Matrix Embedding for Large Payloads

Jessica Fridrich and David Soukal

***Abstract*— Matrix embedding is a previously introduced coding method that is used in steganography to improve the embedding efficiency (increase the number of bits embedded per embedding change). Higher embedding efficiency translates into better steganographic security. This gain is more important for long messages than for shorter ones because longer messages are in general easier to detect. In this paper, we present two new approaches to matrix embedding for large payloads suitable for practical steganographic schemes—one based on a family of codes constructed from simplex codes and the second one based on random linear codes of small dimension. The embedding efficiency of the proposed methods is evaluated with respect to theoretically achievable bounds.**

***Index Terms*— steganography, covering codes, embedding efficiency.**

I. Introduction

THE main requirement for a steganographic scheme is statistical undetectability. Given the knowledge of the embedding algorithm and the source of cover media, the attacker should not be able to distinguish between stego and cover objects with success rate better than random guessing. Steganographic security is mostly influenced by the type of cover media, the method for selection of places within the cover that might be modified, the type of embedding operation, and the number of embedding changes, which is a quantity closely related to the length of the embedded data. Given two embedding schemes that share the first three attributes, the scheme that introduces fewer embedding changes will be less detectable.

Matrix embedding is a general principle that can be applied to most steganographic schemes to improve their embedding efficiency, which is defined as the expected number of random message bits embedded per one embedding change. Matrix embedding was introduced by Crandall [3], analyzed by Bierbrauer [1], and independently discovered by van Dijk et al. [8] and Galand et al. [5]. It was made popular by Westfeld who incorporated a specific implementation of matrix embedding using binary Hamming codes in his F5 algorithm [9]. It is intuitively clear that the gain in embedding efficiency can be larger for short messages than for longer ones. Since in general short messages are more difficult to detect than longer ones, improving the embedding efficiency for increasingly shorter messages becomes progressively less important for the overall security. In current steganographic literature, however, no codes suitable for embedding large payloads were described. The goal of this correspondence is to fill in the gap and present codes that practitioners will find useful for implementation.

Corresponding author: J. Fridrich, Department of Electrical and Computer Engineering, SUNY Binghamton, Binghamton, NY 13902-6000, USA.
D. Soukal is with the Department of Computer Science.

We describe two approaches. The first one is based on simplex codes and codes constructed from them, while the second approach uses random linear codes of small dimension. In Section II, we introduce the terminology and basic concepts of linear codes necessary to explain the embedding methods. In Section III, we briefly describe the principles of matrix embedding and state known bounds on achievable embedding efficiency. Matrix embedding based on simplex codes and random linear codes with small dimension is explained in Section IV. In the same section, the performance is compared to theoretically achievable bounds. Pseudo-codes are used to describe the embedding and extraction algorithms to ease the implementation and make this text self-contained. The paper is concluded in Section V.

II. NOTATION

Throughout this article, we will use some standard concepts and results from Coding Theory that can be found for example in [10]. Vectors or matrices are in bold and the calligraphic font is used for sets. Let $\mathbb{F}_2^n$ denote the space of all n-bit column vectors $\mathbf{x} = (x_1, \dots, x_n)^t$. A binary $[n, k]$ code $\mathcal{C}$ of block length n and dimension k is a k-dimensional vector subspace of $\mathbb{F}_2^n$, where the sum of two vectors and a multiplication of a vector by scalar are defined using the usual binary arithmetics. The k basis vectors written as rows of a matrix form the generator matrix $\mathbf{G}$. The orthogonal complement of an $[n, k]$ code is an $[n, n-k]$ code (called the dual code to $\mathcal{C}$) with an $(n-k) \times n$ generator matrix $\mathbf{H}$ with the property that $\mathbf{Hx} = \mathbf{0}$ for each $\mathbf{x} \in \mathcal{C}$. The matrix $\mathbf{H}$ is called the parity check matrix of $\mathcal{C}$.

For any $\mathbf{x} \in \mathbb{F}_2^n$, the vector $\mathbf{s} = \mathbf{Hx} \in \mathbb{F}_2^{n-k}$ is called the syndrome of $\mathbf{x}$. For each syndrome $\mathbf{s} \in \mathbb{F}_2^n$, the set $\mathcal{C}(\mathbf{s}) = \{\mathbf{x} \in \mathbb{F}_2^n | \mathbf{Hx} = \mathbf{s}\}$ is called a coset. Note that $\mathcal{C}(\mathbf{0}) = \mathcal{C}$. Obviously, cosets associated with different syndromes are disjoint. Also, from elementary linear algebra we know that every coset can be written as $\mathcal{C}(\mathbf{s}) = \mathbf{x} + \mathcal{C}$, where $\mathbf{x} \in \mathcal{C}(\mathbf{s})$ arbitrary. Thus, there are 2^{n-k} disjoint cosets, each consisting of 2^k vectors. Any member of the coset $\mathcal{C}(\mathbf{s})$ with the smallest Hamming weight is called a coset leader and will be denoted as $\mathbf{e}_L(\mathbf{s})$ (the Hamming weight w of a vector $\mathbf{x}$ is defined as the number of ones in $\mathbf{x}$, i.e., $w(\mathbf{x}) = x_1 + \cdots + x_n$).

The distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as the Hamming weight of their difference $d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} - \mathbf{y})$. For any $\mathbf{x} \in \mathcal{C}$, we denote as $\mathcal{B}(\mathbf{x}, R)$ the ball with center $\mathbf{x}$ and radius R, $\mathcal{B}(\mathbf{x}, R) = \{\mathbf{y} \in \mathbb{F}_2^n | d(\mathbf{x}, \mathbf{y}) \le R\}$.

The covering radius R of code $\mathcal{C}$ is defined as

$$R = \max_{\mathbf{x} \in \mathbb{F}_2^n} d(\mathbf{x}, \mathcal{C}), \tag{1}$$

where $d(\mathbf{x}, \mathcal{C}) = \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{x}, \mathbf{c})$ is the distance between $\mathbf{x}$ and the code $\mathcal{C}$. An R-covering of $\mathbb{F}_2^n$ is any subset $\mathcal{C}$ of $\mathbb{F}_2^n$

such that $\bigcup_{\mathbf{x}\in\mathcal{C}} \mathcal{B}(\mathbf{x}, R) = \mathbb{F}_2^n$, where $\mathcal{B}(\mathbf{x}, R)$ is the ball with center $\mathbf{x}$ and radius R. The average distance to code is defined as the average distance between a randomly selected vector from $\mathbb{F}_2^n$ and the code $\mathcal{C}$

$$R_a = \frac{1}{2^n} \sum_{\mathbf{x}\in\mathbb{F}_2^n} d(\mathbf{x}, C). \tag{2}$$

Clearly, $R_a \le R$.

III. Matrix embedding in steganography

We will assume that the *cover image* $\mathbf{X}$ is an element of $\mathcal{G}^n$, where $\mathcal{G}$ is the set of all possible pixel values. For example, in steganography using 8-bit grayscale digital images, $\mathcal{G}$ is the set of all integers in the range $[0, 255]$ and n is the number of pixels. Data embedding consists of modifying the values of selected pixels so that the modified (stego) image $\mathbf{Y}$ conveys the desired secret message. The impact of embedding is captured by a *distortion metric* $D : \mathcal{G}^n \times \mathcal{G}^n \to [0, \infty)$.

In most steganographic schemes, the message (a bit-stream) is communicated through a *bit-assignment function* $s : \mathcal{G} \to \mathbb{F}_q$ that assigns a bit to each possible pixel value. The most common bit-assignment function used in steganography is the least significant bit (LSB) of pixel values

$$s(i) = i \bmod 2. \tag{3}$$

The embedding operation is then designed so that its application to a pixel value modifies its assigned bit. For example, for LSB embedding, the embedding operation is flipping the LSB of the pixel value. Writing the pixels of image $\mathbf{X}$ as a one-dimensional vector, its vector of bits $s(\mathbf{X}) = \mathbf{x} \in \mathbb{F}_2^n$ is obtained by applying s to each element. Everywhere in this paper, we measure the impact of embedding using the Hamming distance $d : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \{0, 1, \ldots, n\}$ between the corresponding bit vectors, which is the number of embedding changes

$$D(\mathbf{X}, \mathbf{Y}) = d(s(\mathbf{X}), s(\mathbf{Y})) \text{ for all } \mathbf{X}, \mathbf{Y} \in \mathcal{G}^n. \tag{4}$$

We now briefly review a few relevant known facts about embedding schemes and covering codes that appeared in [1], [5] and establish some more terminology. Let $\mathcal{M}$ be the set of all messages. An *embedding scheme* on $\mathbb{F}_2^n$ with a distortion bound R is a pair of embedding and extraction functions Emb and Ext,

$$Emb: \mathbb{F}_2^n \times \mathcal{M} \to \mathbb{F}_2^n \text{ and } Ext: \mathbb{F}_2^n \to \mathcal{M}, \tag{5}$$

$$d(\mathbf{x}, Emb(\mathbf{x}, \mathbf{m})) \le R \text{ for all } \mathbf{m} \in \mathcal{M} \text{ and } \mathbf{x} \in \mathbb{F}_2^n, \tag{6}$$

such that for all $\mathbf{m} \in \mathcal{M}$ and all $\mathbf{x} \in \mathbb{F}_2^n$, $Ext(Emb(\mathbf{x}, \mathbf{m})) = \mathbf{m}$. In other words, (5) means that we can embed any message from $\mathcal{M}$ in any binary n-tuple and (6) states that we can do it using at most R changes.

The value $h = \log_2 |\mathcal{M}|$ is called the embedding capacity (in bits) and $\alpha = h/n$ the relative payload (in bits per pixel or bpp). We have the obvious inequality

$$|\mathcal{M}| \le 2^n \text{ or } \alpha \le 1. \tag{7}$$

We further define $\underline{e} = \frac{h}{R}$ as the *lower embedding efficiency* and $e = \frac{h}{R_a}$ as the *embedding efficiency*, where R_a is the expected number of changes over uniformly distributed cover objects $\mathbf{x} \in \mathbb{F}_2^n$ and messages $\mathbf{m} \in \mathcal{M}$. Note that since R is the upper bound on the number of embedding changes, for any embedding scheme $\underline{e} \le e$. The reason why here we used the same symbols R and R_a that already have the meaning of the covering radius and average distance to code will become clear from the matrix embedding theorem below. The matrix embedding theorem is taken from [1], [5] and gives a recipe on how to use an $[n, k]$ code to communicate $n-k$ bits using at most R changes in n pixels.

Theorem 1: (*Matrix embedding*) Let $\mathcal{C}$ be an $[n, k]$ code with a parity check matrix $\mathbf{H}$ and covering radius R. The embedding scheme below can communicate $n-k$ bits $\mathbf{m} \in \mathbb{F}_2^n$ in n pixels with bits $\mathbf{x}$ using at most R changes:

$$\begin{aligned} Emb(\mathbf{x}, \mathbf{m}) &= \mathbf{x} + \mathbf{e}_L(\mathbf{m} - \mathbf{H}\mathbf{x}) = \mathbf{y}, \\ Ext(\mathbf{y}) &= \mathbf{H}\mathbf{y}, \end{aligned}$$

where $\mathbf{m} \in \mathbb{F}_q^{n-k}$ is a sequence of $n-k$ message symbols and $\mathbf{e}_L(\mathbf{m} - \mathbf{H}\mathbf{x})$ is a coset leader of the coset $\mathcal{C}(\mathbf{m} - \mathbf{H}\mathbf{x})$.

Indeed, since $\mathcal{C}$ has covering radius R, we know that $d(\mathbf{x}, \mathbf{y}) = w(\mathbf{e}_L(\mathbf{m} - \mathbf{H}\mathbf{x})) \le R$, which shows that the embedding scheme has (a tight) distortion bound R. To see that $Ext(Emb(\mathbf{x}, \mathbf{m})) = \mathbf{m}$, note that $Ext(Emb(\mathbf{x}, \mathbf{m})) = \mathbf{H}\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{H}\mathbf{e}_L(\mathbf{m} - \mathbf{H}\mathbf{x}) = \mathbf{H}\mathbf{x} + \mathbf{m} - \mathbf{H}\mathbf{x} = \mathbf{m}$.

For an embedding scheme realized using matrix embedding, the expected number of embedding changes for messages uniformly distributed in $\mathbb{F}_2^{n-k}$ is equal to the average weight of all coset leaders of $\mathcal{C}$. It is reasonable to assume that the messages are drawn uniformly at random from $\mathbb{F}_2^{n-k}$ since typically they will be encrypted before embedding. We now show that the expected number of embedding changes is equal to the average distance to the code (2). Because any two words $\mathbf{x}, \mathbf{y}$ from the same coset $\mathcal{C}$ have the same distance from $\mathcal{C}$: $d(\mathbf{x}, \mathcal{C}) = d(\mathbf{y}, \mathcal{C}) = w(\mathbf{e}_L)$, the weight of any coset leader of $\mathcal{C}$, we have for the average distance to code

$$\begin{aligned} \frac{1}{2^n} \sum_{\mathbf{x}\in\mathbb{F}_2^n} d(\mathbf{x}, \mathcal{C}) &= \frac{1}{2^n} \sum_{\mathbf{s}\in\mathbb{F}_2^{n-k}} \sum_{\mathbf{x}\in\mathcal{C}(\mathbf{s})} d(\mathbf{x}, \mathcal{C}(\mathbf{s})) \\ &= \frac{1}{2^n} \sum_{i=1}^{2^{n-k}} 2^k w(\mathbf{e}_L(\mathbf{s})) = \frac{1}{2^{n-k}} \sum_{i=1}^{2^{n-k}} w(\mathbf{e}_L(\mathbf{s})), \end{aligned}$$

which is the average number of embedding changes for messages uniformly chosen from $\mathbb{F}_2^{n-k}$.

We now state a few useful bounds on the embedding efficiency. Because there are $\sum_{i=0}^{R} \binom{n}{i}$ ways in which one can make R or fewer changes in n pixels, we have

$$h = \log_2 |\mathcal{M}| \le \log_2 \sum_{i=0}^{R} \binom{n}{i} = \log_2 V(n, R) \le nH(R/n), \tag{8}$$

where $V(n, R)$ is the volume of a ball of radius R in $\mathbb{F}_2^n$ and $H(x) = -x \log_2 x - (1-x) \log_2 (1-x),\ 0 \le x \le 1/2$, is the binary entropy function. Inequality (8) also gives us an upper bound on the lower embedding efficiency $\underline{e} = \frac{h}{R}$ for a given relative payload $\alpha = \frac{h}{n}$:

$$H^{-1}(\alpha) \le \frac{R}{n} \implies \underline{e} = \frac{h}{R} = \alpha \cdot \frac{n}{R} \le \frac{\alpha}{H^{-1}(\alpha)}. \tag{9}$$

We note that this bound is also an asymptotic bound on the embedding efficiency e

$$e \lesssim \frac{\alpha}{H^{-1}(\alpha)} \tag{10}$$

which holds for almost all $[n, n(1-\alpha)]$ codes because the relative covering radius $\rho = R/n$ and the relative distance to code $\rho_a = R_a/n$ converge with $n \to \infty$. To state this more precisely, we formulate (and prove in the Appendix) the following theorem.

Theorem 2: For any $0 < \alpha < 1$ and any $\epsilon > 0$, the fraction of all binary $[n, (1-\alpha)n]$ codes for which $|\rho - \rho_a| \leq \epsilon$ tends to 1 as $n \to \infty$.

We close this section with one more useful bound on the embedding efficiency for codes restricted to a class of linear codes with relative payload α, or the class of $[n, n(1-\alpha)]$ codes. An upper bound on e requires a lower bound on R and R_a. Because there are $\binom{n}{i}$ possible sums of i columns of the parity check matrix $\mathbf{H}$, the number of cosets whose coset leaders have weight i is at most $\binom{n}{i}$. Thus, the covering radius R must be at least equal to R_n for which

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{R_n - 1} + \xi_n \binom{n}{R_n} = 2^{\alpha n},$$

where $0 \leq \xi_n < 1$ is a real number. Besides the lower bound $R \geq R_n$, we obtain a lower bound for R_a

$$R_a \geq \frac{\sum_{i=1}^{R_n - 1} i\binom{n}{i} + R_n \xi_n \binom{n}{R_n}}{2^{\alpha n}} \tag{11}$$

and an upper bound

$$e = \frac{\alpha n}{R_a} \leq \frac{\alpha n 2^{\alpha n}}{\sum_{i=1}^{R_n - 1} i\binom{n}{i} + R_n \xi_n \binom{n}{R_n}}. \tag{12}$$

IV. Matrix embedding for large payloads

The first example of matrix embedding given by Crandal [3] and Westfeld [9] was realized using $[2^p - 1, 2^p - 1 - p, 3]$ Hamming codes. Here, we can embed p message bits in a block of $2^p - 1$ pixels by performing at most one embedding change (we make no change with probability 2^{-p}). Thus, the embedding efficiency is $e_p = p/(1 - 2^{-p})$. Embedding p bits per $2^p - 1$ pixels means that the relative payload is $\alpha = \frac{p}{2^p - 1}$.

Note that Hamming codes do not lead to any embedding efficiency improvement for messages of relative length $2/3$ or higher. It is possible to use Hamming codes for message lengths larger than $2/3$ using a construction called the direct sum [2]. We can divide the message into two or more segments and embed them in disjoint parts of the cover using Hamming codes with different parameters. For example, given a relative payload 0.8 bpp, we may divide it into two halves and embed the first half in $0.4 \times n$ pixels and the second half in $0.6 \times n$ pixels. In the first part, we do not use matrix embedding and embed with efficiency 2, while in the second part, we may use matrix embedding with Hamming codes with $p = 2$ (because we are embedding at relative message length $0.4/0.6 = 2/3$). This will lead to embedding efficiency of $0.8/(0.4/2 + 0.4/e_2) = 16/7 \doteq 2.286$, which is better than not using matrix embedding at all but still far from theoretically achievable $e \doteq 0.8/H^{-1}(0.8) = 3.292$.

Algorithm 1 Embedding M bits in an N-element cover object using random linear codes.

1) To embed M bits in an N-element cover object, first find n such that $\alpha_n \geq \frac{M}{N} > \alpha_{n-1}$.
2) Read the next $n-k$ bits $\mathbf{x}$ from the cover object (along a stego-key dependent path) and the next message segment $\mathbf{m}$ of the same length.
3) Find any $\mathbf{e}$ that solves $\mathbf{He} = \mathbf{m} - \mathbf{Hx}$.
4) In the list of all 2^k codewords, find the closest codeword to $\mathbf{e}$, denote $\mathbf{c}(\mathbf{e})$.
5) [Embedding modifications] $\mathbf{y} = \mathbf{x} + \mathbf{e} - \mathbf{c}(\mathbf{e})$ is the stego object.
6) If we are at the end of the cover object, stop, otherwise go to 1.
7) [Extraction step] The message bits are extracted by following the same embedding path and calculating $n-k$ bits $\mathbf{m}$ from each block $\mathbf{y}$ of the stego object $\mathbf{m} = \mathbf{Hy}$.

A. Matrix embedding using random linear codes

Since random linear codes asymptotically achieve the bound (10) [5], we may attempt to construct good codes randomly. The downside of random codes is that they lack structure needed for fast encoding. Fortunately, for large relative payloads with $\alpha \to 1$ the codimension of the code will be close to code length and thus the dimension will be small enough to enable fast finding of coset leaders.

Indeed, to find the coset leader of the coset $\mathcal{C}(\mathbf{Hx} - \mathbf{m})$, we can first find an arbitrary vector $\mathbf{e}$ satisfying $\mathbf{He} = \mathbf{Hx} - \mathbf{m}$. If $\mathbf{c}(\mathbf{e})$ is the closest codeword to $\mathbf{e}$, then $\mathbf{e} - \mathbf{c}(\mathbf{e})$ is the coset leader of $\mathcal{C}(\mathbf{Hx} - \mathbf{m})$ because

$$d(\mathbf{e}, \mathbf{c}(\mathbf{e})) = \min_{\mathbf{c} \in \mathcal{C}(\mathbf{Hx} - \mathbf{m})} w(\mathbf{e} - \mathbf{c}) = w(\mathbf{e}_L(\mathbf{Hx} - \mathbf{m})).$$

We note that if $\mathbf{H}$ is generated randomly but already in a systematic form[1], finding $\mathbf{e}$ will be trivial. Thus, the most time consuming part of encoding is determining the closest codeword $\mathbf{c}(\mathbf{e})$. Since there are 2^k codewords, keeping the table of all codewords in memory requires $n2^k$ bits. Finding the closest codeword requires the same order of computations $O(n2^k)$. To keep the complexity and memory requirements low, the code dimension k should be small, e.g., $k \leq 14$. We note that for a fixed k, the relative payload α_n for the class of $[n, k]$ codes is $\alpha_n = \frac{n-k}{n}$. The pseudo-code for embedding is given in Algorithm 1.

While the parameter k can be a public knowledge, the block length n must be communicated to the recipient in the stego image itself because it depends on the message length. This is also the *only* piece of information that needs to be communicated along with the payload[2]. One possibility is to encode n using regular (non-matrix) embedding in a small subset of pixels pseudo-randomly chosen using the shared secret stego-key. The same stego-key can be used to generate the $(n-k) \times n$ matrix $\mathbf{H}$ using a pseudo-random

[1] $\mathbf{H} = [\mathbf{I}_{n-k}, \mathbf{D}]$, where $\mathbf{I}_{n-k}$ is a square $(n-k) \times (n-k)$ identity matrix.

[2] If we limit ourselves, for example, to $n \leq 256$, we would only need 8 bits for this overhead.

TABLE I
SPEED OF EMBEDDING FOR 1.3 MEGA-PIXEL IMAGE WITH FIXED BLOCK LENGTH $n = 100$.

dimension k	speed in seconds
10	0.82
12	2.42
14	8.65

number generator so that it does not have to be communicated. Alternatively, the matrix $\mathbf{H}$ could be even public as long as the message bits are embedded along a pseudo-random path generated from the stego-key.

Figure 1 shows the embedding efficiency of random linear codes for $k = 10$ and $k = 14$ for $n \leq 165$. A nice feature of random codes is that they provide an almost continuously changing family of codes with the same coding algorithm, allowing the sender to choose the code length n to match $\alpha_n = (n-k)/n$ to the relative payload length and thus use the whole embedding space in the cover object.

To see how much the coding improves the embedding efficiency, let us take two relative payloads 0.9 and 0.8. From Figure 1, using random linear codes of dimension 14, the embedding efficiency improves from 2 (no coding) to approximately 2.7 and 3, respectively. Thus, the coding reduces the impact of embedding the two messages as if we were embedding messages of length $\frac{0.9\times 2}{2.7} \doteq 0.67$ and $\frac{0.8\times 2}{3} \doteq 0.53$, respectively, without any coding. This is a significant improvement in view of the fact that the performance of current steganalyzers for some embedding methods may be quite sensitive to the relative payload in this range (see, for example, [6], [7]).

Note that the embedding efficiency of random codes is fairly close to the upper bound (11) for codes of the same length. The strange little "wiggles" in the upper bound are not a computing artifact but a real phenomenon whose explanation can be found in [4].

We can also see in Figure 1 the increase in embedding efficiency as the code dimension is increased from 10 to 14. Better performance could be obtained by further increasing the code dimension at the price of exponentially increasing complexity. Even though typical steganographic algorithms are run off-line on a computer and thus have less stringent requirements on complexity than typical channel coding applications, the code dimension cannot be increased much without severe complexity increase (recall that the complexity of coding is $O(n2^{n(1-\alpha)})$.

In Table IWe give a small example of how fast the embedding based on random codes runs on a computer. We simulated embedding into an image with $N = 1280 \times 1024$ pixels using a random code with block length $n = 100$. We measured the time taken to perform the embedding with dimensions $k = 10, 12$, and 14. The test was performed on Pentium IV running at 3.4 GHz with 1 GB RAM. The algorithm was implemented in C++ and compiled under Linux with GCC 3.4.3.

Algorithm 2 Embedding M bits in an N-element cover object using simplex codes.

1) To embed M bits in an N-element cover object, first find q such that $\frac{2^q-1-q}{2^q-1} \geq \frac{M}{N} > \frac{2^{q-1}-1-q-1}{2^{q-1}-1}$.
2) Read the next $p = 2^q - 1$ bits $\mathbf{x}$ from the cover object and the next message segment $\mathbf{m}$ of length $p = 2^q-1-q$ (follow a pseudo-random path through the image).
3) Find any $\mathbf{e}$ that solves $\mathbf{He} = \mathbf{m} - \mathbf{Hx}$, e.g., using Gaussian elimination.
4) For $\hat{\mathbf{e}} = (0, e_1, \ldots, e_{2^q-1})$ calculate $\mathbf{E} = (\mathbf{1} - 2\hat{\mathbf{e}})\mathbf{H}_{2^q}$ using the fast Hadamard transform.
5) $E_{i_0} = \max\{E_1, \ldots, E_{2^q}\}$, $\mathbf{u}$ = binary expansion of $i_0 - 1$ (LSB is the last).
6) The closest codeword to $\mathbf{e}$ is $\mathbf{c}(\mathbf{e}) = \sum_{i=1}^{q} u_i \mathbf{v}_i^t$, where $\mathbf{v}_i$ is the i-th row of the generator matrix $\mathbf{G}$.
7) [Embedding modifications] $\mathbf{y} = \mathbf{x}+\mathbf{e}-\mathbf{c}(\mathbf{e})$ is the stego object.
8) If we are at the end of the cover object, stop, otherwise go to 1.

B. Matrix embedding using simplex codes

Any structured codes with low dimension and fast decoding algorithms that are quantizers can be used for our purpose. In this section, we study the performance of the dual to Hamming codes — the simplex codes.

In Algorithm 2, we give the pseudo-code for matrix embedding using the simplex codes. The decoding algorithm for simplex codes can be found, for example, in [10]. We note that $\mathbf{H}$ is a parity check matrix of the $[2^q-1, q]$ simplex code, $\mathbf{H}_{2^q}$ is the Hadamard (Sylvester) matrix of order 2^q and the symbol $\mathbf{1}$ is a column vector of 2^q ones.

Other codes derived from simplex codes using common operations on codes, such as lengthening (increasing length by one) or augmenting (adding a codeword to the generator matrix) also give good performance and can be decoded using a simple modification of the decoding algorithm for simplex codes. If we augment the simplex code with an all-one vector $(1, \ldots, 1)$, we obtain a $[2^q - 1, q + 1]$ code, which coincides with the punctured first-order Reed-Muller code [10].

To embed with this code, we need to slightly modify Algorithm 2. We need to run Step 4 with $\mathbf{e}$ prepended with both '0' and '1': $\hat{\mathbf{e}}_0 = (0, e_1, \ldots, e_{2^q-1})$ and $\hat{\mathbf{e}}_1 = (1, e_1, \ldots, e_{2^q-1})$, obtaining now two vectors $\mathbf{c}_0$ and $\mathbf{c}_1$ in Step 6, taking the vector closer to $\mathbf{e}$ as $\mathbf{c}(\mathbf{e})$. To avoid calculating the Hadamard transform twice, note that $(\mathbf{1}-2\hat{\mathbf{e}}_1)\mathbf{H}_{2^q} = (\mathbf{1}-2\hat{\mathbf{e}}_0)\mathbf{H}_{2^q}-2\mathbf{h}_1$, where $\mathbf{h}_1$ is the first row of $\mathbf{H}_{2^q}$.

The embedding efficiency of simplex codes and augmented simplex codes for $q = 3, \ldots, 11$, is shown in Figure 1. Note that their performance is not as good as that of random linear codes. Also, they do not cover the range of α as densely as random codes—their relative payloads are $\alpha_q = \frac{2^q-1-q}{2^q-1}$ and $\alpha_q = \frac{2^q-2-q}{2^q-1}$ for the simplex and augmented simplex codes, respectively. On the other hand, they easily reach into the range of relative payload close to 1 and they do so with low computational complexity $O(q2^q) = O(n \log n)$ in terms of the code length n.

Again, to give an example of the improvement obtained from embedding using these structured codes, for a relative payload 0.94, the application of the augmented simplex code leaves the same impact as an uncoded embedding of a message with relative length $\frac{0.9\times 2}{2.4} = 0.75$, which is an improvement of about 20%.

We note that the parameter q is again the only information that needs to be communicated to the recipient in the same manner as described in the previous section.

V. Conclusions

In this paper, we present two simple coding schemes suitable for matrix embedding of large payloads. The codes can be applied to most steganographic schemes without any other changes to their embedding mechanism to increase their embedding efficiency—the expected number of random bits embedded using one embedding change. This will improve their steganographic security.

We showed that random linear codes provide good embedding efficiency and their relative embedding capacity densely covers the range of large payloads making such codes suitable for practical applications. Matrix embedding using simplex codes is more computationally efficient and can be used even for relative payloads above 0.9.

In this paper, we also introduce a new concept of an average distance to code as it is more relevant and directly related to embedding efficiency as currently used in steganography. We derive asymptotic bounds on the average distance to code to better contrast the performance of the proposed codes to the theoretically achievable embedding efficiency. The average distance to code asymptotically coincides with the covering radius with increasing code length. However, for small code lengths, codes with the smallest average distance to code may not necessarily have the smallest covering radius. We plan to elaborate on this issue in our future work.

Acknowledgements

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under the research grants number FA8750-04-1-0112 and F30602-02-2-0093. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U.S. Government. Special thanks belong to Jürgen Bierbrauer, Petr Lisoněk, and Miroslav Goljan for many useful discussions.

Appendix

Before we give a proof of Theorem 2, we formulate two auxiliary lemmas.

Lemma 1: For any $0 \le \rho < 1/2$ there exists an integer sequence k_n with

$$k_n/n \le 1 - H(\rho) + f(n),$$

where $f(n) \in O(n^{-1}\log n)$, such that the fraction of all binary $[n, k_n]$ codes that are $\lfloor \rho n \rfloor$-coverings tends to 1.

Proof: This lemma is proved in [2, page 325] (Theorem 12.3.5). ∎

Lemma 2: For any $H^{-1}(\alpha) < \rho < 1/2$, the fraction of all binary $[n, (1-\alpha)n]$ codes with covering radius at most $\lfloor \rho n \rfloor$ tends to 1 as $n \to \infty$.

Proof: Let us denote $\rho^\star = H^{-1}(\alpha)$. Because $1 - H(\rho) < 1 - H(\rho^\star)$ and $f(n) \to 0$ as n goes to infinity, there exists n_0 such that for any $n > n_0$,

$$1 - H(\rho) + f(n) \le 1 - H(\rho^\star) = 1 - \alpha.$$

Applying Lemma 1 to ρ, we obtain an integer sequence k_n for which

$$k_n/n \le 1 - H(\rho) + f(n) \le 1 - H(\rho^\star) = 1 - \alpha,$$

for $n > n_0$. Thus, $k_n \le (1-\alpha)n$ and the fraction of all $[n, k_n]$ codes whose covering radius is at most $\lfloor \rho n \rfloor$ tends to one. However the same is true for at least the same fraction of $[n, (1-\alpha)n]$ codes as well. This is so because for any two codes $\mathcal{C}_1 \subset \mathcal{C}_2$, $\mathcal{C}_1$ an $[n, k_1]$ code with covering radius R_1 and $\mathcal{C}_2$ an $[n, k_2]$ code with covering radius R_2, we have $R_2 \le R_1$. ∎

Proof of Theorem 2. Let $\rho^\star = H^{-1}(\alpha)$ and let $\mathcal{C}$ be an $[n, (1-\alpha)n]$ code. From (8) applied to $\mathcal{C}$ (note that $h = \alpha n$), we have for its relative covering radius ρ, $\rho^\star = H^{-1}(\alpha) \le R/n = \rho$. On the other hand, from Lemma 2 it follows that $\rho \le \rho^\star + \epsilon$ for all $n > n_0$, for a fraction of all $[n, (1-\alpha)n]$ codes that goes to 1 as $n \to \infty$.

The average distance to such codes is $R_a = \frac{1}{2^{\alpha n}} \sum_{l=0}^{\rho n} l c_l$, where c_l is the number of coset leaders of weight l. Because $\rho_a \le \rho$, we need a lower bound on ρ_a. Writing

$$R_a = \frac{1}{2^{\alpha n}} \sum_{l=0}^{\lfloor (\rho^\star - \epsilon)n \rfloor} l c_l + \frac{1}{2^{\alpha n}} \sum_{l=\lfloor (\rho^\star-\epsilon)n \rfloor + 1}^{\rho n} l c_l, \qquad (13)$$

we will find a lower bound on the second sum. To do so, we first derive an upper bound on c_l for l satisfying $l < (\rho^\star - \epsilon)n$. We start with

$$c_l \le \binom{n}{l} \le 2^{nH(l/n)}. \qquad (14)$$

The second inequality follows from Lemma 2.4.2 in [2] and holds for any $l < n/2$ for sufficiently large n (e.g., $n > n_1$). Using the fact that $H(x)$ is increasing on $[0, 1/2]$, from Taylor expansion of $H(x)$ at $\rho^\star$,

$$2^{nH(l/n)} \le 2^{nH(\rho^\star - \epsilon)} = 2^{n(\alpha - \epsilon H'(\xi))}, \qquad (15)$$

where $\rho^\star - \epsilon < \xi < \rho^\star$. Finally, because H' is decreasing on the same interval,

$$c_l \le 2^{\alpha n} 2^{-n\epsilon H'(\xi)} < 2^{\alpha n} 2^{-n\epsilon H'(\rho^\star)}, \qquad (16)$$

for any $l < (\rho^\star - \epsilon)n$.

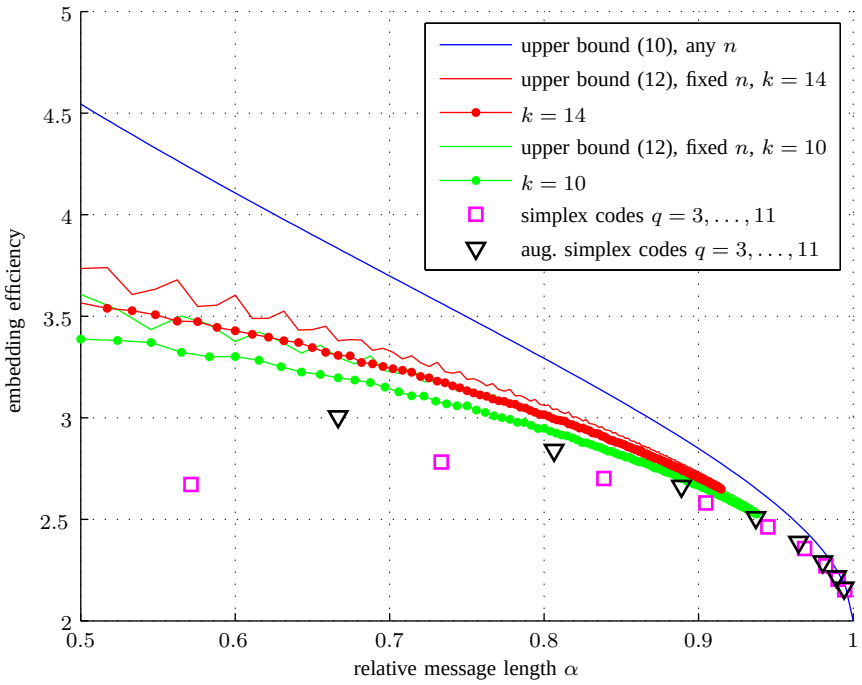

Fig. 1. Embedding efficiency vs. relative capacity (large payload case).

We now obtain a lower bound for R_a. Writing $l_0 = \lfloor(\rho^\star - \epsilon)n\rfloor$, from (13)

$$R_a \geq \sum_{l=l_0+1}^{\rho n} \frac{l c_l}{2^{\alpha n}} \geq (\rho^\star - \epsilon)n \sum_{l=l_0+1}^{\rho n} \frac{c_l}{2^{\alpha n}} \qquad (17)$$
$$= (\rho^\star - \epsilon)n \left(1 - \sum_{l=0}^{l_0} \frac{c_l}{2^{\alpha n}}\right)$$

because $\sum_{l=0}^{R} c_l = 2^{\alpha n}$. Using (16)

$$R_a \geq (\rho^\star - \epsilon)n \left(1 - (\rho^\star - \epsilon)n \cdot 2^{-n\epsilon H'(\rho^\star)}\right) \qquad (18)$$
$$= (\rho^\star - \epsilon)n(1 - \delta(n)),$$

where $\delta(n) \to 0$ exponentially fast with $n \to \infty$. Combining this result with $\rho_a \leq \rho \leq \rho^\star + \epsilon$, we obtain the following bounds for the average distance to code in terms of the relative quantities (for $n > \max(n_0, n_1)$)

$$(\rho^\star - \epsilon)(1 - \delta(n)) \leq \rho_a \leq \rho \leq \rho^\star + \epsilon, \qquad (19)$$

which proves the claim because $\epsilon > 0$ was arbitrary and $\delta(n) \to 0$ for $n \to \infty$.

References

[1] J. Bierbrauer. On crandall's problem. *Personal Communication, (available from http://www.ws.binghamton.edu/fridrich/covcodes.pdf)*, 1998.

[2] G. D. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes*, volume 54. Elsevier, North-Holland Mathematical Library, 1997.

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

[4] J. Fridrich, M. Goljan, and D. Soukal. Wet paper codes with improved embedding efficiency. *IEEE Transactions on Information Security and Forensics (in print)*, 2006.

[5] F. Galand and G. Kabatiansky. Information hiding by coverings. In *Proceedings ITW2003, Paris, France, 2003*, pages 151–154.

[6] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. Delp III and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, San Jose, CA, January 16–19, (to appear)*, January 2006.

[7] A. D. Ker. Steganalysis of LSB matching in grayscale images. *IEEESignal Processing Letters*, 12(6):441–444, June 2005.

[8] M. van Dijk and F. Willems. Embedding information in grayscale images. In *Proceedings of the 22nd Symposium on Information and Communication Theory in the Benelux, Enschede, The Netherlands, May 15–16, 2001*, pages 147–154.

[9] A. Westfeld. High capacity despite better steganalysis (F5—a steganographic algorithm). In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of *LNCS*, pages 289–302. Springer-Verlag, New York, 2001.

[10] F. J. M. Williams and N. J. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1977.