Relationship between Correcting Code and Module Technique in Hiding Secret Data

Phan Huy, Nguyen Thanh, Cheonsick Kim

Relationship between Correcting Code and Module Technique in Hiding Secret Data

Phan Trung Huy [1], Nguyen Hai Thanh [2], Cheonsick Kim [3]

[1] Hanoi University of Science and Technology, Hanoi, Vietnam
huypt-fami@mail.hut.edu.vn,huyfr2002@yahoo.com
[2] Ministry of Education and Training, Hanoi, Vietnam
nhthanh@moet.gov.vn
[3] Dept. of Computer Engineering Sejong University,
98 Gunja-Dong,Gwangjin-Gu, Seoul 143-747, Korea
mipsan@paran.com

Abstract. In this paper, we show the role of modules over rings of finite characteristics in data hiding area. Applications of correcting codes and covering functions in data hiding are shown as special cases of our module approach. Applications of modules over rings of characteristic 2 to design new embedding schemes for hiding secret data in binary images are introduced.

Keywords: module, ring, characteristic 2, data hiding, binary image, steganography, correcting codes, covering function, MSDR.

1 Introduction

Data hiding can be applied in copyright, annotation, and communication, and can be achieved by altering some nonessential pixels in the cover image. For example, in a given color image (including grayscale image), the least-significant bit (LSB) of each pixel can be changed to embed the secret data. However, two color images embedded by secret data are very sensitive and can be easily detected by the human eyes. One of the most challenging problems is hiding the secret data into binary images with a high ratio of secret data, and low image distortion. In case of palette images, ones need to prevent steganalysis, especially to histogram-based attacks (see for examples some analysis in [18], if the alpha ratio of the number of changed pixels to the number of total pixels of a given palette image is lower than 0.1, it is very difficult to guess if the image contains hidden data. In block-based approaches, each binary image is partitioned into binary blocks of the same size N, each block can be seen as an N-bit string of size N. In such a block F of size N, by taking WL scheme [16] one can embed one bit by changing at most one bit of F. From CPT scheme proposed by Chen-Pan-Tseng (2000) [3], in F one can embed $r=\lfloor \log_2(N+1) \rfloor$ bits, by changing at most two bits of F. By correcting codes approach, with notion of covering codes, Crandall [5] refers to an unpublished article by Bierbrauer [1] that brings deep tech-

* This work is partially supported by Vietnamese National Foundation for Science & Technology Development (NAFOSTED)

niques to data hiding area, based on the point of view of a coding theorist [10,12,13]. The connection between linear covering codes and steganography had also appeared in one paper by Galand and Kabatiansky [6], and covering codes in [13] dating back 1994 by R. Struik. In a nice paper [7] written by J. Bierbrauer and J. Fridrich, the authors described and extended the original Bierbrauer's work[1] which show rich contributions of covering functions to data hiding area. As shown in their paper, in binary images or in binary data formed from LSB planes of three component colors Red, Green, Blue of pixels, in true color images, some schemes have reached the maximum secret data ratio ($MSDR_k$) based on the number of secret bits which can be embedded in a block F of N pixels with the restrictions: in F, at most k bits can be changed (k=1,2,3). Several works [2, 4, 5, 6, 7, 9, 14, 15] introduced the powerful applications of approaches based on correcting codes in data hiding.

In this paper, we introduced an application of modules over rings of characteristic 2 to data hiding area, for binary images the main case of our interest. It can be seen that this idea will be easily extended to others characteristics for different formats on multimedia environment. The relationship between two methods in data hiding, by module and by correcting codes methods is considered. It is shown that hiding secret data based on correcting codes is the special case of hiding data by module over rings of characteristic 2. Some new schemes for data hiding by module method are introduced, showing the advantage and flexible of module approach to data hiding area.

The paper is divided into 5 sections. Following the introduction section, section 2 recalls applications of linear correcting codes and covering codes in binary data hiding. Due to [11] we recall the notion "k-maximal secret data ratio" ($MSDR_k$) of secret bits embedded in each block F of N pixels in binary images with the restriction that at most k –pixels can be changed in F. As shown, the results of covering codes which reached these limits. In section 3 we focus on modules over the ring $\mathbf{Z}_2$ of integers modulo 2, the main subject of this paper. Notions of k- base on module and k-embedding scheme are introduced. Some aspects of *MSDR* are considered. To get new 2-bases of $\mathbf{Z}_2$-modules for new 2-embedding schemes applied to binary images, a designing method is given. In section 4 we show the relationship between correcting codes, covering function with module method, and present in details the arguments via an example of using Hamming code (7,4) in data hiding. As experimental results, some 2-bases obtained by our program are presented and their applications in data hiding schemes are discussed. Conclusion is the content of section 5. The general applications of modules over the rings of characteristics q, q>1, specially for the special case of ring $\mathbf{Z}_q$ of integers modulo q, are discussed for future study.

2 Error correcting codes and covering codes

2.1 Correcting codes in steganography

As shown in the survey work [7] of J. Bierbrauer and J. Fridrich, the covering function technique is originated from error correcting codes area and this brings to steganography powerful ways to design high quality schemes for embedding secret

data. For examples, the matrix encoding technique used in F5 algorithm by Westfeld [15] permit us to modify at most 1 pixel among 2^k-1 pixels to hide k secret bits. The distortion of image then is reduced with the high ratio of the embedding scheme, which reached $MSDR_1$. Matrix encoding technique using correcting codes show that after the embedding phase, the syndromes generated by a parity check matrix in the extracting phase will present exactly secret data embedded in stego-images. In covering code technique, the covering radius of the code reflects the maximal number of pixels changed to embed secret data and the dimension of code reflects the capacity of bits can be embedded, by a scheme used this approach.

Given the linear space $W_n = \mathbf{F}^n_q$ of dimension n over the finite field $\mathbf{F}_q$ and a natural number $d > 1$. Due to Hamming's work origined from 1947, an error correcting code over $\mathbf{F}_q$ is defined as a subset $C \subseteq W_n$ for which d is the minimum distance between two distinct code words $x \neq y \in C$. In error correcting codes area, each codeword $v \in C$ can be seen as an (exact) encoded bit string we need to transmit on a noise channel. If there are at most $r = \lfloor d/2 \rfloor$ errors appear on transmission, that is v is changed to $v'=v+e$ for some error vector e whose Hamming weight $w(e) \leq r$, then we can correct v' to recover v after erasing e, by some methods detecting e. The size of C is defined as $|C|$- the number of code words in C.

If the code is linear, we can find some effective ways to detect the error vector e.

A linear code C of type $[n, k, d]$ is a linear subspace of W_n having its dimension $k < n$, with d the minimum (Hamming) distance between distinct code words $x \neq y \in C$. The covering radius ρ of code C is defined as $\rho = max_{v \in \mathbf{W}} d(v;C)$, where $d(v;C)$ denotes the minimum Hamming distance from the vector v to the code C. A parity check matrix of C is a matrix H *of size* $t \times n$, $t=n-k$, which permit us to obtain for any $v \in W_n$ its syndrome vector $s(v) = H.v^T$, where v^T is the column vector form of v, so that $s(v)= 0$ if and only if $v \in C$. The syndrome $s(v)$ is used to detect and correct the error e if $w(e)=k$ (the number of errors), $k \leq r = \lfloor d/2 \rfloor$, appeared on transmission, to recover the correct message v *from the received message* $v'=v+e$ as follows: since $s(v')=H.v'=H.(v+e)^T=H.v^T+H.e^T=s(v)+s(e)=s(e)$, we find e so that $s(e) = s(v')$ and $w(e) \leq r = \lfloor d/2 \rfloor$, and correct v' to $v = v'-e$. To obtain an efficient way recognizing the error e, *the coset technique* is developed.

A coset $C + v$ is the set of all vectors in W_n with the same syndrome $s(v)$. A vector $l_{r(v)}$ of the minimum weight in $C + v$ can be called a *leader of the coset*. We define the syndrome map $s : W_n \rightarrow W_t$ by $s(v) = H . v^T$.

In steganography (see [7, 9]), one can make use of the syndrome map $s : W_n \rightarrow W_t$ as the extracting map of the embedding scheme $[n, t, \rho]$ which can be defined by:

1. Let n and t be positive integers, $t \leq n$, and let X be a finite set of symbols. An embedding scheme of type $[n, t, \rho]$ over X is given by a pair of maps $c : X^t \times X^n \rightarrow X^n$ and $s : X^n \rightarrow X^t$ such that $s(c(p, v)) = p$ for all (plaintext) $p \in X^t$ and v *(stego-block)* $\in X^n$. Maps c and s are the embedding and the extracting maps, respectively. The covering radius of the scheme is defined as $\rho = \max\{d(v, c(p, v)) \mid p \in X^t, v \in X^n\}$, where d is the Hamming distance.

2. The embedding scheme $[n, t, \rho]$ allows us to hide p (as a string of t secret symbols) into v (as an n-string v of n cover symbols), by changing at most ρ *of* n cover symbols. The following algorithm shows this idea.

Coset algorithm. *Given p,v*

a) Compute $u = s(v) - p$,

b) Set $v' = c(p; v) = v - l_u$, where l_u is a leader of the coset $C + u$ of all the vectors in W_n with the same syndrome of u. The Hamming weigh $w(l_u) = j \le \rho$ together with vector l_u show the exact j positions on which we need to change with the cover vector v by equation $v' = v - l_u$. So, $s(l_u) = u$. This provides us in the extract phase: by using (*a*) above, the syndrome $s(v') = s(v) - s(l_u) = s(v) - u = p$ is obtained, the exact plaintext as claimed.

Concretely (see, such as [7]), in case $X = \mathbf{F}_2$ (also $\mathbf{Z}_2$) for designing an embedding scheme $[n, t, \rho]$, we can use a covering function $COV(\rho,n,t)$ which permit us to embed any t-bit string p in any n-bit string v (as a block of n bits) by changing at most ρ positions on v.

Good covering codes

In [7] the authors show some interest classes of cover functions:

1. $COV(2, 5 \cdot 2^{a-1} - 1, 2a + 1)$ for $a \ge 1$ by equation (1).

2. $COV(2, 6 \cdot 4^{a-1} - 1, 4a)$, $a \ge 2$ by equation (4). The first members of this family are COV (2, 23, 8), COV (2, 95, 12).

By our interest, for high quality of stego-images, in this paper we focus only on small values, especially for $\rho = 1, 2$.

2.2 k-maximal secret data ratio of embedded bits

In this part, given an image G, for simplicity we only concentrate on one fixed block $F = (F_1, F_2, .., F_N)$ of N pixels of G, and F is considered as a vector of dimension N). In F each entry F_i can be understood as a pixel whenever the index i is referring, also as the color of this pixel whenever its color value is mentioned. Suppose each color F_i can be changed to q-1 new colors F_i' closed to F_i, by one of q-1 different ways. In the case of binary images, $q=2$. In the general case $q \ge 2$ for color images.

We consider here *k-embedding schemes* in which secret bits can be embedded in each block F by changing at most k entries, with k small, $k = 1,2$. Together with F, each new block F' after changing pixels of F is called a *configuration*. Denote by $MSDR_k$ the *k-Maximal Secret Data Ratio* which presents the *largest number of embedded bits in each block F of N pixels by changing colors of at most k pixels in F.*

In the case $k=1$, since we change colors in at most one element in F, with N elements, $1+(q-1)N$ ways can be taken. This means that for any 1-embedding scheme, we can hide at most

$$MSDR_1 = \lfloor \log_2(1+(q-1).C(N,1)) \rfloor \text{ secret bits in each block } F.$$

In the cases $k=2$ or 3, similarly, we have

$$MSDR_2 = \lfloor \log_2(1+(q-1).C(N,1)+(q-1)^2.C(N,2)) \rfloor \text{ and}$$

$$MSDR_3 = \lfloor \log_2(1+(q-1).C(N,1)+(q-1)^2.C(N,2)+(q-1)^3.C(N,3)) \rfloor .$$

For example, in binary images, if $N = 5$ then $MSDR_2 = 4$. In the case of grey images, with $N=1$, $q=16$, $MSDR_1 = \lfloor \log_2(1+15.1) \rfloor = 4$. That means in each pixel F (a block of 1 pixel) if its color has 15 other ways to change, then any 4 secret bits can be hidden in F by some appropriate change. This is the case achieved in [8].

3 Modules over rings of characteristic 2 in hiding secret data

3.1 Application of modules in hiding secret data

Each (right) module M over the ring $\mathbf{Z}_q$ is an additive abelian group M with zero 0 together with a scalar multiplication "." to assign each couple $(m,\boldsymbol{t})$ in $M \times \mathbf{Z}_q$ with an element $m.\boldsymbol{t}$ in M. Let $\mathbf{Z}_q = \{\mathbf{0},\mathbf{1},..,\boldsymbol{q}\text{-}\mathbf{1}\}$. We need some following basic properties, which will be used in the sequent:

P1) $m.\mathbf{0} = 0$; $m.\mathbf{1}=m$;

P2) $m+n = n+m$ for all m,n in M.

P3) $m.(\boldsymbol{t}+\boldsymbol{l}) = m.\boldsymbol{t} + m.\boldsymbol{l}$ for all m in M, $\boldsymbol{t},\boldsymbol{l}$ in $\mathbf{Z}_q$.

Definition 1. Given a natural number $v > 0$, a subset $U \subseteq M\text{-}\{0\}$, we call U *a v-base of M* if for any $x \in M - \{0\}$, x can be presented by a linear combination of at most v elements in U. That means there exist n elements $u_1,u_2,..,u_n$ in U, $n \le v$, together with $\boldsymbol{t}_1,\boldsymbol{t}_2,..,\boldsymbol{t}_n$ in $\mathbf{Z}_q$ such that $x = u_1.\boldsymbol{t}_1+u_2.\boldsymbol{t}_2+..+u_n.\boldsymbol{t}_n$.

We call *a k-embedding scheme* any embedding scheme that permits in each block F of N elements ones can change at most k elements to hide data. In the case v=1, it is obvious that $U=M\text{-}\{0\}$ is the unique 1-base.

In this paper, the main of our interest are the cases v = 1,2 for binary images (according to the characteristic q =2 and $M = \mathbf{Z}_2 \times \mathbf{Z}_2 \times .. \times \mathbf{Z}_2$ is the n-fold cartesian product of $\mathbf{Z}_2$, which can be seen as a (right) $\mathbf{Z}_2$-module. For binary image we have q=2. The addition in $\mathbf{Z}_2$ can be seen as the operation XOR (exclusive –OR) on bits, Each element $x=(x_1,x_2,..,x_n)$ in M can be presented as an n-bit string $x=x_1x_2..x_n$, with operations defined as follows:

D1) $x+y = z_1z_2..z_n$ where $z_i=x_i \oplus y_i$, i=1,..,n , For $x=x_1x_2..x_n$, $y=y_1..y_n$ in M, $\boldsymbol{k}$ in $\mathbf{Z}_2$,

D2) $x.\boldsymbol{k}= z_1z_2..z_n$ where $z_i=x_i.\boldsymbol{k} = x_i$ AND $\boldsymbol{k}$.

Given a binary image G, we set $C_G = \mathbf{Z}_2 = \{\mathbf{0},\mathbf{1}\}$ as the set of two colors of G. The *color changing function* Next: $\mathbf{Z}_2 \rightarrow \mathbf{Z}_2$ is given by:

(3.1) $\mathbf{c}' = \boldsymbol{c}+\mathbf{1} =$ Next(c), for all $\boldsymbol{c}$ in $\mathbf{Z}_2$ and *changing a color* $\boldsymbol{c}$ means that $\boldsymbol{c}$ is replaced by $\boldsymbol{c}'$ = Next($\boldsymbol{c}$).

As one can see, an 1-embedding scheme can be reduced from a 2-embedding scheme. Hence, at first we consider 2-embedding schemes.

3.2 Application of 2-bases for 2-embedding schemes

Let $U \subseteq M\text{-}\{0\}$, U be a 2-base of a $\mathbf{Z}_2$ –module M. Suppose $|U| = n$. Consider any binary block $F = (F_1, F_2,.., F_s)$ of a given binary image G, and binary secret key $K = (K_1,K_2,..,K_s)$, $F_i, K_i \in Z_2$, i=1, .., N.

Suppose $s \ge n$. For F, we can assign a surjective function $h_F : \{1,2,..,N\} \rightarrow U$ as a weight function of indexes i of F_i. Since F is fixed in the scope, for simplicity we

write h instead. We can embed any secret element $d \in M$ in F *by changing colors of at most 2 elements* in F as the following 2-embedding scheme.

3.2.1 Embedding a secret element d

Set $S[F,K] = \sum_{1 \le i \le N} h(i).T_i$, by taking operations on the $\mathbf{Z}_2$ – right module M.

Step 0) Given a secret key as a binary vector $K=(k_0, k_1,.., k_N)$, $k_i \in \mathbf{Z}_2$.

Change the color F_i of each $F_i \in F$ into a *marked color* $T_i=F_i+k_i$ (in $\mathbf{Z}_2$).

We present this computation by $T=F \oplus K$;

Step 1) Compute $m = S[F,K]$;

Step 2) Compare m and d:

- Case $m = d$: keep F intact;

- Case $d \neq m$: then find $d - m = a$, for some $a \in M-\{0\}$. There are two following cases happen:

i) $a \in U$: since h is surjective, there exists F_q in F , such that $h(q) = a = d-m$. Then change the color F_q to new color F_q'=Next(F_q)=F_q+**1**.

ii) $a \notin U$: Since U is a 2-base of M - a $\mathbf{Z}_2$-module - we can find (successfully) two elements x,y in U *such that* $a=x+y$, and therefore find two entries F_p,F_q in $\boldsymbol{F}$ such that $h(p)=x$, $h(q)=$y.

Then we change F_p to new color F_p'=F_p+**1**, and change F_q to new color F_q'=F_q+**1**;

3.2.2 Extracting the secret element embedded in F.

Step 1) Computing $u=S[F,K]$;

Step 2) Return u as the secret element d embedded in S (that is $u=d$) .

Correctness of the method.

Theorem 1. *The element u extracted in step 1 of the extracting stage 3.2.2 above is exactly the secret element d hidden into S in the embedding stage 3.2.1.*

Proof. We need consider only the case $d \neq m$ and prove that $u=d$.

Indeed, if the step (2i) in **3.2.1** is taken place, after changing the color F_q to F_q' =F_q+**1** by step 2(ii) in **3.2.1** with $h(q)=a=d-m$**,** we get T_q'= T_q+**1.** Then,

$u = \sum_{1 \le q \neq i \le N} h(i).T_i + h(q).(T_q+\mathbf{1}) = \sum_{1 \le q \neq i \le N} h(i).T_i + h(q).(T_q) + h(q).\mathbf{1}$

$= \sum_{1 \le i \le N} h(i).T_i + h(q) = m+d-m = d.$

For the case that the step (2ii) in **3.2.1** is taken place, using $d - m = a = x + y$, $h(p)=x$, $h(q)=y$, by the same arguments we deduce T_p'=T_p+**1** and T_q'=T_q+**1**. Therefore by properties of modules

$u = \sum_{1 \le p,q \neq i \le N} h(i).T_i + h(p).T_p' + h(q).T_q = \sum_{1 \le p,q \neq i \le N} h(i).T_i + h(p).T_p + h(p) + h(q).T_q + h(q)$

$= \sum_{1 \le i \le N} h(i).T_i + h(p).T_p + h(q).T_q + h(p) + h(q)$

$= \sum_{1 \le i \le N} h(i).T_i + h(p) + h(q) = m + a = m + d - m = d.$ This completes the proof. ||

Example 1. The subset U={0001,0010,0100,1000,1111} is a 2-base of the module

$M = \mathbf{Z}_2 \times \mathbf{Z}_2 \times \mathbf{Z}_2 \times \mathbf{Z}_2$. Therefore, we can use it to hide data. In any block F of 5 pixels, we can change at most two pixels to hide 4 bits. That is the $MSDR_2$ is obtained: $MSDR_2 = \lfloor \log_2(1+5(5+1)/2) \rfloor = 4 = \lfloor \log_2(|M|) \rfloor$.

Remark 1. Applications of k -bases for k-embedding schemes are similarly established, for any $k>0$, hence we do not mention in details.

3.3 Designing 2 – bases of $\mathbf{Z}_2^n$

In this part we introduce a method to design 2-bases of $\mathbf{Z}_2^n$ inductively.

Denote by $V_n = \mathbf{Z}_2^n = \mathbf{Z}_2 \times \mathbf{Z}_2 \times .. \times \mathbf{Z}_2$ the $\mathbf{Z}_2$- module whose elements can be presented simply as the form $b = b_n b_{n-1}..b_1$, an n- bit strings.

Denote by $PR_k(V_n)$ the projection getting k right components in V_n and $PL_k(V_n)$ the projection getting k left components in V_n.

Concretely, $PR_k(b_n b_{n-1}..b_r..b_1) = b_k b_{k-1}..b_1$ and $PL_k(b_n b_{n-1}..b_2 b_1) = b_n b_{n-1}.. b_{n-k+2} b_{n-k+1}$.

Denote by $CL_{k,t}$ a class of 2-bases X of V_n satisfying $|X| = 2^k + 2^t - 3, n = k+t$, and $PL_k(X) = \mathbf{V}_k$.

Denote by $CR_{k,t}$ a class of 2-bases X of V_n satisfying $|X| = 2^k + 2^t - 3, n = k+t$, and $PR_t(X) = \mathbf{V}_t$.

Lemma 2. If X is a *2-base of* V_n such that $PL_m(X) = V_m$ for some integer $0 < m < n$, then there exist a *2-base* Y *of* V_{n+1} such that $PL_m(Y) = V_m$ *and* $|Y| = |X| + 2^r$, $m+r=n$.

Proof. Define Y is the set of all n+1-bit string x in one of two following forms:

(*i*) $b_n b_{n-1}..b_1 0$, with any $b_n b_{n-1}..b_1$ in X *and* whose right most bit is 0.

(*ii*) $00..0 x_r x_{r-1}..x_1 1$ with any $x_r x_{r-1}..x_1 \in V_r$.

By assumption on X, any element of the form $y_1..y_m 0$ in V_{n+1} can be presented as a linear combination of at most two elements in Y. For any element of the form $y_1..y_m x_1..x_r 1$ in V_{n+1}, by $PL_m(X) = V_m$ there exists n-bit string of X of the form $y_1..y_m v_1..v_r$, for some r-bit string $v_1..v_r$. So $y_1..y_m v_1..v_r 0$ belongs to Y by definition.

Define $u_1..u_r = x_1..x_r \oplus v_1..v_r$. Then $y_1..y_m x_1..x_r 1 = 00..0 u_1..u_r 1 \oplus y_1..y_l v_1..v_r$. Hence Y is a 2-base of V_{n+1} . Obviously, $|Y| = |X| + 2^r$, $m+r=n$ and $PL_m(Y) = V_m$. ||

Remark 2. By duality, from a 2-base X satisfying $PR_r(X) = V_r$, we can define a 2-base $\mathbf{Z}$ of V_{n+1} having all elements in one of two forms: $1 y_1..y_m 00..0$, with any m-bit string $y_1..y_m$ in V_m , and $0 x_1..x_m u_1..u_r$, with any n-bit string $x_1..x_m u_1..u_r$ in X.

Theorem 3. *For any $n \geq 4$, there exist 2-bases X in $CL_{m,r}$, Y in $CR_{m,,r}$ such that if $m,r > 1$, $m+r=n$ then $|X|, |Y| \leq 2^m + 2^r - 3$.*

Proof. Firstly, one can see that:

a) $CL_{2,2}$ contains the set $Z = \{0001, 0010, 0100, 1110, 1001\}$, this set satisfies the claim $|Z| = 5 = 2^2 + 2^2 - 3$, and $CR_{2,2}$ contains the set $T = \{1000, 0100, 0010, 0111, 1001\}$ with $|T| = 5$ which satisfies the claim.

b) For all $CL_{m,r}$, $CR_{m,r}$, $m+r=n$, $n \geq 4$, m, $r > 1$, one can prove easily the theorem by induction on n, starting from two 2-bases Z, T above and applying Lemma 2 together with Remark 2 by duality. □

Example 1.

a) the class $CL_{2,1}$ contains the set X={110,100,010,001}, and $CR_{1,2}$ contains the set Y = {011,001,010,100}.

Generally, for any n>0, we can define the set X contains all n+1- bit strings in one of two forms: $b_nb_{n-1}..b_10 \neq 00..0$ and 00..01. Then X belongs to $CL_{n,1}$. We define the set Y contains all n+1- bit strings of two forms: $0b_nb_{n-1}..b_1 \neq 00..0$ and 100..0, then Y belongs to $CL_{n,1}$.

b) $CL_{2,3}$ contains the set of 9 elements X = {00010, 00100, 01000, 11100, 10010, 00001, 00011, 00101, 00111}

c) $CL_{3,3}$ contains the set of 13 elements {000010, 000100, 001000, 011100, 010010, 000001, 000011, 000101, 000111, 100000, 110000, 101000, 111000}.

4 Correcting codes and module method in data hiding

4.1. Correcting code and covering function as special cases of module method

In data hiding area, by the essential relation between correcting code and covering function, for simplicity we need only to show the relation between correcting codes and module methods. Indeed, by the coset algorithm in the part 2.1, section 2, in the extract phase, a plaintext p hidden in the cover-string v' can be extracted from v' by computing the syndrome of v' as $p = s(v')=H.v'^T$, where $v'= v - l_u$ and H is the parity checking matrix of size $t \times n$ which can be presented as $H = (C_1,..,C_i,..,C_n)$, where each column vector C_i with size $t \times 1$ is considered an element in W_t, a $\mathbf{Z}_2$ - module. Suppose $v'= (x_1,..,x_n)$, $x_i \in \mathbf{Z}_2$, i=1,..,n. It is obvious that the syndrome $s(v') = H.v' = \Sigma_{1\leq i\leq n} C_i.x_i$ is exactly the sum we compute by module method, where the weight function h: {1,..,n} $\rightarrow M$-{0} is defined by $h(i) = C_i$ for all i=1,..,n with the $\mathbf{Z}_2$-module $M = W_t$. Computing $v'= v - l_u$ means that we need to find some positions on v to flip if in the coset leader l_u of $u = s(v)$-p, the corresponding positions are different from 0. In the following example, we present the arguments in details.

Example 2. The Hamming code can supply us an instance for module approach in hiding data. In details, we have the embedding scheme [7,3,1] by using the Hamming code (7, 4), taking W_3-{0} = {1,2,..,7} and considering each column $C_1,C_2,..,C_7$ of the parity checking matrix H of size 3×7 as a 3-bit presentation of these numbers

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$\quad C_1 \quad C_2 \quad C_3 \quad C_4 \quad C_5 \quad C_6 \quad C_7$$

Fig.1. Parity checking matrix H in Hamming code (7,4)

Each block F of the binary image can be seen as a column vector u of 7 entries: $u= (x_1,x_2,..x_7)^T$. Then taking operations on $\mathbf{Z}_2$-module we can write $H.u = C_1.x_1 + C_2.x_2 + ... + C_7.x_7$ where each column C_i can be seen as a vector in $V_3=\mathbf{Z}_2^3$ and the set C={C_1, C_2, C_3, C_4, C_5, C_6, C_7} is nothing but an 1-base in $\mathbf{Z}_2$ - module V_3. For security reason, one can choose an extra binary key k - as a column vector of 7 entries,

$k=(k_1,k_2,..,k_7)^T$ and taking operation XOR, with u we get $v = u \oplus k = (y_1,y_2,..,y_7)$ and $H.v = C_1.y_1+C_2.y_2+..+C_7.y_7$.

Replacing a position x_j in u by $x_j \oplus 1$ implies that the same position y_j in v is replaced by $y_j \oplus 1$. This gives us the marked vector v' satisfying the equation $H.v' = H.v \oplus C_j$. Now, suppose $H.v$=e and we need to hide a vector d (of 3 bits) which is considered as an element in W_3. We can flip at most one position x_j in u to hide d as follows:

Case $d=e$, the block u is kept intact, so that in the extracting phase, ones recover $H.v = d$. Case $d \neq e$, or equivalent, $e\text{-}d \neq 0$, we can find in C (an 1-base of W_3) a vector C_j so that $C_j = e\text{-}d$ (that means $C = e \oplus d$ in W_3). After flip x_j to $x_j \oplus 1$ in u, we have $H.v' = H.v \oplus C_j = e \oplus e \oplus d = d$, the result one needs to recover in the extract phase. Let us remark that the sum $C_1.y_1+C_2.y_2+..+C_7.y_7$ is exactly the result we get by the steps in **3.2.1** where $C_i.y_i$ is nothing but $h(i).T_i$ in that steps, with $h(i)=C_i$ *for* $i=1,2,..,7$.

4.2. Experimental results for finding 2-bases

As some results generated from our program, we obtained:

(i) The 2-base X={0001, 0010, 0100, 1110, 1001} and *Y*= {1000, 0100, 0010, 0111, 1001} provide us 2- embedding schemes which permit in each block of 5 pixels, by changing at most two pixels one can hide 4 bits. Hence regarding 2-embedding schemes, our schemes can be seen as some expanded for the list COV $(2, 6 \cdot 4^{a-1} - 1, 4a)$ with a=1, by equation (4)[7] mentioned in section 2.

(ii) The 2-base X = {1, 2, 3, 4, 5, 6, 8, 16, 24, 37, 45, 53, 58} and *Y*= {1, 2, 3, 4, 5, 6, 8, 16, 24, 32, 47, 55, 63} in $W_6 = \mathbf{Z}_2^6$ (the numbers can be seen as 6-bit strings). We provide new 2-bases with 13 elements, which permit us to hide 6 bits in each block of 13 pixels. Then we can extend them by the techniques mentioned in Lemma 2 to obtain new 2-bases for hiding 7-bit strings.

5. Conclusion

By flexibility of module approach for which correcting codes and covering function techniques are considered as the special cases, we can offer more new and powerful schemes to hide data without of using radius or distances as in correcting and covering codes.

In color images (24bpp with three channels Red, Green, Blue), or in palette images, especially for grayscale images, ones can obtain a higher ratio of secret bits hiding in each block of images by using some other module, such as $\mathbf{Z}_q$, $q>2$.

Several works consider some ways to hide as much as secret data in each block of pixels with low image distortion if it is possible. In preventing from steganalysis attacks, ones need a very high quality of stego-images, generally in palette images, hiding bits in each pixel is not good enough for security reason. In these situations, changing only a small k = 1, 2, 3 pixels in each block of pixels, using a huge number of key matrices to prevent effectively from exhausted attacks, by some k- embedding scheme modified for color images we can obtain stego-images with the high quality. These will be studied in future works.

References

1. J. Bierbrauer: Crandall's problem, unpublished, available from http://www.ws.binghamton.edu/fridrich/covcodes.pdf 1998.
2. Chang, C.C., Kieu, T.D., Chou, Y.C.: *A High Payload Steganographic Scheme based on (7,4) Hamming Code for Digital Images,* In: Electronic Commerce and Security 2008 Symposium, pp. 16-21 (2008).
3. Y.Chen, H.Pan, Y.Tseng. *A secure of data hiding scheme for two-color images*. In IEEE symposium on computers and communications, 2000.
4. Cheonsick Kim, Dongkyoo Shin, and Dongil Shin, Data Hiding in a Halftone Image U*sing Hamming Code (15,11),* ACIIDS 2011, LNAI 6592, pp 372-381, 2011.
5. R. Crandall: *Some notes on steganography*. Posted on steganography mailing list http://os.inf.tu-dresden.de/_westfeld/crandall.pdf (1998).
6. Galand and G. Kabatiansky: *Information hiding by coverings,* Proceedings of the IEEE Information Theory Workshop 2004, 151–154.
7. Jürgen Bierbrauer and Jessica Fridrich, *Constructing good covering codes for applications in Steganography*, Transactions on Data Hiding and Multimedia Security III , Lecture Notes in Computer Science, 2008, Volume 4920/2008, 1-22,
8. C.F. Lee, H.L. Chen, *A novel data hiding scheme based on modulus function*, The Journal of Systems and Software 83 (2010) 832–843.
9. M.B. Ould MEDENI, El Mamoun SOUIDI, A Novel Steganographic Protocol from *Error-correcting Codes*, *Journal of Information Hiding and Multimedia Signal Processing* 2010 ISSN 2073-4212. Ubiquitous International Volume 1, Number 4, October 2010
10. A. W. Nordstrom and J. P. Robinson: *An optimum nonlinear code*, Information and Control 11 (1967), 613–616.
11. Phan Trung Huy, Hai Thanh Nguyen : *On the Maximality of Secret Data Ratio in CPTE Schemes*. ACIIDS (1) 2011, LNCS/LNAI series 2011. pp. 88-99.
12. F. P. Preparata: A class of optimum nonlinear double-error-correcting codes, Information and Control 13 (1968), 378–400.
13. R. Struik: Covering Codes, Ph.D. dissertation, Eindhoven 1994.
14. Weiming Zhang, Xinpeng Zhang, and Shuozhong Wang, Maximizing Steganographic Em*bedding Efficiency by Combining Hamming Codes and Wet Paper Codes*, K. Solanki, K. Sullivan, and U. Madhow (Eds.): IH 2008, LNCS 5284, pp. 60–71, 2008. c_Springer-Verlag Berlin Heidelberg 2008.
15. A. Westfeld: *High Capacity Despite Better Steganalysis (F5–A Steganographic Algorithm)*, in I. S.Moskowitz (ed.): Information Hiding. 4th InternationalWorkshop, LNCS vol. 2137, Springer-Verlag, Berlin Heidelberg (2001), 289–302.
16. M.Y.Wu, J.H.Lee. *Anovel data embedding method for two-color fascimile images*. In Proceedings of international symposium on multimedia information processing. Chung-Li, Taiwan, R.O.C, 1998
17. Xinpeng Zhang and Shuozhong Wang, *Analysis of Parity Assignment Steganography in palette Images,* R. Khosla et al. (Eds.): KES 2005, LNAI 3683, pp. 1025–1031, 2005. Springer-Verlag, Berlin- Heidelberg (2005).
18. Xinpeng Zhang, Shuozhong Wang, Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security, Pattern Recognition Letters 25, 331–339 (2004).