$\mathbb{Z}_2\mathbb{Z}_4$-ADDITIVE PERFECT CODES IN STEGANOGRAPHY

Helena Rifà-Pous

Department of Computer Science and Multimedia
Universitat Oberta de Catalunya, 08018-Barcelona, Spain

Josep Rifà and Lorena Ronquillo

Department of Information and Communications Engineering
Universitat Autònoma de Barcelona, 08193-Bellaterra, Spain

Abstract. Steganography is an information hiding application which aims to hide secret data imperceptibly into a cover object. In this paper, we describe a novel coding method based on $\mathbb{Z}_2\mathbb{Z}_4$-additive codes in which data is embedded by distorting each cover symbol by one unit at most (± 1-steganography). This method is optimal and solves the problem encountered by the most efficient methods known today, concerning the treatment of boundary values. The performance of this new technique is compared with that of the mentioned methods and with the well-known rate-distortion upper bound to conclude that a higher payload can be obtained for a given distortion by using the proposed method.

1. Introduction and preliminary results

Steganography is a scientific discipline within *data hiding*, which hides information imperceptibly into innocuous media. A comprehensive overview of the core principles and the mathematical methods that can be used for data hiding can be found in [6].

An interesting steganographic method is known as *matrix encoding*, introduced by Crandall [3] and analyzed by Bierbrauer et al. [1]. Matrix encoding requires the sender and the recipient to agree in advance on a parity check matrix H, and the secret message is then extracted by the recipient as the syndrome (with respect to H) of the received cover object. This method was made popular by Westfeld [9], who incorporated a specific implementation using Hamming codes. The resulting method is known as the F5 algorithm and it can embed t bits of message in $2^t - 1$ cover symbols by changing, at most, one of them.

There are several parameters which are used to evaluate the performance of a steganographic method over a cover message of N symbols: the *average distortion* $D = \frac{R_a}{N}$, where R_a is the expected number of changes over uniformly distributed messages; the *embedding rate* $E = \frac{t}{N}$, which is the amount of bits that can be hidden in a cover message; and some authors use instead the *embedding efficiency*, which is the average number of embedded bits per change. In our case we will use

the average distortion and the embedding rate. Given two methods with the same embedding rate, the one with smaller average distortion will be said to perform better than the other. A scheme with block length N, embedding rate E, and average distortion D is called *optimal*, if all other schemes with the same block length N have embedding rate $E' \leq E$ or average distortion $D' \geq D$. Following the terminology used by Fridrich et al. [4], the tuple (D, E) will be called *CI-rate*.

As Willems et al. in [10], we will also assume that a discrete source produces a sequence $\mathbf{x} = (x_1, \ldots, x_N)$, where N is the block length, $x_i \in \aleph = \{0, 1, \ldots, 2^B - 1\}$, and $B \in \{8, 12, 16\}$ depends on the kind of source. The secret message $\mathbf{s} \in \{1, \ldots, M\}$ produces a composite sequence $\mathbf{y} = f(\mathbf{x}, \mathbf{s})$, where $\mathbf{y} = (y_1, \ldots y_N)$ and each $y_i \in \aleph$, by distorting $\mathbf{x}$. This distortion will be assumed to be of squared-error type (see [10]). In these conditions, we may deal with "binary steganography", in which information is carried by the least significant bit (LSB) of each x_i and the appropriate solution comes from using binary Hamming codes [9], later improved using product Hamming codes [7]; or we may deal with "± 1-steganography", where $y_i = x_i + c$ for $c \in \{0, +1, -1\}$ and the information is carried by the two LSBs of x_i. Let the absolute value of c be the *amplitude* of an embedding change.

There are some steganographic techniques [8] in which messages carrying hidden information are statistically indistinguishable from those not carrying hidden data. However, in general, the embedding becomes statistically detectable rather quickly with the increasing amplitude of embedding changes, and our interest goes to avoid changes of amplitude greater than one. With this assumption, the embedding rate of our ± 1-steganographic scheme will be compared with the upper bound $H(D) + D$ [10], where $H(D)$ is the binary entropy function $H(D) = -D \log_2(D) - (1 - D) \log_2(1 - D)$ and $0 \leq D \leq 2/3$ is the average distortion. One of the purposes of steganographers is designing schemes in order to approach this upper bound.

In most papers, ± 1-steganography has been treated using ternary codes. Willems et al. [10] proposed a scheme based on ternary Hamming and Golay codes, which were proved to be optimal except for a remark which exposed a problem related to boundary values. Fridrich et al. [4] proposed a method based on rainbow colouring graphs using q-ary Hamming codes, where q is a prime power. This method performed better than the scheme from [10] when q is not a power of 3. However, the authors of both methods suggest making a change of magnitude greater than one in order to avoid having to apply the change $x_i - 1$ and $x_i + 1$ to a host sequence of value $x_i = 0$ and $x_i = 2^B - 1$, respectively. Note that this would introduce larger distortion and therefore make the embedding more detectable. The treatment of boundary grayscale values in steganography is important and, as far as we know, not many papers have paid attention to this issue.

In this paper we also consider ± 1-steganography. Our new method is based on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes which, although they are not linear, they have a representation using a parity check matrix that makes them as computationally efficient as the Hamming codes. As we will later show, this new method is optimal and performs better than the method obtained by direct sum of ternary Hamming codes from [10] and the method based on rainbow colouring of graphs using q-Hamming codes [4] for the specific case $q = 3$. Furthermore, the proposed method also deals better with boundary grayscale values, because the magnitude of embedding changes is under no circumstances greater than one.

To make this paper self-contained, we review in Section 2 a few elementary concepts on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes, relevant for our study. The new steganograph-

ic method based on these codes is described in Section 3, whereas an improvement to better deal with the extreme grayscale values problem is given in Section 4. The paper is concluded in Section 5.

2. $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes

Any non-empty subgroup $\mathcal{C}$ of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ is a $\mathbb{Z}_2\mathbb{Z}_4$-*additive code*, where $\mathbb{Z}_2^\alpha$ denotes the set of all binary vectors of length α and $\mathbb{Z}_4^\beta$ is the set of all quaternary vectors of length β. Let $\Phi : \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta \to \mathbb{Z}_2^n$, where $n = \alpha + 2\beta$, be the *extended Gray map* given by applying the usual *Gray map* $\phi(0) = (0,0)$, $\phi(1) = (0,1)$, $\phi(2) = (1,1)$, and $\phi(3) = (1,0)$ to the quaternary coordinates.

A $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathcal{C}$ is isomorphic to an abelian structure like $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$. Therefore, $\mathcal{C}$ has $|\mathcal{C}| = 2^\gamma 4^\delta$ codewords, where $2^{\gamma+\delta}$ of them are of order two. We call such code $\mathcal{C}$ a $\mathbb{Z}_2\mathbb{Z}_4$-*additive code of type* $(\alpha, \beta; \gamma, \delta)$ and its binary image C is a $\mathbb{Z}_2\mathbb{Z}_4$-*linear code of type* $(\alpha, \beta; \gamma, \delta)$ which may not be linear. Note that the Lee distance of $\mathcal{C}$ coincides with the Hamming distance of $C = \Phi(\mathcal{C})$.

The $\mathbb{Z}_2\mathbb{Z}_4$-*additive dual code* of $\mathcal{C}$, denoted by $\mathcal{C}^\perp$, is defined as the set of vectors in $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ that are orthogonal to every codeword in $\mathcal{C}$, where the inner product in $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ is defined by:

$$\langle u, v \rangle = 2\left(\sum_{i=1}^{\alpha} u_i v_i\right) + \sum_{j=\alpha+1}^{\alpha+\beta} u_j v_j \in \mathbb{Z}_4, \tag{1}$$

where $u, v \in \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$ and computations are made considering the zeros and ones in the α binary coordinates as quaternary zeros and ones, respectively.

The binary code $C_\perp = \Phi(\mathcal{C}^\perp)$, of length $n = \alpha + 2\beta$, is called the $\mathbb{Z}_2\mathbb{Z}_4$-*dual code* of C.

A $\mathbb{Z}_2\mathbb{Z}_4$-additive code $\mathcal{C}$ is said to be *perfect* if code $C = \Phi(\mathcal{C})$ is a perfect binary code, that is a binary code of minimum distance 3, where all vectors in $\mathbb{Z}_2^n$ are within distance one from a unique codeword.

It is well known [2] that for any $m \geq 2$ and each $\delta \in \{0, \ldots, \lfloor \frac{m}{2} \rfloor\}$ there exists a perfect $\mathbb{Z}_2\mathbb{Z}_4$-linear code C of binary length $n = 2^m - 1$, such that its $\mathbb{Z}_2\mathbb{Z}_4$-dual code is of type $(\alpha, \beta; \gamma, \delta)$, where $\alpha = 2^{m-\delta} - 1$, $\beta = 2^{m-1} - 2^{m-\delta-1}$ and $\gamma = m - 2\delta$ (note that the binary length can be computed as $n = \alpha + 2\beta$). This allows us to write the parity check matrix H of any $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect code for a given value of δ. Matrix H can be represented by taking as columns all possible vectors in $\mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$, up to sign changes. In this representation, there are α columns which correspond to the binary part of vectors in $\mathcal{C}$, and β columns of order four which correspond to the quaternary part. We agree on a representation of the α binary coordinates as coordinates in $\{0, 2\} \in \mathbb{Z}_4$. Note that the binary Hamming code is a particular case of perfect $\mathbb{Z}_2\mathbb{Z}_4$-linear codes, specifically, when $\beta = 0$.

3. Steganography based on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes

Let us take a $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect code and consider its additive dual, which is of type $(\alpha, \beta; \gamma, \delta)$. As stated in the previous section, this gives us a parity check matrix H which has γ rows of order two and δ rows of order four.

For instance, for $m = 4$ and according to [2], there are three different $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes of binary length $n = 2^4 - 1 = 15$ which correspond to the possible values of $\delta \in \{0, \ldots, \lfloor \frac{m}{2} \rfloor\} = \{0, 1, 2\}$. For $\delta = 0$, the corresponding $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect code is the usual binary Hamming code, while for $\delta = 2$ the

$\mathbb{Z}_2\mathbb{Z}_4$-additive perfect code has parameters $\alpha = 3$, $\beta = 6$, $\gamma = 0$, $\delta = 2$ and the following parity check matrix:

$$H = \left(\begin{array}{ccc|cccccc} 2 & 0 & 2 & 0 & 1 & 1 & 1 & 1 & 2 \\ 2 & 2 & 0 & 1 & 0 & 1 & 2 & 3 & 1 \end{array} \right). \tag{2}$$

Let $\mathbf{h}_i$, for $i \in \{1, \ldots, \alpha+\beta\}$, denote the i-th column vector of H. Note that the all twos vector $\mathbf{2}$ is always one of the columns in H and, for the sake of simplicity, it will be written as column $\mathbf{h}_1$. We group the remaining first α columns in H in such a way that, for any $2 \leq i \leq (\alpha+1)/2$, vector $\mathbf{h}_{2i}$ is paired up with its complementary vector $\bar{\mathbf{h}}_{2i} = \mathbf{h}_{2i+1}$, where $\bar{\mathbf{h}}_{2i} = \mathbf{h}_{2i} + \mathbf{2}$.

To use these $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes in steganography take $N = 2^{m-1} = \frac{\alpha+1}{2} + \beta$ and let $\mathbf{x} = (x_1, \ldots, x_N)$ be an N-length source of grayscale symbols such that $x_i \in \aleph = \{0, 1, \ldots, 2^B - 1\}$, where, for instance, $B = 8$ for grayscale images. We assume each grayscale symbol x_i is represented as a binary vector $(v_{(B-1)i}, \ldots, v_{1i}, v_{0i})$, obtained by first representing x_i in base 4 and then applying the Gray map ϕ to every quaternary symbol in that representation. For example, value 239 is represented as the quaternary vector (3233), which then gives rise to the binary vector (10111010) after applying ϕ. We will use the two least significant bits (LSBs), v_{1i}, v_{0i}, of every grayscale symbol x_i in the source, for $i > 1$, as well as the least significant bit v_{01} of symbol x_1 to embed the secret message.

Each grayscale symbol x_i will be associated with one or more columns $\mathbf{h}_i$ in H:

1. Symbol x_1 is associated with $\mathbf{h}_1$ by taking its least significant bit, v_{01}.
2. Symbol x_i, for $2 \leq i \leq (\alpha+1)/2$, is associated with $\mathbf{h}_i$ and $\bar{\mathbf{h}}_i$, by taking, respectively, the two least significant bits, v_{1i}, v_{0i}, of x_i.
3. Symbol x_j, for $\alpha < j \leq N$, is associated with $\mathbf{h}_{j+(\alpha-1)/2}$ by taking its two least significant bits v_{1j}, v_{0j} and interpreting them as $\phi^{-1}(v_{1j}, v_{0j})$ in $\mathbb{Z}_4$.

In this way, the N-length packet $\mathbf{x}$ of symbols is translated into a vector $\mathbf{w} \in \mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$. The embedding process we propose is based on the matrix encoding method. The secret message can be any vector $\mathbf{s} \in \mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$. Vector $\epsilon \cdot \mathbf{h}_i$ indicates the changes needed to embed $\mathbf{s}$ within $\mathbf{x}$; that is $H\mathbf{w}^T + \epsilon \cdot \mathbf{h}_i = \mathbf{s}$, where ϵ is an integer whose value will be described bellow, $H\mathbf{w}^T$ is the syndrome vector of $\mathbf{w}$ and $\mathbf{h}_i$ is a column vector in H. We may have the following situations, depending on which column $\mathbf{h}_i$ needs to be modified:

1. If $\mathbf{h}_i = \mathbf{h}_1$, then the embedder has to change the least significant bit of x_1 by adding or subtracting one unit to/from x_1, depending on which operation will flip its least significant bit, v_{01}.
2. If $\mathbf{h}_i$ is among the first α column vectors in H and $2 \leq i \leq \alpha$, then ϵ can only be $\epsilon = 1$. In this case, since $\mathbf{h}_i$ was paired up with its complementary column vector $\bar{\mathbf{h}}_i$, then this situation is equivalent to make $(v_{1i}, 1 + v_{0i})$ or $(1 + v_{1i}, v_{0i})$, where v_{1i} and v_{0i} are the least significant bits of the symbol x_i which had been associated with those two column vectors. Hence, after the inverse of Gray map, by changing one or another we are actually adding or subtracting one unit to/from x_i. Note that a problem may crop up at this point if we need to add 1 to a symbol x_i of value $2^B - 1$ or subtract 1 from a symbol of value 0.
3. If $\mathbf{h}_i$ is one of the last β columns in H, then this situation corresponds to add $\epsilon \in \{0, 1, 2, 3\}$. Note that because we are using a $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect code, ϵ will never be 2. Hence, the embedder should add ($\epsilon = 1$) or subtract ($\epsilon = 3$)

one unit to/from symbol $x_{i-(\alpha-1)/2}$. Once again, a problem may arise with boundary values.

Example 1. Let $\mathbf{x} = (239, 251, 90, 224, 226, 187, 229, 180)$ be an N-length source of grayscale symbols, where $x_i \in \{0, \ldots, 255\}$ and $N = 8$, and let H be the matrix in (2). The source $\mathbf{x}$ is then translated into the vector $\mathbf{w} = (010|202310)$ in the way specified above. Let $\mathbf{s} = (02)^T$ be the vector representing the secret message we want to embed in $\mathbf{x}$. We then compute $H\mathbf{w}^T = (23)^T$ and see, by the matrix encoding method, that $\epsilon = 3$ and $\mathbf{h}_i = \mathbf{h}_9$. According to the described method, we should subtract 1 from x_8. In this way, x_8 becomes 179, and then $\mathbf{w}' = (010|202313)$, which has the expected syndrome $(02)^T$.

The problematic cases related to boundary values are also present in methods from [4] and [10], but their authors assume that the probability of gray value saturation is not too large. We argue that, though rare, this gray saturation can still occur. However, in order to compare our proposal with these others we will not consider these problems either until next section. Therefore, we proceed to compute the values of the average distortion D and the embedding rate E.

Our method is able to hide any secret vector $\mathbf{s} \in \mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$ into the given N symbols. Hence, the embedding rate is $(\gamma + 2\delta)$ bits per N symbols, $E = \dfrac{\gamma + 2\delta}{N} = \dfrac{m}{2^{m-1}}$.

Concerning the average distortion D, we are using a perfect code of binary length $2^m - 1$, which corresponds to $N = 2^{m-1}$ grayscale symbols. There are $N - 1$ symbols x_i, for $2 \leq i \leq N$, with a probability $2/2^m$ of being subjected to a change; a symbol x_1 with a probability $1/2^m$ of being the one changed; and, finally, there is a probability of $1/2^m$ that neither of the symbols will need to be changed to embed the secret message $\mathbf{s}$. Hence, $D = \dfrac{2N - 1}{N2^m} = \dfrac{2^m - 1}{2^{2m-1}}$.

The described method has a CI-rate $(D_m, E_m) = \left(\dfrac{2N-1}{2N^2}, \dfrac{1 + \log(N)}{N}\right)$, where $N = 2^{m-1}$ and m is any integer $m \geq 2$.

It is shown in [10] that the linear ternary perfect codes (Hamming or Golay) are optimal in the sense that they achieve the smallest possible distortion at a given embedding rate for a fixed block length. This property is not exclusive of these codes and we will prove, in the next proposition, that the method we have described using $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes also satisfies it.

Proposition 1. *The proposed embedding method based on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes is optimal.*

Proof. Consider a code with length $N = \frac{\alpha+1}{2} + \beta$. Suppose that the source produces a sequence $\mathbf{x}$ and assume that we have a steganographic scheme with embedding rate $E_m = \frac{1+\log(N)}{N} = \frac{\log(2N)}{N}$. Hence, there are $2N$ composite sequences $\mathbf{y}$, each one of them representing a different message. There is only one sequence $\mathbf{x}$ which does not need to be distorted, that is $\mathbf{y} = \mathbf{x}$. The smallest possible nonzero distortion is $1/N$ and it is achieved by 2β sequences $\mathbf{y}$ which differ from $\mathbf{x}$ in exactly one of the β quaternary coordinates, after multiplying by 1 or 3, that is $|\mathbf{y} - \mathbf{x}| = 1$ or $|\mathbf{y} - 3\mathbf{x}| = 1$, and the same distortion is also achieved by a sequence $\mathbf{y}$ which differs from $\mathbf{x}$ in exactly one of the α binary coordinates (i.e. they differ in a bit). So there are $\alpha + 2\beta = 2N - 1$ composite sequences $\mathbf{y}$ achieving the smallest possible nonzero distortion $1/N$, and this gives us the smallest possible maximum average

distortion $D = \dfrac{(2N-1)(1/N)}{2N} = \dfrac{2N-1}{2N^2}$, which coincides with the distortion in our method. □

Note that we are only able to generate an embedding scheme for natural values of $m \geq 2$. However, we can use the direct sum of codes [5] to obtain codes whose CI-rates are convex combinations of CI-rates of boths codes. Thus given any non-allowable parameter D for the average distortion, we can take two codes with CI-rates (D_1, E_1) and (D_2, E_2), respectively, where $D_1 < D < D_2$, and their direct sum generates a code with a new CI-rate (D, E), with $D = \lambda D_1 + (1-\lambda)D_2$ and $E = \lambda E_1 + (1-\lambda)E_2$. From a graphic point of view, this is equivalent to draw a line between two contiguous points (D_1, E_1) and (D_2, E_2), as it is shown in Figure 1.

Proposition 2. *For $m \geq 4$, the CI-rate given by the method based on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes improves the CI-rate obtained by direct sum of ternary Hamming codes with the same average distortion.*

Proof. Optimal embedding (of course, in the allowable values of D) can be obtained by using ternary codes, as it is shown in [10]. The CI-rate of these codes is $(D_\mu, E_\mu) = \left(\dfrac{2}{3^\mu}, \dfrac{2\mu\log(3)}{3^\mu - 1}\right)$ for any integer μ. Our method, based on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes, has CI-rate $(D_m, E_m) = \left(\dfrac{2N-1}{2N^2}, \dfrac{1+\log(N)}{N}\right)$, for $N = 2^{m-1}$ and any integer $m \geq 2$.

Take, for any $m \geq 2$, two contiguous values for μ such that $D_{\mu+1} < D_m < D_\mu$ and write $D_m = \lambda D_{\mu+1} + (1-\lambda)D_\mu$, where $0 \leq \lambda \leq 1$.

We want to prove that, for $m \geq 4$, we have $E_m \geq \lambda E_{\mu+1} + (1-\lambda)E_\mu$, which is straightforward. However, since it is neither short nor contributes to the well understanding of the method, we do not include all computations here. □

4. Solving the extreme grayscale values problem

In Section 3 we described a problem which may arise when, according to our method, the embedder is required to add one unit to a source symbol x_i containing the maximum allowed value $(2^B - 1)$, or to subtract one unit from a symbol x_i containing the minimum allowed value, 0. To face this problem, we will use the complementary column vector $\bar{\mathbf{h}}_i$ of columns $\mathbf{h}_i$ in matrix H, where $\bar{\mathbf{h}}_i = 3\mathbf{h}_i + \mathbf{2}$ and $\mathbf{h}_i$ is among the last β columns in H. Note that $\mathbf{h}_i$ and $\bar{\mathbf{h}}_i$ can coincide.

The first α column vectors in H will be paired up as before, and the association between each x_i and each column vector $\mathbf{h}_i$ in H will be also the same as in Section 3. However, given an N-length source of grayscale symbols $\mathbf{x} = (x_1, \ldots, x_N)$, a secret message $\mathbf{s} \in \mathbb{Z}_2^\gamma \times \mathbb{Z}_4^\delta$ and the vector $\epsilon \cdot \mathbf{h}_i$, such that $H\mathbf{w}^T + \epsilon \cdot \mathbf{h}_i = \mathbf{s}$, indicating the changes needed to embed $\mathbf{s}$ within $\mathbf{x}$, we can now make some variations on the kinds of changes to be done for the specific problematic cases:

- If $\mathbf{h}_i$ is among the first α columns in H, for $2 \leq i \leq \alpha$, and the embedder is required to add 1 to a symbol $x_i = 2^B - 1$, then the embedder should instead subtract 1 from x_i as well as perform the appropriate operation ($+1$ or -1) over x_1 to have v_{01} flipped. Likewise, if the embedder is required to subtract 1 from a symbol $x_i = 0$, then (s)he should instead add 1 to x_i and also change x_1 to flip v_{01}.

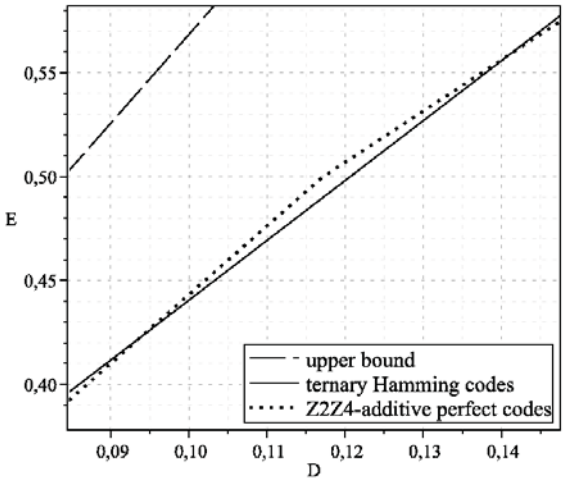

FIGURE 1. CI-rate (D, E), for $B = 8$, of steganographic methods based on ternary Hamming codes and on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes, compared with the upper bound $H(D) + D$, where E is the embedding rate, D is the average distortion and $H(D)$ is the binary entropy function.

- If $\mathbf{h}_i$ is one of the last β columns in H, and the embedder has to add 1 to a symbol $x_i = 2^B - 1$, (s)he should instead subtract 1 from the grayscale symbol associated to $\bar{\mathbf{h}}_i$ and also change x_1 to flip v_{01}. If the method requires subtracting 1 from $x_i = 0$, then we should instead add 1 to the symbol associated to $\bar{\mathbf{h}}_i$ and, again, change x_1 to flip v_{01}.

Example 2. Let $\mathbf{s}$ and $\mathbf{x}$ be as in Example 1, except for the value of x_8 which is now $x_8 = 0$. The packet $\mathbf{x}$ is translated into vector $\mathbf{w} = (010|202310)$. However, now we are not able to make $x_8 - 1$. Instead of this, we will add one unit to x_3, which is the symbol associated with $\bar{\mathbf{h}}_9 = \mathbf{h}_4$, and subtract one unit from x_1 so as to have its LSB flipped. Therefore, we obtain $\mathbf{x}' = (238, 251, 91, 224, 226, 187, 229, 0)$ and then $\mathbf{w}' = (110|302310)$

The method above described has the same embedding rate $E = \dfrac{m}{2^{m-1}}$ as the one from Section 3 but a slightly worse average distortion. We will take into account the squared-error distortion defined in [10] for our reasoning.

As before, among the total number of grayscale symbols $N = 2^{m-1}$, there are $N - 1$ symbols x_i, for $2 \leq i \leq N$, with a probability $2/2^m$ of being changed; a symbol x_1 with a probability $1/2^m$ of being the one changed; and, finally, there is a probability of $1/2^m$ that neither of the symbols will need to be changed.

As one may have noted in this scheme, performing a certain change to a symbol x_i, associated with a column $\mathbf{h}_i$ in H, has the same effect as performing the opposite change to the grayscale symbol associated with $\bar{\mathbf{h}}_i$ and also changing the least significant bit v_{01} of x_1. This means that with probability $\frac{2^B-2}{2^B}$ we will change a symbol x_i, for $2 \leq i \leq N$, a magnitude of 1; and with probability $\frac{2}{2^B}$ we will change two other symbols also a magnitude of 1. Therefore,

$R_a = (N-1)\frac{2}{2^m}\left(\frac{2^B-2}{2^B}+2\frac{2}{2^B}\right)+\frac{1}{2^m}$ and the average distortion is thus $D = \frac{2N-1+\frac{N-1}{2^{B-2}}}{N2^m}$. Hence, the described method has CI-rate:

$$(D_m, E_m) = \left(\frac{2N-1+\frac{N-1}{2^{B-2}}}{2N^2}, \frac{1+\log(N)}{N}\right).$$

With the aim of providing a possible solution to the boundary grayscale values problem, the authors of [10] and [4] suggested to perform a change of magnitude greater than 1. However, the effects of doing this were out of the scope of ± 1-steganography.

5. Conclusions

We have presented a new method for ± 1-steganography, based on $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes. These codes correspond, through the Gray map, to binary perfect codes, which can be nonlinear but they still have a parity check matrix representation which makes them computationally efficient to work with.

As shown in sections 3 and 4, this new scheme is optimal and performs better than the one obtained by simple direct sum of ternary Hamming codes from [10] and the one based on rainbow colouring of graphs using q-ary Hamming codes [4], for $q = 3$.

If we consider the special cases in which the technique might require to subtract one unit from a grayscale symbol containing the minimum allowed value, or to add one unit to a symbol containing the maximum allowed value, our method performs even better than those aforementioned schemes. This is so because unlike them, our method never applies any change of magnitude greater than 1, but two changes of magnitude 1 instead. This is better in terms of distortion and therefore makes the embedding less statistically detectable.

As for further research, since the approach based on product Hamming codes in [7] improved the performance of basic LSB steganography and the basic $F5$ algorithm, we would also expect a considerable improvement of the CI-rate by using product $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes or subspaces of product $\mathbb{Z}_2\mathbb{Z}_4$-additive perfect codes in ± 1-steganography.

Acknowledgments

The authors wish to thank the anonymous referees for useful and valuable comments which have improved some proofs in this paper.

References

[1] J. Bierbrauer and J. Fridrich, *Constructing good covering codes for applications in steganography*, in "Trans. on Data Hiding and Multimedia Security III," (2008), 1–22.

[2] J. Borges and J. Rifà, *A characterization of 1-perfect additive codes*, IEEE Trans. Inform. Theory, **45** (1999), 1688–1697.

[3] R. Crandall, *Some notes on steganography*, available from http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0203/steganografia/LINKS%20LOCALI/matrix-encoding.pdf, 1998.

[4] J. Fridrich and P. Lisoněk, *Grid colorings in steganography*, IEEE Trans. Inform. Theory, **53** (2007), 1547–1549.

[5] F. J. MacWilliams and N. J. A. Sloane, "The Theory of Error-Correcting Codes," North-Holland Publishing Company, 1977.

[6] P. Moulin and R. Koetter, *Data-hiding codes*, Proc. IEEE, **93** (2005), 2083–2126.
[7] H. Rifà-Pous and J. Rifà, *Product perfect codes and steganography*, Digit. Signal Process., **19** (2009), 764–769.
[8] B. Ryabko and D. Ryabko, *Asymptotically optimal perfect steganographic systems*, Probl. Inform. Transm., **45** (2009), 184–190.
[9] A. Westfeld, *High capacity despite better steganalysis (F5 - A steganographic algorithm)*, Lecture Notes in Comput. Sci., **2137** (2001), 289–302.
[10] F. M. J. Willems and M. van Dijk, *Capacity and codes for embedding information in grayscale signals*, IEEE Trans. Inform. Theory, **51** (2005), 1209–1214.