Wet Paper Codes with Improved Embedding Efficiency

Jessica Fridrich, *Member, IEEE*, Miroslav Goljan, and David Soukal

***Abstract*— Wet paper codes were previously proposed as a tool for construction of steganographic schemes with arbitrary (non-shared) selection channels. In this paper, we propose a new approach to wet paper codes using random linear codes of small codimension that at the same time improves the embedding efficiency (number of random message bits embedded per embedding change). Practical algorithms are given and their performance is evaluated experimentally and compared to theoretically achievable bounds. An approximate formula for the embedding efficiency of the proposed scheme is derived. The proposed coding method can be modularly combined with most steganographic schemes to improve their security.**

***Index Terms*— Steganography, covering codes, embedding efficiency, wet paper codes, matrix embedding, selection channel.**

I. Introduction

In steganography, the detectability of hidden data in a stego object is mostly influenced by four basic ingredients—1) the cover object, 2) the selection rule used to identify individual elements of the cover that might be modified during embedding, 3) the type of embedding operation that modifies the cover elements, and 4) the number of embedding changes (related to the message length). In this paper, we present a coding method that allows using arbitrary (non-shared) selection channels while decreasing the number of embedding changes. This is achieved using random linear codes of small codimension. Although the exposition is general and applies to many different types of cover objects, for simplicity, we assume that the cover object is a grayscale digital image.

The placement of embedding changes within the cover image is called the selection channel. Flat and less textured areas of the cover image allow the steganalyst to use better statistical models and thus more accurately detect the impact of embedding. Therefore, it appears that adaptive embedding that is confined to textured areas should improve steganographic security. However, it is not clear if this is, indeed, the case as narrowing the selection channel using publicly available information may help the attacker [1]. For example, the attacker may use the unmodified image segments to calibrate certain detection statistics as well as narrow down the analysis to smaller segments in the image with a higher density of embedding changes. To prevent the attacker from doing so, the selection channel should not be publicly available even in any "partial form." A possible remedy is to select it according to some side information that is in principle unavailable to the attacker (e.g., random) or that cannot be well estimated from the stego image, such a high resolution (unquantized) version of the cover object [2]. Steganography with non-shared selection channels requires codes for memories with defective cells [3], also called wet paper codes.

Assuming two embedding methods share the same source of cover images, the same selection channel and embedding operation, the one that introduces fewer embedding changes will be less detectable as it decreases the chance that any statistics used by an attacker will be sufficiently disturbed to mount a successful steganalysis attack. Thus, it is in the interest of the sender to embed data while imposing as few changes on the cover image as possible.

Minimizing the number of embedding changes leads to covering codes (or a more common term used in steganography—matrix embedding). This was discovered by Crandall in 1998 [4], later analyzed in an unpublished article by Bierbrauer [5], and recently independently discovered by Galand et al. [6].

This paper provides a new tool for steganography—a coding method that empowers the steganographer with the ability to use arbitrary selection channels while substantially decreasing the number of embedding changes, assuming the embedded message length is shorter than 70% of maximal embedding capacity. The method can be flexibly incorporated as a module into majority of existing steganographic methods.

To make this paper self-contained, in Section II, we review a few elementary concepts from coding theory that will be needed for the rest of the paper. A previously proposed approach to wet paper codes based on syndrome coding using random linear codes is briefly described in Section III. In the same section, we derive bounds on achievable embedding efficiency for linear codes. Section IV explains the proposed coding method whose embedding efficiency is calculated in Section V. Experimental results and their analysis and interpretation appear in Section VI. In Section VII, we discuss how the proposed technique improves steganographic security. Finally, the paper is concluded in Section VIII.

II. Basics of coding

We now review some elementary concepts from coding theory that are relevant for our study. A good introductory text to this subject is, for example, [7]. Throughout the text, boldface symbols denote vectors or matrices.

A binary code C is any subset of the space of all n-bit column vectors $\mathbf{x} = (x_1, \dots, x_n)^t \in \{0,1\}^n$. The vectors in

Corresponding author: J. Fridrich, Department of Electrical and Computer Engineering, SUNY Binghamton, Binghamton, NY 13902-6000, USA.

M. Goljan is with the Department of Electrical and Computer Engineering, SUNY Binghamton, Binghamton, NY 13902-6000, USA.

D. Soukal is with the Department of Computer Science, SUNY Binghamton, Binghamton, NY 13902-6000, USA.

C are called codewords. The set $\{0,1\}^n$ forms a linear vector space if we define the sum of two vectors and a multiplication of a vector by scalar using the usual arithmetics in the finite field GF(2); we will denote this field by $\mathbb{F}_2$. For any $C, D \subset \mathbb{F}_2^n$ and vector $\mathbf{x}$, $C+D = \{\mathbf{y} \in \mathbb{F}_2^n | \mathbf{y} = \mathbf{c}+\mathbf{d},\ \mathbf{c} \in C,\ \mathbf{d} \in D\}$, $\mathbf{x}+C = \{\mathbf{y} \in \mathbb{F}_2^n | \mathbf{y} = \mathbf{x}+\mathbf{c},\ \mathbf{c} \in C\}$.

The Hamming weight w of a vector $\mathbf{x}$ is defined as the number of ones in $\mathbf{x}$. The distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is the Hamming weight of their difference $d(\mathbf{x},\mathbf{y}) = w(\mathbf{x}-\mathbf{y})$. For any $\mathbf{x} \in C$ and a positive real number r, we denote as $B(\mathbf{x},r)$ the ball with center $\mathbf{x}$ and radius r, $B(\mathbf{x},r) = \{\mathbf{y} \in \mathbb{F}_2^n | d(\mathbf{x},\mathbf{y}) \le r\}$. We also define the distance between $\mathbf{x}$ and set $C \subset \mathbb{F}_2^n$ as $d(\mathbf{x},C) = \min_{\mathbf{c}\in C} d(\mathbf{x},\mathbf{c})$. The covering radius R of C is defined as $R = \max_{\mathbf{x}\in\mathbb{F}_2^n} d(\mathbf{x},C)$. The average distance to code C, defined as $R_a = 2^{-n}\sum_{\mathbf{x}\in\mathbb{F}_2^n} d(\mathbf{x},C)$, is the average distance between a randomly selected vector from $\mathbb{F}_2^n$ and the code C. Clearly, $R_a \le R$.

Linear codes are codes for which C is a linear vector subspace of $\mathbb{F}_2^n$. If C has dimension k, we call C a linear code of length n and dimension k (and codimension $n-k$), or we say that C is an $[n,k]$ code. Each linear code C of dimension k has a basis consisting of k vectors. Writing the basis vectors as rows of an $k \times n$ matrix $\mathbf{G}$, we obtain a generator matrix of C. Each codeword can be written as a linear combination of rows from $\mathbf{G}$. There are 2^k codewords in an $[n,k]$ code.

Given $\mathbf{x},\mathbf{y} \in \mathbb{F}_2^n$, we define their dot product $\mathbf{x}\cdot\mathbf{y} = x_1y_1 + \cdots + x_ny_n$, all operations in GF(2). We say that $\mathbf{x}$ and $\mathbf{y}$ are orthogonal if $\mathbf{x}\cdot\mathbf{y} = 0$. Given a code C, the dual code of C, denoted as $C^\perp$, is the set of all vectors $\mathbf{x}$ that are orthogonal to all vectors in C. The dual code of a $[n,k]$ code is a $[n,n-k]$ code with an $(n-k)\times n$ generator matrix $\mathbf{H}$ with the property that $\mathbf{Hx} = \mathbf{0}$ for each $\mathbf{x} \in C$. The matrix $\mathbf{H}$ is called the parity check matrix of 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 $\mathbf{s} \in \mathbb{F}_2^{n-k}$, the set $C(\mathbf{s}) = \{\mathbf{x} \in \mathbb{F}_2^n | \mathbf{Hx} = \mathbf{s}\}$ is called a coset. Note that $C(\mathbf{0}) = C$. Obviously, cosets associated with different syndromes are disjoint. Also, from elementary linear algebra we know that every coset can be written as $C(\mathbf{s}) = \mathbf{x}+C$, where $\mathbf{x} \in C(\mathbf{s})$ arbitrary. Thus, there are 2^{n-k} disjoint cosets, each consisting of 2^k vectors. Any member of the coset $C(\mathbf{s})$ with the smallest Hamming weight is called a coset leader and will be denoted as $\mathbf{e}_L(\mathbf{s})$.

Lemma 1: Given a coset $C(\mathbf{s})$, for any $\mathbf{x} \in C(\mathbf{s})$, $d(\mathbf{x},C) = w(\mathbf{e}_L(\mathbf{s}))$. Moreover, if $d(\mathbf{x},C) = d(\mathbf{x},\mathbf{c}')$ for some $\mathbf{c}' \in C$, the vector $\mathbf{x}-\mathbf{c}'$ is a coset leader.

Proof: $d(\mathbf{x},C) = \min_{\mathbf{c}\in C} w(\mathbf{x}-\mathbf{c}) = \min_{\mathbf{y}\in C(\mathbf{s})} w(\mathbf{y}) = w(\mathbf{e}_L(\mathbf{s}))$. The second equality follows from the fact that if $\mathbf{c}$ goes through the code C, $\mathbf{x}-\mathbf{c}$ goes through all members of the coset $C(\mathbf{s})$. ■

Lemma 2: If C is an $[n,k]$ code with a $(n-k)\times n$ parity check matrix $\mathbf{H}$ and covering radius R, then any syndrome $\mathbf{s} \in \mathbb{F}_2^{n-k}$ can be written as a sum of at most R columns of $\mathbf{H}$ and R is the smallest such number. Thus, we can also define the covering radius as the maximal weight of all coset leaders.

Proof: Any $\mathbf{x} \in \mathbb{F}_2^n$ belongs to exactly one coset $C(\mathbf{s})$ and from Lemma 1 we know that $d(\mathbf{x},C) = w(\mathbf{e}_L(\mathbf{s}))$. But the weight $w(\mathbf{e}_L(\mathbf{s}))$ is the smallest number of columns in $\mathbf{H}$ that must be added to obtain $\mathbf{s}$. ■

III. Wet paper codes and covering codes

Let us assume that the cover image $\mathbf{x}$ consists of n pixels x_i, $x_i \in \{0,1,\ldots,255\}$. The sender selects k changeable pixels x_j, $j \in J \subset \{1,2,\ldots,n\}$, $|J| = k$, which is the selection channel. The changeable pixels may be used and modified independently from each other by the sender to communicate a secret message to the recipient, while the remaining pixels are not modified during embedding. We stress that the selection channel is not shared with the recipient.

It is further assumed that the sender and the recipient agree on a mapping $b : \{0,1,\ldots,255\} \to \{0,1\}$, for example, $b(x) =$ the LSB of x (Least Significant Bit). During embedding, the sender either leaves the changeable pixels x_j, $j \in J$, unmodified or replaces x_j with y_j to modify its bit from $b(x_j)$ to $b(y_j)$. The vector of cover image bits $\mathbf{b_x} = (b(x_1),\ldots,b(x_n))^t$ changes to $\mathbf{b_y} = (b(y_1),\ldots,b(y_n))^t$, where $\mathbf{x}^t$ denotes the transpose of $\mathbf{x}$. To communicate m bits $\mathbf{m} \in \mathbb{F}_2^m$, the sender modifies some changeable pixels x_j, $j \in J$, so that

$$\mathbf{Db_y} = \mathbf{m}, \tag{1}$$

where $\mathbf{D}$ is an $m \times n$ binary matrix shared by the sender and the recipient. Equation (1) can be further rewritten to

$$\mathbf{Dv} = \mathbf{m} - \mathbf{Db_x} \tag{2}$$

using the variable $\mathbf{v} = \mathbf{b_y} - \mathbf{b_x}$ with non-zero elements corresponding to the pixels the sender must change to satisfy (1). In (2), there are k unknowns v_j, $j \in J$, while the remaining $n-k$ values v_i, $i \notin J$, are zeros. Thus, on the left hand side, the sender can remove from $\mathbf{D}$ all $n-k$ columns $\mathbf{d}_i$, $i \notin J$, and also remove from $\mathbf{v}$ all $n-k$ elements v_i with $i \notin J$. Keeping the same symbol for $\mathbf{v}$, (2) now becomes

$$\mathbf{Hv} = \mathbf{s}, \tag{3}$$

where $\mathbf{H}$ is an $m \times k$ matrix consisting of those columns of $\mathbf{D}$ corresponding to indices J, $\mathbf{v}$ is an unknown $k \times 1$ vector, and $\mathbf{s} = \mathbf{m} - \mathbf{Db_x}$ is the $m \times 1$ right hand side. Thus, the sender needs to solve a system of m linear equations with k unknowns in GF(2). The question of solvability of (3) has been investigated in detail in [8] and is briefly reviewed below.

In the language of coding theory, assuming $\mathbf{H}$ is a parity check matrix of some $[k, k-m]$ linear code, solving (3) is decoding the noisy codeword $\mathbf{v}$ from its syndrome $\mathbf{s}$. Minimizing the number of embedding changes means finding such a solution $\mathbf{v}$ to (3) (possibly out of many) with the minimal weight—a coset leader.

The matrix $\mathbf{H}$ is obtained from $\mathbf{D}$ as a column sub-matrix as defined by the selection channel. Because the selection channel can be arbitrary, e.g., even random or dependent on the cover, it is difficult to impose structure on $\mathbf{D}$ that would be inherited by $\mathbf{H}$ and that would help us solve (3). Moreover, we need a whole class of good codes for various values of n, k, and m. We now quote results from [8] that show that, asymptotically, random linear codes enable communication of k bits and that with increasing code length n they also achieve the best possible embedding efficiency for a fixed

relative message length $\alpha = m/k$ and fixed rate $r = k/n$. This makes such codes ideal for steganographic applications provided there exist efficient coding algorithms (Section IV-A).

Let us assume that the sender always tries to embed as many bits as possible by adding rows to $\mathbf{D}$ while (3) still has a solution. For random binary matrices whose elements are i.i.d. realizations of a random variable uniformly distributed in $\{0, 1\}$, the expected value of the maximum message length $m_{\max}$ that can be communicated in this manner is [8]

$$m_{\max} = k + O(2^{-k/4}) \tag{4}$$

as $k \to \infty$, $k < n$. Thus, these variable-rate random linear codes asymptotically achieve the maximal embedding capacity.

To address the embedding efficiency of the syndrome coding approach above, let R be the covering radius of the linear code with parity check matrix $\mathbf{H}$. From Lemma 2, we know that every syndrome can be generated by adding at most R columns, where R is the covering radius. Because there are $\binom{k}{i}$ different sums of i columns of $\mathbf{H}$, we obtain the sphere-covering bound

$$2^m \leq \sum_{i=0}^{R} \binom{k}{i} = V(k, R) \leq 2^{kH(R/k)}, \tag{5}$$

where $V(k, R)$ is the volume of a ball of radius R in $\mathbb{F}_2^k$ and $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function. The second inequality is a frequently used bound in coding (e.g., Lemma 2.4.3 in [9]). Since we are embedding m bits using k changeable pixels, the relative message length is $\alpha = m/k$. We define the lower embedding efficiency $\underline{e}$ as the ratio $\underline{e} = m/R$, which is the number of embedded bits per embedding change in the worst case when we have to make all R changes. For practical purposes, steganographers are likely to be more interested not in the worst case but the average case. Thus, we define the embedding efficiency as $e = m/R_a$, where R_a is the average distance to code. Obviously, $\underline{e} \leq e$, which is why $\underline{e}$ is called the lower embedding efficiency. The inequality (5) enables us to derive an upper bound on $\underline{e}$ and eventually on e.

From (5),

$$\begin{aligned} \alpha &\leq H(R/k) \\ H^{-1}(\alpha) &\leq R/k \\ \underline{e} = \frac{m}{R} &\leq \frac{\alpha}{H^{-1}(\alpha)}, \end{aligned} \tag{6}$$

where $H^{-1}(x)$ is the inverse of $H(x)$ on $[0, 1/2]$. Thus, we have obtained an upper bound on the lower embedding efficiency for a fixed relative message length α. It is possible to show that $\frac{\alpha}{H^{-1}(\alpha)}$ is also an asymptotic upper bound on e. The proof of this statement can be found in [10].

Furthermore, it is known that the upper bound is asymptotically achievable using almost all random linear codes $[k, k - \alpha k]$ with $k \to \infty$ (see Theorem 12.3.5. in [9], page 325).

We conclude that random linear codes are good candidates for wet paper codes with minimal number of embedding changes. In the next section, we introduce a practical embedding method and study its performance.

IV. Combining wet paper codes with matrix embedding

We repeat that the sender's goal is to find a solution to (3) $\mathbf{Hv} = \mathbf{s}$ with the smallest weight $w(\mathbf{v})$. Equivalently, the sender tries to find the coset leader of the coset $C(\mathbf{s})$. This is, however, in general an NP complete problem [11]. A common approach to overcome the complexity issue is to use structured codes. However, as discussed before, $\mathbf{H}$ is obtained from $\mathbf{D}$ through a selection process over which the sender may not have any control. It is possible, though, to impose some stochastic structure on $\mathbf{D}$ that would be transferred to $\mathbf{H}$, such as the distribution of weights of columns. Indeed, this idea was utilized for efficient implementation of wet paper codes in [12], where the column weights of $\mathbf{D}$ were required to follow the robust soliton distribution as in LT codes [13]. While such low density parity check matrices indeed work quite well to quickly solve (3), attempts to adjust the LT process to improve the embedding efficiency were only moderately successful (Section 4.2 in [12]). It is possible, though, that other stochastic properties may be imposed on $\mathbf{D}$ that will allow finding solutions of (3) with small weight efficiently. This topic will be the subject of our future effort.

An obvious simple measure to avoid large computational complexity is to use codes of small codimension where exhaustive searches are computationally feasible. In this case, however, we need to study how much is lost on optimality of coding as the results in Section III are only asymptotic.

A. Random codes of small codimension

Let us assume that the cover image has n pixels and k changeable pixels and that we wish to communicate m message bits. The sender and receiver agree on a small integer p (e.g., $p < 20$) and using the stego key divide the cover image into $n_B = m/p$ disjoint pseudo-random blocks of cardinality $n/n_B = pn/m$ (for simplicity we assume the quantities above are all integers). Each block will contain on average $k/n \times pn/m = pk/m = p/\alpha$ changeable pixels, where $\alpha = m/k$, $0 \leq \alpha \leq 1$, is the relative message length. The sender will use a pseudo-random binary $p \times pn/m$ matrix $\mathbf{D}$ for embedding up to p bits. The matrix $\mathbf{D}$ can be the same for each block, publicly available, or also generated from a secret stego key. Note that since duplicates and zero columns in $\mathbf{D}$ do not help, as long as[1] $n/n_B = pn/m < 2^p$, we can generate $\mathbf{D}$ so that its columns are non-zero and mutually different.

As described in Section III, in each block the sender forms a binary sub-matrix $\mathbf{H}$ of $\mathbf{D}$ and the syndrome $\mathbf{s}$ from the set of all changeable pixels in that block. The matrix $\mathbf{H}$ will have exactly p rows and, on average, p/α columns. Let $C_1 \subset \mathbb{F}_2^p$ be the set of all columns of $\mathbf{H}$, and $C_{i+1} = C_1 + C_i - (C_1 \cup \cdots \cup C_i) - \{\mathbf{0}\}$, for $i = 1, \ldots, p$. Note that $C_i = \emptyset$ for $i > R$,

[1] This will be satisfied for embedding in typical digital media files because we use $p \approx 20$ (see below).

Algorithm 1 Meet-in-the-middle algorithm for finding coset leaders

```
If (s ∈ C_1) {
  v_{j_1} = 1;
  set v_j = 0 for all other j;
  return;
} else {
  l = r = 1;
}
while ((s + C_l) ∩ C_r == ∅) {
  if (l == r) {
    r = r + 1;
    if (C_r not created) create C_r;
  } else {
    l = l + 1;
    if (C_l not created) create C_l;
  }
}
// there is a solution v of weight l +
r determined by any vector from the intersection
```

where R is the covering radius of $\mathbf{H}$. Also note that C_i is the set of syndromes that can be obtained by adding i columns of $\mathbf{H}$ but no less than i (equivalently, C_i is the set of all coset leaders of weight i).

Let $\mathbf{s} = \mathbf{h}_{j_1} + \cdots + \mathbf{h}_{j_r}$, where r is the minimal number of columns of $\mathbf{H}$ adding up to $\mathbf{s}$, $r \leq R$. Then, $\mathbf{s} + \mathbf{h}_{j_1} + \cdots + \mathbf{h}_{j_{\lfloor r/2 \rfloor}} = \mathbf{h}_{j_{\lfloor r/2 \rfloor + 1}} + \cdots + \mathbf{h}_{j_r}$, which implies $(\mathbf{s} + C_{\lfloor r/2 \rfloor}) \cap C_{r - \lfloor r/2 \rfloor} \neq \emptyset$ and $\mathbf{v}$ with zeros everywhere except for indices $j_1, \ldots, j_r$ solves (3). This leads to the Algorithm 1 for finding the coset leader.

After the solution $\mathbf{v}$ is found, the sender modifies the pixels in the block accordingly—the non-zero elements of $\mathbf{v}$ determine pixels x_i within the block where embedding changes must take place. We remind that the modified block of pixels in the stego image is denoted $\mathbf{y}$.

The extraction algorithm is very simple. The recipient knows n from the stego image and knows p as this is a publicly shared parameter. Since the message length m is used in dividing the image into blocks, it needs to be communicated in the stego image as well. This can be arranged in many different ways, for example, by isolating from the image a small subset (using the stego key) and embedding $\log_2 m$ bits in it using standard wet paper code from Section III. Knowing m, the recipient uses the secret stego key and partitions the rest of the stego image into the same disjoint blocks as the sender and extracts p message bits $\mathbf{m}$ from each block of pixels $\mathbf{y}$ as $\mathbf{m} = \mathbf{D}\mathbf{y}$.

B. Algorithm complexity and implementation issues

In Algorithm 1, in the worst case, we need to calculate all sets $C_1, \ldots, C_{\lceil R/2 \rceil}$. The cardinalities of C_i increase with i, achieve a maximum for $i \approx R_a$, and then quickly fall off to zero for $i > R_a$. With increasing length of the code (or increasing p), $R_a \to R$. This means that the above algorithm avoids computing the largest of the sets C_i. Nevertheless, we will need to keep in memory the sets $C_i, i = 1, \ldots, \lceil R/2 \rceil$ and the indices $j_1, \ldots, j_i$ for each element of C_i. Because on average $|C_1| = p/\alpha$, we have on average $|C_i| \leq \binom{p/\alpha}{i}$. Thus, the total memory requirements are bounded by $O(R/2 \cdot \binom{p/\alpha}{R/2}) \approx O(p \cdot 2^{p/\alpha H(R\alpha/2p)}) \approx O(p2^{\beta p})$, where $\beta = \frac{H(H^{-1}(\alpha)/2)}{\alpha} < 1$, because $R \approx p/\alpha H^{-1}(\alpha)$ for large p from (6). For example, for $\alpha = 1/2, \beta = 0.61$. To obtain a bound on the computational complexity, note that we need to compute $C_1 + C_i$ for $i = 1, \ldots, R/2$. Thus, the computational complexity is bounded by $O(R/2 \cdot p/\alpha \cdot \binom{p/\alpha}{R/2}) \approx O(p^2 2^{\beta p})$.

At this point, we note that we studied other approaches for finding coset leaders, such as the method based on non-expurgated syndrome trellis proposed by Wadayama [14]. Because the computational complexity of Wadayama's method is $O(p2^p)$, it is asymptotically slower than the meet-in-the-middle method.

We now comment on the solvability of (3). The equation $\mathbf{H}\mathbf{v} = \mathbf{s}$ will have a solution for all $\mathbf{s} \in \mathbb{F}_2^p$ if and only if $\text{rank}(\mathbf{H}) = p$. The probability of this is $1 - O(2^{p(1-k/m)})$, as this is the probability that a random binary matrix with dimension $p \times p/\alpha$, $\alpha = m/k$, will have full rank (see, for example, [15]). This probability quickly approaches 1 with decreasing message length m or with increasing p (for fixed m and k) because $k > m$.

For k/m close to 1 ($m \sim k$), the probability that $\text{rank}(\mathbf{H}) < p$ may become large enough to encounter a failure to embed all p bits in some blocks. For example, for $p = 18$ and $k/m = 2$, $n = 10^6$, $k = 50,000$, the probability of failure is about 0.0043. The fact that the number of columns in $\mathbf{H}$ varies from block to block also contributes to failures. We note that the probability of failure very quickly decreases with increasing k/m.

To make the method applicable to as wide range of the parameters k, n, and m as possible, the encoder needs to communicate the number of bits embedded in each block. Let us assume k, n, and m are fixed. For the i-th block, let p_i be the largest integer for which the first p_i rows of $\mathbf{H}$ form a matrix of rank p_i. Furthermore, let $f(q), q = 0, \ldots, p-1, p$, be the probability distribution of p_i over the blocks and random matrices $\mathbf{H}$. The information necessary to communicate p_i is $H(f)$, the entropy of f. Denoting by $E\{f\}$ the mean value of the distribution f, the average number of bits that can be encoded per block is thus $E\{f\} - H(f) \leq p$. Thus, the pure payload $m' = m(E\{f\} - H(f))/p$ that can be embedded is slightly smaller than m. From Table I, we show the embedding capacity loss for some typical values of m/k. We see that while this loss is negligible for $m/k \leq 0.6$, it imposes a limit on the maximal relative message length that can be embedded using this method to $\alpha_{\text{max}} = m'/k < 0.698$. In other words, payloads longer than roughly 70% of the maximal embeddable message cannot be embedded using this approach.

From the practical point of view, the sequence p_i should be compressed[2] and then embedded, for example, one bit per block, as the first bit in each block. The decoder first extracts

[2]In practice, the compressed bit-stream will be slightly larger than $H(f)$. Since f is not known to the decoder beforehand, adaptive coders, such as adaptive arithmetic coder, can be used.

TABLE I

DECREASE OF EMBEDDING CAPACITY DUE TO NON-SOLVABILITY OF (3) AS A FUNCTION OF THE RELATIVE MESSAGE LENGTH m/k. THE VALUES WERE OBTAINED EXPERIMENTALLY FOR A COVER IMAGE WITH $n = 10^6$ PIXELS, $k = 50,000$ RANDOMLY SELECTED CHANGEABLE PIXELS, AND $p = 18$.

m/k	0.3	0.4	0.5	0.6	0.7	0.8	0.9
m'/k	0.3	0.4	0.5	0.591	0.660	0.696	0.698

TABLE II

EMBEDDING TIME IN SECONDS FOR DIFFERENT RELATIVE MESSAGE LENGTH $\alpha = m/k$ FOR VARIOUS VALUES OF p. THE VALUES WERE OBTAINED EXPERIMENTALLY FOR A COVER IMAGE WITH $n = 10^6$ PIXELS AND $k = 50,000$ CHANGEABLE PIXELS.

m/k	0.1	0.2	0.25	0.33	0.5
$p = 17$	0.73	2.29	2.30	2.00	2.24
$p = 18$	1.17	4.58	4.19	3.58	3.80
$p = 19$	5.28	10.35	7.99	6.59	9.05
$p = 20$	15.74	17.37	12.82	10.19	18.68

p bits from each block, decompresses the bit sequence formed by the first bits from each block, reads p_i for all blocks, and then discards $p - p_i$ bits from the end of each block message chunk together with the first bit.

In general, the embedding efficiency improves[3] with the increasing value of p. However, a practical limit on the largest usable p is imposed by the exponentially increasing complexity and memory requirements. Table II shows the embedding time for a one-mega-pixel image (image with $n = 10^6$ pixels), for $k = 50,000$ changeable pixels, for some values of p on a PC equipped with a 3.4 GHz Intel Pentium IV processor. From our simulations, we recommend $p \leq 19$ to keep the embedding time of the order of seconds.

V. EMBEDDING EFFICIENCY

We know from Section III that with increasing code length, random linear codes asymptotically achieve the theoretical upper bound (6) on the embedding efficiency. However, the computational complexity of the proposed coding method imposes a limit on the practically usable code length. In this section, we derive an approximate but sufficiently accurate expression for the embedding efficiency of the proposed method.

Given two integers p and n, let $\mathcal{H}(p,n)$ be an ensemble of all binary matrices of dimension $p \times n$ with n different non-zero columns. The average number of embedding changes for a given matrix $\mathbf{H} \in \mathcal{H}(p,n)$ is the average distance R_a to the code represented by $\mathbf{H}$ (here calculated in the syndrome space using the sets C_i defined in Section IV-A)

$$R_a = 2^{-p}(|C_1| + 2|C_2| + \cdots + R|C_R|). \tag{7}$$

Let $c_i(p,n)$, $i = 1, \ldots, p$, be the expected value of $|C_i|/2^p$ over matrices $\mathbf{H}$ drawn uniformly from $\mathcal{H}(p,n)$. The expected value of R_a over matrices $\mathbf{H}$ drawn uniformly from $\mathcal{H}(p,n)$ is denoted $r_a(p,n) = \sum_{i=1}^{p} i c_i(p,n)$. We know that $c_1 = n$ and from the lemma below we calculate c_2. The remaining c_i for $i > 2$ will be obtained approximately using a recurrent formula.

Lemma 3: Let $\mathcal{C}$ be a collection of subsets of $\mathbb{F}_2^p$ consisting of n different non-zero vectors. Then,

$$2^p c_2 = E\{|C + C - \{0\}|\} = (2^p - 1)\left(1 - \prod_{i=1}^{n-1} \frac{2^p - 2i}{2^p - i}\right),$$

where the expected value is taken over all uniformly chosen $C \in \mathcal{C}$.

Proof: Each non-zero $\mathbf{x} \in \mathbb{F}_2^p$ can be written as sum of two vectors $\mathbf{a}_i$ and $\mathbf{b}_i$, $\mathbf{a}_i < \mathbf{b}_i$, in 2^{p-1}different ways, where the ordering of $\mathbb{F}_2^p$ is, for example, lexicographic. Let M be a $2^{p-1} \times 2$ array of vectors, $M_{i,1} = \mathbf{a}_i, M_{i,2} = \mathbf{b}_i$. We now count how many different n-tuples of vectors from $\mathbb{F}_2^p$ exist such that no sum of two gives $\mathbf{x}$. There are two types of such n-tuples—those containing $\mathbf{x}$ and those not containing $\mathbf{x}$. If the n-tuple contains $\mathbf{x}$, then we can choose the remaining $n-1$ vectors from $2^{p-1} - 1$ rows of matrix M (skipping the row $\mathbf{0}, \mathbf{x}$). This gives us $\binom{2^{p-1}-1}{n-1}2^{n-1}$ n-tuples (the factor 2^{n-1} appears because in each row, we can select either $\mathbf{a}_i$ or $\mathbf{b}_i$). If the n-tuple does not contain $\mathbf{x}$, we select all n elements from among $2^{p-1} - 1$ rows of M, giving us $\binom{2^{p-1}-1}{n}2^n$ n-tuples. Thus, there are total $N_{p,n} = \binom{2^{p-1}-1}{n-1}2^{n-1} + \binom{2^{p-1}-1}{n}2^n$ n-tuples of vectors for which no sum of two gives $\mathbf{x}$. We can calculate the probability that a given non-zero vector belongs to $C + C$ where C is randomly chosen from $\mathcal{C}$, $\Pr(\mathbf{x} \notin C + C) = N_{p,n}/|\mathcal{C}|$, where $|\mathcal{C}| = \binom{2^p-1}{n}$. Thus, (the last expression is obtained after some simple algebra)

$$\begin{aligned}
E\{|C + C - \{0\}|\} &= \frac{1}{|\mathcal{C}|}\sum_{C \in \mathcal{C}} |C + C - \{0\}| \\
&= \frac{1}{|\mathcal{C}|}\sum_{\mathbf{x}, \mathbf{x} \neq \mathbf{0}} |\{C \in \mathcal{C} | \mathbf{x} \in C + C\}| \\
&= (2^p - 1)\Pr(\mathbf{x} \in C + C) \\
&= (2^p - 1)(1 - \Pr(\mathbf{x} \notin C + C)) \\
&= (2^p - 1)\left(1 - \prod_{i=1}^{n-1} \frac{2^p - 2i}{2^p - i}\right).
\end{aligned}$$

■

The expected values $c_i(p,n)$, for $i > 2$, will be calculated only approximately. Let U_i be the set of all vectors in $\mathbb{F}_2^p$ that can be generated by adding i or fewer columns of a given matrix $\mathbf{H} \in \mathcal{H}(p,n)$. Then, $C_i = U_i - U_{i-1} = C_i^\star - U_{i-1}$, where $C_i^\star$ is the set of all vectors obtainable by adding exactly i different columns of $\mathbf{H}$. There are up to $\binom{n}{i}$ vectors in $C_i^\star$. We now make a simplifying assumption that $C_i^\star$ is obtained by random sampling of $\binom{n}{i}$ elements with replacement from the set $\{0, 1, \ldots, 2^p\}$. Under this assumption, $E\{|C_i^\star|\} = \text{Ball}\left(\binom{n}{i}, 2^p\right)$, where $\text{Ball}(k,N)$ denotes the expected number of occupied bins after throwing k balls in N bins, $\text{Ball}(k,N) = N - N(1 - 1/N)^k$. It also follows from this assumption that among all $|C_i^\star|$ vectors there will be on average $|C_i^\star| \times |U_{i-1}|/2^p$ vectors in U_{i-1}. Denoting

[3]Detailed analysis of how the embedding efficiency depends on p is in Section V.

with lower case letters the expected values of cardinalities of corresponding sets divided by 2^p, we write the following recurrent formula for $c_i(p,n)$, $u_i(p,n) = E\{2^{-p}|U_i|\}$, and $c_i^\star(p,n) = E\{2^{-p}|C_i^\star|\}$ for $i = 3,\dots,p$ (we skip the arguments p, n for better readability)

$$\begin{aligned} c_i^\star &= \text{Ball}\left(\binom{n}{i}, 2^p\right) \\ u_i &= c_i^\star + u_{i-1} - c_i^\star u_{i-1} \\ c_i &\approx u_i - u_{i-1}, \end{aligned} \tag{8}$$

supplied with the initial conditions

$$\begin{aligned} c_1 &= n2^{-p} \\ u_1 &= c_1 \\ c_2 &= (1-2^{-p})\left(1 - \prod_{i=1}^{n-1}\frac{2^p-2i}{2^p-i}\right) \\ u_2 &= c_1 + c_2. \end{aligned}$$

The expression for c_2 follows from Lemma 3.

Having obtained $c_i(p,n)$, we can now calculate the average number of embedding modifications of the algorithm from Section IV-A. The number of changeable pixels in each block is a random variable κ that follows hyper-geometric distribution

$$\Pr(\kappa = j) = \frac{\binom{k}{j}\binom{n-k}{n/n_B-j}}{\binom{n}{n/n_B}},\ j = 0,\dots,n/n_B. \tag{9}$$

Thus, the average number of embedding changes is

$$\begin{aligned} R_a(p) &= E\{r_a(p,\kappa)\} = E\left\{\sum_{i=1}^{p} i c_i(p,\kappa)\right\} \\ &= \sum_{j=1}^{n/n_B}\sum_{i=1}^{p} i c_i(p,j)\Pr(\kappa = j). \end{aligned} \tag{10}$$

Finally, the average embedding efficiency is

$$e(p) = p/R_a(p). \tag{11}$$

We have verified the accuracy of (10) using computer experiments in the range $5 \le p \le 18$ and $2 \le k/m \le 15$. In this range, it was possible to calculate $|C_i|$ in each block, averaging over $10,000$ blocks (for $n = 10^6$ and $k = 50,000$). The difference Δ between $R_a(p)$ obtained using simulations and using (10) is shown in Figure 1. We see that Δ is the largest for small values of p and k/m. We can see, however, that the accuracy quickly improves with increasing p and becomes less than 10^{-2} for $p > 10$ across the whole tested range of the relative message length α. We will use this formula to explore the properties of the proposed embedding algorithm for a wider range of p.

VI. Experiments and their interpretation

Figure 2, shows the embedding efficiency as a function of the ratio $\alpha^{-1} = k/m$ for a cover image with $n = 10^6$ pixels and $k = 50,000$ changeable pixels for $p = 4,\dots,20$. It was obtained by averaging over 100 embeddings of a random message bit-stream in the same cover image with the same parameters k, n, and m. The solid curve is the asymptotic upper bound (6).

The efficiency increases with shorter messages for a fixed p. Once the number of changeable pixels in each set exceeds 2^p, the embedding efficiency starts saturating at $p/(1-2^{-p})$, which is the value that all curves in Figure 2 reach asymptotically with decreasing α. This is because the p/α columns of $\mathbf{H}$ eventually cover the whole space $\mathbb{F}_2^p$ and thus we embed every non-zero syndrome $\mathbf{s} \neq \mathbf{0}$ using one embedding change (when $\mathbf{s} = \mathbf{0}$ no embedding changes are necessary).

Notice that for fixed α, the embedding efficiency increases in a curious non-monotone manner with increasing p. To see this interesting phenomenon more clearly, we plot e as a function of p for various fixed relative message lengths α. The result is shown in Figure 3. The diagram shows the expected value of embedding efficiency obtained from (11) as a function of $p = 4,\dots,80$. Each curve corresponds to a different value of $\alpha = 1/2, 1/3, \dots, 1/200$. The diagram was generated using the approximate formula (8) because it is not computationally feasible to obtain accurate estimates of c_i by direct calculation for such large values of p.

We see from Figure 3 that with increasing value of p the embedding efficiency increases and reaches the asymptotic value given by the bound (6). However, this increase is not monotone. In fact, it is not always true that increasing p will improve the embedding efficiency. For $p = 19$ and $\alpha = 1/10$ (embedding at 10% of embedding capacity), we obtain an improvement only after we increase p beyond 24. Without this knowledge, we may increase p from 19 to 22 hoping to improve the performance because, in general, increasing p improves embedding efficiency. However, in this case we only increase the embedding time while the embedding efficiency, in fact, decreases!

We now provide brief qualitative explanation of the origin of the non-monotone behavior as well as some quantitative description of the diagram. Due to space limitations, details of the exposition are sometimes omitted, only outlining the main ideas.

For $\mathbf{H} \in \mathcal{H}(p, p/\alpha)$ and $i \le p$, $|C_i| \le \binom{p/\alpha}{i} \le 2^{\frac{p}{\alpha}H(i\alpha/p)}$. For any small $\epsilon > 0$, let $i(\epsilon) = (1-\epsilon)\frac{p}{\alpha}H^{-1}(\alpha)$. Thus,

$$|C_{i(\epsilon)}| \le 2^{\frac{p}{\alpha}H\left((1-\epsilon)H^{-1}(\alpha)\right)} = 2^p 2^{-\frac{p}{\alpha}\epsilon H'(\xi)},$$

where $\xi \in (H^{-1}(\alpha)-\epsilon, H^{-1}(\alpha))$ from Taylor expansion of $H(x)$ at $H^{-1}(\alpha)$. Thus, with $p \to \infty$ we have $c_{i(\epsilon)}(p, p/\alpha) \le 2^{-\frac{p}{\alpha}\epsilon H'(\alpha)} \to 0$ because $H'(x) > 0$ is decreasing on $[0, 1/2)$. A little more careful argument can be made to show that $E\{c_{i(\epsilon)}(p,\kappa)\} \to 0$, where κ is the random variable (9).

As the upper bound on $c_{i(\epsilon)}$ is exponential in p and because we know that $\sum_{i=1}^{p} c_i = 1$, we can say that with increasing p, the peak of the distribution c_i moves to

$$k \doteq \frac{p}{\alpha}H^{-1}(\alpha), \tag{12}$$

which is the asymptotic covering radius of matrices from $\mathcal{H}(p,\kappa)$. This also implies that the embedding efficiency $e = p/R_a(p) \to \frac{\alpha}{H^{-1}(\alpha)} = \lambda$ with $p \to \infty$.

The wave character of $e(p)$ is caused by the corresponding wave character exhibited by R_a. To obtain an insight, inspect

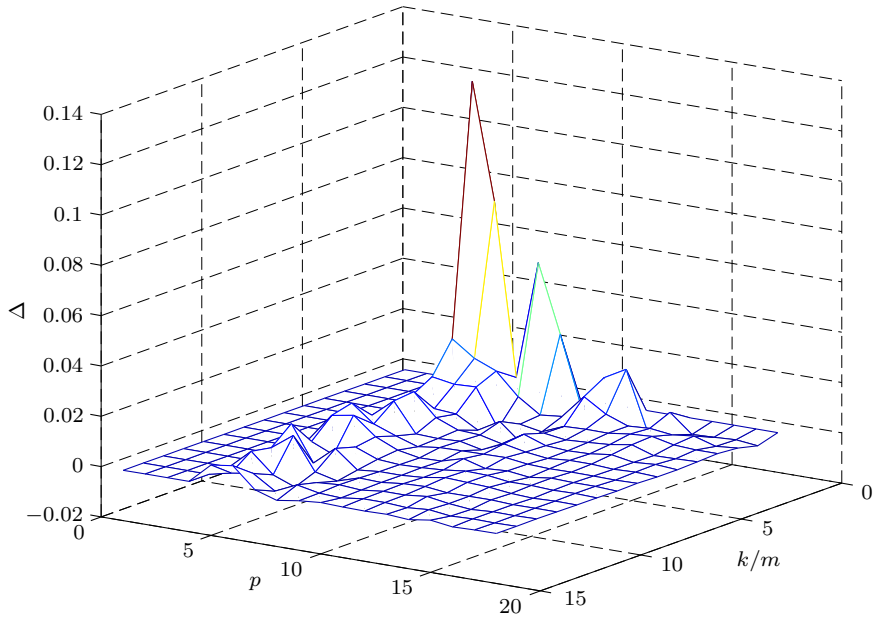

Fig. 1. Error Δ of the approximate formula (10) for the average number of embedding changes as a function of the relative message length $\alpha = m/k$ and parameter p.

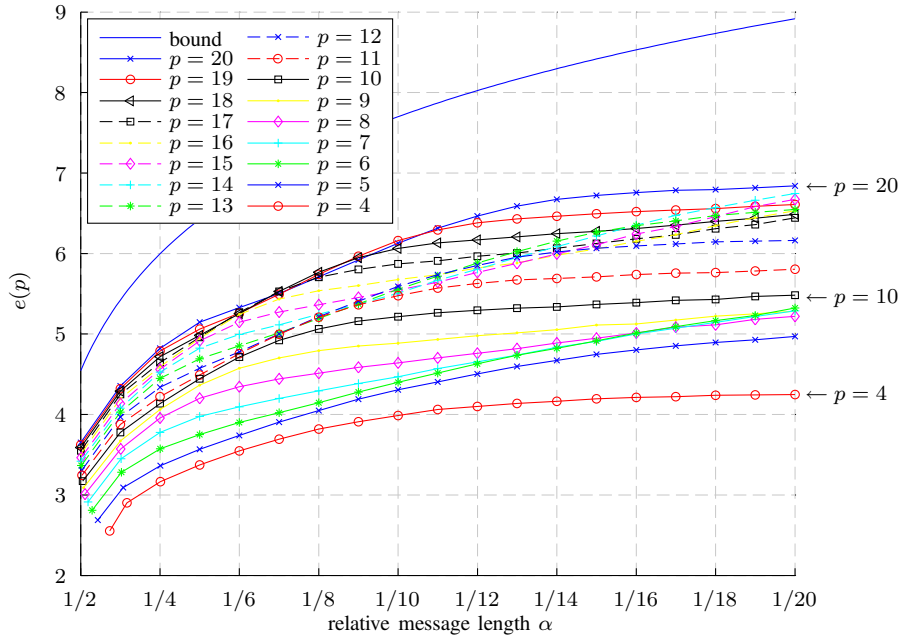

Fig. 2. Embedding efficiency $e(p)$ as a function of relative message length α for $p = 4, \ldots, 20$.

Figure 4, which shows a zoomed portion of one "wave" exhibited by $e(p)$ for $\alpha = 1/200$ and the corresponding distribution $c_i(p, p/\alpha)$. Note that the local maxima (the circled points No. 1, 7 in the figure) attained for $p = 34$ and $p = 47$ correspond to cases when $c_3(p, p/\alpha) \approx 1$ and $c_4(p, p/\alpha) \approx 1$, respectively. In these cases, the average distance to code R_a is very close to an integer and the distribution c_i undergoes very small changes. As a result, the growth in R_a is the smallest and thus the ratio p/R_a experiences a peak. With p increasing from $p = 34$ to 47, R_a quickly starts moving from 3 to 4. Because $e = p/R_a$, the decrement in e is largest when the increment in R_a is the largest (point No. 3), which occurs when the peaks switch their places. After they switch places, the changes to the distribution c_i become very subtle again and during this period e experiences growth.

Because the local peaks in e correspond to cases when R_a is an integer, the peaks lie on lines $y = p/k$ for k integer. The lines for $k = 1, 2$, and 3 are shown in Figure 3. Finally, since for large p, $R_a \approx R \approx p/\alpha H^{-1}(\alpha) = p/\lambda$, and because at the peaks R_a is an integer, the peaks will asymptotically occur at $p_k = k\lambda$, which gives an asymptotic expression for the "wavelength" $p_{k+1} - p_k \to \lambda$. This completes the quantitative explanation of Figure 3.

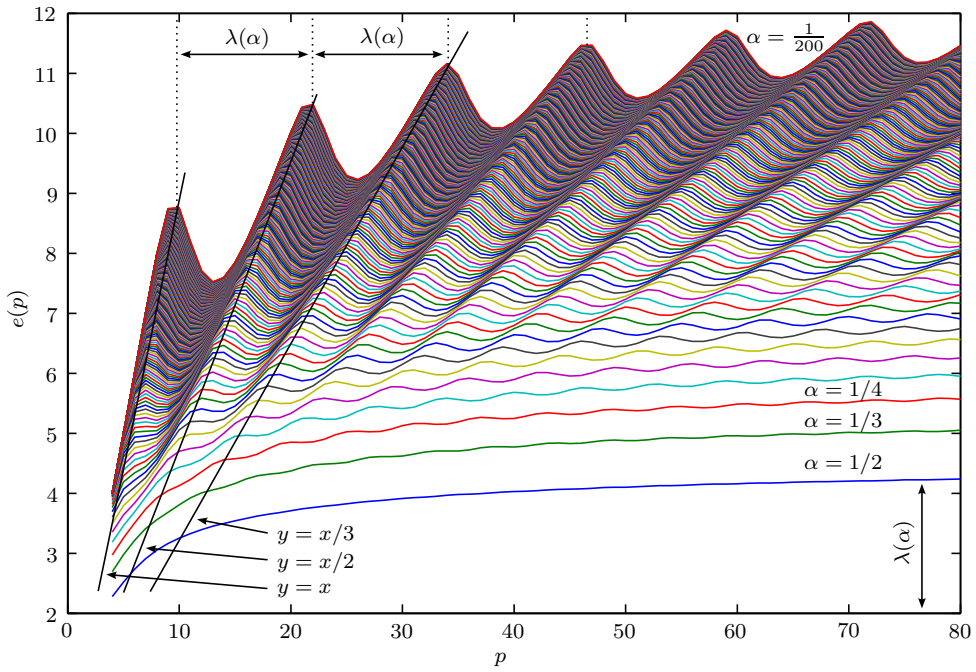

Fig. 3. Embedding efficiency $e(p)$ as a function of p for relative message length$\alpha = 1/2, 1/3, \ldots, 1/200$.

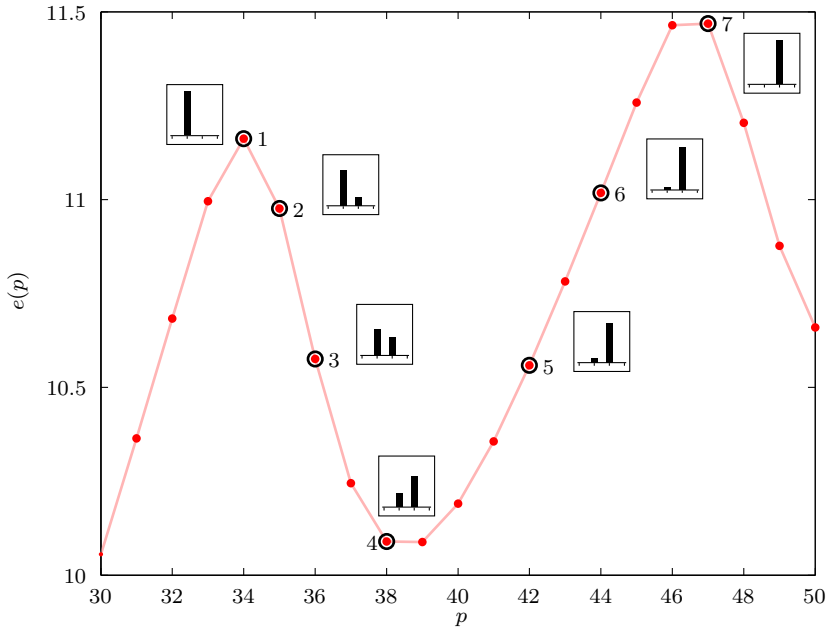

Fig. 4. Enlarged portion of Figure 3 for embedding efficiency $e(p)$ in the range $p \in [30, 50]$ for $\alpha = 1/200$. The distribution c_i for different values of p is shown in small boxes.

VII. Applications

The proposed approach to wet paper codes provides an efficient, general, and elegant tool to solve the problem of non-shared selection channel, which is quite common in steganography. For example, the sender can now use an arbitrary side information, such as a high-resolution version of the cover, for selecting the placement of embedding changes within the cover image and further minimize the embedding distortion. In Perturbed Quantization embedding (PQ) proposed in [16], the sender only selects those pixels/DCT coefficients in the image file whose unquantized values lie close to the middle of quantization intervals. This selection channel helps minimize the total distortion due to quantization (e.g., as in lossy compression) *and* embedding. The concept of PQ has been shown to produce steganographic schemes with substantially improved security [16].

Another application that greatly benefits from the proposed method is adaptive steganography. In adaptive steganography, the pixels are selected for embedding based on their local context (neighborhood). However, the act of embedding itself will modify the cover image and thus the recipient may not be able to identify the same set of message-carrying pixels from the stego image. This problem becomes especially pronounced and hard to overcome for data hiding in binary images. This is why we selected this application to demonstrate the benefits of the proposed method.

Binary images contain pixels of only two colors—black and white. Thus, they provide little space for data hiding. In particular, the embedding changes must be confined to portions of the boundary between both colors. In [17], Wu et al. proposed a measure that captures the perceptual impact of changes in binary images. This measure, called flippability score, assigns to each pixel a scalar score calculated from its 3×3 neighborhood. Ideally, the sender should only modify pixels with the highest score to avoid introducing disturbing artifacts.

In Figure 5, we show an example of a binary image "Clinton's signature." It is a 48×288 image similar to what may be captured when a person signs a credit card bill on a digitized pad at a grocery store. Data embedding can help secure the digitized signature against malicious tampering by embedding a Message Authentication Code (MAC) into the image itself. Using Wu's flippability measure, the image contains 250 pixels with flippability score 0.625, 32 pixels with score 0.375, 3 pixels with 0.25, 86 with score 0.125, 382 with score 0.1, 662 with 0.05, 71 with 0.01, and the remaining 12338 pixels have score 0.

Let us suppose that we intend to embed a 64-bit MAC, together with 16 bit header data into the image. This makes the total payload $m = 80$ bits. Also, we prefer to only use the pixels with the highest flippability score of 0.625. Thus, using the notation established in this paper, there are $k = 250$ changeable pixels and $n = 48 \times 288 = 13824$ total pixels. The relative message length $\alpha = m/k = 80/250 = 0.32$.

We stress that for this binary embedding, application of wet paper codes is a necessity because the act of embedding may change the flippability score of many pixels. From Figure 2, we see that for $p = 18$ and $\alpha = 0.32$ we can embed with embedding efficiency of roughly $e(18) = 4.4$. This means that wet paper codes with improved embedding efficiency would introduce about $80/4.4 \doteq 18$ embedding changes. This should be contrasted with regular wet paper codes that would introduce about $80/2 = 40$ changes.

While this decrease in embedding distortion is quite substantial by itself, we point out another important consequence of improved embedding efficiency. We can now embed up to $2.2 \times 80 = 176$ bits with the same embedding distortion as the one due to embedding 80 bits using regular wet paper codes. Thus, instead of decreasing the embedding distortion, we may choose to improve the robustness of embedded data by applying strong error correction code to the payload. Therefore, the improved embedding efficiency can be utilized either for decreasing the visual impact of embedding or to improve the robustness of the embedded data to channel noise.

VIII. Conclusions

In this paper, we combine wet paper codes with matrix embedding, providing a general tool for constructing steganographic schemes with arbitrary selection channels and improved embedding efficiency. This is achieved using random linear codes of small codimension, where the coding can be done using efficient exhaustive searches (the meet-in-the-middle method). We derive an approximate formula for the embedding efficiency of the proposed method, compare it with embedding efficiency obtained experimentally, and evaluate its performance with respect to theoretically achievable bounds.

While analyzing the performance of the proposed method, we have discovered a curious transient phenomenon. While the embedding efficiency globally increases with increasing code block length and eventually reaches the theoretical upper bound, it does so in a non-monotone manner. Because understanding this transient phenomenon is important for practical implementations, we analyze and quantify the non-monotone behavior, providing insight that practitioners might find valuable.

We also discuss how the proposed coding method improves steganographic security within the context of Perturbed Quantization and for data hiding in binary images. Improved embedding efficiency can be used to decrease the embedding distortion or to improve robustness of data hiding schemes to channel noise. This is because instead of decreasing the distortion, we can embed larger amount of bits and apply strong error correction algorithms to the embedded payload.

In our future effort, reflecting on our previous work on application of LT codes to wet paper codes, we plan to investigate low density parity check codes and their iterative decoding algorithms with the intention to obtain good quantizers suitable for steganography with non-shared selection channels with improved embedding efficiency.

IX. 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

Fig. 5. Binary 48×288 image "Clinton's signature."

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 Petr Lisoněk and Alexander Barg for many useful discussions.

REFERENCES

[1] A. Westfeld, "High capacity despite better steganalysis (F5—A steganographic algorithm)," in *Proceedings, Information Hiding: 4th International Workshop, IHW 2001* (I. S. Moskowitz, ed.), vol. 2137 of *Lecture Notes in Computer Science*, (Pittsburgh, PA, USA), pp. 289–302, Springer-Verlag, Apr. 25–27 2001.

[2] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed Quantization steganography using Wet Paper Codes," in *MM&Sec '04: Proceedings of the 2004 multimedia and security workshop on Multimedia and security* (J. Dittmann and J. Fridrich, eds.), Proceedings of ACM, (New York, NY, USA), ACM Press, Dec. 6 2004.

[3] A. Kuznetsov and B. Tsybakov, "Coding in a memory with defective cells," vol. 10, pp. 132–138, 1974.

[4] R. Crandall, "Some notes on steganography." Posted on Steganography Mailing List. `http://os.inf.tu-dresden.de/~westfeld/crandall.pdf`, 1998.

[5] J. Bierbrauer, "On Crandall's problem." Personal communication, 1998.

[6] F. Galand and G. Kabatiansky, "Information hiding by coverings," in *Proc. ITW2003*, (Paris, France), pp. 151–154, 2003.

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

[8] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on wet paper," in *IEEE Trans. on Sig. Proc., Third Supplement on Secure Media*, vol. 53, pp. 3923–3935, Oct. 2005.

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

[10] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," in *submitted to IEEE Transactions on Information Security and Forensics*, 2005.

[11] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inf. Theory*, vol. 24, pp. 384–386, May 1978.

[12] J. Fridrich, M. Goljan, and D. Soukal, "Efficient Wet Paper Codes," in *Proceedings, Information Hiding: 7th International Workshop, IHW 2005*, Lecture Notes in Computer Science, (Barcelona, Spain), Springer-Verlag, 2005.

[13] M. Luby, "LT codes," in *Proc. The 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–282, Nov. 16–19 2002.

[14] T. Wadayama, "An algorithm for calculating the exact bit error probability of a binary linear code over the binary symmetric channel," in *IEEE Trans. Inform. Theory*, vol. 50, pp. 331–337, Feb. 2004.

[15] R. Brent, S. Gao, and A. Lauder, "Random Krylov spaces over finite fields," in *SIAM J. Discrete Math.*, vol. 16(2), pp. 276–287, 2003.

[16] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed Quantization steganography," in *to appear in ACM Multimedia & Security Journal*, 2005.

[17] M. Wu and B. Liu, "Data hiding for binary image for authentication and annotation," in *IEEE Trans. on Multimedia*, vol. 6, pp. 528–538, Aug. 2004.