A Modified Product Code Over $\mathbb{Z}_4$ in Steganography with Large Embedding Rate

Lingyu Zhang[1,2], Deyuan Chen[3]
[1] The State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Computer and Communication Engineering, Liaoning Shihua University
Fushun, P.R. 113001, China
[e-mail: zhanglingyu@iie.ac.cn]
[3] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences
Beijing 100093, China
[e-mail: chendy@ucas.ac.cn]
*Corresponding author: Lingyu Zhang

Received January 2, 2016; revised March 31, 2016; revised April 6, 2016; accepted April 28, 2016; published July 31, 2016

Abstract

The way of combination of Product Perfect Codes (PPCs) is based on the theory of short codes constructing long codes. PPCs have larger embedding rate than Hamming codes by expending embedding columns in a coding block, and they have been proven to enhance the performance of the F_5 steganographic method. In this paper, the proposed modified product codes called MPCs are introduced as an efficient way to embed more data than PPCs by increasing $2^{r_2-1}-r_2$ embedding columns. Unlike PPC, the generation of the check matrix $\boldsymbol{H}$ in MPC is random, and it is different from PPC. In addition a simple solving way of the linear algebraic equations is applied to figure out the problem of expending embedding columns or compensating cases. Furthermore, the MPCs over $\mathbb{Z}_4$ have been proposed to further enhance not only the performance but also the computation speed which reaches $O(n_1+\sigma)$. Finally, the proposed $\mathbb{Z}_4$-MPC intends to maximize the embedding rate with maintaining less distortion , and the performance surpasses the existing improved product perfect codes. The performance of large embedding rate should have the significance in the high-capacity of covert communication.

Keywords: Product perfect codes (PPCs), modified product codes (MPCs), larger embedding rate, $\mathbb{Z}_4$-MPCs

This work was supported by the State Key Program of National Natural Science Foundation of China (Grant No. 61032006), the National Science Foundation of China (Grant No. 61072045, 61271282), and the Award Foundation of Chinese Academy of Sciences (Grant No. 2069901) , in part by "Science100 Program" of the Chinese Academy of Sciences under Grant Y12901HEA2, the NSFC under 61170281 and 61303259, and the Strategic Priority Research Program of Chinese Academy of Sciences under XDA06030600.

http://dx.doi.org/10.3837/tiis.2016.07.027 **ISSN : 1976-7277**

1. Introduction

As a secret communication method, steganography allows the sender to hide the secret messages into the covers of carrier media files such as digital images, audios or videos etc., and send them to the receiver via public channel. Despite that steganographic algorithm only modifies the most significant components, the attackers can detect the suspicious clues by using steganalysis techniques. Generally, the large hidden data can make the steganographic carriers become vulnerable to be detected. One way of improving security is to reduce the amount of embedding changes when we embed the less number of secret bits into larger covers. Thus it is a key point for steganographer to obtain algorithms with high performance, and the high embedding efficiency is needed. Many existing works of steganography followed this principle. Based on R. Crandall introduced matrix encoding [6], A. Westfeld proposed a famous steganography F5 [1] whose embedding efficiency is higher than 2 and can be even higher when dimension k is larger. R.Y.Zhang et al. have improved the computing efficiency or embedding efficiency of BCH [3] through specific mapping way to find the flipping positions. Later V.Sachnev and H.J.Kim [4] redesigned the BCH-based data hiding scheme to embed more bits into the combined blocks rather than a single block for improving the embedding efficiency in JPEG steganography. Up to now, many outstanding works literatures [13][14][17-19] are proposed to improve the average distortion (embeddable secret bits/cover size), and better visual quality can be obtained.

However, these statements mentioned above can result in good embedding efficiencies, but low embedding rate. In a practical application, the real-time transporting or communicating message is urgently needed, e. g., near field communication (NFC) has important implications [20-21]**.** Usually, because NFC needs relatively less amount of communication time, the requirement of mass capacity is necessary. In addition, during communicating with the store mediums, i.e., audios, videos, images, information hiding with high capacity has become an important research content in these field. In certain context, high embedding rate codes are usually needed. Fridrich *et al*. proposed the "Matrix embedding for large payload" [2], these practical schemes are based on simplex codes with respect to small dimensions. In 2009, H. Rifà-Pous and J. Rifà proposed PPCs [5] for steganography, they embedded the secret messages into each row and each column based on Hamming codes. Specially, the embedding efficiency of PPCs have always surpassed the simplex codes, and with a small dimensions ($r \leq 3$), the embedding rate of PPCs was larger than simplex codes too. In 2010, J. Rifà *et al*., proposed product perfect $\mathbb{Z}_2\mathbb{Z}_4$-linear codes [9], the N-length packet x of symbols is translated into vector $w \in Z_2^{\alpha} \otimes Z_4^{\beta}$ of α binary and β quaternary coordinate. The codeword of this code are all the $n \times N$ matrics, where $N = 2^{m-1}, n = 2^m - 1, m \geq 2$, such that all rows are codeword in $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect code of type $(\alpha, \beta; \gamma, \delta)$, and so it is the first column after applying the inverse of the extended Gray map Φ. However, a recursively generated way is applied by taking the codes whose codeword are all the l-dimensional matrices. When $l = n \times (n \times \ldots, n \times N)$, the upper bound of embedding rate and average distortion converge to 1.5 and 0.38 based on CI-rate [9]. Especially, when $l = n \times N$, the performance is degraded to 1.33 and 0.57 respectively. Later, Ze-li Zhao *et al*. combined F_5 with wet paper codes (F_5-EMD-WPC) [10] method to improve PPCs. That method also utilizes the structure of the PPCs, the difference is that the codes in the first c columns are composed by EMD+WPC and its compensating case in the corresponding row is the same as PPCs'. In a 3×3 embeddable block, its performance upper bound are 0.97 and 0.375 respectively. In this paper, the

proposed $\mathbb{Z}_4$-MPCs obtain superior performance in the following aspects. The first superiority is that $\mathbb{Z}_4$-MPCs utilize the shared key to randomly build check matrix H instead of only using fixed way to generate H in PPCs, and the embedding rate of $\mathbb{Z}_4$-MPCs can be enlarged by increasing $2^{r_2-1}-r_2$ embedding columns in a $C_1 \otimes C_2$ product code, where C_1 and C_2 are Hamming codes, respectively. Furthermore, MPCs is expanded over $\mathbb{Z}_4$ called $\mathbb{Z}_4$-MPC. It utilizes the Gray map to form double layer embedding channels for doubling the embedding rate. In the course of column embedding, the message of corresponding row can be modified, and a compensation is needed for correct the message. The theory is defined as Lemma 1 which is based on the simple solution of a linear equation. Second, in order to decrease the number of flipping, the column embedding operation is restricted to the first r_1 rows by setting wet spots for the last k_1 positions in each column. Thus, there are at most r_1 rows needed to be changed for the compensation, and the the growth rate of number of flipping becomes slow. Third, the embedding rate performance upper bound of $\mathbb{Z}_4$-MPCs achieves 1.78 *bpp*, and the average distortion achieves 0.57 which surpass the performance of all the existing improved product perfect codes. Finally, because the values of k free variables of a linear equation in $q^k | q = 4$ solution space are known by setting wet spots, the complexity of one time processing during coding in each row, column and compensation is O(1), and the total computational-complexity of this proposed reaches $O(n_1 + \sigma)$, $0 \le \sigma \le r_1 \times k_2$, σ is a positive integer. The complexity of the proposed is lower than PPC which computational-complexity reaches $O(2^{k_2}(n_1 + c))$, $1 \le c \le 2^{r_2-1} - 1$. Specially, when the proposed code with small dimension ($r = 2$) is needed, the Lee distance can be utilized in row embedding course. The upper bound of average distortion can be decreased 3%, and the computational-complexity is $O(q^2 n_1 + \sigma)$.

The remainder of the paper is organized as follows. Section 2 is review of the details of the product perfect codes. In Section 3, we proposed $\mathbb{Z}_4$-MPCs and gave their detail of embedding and extracting coding processes. In Section 4, the performance of the proposed method is compared with the other improved product codes. Finally, Section 5 gives a conclusion of this method.

2. Review Product Perfect Codes

In this section, we review the theory of product perfect codes, which is the foundation of our proposed method. Let $\mathbb{F}_q$ be a finite field of q elements, where q is a prime power. Let code C_1 and C_2 be two binary linear codes of length n_1 and n_2, where $n_1 = 2^{r_1} - 1$ and $n_2 = 2^{r_2} - 1$, respectively. The product code $C_1 \otimes C_2$ is the tensor product of the two linear code $\boldsymbol{u} \in C_1$ and code $\boldsymbol{v} \in C_2$. The tensor product is generated by the vectors in the form of:

$$\boldsymbol{u} \otimes \boldsymbol{v} = \left(u_i v_j \middle| 1 \le i \le n_1, 1 \le j \le n_2\right), \tag{1}$$

where $\boldsymbol{u} = (u_1, u_2, \dots, u_{n_1}) \in C_1$ and $\boldsymbol{v} = (v_1, v_2, \dots, v_{n_2}) \in C_2$. The product code $C_1 \otimes C_2$ is defined as a set of tensor product in which each row is an element in C_2 and each column is an element in C_1 , where the length of product code is $n = n_1 \times n_2$ and the dimension of product code is $k = (n_1 - r_1) \times (n_2 - r_2)$. The cover image is divided into disjoint matrix blocks of $n_1 \times n_2$ pixels.

Embedding:

Step I In each row, there would be changed at most only one bit in code C_1. Perform the embedding row by row.

Step II After processing all the rows, they start embedding from the first column to the $2^{r_2-1}-1$th column. When the pixel is changed in the jth column and ith row, there are adjustments to ensure the correct extraction in this row. The details of the adjustments are demonstrated in [5].

Extracting:

The message of rows and the first $2^{r_2-1}-1$th columns can be calculated from $H_r \cdot \boldsymbol{y}^T$, where $\boldsymbol{y}^T$ is the stego-code in each row or column, H_r is the check matrix defined in [5], and the best results is obtained when $r_1 = r_2$.

3. Modified Product Codes

The PPCs only utilize the column embedding within the first $2^{r_2-1}-1$ columns. Actually, in order to improve the performance we can utilize at most $k_2 = n_2 - r_2$ columns to be embedded. Obviously, the inequality $k_2 \geq 2^{r_2-1}-1$ is always satisfied, where $r_2 \geq 2$. When $r_2 = 2$, the embedding rate of MPC is degenerated to PPC's.

3.1 Theories of Modified Product Codes

In the theory of Modern Algebra, the 2^k codewords form a k-dimension subspace C of the n-dimension linear vector space $\sum^n$, and the k independently codewords form a basis, that is, the basis can expand the $\sum^n$. The 2^k codewords can be obtained by a $r \times (2^r - 1)$ check matrix which rank is $n-k$. If the information element constantly and randomly appears at the first k bits of codeword $\forall c \in C$, the C can be called systematic codes and its check matrix can be called a systematic check matrix, otherwise C can be called nonsystematic codes and has the corresponding nonsystematic check matrix. In this paper, the used check matrix is considered as a systematic or nonsystematic check matrix in any way is randomly generated by a key, and the $r \times (2^r - 1)$ check matrix H is defined as follows:

$$\mathbf{H} = (\boldsymbol{h}_1, \dots, \boldsymbol{h}_k, \dots, \boldsymbol{h}_n), \tag{2}$$

where $\boldsymbol{h}_i | i \in \{1, \dots, n\}$ is a binary column vector. A certain restriction must be concluded that during the process of generating check matrix $\mathbf{H}$ by a key, the row rank should be equal to r. If the condition is invalid, change key until the restriction is valid.

There are several lemmas which are utilized for the following algorithm to be defined as follows: Suppose we have already previously modified the ith row during the row embedding, and we want to embed message into jth column in the course of column embedding. The column embedding will generate two error patterns for j th column and the i th row respectively resulting from a pixel changed in coordinate (i, j), then the row code and column code have been modified. In order to correct message in ith row, we must find some compensatory error patterns to accurately extract the embeddable message $\boldsymbol{m}$ in ith row. This process is defined as a compensation for the ith row. Suppose the cover code of ith row is $\boldsymbol{x}$, the error pattern is $\boldsymbol{e}$, the stego code is $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$, the equality $\mathbf{H} \cdot \boldsymbol{y}^T = \boldsymbol{m}$ holds, where $\boldsymbol{m}$ denotes the embedded message of ith row. Now, if the pixel to be changed locates in position (i, j), the error pattern of ith row is $\boldsymbol{e}' = (0, \dots, e_j, \dots, 0) | e_j = 1, j \in \{1, \dots, k\}$, and the stego code of this row is changed to $\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{e}'$, where $\mathbf{H}\boldsymbol{y}'^T = \boldsymbol{m}' \neq \mathbf{H}\boldsymbol{y}^T$. The lemma 1 demonstrates the fact that there exists compensatory error pattern set $\{\boldsymbol{e}'' = (0, \dots, e_j, \dots, 0) | e_j = 1, j \in \{k+1, n\}\}$, which make the equation $\mathbf{H}\boldsymbol{y}'^T + \sum \boldsymbol{e}''^T = H\boldsymbol{y}^T$ hold. The lemma is proved as follows:

Lemma 1. Suppose known conditions are $\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{e}'$, $\mathbf{H}\boldsymbol{y}'^T = \boldsymbol{m}' \neq \boldsymbol{m}$, where $\boldsymbol{m} = \mathbf{H}\boldsymbol{y}^T$, the error pattern of code $\boldsymbol{y}'$ is $\boldsymbol{e}' = (0, \dots, e_j, \dots, 0) | e_j = 1, j \in \{1, \dots, k\}$. If there exists compensatory error pattern set $\{\boldsymbol{e}'' = (0, \dots, e_j, \dots, 0) | e_j = 1, j \in \{k+1, n\}\}$, then the equality $\mathbf{H}\boldsymbol{y}'^T + \mathbf{H}\sum \boldsymbol{e}''^T = \boldsymbol{m}$ holds.

Proof.

Suppose the compensation error pattern set is $\{\boldsymbol{e}'' | \mathbf{H}\boldsymbol{e}''^T \in h_i | i \in \{k+1, \dots, n\}\}$. Based on the known conditions, the equation $\mathbf{H}\sum \boldsymbol{e}''^T = \mathbf{H}E^T = \boldsymbol{m} - \mathbf{H}\boldsymbol{y}'^T$ needs to be solved. If and only if the free variables of the linear equation are designated as the first k elements by setting $E = (0, \dots, 0, e_{k+1}, \dots, e_n)$, the solution E is obtained by the Gaussian Elimination method. The equation $\mathbf{H}\boldsymbol{y}'^T + \mathbf{H}\sum \boldsymbol{e}''^T = \boldsymbol{m}$ is proved. □

When we do column embedding for the first k columns, the chance of flipping for the compensation only happened in the last r pixels of ith row based on the Lemma 1. The result is that the compensation is valid and the message $\boldsymbol{m}$ in the ith row can be correct extracted.

Lemma 2. If the unknown error patter is set $E = (0, \dots, 0, e_1, \dots, e_r)$, the equation $-\mathbf{H} \cdot \boldsymbol{x}^T + \boldsymbol{m} = \mathbf{H} \cdot E^T$ can be obtain the unique solution over $\mathbb{Z}_4$.

Lemma 3. The solution of equation $-\mathbf{H} \cdot \boldsymbol{x}^T + \boldsymbol{m} = \mathbf{H} \cdot E^T$ can be computed by Gauss Elimination based on minimum Lee Distance which is defined as,

$$d_{min} = min\{d_L(C_1, C_2)\}, \tag{3}$$

where $C_1, C_2 \in C$ be the codewords over $\mathbb{Z}_4$, $d_L(C_1, C_2) = w_L(C_1 - C_2)$, $w_L(0) = 0$, $w_L(1) = 1, w_L(2) = 2, w_L(3) = 1$. d_L denotes Lee Distance between C_1 and C_2, w_L denotes Lee Weight. d_{min} reflects the error-correcting ability of $\mathbb{Z}_4$-Hamming code.

Lemma 4. There are two error patterns $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$with the same Lee Distance computed from equation $-\mathbf{H} \cdot \boldsymbol{x}^{\mathrm{T}} + \boldsymbol{m} = \mathbf{H} \cdot E^{\mathrm{T}}$. Then whatever which error pattern is chosen the message $\boldsymbol{m}$ can be exactly fetched.

Proof. Suppose $\boldsymbol{y}_1 = \boldsymbol{x} + \boldsymbol{e}_1$, $\boldsymbol{y}_2 = \boldsymbol{x} + \boldsymbol{e}_2 | \boldsymbol{e}_1 \neq \boldsymbol{e}_2$, then $\mathbf{H}\boldsymbol{y}_1{}^T = \mathbf{H}(\boldsymbol{x} + \boldsymbol{e}_1)^T$, $\mathbf{H}\boldsymbol{y}_2{}^T = \mathbf{H}(\boldsymbol{x} + \boldsymbol{e}_2)^T$. Because $\mathbf{H}\boldsymbol{e}_1{}^T = \mathbf{H}\boldsymbol{e}_2{}^T$, then $\mathbf{H}\boldsymbol{y}_1{}^T = \mathbf{H}\boldsymbol{y}_2{}^T$ and the equation $\boldsymbol{m} = \mathbf{H}\boldsymbol{y}_1{}^T = \mathbf{H}\boldsymbol{y}_2{}^T$ is proved. □

Lemma 5. Suppose the $[n, k]$ Hamming code over $\mathbb{Z}_4$, suppose e of the code denotes the maximum number of error correction, and which is also denoted as cover radius ρ of q^k codewords $\{C_j | 0 \leq j \leq q^k - 1, q = 4\}$. The cover radius ρ can be computed by the follows:

1) Suppose the number of Cosets is q^{n-k}, set $sum \coloneqq 0$, $i \coloneqq 0$.

2) $sum \coloneqq sum + (q-1)^i \times C_n^i$. If $sum < q^{n-k}$, $i \coloneqq i + 1$ and do 2), else do 3).

3) $\rho \coloneqq i$.

Lemma 6. Suppose the $[n, k]$ Hamming code over $\mathbb{Z}_4$, $r = n - k$, if the inequation $q^r < (q-1)^i \times C_n^i | q = 4, r \geq 2$ holds, then inequation $i \geq 2$ holds, where i is positive integer and the cover radius ρ of this code is 2.

Proof. Based on mathematical induction, the inequation $q^r < (q-1)^i \times C_n^i | q = 4, r \geq 2, i \geq 2$ is obvious true. Based on Lemma 5, the cover radius $\rho = 2$ is proved. □

3.2 The Procedure of Coding

Before the introduction of the embedding and extracting process, a generic flow chart of this

proposed scheme is depicted in **Fig. 1.**

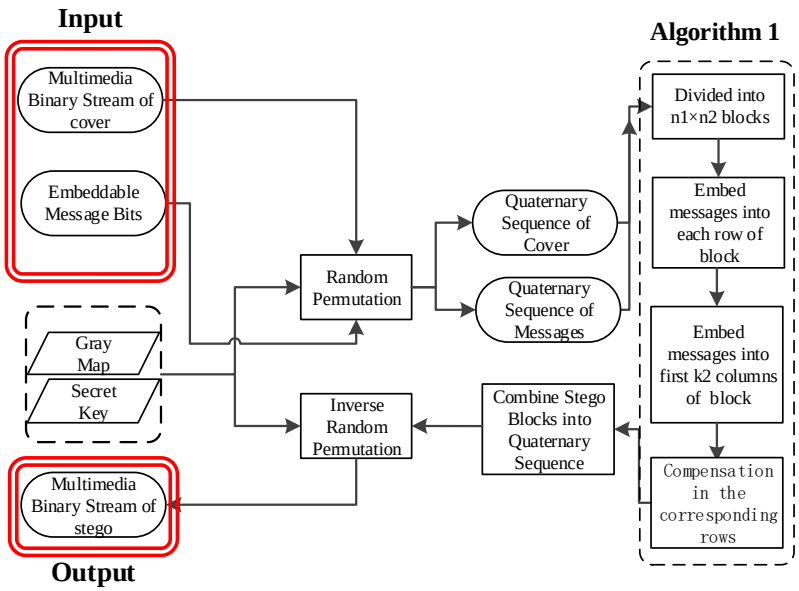


Fig. 1. Flow chart of $\mathbb{Z}_4$-MPC steganographic scheme.

In this paper, the MPCs can be similarly defined as $\mathbb{Z}_4$-MPCs over $\mathbb{Z}_4$ which belong to double-layered embedding channel method. We assume a cover sequence $\boldsymbol{x} = (f(x_1), \ldots, f(x_N))$, where N is the block length, $x_i \in \aleph = \{0,1, \ldots, 2^B - 1\}$, where $B = 8$. The cover belongs to the kind of 256 Gray scale image. Note that the function f denotes the needed significant bits of cover x_i,where $f = \{\Phi(SLSB(x_i), LSB(x_i)) | i \in \{1, \ldots, N\}\}$, LSB and SLSB denote the least and the second least significant bit respectively, and Φ denotes the usual Gray map from $\mathbb{F}_2$ to $\mathbb{Z}_4$: $\Phi(0,0) = 0$, $\Phi(0,1) = 1$, $\Phi(1,1) = 2$, $\Phi(1,0) = 3$. Let message $\boldsymbol{m}$ with respect to quaternary sequence be hided into the cover $\boldsymbol{x}$, the corresponding stego sequence is $\boldsymbol{y} = \boldsymbol{x} + E$, and the equality $\mathbf{H}\boldsymbol{y} = \boldsymbol{m}$ holds. The cover image is segmented into nonoverlapping blocks, which size is $(2^{r_1} - 1) \times (2^{r_2} - 1)$, and the cover sequence $\boldsymbol{x}$ is arranged to $(x_{1,1}, \ldots, x_{1,n_2}), \ldots, (x_{n_1,1}, \ldots, x_{n_1,n_2}) | x_{i,j} = f(x_k), x_k \in \aleph, k = (i-1) \times n_2 + j$. The structure of the proposed code is shown in **Fig. 3**. The implementation is based on a simple solution way of the linear algebraic equations which will be described in the following steps. The coding procedure of $\mathbb{Z}_4$-MPCs is given in Algorithm 1.

Algorithm 1 $\mathbb{Z}_4$-MPCs Algorithm.

Embedding:

Step I For ith row, let $X_i^r = (x_{i,1}, \ldots, x_{i,2^{r_2}-1})$, sign r denotes the operation of row embedding, E denotes the unknown error pattern, $\boldsymbol{m}_i \in \mathbb{Z}_4$denotes the embeddable message in this row, and the check matrix symbolized by $\mathbf{H}$. Solve the equation according to the matrix encoding method [2] in such a way that $-\mathbf{H} \cdot X_i^{rT} + \boldsymbol{m}_i = \mathbf{H} \cdot E^T$ holds. The solution is obtained by Lemma 2 or Lemma 3. In Lemma 3, although there exists some cases that some error patterns have the same Lee Distance (d_{min}), we can choose the error pattern with minimum number of flipping as the final solution, and because that error patterns can be mapped to the same syndrom which can be utilized to exactly fetch the embeddable message in the end based on Lemma 4. Finally, the stego sequence in ith row is $Y_i^r = X_i^r + E$. We apply this embedding procedure for $2^{r_1} - 1$ times.

Step II For jth column $(j \in \{1, \dots 2^{r_2} - 1 - r_2\})$, let $X_j^c = (x_{1,j}, \dots, x_{2^{r_1}-1,j})$, sign c denotes the operation of column embedding, $m_j \in \mathbb{Z}_4$ denotes the embeddable message in jth column. Solve the equation $-\mathbf{H} \cdot X_j^{cT} + \boldsymbol{m}_j = \mathbf{H} \cdot E^T$, and let $E = (e_1, \dots, e_{r_1}, 0, \dots, 0)$. The last k_1 free variables are set zero, and there exist a unique solution E. The stego sequence in jth column is $Y_j^c = X_j^c + E$. The stego sequence in ith row is changed to $Y_i^{r\prime} = Y_i^r + \boldsymbol{e}'$, where $\boldsymbol{e}' = (0, \dots, e_j, \dots, 0) | e_j = 1, j \in \{1, \dots, k\}$. In such case, the changed coordinate in (i, j)th will generate non-desirable effects in the ith row. Based on lemma 1, there exists error pattern set $\{\boldsymbol{e}'' | \mathbf{H}e''^T \in h_i | i \in \{k_2 + 1, \dots, n_2\}\}$ to be used to accurate extract the embeddable row message $\boldsymbol{m}_i$.

Step III Repeat the procedure with Step II until the $(2^{r_2} - 1 - r_2)$th column is finished.

Extracting:

The message of rows and the first $2^{r_2} - 1 - r_2$ columns can be calculated from $H \cdot \boldsymbol{y}^T$, where $\boldsymbol{y}^T$ is the stego-code in each row or column.

In Step I, there exists two ways to embed message bits into each row based on Lemma 2 or Lemma 3. The method of Lemma 2 can obtain the optimization of computation speed, and the the method of Lemma 3 can obtain the optimization of number of flipping under the tolerable performance. The solution space based on Lemma 3-Lee Distance (d_{min}) is q^{k_2}, which will make the computational speed grow exponentially. However, the number of flipping of improved method is decreased, and the running speed under low dimension code is acceptable e. g., for a 512×512 gray image, when $r_1 = r_2 = 2$ and suppose the payload is $\frac{26214}{512 \times 512} = 0.1\ bpp$, the running speed reaches to 10″ under MATLAB running environment. By synthesizing each kind of situation, when the embedding capacity is too much, the smaller dimension $\mathbb{Z}_4$-MPC with $r_1 = r_2 = 2$ can be competent to the task. We take an instance to describe the procedure of embedding. Suppose the check matrix $H = (1\ 0\ 1; 1\ 1\ 0)$. In order to simplify the course, we only embed message bits into the first row which cover code is ${x_1}^{row} = (3,1,2)$, and the first column which cover code is ${x_1}^{column} = (3,0,2)$ in the 3×3 cover block. Suppose the embeddable message bits in this row is $\mathrm{m}_{r1} = (1,1)$. Compute the equation $-\mathbf{H}{\boldsymbol{x}_1}^{rowT} + {\boldsymbol{m}_{r1}}^T = \mathbf{H}E^T$, where E is error pattern. Solved it by d_{min} and Gaussian Elimination, then $E = (010)$ and stego code ${\boldsymbol{y}_1}^{row} = E + {\boldsymbol{x}_1}^{row} = (3,2,2)$. Suppose the embeddable message bits in this column is $\mathrm{m}_{c1} = (2,0)$. Compute the equation $-\mathbf{H}{\boldsymbol{x}_1}^{columnT} + {\boldsymbol{m}_{c1}}^T = \mathbf{H}E^T$, and obtain the $E = (1,0,0)$. Then ${\boldsymbol{y}_1}^{column} = {\boldsymbol{x}_1}^{column} + E = (0,0,2)$, and ${\boldsymbol{y}_1}^{row} = (0,2,2)$. Based on Lemma 1, the corresponding compensation for this row is needed. Compute equation $\mathbf{H}{\boldsymbol{y}_1}^{rowT} + \mathbf{H}E = \boldsymbol{m}_{r1}$, where $E = (0, e_2, e_3)$. The solution $E = (0,3,3)$, and ${\boldsymbol{y}_1}^{row} = {\boldsymbol{y}_1}^{row} + E = (0,1,1)$. The variation in this row is $(+1,0,-1)$. Two positions are changed for embedding 8 bits messages. In the other rows, there exists row embedding procedures, and the total expected number of flipping and the number of embeddable bits are analysised in section 4.

3.3 The Superiority of Code Structure Characteristics

3.3.1 High Capacity Information Hiding for Covert Communication

The $\mathbb{Z}_4$-MPC improves the embedding rate (bits of secret messages /code length, bits per pixel, bpp), and the value not only exceeds other improved product codes but also the existing large payload code scheme. The value is realized by the follows: First, the number of embedding column channels is increased by expanding $2^{r_2-1} - r_2$ embedding columns in a $C_1 \otimes C_2$

product code over $\mathbb{F}_2$ being referred to as single-layered steganographic algorithm. Second, the novel data hiding method transforms the secret data into a stream of secret digits by using a q-ary notational system in order to generating the modified product code over $\mathbb{Z}_4$. The method is referred to as double-layered steganographic algorithm, and the embedding rate obtain a significant rise. Actual structure of embedding channels is shown in **Fig. 2**.

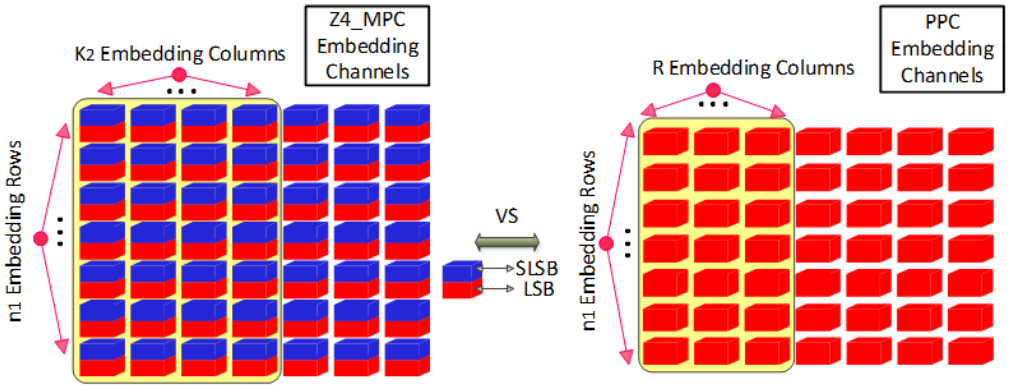


Fig. 2. Actual embedding channels : $[n_1, k_1]\otimes[n_2, k_2]$ modified product code over $\mathbb{Z}_4$($\mathbb{Z}_4$-MPC) vs $[n_1, k_1]\otimes[n_2, k_2]$ product code over $\mathbb{F}_2$(PPC)

In PPC, the number of embedding columns is $R = 2^{r_2-1} - 1 \leq k_2$.

3.3.2 The Superiority of Cluster Structure

In the proposed algorithm, the distorted region of a $n_1 \times n_2$ emeddable block can concentrate on a certain region ranging from the coordinate $(1,1)$ to coordinate (r_1, k_2). First, the modified vector of each row and column embedding is computed by equation $H\boldsymbol{x}^T + HE^T = \boldsymbol{m}$, where $E = (0, \ldots, 0, e_{k_2+1}, \ldots, e_{n_2})$ or $E = (e_1, \ldots, e_{r_1} 0, \ldots, 0)$. The corresponding region is depicted in the solid or dotted line borders in **Fig. 3**. Second, similarly, the compensated changes based on lemma 1 coincide with the changes of row embedding within the region ranging from the coordinate $(1, k_2 + 1)$ to coordinate (r_1, n_2), and the expected number of changes of compensation needn't be computed extraly. The distribution of the distorted region is more concentrated than the random distribution generated by Coset algorithm [5], and the region is shown in the dotted borders in **Fig. 3**.

The characteristic of high payload could be applied in real time communication e.g., the NFC. A large embedding rate code is necessary. At the same time, high payload performance of this code can make one cover element carrying more message bits, the result is that the number of changed positions is decreased, and because of the small dimension of this proposed code the distribution of the changed position is more concentrated. Whatever the type of media cover or resolution is, the proposed code can be utilized as long as the mapping relation would be existing. Furthermore, for convenience, the proposed code is applied to adaptive steganography in 512×512 image which is selected from BOSS1.01 for describing the characteristic of cluster, and the chosen candidate embeddable blocks (the image can be divided into $\left\lfloor \frac{512\times512}{n_1\times n_2} \right\rfloor$ blocks) are desided by a block-locating function which make the embedding positions concentrate on regions of complex textrue. The adaptive steganography being applied in video, image or other multimedia cover is worth to have further research. The embedding effect combining with the cost function to forming an adaptive steganograpy is shown in **Fig. 4**.

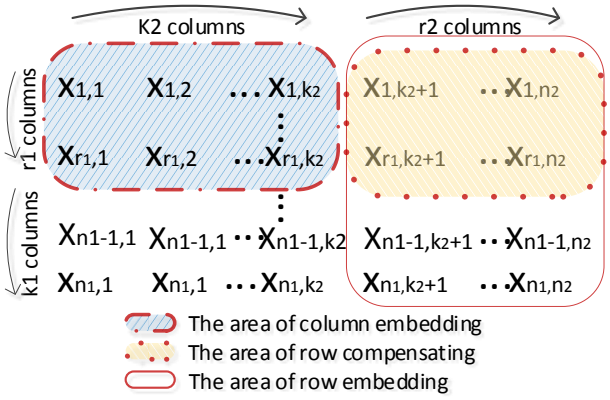
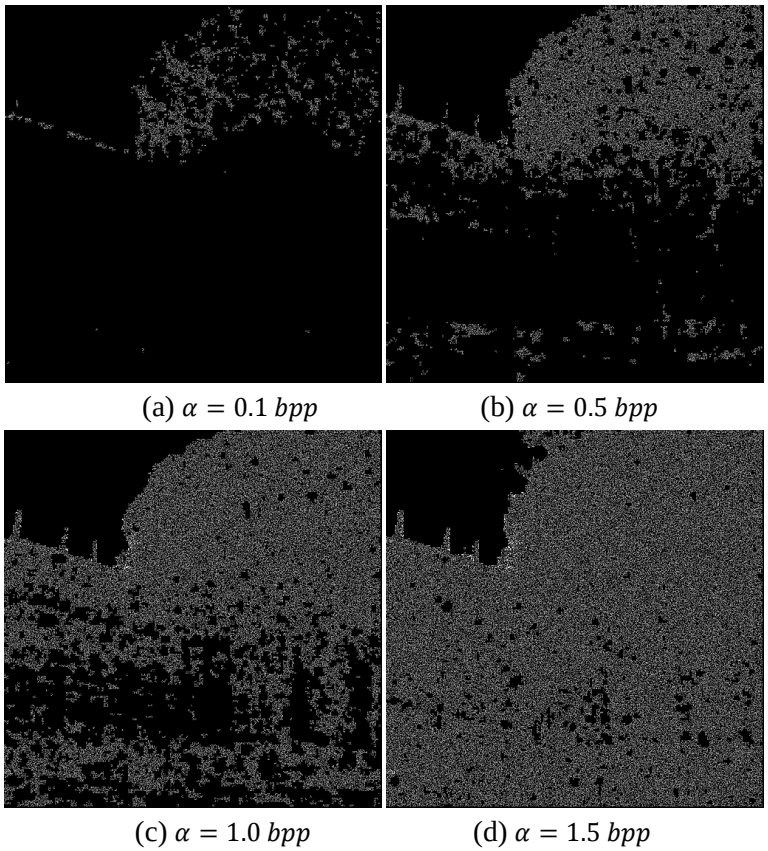


Fig. 3. The disturbed area of $\mathbb{Z}_4$-MPC



(a) $\alpha = 0.1\ bpp$ (b) $\alpha = 0.5\ bpp$

(c) $\alpha = 1.0\ bpp$ (d) $\alpha = 1.5\ bpp$

Fig. 4. Actual embedding changes executed by Z4-MPC($r = 2$) with payload rate(embeddable bits/cover size) $\alpha \in \{0.1, 0.5, 1.0, 1.5\}\ bpp$

The disturbed region of adaptive $\mathbb{Z}_4$-MPC is converged on complex texture area. We can clearly see that when payload increases the distorted region gets bigger. However, the expanded region revolves around the complex texture area because of the small dimension code can easily capture low cost area. The characteristic of clustering structure indicates that this proposed codes with smallest dimension can make the flipping positions are highly

converged in region of complex texture. The characteristic is similar to synchronizing embedding [11], which reveals the fact that executing the embedding changes in a group of adjacent pixels will likely have a smaller statistical impact than changing the same number of isolated pixels.

4. Performance Analysis and experimental results

Two parameters need to be computed for the comparison among these algorithms: The embedding rate α, which is the bits of embeddable message hidden in a cover pixel; The average distortion D_a, which is the expected number of changes for per pixel. We utilize the performance parameters of Hamming code over $\mathbb{Z}_4$ to compute the $\mathbb{Z}_4$-MPCs'. The two parameters α and D_a of $\mathbb{Z}_q$-MPC ($q = 4$) are defined as follows:

4.1 Embedding rate α and Average distortion D_a

The Embedding rate (embeddable bits/$(n_1 \times n_2)$) of a $[n_1, k_1]\otimes[n_2, k_2]$ $\mathbb{Z}_4$-MPC is computed by the follow formula:

$$\alpha = \frac{[r_2 \times (2^{r_1} - 1) + r_1 \times (2^{r_2} - 1 - r_2)] \times \lceil log_2 q \rceil}{(2^{r_1} - 1) \times (2^{r_2} - 1)}. \tag{4}$$

Let D denotes the expected number of changes, which includes two parts: the expected number of changes d_1 in all rows, the expected number of changes d_2 in the first k_2 columns. Besides, the expected number of changes of the compensation needn't be extra computed because the area of compensation embedding coincides with the area of row embedding, which is illustrated in **Fig. 3**. However, if the expected number of changes in rows is computed by Lemma 3, the expected number of changes of the compensation needs to be computed. The parameter D_a is computed as follows:

$$\frac{D}{(2^{r_1} - 1) \times (2^{r_2} - 1)}. \tag{5}$$

1) Compute parameter d_1

The total expected number of changes in all $(2^{r_1} - 1)$ rows based on Lemma 2 is d_1 which is defined as follows:

$$\frac{\sum_{i=1}^{r_2} i \times (q-1)^i \times C_{r_2}^i}{q^{r_2}} \times (2^{r_1} - 1). \tag{6}$$

When $r_1 = r_2 = 2$, $d_1 = \frac{3}{2} \times 3 = 4.5$. If the Lee Distance of Lemma 3 is utilized, $d_1 = \frac{1\times9+2\times6}{16} \times 3 = 3.93$ based on the Coset table of Hamming code [12] over $\mathbb{Z}_4$. The expected number of changes in rows can be decreased. The total expected number of changes in all rows is ${d_1}^{Lee}$,

$$\frac{\sum_{i=1}^{\rho-1} i \times (q-1)^i \times C_{2^{r_2}-1}^i + \rho \times (q^{r_2} - \sum_{j=0}^{\rho-1}(q-1)^j \times C_{2^{r_2}-1}^j)}{q^{r_2}} \times (2^{r_1} - 1). \tag{7}$$

Based on Lemma 6, $\rho = 2$ always holds.

2) Compute parameter d_2

Let d_{r_1} denotes the expected number of changes within the first r_1 rows in each column, and it is defined as follows:

$$\frac{\sum_{i=1}^{r_1} i \times (q-1)^i \times C_{r_1}^i}{q^{r_1}}. \tag{8}$$

Then d_2 is:

$$d_{r_1} \times \frac{1}{(r_1+1)} \times \left(\frac{2^{r_2}-1}{2^{r_2}} - \frac{1}{2^{r_2}} \times \frac{1}{q}\right) \times (2^{r_2}-1-r_2). \tag{9}$$

The occasions of each pixel to be changed locating in (i,j)th position ($j \in \{1,\dots,k_2\}$) during the column embedding is $\frac{1}{(r_1+1)}$. If there is no previously changed coordinate in (i,j)th position, this situation increases by one the amount of expected number of changes with the probability of $\frac{2^{r_2}-1}{2^{r_2}}$; If there is previously flipping in this coordinate, the encounter will make the changes counteracted, this situation decreases by one the amount of expected number of changes and happens in $\frac{1}{2^{r_2}} \times \frac{1}{q}$ occasions, and there are $2^{r_2}-1-r_2$ columns to be embedded. If the Step I of $\mathbb{Z}_4$-MPCs Algorithm is based on Lemma 2, because the changed positions are located within the last r_2 columns in row embedding procedure, we need not add extra computation for the compensation. Finally, the total expected number of changes is $D = d_1 + d_2$.

3) Compute parameter d_3

If the Step I of $\mathbb{Z}_4$-MPCs Algorithm is based on Lemma 3, the expected number of changes of the compensation within the first r_1 rows is d_3,

$$\left(\frac{\sum_{i=1}^{r_2} i \times (q-1)^i \times C_{r_2}^i}{q^{r_2}} - {d_1}^{Lee}\right) \times r_1. \tag{10}$$

Finally, the total expected number of changes is $D = d_1 + d_2 + d_3$. Summing this up, the average distortion of this proposed is $\frac{d_1+d_2}{(2^{r_1}-1)\times(2^{r_2}-1)}$ or $\frac{{d_1}^{Lee}+d_2+d_3}{(2^{r_1}-1)\times(2^{r_2}-1)}$, and the embedding rate is $\frac{[r_2\times(2^{r_1}-1)+r_1\times(2^{r_2}-1-r_2)]\times\lfloor log_2 q\rfloor}{(2^{r_1}-1)\times(2^{r_2}-1)}, q = 4$.

Comparison of performance has been made among the PPC method, F5-W-EMD method, Product Perfect $\mathbb{Z}_2\mathbb{Z}_4$-linear method ($\mathbb{Z}_2\mathbb{Z}_4$-PPC) and this $\mathbb{Z}_4$-MPC method. The optimal performance of one algorithm is that it obtains higher embedding rate with less distortion. There are also some parameter settings to be introduced among these methods: For the Product Perfect $\mathbb{Z}_2\mathbb{Z}_4$-linear method, if the parameter $= n \times N$, it denotes the smallest size of embeddable block. The code actually embeds $(n+1) \times m$ bits in $n \times N = n \times (n+1)/2$ symbols, and an upper bound for the average distortion is $\frac{2n+\frac{n-1}{2^{B-2}}}{(n+1)^2} \times (\frac{n+3}{\left(n+\frac{n-1}{2^{B-1}}\right)\times(n+1)} + 1)$, where B = 8 for 256 grayscale images. In a similar way, they repeat this method over and over and generalize the computations of the average distortion and the embedding rate by taking the code whose codewords are all the l-dimensional matrices, where $l \to n \times (n \times \dots n \times N), \xi = 0$, the performance is closer to the ideal optimality but the size gets infinity. Finally, the performance converges to a CI-rate of $(\frac{2n+\frac{n-1}{2^{B-2}}}{(n+1)^2} \times \left(\frac{1}{1-\xi}\right), \frac{mn}{N(n-1)})$. In **Fig. 5**, we suppose $\xi = 0$

which make the code converge to an ideal value; The F5-W-EMD method achieves the average distortion and embedding rate are $(\frac{n\frac{2^x-1}{2^x}+\frac{n(2^{x+1}3\times2^{x-1}-2)(2^{x-1}-1)}{(n+1)2^{x+1}}}{n(2^x-1)}, \frac{nx+(2^{x-1}-1)\left(log_2(n+1)+\frac{n}{n+1}\right)}{n(2^x-1)})$, and This method can be obtained the optimal performance when the size of block is $n \times (2^x - 1)$, $n = 2^x - 1$; For the $\mathbb{Z}_4$-MPC and PPC method, when we let $r_1 = r_2 = r, n = 2^r - 1$, the optimal performance can be obtained.

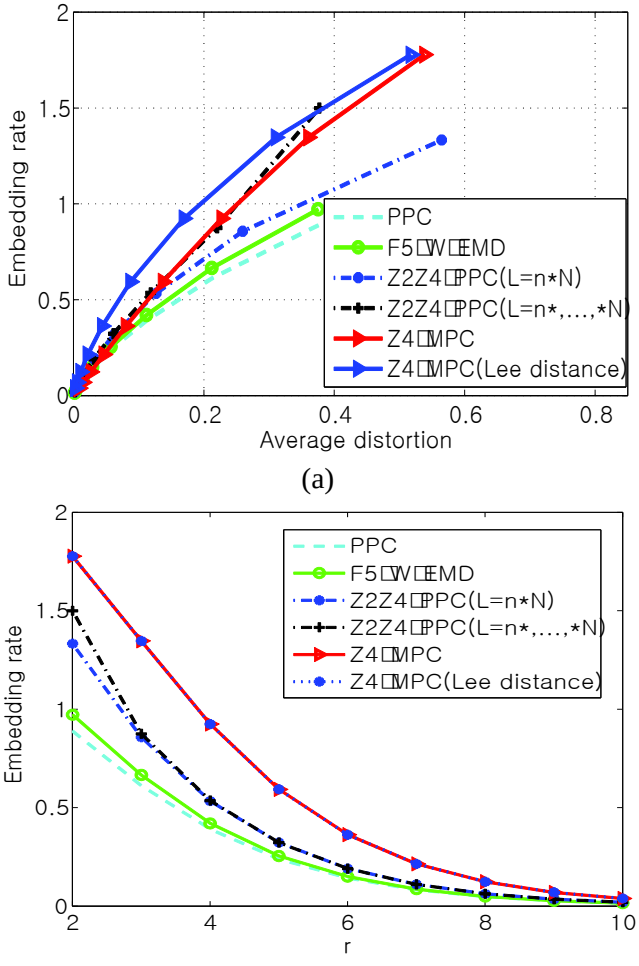


Fig. 5. Performance comparison among PPC, F_5-W-EMD, $\mathbb{Z}_2\mathbb{Z}_4$-PPC and $\mathbb{Z}_4$-MPC steganographic methods.

By using these parameter settings, the results of comparison between these algorithms are shown in **Fig. 5(a)**. The x axis represents the average distortion D_a, while the y axis denotes the embedding rate α. In **Fig. 5(b)**, the result shows the comparison of embedding rates among these algorithms. As seen from the figures, the performance of this proposed method has far exceeded the others. $\mathbb{Z}_4$-MPC (Lee distance, based on lemma 3) can obtain less average distortion than $\mathbb{Z}_4$ -MPC(based on lemma 2), but the former also obtains higher computatinal-complexity.

4.2 The Evaluation of Imperceptibility

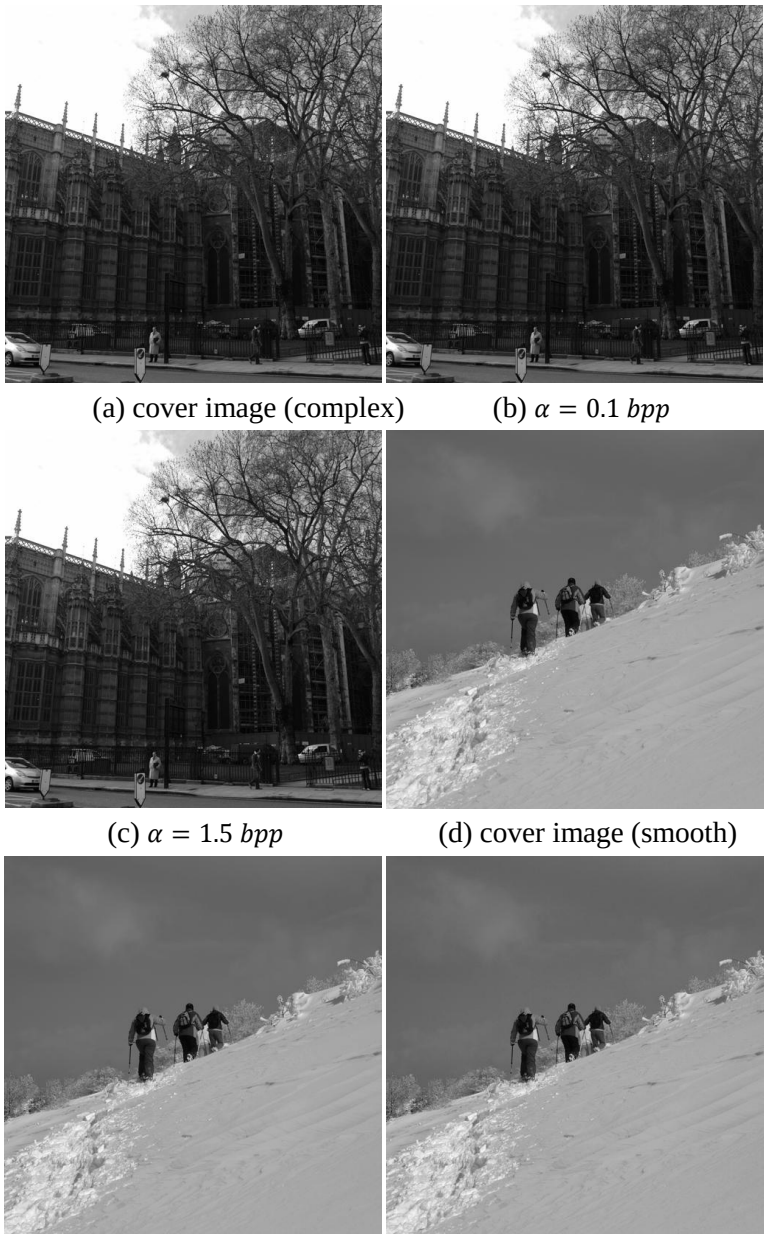
The method of objective evaluating of image quality are many, e.g., Mean Square error (MSE), Laplace MSE, Signal-Noise Rate (SNR) and Peak Signal-Noise Rate (PNSR). Up to now, the widely distortion metrics in image or audio encoding domain is PNSR. In this section, suppose a $M \times N$ image, the cover image is denoted as $p(x, y)$, and the stego image is denoted as $\hat{p}(x, y)$. The evaluation of mathematical expression is given as follow:

$$PNSR = 10\log\frac{M \times N \max_{M,N} p^2(x, y)}{\sum_{M,N}(p(x, y) - \hat{p}(x, y))^2}. \tag{11}$$

PNSR evaluates the approximate degree between raw image and processed image, and it is a statistically linear description which effects on vision quality. Two 512×512 raw images from BOSSBase 1.01 with different complex texture are selected to test their imperceptibility. Suppose payload rate (embeddable bits/cover size) $\alpha \in \{0.1,0.3,0.5,0.8,1.0,1.2,1.5,1.7\}$ *bpp*, and the code $\mathbb{Z}_4$-MPC ($r = 2$) is selected. The results of their corresponding PNSR are shown in **Table 1**, and their comparison charts between raw images and stego images are shown in **Fig. 6**.

Table 1. Comparison of PNSR between different payload rates

Texture	Payload Rate (*bpp*)	PNSR(*db*)
Complex	0.1	61.0396
	0.2	58.4147
	0.3	56.6739
	0.5	54.4426
	0.8	52.4167
	1.0	51.4525
	1.2	50.6667
	1.5	49.6903
	1.7	48.6297
Smooth	0.1	60.5038
	0.2	57.4886
	0.3	55.7213
	0.5	53.5231
	0.8	51.4668
	1.0	50.4904
	1.2	49.6945
	1.5	48.7276
	1.7	48.1966



(a) cover image (complex) (b) $\alpha = 0.1\ bpp$

(c) $\alpha = 1.5\ bpp$ (d) cover image (smooth)

(e) $\alpha = 0.1\ bpp$ (f) $\alpha = 1.5\ bpp$

Fig. 6. Comparision charts of imperceptibility with different texture features and different payload rates before and after embedding message.

Generally, the bigger PNSR the higher quality of reconstructed image. When PSNR is higher than 28 *db*, there is no significant difference of image quality. When PSNR is higher than 40 db, the naked eye couldn't tell the difference and the two compared images are considered similar. As seen from the results of testing data shown in **Table 1** we conclude that the values of PNSR are almost belong to 61~48 *db*. With the increase in payload the PSNR is decline, when the payload rate reaches 1.5 *bpp* , the value is around 50 *db*. The smooth image is also selected to test the imperceptibility. As seen from the results, the value of PNSR of the smooth image is lower than the complex texture image, and the rate of decline is less than

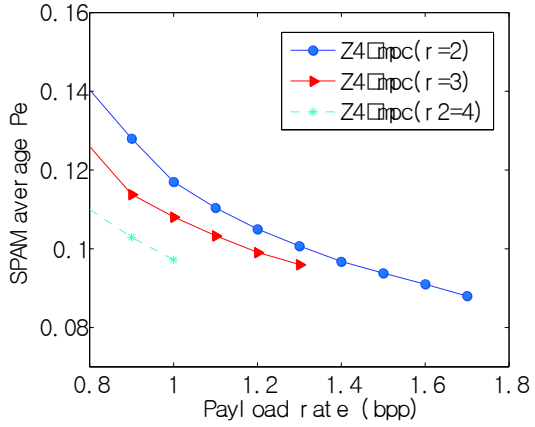
0.5%. We can see from the figures shown in **Fig. 6** that the image quality between raw image and processed image are almost the same. With the increase in payload the decline of image quality is small, specially, the image quality is hardly impact when complex texture changes. Some conclusions are given that the proposed codes can be utilized not only in image steganography domain but also in high-capacity covert communication domain, e. g., real-time voice NFC or other multimedia data hiding domain, and the further research should be enhanced.

4.3 The Testing of Superiority of The Cluster Structure in image steganography

In the results of our experiments and the research of secure estimated payload [22][23] in image steganography domain, the payload rate should not too big because of considering security. However, our proposed codes has the characteristic of large payload which is more specifically suited to application in real time NFC domain. The reason for proposing combining this code with cost function intends to discuss the superiority of the cluster structure. In order to prove the superiority, HUGO+ $\mathbb{Z}_4$-MPCs $(2 \leq r \leq 4)$ are selected to test the performance. In our experiments, 10000 images are obtained from the BOSSbase 1.01. We randomly choose 5000 of them for the training purposes and the left 5000 images are prepared for testing. As a feature set, we used the SPAM features [16], the ensemble classifier are trained by fusing decisions of weak base learners trained on subsets of the feature space. The testing error is $\overline{P_e}$ which is defined as follows:

$$\overline{P_e} = \min_{P_{FA}} \frac{1}{2}\left(P_{FA} + P_{MD}(P_{FA})\right), \tag{12}$$

where P_{FA} and P_{MD} are false alarm rate and missing alarm rate respectively. When payload rate $\alpha \in \{0.8,0.9,1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.7\}$ *bpp*, the results of average testing errors are shown in **Fig. 6.** As seen in **Fig. 6**, the security performance of this proposed code with $r = 2$ can obtain the optimal performance. The results show that when the dimension of the code gets smaller, the security performance is more improved because small dimension codes have better cluster performance. When the payload rate increases, the security performance significantly declined. If the payload rate is larger than 1.0 *bpp*, the changes are almost distributed in the whole region. Then the performance of these codes tends to slightly decrease.



(a)

Fig. 6. The comparison of security of various adaptive steganographic schemes with $0.8 \leq \alpha \leq 1.7$ *bpp* and $r \leq 4$ by means of average error Pe of steganalyzers utilizing SPAM features.

5. Conclusion

Although the adaptive steganography technique with STC has been widely used in some store mediums, i.e., audios, videos, images, some domains such as NFC in acoustic communication without store mediums has urgently needed some high-capacity steganographic schemes for covert communicating secret messages. Our proposed codes obviously have the characteristic of large embedding rate and can be easily used in covert communication or data hiding domain because of the low computational-complexity. However, we also have a further research for testing the performance of codes with large embedding rate by combining the codes with adaptive scheme [7][8]. The result is that this proposed method can be more security when the code dimension getting smaller. But when the payload rate increases, the security performance significantly declined. The test of this proposed only verifies the superiority of clustering performance of codes with large embedding rates. In literature [24], it is shown that the distortion measure seems to fail to capture the statistical detectable correctly and the algorithms are more detectable for large payload. Given all this, the proposed codes with large embedding rate are not qualified to being used in application of steganography for storage media. But steganography for real time non-storage media should be considered to be applied because of the high-capacity and optimal imperceptibility, and the proposed method is expected to further research in the later work.

References

[1] A. Westfeld, "F5-a steganographic algorithm: High capacity despite better steganalysis," in *Proc. of Information Hiding, 4th International Workshop I. S*, Moskowitz, Ed. New York, Springer-Verlag, vol. 2137, Lecture Notes Comput. Sci., pp. 289-302, 2001. Article (CrossRef Link)

[2] J. Fridrich, and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Inf. Security Forensics*, vol. 1, no. 3, pp. 390–394, 2006. Article (CrossRef Link)

[3] R.Y. Zhang, V. Sachnev, and H. J. Kim, "Fast BCH syndrome coding for steganography," in *Proc. of Information Hiding*, Darmstadt, Germany, pp. 48–58, 2009. Article (CrossRef Link)

[4] V. Sachnev, and H. J. Kim, "Modified BCH data hiding scheme for JPEG steganography," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 89-98, 2012.

[5] H. Rifà-Pous, and J. Rifà, "Product perfect codes and steganography," *Digital Signal Processing*, vol. 19, no 2009, pp. 764-769, 2009. Article (CrossRef Link)

[6] R. Crandall, "Some Notes on Steganography," *Posed on Steganography Mailing List*, http://os.inf.tu-dresden.de/~westfeld/cranda ll.pdf, 1998. Article (CrossRef Link)

[7] T. Pevn, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. of Information Hiding, Lecture Notes in Computer Science, R. Bhme, P. Fong, and R. SafaviNaini, Eds. Springer Berlin / Heidelberg*, pp. 161–177, 2010. Article (CrossRef Link)

[8] V. Holub, J. Fridrich, "Digital Image Steganography Using Universal Distortion," in *Proc. of 1st IH&MMSec Workshop*, Montpellier, France, pp.59-68, June 17-19, 2013. Article (CrossRef Link)

[9] J. Rifà and L. Ronquillo, "Product Perfect $\mathbb{Z}_2\mathbb{Z}_4$-linear codes in Steganography," in *Proc. of International Symposium on Information Theory and its Applications (ISITA)*, Taiwan, pp. 696-701, 2010. Article (CrossRef Link)

[10] Z. Zhao and F. Gao, "An improved steganographic method of product perfect codes," in *Proc. of Signal Processing, Communication and Computing (ICSPCC)*, Xian, pp. 1-5, 2011. Article (CrossRef Link)

[11] T. Denemark and J. Fridrich, "Improving Steganographic Security by Synchronizing the Selection Channel," in *Proc. of 3rd IH&MMSec. Workshop*, Portland, Oregon, pp. 5-14, 2015. Article (CrossRef Link)
[12] Zuyun Fu, *Information Theory*, Publishing House of Electronics Industry, China, 2007.
[13] X. Zhang and S. Wang, "Efficient Steganographic Embedding by Exploiting Modification Direction," *IEEE Communications Letters*, vol. 10, no. 11, pp. 781-783, 2006. Article (CrossRef Link)
[14] W. M. Zhang, X. P. Zhang and S. Z. Wang, "A Double Layered "Plus-Minus One Data Embedding Scheme," *IEEE Signal Processing Letters*, vol. 14, no. 11, pp.848-851, 2007. Article (CrossRef Link)
[15] J. Fridrich, M. Goljan, P. Lisonek, D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Process*, vol. 53, no. 10, pp. 3923-3953, 2005. Article (CrossRef Link)
[16] Pevny T, Bas P, Fridrich J, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp.215-224, 2010. Article (CrossRef Link)
[17] Xinpeng Zhang and Shouzhong Wang, "Dynamical Running Coding in Digital Steganography," *IEEE Signal Processing Letters*, vol. 13, no. 3, pp. 165-168, 2006. Article (CrossRef Link)
[18] C. Qin, C. C. Chang, Y. H. Huang, and L. T. Liao, "An Inpainting-Assisted Reversible Steganographic Scheme Using a Histogram Shifting Mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp.1109-1118, 2013. Article (CrossRef Link)
[19] C. Qin, C. C. Chang and T. J. Hsu, "Reversible Data Hiding Scheme Based on Exploiting Modification Direction with Two Steganographic Images," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5861-5872, 2015. Article (CrossRef Link)
[20] Zhou Peng, Nadeem T, Kang P, *et al*, "EZCab: A cab booking application using short-range wireless communication," in *Proc of IEEE Int Conf Pervasive Computing and Communications (PerCom)*, Piscataway, NJ:IEEE, pp.27-38, 2015. Article (CrossRef Link)
[21] Li Chuan, Hutchins D A, Green R J, "Short-range ultrasonic digital communications in air," *IEEE Trans on Ultrasonic, Ferroelectrics and Frequency Control*, vol. 55, no. 4, pp. 908-918, 2008. Article (CrossRef Link)
[22] T. Filler, A. Ker and J. Fridrich, "The square root law of steganographic capacity for markov covers," in *Proc. of Media Forensics and security XI*, pp. 801-811, 2009. Article (CrossRef Link)
[23] Lingyu Zhang, Diao Chen, Yun Cao and Xianfeng Zhao, "A Practical Method to Determine Achievable Rates for Secure Steganography," in *Proc. of 17th IEEE International Symposium on Cyberspace Safety and Security (CSS)*, NewYork, USA, pp.1274-1281, 2015. Article (CrossRef Link)
[24] Tomas Filler, Jan Judas, Jessica Fridrich, "Minimizing Additive Distortion in Steganography using Syndrome-Trellis Codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920-935, 2011. Article (CrossRef Link)

Lingyu Zhang her B.E. degree in School of Computer and Communication Engineering from Liaoning Shihua University, Fushun, liaoning, P. R. China, in June 2007. She is currently pursuing a doctoral degree at the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. Her current research interests are information hiding including steganography and steganalysis, multimedia security and network security etc.

Deyuan Chen is an associate professor at University of the Chinese Academy of Sciences, Beijing, P. R. China. He received both his B.E. and PhD degree from University of the Chinese Academy of Sciences, Beijing, P. R. China. His current research interests include joint channel encryption coding, channel coding and image coding. He currently also serves as a reviewer for several journals including *Chinese Journal of Electronics* and *Chinese Journal of Computers.*