International Journal of Innovative
Computing, Information and Control
Volume **15**, Number **4**, August **2019**

 ISSN 1349-4198
pp. 1565–1580

AN EFFICIENT MATRIX EMBEDDING TECHNIQUE BY USING SUBMATRIX TRANSFORM FOR GRAYSCALE IMAGES

Hsi-Yuan Chang[1,2], Jyun-Jie Wang[3], Chi-Yuan Lin[3,*]
and Chin-Hsing Chen[1,2]

[1]Institute of Computer and Communication Engineering
[2]Department of Electrical Engineering
National Cheng Kung University
No. 1, University Road, Tainan 70101, Taiwan
hi168.hi168@yahoo.com.tw; chench@eembox.ncku.edu.tw

[3]Department of Computer Science and Information Engineering
National Chin-Yi University of Technology
No. 57, Zhongshan Road, Taiping District, Taichung 41170, Taiwan
jjwang@ncut.edu.tw; *Corresponding author: chiyuan@ncut.edu.tw

Received December 2018; revised April 2019

Abstract. *Matrix embedding (ME) is a high performance technique for steganography. Unlike optimal matrix embedding algorithms, which require maximum likelihood (ML) decoding to determine the minimum modified changes, this study proposes an adaptive suboptimal algorithm, called the submatrix transformation matrix embedding (STME) algorithm. The STME algorithm combines the original ME technique with adaptive techniques to improve embedding efficiency and complexity. Several concerns are related to cover location selection, such as the modification of less significant covers, changeable parts of the cover, and forced modification of the cover when embedding a secret message into the cover. The STME algorithm can embed q-ary message vectors at arbitrary specified cover locations. Consequently, the embedded message can be recovered at the receiver, without any damage to the associated cover locations. The simulation results indicate that the STME algorithm offers a trade-off between computational time complexity and embedding efficiency. Moreover, the experimental results show that the STME algorithm has the advantage of adaptive embedding, unlike conventional ME algorithms. The results also show efficiency difference between the optimal ME algorithm and the STME algorithm.*

Keywords: Matrix embedding, Steganography, Maximum likelihood (ML) decoding, Suboptimal algorithm

1. **Introduction.** Researchers have developed numerous embedding techniques in data hiding [1]; this study focused on the steganography technique. Steganography is a crucial measure in the field of secure communication [2]. A steganographic scheme must meet the requirements of statistical undetectability and offer high embedding efficiency [3], which leads to efficient steganographic security. Another problem in steganography is the computational complexity of embedding algorithms. A steganographic scheme requires efficient embedding algorithms [3-6] with high embedding efficiency. Embedding efficiency, which is the average number of embedded bits per one embedded change, is a critical subject in steganography.

DOI: 10.24507/ijicic.15.04.1565

One of the most effective steganographic techniques is matrix embedding (ME) [7,8]. ME with linear block codes, also called syndrome codes [9] or coset codes [10,11] generalized the concept of ME and defined parity check matrix codes as steganographic codes, also called stego codes. [7-11] embed and extract messages by using the parity check matrices of linear block codes. These methods lead to high embedding efficiency because of the characteristics of linear block codes. Finding the coset leader is difficult for a sufficiently large linear code because the complexity of the maximum likelihood (ML) decoding increases exponentially. The present work replaces the ML algorithm with a suboptimal embedding algorithm to improve this disadvantage. Some special cases involve constructive and fast embedding algorithms [3-6]. [3] proposed a highly efficient embedding scheme, namely an ME-based embedding technique for large payloads, and demonstrated using simple codes and random codes. [3] proposed an embedding algorithm based on maximum likelihood (ML) estimation that was capable of embedding substantial amounts of data. By implementing the advantages of ML estimations, the proposed algorithm achieved optimal embedding efficiency. However, the algorithm exhibited disadvantages in the embedding of low data rates, during which the search complexity constantly increased by 2^n. This resulted in substantial increases in the search frequency, thereby rendering the algorithm unusable. [3] resulted in superior steganographic security for large payloads. [3] used structured, simple codes, that is, fast and efficient Hadamard decoding, suitable for large code lengths to produce efficient ME codes and to approach the embedding bounds for large payloads. Furthermore, [4,5] proposed two schemes, called the tree-based parity check (TBPC) and block-overlapping parity check (BOPC), to reduce distortion on a cover object on the basis of some special structures. The TBPC algorithm in [4] exhibits the lowest and second-highest embedding time complexity and embedding efficiency values for these four articles. In [5], the embedding algorithm was used in halftone images. Due to limited changes in halftone images, the embedding algorithm did not exhibit high embedding efficiency, thereby only showing the third-highest embedding efficiency of these four articles. However, the algorithm achieved a 0.5 embedding rate. [6] concealed a large amount of data in a binary image by using a steganographic scheme, which uses a secret key and a weight matrix to increase its security; the weight matrix increased the embedding rate, and an XOR operator decreased the time complexity. The main contribution of [6] was the usage of safe embedding technology, which demonstrated similar embedding efficiency with the algorithm proposed in [5] but only exhibited a 0.125 embedding rate. [12] utilized a majority vote strategy to further improve the computational complexity of TBPC. The properties of the aforementioned four articles can be explained as follows. 1) In algorithm embedding, the computational time complexity increased in accordance with the embedding efficiency, as shown in [3]. 2) Low computational time complexity is attributed to low embedding efficiency, as displayed in [4]. 3) The embedding efficiency and computational time complexity of the embedding algorithm is influenced by the implementations of calculation, as shown in [5,6]. The STME algorithm proposed in this thesis consists of a variable (B), which determines the selection target number of candidate vectors. When the numerical value of B is large, the embedding algorithm demonstrates optimal embedding efficiency, similar to that of the ML algorithm, as shown in [3]. However, this causes excessive computational time complexity. If B is small, then the computational time is low and the embedding efficiency of the embedding algorithm is small. If B is set as a small number, the computational time of the algorithm will be similar to that in [4]. Furthermore, the STME embedding algorithm can conduct embedding in suitable carrier positions according to different carriers, thereby enabling its usage in special occasions such as those stated in [5,6]. [13] used RM codes to embed data in binary host images. The authors propose a novel low-complexity embedding algorithm

that uses a modified majority-logic algorithm to decode RM codes, in which a message-passing algorithm is performed on the highest order of information bits in the RM codes. [14] considers the problem of encoding a finite set of vectors into a small number of bits while approximately retaining information on the angular distances between the vectors. By deriving improved variance bounds, the method gets fast for embedding speed. Binary embedding [15] refers to methods for embedding points in R^d into vertices in the Hamming cube of dimension $O(d)$, such that the normalized Hamming distance between the codes preserves a prespecified distance between vectors in the original space.

This paper proposes submatrix transformation matrix embedding (STME) as an effective and simple method for addressing decoding concerns. STME can replace ML decoding to decrease complexity. The STME algorithm offers a trade-off between complexity and efficiency. Another motivation for using the STME algorithm is the requirement of using an adaptive embedding scheme. For example, STME can process crucial messages that are not permitted to be changed. In the case of a cover in the transmitter, if the secure message is embedded into a specific region of the cover, distortion can be minimized. Consequently, this study proposes an adaptive ME to assign certain blocks that cannot be changed in some specified cover positions. Although an adaptive ME algorithm can be applied in most cases, its use is crucial in a number of cases, including quantization, filtering, lossy compression, dither, and sampling. The integrity of the original cover signal is not the primary concern of this signal processing approach.

Searching for solutions, one of the main concerns in an adaptive matrix embedding algorithm, can be subdivided into two problems. The first problem is that the embedder must solve a system of linear equations over $GF(q)$, and, given a parity check matrix, a problem must be solved using the Gauss elimination method [16,17]. The second problem is the search for an adaptive solution. After solving the first problem, the solution with the least embedding distortion must be located. A search for the toggle vector, that is, the modified vector for the cover object, of a minimum Hamming distance is equivalent to a decoding problem or a binning scheme problem. The STME offers an efficient algorithm with low complexity to manage these two problems simultaneously. Although this adaptive embedding can be achieved, the disadvantage of the STME algorithm is a loss of some embedding efficiency. In addition to improving the embedding efficiency for several applications, cover objects must allow the use of adaptive selection and the nonbinary embedding of messages. The proposed STME algorithm uses q-ary linear block codes to satisfy the requirements for adaptive embedding and nonbinary embedding. Embedding efficiency is also increased using q-ary ME in the grayscale signal domain, that is, ME by using q-ary linear block codes combined with LSB [18] or $\pm\lfloor(q-1)/2\rfloor$ embedding [19,20]. [21] has proposed ternary Hamming and ternary Golay and Hamming covering codes to improve the performance of ± 1 embedding.

The remainder of this paper is organized as follows. Section 2 provides a brief description of related work. Section 3 describes the issues of grayscale signal embedding systems. Section 4 presents the proposed adaptive suboptimal embedding algorithm. Section 5 provides the experimental results and a constructive discussion, with an analysis of the performance levels of various suboptimal algorithms. Finally, Section 6 offers the conclusion.

2. **Related Work.** This section presents a brief discussion on a number of known results of steganography. Least significant bit (LSB) embedding and $\pm\lfloor(q-1)/2\rfloor$ LSB embedding in grayscale covers are two simple embedding schemes. A straightforward method for steganography is LSB embedding, in which the message vectors directly substitute the least significant cover values at the bit level. For example, in 8-bit grayscale images,

each pixel in the image consists of 8 bits according to a weighting from LSB to MSB, consecutively. This method can be detected using statistical analyses. A more efficient steganographic scheme, namely the $\pm\lfloor(q-1)/2\rfloor$ embedding scheme, embeds the message in the grayscale domain instead of in the LSB. We first describe the LSB substitution embedding method. Let $u = (u_1, \ldots, u_n)$ be a cover block, in which $u_i \in \{0, 1, \ldots, 2^{d_p} - 1\}$ is a d_p-bit grayscale cover pixel as follows:

$$u_i = \sum_{j=1}^{d_p} u_{i,j} 2^{j-1} \quad i = 1, 2, \ldots, N, \tag{1}$$

where $u_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,d_p})_2$ represents the base-2 vector and d_p is an arbitrary positive integer. A secret message $s_i = (s_{i,1}, \ldots, s_{i,t})$ of length t must be embedded into the t-bit LSBs of the cover u_i to obtain stego $l_i \in \{0, 1, \ldots, 2^{d_p} - 1\}$, which is a cover in which the secret message has been embedded. The stego pixel l_i with t-bit message in the LSB position is defined as follows:

$$l_i \triangleq LSB(u_i, t) = \sum_{j=1}^{t} s_{i,t} 2^{j-1} + \sum_{j=t+1}^{d_p} u_{i,j} 2^{j-1} = (s_i)_{10} + \sum_{j=t+1}^{d_p} u_{i,j} 2^{j-1}, \tag{2}$$

where $l_i = (s_{i,1}, \ldots, s_{i,t}, u_{i,t+1}, \ldots, u_{i,d_p})_2$ is the representation of base-2 vector. In the extractor, each stego pixel l_i is extracted using

$$s_{i,j} = \left\lfloor \frac{l_i}{2^{j-1}} \right\rfloor \ Mod\ 2, \quad j = 1, 2, \ldots, t. \tag{3}$$

Suppose that the LSB substitution method is used to generate the stego pixel l_i. The payload of this method is precisely t bits of secret message; that is, it is capable of embedding 2^t symbols. In a number of applications, the payload is unnecessary for some requirements; the secret message may express 2^t symbols. LSB substitution is not a convenient or practical method. Moreover, for an $LSB(u_i, 1)$ case, the LSB substitution only changes the LSB $u_{i,1}$ of u_i. For an even u_i case, the pairs of u_i and $u_i + 1$ are only candidates for stego pixel l_i. In particular, the stego pixel l_i is never changed to $u_i - 1$. A number of statistical analyses can detect the LSB method by using this fact. The LSB substitution method should be replaced using the $\pm\lfloor(q-1)/2\rfloor$ embedding scheme. Next, we describe the $\pm\lfloor(q-1)/2\rfloor$ embedding scheme at the grayscale level.

Assume that we use a cover vector $u_{i'} \in F_q$, where $u_{i'}$ can be obtained from a grayscale pixel u_i as $u_{i'} = u_i\ Mod\ q$ to embed a message. For example, in the case of ± 1 embedding, we have three possibilities for each cover, $\{u_{i'} + 0, u_{i'} + 1, u_{i'} + 2\} = \{u_{i'} - 1, u_{i'}, u_{i'} + 1\}$. This means that the cover $u_{i'}$ is unchanged or modified using ± 1. Similarly, we can use the toggle $e_i \in \{0, 1, \ldots, q-1\}$ to obtain the stego $l_i \in \{0, 1, \ldots, 2^{d_p} - 1\}$ as

$$l_i\ Mod\ q = u_{i'} - e_i = u_i\ Mod\ q - e_i\ Mod\ q = u_i - e_i\ Mod\ q. \tag{4}$$

Equation (4) demonstrates that the module operation is distributive over addition; therefore, the change e_i can be added to or subtracted from the cover u_i to obtain the stego in the grayscale domain. For the grayscale level case, the stego pixel can be further represented as follows:

$$l_i = \begin{cases} u_i - e_i + q & \text{if } e_i > \lfloor(q-1)/2\rfloor \\ u_i - e_i & \text{if } e_i \le \lfloor(q-1)/2\rfloor \end{cases}. \tag{5}$$

The calculation of Equation (5) is referred to as the $\lfloor(q-1)/2\rfloor$ embedding scheme, which has a maximum embedding distortion of up to $\lfloor(q-1)/2\rfloor$. Finally, the $\lfloor(q-1)/2\rfloor$ embedding scheme offers superior performance compared to the LSB substitution scheme;

Table 1 shows the embedding efficiency of LSB substitutions and $\pm\lfloor(q-1)/2\rfloor$ embedding schemes, where R_m required to represent the embedding bits per each block is from 0.1 to 0.6. For the approximate embedding rate R_m, the $\pm\lfloor(q-1)/2\rfloor$ embedding schemes have lower embedding distortion at the grayscale level, that is, MSE in Euclidean distance, than LSB substitution has. Although the embedding efficiency can be improved using $\pm\lfloor(q-1)/2\rfloor$ embedding schemes, a more efficient strategy that combines the $\pm\lfloor(q-1)/2\rfloor$ embedding schemes with ME is proposed in Section 3.

TABLE 1. Comparing the performance of LSB substitution and $\pm\lfloor(q-1)/2\rfloor$ embedding algorithms for embedding rate values from 0.1 to 0.6

Embedding scheme	R_m in bpp	D in MSE	η
LSB(1)	1	0.5037	1.9853
LSB(2)	2	2.4783	0.8070
LSB(3)	3	10.3563	0.2896
LSB(4)	4	42.3443	0.0945
LSB(5)	5	170.4	0.0293
±1	1.5849	0.661	2.3977
±2	2.3219	1.9987	1.1617
±3	2.8073	4.0458	0.6939
±4	3.1699	6.6503	0.4766
±5	3.4594	10.0813	0.3432
±6	3.7004	14.0244	0.2639
±7	3.9069	18.6609	0.2093
±8	4.087	24.0394	0.17
±9	4.2479	29.8337	0.1424
±10	4.3923	36.5508	0.1202
±11	4.5236	43.9752	0.1029
±12	4.6439	52.1448	0.089
±13	4.7549	60.2206	0.0789
±14	4.858	71.6853	0.0677
±15	4.9542	77.3044	0.064

3. $\pm\lfloor(q-1)/2\rfloor$ Embedding Scheme Combined with q-ary Matrix Embedding.

This section presents a description of a q-ary embedding scheme that modifies grayscale to enhance embedding efficiency. This section discusses the efficiency bound for ME by using q-ary linear block codes. Furthermore, this section discusses adaptive issue and a merging of ME technique and $\pm\lfloor(q-1)/2\rfloor$ embedding in grayscale.

3.1. Matrix embedding by using q-ary linear block codes.

This subsection discusses a matrix embedding scheme that uses linear block codes and defines some bounds of embedding efficiency for that ME. In the matrix embedding problem, the embedder embeds a message vector from a particular syndrome with regard to linear block codes in an arbitrary cover vector to generate a stego and transmits that stego to the receiver. In the receiver, the message vector is extracted using a parity check matrix. Next, we discuss the concepts of q-ary ME. For a matrix embedding scheme, a q-ary (n,k) linear block code C can be specified as the null space of a given parity check matrix $H \in \{0,1,\ldots,q-1\}^{m\times n}$, where $m=n-k$, as

$$C=\left\{r|Hr^T=0, r\in F_q^n\right\}, \tag{6}$$

where the code C is of q^k codewords. The payload of the code C is defined as the number $|M| = q^m$ of messages. A more convenient measure in a matrix embedding scheme is the embedding rate

$$R_m = \frac{\log_2 q^m}{n} \text{ bits/symbol.} \tag{7}$$

Based on Equation (6), the syndrome $s \in F_q^m$ of the vector r, in the case of a nonzero Hu^T, is defined as $s = Hr^T$. Furthermore, the set composed of all the vectors r, corresponding to the identical s, is referred to as the coset of the code C, defined as $C^s = \left\{r|Hr^T = s\right\} = \{c + e|c \in C\}$, where e denotes the coset leader with the minimum Hamming weight. The minimum error quantizer, over a q-ary symmetric source channel $r = c + e$, quantizes the r to the nearest codeword $c \in C$, and the quantization error, e, is defined as follows:

$$e \triangleq f\left(Hr^T\right), \tag{8}$$

where $f(\cdot)$ is a syndrome decoding function. We also define the set of the coset leader as $E_0 = \left\{e_i|i = 1, \ldots, q^{(n-k)}\right\}$, which consists of all the coset leader e_i for each coset. Given an n tuple source vector r, the average distortion for a vector is defined as follows:

$$d = \frac{E[d(c, r)]}{n} = \frac{D}{n} \text{ changes/symbol,} \tag{9}$$

where $E[\cdot]$ represents the expected value and c represents a nearest quantized codeword existing in the code C. For a good q-ary (n, k) linear embedding code, the equation is approximately defined as follows:

$$R_m \approx h_q(\delta), \tag{10}$$

where $h_q(\delta) = \delta \log_2(1/\delta) + (1 - \delta) \log_2(1/(1 - \delta)) + \delta \log_2(q - 1)$ denotes a q-ary entropy function, that is, the optimal embedding rate R_m at the low bound δ of distortion. For the code C, the minimum average distortion is up to

$$\delta = h^{-1}(R_m) = h_q^{-1}((\log_2 q^m)/n), \tag{11}$$

where $h_q^{-1}(\cdot)$ is the q-ary inverse entropy function. Equation (11), also referred to as the rate-distortion function for an embedding scheme, uses q-ary linear block codes. The low bound δ of the average distortion for each bit in a code block is $\delta \leq d$, where $d = D/n$. When performing q-ary embedding to a cover vector, the embedding efficiency is defined as follows:

$$\eta = \frac{R_m}{d} = \frac{\log_2 q^m}{D} \text{ bits/changes.} \tag{12}$$

Using Equations (11) and (12) obtains the asymptotic upper bound, as follows:

$$\eta_\delta = \frac{\log_2 q^m}{n\delta} = \frac{R_m}{h^{-1}(R_m)}. \tag{13}$$

Next, we discuss the bounding interval of the decoding influence. The optimal algorithm has the maximal probability to decode. The suboptimal algorithm has a lower probability to decode and is proposed in Section 4. For the code C, the embedding efficiency between the optimal and suboptimal algorithms can be expressed as follows:

$$\frac{\log_2 q^m}{nh^{-1}(R_m)} \geq \frac{\log_2 q^m}{D_{opt}} \geq \frac{\log_2 q^m}{D_{sub}}, \tag{14}$$

where D_{opt} and D_{sub} represent the average distortion estimated for each block in the optimal decoding and the suboptimal decoding, respectively. Equation (14) can be expressed in an alternative form as $\eta_\delta \geq \eta_{opt} \geq \eta_{sub}$. Thus, as the measure of efficiency, the interval measure parameters are defined as follows:

$$\varepsilon_{sub} = \eta_{opt} - \eta_{sub}. \tag{15}$$

For an efficient suboptimal embedding code, the value ε_{sub} should be as low as possible. Assume that the distortion of an adaptive embedding is d_{apt} for an n tuple cover vector u; the adaptive embedding technique exhibits an inferior distortion relative to the original embedding version. This occurs because the changeable part contains a part of the subset of u. The average distortion $D_{apt} = nd_{apt}$ of each block for an adaptive embedding is larger than that of the original embedding version. However, the embedding algorithm achieves optimal embedding distortion, D_{opt}, or suboptimal embedding distortion, D_{sub}.

3.2. **An optimal solution for adaptive matrix embedding.** We generalize the matrix embedding problems discussed in this subsection as follows. A matrix embedding scheme can use the q-ary (n, k) linear block code C to achieve the quantization problem. An effective embedding scheme can approach the rate-distortion bound on ME for any chosen embedding rate R_m. This section proposes a solution for a matrix embedding scheme and demonstrates that this solution can be classified as an optimal solution or as an adaptive solution.

It was assumed that the embedding scheme entailed embedding m message symbols to the n tuple cover vectors u' as stego vectors $l' \in C^l$ by using q-ary (n, k) linear block codes. The n tuple toggle vector $x = u' - l'$, which is to be modified according to the positions of nonzero values, is the distance between vector u' and stego vector l'. The process must determine the minimum weight of toggle vector $x \in C^x$. Suppose that the toggle vector x exists and that it is searched with the minimal weight, that is, $x = e_{opt}$ as

$$e_{opt} = \arg\min_{l \in C^l} d(u', l'). \tag{16}$$

In other words, the cover vector u' and the stego vector l' are of a minimal weight vector e_{opt}, that is, the coset leader in C^x. From the decoding viewpoint, given a cover vector u' and a message $s_{l'}$, the coset leader e_{opt} can be discovered through a quantization error function, expressed as follows:

$$e_{opt} = f_{opt}\left(Hu'^T - s_{l'}\right) = f_{opt}(s_{u'} - s_{l'}) = f_{opt}(s_x). \tag{17}$$

We also rewrite Equation (17) with regard to H and in terms of a minimum weight vector x, that is, the coset leader e_{opt}, as

$$s_x = Hx^T. \tag{18}$$

Once discovered, the coset leader e_{opt} is subtracted from the cover vector u' as $l' = u' - e_{opt}$. Essentially, l' is the stego vector closest to the cover vector u' and contains the message vector $s_{l'}$. The procedure of determining the coset leader e_{opt} in the toggle coset C^x is the so-called optimal solution for ME. Because of the constraint imposed on the location selection to embed the message $s_{l'}$ in an adaptive embedding algorithm, the optimal solution may not be the minimum weight vector $x = e_{opt}$ in the coset C^x, whereas the process uses ML decoding to locate the intended toggle vector. Given that a toggle vector $x = (x_1, \ldots, x_n)$, the index set $S \subseteq \{1, 2, \ldots, n\}$. In addition, if the changeable cover locations in u' are confined to u'_i, where $i \in S$, then e_{opt} is no longer the optimal modification vector but is instead defined as follows:

$$e_{apt} = \arg\min_{x \in c^x : Hx^T = s_{l'}} w_H(x), \tag{19}$$

where $\{x_i = 0 | i \notin S\}$. The determination of e_{apt} is dependent on the selection locations of S, that is, e_{apt} may not exist. In other words, a search is conducted within the confined region C^x for the intended toggle vector x. Once discovered, the coset leader $e_{apt} \in C^x$ is subtracted from the cover u' as $l_{apt} = u' - e_{apt}$. Essentially, $l_{apt} \in C^l$ is the vector closest to the vector u' within the F_q^n dimensional space under the location constraint and contains

the message vector s_l. Once e_{apt} is known, the adaptive optimal embedding sequence l' can be discovered. It is difficult to determine e_{apt} in the case of a q-ary (n, k) linear block code C with sufficiently large length because the complexity of the ML decoding increases as q^k. To resolve this disadvantage, another adaptive suboptimal embedding algorithm with low complexity is proposed in Section 4 to replace the optimal algorithm for adaptive embedding.

3.3. **Combination matrix embedding with $\pm\lfloor(q-1)/2\rfloor$ embedding.** As stated in the previous subsection, a linear block code can obtain the adaptive stego block $l' = u' - e_{apt}$, in which a number of tuples may be modified. We only need to change components in a block slightly to apply them to the grayscale signal. For an adaptive toggle block, $e_{apt} = (e_1, \ldots, e_n)$, each component $u_{i'} \in F_q$ in the cover block u' is of the embedding distortion up to $\{0, \ldots, q-1\}$. However, to decrease the embedding distortion, we may embed the message at the grayscale level instead of in the F_q domain. To decrease the embedding distortion, we use the $\pm\lfloor(q-1)/2\rfloor$ embedding scheme, which is of the maximum distortion $\lfloor(q-1)/2\rfloor$, to embed the message vectors $s_{l'}$. For a grayscale cover block, $u = (u_1, \ldots, u_n)$, assume that $e_{apt} \in F_q^n$ is its changeable minimum weight vector, which is used to embed the message vector $s_l \in F_q^m$. We form the stego block as follows:

$$l' = u' - e_{apt} = (u'_1, \ldots, u'_i, \ldots, u'_n) - (e_1, \ldots, e_i, \ldots, e_n) = (l'_1, \ldots, l'_i, \ldots, l'_n), \tag{20}$$

where $(\cdot)_{10}$ denotes a decimal system and the toggle vector is $e_i \in \{-\lfloor(q-1)/2\rfloor, \ldots, \lfloor(q-1)/2\rfloor\}$. The choice of adding or subtracting the amount of e_i may be used to change the cover vectors u_i and obtain the grayscale stego block $l_{apt} = ((l'_1)_{10}, \ldots, (l'_n)_{10})$; therefore, the maximum embedding distortion of stego vector $(l_i)_{10}$ is $|\lfloor(q-1)/2\rfloor|$. In other words, the maximum embedding distortion is the amount that e_i can change l'_i to the grayscale level domain by modifying the u_i. At the receiver, the stego block l_{apt} is mapped from the grayscale level domain into the finite field domain F_q^n by using module operations. Finally, the secret message $s_{l'}$ is extracted using l_{apt} and H as

$$s_{l'} = H(l_{apt} \ Mod \ q)^T = H(u \ Mod \ q - e_{apt} \ Mod \ q)^T = H(u' - (u' - l'))^T. \tag{21}$$

4. **Adaptive Suboptimal Embedding Algorithm.** This section presents a discussion on an adaptive suboptimal algorithm called the STME algorithm, which has two advantages: decoding complexity is decreased and an unchangeable specific cover can be maintained. In other words, the secret logo can be embedded by changing the specified locations of cover (i.e., adaptive embedding).

The proposed algorithm obtains a toggle vector by using a different method than that of conventional ME, which uses the ML decoding algorithm. The proposed algorithm uses the geometric interpretation of a suboptimal embedding algorithm to search for a toggle vector with low weight (Figure 1). This easily obtains the suboptimal toggle vector e_{sub}, that is, the so-called STME algorithm. This suboptimal algorithm locates a suboptimal toggle vector, e_{sub}, where $w(e_{sub}) \geq w(e_{opt})$, and $w(\cdot)$ denotes the Hamming weight, within the toggle coset C^x. However, $w(e_{sub})$ must be as close to $w(e_{opt})$ as possible. Finally, the stego vector l', obtained by subtracting e_{sub} from the cover vector u, and e_{sub} cannot be ensured as the optimal toggle vectors. Generally, optimal algorithms have optimal decoding performance, but they can be implemented only with great difficulty and may be unrealizable because of their high complexity. The STME algorithm is not only a suboptimal embedding algorithm but also an adaptive embedding algorithm; it is illustrated using an example from $((q^m-1)/(q-1), (q^m-1)/(q-1)-m, 3)$ Hamming codes. Any q-ary $((q^m-1)/(q-1), (q^m-1)/(q-1)-m, 3)$ Hamming code of length $(q^m-1)/(q-1)$ can be specified as the null space of a given parity check matrix H

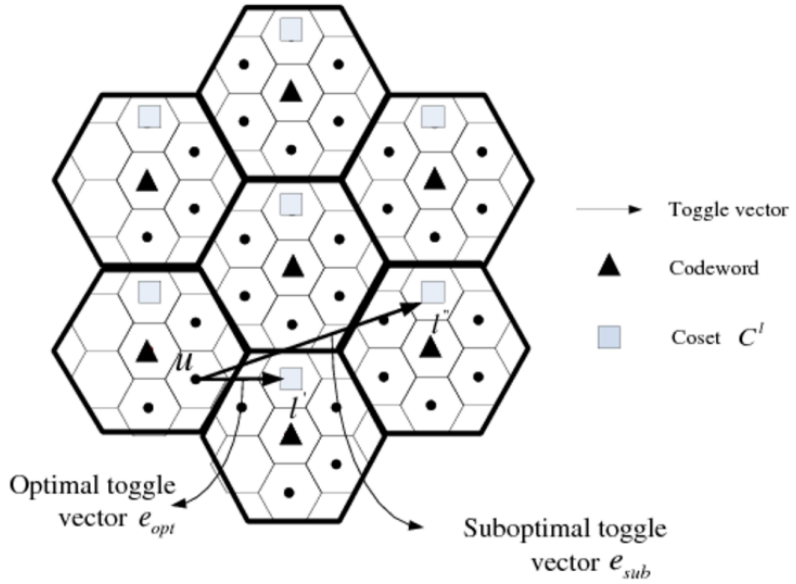


FIGURE 1. For the cover vector u, the closest stego vector l' is found using message symdrome $s_{l'}$.

for embedding a secret message $s_{l'}$ with m q-ary symbols. Consider a cover u' of length $(q^m - 1)/(q - 1)$. Using H to estimate the cover syndrome $s_{u'} = H(u')^T$, a Hamming code results in the difference $s_x = s_{u'} - s_{l'}$, called the toggle syndrome, between cover syndrome $s_{u'}$ and the secret message, syndrome $s_{l'}$. The toggle vector x corresponding to the toggle syndrome s_x obtained using parity check matrix H is a modified vector for the cover object. However, the modified vector x, that is, the toggle vector, is not unique. We may select arbitrary vector x within coset C^x as the toggle vector. In other words, the stego vector l' can be expressed as $l' = u' - f(s_x)$ in the vector domain, where $f(\cdot)$ denotes a decoding function. When $s_{l'} = s_{u'}$, the stego l' (obtained by changing only one location of cover u') is the same as the secret logo. At the receiver, the secret logo is extracted by evaluating $s_{l'} = H(l')^T$. In an adaptive embedding case, given a set $S \subseteq \{1, 2, \ldots, n\}$ as the changeable location index, the equation $H(u')^T - s_{l'}$ is solved for a legitimate toggle vector $x = (x_1, \ldots, x_n)$, where $x_i = 0$ and $i \notin S$, by using the Gaussian elimination method and searching for the solution with minimum distortion. Consider a cover $u' = (0, 2, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0)$, a secret logo $s_{l'} = (1, 1, 1)$, and the ternary $(13, 10)$ Hamming code with parity check

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{bmatrix} = [\theta_1\ \theta_2\ \cdots\ \theta_{13}], \tag{22}$$

where θ_i denotes the column vector of H. We assume that the changeable cover set is $S = \{1, 2, 4, 8, 10, 13\}$ and the other is an unchangeable cover index. The embedding algorithm uses the parity check matrix H to embed three symbols of the secret logo message $s_{l'} = (1, 1, 1)$ into the cover u' with length $n = 13$. In the case of a conventional ME algorithm, the syndrome of u' is $s_{u'} = H(u')^T = (0, 2, 0)^T$, and the difference between $s_{u'} = (0, 2, 0)^T$ and $s_{l'} = (1, 1, 1)^T$ is $s_x = s_{u'} - s_{l'} = (2, 1, 2)^T$. The equation $Hx^T = (2, 1, 2)^T$ is solved for a least weighting vector x. If all the bits within the cover are permitted to be changed, that is, $\{x_i = 1, i \in S\}$, x is subsequently discovered as $x = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0)$, and the 12th symbol is modified for the purpose of embedding $s_{l'}$. Essentially, the operation is tantamount to subtracting the 12th column vector of H and multiplying a scale $2 \in F_3$

by $s_{u'}$. This example of ME can be realized using $(13, 10)$ Hamming code. We can embed three secret logo symbols in a block of 13 pixels by performing at most one embedding change. Thus, the embedding efficiency is $\eta = (3\log_2 3)/(1-3^{-3}) = 4.9378$. In fact, the changeable location set within $\{u'_i | i \in S = \{1, 2, 4, 8, 10, 13\}\}$, that is, the 3rd, 5th, 6th, 7th, 9th, 11th, and 12th covers are those that are not allowed to be changed. In other words, the set of vectors $x = (x_1, x_2, \ldots, x_{13})$, where $\{x_i = 0 | i = 3, 5, 6, 7, 9, 11, 12\}$, is the only legitimate set of toggle vectors. These toggle vectors are necessary to determine if there is a linear combination between the column vectors 1, 2, 4, 8, 10, and 13 of H to form the toggle syndrome $s_x = (2, 1, 2)^T$, the difference between $s_{u'} = (0, 2, 0)^T$ and $s_{l'} = (1, 1, 1)^T$. Observed by using these legitimate column vectors of H, one of the solutions of $Hx^T = (2, 1, 2)^T$ is $x = (2, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ because the linear combination of 1, 2, and 4 of H is $s^T = 2\theta_1 + \theta_2 + 2\theta_4 = (2, 1, 2)^T$. This toggle vector is one of the toggle sets; it shows a way to embed the secret logo $s_{l'}$ in the cover u'. Although we can find another legitimate toggle vector in these changeable locations, the result raises the embedding change from 1 to 3. However, in most cases, it is unlikely to yield a scale column vector of H as a linear combination of others in the same manner. Consider n symbols of the cover u': an index set S corresponds to the column vectors, for selection of an $m \times n$ matrix H. In the case of $|S| \geq m$, the set $\theta = \{\theta_i | i \in S_\theta\}$, where S_θ of size m denotes the subset of S (composed of m linearly independent (LI) column vectors within S), the toggle vector required is a linear combination of the LI set θ, with m symbols of coefficient λ. Subsequently, this study proposes an effective approach for satisfying the requirement of the adaptive ME algorithm. During data embedding, the equation $Hx^T = s_x$ is derived to search for near optimal solution. The equation can be solved using the STME algorithm, as follows: suppose that $n - k$ LI row vectors are present within the parity check matrix H, that is, $Rank(H) = n - k = m$ of a (n, k) linear code over F_q, $S \subseteq \{1, 2, \ldots, n\}$, $|S| \geq m$, $S_\theta \subseteq S$, and $|S_\theta| = m$. Selected randomly from H and verified as LI, m column vector $\theta = \{\theta_i | i \in S_\theta\}$, where $|\theta| = m$ is used as a basis for representing an arbitrary m toggle vectors s_x. Assuming that an m syndrome with vectors $s_x = (s_{x,1}, \ldots, s_{x,m})$ corresponds to H, we can obtain an independent matrix $\theta = \{\theta_i | i \in S_\theta \subset n, |S_\theta| = m\}$ from m columns out of H, such that

$$s_x^T = \begin{bmatrix} \theta_{1,i_1} & \theta_{1,i_2} & \cdots & \theta_{1,i_m} \\ \theta_{2,i_1} & \theta_{2,i_2} & \cdots & \theta_{2,i_m} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{m,i_1} & \theta_{m,i_2} & \cdots & \theta_{m,i_m} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{bmatrix}, \tag{23}$$

where $\lambda_i \in F_q$ and $\theta_i = (\theta_{1,i}, \theta_{2,i}, \ldots, \theta_{m,i})^T$. Given θ and s_x, the coordinates $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ corresponding to basis θ can be evaluated as follows:

$$\lambda^T = \begin{bmatrix} \theta_{1,i_1} & \theta_{1,i_2} & \cdots & \theta_{1,i_m} \\ \theta_{2,i_1} & \theta_{2,i_2} & \cdots & \theta_{2,i_m} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{m,i_1} & \theta_{m,i_2} & \cdots & \theta_{m,i_m} \end{bmatrix}^{-1} \begin{bmatrix} s_{x,1} \\ s_{x,2} \\ \vdots \\ s_{x,m} \end{bmatrix}. \tag{24}$$

By using (24), we can immediately obtain a solution $x' = (x_1, \ldots, x_n)$ for $Hx'^T = s_x$. Subsequently, we construct the ith component of x', as follows: if $i \in S_\theta$, then $x_i = \lambda_i$, and, if $i \in S_\theta$, then $x_i = 0$. It then follows that $Hx'^T = \theta\lambda^T = s_x$, where θ is the submatrix of H. The original equation $Hx^T = s_x$ can be resolved for x with a linear combination of column vectors from H. Discovering the least weight $w(x)$ that corresponds to the minimum embedding distortion obtains the least weight coordinate vector λ, associated with a randomly selected basis θ. The adaptive toggle vector x corresponds to the least

weight coordinate vector λ. The output l' of the embedder, that is, the stego vector, is subsequently obtained as $l' = u' - x$. Finally, at the receiver, the secret message $s_{l'}$ is extracted, as follows:

$$\hat{s}_{l'} = Hl'^T = H(u' - x)^T = Hu'^T - \theta\lambda^T = s_{l'}. \tag{25}$$

These embedding procedures are illustrated using the following algorithm.

Algorithm: Adaptive STME algorithm

Encoder: Given a q-ary (n, k) linear block code with $H = [\theta_1 \cdots \theta_i \cdots \theta_n]$, $rank(H) = m$, cover vector u', m message $s_{l'}$, changeable location index S, basis index $S_\theta \subseteq S$, and constant B.

1. Calculate the syndrome
   $s_{u'} = Hu'^T$.
2. The value derived from $s_{l'}$ is subtracted from $s_{u'}$ to obtain s_x.
3. The vector $x = (x_1, \ldots, x_n)$ corresponding to s_x is obtained as follows.
   1) Let $j = 1$, and randomly select m column vectors out of H as $\theta^{(j)} = \left\{\theta_i^{(j)} | i \in S_\theta^{(j)} \subseteq S^{(j)}\right\}$.
   2) Determine whether $\theta^{(j)}$ are LI $\det\left(\theta^{(j)}\right) \neq 0 \rightarrow s_x = \sum_{i \in S_\theta^{(j)}} \lambda_i^{(j)} \theta_i^{(j)}$ and determine $\lambda^{(j)} = \left\{\lambda_i^{(j)} | i \in S_\theta^{(j)}\right\}$. If $\det\left(\theta^{(j)}\right) = 0$, then this is not solvable: revert to step 1).
   3) In the event that $j = B$, then $\lambda = \left\{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(B)}\right\}$ is gained. Proceed to Step 4. Otherwise, $j = j + 1$, and revert to Step 1.
   4) Select a minimum coefficient vector $\lambda' = \{\lambda_i'\}$ from the candidate set.
   5) Obtain $x = (x_1, \ldots, x_n)$ and $\lambda_{i'} \in \lambda_{\min}$ as
   $$x_i = \begin{cases} \lambda_i' & i \in S_\theta^{(j)} \\ 0 & i \notin S_\theta^{(j)} \end{cases}. \tag{26}$$
4. Find the adaptive stego vector as $l' = u' - x$.

Decoder: Recover the message $s_{l'}$ by using l' and H.

5. Extract the embedded data by performing
   $s_{l'} = Hl'^T$.

---

5. **Simulation Results.** The experimental results demonstrate the computational complexity and the embedding efficiency of various embedding algorithms, soving probability for STME algorithm and the embedding efficiency of STME algorithm in comparison with a variation of suboptimal embedding algorithms. The simulations were developed using MATLAB-R2009 and executed using an Intel E8300 2.83G CPU on a computer with win7 operating system and 2G DRAM. In the experiment, for each relative message length, we ran various embedding algorithms based on random codes with a 1/2 code rate. All cover samples were used to carry secret messages s_l block by block. These results were simulated in cover block u of 10^5 and these samples for this simulation were selected uniformly and randomly.

5.1. **Solving probability.** STME algorithm solutions yield a probability level owing to coordinate transform of the chosen submatrix from the parity check matrix of devising the adaptive STME algorithm using random codes. For the STME algorithm, these locations arbitrarily chosen in a cover block to change are called the adaptive STME

embedding. Consider the case of solvability of the random matrix H with an embedding rate $R_m = (k \log_2 q)/n$ and determine the probability of solvability. The term H is an $m \times n$ random matrix over F_q consisting of n columns. The H can be deleted from a number of columns to form a $(n', k) = (n - i, k)$ random code fixed at embedding rate $1/2$, where i is the number of deleted columns. In fact, the likely solving probability for transform matrix is small when deleting a large number of columns. For $m = 16, 14, 12, 10, 8$, there is some approximate probability of solution for the adaptive STME algorithm. Moveover, in the case of the same q-ary, this solution probability increases as the message size m decreases, as shown in Figure 2.

5.2. **Embedding efficiency η for various number B of the candidate bases.** The experiments in this study involved using a q-ary $(16, 8)$ random embedding code for the ataptive STME algorithm. Figures 3 and 4 demonstrate two objectives. The first is to

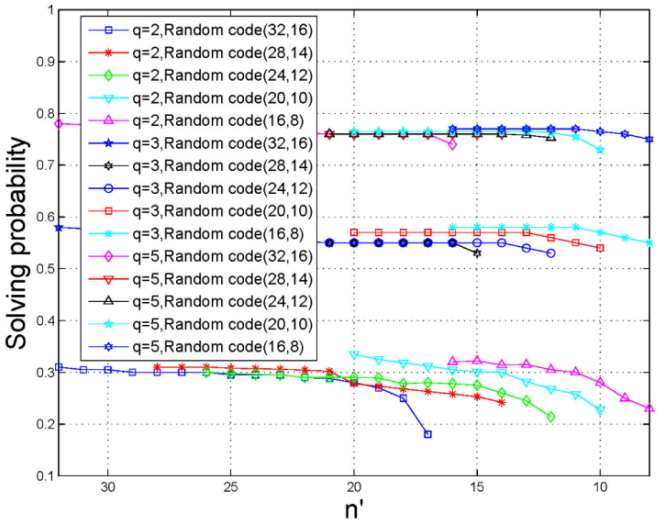


FIGURE 2. (color online) Solving probability for performing adaptive STME algorithm

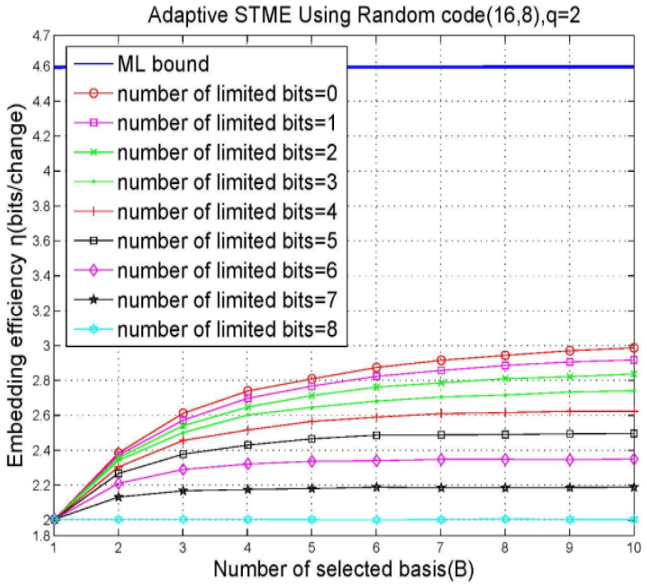


FIGURE 3. Embedding efficiency η versus candidate bases number B

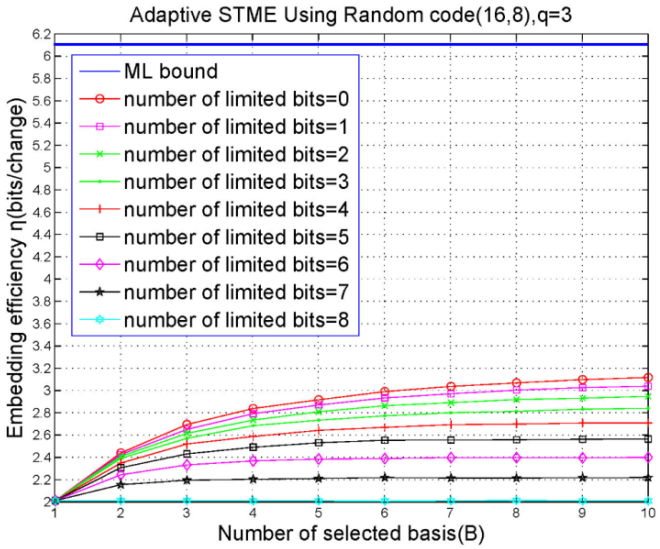
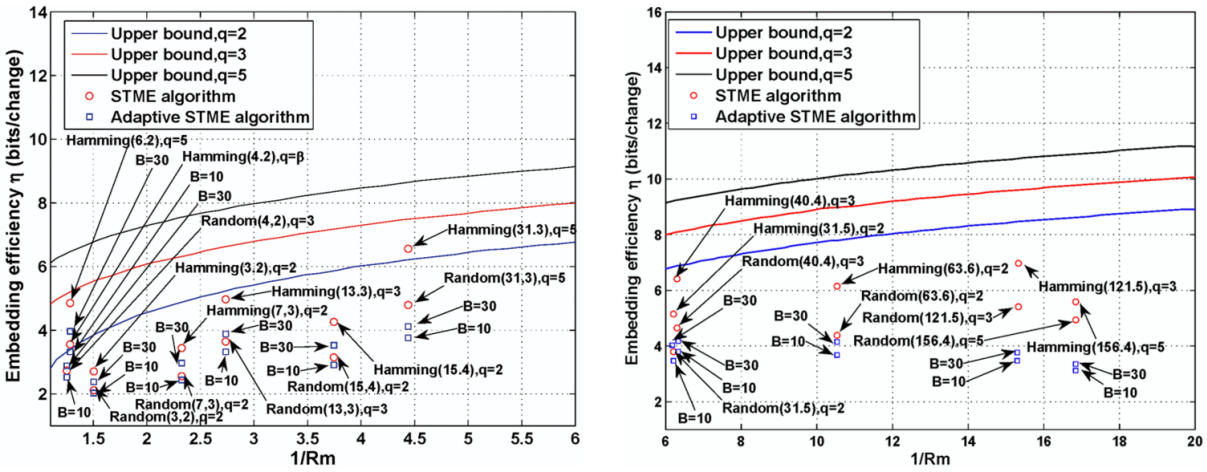


FIGURE 4. Embedding efficiency η versus candidate bases number B



(a) (b)

FIGURE 5. (color online) Embedding efficiency versus inverse embedding rate

show the embedding efficiency η for different B of the candidate bases for the STME algorithm, with B ranging from 1 to 10. The embedding efficiency η increases as the number B increases. The results show that a high B has an insignificant effect on η. The second is to show that the forbidden alteration location ranges from 0 to 8. When the forbidden alteration location is large, η decreases with the number of locations. To achieve the trade-off between the computational complexity and the embedding efficiency in variety of applications, the adaptive STME algorithm can be suitably devised by altering the number of the candidate bases and the forbidden alteration location.

5.3. **Embedding efficiency.** Figure 5 compares the embedding efficiency of various suboptimal algorithms at various inverse embedding rates $1/R_m$. The STME and adaptive STME algorithms are proposed to embed the m vectors of secret messages for each cover block with exiting codes. The number of random forbidden alteration locations for the adaptive STME algorithm consists of 20% of the length n. Figure 5 demonstrates three

subjects: 1) the larger q-ary for various embedding algorithms has high embedding efficiency; 2) when B is decreasing, the embedding efficiency of adaptive STME algorithm is decreasing; and 3) the STME and adaptive STME algorithms are suitable for design at various embedding rates.

5.4. **Computational time complexity.** Since the decoding complexity of optimal embedding algorithms, that is, decoding with ML algorithm, for ME is bounded by finding the coset leader for a linear code, this embedding scheme is too difficult to implement for large linear code. [3] proposed two matrix embedding methods based on random linear codes and simplex codes for near optimal embedding. The decoding algorithms for (n, k) simplex codes in [3] have time complexity $O(n \log n)$, where n is the code length and k is the dimension of the code. Although the embedding efficiency of [3] is close to the upper bound of optimal embedding efficiency for large payloads, a drawback is that the relative payload only works in large payloads. The candidate toggle vectors of the STME algorithm are obtained by searching the part of the coset corresponding to the secret logo s_l and the adaptive STME algorithm is respresented by coordinate transformation only for some tuples in the cover u. Because of this feature, the STME algorithm can be used to improve the limited embedding rate. [4-6] have proposed three embedding algorithms based on various suboptimal decoding methods, called suboptimal embedding algorithm. These methods are unlikely to employ the optimal embedding, (i.e., ML algorithm) to

Table 2. Comparison of the performance and computational time of various embedding algorithms

q-ary	code(n, k)	R_m	η	sec
$q = 2$	[3], $k = 9$	0.982	2.2818	9.36
$q = 2$	[3], $k = 10$	0.99	2.1926	5.3
$q = 2$	[3], $k = 11$	0.995	2.166	3.38
$q = 2$	[3], $k = 12$	0.997	2.1144	2.58
$q = 2$	[4]	0.242	2.7957	0.76
$q = 2$	[5]	0.5161	2.6056	9.41
$q = 2$	[6]	0.125	2.6244	74.33
$q = 2$	ML Random$(16, 8)$	0.5	3.1189	229.2
$q = 2$	ML Random$(16, 12)$	0.25	3.2362	7497
$q = 2$	STME Random$(16, 8)$, $B = 10$	0.5	3.0479	33.0465
$q = 2$	STME Random$(16, 12)$, $B = 10$	0.25	3.111	30.9746
$q = 2$	STME Random$(16, 8)$, $B = 100$	0.5	3.1172	139.443
$q = 2$	STME Random$(16, 12)$, $B = 100$	0.25	3.1761	131.3333
$q = 3$	ML Random$(16, 8)$	0.7925	3.7394	19024.7
$q = 3$	ML Random$(16, 12)$	\	\	\
$q = 3$	STME Random$(16, 8)$, $B = 10$	0.7925	3.1734	19.6957
$q = 3$	STME Random$(16, 12)$, $B = 10$	0.3962	3.5617	18.8699
$q = 3$	STME Random$(16, 8)$, $B = 100$	0.7525	3.709	147.3158
$q = 3$	STME Random$(16, 12)$, $B = 100$	0.3962	3.8261	139.6292
$q = 5$	ML Random$(16, 8)$	1.161	4.4581	8.85×10^7
$q = 5$	ML Random$(16, 12)$	\	\	\
$q = 5$	STME Random$(16, 8)$, $B = 10$	1.161	2.966	15.9181
$q = 5$	STME Random$(16, 12)$, $B = 10$	0.5805	3.5277	15.2299
$q = 5$	STME Random$(16, 8)$, $B = 100$	1.161	3.8227	141.6984
$q = 5$	STME Random$(16, 12)$, $B = 100$	0.5805	4.4305	129.3477

find the coset leader. Although they have rapid embedding time complexity, their embedding efficiencies are inferior to [3] and the adaptive STME algorithm shown in Table 2. [4-6] have the disadvantage which uses the linear codes with poor structure, so that the less performance result. Because the STME only searches a certain number of toggle vectors, the computational complexity requires $O(nB)$, where B (the number of the LI bases) is a constant. The STME algorithm can use linear block codes over F_q to embed, and thus the embedding efficiency is superior to that of embedding over F_2. Our method is superior to [4-6] in embedding efficiency (Table 2). Moreover, the STME algorithm offers a trade-off between embedding efficiency and computational time complexity by altering the constant B. Table 2 shows the speed and operation of embedding for various suboptimal embedding algorithms for random messages fixed at 10^5 bits. The STME algorithm incurs constant complexity regarding the number B of LI bases. By contrast, the complexity cost, with the order of computations $O(nq^k)$, of the ML embedding algorithm plays a crucial role in evaluating optimization for seeking the optimal toggle vector or the coset leader. Table 2 shows that, compared with using the embedding efficiency of the ML embedding algorithm, the STME algorithm performs poorly for random codes. Although the STME algorithm sacrifices a certain degree of efficiency, its computational complexity is superior to that of the ML embedding algorithm, according to the number B of LI bases. The embedding scheme that used the ML algorithm expended most of its computational time on the exponential complexity of the optimal decoding.

6. **Conclusion.** This study proposes the STME algorithm to produce an embedding method that uses $\pm\lfloor (q-1)/2 \rfloor$ embedding technique to modify cover objects and applies it to an arbitrary selection of cover locations. The proposed scheme features decreased embedding complexity in comparison with ML embedding scheme, and it can work as an adaptive embedding method for various applications. In the experiment, we used q-ary random codes and q-ary Hamming codes to implement the STME algorithm. Although the STME algorithm caused some loss of embedding efficiency because our method used suboptimal decoding, the experimental results confirm that the STME algorithm is of low computational complexity compared with the optimal matrix embedding scheme. Moreover, the results also indicate that efficiency is close to the ML embedding by increasing the number of LI sets. Finally, the results show the embedding efficiency levels for STME and STME with specified changeable locations by using random codes and Hamming codes at various embedding rates.

REFERENCES

[1] P. Moulin and R. Koetter, Data-hiding codes, *Proc. of IEEE*, vol.93, no.12, pp.2083-2126, 2005.

[2] C. Cachin, An information-theoretic model for steganography, *Information Hiding: 2nd International Workshop*, vol.1525, pp.306-318, 1998.

[3] J. Fridrich and D. Soukal, Matrix embedding for large payloads, *IEEE Trans. Information Forensics and Security*, vol.1, no.3, pp.390-394, 2006.

[4] R. Y. M. Li, O. C. Au, K. K. Lai, C. K. M. Yuk and S. Y. Lam, Data hiding with tree based parity check, *Proc. of IEEE Int. Conf. Multimedia and Expo (ICME07)*, pp.635-638, 2007.

[5] R. Y. M. Li, O. C. Au, C. K. M. Yuk, S. K. Yip and S. Y. Lam, Halftone image data hiding with block-overlapping parity check, *Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP07)*, vol.2, pp.193-196, 2007.

[6] Y. C. Tseng, Y. Y. Chen and H. K. Pan, A secure data hiding scheme for binary images, *IEEE Trans. Commun.*, vol.50, no.8, pp.1227-1231, 2002.

[7] R. Crandall, Some notes on steganography, *Steganography Mailing List*, 1998.

[8] J. Bierbrauer, *On Crandall's Problem*, 1998.

[9] M. Khatirinejad and P. Lisonek, Linear codes for high payload steganography, *Discrete Applied Math.*, vol.157, no.5, pp.971-981, 2009.

[10] G. Cohen, I. Honkala, S. Litsyn and A. Lobstein, *Covering Codes*, North-Holland, Amsterdam, The Netherlands, 1997.
[11] W. Zhang and S. Li, A coding problem in steganography, *Des. Codes Cryptogr.*, vol.46, no.1, pp.67-81, 2008.
[12] C. Hou, C. Lu, S. Tsai and W. Tzeng, An optimal data hiding scheme with tree-based parity check, *IEEE Trans. Image Process.*, vol.20, no.3, pp.880-886, 2011.
[13] T. Yang and H. Chen, Matrix embedding in steganography with binary Reed-Muller codes, *IET Image Processing*, vol.11, pp.522-529, 2017.
[14] S. Dirksen and A. Stollenwerk, Fast binary embeddings with Gaussian circulant matrices: Improved bounds, *Discrete and Computational Geometry*, pp.599-626, 2018.
[15] S. Kim and S. Choi, Sparse circulant binary embedding: An asymptotic analysis, *IEEE Trans. Signal Processing Letters*, vol.25, no.3, pp.432-436, 2018.
[16] B. A. Lamacchia and A. M. Odlyzko, Solving large sparse linear systems over finite fields: Advances in cryptology, *Lecture Notes in Computer Science*, vol.537, pp.109-133, 1991.
[17] D. H. Wiedemann, Solving sparse linear equations over finite fields, *IEEE Trans. Inf. Theory*, vol.32, no.1, pp.54-62, 1986.
[18] A. Ker, Steganalysis of LSB matching in grayscale images, *IEEE Singal Process. Lett.*, vol.12, no.6, pp.441-444, 2005.
[19] D. Soukal, J. Fridrich and M. Goljan, Maximum likelihood estimation of secret message length embedded using PMK steganography in spatial domain, *Proc. of IST/SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VII*, vol.5681, pp.595-606, 2005.
[20] P. W. Wong, H. Chen and Z. Tang, On steganalysis of plus-minus one embedding in continuous-tone images, *Proc. of IST/SPIE Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VII*, vol.5681, pp.643-652, 2005.
[21] F. Willems and M. van Dijk, Capacity and codes for embedding information in gray-scale signals, *IEEE Trans. Inf. Theory*, vol.51, no.3, pp.1209-1214, 2005.