:

$$\mathrm{E} = \sum_i \ h(i)\ log\frac{N}{h(i)} \tag{15}$$

When the relative entropy **E** lying between two probability distribution functions is zero, the system is perfectly secure. In **Table 4**, the entropy of original image (Entropy O), stego image (Entropy S), and the difference between Entropy O and Entropy S are enumerated. It appears that when the number of bits in the secret message increases, the relative entropy in stego image also increases. Because the differential entropy is approximated to zero, it means the proposed scheme endures an attack of steganalysis tools.

Table 4. Comparison of relative entropy between original images and stego images.

Image	ER	Entropy O	Entropy S	Difference
Lena	0.3	7.3938	7.3937	0.0001
	0.4	7.3938	7.3939	0.0001
Baboon	0.3	7.3579	7.3579	0.0000
	0.4	7.3579	7.3582	0.0003
Boats	0.3	7.1237	7.1256	0.0018
	0.4	7.1237	7.1257	0.0020
Goldhill	0.3	7.4777	7.4794	0.0016
	0.4	7.4777	7.4791	0.0013
Tiffany	0.3	6.6055	6.6064	0.0008
	0.4	6.6055	6.6066	0.0010

RS steganalysis [22] was developed with the intent to detect embedded secret messages using LSB replacement. For RS steganalysis, the discrimination function (DF) *f* was used (Eq.(16)), which is to capture the smoothness or “regularity” of the group of pixels.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \tag{16}$$

A pixel groups can be classified as three types: R, S, and U.

- Regular groups: G ∊ R ⇔ *f*(F(G)) > *f*(G).
- Singular groups: G ∊ S ⇔ *f*(F(G)) < *f*(G).
- Regular groups: G ∊ U ⇔ *f*(F(G)) = *f*(G).

where F(G) is ($F_{M(1)}(x_1)$, $F_{M(2)}(x_2)$,…, $F_{M(n)}(x_n)$). Thus, total number of R groups will be increase than the total number of S groups. When the parameters satisfy Eq.(17), it indicates that there is no hidden data in the respective image. When an image has hidden data, R_{-M} and S_M increases, whereas R_M and S_M decrease and detected by RS steganalysis.

$$R_M \approx R_{-M} \text{ and } S_M \approx S_{-M} \tag{17}$$

According to increase ER, the difference between R_M and S_M is to zero. After flipping the LSB of about 43% of pixels, it become $R_M \neq S_{M,}$ where M = [1 1 0 0 1]. As shown in **Fig. 6** (a), it is a few to find the $R_M \approx R_{-M}$ and $S_M \approx S_{-M}$. Meanwhile, in **Fig. 6** (b),(c), and (d), we can see $R_M \approx R_{-M}$ and $S_M \approx S_{-M}$ until about 100%. Therefore, HDH, H*k*_*m*DH, and SH*k*_*m*DH have safety zones avoiding steganalysis detection.

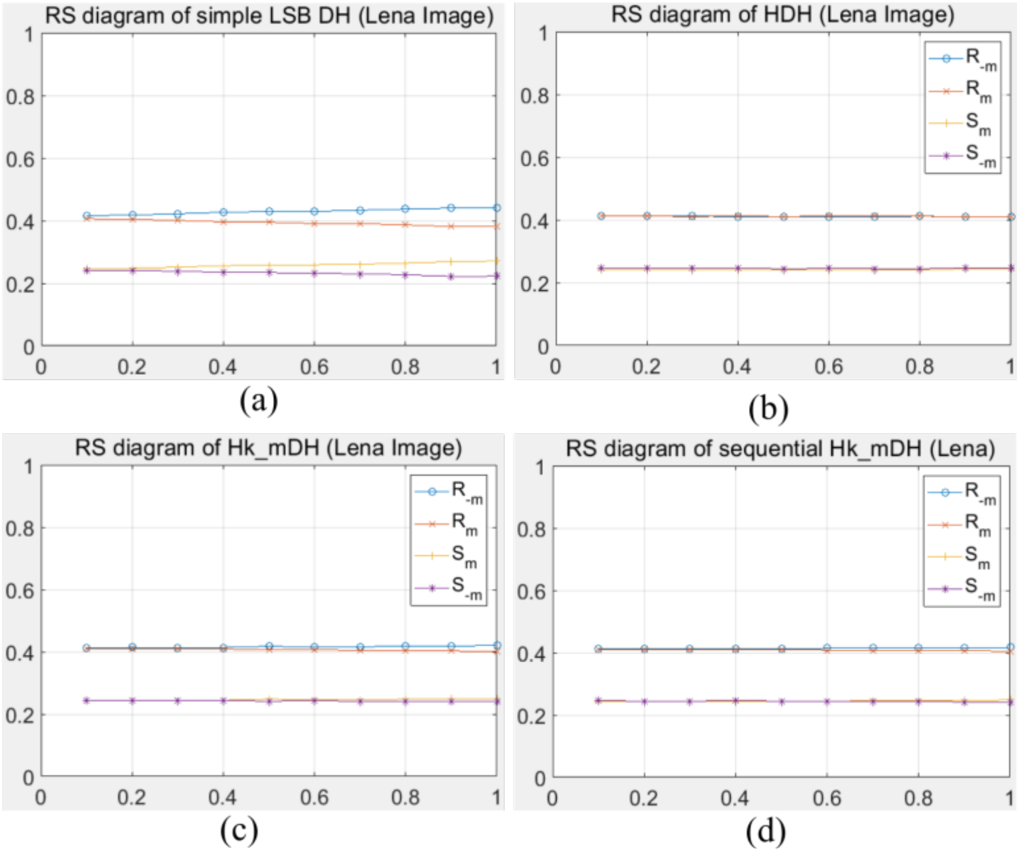


Fig. 6. Comparison of RS diagram between schemes (when ER = 0.43).

5. Conclusion

H*k*_*m*DH is better than previous HC based DHs in the aspect of ER. In this paper, the reason why we proposed SH*k*_*m*DH was that H*k*_*m*DH is not applied sequentially. As a result, could the sequential approach increases the ER to 0.75 when m = 3. In SH*k*_*m*DH, a collision may occur when the same positions in two overlapping blocks need to be corrected at the same time. In this paper, we solve the collision problem using OPAP and LSB. In addition, we demonstrated the superiority of SH*k*_*m*DH via experiments and proofs.

Acknowledgements

This research was supported in part by Ministry of Science and Technology (MOST), under Grant 107-2221-E-259-007-. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1B07047395), National Research Foundation of Korea (NRF) funded by (2015R1D1A1A01059253), and was supported under the framework of international cooperation program managed by NRF (2016K2A9A2A05005255).

References

[1] P. Moulin and R. Koetter, “Data-hiding codes,” in *Proc. of IEEE*, vol. 93, pp. 2083–2126, 2005. Article (CrossRef Link).

[2] W. Bender, D. Gruhl, N. Morimoto, A. Lu, “Techniques for data hiding,” *IBM Systems Journal*, vol. 35, no. 3.4, pp. 313 – 336, 1996. Article (CrossRef Link).

[3] J. Fridrich, M. Goljan, D. Hogea, “Steganalysis of JPEG Images: Breaking the F5 Algorithm,” in *Proc. of Petitcolas F.A.P. (eds) Information Hiding. IH 2002. Lecture Notes in Computer Science*, vol. 2578. Springer, Berlin, Heidelberg, pp.310–323, 2003. Article (CrossRef Link).

[4] Z. Xia, X. Wang, X. Sun, Q. Liu, N. Xiong, “Steganalysis of LSB matching using differences between nonadjacent pixels,” *Multimed. Tools Appl.*, vol. 75, no. 4, pp. 1947–1962, February 2016. Article (CrossRef Link).

[5] C.K. Chan, L.M. Cheng, “Hiding data in images by simple LSB substitution,” *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, March 2004. Article (CrossRef Link).

[6] C.N. Yang, S.C. Hsu, C. Kim, “Improving stego image quality in image interpolation based data hiding,” *Computer Standards & Interfaces*, vol. 50, pp. 209–215, February 2017. Article (CrossRef Link).

[7] R. Crandall, “Some notes on steganography,” 30 November 2018.

[8] A. Westfeld, “F5–A Steganographic Algorithm: High Capacity Despite Better Steganalysis,” in *Proc. of the 4th International Workshop on Information Hiding*, pp. 289–302, 2001. Article (CrossRef Link).

[9] J. Bierbrauer, J. Fridrich, “Constructing good covering codes for applications in steganography,” *Trans. on Data Hiding and Multimedia Security III*, pp. 1–22, 2008. Article (CrossRef Link).

[10] M. van Dijk and F. M. J. Willems, “Embedding information in grayscale images,” in *Proc. of the 22nd Symposium on Information and Communication Theory in the Benelux*, pp. 147–154, (Enschede, The Netherlands), May 2001. Article (CrossRef Link).

[11] W. Zhang, S. Wang, X. Zhang, “Improving embedding efficiency of covering codes for applications in steganography,” *IEEE Commun. Lett.*, vol. 11, pp. 680–682, 2007. Article (CrossRef Link).

[12] R. Zhang, V. Sachnev, B.M. Bakke, H.J. Kim, J. Heo, “An efficient embedder for BCH coding for steganography,” *IEEE T. Inform. Theory*, vol. 58, no. 12, pp. 7272–7279, December 2012. Article (CrossRef Link).

[13] X. Zhang, S.Wang, “Efficient steganographic embedding by exploiting modification direction,” *IEEE Communications Letters*, vol. 10, no. 11, pp. 781–783, 2006. Article (CrossRef Link).

[14] J. Fridrich, P. Lisonˇek, “Grid coloring in steganography,” *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1547–1549, 2007. Article (CrossRef Link).

[15] J. Mielikainen, “LSB matching revisited,” *IEEE Signal Proc. Let.*, vol. 13, pp. 285–287, 2006. Article (CrossRef Link).

[16] W. Zhang, S. Wang, X. Zhang, “Improving embedding efficiency of covering codes for applications in steganography,” *IEEE Communications Letters*, vol. 11, no. 8, 680–682, 2007. Article (CrossRef Link).

[17] C. Kim, C.N. Yang, “Data hiding based on overlapped pixels using Hamming code,” *Multimedia Tools Appl.*, vol. 75, no. 23, pp. 15651–15663, December 2016. Article (CrossRef Link).

[18] C. Kim, D. Shin, C.N. Yang, Y.S. Chou, “Improving capacity of Hamming (n, k)+1 stego-code by using optimized Hamming + k,” *Digital Signal Processing*, vol. 78, pp. 284–293, July 2018. Article (CrossRef Link).

[19] C. Kim, D. Shin, C.N. Yang, Y.S. Chou, “Capacity enhancement of Hamming+k data hiding by pixel overlapping approach,” in *Proc. of the IEEE 17th International Conference on Communication Technology (ICCT)*, pp.1242–1246, 2017. Article (CrossRef Link).

[20] “Image database ref.,” 31 May 2017. Article (CrossRef Link).

[21] A. Rashid, N. Salamat, V.B.S. Prasath, “An Algorithm for Data Hiding in Radiographic Images and ePHI/R Application,” *Technologies*, vol. 6, no. 1, pp.1–12, 2018. Article (CrossRef Link).

[22] J. Fridrich, M. Goljan, R. Du, “Reliable Detection of LSB Steganography in Grayscale and Color Images,” in *Proc. of the ACM Workshop on Multimedia and Security, Ottawa, Canada*, pp. 27-30, October 5, 2001. Article (CrossRef Link).

Cheonshik Kim received his B.S. degree in Computer Engineering from Anyang University, Korea, in 1995; his M.S. degree in Computer Engineering from Hankuk University of Foreign Studies (HUFS), Korea, in 1997; and his Ph.D. degree in Computer Engineering from HUFS in 2003. He works at Department of Computer Engineering, Sejong University, Korea. He won a research award from the IEEK in 2012. He is a member of IEEE. His research interests include Multimedia Systems, Data Hiding, and Watermarking. He wrote about 100 papers including conference papers and journal papers. He was a subject of biographical record in the marquis who's who in the world 2013 - 2019.

Dongkyoo Shin received a B.S. in Computer Science from Seoul National University, Korea, in 1986, an M.S. in Computer Science from Illinois Institute of Technology, Chicago, Illinois, in 1992, and a Ph.D. in Computer Science from Texas A&M University, College Station, Texas, in 1997. He is currently an Associate Professor in the Department of Computer Engineering at Sejong University in Korea. From 1986 to 1991, he worked in Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he worked in the Multimedia Research Institute of Hyundai Electronics Co., Korea as a Principal Researcher. His research interests include Multimedia System, Information Security, Data Mining and Machine Learning.

Ching-Nung Yang obtained his Ph. D. degree in Electrical Engineering from National Cheng Kung University. His B.S. and M.S. degrees, both were awarded in Department of Telecommunication Engineering from National Chiao Tung University. Dr. Yang served in National Dong Hwa University since 1999. His current title is Professor in Department of Computer Science and Information Engineering. He had been Visiting Professor to University of Missouri Kansas City, University of Milan, and University of Tokyo. He is currently a Fellow of IET (IEE) and an IEEE senior member. Professor Yang has done extensive researches on visual cryptography and secret image sharing, and is the chief scientist in both areas. He has authored two books and has published over 250 professional research papers (including more than 110 SCI-indexed journal papers) in the areas of information security and coding theory.

Yi-Cheng Chen is a graduate student at the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. His research includes information security and blockchain technology.

Song-Yu Wu is a graduate student at the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. His research includes multimedia security and error correcting code.