Improving success probability and embedding efficiency in code based steganography

Morgan Barbier* Carlos Munuera†‡

February 8, 2013

Abstract

For stegoschemes arising from error correcting codes, embedding depends on a decoding map for the corresponding code. As decoding maps are usually not complete, embedding can fail. We propose a method to ensure or increase the probability of embedding success for these stegoschemes. This method is based on puncturing codes. We show how the use of punctured codes may also increase the embedding efficiency of the obtained stegoschemes.

1 Introduction

Steganography is the art of transmitting information in secret, so that even the existence of communication is hidden. It is realized by embedding the messages to be protected into innocuous cover objects such as digital images. In order to minimize the possibility of being detected by third parties, the number of embedding changes in the cover must be small enough. Consequently, to obtain a high payload, steganographers should design stegosystems able to embed as much information as possible per cover change. In other words, we seek for embedding methods giving high embedding efficiency.

Given a cover image and a secret message, we first select the placement and intensity of allowed embedding changes in the cover. This is done by means of a *selection rule*. After this step the cover is transformed into a finite sequence $x_1, \dots, x_n$ of symbols from an alphabet $\mathcal{A}$ (usually $\mathcal{A} = \mathbb{F}_2$). We refer to this sequence as the *cover vector* $\mathbf{x} = (x_1, \dots, x_n)$. The secret message will be also a vector $\mathbf{m} = (m_1, \dots, m_r) \in \mathcal{A}^r$. The algorithms used for embedding the information $\mathbf{m}$ into $\mathbf{x}$ and later recovering $\mathbf{m}$ from $\mathrm{Emb}(\mathbf{x}, \mathbf{m})$)form the

*University of Caen - GREYC `morgan.barbier@unicaen.fr`

†University of Valladolid - Department of Applied Mathematics `cmunuera@arq.uva.es`

‡This work was supported by Junta de Castilla y León under grant VA065A07 and by Spanish Ministry for Science and Technology under grants MTM2007-66842-C02-01 and MTM 2007-64704.

stegoscheme associated to the stegosystem.

Crandall first noted that error correcting codes can be used to construct stegoschemes of high embedding efficiency. From this discovering many stegoschemes have been proposed using different types of codes. Relations between codes and stegoschemes will be treated in more detail in Section 2. In general, in code based steganography (also called *matrix embedding*), given a code $\mathcal{C}$ (or a family of such codes, as explained below), the embedding is realized by using a decoding map of $\mathcal{C}$. It turns out that every decoding map of $\mathcal{C}$ provides a realization of it as a stegoscheme. This nice idea encountered in practice a serious problem: the absence of effective complete decoding methods. Indeed, for nearly all currently known decoding algorithms, most errors are impossible to decode, which means that when using these algorithms, the majority of times the embedding process fails. In contrast, complete decoding algorithms, for which this problem does not occur, are computationally infeasible, so their practical applications are reduced to either perfect codes or codes of small length, which offer low embedding efficiency.

In this paper we propose a novel method to ensure embedding success, in principle for using any code. It is based on puncturing the original code as many times as necessary to get a new code whose covering radius equals to the correction capability of the original one. As we shall see, this method may also improve the embedding efficiency so that it guaranties embedding success and may give high embedding efficiency.

The organization of the paper is as follows. In Section 2 we recall the connection between coding theory and steganography, as well as we point the above mentioned drawback of stegoschemes obtained by this method, concerning decoding maps. Our method is exposed in detail in Section 3. Finally Section 4 deals to the parameters of the new stegoschemes. Some numerical experiments concerning BCH codes are reported.

2 From coding theory to steganography

2.1 Stegoschemes and codes

Let $\mathcal{A}$ be a finite alphabet with q elements and let $n \geq r$ be two positive integers. The purpose of a stegoscheme $\mathcal{S}$ is to embed a message $\mathbf{m} \in \mathcal{A}^r$ into a cover vector $\mathbf{x} \in \mathcal{A}^n$, making as few changes as possible in $\mathbf{x}$, and later extract the information hidden in the modified vector. Formally, a (n, r)-stegoscheme $\mathcal{S}$ is defined as a couple of functions $\mathcal{S} = (\mathrm{Emb}, \mathrm{Ext})$

$$\mathrm{Emb} : \mathcal{A}^n \times \mathcal{A}^r \longrightarrow \mathcal{A}^n \quad \text{and} \quad \mathrm{Ext} : \mathcal{A}^n \longrightarrow \mathcal{A}^r,$$

such that $\mathrm{Ext}(\mathrm{Emb}(\mathbf{x}, \mathbf{m})) = \mathbf{m}$, for all $(\mathbf{x}, \mathbf{m}) \in \mathcal{A}^n \times \mathcal{A}^r$. This condition guarantees that we always retrieve the right message $\mathbf{m}$ from the stego vector

$\mathrm{Emb}(\mathbf{x},\mathbf{m})$. The stegoscheme is called *proper* if $d(\mathbf{x},\mathrm{Emb}(\mathbf{x},\mathbf{m})) \leq d(\mathbf{x},\mathbf{v})$ for all $\mathbf{v}$ such that $\mathrm{Ext}(\mathbf{v}) = \mathbf{m}$, where d stands for the Hamming distance. This means that the number of embedding changes is the minimum possible allowed by the extracting map.

The first stegoscheme based on coding theory was proposed by Crandall in 1998, by using the family of binary Hamming codes, see [Cra98]. Let m be a positive integer and let H be a parity check matrix of the binary Hamming code of length $n = 2^m - 1$. The embedding and extracting functions are as follows

$$\begin{array}{rcl} \mathrm{Emb} : \mathbb{F}_2^n \times \mathbb{F}_2^m & \longrightarrow & \mathbb{F}_2^n \\ (\mathbf{x},\mathbf{m}) & \longmapsto & \mathbf{x} - \mathrm{cl}(\mathbf{x}H^T - \mathbf{m}), \end{array}$$

$$\begin{array}{rcl} \mathrm{Ext} : \mathbb{F}_2^n & \longrightarrow & \mathbb{F}_2^m \\ \mathbf{v} & \longmapsto & \mathbf{v}H^T, \end{array}$$

where $\mathrm{cl}(\mathbf{z})$ stands for a *coset leader* of $\mathbf{z} \in \mathbb{F}_2^m$. That is an element of minimum Hamming weight among those $\mathbf{v} \in \mathbb{F}_2^n$ verifying $\mathbf{v}H^t = \mathbf{z}$. The fact that $\mathrm{Ext}(\mathrm{Emb}(\mathbf{x},\mathbf{m})) = \mathbf{m}$ is straightforward.

This stegoscheme is so efficient that Westfeld developed the famous software F5 based on it [Wes01]. The notion of *efficiency*, which is the main subject of this paper, will be detailed in Section 2.2. After Crandall's discovery, other authors proposed the same model of stegoscheme, based on different types of codes: BCH [SW06, ZSK09], Reed-Solomon [FG09], trellis codes [FJF10], etc.

More generally, in [MB11] it is shown that a (n,r)-stegoscheme is equivalent to a family $\{(\mathcal{C}_{\mathbf{m}}, \mathrm{dec}_{\mathbf{m}})\}_{\mathbf{m}\in\mathcal{A}^r}$ where the $\mathcal{C}_{\mathbf{m}}$'s are nonempty disjoint codes in $\mathcal{A}^n$ (not necessarily linear), $\mathrm{dec}_{\mathbf{m}}$ is a decoding map for $\mathcal{C}_{\mathbf{m}}$ and $\cup\mathcal{C}_{\mathbf{m}} = \mathcal{A}^n$. Let us remember that a *decoding map* for a code $\mathcal{C} \subseteq \mathcal{A}^n$ is just a map $\mathrm{dec} : \mathcal{X} \subseteq \mathcal{A}^n \longrightarrow \mathcal{C}$. If $\mathcal{X} = \mathcal{A}^n$ then the decoding is said to be *complete* (as every received vector can be decoded). If dec verifies the additional property that $d(\mathbf{x},\mathrm{dec}(\mathbf{x})) = d(\mathbf{x},\mathcal{C})$ for all $\mathbf{x} \in \mathcal{X}$, then dec is called *minimum distance decoding*. Given such a family, the corresponding stegoscheme has embedding and extracting functions

$$\mathrm{Emb}(\mathbf{x},\mathbf{m}) = \mathrm{dec}_{\mathbf{m}}(\mathbf{x}) \quad \text{and} \quad \mathrm{Ext}(\mathbf{v}) = \mathbf{m} \text{ if } \mathbf{v} \in \mathcal{C}_{\mathbf{m}}.$$

If dec is a minimum distance decoding, then the obtained stegoscheme is proper. This method allows us to construct a large amount of stegoschemes, to which we collectively refer as *code-based* stegoschemes. If $\mathcal{A}$ is a field, $\mathcal{A} = \mathbb{F}_q$, and $\mathcal{C}_{\mathbf{0}}$ is $[n, n-r]$ linear, it is natural to consider the partition of $\mathbb{F}_q^n$ given by the translates of $\mathcal{C}_{\mathbf{0}}$ (or equivalently, the cosets of $\mathbb{F}_q^n/\mathcal{C}_{\mathbf{0}}$). Note that given a decoding map $\mathrm{dec}_{\mathbf{0}}$ of $\mathcal{C}_{\mathbf{0}}$, the function $\mathrm{dec}_{\mathbf{m}}(\mathbf{x}) = \mathrm{cl}(\mathbf{m}) + \mathrm{dec}_{\mathbf{0}}(\mathbf{x} - \mathrm{cl}(\mathbf{m}))$ is a decoding map for the translate $\mathcal{C}_{\mathbf{m}} = \mathrm{cl}(\mathbf{m}) + \mathcal{C}_{\mathbf{0}}$. By using the systematic writing of $\mathcal{C}_{\mathbf{0}}$, say in the first $n-r$ positions, we can avoid the need of computing coset leaders in the embedding process (unless they are required by $\mathrm{dec}_{\mathbf{0}}$), as $\mathrm{Emb}(\mathbf{x}, \mathbf{m}) = (\mathbf{0}, \mathbf{m}) + \mathrm{dec}_{\mathbf{0}}(\mathbf{x} - (\mathbf{0}, \mathbf{m}))$, see [MB11]. The same idea can be applied if $\mathcal{C}_{\mathbf{0}}$ is a group (nonlinear) code or a general systematic code.

Therefore, given a linear code $\mathcal{C}$, each decoding map of $\mathcal{C}$ provides a *realization* of $\mathcal{C}$ as stegoscheme. Recall that there is a universal decoding method for linear codes, the so-called *syndrome-leader decoding*, based on pre-computing all syndromes and leaders. For example, Crandall's stegoscheme is obtained from Hamming codes by using syndrome-leader decoding.

2.2 Embedding Efficiency

The behavior of a stegoscheme, and subsequently the comparison of two of them, is based on its parameters. Let $\mathcal{S}$ be a (n, r)-stegoscheme which embeds a message of $\mathcal{A}^r$ in a vector of $\mathcal{A}^n$ with T modifications at most and $\tilde{T}$ modifications in average. The *relative payload, change rate* and *average change rate* of $\mathcal{S}$ are respectively defined as

$$a = \frac{r}{n}\ , \ R = \frac{T}{n} \ \text{ and } \ \tilde{R} = \frac{\tilde{T}}{n}.$$

The *embedding efficiency* and *average embedding efficiency* of $\mathcal{S}$ are defined as

$$e = \frac{r}{T} \text{ and } \tilde{e} = \frac{r}{\tilde{T}}.$$

Recall that when $\mathcal{S}$ is arising from a linear or systematic code $\mathcal{C}$ (by the method above explained), then r is the redundancy of $\mathcal{C}$, and $T, \tilde{T}$ are the covering and average radii of $\mathcal{C}$, see for example [MB11].

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two stegoschemes defined over the same alphabet $\mathcal{A}$. If they have the same relative payload, then both embed as much information by using the same quantity of cover-medium. Thus $\mathcal{S}_1$ is better than $\mathcal{S}_2$ if and only if it produces less distortion in the cover, that is, if and only if the embedding efficiency of $\mathcal{S}_1$ is greater than the embedding efficiency of $\mathcal{S}_2$. So we look for stegoschemes with the biggest embedding efficiency for a fixed relative payload. There exists an upper on the embedding efficiency for a fixed relative payload as follows.

Theorem 1. *Let $\mathcal{S}$ be a q-ary stegoscheme with relative payload a and embedding efficiency e. Then*

$$e \leq \frac{a}{\mathcal{H}_q^{-1}(a)},$$

where $\mathcal{H}_q^{-1}$ is the inverse function of the q-ary entropy $\mathcal{H}_q(x) = x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x)$.

See [CMB$^+$08, Ch. 12, page 454]. The same result holds asymptotically for the average embedding efficiency.

2.3 Realizations of codes as stegoschemes

Let $\mathcal{C}$ be a $[n, k]$ linear code over $\mathcal{A} = \mathbb{F}_q$. As seen in the previous section, each decoding map dec of $\mathcal{C}$ provides a realization of $\mathcal{C}$ as a (n, r) stegoscheme, $\mathcal{S}_{\text{dec}}$. All of them are different although they share the same theoretical parameters: $r = n - k$ is the redundancy of $\mathcal{C}$, R is the covering radius of $\mathcal{C}$ and $\tilde{R}$ is the average radius of $\mathcal{C}$. This method of making stegoschemes has, in practice, a serious difficulty: the dramatic absence of computationally efficient complete decoding maps for a given code. As already mentioned above, there is a universal complete decoding method for linear codes, namely syndrome-leader decoding. For binary codes we can also use gradient decoding, based on minimal codewords. However, these two methods require the use (computation and storage) of tables, usually of a very large size, so they have mainly a theoretical interest, being their practical applications restricted to the case of perfect codes or codes of small length n. Note that stegoschemes of high efficiency have large n. Algebraic decoding methods (not based on using tables) may be computationally efficient, but they are far to be complete; usually they decode up to the packing radius $t = \lfloor (d-1)/2 \rfloor$, where d is the minimum distance, or even less. For example, Berlekamp-Massey algorithm for decoding for BCH codes, decodes errors of weight up to one half of the designed minimum distance. For BCH two-error correcting codes (which are quasi-perfect) a complete decoding method is available in [Har71] by using small tables and these codes have been proposed as good candidates for stegoschemes, see [ZSK09]. For other codes most vectors cannot be decoded. Consequently, most vectors of $\mathcal{A}^n$ cannot be used as a cover vectors for embedding. Remark that when using codes for error-correction, error patterns of low weight are more probable while for embedding purposes all vectors are, in principle, equally likely as covers.

Example 1. *Let $\mathcal{C} = \mathrm{BCH}_m(3)$ be the primitive binary triple error correcting BCH code of length $n = 2^m - 1$. Suppose first we want to decode $\mathcal{C}$ by using syndrome-leader decoding. To that end we need a table containing all syndromes and leaders. Since the total number of cosets is $2^{3m} = (n+1)^3$, the size of this table, in megabits, is the given in the following Table 1.*

m	size in Mb
5	1.507
6	21.234
7	310.378
8	4680.843
9	72209.138
10	1130650.141

Table 1: Size of a syndrome-leader table for the code $\mathrm{BCH}_m(3)$.

It is clear that we need other decoding methods even for small values of m. Let us consider the minimum distance decoding map dec *obtained by means of*

Berlekamp-Massey algorithm. It corrects up to 3 errors. Since the covering radius of this code is 5*, see [Hel78], error patterns of weight* ≤ 5 *can occur. Let* A_j *denote the number of cosets having leaders of weight* j*. Since every vector of weight* ≤ 3 *is a coset leader and there are* $2^{3m} = (n+1)^3$ *cosets, then the number of cosets whose vectors cannot be decoded by* dec *is*

$$A_4 + A_5 = (n+1)^3 - \sum_{j=0}^{3} \binom{n}{j} = \frac{n(n+1)(5n+13)}{6}.$$

Since the leading coefficient of the numerator is 5*, asymptotically* 5/6 *of error patterns cannot be decoded. Consequently, the stegoscheme based on* $\mathcal{C}_m$ *and* dec*, successfully embeds a message into a random cover vector with probability close to* 1/6*. Let us note, however, that it is known a computationally efficient decoding algorithm for* $\mathcal{C}$*, see [VDHB76]. In fact BCH(2) and BCH(3) are the only codes in the family of BCH codes for which such a complete decoding is available.*

Given a (n, r) stegoscheme $\mathcal{S} = (\mathrm{Emb}, \mathrm{Ext})$, the previous considerations allow us to define the *embedding probability* $p_{\mathcal{S}}$ of $\mathcal{S}$ as the probability that $\mathrm{Emb}(\mathbf{x}, \mathbf{m})$ can be computed over all possible pairs $(\mathbf{x}, \mathbf{m}) \in \mathcal{A}^n \times \mathcal{A}^r$ (considered as equally likely). We define also the *embedding efficiency of* $\mathcal{S}$ *relative to the embedding probability* $p_{\mathcal{S}}$ (resp. *average embedding efficiency of* $\mathcal{S}$ *relative to the embedding probability*) as

$$e_{rel} = e.p_{\mathcal{S}} \text{ and } \tilde{e}_{rel} = \tilde{e}.p_{\mathcal{S}}.$$

Example 2. *(Example 1 continued). Let* $\mathcal{S}$ *be the stegoscheme obtained from* $\mathcal{C} = \mathrm{BCH}_m(3)$*. The parameters of* $\mathcal{S}$ *can be obtained from the parameters of* $\mathcal{C}$*. In particular,* $n = 2^m - 1$*,* $r = 3m$ *(the redundancy of* $\mathcal{C}$*) and* $T = 5$ *(the covering radius of* $\mathcal{C}$*). In the same way, we can obtain the average change rate* $\tilde{T}$ *of* $\mathcal{S}$ *as the average radius of* $\mathcal{C}$*,*

$$\tilde{T} = 2^{-3m} \sum_{j=0}^{5} j A_j.$$

Since the values of A_4 *and* A_5 *are not known in general, we can not compute exactly this number. However, we have the following bounds [VDHB76]*

$$\frac{5n(5n+13)}{6} \leq A_4 \leq \frac{n(5n^2+10n-3)}{6},$$
$$\frac{4n(n+2)}{3} \leq A_5 \leq \frac{n(n-4)(5n+13)}{6}.$$

The embedding probability can be deduced from our computations in Example 1, as

$$p_{\mathcal{S}} = 2^{-3m} \sum_{j=0}^{3} \binom{n}{j} \approx \frac{1}{6}.$$

The following Table 2 collects the efficiency and relative efficiency of these stegoschemes for some small values of m*.*

m	n	r	$\tilde{T}$	e	$\tilde{e}$	$p_{\mathcal{S}}$	e_{rel}	$\tilde{e}_{rel}$
4	15	10	3.33	2	3.003	0.141	0.281	0.422
5	31	15	4.28	3	3.504	0.152	0.457	0.532
6	63	18	4.06	3.6	4.433	0.159	0.573	0.704
7	127	21	3.85	4.2	5.454	0.162	0.684	0.883
8	255	24	3.84	4.8	6.250	0.163	0.791	0.891
9	511	27	3.84	5.4	7.031	0.166	0.896	1.167
10	1023	30	3.84	6	7.812	0.166	0.996	1.296

Table 2: Embedding efficiency of stegoschemes based on $\mathrm{BCH}_m(3)$ codes.

3 A method to ensure embedding success

As we highlighted in the previous section, it is important to have a stegoscheme with an embedding probability as close to one as possible. In this section, we propose to modify the stegoscheme arising from a given code in order to ensure success (or at least to improve its probability).

3.1 Puncturing codes

Let $\mathcal{C}$ be a $[n, n-r, d]$ code of covering radius ρ. Let dec be a decoding map of $\mathcal{C}$, which can decode up to t errors. The main idea of our method is to imitate, in a sense, the behavior of perfect codes. To that end, we will puncture $\mathcal{C}$ as many times as necessary to obtain a code $\mathcal{C}'$ with covering radius $\rho' = t$. Then we can use dec to deduce a decoding map for $\mathcal{C}'$, as explained in the next paragraph. Let us remember that given a code $\mathcal{C}$, *puncturing* $\mathcal{C}$ at a position i is to delete the i-th coordinate in each codeword. If G is a generator matrix for $\mathcal{C}$, then a generator matrix for the punctured code is obtained from G by deleting column i (and dependent rows if necessary), see [HP03, Sect. 1.5.1].

3.2 Decoding punctured codes

Let $\mathcal{C}$ be a binary linear code and $\mathcal{C}'$ be the code obtained from $\mathcal{C}$ by puncturing at a set of positions $\mathcal{P} \subset \{1, \ldots, n\}$. Let $\overline{\mathcal{P}} = \{1, \ldots, n\} \setminus \mathcal{P}$, $\pi_{\overline{\mathcal{P}}}$ be the projection on the coordinates of $\overline{\mathcal{P}}$ and dec a decoding map for $\mathcal{C}$. The following algorithm provides a decoding map for $\mathcal{C}'$.

Proposition 1. *The algorithm 1 is correct and runs in $\mathcal{O}(q^{|\mathcal{P}|}c_{\text{dec}})$, where c_{dec} is the complexity of* dec *the decoding map of $\mathcal{C}$.*

Proof. Let $\mathbf{y}' \in \mathbb{F}_q^{n-|\mathcal{P}|}$ and $\mathbf{c}' \in \mathcal{C}'$ be a codeword such that $d(\mathbf{y}', \mathbf{c}') \leq t$. Then there exist $\mathbf{c} \in \mathcal{C}$ such that $\pi_{\overline{\mathcal{P}}}(\mathbf{c}) = \mathbf{c}'$. Let us to note that $\mathbf{y} \in \mathbb{F}_q^n$ such that $\pi_{\mathcal{P}}(\mathbf{y}) = \pi_{\mathcal{P}}(\mathbf{c})$ and $\pi_{\overline{\mathcal{P}}}(\mathbf{y}) = \mathbf{y}'$. Then $\mathbf{c}'$ is in the returned list. The statement of the complexity is obvious. $\square$

Algorithm 1: Decoding algorithm for the punctured code $\mathcal{C}'$.

Input : The punctured set $\mathcal{P}$, the received vector $\mathbf{y}' \in \mathbb{F}_q^{n-|\mathcal{P}|}$ and the decoding map dec of $\mathcal{C}$.

Output: The list of all codewords $\mathbf{c}'$ of $\mathcal{C}'$ such that $d(\mathbf{c}', \mathbf{y}') \leq t$.

$\mathcal{L} \leftarrow \emptyset$;

// for $q^{|\mathcal{P}|}$ elements

foreach $\boldsymbol{y} \in \mathbb{F}_q^n$ *such that* $\pi_{\overline{\mathcal{P}}}(\boldsymbol{y}) = \boldsymbol{y}'$ **do**

$\quad \mathcal{L} \leftarrow \mathcal{L} \cup \mathrm{dec}(\mathbf{y})$;

end

return $\{\pi_{\overline{\mathcal{P}}}(\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{L}\}$

If dec corrects up to t errors, the previous algorithm provides a decoding map for $\mathcal{C}'$ correcting t errors as well: as the returned list is nonempty, simply take one of the vectors closest to $\mathbf{y}'$.

Note that the statement about complexity in Proposition 1 has been computed considering the worst case. In following, we propose a little improvement for the average case.

Let $\mathbf{y}' \in \mathbb{F}_q^{n-|\mathcal{P}|}$, $\mathbf{y}' \notin \mathcal{C}'$, be the vector to be decoded and let $\mathbf{c}' \in \mathcal{C}'$ be the closest codeword of $\mathbf{y}'$. Then $d(\mathbf{c}', \mathbf{y}') = d(\mathcal{C}', \mathbf{y}')$. Let $\mathbf{e}' \in \mathbb{F}_q^{n-|\mathcal{P}|}$ be such that $\mathbf{y}' = \mathbf{c}' + \mathbf{e}'$, hence $\mathrm{wt}(\mathbf{e}') \leq \rho' = t$. There exists a "prefix" vector $\mathbf{p} \in \mathbb{F}_q^{|\mathcal{P}|}$ such that $\mathbf{c} = (\mathbf{p}, \mathbf{c}') \in \mathcal{C}$, with $\pi_{\mathcal{P}}(\mathbf{c}) = \mathbf{p}$ and $\pi_{\overline{\mathcal{P}}}(\mathbf{c}) = \mathbf{c}'$. Finally, let $\mathbf{y}, \mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{y} = (\mathbf{p}, \mathbf{y}') = \mathbf{c} + \mathbf{e}$. As stated before, Algorithm 1 works by considering all possible prefixes. However, since $\mathbf{e} = (\mathbf{0}, \mathbf{e}')$, and hence $\mathrm{wt}(\mathbf{e}) = \mathrm{wt}(\mathbf{e}')$, this algorithm can be slightly changed to avoid unnecessary iterations. In fact, the minimum number of iterations depends on $\mathrm{wt}(\mathbf{e}')$. It is simple to see that the number of iterations needed to decode is

$$\left\lceil \frac{q^{|\mathcal{P}|}}{V_q(|\mathcal{P}|, t - \mathrm{wt}(\mathbf{e}'))} \right\rceil .$$

Furthermore, all this iterations should be parallelized to speed up the algorithm. Of course, the value $\mathrm{wt}(\mathbf{e}')$ is not known a priori, but for each codeword $\mathbf{c}_i \in \mathcal{L}$, then distance $d(\mathbf{c}_i, \mathbf{y})$ gives an upper bound to the distance between $\mathbf{y}$ and the closest codeword $\mathbf{c}$.

Example 3. *(Continued from previous examples). Consider the code* $BCH_4(3)$ *of parameters* $[15, 5]$*. Here* $t = 3$ *and, as we can see in Table 5, we get* $|\mathcal{P}| = 3$.

Let $\mathcal{C}$ *be the binary* $BCH_4(3)$ *code,* $\mathcal{P} = \{1, 2, 3\}$ *be a punctured set and* $\mathcal{C}'$ *be the punctured code of* $\mathcal{C}$ *at the position given by* $\mathcal{P}$*. We propose to decode on* $\mathcal{C}'$*,* $\boldsymbol{y}' = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0)$ *the received word. We obtain* $\boldsymbol{c}_1 = (1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0)$ *by the decoding of* $\boldsymbol{y}_1$*. We deduce that* $\boldsymbol{c}_1' = \pi_{\bar{\mathcal{P}}}(\mathbf{c}_1) = (1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0)$ *is a codeword of* $\mathcal{C}'$*. Since* $d(\boldsymbol{c}_1', \boldsymbol{y}') = 2$,

(0, 0, 0)	(0, 0, 1)	(0, 1, 1)	(1, 1, 1)
	(0, 1, 0)	(1, 0, 1)	
	(1, 0, 0)	(1, 1, 0)	

Table 3: All binary vectors of length 3 sorted by weights.

then only

$$\left\lceil \frac{2^3}{V_q(3,1)} \right\rceil = 2$$

decoding calls is needed. Let $\boldsymbol{y}_2 = (1,1,1,1,1,1,1,1,1,1,1,1,0,0,0)$ *be the second, and last, virtual received word of* $\mathcal{C}$*. Note that the prefix used in this case,* $(1,1,1)$*, is the farthest from the previous one, that is* $(0,0,0)$*. The decoding on* $\mathcal{C}$ *gives us* $\boldsymbol{c}_2 = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)$*, which allows us to compute* $\boldsymbol{c}_2' = \pi_{\overline{\mathcal{P}}}(\boldsymbol{c}_2) = (1,1,1,1,1,1,1,1,1,1,1,1)$*. Since* $d(\boldsymbol{c}_2', \boldsymbol{y}') = 3 > d(\boldsymbol{c}_1', \boldsymbol{y}')$*, we deduce that the closest codeword is* $\mathbf{c}_1'$*.*
Another way to see our trick on this example, is after the computation of $\boldsymbol{c}_1$*, we observe that* $d(\boldsymbol{c}_1', \boldsymbol{y}') = 2$*, then the closest codeword from* $\boldsymbol{y}'$ *is at most at distance 2. Since the decoding algorithm on* $\mathcal{C}$ *can correct 3 errors, there is an extra error for the prefix set, see Table 3. Thus the decoding with prefix* $(0,0,0)$ *manage the prefix vectors of weight 0 and 1. Moreover, the decoding with prefix* $(1,1,1)$ *manage the prefix vectors of weight 2 and 3, so all prefix vectors are managed only by these 2 decodings.*
With this trick, we need only 2 decoding processes whereas our previous algorithm needs 8.

3.3 On the number of punctured positions

Since the decoding complexity of the punctured code $\mathcal{C}'$ is exponential on the number of punctured positions, we want to minimize this quantity. To that end we introduce some new notation. For a given integer $j \leq \rho$, let

$$\begin{aligned}
\mathcal{Y}_j &= \left\{\mathbf{y} \in \mathbb{F}_q^n : d(\mathbf{y}, \mathcal{C}) \geq \rho - j\right\} \\
\mathcal{E}_j &= \left\{\mathbf{y} - \mathbf{c} \in \mathbb{F}_q^n : \mathbf{y} \in \mathcal{Y}_j, \mathbf{c} \in C \text{ and } d(\mathbf{y}, \mathbf{c}) = d(\mathbf{y}, \mathcal{C})\right\} \\
\mathcal{P}_j &= \bigcap_{\mathbf{e} \in \mathcal{E}_j} \operatorname{supp}(\mathbf{e}).
\end{aligned}$$

Proposition 2. *Let* j *be a positive integer and let* $\mathcal{C}_j'$ *be the code obtained by puncturing* $\mathcal{C}$ *at the positions of* $\mathcal{P}_j$*. The covering radius of* $\mathcal{C}_j'$ *satisfies*

$$\rho_j' \leq \max\left\{\rho - j - 1, \rho - |\mathcal{P}_j|\right\}.$$

Proof. Assume there exists $\mathbf{y}' \in \mathbb{F}_q^{n-|\mathcal{P}_j|}$ such that $d(\mathbf{y}', \mathcal{C}') > \max\left\{\rho - j - 1, \rho - |\mathcal{P}_j|\right\}$. Let $(\mathbf{c}', \mathbf{e}') \in \mathcal{C}' \times \mathbb{F}_q^{n-|\mathcal{P}_j|}$ such that $\mathbf{y}' = \mathbf{c}' + \mathbf{e}'$ and $\operatorname{wt}(\mathbf{e}') = d(\mathbf{y}', \mathbf{c}') = d(\mathbf{y}', \mathcal{C}') > \max\left\{\rho - j - 1, \rho - |\mathcal{P}_j|\right\}$. Then there exist

$\mathbf{y} \in \mathcal{Y}_j$ and $(\mathbf{c}, \mathbf{e}) \in \mathcal{C} \times \mathcal{E}_j$ such that $\pi_{\overline{\mathcal{P}}_j}(\mathbf{y}) = \mathbf{y}'$ and

$$\begin{cases} \mathbf{y} & = & \mathbf{c} + \mathbf{e}, \\ \pi_{\overline{\mathcal{P}}_j}(\mathbf{e}) & = & \mathbf{e}'. \end{cases}$$

Since $\mathbf{e} \in \mathcal{E}_j$, we have $\mathcal{P}_j \subset \mathrm{supp}(\mathbf{e})$ and so $\mathrm{wt}(\mathbf{e}) = \mathrm{wt}(\mathbf{e}') + |\mathcal{P}_j| > \rho$, which is impossible by the definition of covering radius. □

Therefore, greater j implies smaller $\rho - j - 1$ but also greater $\rho - |\mathcal{P}_j|$. For steganographic purposes we must puncture to obtain a code $\mathcal{C}'$ with a covering radius equal to the correction capacity of the decoding map of $\mathcal{C}$. Since puncturing $n-1$ times leads to a code of covering radius 0, this is always possible.

Algorithm 2: Algorithm to compute the punctured code $\mathcal{C}'$.

Input : A code $\mathcal{C}$, its covering radius ρ and a positive integer t.
Output: A couple $(\mathcal{C}', \mathcal{P})$, where $\mathcal{C}'$ is obtained from $\mathcal{C}$ by puncturing at the positions of $\mathcal{P}$ and has covering radius t.

$(\mathcal{C}', \rho') \leftarrow (\mathcal{C}, \rho)$;
$\mathcal{P} \leftarrow \emptyset$;
while $t \neq \rho'$ **do**
    $CL \leftarrow$ leaders of cosets of $\mathcal{C}'$ with weight greater than t;
    $i_{\max} \leftarrow$ the position which occurs the most of time in the support of CL;
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{i_{\max}\}$;
    $\mathcal{C}' \leftarrow \mathcal{C}'$ punctured at the position $i_{\max}$;
    $\rho' \leftarrow$ covering radius of $\mathcal{C}'$;
end
return $(\mathcal{C}', \mathcal{P})$

The main drawback of this algorithm is that it could be very expensive in time and memory complexities, since it requires the computation of the whole set of the coset leaders. Note however, that this computation must be done just once for code (as for any other parameter of $\mathcal{C}$).

3.4 A stegoscheme based on a punctured code

Let $\mathcal{C}$ be a code of covering radius ρ and dec a decoding map correcting t errors. Assume we know the set $\mathcal{P}$ of positions to obtain $\mathcal{C}'$, the punctured code of covering radius t, which can be computed by using Algorithm 2. Moreover let dec' be a decoding map for $\mathcal{C}'$ obtained from Algorithm 1. Finally, let H' be a parity check matrix of $\mathcal{C}'$. It can be easily obtained from a generator matrix G' which, in turn, can be easily obtained from a generator matrix G of $\mathcal{C}$. We have all the ingredients to define a stegoscheme $\mathcal{S}'$ based on the matrix embedding principle with the punctured code $\mathcal{C}'$. Let n' and r' be, respectively, the length and redundancy of $\mathcal{C}'$. The embedding map of $\mathcal{S}'$ is

$$\begin{array}{rcccl} \mathrm{Emb}: \mathbb{F}_2^{r'} & \times & \mathbb{F}_q^{n'} & \longrightarrow & \mathbb{F}_q^{n'} \\ (\mathbf{m} & , & \mathbf{v}) & \longmapsto & \mathbf{y} + \mathrm{dec}'(\mathbf{v} - \mathbf{y}), \end{array}$$

where $\mathbf{y}$ is an element of $\mathbb{F}_q^{n'}$ such that $\mathbf{y}H'^T = \mathbf{m}$. Recall that when $\mathcal{C}'$ is in systematic form, then one can simply take $\mathbf{y} = (\mathbf{0}, \mathbf{m})$. The extracting map is

$$\begin{array}{rcl} \mathrm{Ext}: \mathbb{F}_q^{n'} & \longrightarrow & \mathbb{F}_2^{r'} \\ \mathbf{y} & \longmapsto & \mathbf{y}H'^T. \end{array}$$

It is simple to check that $\mathrm{Ext}\,(\mathrm{Emb}\,(\mathbf{m}, \mathbf{v})) = \mathbf{m}$ so that couple $(\mathrm{Emb}, \mathrm{Ext})$ defines a true stegoscheme $\mathcal{S}'$.

3.5 Tradeoff

In previous sections we have suggested the use of puncturing up to ensuring embedding success. However, since complete decoding is known to be an NP-hard problem [BMVT78], in specific situations this goal may be too ambitious.

Indeed, we can simply use the puncturation principle to increase the embedding probability, but not necessarily up to one. For example, the sender could have a target embedding probability $p_{\mathcal{S}} < 1$, which is not reachable by the original stegoscheme. Then we can stop the puncturation process when the target embedding probability is reached. Analogously, since the complexity of the decoding algorithm for punctured codes is exponential in the number of the punctured positions, the steganographer may also limit this number to a preset maximum, according to available computing resources. In short, the proposed method is totally versatile to be modified according to the requirements of the sender. In the rest of this article, we puncture up to have an embedding probability $p_{\mathcal{S}} = 1$.

4 Parameters of the new stegoschemes

Keeping notations as in the previous section, let $\mathcal{C}$ be a code for which there exists an efficient algorithm capable of correcting t errors. Let $\mathcal{C}'$ be its punctured of covering radius t, and let $\mathcal{S}, \mathcal{S}'$ be the stegoschemes obtained from $\mathcal{C}$ and $\mathcal{C}'$ respectively. The parameters of $\mathcal{S}'$ are

$$a' = \frac{r'}{n'},\ R' = \frac{t}{n'},\ e' = e'_{rel} = \frac{r'}{t},\ \tilde{R}' = \frac{\tilde{T}'}{n'},\ \tilde{e}' = \tilde{e}'_{rel} = \frac{r'}{\tilde{T}'},$$

where $\tilde{T}'$ is the average covering radius of $\mathcal{C}'$. It is not easy to obtain general closed formulas for these parameters, since they depend on the number and location of punctured positions, which in turn depend on the code $\mathcal{C}$ and the number t. However, except for perfect codes (for which t is just the covering

m	BCH$_m(2)$						punctured BCH$_m(2)$					
	n	r	a	$\tilde{R}$	e	$\tilde{e}$	n'	r'	a'	$\tilde{R}'$	e'	$\tilde{e}'$
4	15	8	0.533	0.164	2.67	3.25	11	4	0.363	1.375	2	2.909
5	31	10	0.323	0.0801	3.33	4.03	28	7	0.250	1.766	3.5	3.965
6	63	12	0.190	0.0396	4	4.81	59	8	0.135	1.777	4	4.501
7	127	14	0.110	0.0197	4.66	5.61	123	10	0.081	1.879	5	5.322
8	255	16	0.0627	0.0098	5.34	6.41	251	12	0.047	1.939	6	6.189
9	511	18	0.0352	0.00489	6	7.2	507	14	0.027	1.969	7	7.110
10	1023	20	0.196	0.00244	6.66	8	1018	15	0.014	1.969	7.5	7.617
11	2047	22	0.0107	0.00122	7.34	8.80	2042	17	0.008	1.984	8.5	8.566

Table 4: Parameters of stegoschemes arising from binary BCH$_m(2)$ codes and punctured BCH$_m(2)$ codes.

m	BCH$_m(3)$						punctured BCH$_m(3)$					
	n	r	a	$\tilde{R}$	e	$\tilde{e}$	n'	r'	a'	$\tilde{R}'$	e'	$\tilde{e}'$
4	15	10	0.667	0.222	2	3	12	7	0.583	0.191	2.33	3.05
5	31	15	0.484	0.138	3	3.5	25	9	0.36	0.0985	3	3.65
6	63	18	0.286	0.0645	3.6	4.43	56	11	0.197	0.0434	3.67	4.52
7	127	21	0.165	0.0303	4.2	5.45	121	15	0.124	0.0230	5	5.38
8	255	24	0.0941	0.0151	4.8	6.25	248	17	0.0686	0.0112	5.66	6.11
9	511	27	0.0529	0.00751	5.4	7.03	504	20	0.0397	0.00572	6.66	6.94

Table 5: Parameters of stegoschemes arising from binary BCH$_m(3)$ codes and punctured BCH$_m(3)$ codes.

radius of $\mathcal{C}$ and we do not need to puncture) we have $p_{\mathcal{S}'} = 1$, $n' < n$, $a \geq a'$ and $T' = t < T$, $\tilde{T}' < \tilde{T}$. The embedding efficiency of $\mathcal{S}'$ may be larger or smaller than the embedding efficiency of $\mathcal{S}$, for a relative payload fixed.

4.1 Numerical experiments

In order to see some concrete results showing the performance of our method, we list some numerical results obtained for primitive two and three error correcting binary BCH codes, taking the number t given by the BCH bound (that is, allowing efficient decoding by means of Berlekamp-Massey algorithm, as usual). These codes have been studied by many authors[VDHB76, Hel78] and proposed as good candidates for constructing stegoschemes. It is well known that they have covering radii 3 and 5 respectively, and that BCH$_m(2)$ is quasi-perfect. By using Algorithm 2, we have determined the couple $(\mathcal{C}', \mathcal{P})$ for some two and three error-correcting binary BCH codes and subsequently computed the parameters of the corresponding stegoschemes.

The obtained results are listed in Tables 4 (two-error-correcting, $t = 2$) and 5 (three-error-correcting, $t = 3$). Notations in these tables are same used throughout this article. In both cases we puncture BCH codes to get other codes whose covering radii are equal to the correction capabilities of BCH codes. Then $R = 3, R' = 2$ for two-error-correcting and $R = 5, R' = 3$ for three-error-correcting BCH codes. We do not include the probability of embedding success for punctured codes, since in all cases such probability is exactly 1. For a better understanding of these results, and a comparison with stegoschemes coming

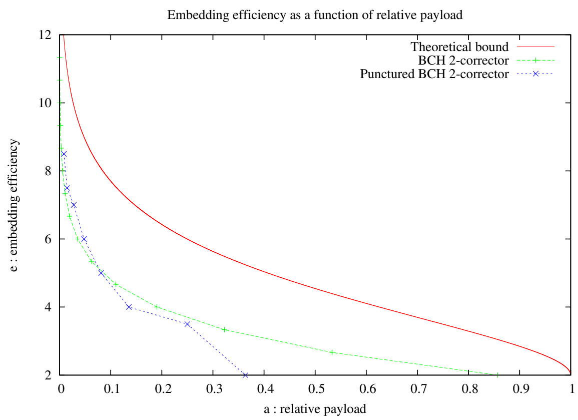

Figure 1: Comparison between stegoschemes based on the binary $\mathrm{BCH}_m(2)$ and their punctured associated codes.

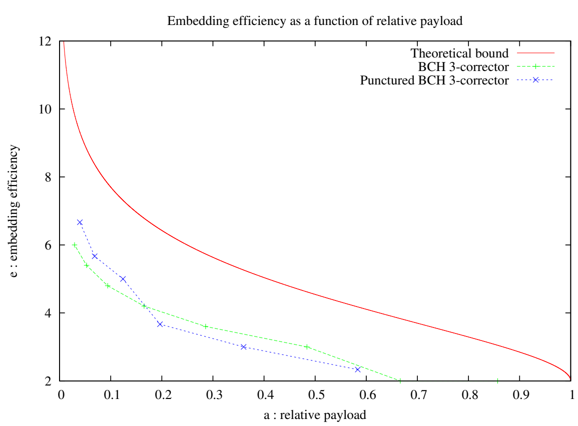

Figure 2: Comparison between stegoschemes based on the binary $\mathrm{BCH}_m(3)$ and their punctured associated codes.

from original BCH codes, we also include a graphical representation in Figures 1 and 2.

As we see, the new stegoschemes may have a better performance in terms of embedding efficiency. For example, in Table 4, we can see how the stegosystem based on $\text{BCH}_9(2)$ has the same embedding efficiency as the stegoscheme based on $\text{BCH}_8(2)'$, but with a smaller relative payload. Thus we conclude that the stegoschemes based on punctured codes may be better than the original ones *in the worst case*, although they do not appear to be better in terms of *average* embedding efficiency.

All computations have been performed by using the system MAGMA [BCP97], where BCH codes are managed in systematic form. We remark that all punctured positions have resulted to be among the first $n - r$ (systematic) positions. Given our limited computer resources, we have not used Algorithm 2 in full to obtain the puncturing of $\text{BCH}_8(3)$ and $\text{BCH}_9(3)$. Instead we have punctured these codes just at the first positions, up to obtain a code with covering radius R' equal to $t = 3$ (and then true results may be somewhat better than those shown in Table 5).

5 Conclusion

We have proposed a method to ensure or increase the probability of embedding success for stegoschemes arising from error correcting codes. This method is based on puncturing codes. As we have seen, the use of these punctured codes can also increase the embedding efficiency of the obtained stegoschemes.

References

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BMVT78] Elwyn Berlekamp, Robert McEliece, and Henk Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. on Information Theory*, IT-24(3):384–386, May 1978.

[CMB+08] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008.

[Cra98] Ron Crandall. Some notes on Steganography, 1998. Posted on the steganography mailing list.

[FG09] Caroline Fontaine and Fabien Galand. How Reed-Solomon codes can improve steganographic schemes. *EURASIP J. Inf. Secur.*, 2009:1–10, 2009.

[FJF10] Tomas Filler, Jan Judas, and Jessica Fridrich. Minimizing embedding impact in steganography using trellis-coded quantization. *Media Forensics and Security II*, 7541(1):754105, 2010.

[Har71] Carlos Hartmann. A note on the decoding of double-error-correcting binary BCH codes of primitive length. *IEEE Trans. on Information Theory*, 17(6):765 – 766, November 1971.

[Hel78] Tor Helleseth. All binary 3-error-correcting BCH codes of length $2^m - 1$ have covering radius 5. *IEEE Trans. on Information Theory*, IT-24:257–258, March 1978.

[HP03] Cary Huffman and Vera Pless. *Fundamentals of error-correcting codes*. Cambridge University Press, Cambridge, 2003.

[MB11] Carlos Munuera and Morgan Barbier. Wet paper codes and the dual distance in steganography. *Advances in Mathematics of Communications*, 2011. To be published.

[SW06] Dagmar Schönfeld and Antje Winkler. Embedding with syndrome coding based on BCH codes. In *Proceedings of the 8th workshop on Multimedia and security*, MM&Sec '06, pages 214–223, New York, NY, USA, 2006. ACM.

[VDHB76] Jose Antonio Van Der Horst and Tony Berger. Complete decoding of triple-error-correcting binary BCH codes. *IEEE Trans. on Information Theory*, pages 138–147, 1976.

[Wes01] Andreas Westfeld. F5 - A steganographic algorithm. In Ira Moskowitz, editor, *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 289–302. Springer Berlin / Heidelberg, 2001.

[ZSK09] Rongyue Zhang, Vasiliy Sachnev, and Hyoung Kim. Fast BCH syndrome coding for steganography. In Stefan Katzenbeisser and Ahmad-Reza Sadeghi, editors, *Information Hiding*, volume 5806 of *Lecture Notes in Computer Science*, pages 48–58. Springer Berlin / Heidelberg, 2009.