Correspondence

Generalization of the ZZW Embedding Construction for Steganography

Weiming Zhang and Xin Wang

***Abstract*—We generalize the Zhang, Zhang, and Wang (ZZW) construction to produce larger code families for steganography from existing codes, which can cover the range of relative payloads more densely. We also prove that the expanded code family keeps the similar asymptotic property as the code family produced by the ZZW construction, that is, the codes follow the theoretic upper bound on embedding efficiency as relative payload tends to zero.**

***Index Terms*—Embedding efficiency, matrix coding, steganography, wet paper codes (WPCs), Zhang, Zhang, and Wang (ZZW) construction.**

I. Introduction

Steganography aims to embed secret messages into innocuous cover objects, such as digital images, for covert communication. In the embedding, only slight modifications on pixels are allowed to resist detection. For example, least significant bit (LSB) steganography, the most popular steganopgraphic technique, uses only the least significant bits of pixel values as carriers, in which the maximum amplitude of embedding changes on one pixel is one. In this case, the average number of bits carried by each pixel is called relative payload, and the average number of bits embedded per change is called embedding efficiency, which are used to measure the performance of an embedding scheme.

Given a relative payload, steganography desires to make the embedding efficiency as high as possible, which can be formulated as a coding problem [1], [2]. Many methods on binary embedding coding have been constructed using structured covering codes [2]–[5] or random codes [6], [7].

Recently, Zhang *et al.* [8] proposed a method to produce new families of codes with high embedding efficiency from existing codes, which was referred to as the Zhang, Zhang, and Wang (ZZW) construction by Fridrich [9]. In [9], Fridrich proves that the embedding efficiency of codes produced by the ZZW construction follows the upper bound on embedding efficiency as relative payload decreases to zero.

The drawback of the ZZW construction is that the generated codes can only provide sparse values of relative payloads, as shown in Fig. 1. However, for steganographic applications, we usually need codes for various payloads. In this paper, we generalize the ZZW construction to produce larger code families from existing codes, which can cover the range of relative payloads more densely. We also prove that the new construction has the same property as the ZZW construction as proved in [9], that is, the codes in a code family keep a steady distance to the upper bound on embedding efficiency when the relative payload decreases.

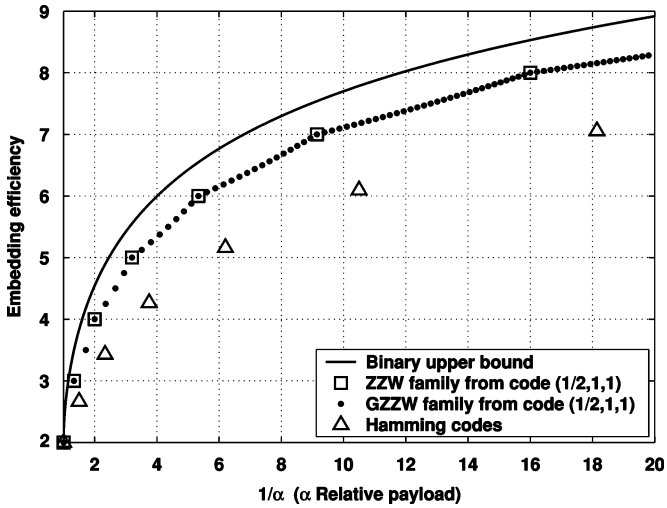

Fig. 1. Performance of code families generated from code $(1/2, 1, 1)$ by the ZZW construction and by the GZZW construction, which are compared with Hamming codes.

In the rest of this paper, we introduce some notations and the ZZW construction in Section II. In Section III, we describe the generalized ZZW (GZZW) construction and its performance. The asymptotic behavior of the proposed method is analyzed in Section IV. The paper is concluded in Section V.

II. ZZW Construction

The ZZW construction is based on matrix coding [3] and wet paper codes (WPCs) [10], [11]. Therefore, we first briefly introduce matrix coding and WPCs.

A. Matrix Coding

We will use calligraphic font for codes, and boldface font for matrices and vectors, and take images as covers to describe the proposed method. To embed data, the cover image is divided into disjoint segments of n pixels. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be the LSBs of pixels which are used as carriers for binary embedding. Because the embedded message is usually encrypted first, it can be considered as a binary random sequence. The message block is denoted by $\mathbf{m} = (m_1, \ldots, m_m)$. If a code $\mathcal{S}$ of length n can embed m bits of messages into n pixels using on average R_a changes, we say that $\mathcal{S}$ is (R_a, n, m).

The data embedding method derived from a linear code is called matrix coding [1], [3]. Let $\mathcal{C}$ be a binary $[n, n-m]$ linear code having the covering radius R and a parity check matrix $\mathbf{H}$. With $\mathbf{H}$, m bits of messages $\mathbf{m} \in \mathcal{F}_2^m$ can be embedded into n bits of cover symbols $\mathbf{x} \in \mathcal{F}_2^n$ by at most R changes. In fact, the key idea of matrix coding is to represent the message $\mathbf{m}$ with syndrome $\mathbf{H}\mathbf{x}^T$. If $\mathbf{H}\mathbf{x}^T = \mathbf{m}^T$, $\mathbf{m}$ is embedded into $\mathbf{x}$ without any change; otherwise we only need modify at most R bits of $\mathbf{x}$ to make $\mathbf{H}\mathbf{x}^T = \mathbf{m}^T$ hold, because the covering radius of code $\mathcal{C}$ is R. For perfect codes such as Hamming

Manuscript received March 15, 2009; revised May 10, 2009. First published June 10, 2009; current version published August 14, 2009. This work was supported by the Natural Science Foundation of China (60803155) and by the High-Tech Research and Development Program of China (2007AA01Z477). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nasir Memon.

The authors are with Department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China (e-mail: zwmshu@gmail.com).

Digital Object Identifier 10.1109/TIFS.2009.2024720

and Golay codes, the average number of changes can be calculated by $R_a = (1/2^m)\sum_{i=0}^{R} i\binom{n}{i}$.

For example, the covering radius of $[2^k-1, 2^k-k-1]$ Hamming code is one for any positive integer k. Therefore, by the parity check matrix of the Hamming code, we can embed any k bits into 2^k-1 bits with at most one change. The average number of changes is $(2^k-1)/2^k$ for Hamming codes, so we obtain a family of codes $((2^k-1)/2^k, 2^k-1, k)$, $k \geq 1$.

In general, for a code (R_a, n, m), we define the relative payload as $\alpha = m/n$, the change rate as $c = R_a/n$, and the embedding efficiency as $e = m/R_a = \alpha/c$. The largest relative payload that can be embedded by using the change rate c is $H_2(c)$ [1], so the embedding efficiency is bounded from above by

$$e(c) \leq \frac{H_2(c)}{c}, \qquad 0 \leq c \leq \frac{1}{2}. \tag{1}$$

When taking embedding efficiency e as a function of relative payload α, this bound is stated [6]

$$e(\alpha) \leq \frac{\alpha}{H_2^{-1}(\alpha)}, \qquad 0 \leq \alpha \leq 1 \tag{2}$$

where $H_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function, and $H_2^{-1}(y)$ is the inverse function of $H_2(x)$.

B. WPCs

By WPCs, the sender can embed messages into the cover $\mathbf{x} = (x_1, \ldots, x_n)$ by an arbitrarily selected channel with k changeable bits $x_j, j \in J \subset \{1, 2, \ldots, n\}, |J| = k$, which is not shared with the recipient. The secret message $\mathbf{m} = (m_1, \ldots, m_m)$ can be embedded into $\mathbf{x}$ by only modifying some bits in the selection channel J. In the terms of WPCs, we say the bits in the selection channel are dry and other bits are wet.

WPCs can be implemented by matrix coding based on random linear codes [10], in which, to embed $\mathbf{m}$ into $\mathbf{x}$, the sender changes some bits in the selection channel and modifies the cover-object $\mathbf{x}$ to satisfy

$$\mathbf{D}\mathbf{x}^T = \mathbf{m}^T \tag{3}$$

where $\mathbf{D}$ is an $m \times n$ binary pseudorandom matrix shared by the sender and the recipient. Therefore, the recipient can extract $\mathbf{m}$ from the modified $\mathbf{x}$ by only calculating $\mathbf{D}\mathbf{x}^T$ without any knowledge about the selection channel J.

It is pointed out [10] that we can communicate about k bits of messages by the above manner when the number of the changeable bits k $(k < n)$ is large enough. The encoding procedure can be executed by Gaussian elimination [10]. However, due to the high computational complexity of Gaussian elimination, the message has to be divided into small segments and then be embedded into disjoint blocks of the cover. By imposing the column weights of $\mathbf{D}$ to follow the distribution as in Luby transform (LT) codes, Fridrich *et al.* [11] also proposed an efficient method with low complexity. Therefore, by LT process, we can communicate a long message by one time, which can greatly simplify the implementation.

C. ZZW Construction

From any code (R_a, n, m) (not necessarily linear), the following ZZW construction generates a family of codes that are $(R_a, n2^k, m + R_a k)$, where k is an integer, $k \geq 0$.

To apply the ZZW construction, we should divide the cover image into disjoint blocks of $n2^k$ pixels. Without loss of generality, assume the cover image consists of L such blocks. Write the LSBs of each block as a matrix as

$$\begin{matrix} x_{1,1}, \ldots, x_{1,n} \\ x_{2,1}, \ldots, x_{2,n} \\ \cdots \\ x_{2^k,1}, \ldots, x_{2^k,n}. \end{matrix} \tag{4}$$

The block (4) is decomposed into two embedding channels in following manner.

First, compress each column into one bit with exclusive-or operation

$$y_i = \bigoplus_{j=1}^{2^k} x_{j,i} \qquad i = 1, 2, \ldots, n. \tag{5}$$

Take $(y_1, \ldots, y_n)$ as the first embedding channel, and apply a (R_a, n, m) code to it. Thus we can embed m bits of messages with R_a changes on average.

Second, take the first 2^k-1 elements from every column, and write

$$\mathbf{x}_1 = (x_{1,1}, \ldots, x_{2^k-1,1}), \ldots, \mathbf{x}_n = (x_{1,n}, \ldots, x_{2^k-1,n}). \tag{6}$$

In the embedding process of the first channel, if some y_i needs to be modified, we can flip any one of the 2^k bits in the ith column to change y_i, and therefore, we can map the ith column into any k bits of syndrome by $\mathbf{H}\mathbf{x}_i^T$, where $\mathbf{H}$ is the parity check matrix of the $[2^k-1, 2^k-k-1]$ Hamming code. In fact, if $\mathbf{H}\mathbf{x}_i^T$ is just the k bits we need, we flip the 2^kth bit $x_{2^k,i}$; otherwise we can freely change $\mathbf{H}\mathbf{x}_i^T$ to one of other 2^k-1 syndromes by changing one of the first 2^k-1 bits in this column. With this in mind, we construct the second embedding channel as follows:

$$\mathbf{H}\mathbf{x}_1^T, \mathbf{H}\mathbf{x}_2^T, \ldots, \mathbf{H}\mathbf{x}_n^T. \tag{7}$$

The channel (7) consists of nk bits. Because in the embedding of the first step there are on average R_a of y_i's to be changed, with these changes, the corresponding $R_a k$ bits in the second embedding channel (7) can be modified freely as the above analysis. Forbidding any change to the rest of the $(n-R_a)k$ bits, we construct a typical wet paper cover with $R_a k$ dry positions and $(n-R_a)k$ wet positions [10]. There are L blocks in total, each of which can introduce such a wet paper cover. We can cascade them to employ wet paper coding [10], [11], and embed additional $R_a k$ bits on average in every block without extra changes. The recipient can extract these embedded bits without any knowledge about the dry positions.

Combining the above two steps, we embed on average $m + R_a k$ bits of messages into every length-$n2^k$ block with R_a changes. Thus, we have a family of codes $(R_a, n2^k, m+R_a k), k \geq 0$, which have relative payload $\alpha(k)$, change rate $c(k)$, and embedding efficiency $e(k)$ such that

$$\begin{aligned} \alpha(k) &= \frac{m + R_a k}{n2^k}, \quad c(k) = \frac{R_a}{n2^k} \\ e(k) &= \frac{m + R_a k}{R_a} = e(0) + k, \qquad k \geq 0. \end{aligned} \tag{8}$$

For instance, the simplest code is $(1/2, 1, 1)$, which embeds one secret bit into each pixel with on average $1/2$ changes. With this trivial code, ZZW construction generates a family of codes $(1/2, 2^k, 1+k/2)$ with relative payload $(k+2)/2^{k+1}$ and embedding efficiency $k+2$, as shown in Fig. 1. This family outperforms almost all codes derived from structured covering codes [2]–[5]. The families generated from

low-density generator matrices (LDGM) codes [7] follow the bound even closer.

On the other hand, Fig. 1 also shows that the code family produced from code $(1/2, 1, 1)$ by ZZW construction has only sparse values of relative payloads. In Section III, we will generalize the ZZW construction to get a larger families of codes, which can cover the range of relative payloads densely and keep high embedding efficiency.

III. GZZW Construction

A. GZZW Construction for Binary Embedding

To use Hamming matrix coding, the ZZW construction has to divide the cover into blocks with 2^k rows. That is why relative payloads yielded by the ZZW construction are sparse. Now we propose an improved construction by allowing q rows in every block, where q is an arbitrary positive integer.

Assume $2^k \le q < 2^{k+1}$ and write $q = 2^k + p$, where $k \ge 0$ and $0 \le p < 2^k$. When $q = 2^k$, i.e., $p = 0$, we use the ZZW construction. For $p > 0$, each cover block has the following form:

$$\begin{array}{llll} x_{1,1}, & \ldots, & x_{1,n} \\ \ldots & & \\ x_{2^k,1}, & \ldots, & x_{2^k,n} \\ x_{2^k+1,1}, & \ldots, & x_{2^k+1,n} \\ \ldots & & \\ x_{2^k+p,1}, & \ldots, & x_{2^k+p,n}. \end{array} \tag{9}$$

We will decompose the cover into three embedding channels, the first two of which are from the ZZW construction.

First, adjust (9) to the block condition of ZZW construction. Let

$$z_{j,i} = x_{j,i} \oplus x_{2^k+j,i}, \quad j = 1, 2, \ldots, p, \quad i = 1, 2, \ldots, n. \tag{10}$$

In the ith column of (9), clip the last p bits and replace $x_{j,i}$ by $z_{j,i}$, where $1 \le i \le n$ and $1 \le j \le p$. Thus the block (9) is adjusted to

$$\begin{array}{llll} z_{1,1}, & \ldots, & z_{1,n} \\ \ldots & & \\ z_{p,1}, & \ldots, & z_{p,n} \\ x_{p+1,1}, & \ldots, & x_{p+1,n} \\ \ldots & & \\ x_{2^k,1}, & \ldots, & x_{2^k,n}. \end{array} \tag{11}$$

Based on a code (R_a, n, m), ZZW construction can embed $m + R_a k$ bits of messages into (11) with on average R_a changes. In the embedding process, if some $z_{j,i} = x_{j,i} \oplus x_{2^k+j,i}$ need to be changed, we can flip $x_{j,i}$ or $x_{2^k+j,i}$. Therefore, we have a change-free (dry) bit $x_{j,i}$, that is, if $x_{j,i}$ equals the needed bit, $x_{2^k+j,i}$ will be flipped; otherwise, $x_{j,i}$ will be flipped. Now we can construct the third embedding channel by taking first p bits from every column

$$x_{1,1}, \ldots x_{p,1}, \ldots, x_{1,n}, \ldots, x_{p,n}. \tag{12}$$

Note that, by the ZZW construction, on average R_a columns in block (11) are modified. In each of such columns, only one bit should be changed, and the changed position in a column is uniformly distributed, because the embedded messages are random. Therefore, the probability is $p/2^k$ for the case of the changed bit belonging to the first p bits of the column, which implies that there are $R_a p/2^k$ dry positions on average in the third embedding channel (12). We can cascade the third embedding channels constructed from every block and use WPCs to embed on average $R_a p/2^k$ bits more in every block. Thus we obtain a family of codes $(R_a, n(2^k + p), m + R_a(k + p/2^k))$, $k \ge 0$, $0 \le p < 2^k$. Because $q = 2^k + p$, we can also write the family as

$$\left(R_a, nq, m + R_a\left(\lfloor \log_2 q \rfloor + \frac{q}{2^{\lfloor \log_2 q \rfloor}} - 1\right)\right), \qquad q \ge 1 \tag{13}$$

which have relative payload $A(q)$, change rate $C(q)$, and embedding efficiency $E(q)$ such that

$$\begin{aligned} A(q) &= \frac{m + R_a\left(\lfloor \log_2 q \rfloor + \frac{q}{2^{\lfloor \log_2 q \rfloor}} - 1\right)}{nq} \\ C(q) &= \frac{R_a}{nq} \\ E(q) &= \frac{m}{R_a} + \lfloor \log_2 q \rfloor + \frac{q}{2^{\lfloor \log_2 q \rfloor}} - 1, \qquad q \ge 1. \end{aligned} \tag{14}$$

We call this method GZZW construction. When taking $q = 2^k$ in the GZZW construction, we just derive the ZZW construction. As shown in Fig. 1, the GZZW family from code $(1/2, 1, 1)$ expands the ZZW family from the same code extensively, and the embedding efficiency of the new codes are also close to the bound.

B. Practical Implementation Issues

The above GZZW construction decomposes the original cover into three embedding channels. In the first one, we embed messages with a code (R_a, n, m). For the other two channels, we use wet paper coding, which needs two pseudorandom matrices D_1 and D_2. The random matrices can be generated with a stego-key shared by the sender and the recipient. The width of the matrix equals the length of the embedding channel, and the height of the matrix equals the length of the message embedded in each channel. Therefore, the sender should also communicate the lengths of the messages embedded in the second and the third steps. In addition, if we use the LT process for the wet paper coding, the distribution of the column weights also depends on the length of the embedded message. Fridrich *et al.* [11] proposed to communicate the message lengths as follows. Assume the message lengths can be encoded with h bits. The sender divides the cover $\mathbf{x}$ into two pseudorandom disjoint subsets, such that $\mathbf{x}_h$ for the h bits and $\mathbf{x} - \mathbf{x}_h$ for the main message. In practice, taking 40 pixels for $\mathbf{x}_h$ is sufficient, because we can embed 40 bits into the LSBs of these pixels with code $(1/2, 1, 1)$.

When analyzing the embedding efficiency and relative payload for the GZZW construction in Section III-A, we assume the wet paper coding can always embed maximum message length, that is, the message length m equals the number of the changeable bits k. However, wet paper coding based on the LT process requires that $m < k$, which means there is a capacity loss in the LT process [11]. This capacity loss decreases as k increases. For example, the loss is about 10% for $k = 1000$ and about 5% for $k = 10\,000$. Therefore, when the embedded message (or the cover) is short, we cannot approach the optimal performance as derived in Section III-A by using the LT process.

The alternate method for wet paper coding is Gaussian elimination which has a negligible capacity loss for each embedding process [10]. However, because of the high computational complexity of Gaussian elimination, we have to divide the cover and the message into small blocks, for example, embedding less than 1000 bits into each block. Therefore, we need to communicate the length of the message for every block, which will also induce capacity loss and make the implementation complex.

We use the following embedding manner for wet paper coding in the GZZW construction. When the number of changeable bits $k > 1000$, we use LT process; and when $k \le 1000$, we use the fast Gaussian

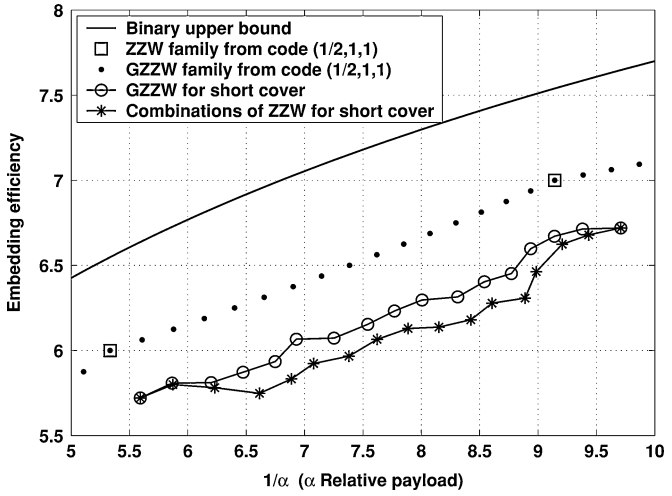

Fig. 2. Experimental results on the GZZW construction (the parameter $q = 16, 17, \ldots, 32$) for a short cover, which is compared with the combinations of the ZZW construction.

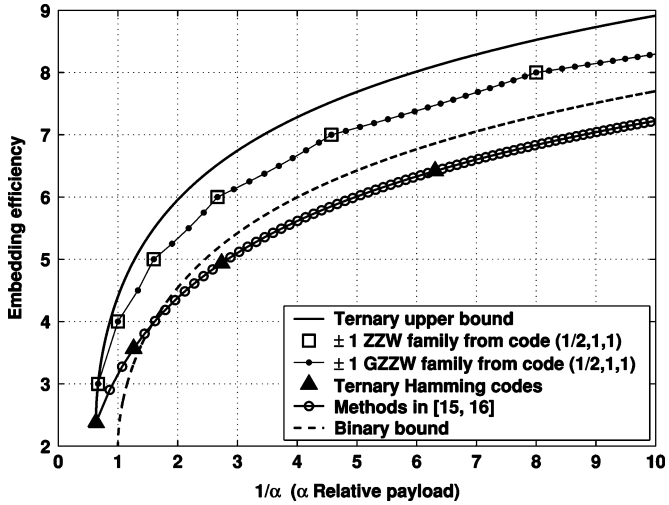

Fig. 3. Performance of code families generated from code $(1/2, 1, 1)$ by ± 1 ZZW construction and by ± 1 GZZW construction, which are compared with ternary Hamming codes and the methods in [15] and [16].

elimination proposed in [10]. To show how the performance drops for the case of short messages (or short covers), we embed messages into a small image with only 2^{16} pixels by the GZZW construction with the parameter $q = 16, 17, \ldots, 32$. In these cases, the number of changeable bits in the second channel is lager than 1000 and small than 10 000, and in the third channel we have only less than 1000 changeable bits. Therefore, we use the LT process in the second step and Gaussian elimination in the third step. The experimental results drawn in Fig. 2 show that the embedding efficiency drops about 0.4 from the values derived in Section III-A.

As mentioned in Section III-A, the GZZW construction generates more codes with dense relative payload than the ZZW construction does. Another solution to achieve arbitrary relative payload between two existing codes is linear combination. For example, assume there are two codes, S_1 and S_2, with relative payload 0.8 and 0.4, respectively, but we need relative payload 0.5. By using S_1 for 1/4 of the cover and S_2 for the remaining 3/4 of the cover, relative payload 0.5 can be attained. To compare the GZZW construction and the combination of ZZW construction, we also embedded messages into the image consisting of 2^{16} pixels by the combination of two code in the ZZW family, i.e., the codes with parameter $q = 16$ and $q = 32$. As shown in Fig. 2, the combination of ZZW construction yields a convex profile of performance.

C. GZZW Construction for ± 1 Embedding

In ± 1 embedding, the allowable modification on a pixel value is $+1$ or -1. Therefore each pixel can carry $\log_2 3$ bits of information by choosing $+1$, -1 or no change, that is, a ternary digit, with the pixel value modulo 3, so ± 1 embedding essentially involves a ternary coding problem. If a ± 1 embedding scheme has relative payload α, change rate c, and embedding efficiency $e = \alpha/c$, then e has the upper bound [12] such that

$$e(c) \leq \frac{H_3(c)}{c} = \frac{H_2(c) + c}{c} = \frac{H_2(c)}{c} + 1, \leq c \leq \frac{2}{3}. \qquad (15)$$

As a function of relative payload α, the embedding efficiency e is bounded from above by

$$e(\alpha) \leq \frac{\alpha}{H_3^{-1}(\alpha)}, \qquad 0 \leq \alpha \leq \log_2 3 \qquad (16)$$

where $H_3(x)$ is the ternary entropy function, and $H_3^{-1}(y)$ is the inverse function of $H_3(x)$.

A ± 1 embedding method based on binary codes was proposed in [13] and [14], which embeds messages in the LSB layer and the second LSB layer of pixels by using binary codes and WPCs, respectively. In fact, if the LSB of a pixel value g needs to be changed, its LSB can be flipped with $g+1$ or $g-1$. By the choice of ± 1, we can control the value of the second LSB of g, i.e., $\lfloor g/2 \rfloor \bmod 2$. Therefore, if R_a bits should be changed in the LSB embedding, we will have R_a change-free bits in the second LSB layer by the choice of ± 1.[1] With WPCs, additional R_a bits can be embedded in the second LSB layer without introducing new changes. That means that, applying the GZZW construction to the double layered embedding, we can obtain ± 1 embedding codes with relative payload $A_{\pm 1}(q)$ and embedding efficiency $E_{\pm 1}(q)$ as follows:

$$A_{\pm 1}(q) = \frac{m + R_a \left(\lfloor \log_2 q \rfloor + \frac{q}{2^{\lfloor \log_2 q \rfloor}} \right)}{nq}$$
$$E_{\pm 1}(q) = \frac{m}{R_a} + \lfloor \log_2 q \rfloor + \frac{q}{2^{\lfloor \log_2 q \rfloor}} = E(q) + 1. \qquad (17)$$

Using the same method as ± 1 ZZW construction [8], the embedding channel in the second LSB layer can be combined with the third embedding channel of GZZW construction, which will simplify the implementation. We refer above GZZW construction for ± 1 embedding as to "± 1 GZZW."

In Fig. 3, we compare the performance of the code family constructed from code $(1/2, 1, 1)$ by ± 1 GZZW construction and by ± 1 ZZW construction, and some previous methods in [12], [15], and [16]. The exploring modification direction (EMD) method [15] or grid coloring method [16] generates the same family of codes, embedding $\log_2(2n+1)$ bits into n pixels with $2n/(2n+1)$ changes on average, $n \geq 1$, which includes the ternary Hamming codes [12]. The ± 1 ZZW family from code $(1/2, 1, 1)$ outperforms the methods in [15] and [16], but provides fewer allowable values of relative payloads. By the generalized construction, i.e., ± 1 GZZW, the code family is expanded at various relative payloads and keeps following the upper bound.

[1]Note that change in only one direction is allowed when the pixel value is saturated, such as 0 or 255 for the 8-bit gray-scale images. In these cases, the second LSB will always be labelled wet, which will decrease the payload. Nevertheless, if these situations rarely occur, the effect on the overall performance can be negligible.

IV. Asymptotic Property of the GZZW Construction

The performance of the steganographic scheme in zero-payload limit is important for analyzing the security of batch steganography [17]. Fridrich [9] proves that the difference between the embedding efficiency of ZZW construction (8) and the upper bound (2) approaches a finite limit as $k \to \infty$.

Theorem 1 [9]: Under the notation established in Section II

$$\begin{aligned}\lim_{k\to\infty}&\left(\left(\frac{\alpha(k)}{H_2^{-1}(\alpha(k))}\right)-e(k)\right)\\&=\frac{1}{\ln 2}-\frac{m}{R_a}+\log_2\frac{n}{R_a}\\&\triangleq\lambda(R_a,n,m).\end{aligned}\tag{18}$$

The code family obtained by the ZZW construction is a subset of the code family by the GZZW construction. An important and interesting question is whether the expanded code family keeps the same asymptotic property. To make the asymptotic analysis simple, we use the upper bound on embedding efficiency at corresponding change rate (1), and derive the following result.

Theorem 2: Under the notation established in Sections II and III

$$\begin{aligned}\liminf_{q\to+\infty}&\left(\frac{H_2(C(q))}{C(q)}-E(q)\right)\\&=\lambda(R_a,n,m)\end{aligned}\tag{19}$$

$$\begin{aligned}\limsup_{q\to+\infty}&\left(\frac{H_2(C(q))}{C(q)}-E(q)\right)\\&=\lambda(R_a,n,m)+\frac{\ln 2-\ln\ln 2-1}{\ln 2}\\&\approx\lambda(R_a,n,m)+0.086.\end{aligned}\tag{20}$$

In particular, the case of $q=2^k$, $k\geq 0$, corresponds to the ZZW code family, for which $C(q)=c(k)$, $E(q)=e(k)$, and we have

$$\lim_{k\to\infty}\left(\left(\frac{H_2(c(k))}{c(k)}\right)-e(k)\right)=\lambda(R_a,n,m).\tag{21}$$

Theorem 2 implies that the distance between the embedding efficiency of a GZZW code family and the upper bound fluctuates with slight amplitude within 0.086 as q tends to infinity.

Based on the property on ZZW construction in Theorem 1, Fridrich [9] proposed to use the value $\lambda(R_a,n,m)=\lambda(\mathcal{S})$ for comparing codes in the zero-payload limit. Define $\mathcal{S}_1\prec\mathcal{S}_2$ if and only if $\lambda(\mathcal{S}_1)\leq\lambda(\mathcal{S}_2)$. In the expanded code family generated by the GZWW construction, the inferior limit of the distance to the bound is also $\lambda(R_a,n,m)$, and the superior limit is larger only by a small constant. Thus, it is still reasonable to order codes for their asymptotic performance by using $\lambda(R_a,n,m)$.

Note that, for ± 1 GZZW construction in Section III-C, we can prove the same asymptotic result as Theorem 2, because the embedding efficiency $E_{\pm 1}(q)$ in (17) increased by 1 compared with $E(q)$ in (14), and the ternary bound (15) is also larger by 1 than the binary bound (1).

V. Conclusion

A constructing method is proposed to produce new codes from old codes for steganography, which includes the ZZW construction [8] as a special case and therefore yields larger code families. The embedding efficiency of codes from the new construction also follows the upper bound on embedding efficiency as codes from the ZZW construction. We derive the superior limit and inferior limit of the distance to the bound on embedding efficiency as a function of change rate. Note that, when change rate tends to zero, relative payload also decreases to zero. Therefore, the results in Theorem 2 can be used to analyze the asymptotic property of codes in zero-payload limit. In fact, for ZZW construction ($q=2^k$), we derive the same limit when taking the bound as a function of relative payload and a function of change rate, respectively, as shown in Theorems 1 and 2.

Appendix

Proof of Theorem 2: First, we calculate that

$$\frac{H_2(C(q))}{C(q)}-E(q)=f(q)+g(q)+\log_2\frac{n}{R_a}-\frac{m}{R_a}\tag{22}$$

where

$$f(q)=\log_2 q-\left(\lfloor\log_2 q\rfloor+\frac{q}{2^{\lfloor\log_2 q\rfloor}}-1\right)\tag{23}$$

$$g(q)=-\left(\frac{1}{C(q)}-1\right)\log_2(1-C(q)).\tag{24}$$

Note that $C(q)$ tends to zero as $q\to\infty$; hence, by L'Hôpital's rule, we have

$$\lim_{q\to+\infty}g(q)=\frac{1}{\ln 2}.\tag{25}$$

So we only need to prove

$$\liminf_{q\to+\infty}f(q)=0\tag{26}$$

$$\limsup_{q\to+\infty}f(q)=\frac{\ln 2-\ln\ln 2-1}{\ln 2}.\tag{27}$$

Define function

$$f(x)=\log_2 x-\lfloor\log_2 x\rfloor-\frac{x}{2^{\lfloor\log_2 x\rfloor}}+1\tag{28}$$

where $x>0$ is a real number. Let $x\in[2^k,2^{k+1}-1]$, where k is a positive integer. Then

$$f(x)=\log_2 x-k-\frac{x}{2^k}+1.\tag{29}$$

The derivation

$$f'(x)=\frac{1}{x\ln 2}-\frac{1}{2^k}.\tag{30}$$

It is easy to see

$$\begin{aligned}&f'(x)>0 \text{ when } x\in\left[2^k,\frac{2^k}{\ln 2}\right) \text{ and}\\&f'(x)<0 \text{ when } x\in\left(\frac{2^k}{\ln 2},2^{k+1}-1\right]\end{aligned}\tag{31}$$

so $f(x)$ is increased when $x \in [2^k, (2^k/\ln 2)]$ and decreased when $x \in [(2^k/\ln 2), 2^{k+1}-1]$. Therefore, $f(x)$ has the maximum at $x = (2^k/\ln 2)$ and we obtain

$$\min(f(2^k), f(2^{k+1}-1)) \le f(x) \le f\left(\frac{2^k}{\ln 2}\right). \tag{32}$$

For any $q \ge 2$, let $k = \lfloor \log_2 q \rfloor$, and obviously $q \in [2^k, 2^{k+1}-1]$; thus,

$$\min(0, f(2^{k+1}-1)) \le f(q) \le f\left(\frac{2^k}{\ln 2}\right). \tag{33}$$

Since $k = \lfloor \log_2 q \rfloor \to +\infty$ as $q \to +\infty$, then we obtain

$$\liminf_{q\to+\infty} f(q) \ge \min(0, \liminf_{k\to+\infty}(f(2^{k+1}-1))) = 0 \tag{34}$$

$$\limsup_{q\to+\infty} f(q) \le f\left(\frac{2^k}{\ln 2}\right) = -\log_2 \ln 2 - \frac{1}{\ln 2} + 1. \tag{35}$$

Choose a subsequence $\{q = 2^k | k \ge 1\}$. It is easy to see

$$\lim_{k\to+\infty} f(2^k) = 0 \tag{36}$$

so

$$\liminf_{q\to+\infty} f(q) = 0. \tag{37}$$

On the other hand, choose a subsequence $q = \lfloor (2^k/\ln 2) \rfloor, k \ge 1$. According to the basic inequality $x - 1 \le \lfloor x \rfloor \le x$, we have

$$\begin{aligned} &\log_2\left(\frac{1}{\ln 2} - \frac{1}{2^k}\right) - \frac{1}{\ln 2} + 1 \\ &\quad \le f\left(\left\lfloor \frac{2^k}{\ln 2} \right\rfloor\right) \\ &\quad \le -\log_2 \ln 2 - \frac{1}{\ln 2} + 1. \end{aligned} \tag{38}$$

Therefore,

$$\lim_{k\to+\infty} f\left(\left\lfloor \frac{2^k}{\ln 2} \right\rfloor\right) = -\log_2 \ln 2 - \frac{1}{\ln 2} + 1. \tag{39}$$

Then we achieve

$$\begin{aligned} \limsup_{q\to+\infty} f(p) &= -\log_2 \ln 2 - \frac{1}{\ln 2} + 1 \\ &= \frac{\ln 2 - \ln\ln 2 - 1}{\ln 2}. \end{aligned} \tag{40}$$

In particular, combing (22), (25), and (36), we have

$$\begin{aligned} &\lim_{k\to\infty}\left(\frac{H_2(C(2^k))}{C(2^k)} - E(2^k)\right) \\ &\quad = \lim_{k\to\infty}\left(\frac{H_2(c(k))}{c(k)} - e(k)\right) \\ &\quad = \frac{1}{\ln 2} - \frac{m}{R_a} + \log_2 \frac{n}{R_a}. \end{aligned} \tag{41}$$

REFERENCES

[1] F. Galand and G. Kabatiansky, "Information hiding by coverings," in *Proc. IEEE Information Theory Workshop*, 2003, pp. 151–154.

[2] J. Bierbrauer and J. Fridrich, "Constructing good covering codes for applications in steganography," *LNCS Trans. Data Hiding Multimedia Security*, vol. 4902, pp. 1–22, 2008.

[3] R. Crandall, Some notes on steganography. Posted on Steganography Mailing List, 1998 [Online]. Available: http://os.inf.tu-dresden.de/westfeld/crandall.pdf

[4] D. Schönfeld and A. Winkler, "Embedding with syndrome coding based on BCH codes," in *Proc. ACM 8th Workshop on Multimedia and Security*, 2006, pp. 214–223.

[5] C. Munuera, "Steganography and error-correcting codes," *Signal Process.*, vol. 87, pp. 1528–1533, 2007.

[6] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 3, pp. 390–394, Sep. 2006.

[7] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, CA, 2007, vol. 6050, pp. 02–03.

[8] W. Zhang, X. Zhang, and S. Wang, "Maximizing steganographic embedding efficiency by combining Hamming codes and wet paper codes," in *Proc. 10th Information Hiding Conf. (LNCS 5284)*, 2008, pp. 60–71.

[9] J. Fridrich, "Asymptotic behavior of the ZZW embedding construction," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 151–154, Mar. 2009.

[10] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3923–3935, Oct. 2005.

[11] J. Fridrich, M. Goljan, and D. Soukal, "Efficient wet paper codes," in *Proc. 7th Int. Workshop on Information Hiding (LNCS 3727)*, 2005, pp. 204–218.

[12] F. Willems and M. Dijk, "Capacity and codes for embedding information in gray-scale signals," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 1209–1214, Mar. 2005.

[13] X. Zhang, W. Zhang, and S. Wang, "Efficient double-layered steganographic embedding," *IET Electron. Lett.*, vol. 43, no. 8, pp. 482–483, 2007.

[14] W. Zhang, X. Zhang, and S. Wang, "A double layered "plus-minus one" data embedding scheme," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 848–851, Nov. 2007.

[15] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Commun. Lett.*, vol. 10, no. 11, pp. 781–783, Nov. 2006.

[16] J. Fridrich and P. Lisoněk, "Grid coloring in steganography," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1547–1549, Apr. 2007.

[17] A. D. Ker, "Batch steganography and pooled steganalysis," in *Proc. 8th Information Hiding Conf. (LNCS 4437)*, 2006, pp. 265–281.