Maximum likelihood decoding algorithm for some Goppa and BCH Codes: Application to the matrix encoding method for steganography

Thierry P. Berger
XLIM (UMR CNRS 7252), Université de Limoges
Limoges, France
Email: thierry.berger@unilim.fr

Mohamed Bouye Ould Medeni
LMIA, Université Mohammed V- Agdal
Rabat, Morocco
Email: sbaimedeni@yahoo.fr

***Abstract*—The idea of "Matrix encoding" was introduced in steganography by Crandall in 1998. The implementation was then proposed by Westfeld with steganography algorithm F5. Matrix encoding using linear codes (syndrome coding) is a general approach to improving embedding efficiency of steganographic schemes. The covering radius of the code corresponds to the maximal number of embedding changes needed to embed any message. Steganographers, however, are more interested in the average number of embedding changes rather than the worst case. In fact, the concept of embedding efficiency - the average number of bits embedded per embedding change - has been frequently used in steganography to compare and evaluate performance of steganographic schemes. The aim of this paper is to transform some algebraic decoding algorithms up to the error correcting capacity into a maximum likelihood decoder by the use of a limited exhaustive search. This algorithm is directly inspired from those proposed by N. Courtois, M. Finiasz, and N. Sendrier in the context of electronic signature. It remains exponential, however it becomes practicable for for some small BCH and Goppa codes (typically, with an error correcting capacity until 4).**

Keywords - Steganography; Error-correcting Codes; Complete Decoding; Goppa Codes; Embedding Efficient.

I. INTRODUCTION

Research on hiding data into digital multimedia objects, such as images, audios, and videos, has advanced considerably over the past decade. Steganography refers to the science of covert communication, and steganalysis is the opposite of steganography. Nowadays, a large number of steganography tools have been developed based on replacement of the least significant bit (LSB) with secret message because of its extreme simplicity.

An interesting steganographic method is known as matrix encoding, introduced by Crandall [4]. 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 [17], who incorporated a specific implementation using Hamming codes in his F5 algorithm. This steganographic scheme can embed m bits of message in 2^m-1 cover symbols by changing at most one of them.

There are three parameters to evaluate the performance of a steganographic method over a cover vector of n symbols. The first one is average distortion $D = \frac{r_a}{n}$, where r_a is the expected number of changes over uniformly distributed messages. The second one is the embedding rate $\epsilon_r = \frac{k}{n}$, which is the amount of bits that can be hidden in a cover vector [2] (k is the number of bits of the hidden message). The third one is the embedding efficiency $\epsilon_{\mathit{eff}} = \frac{k}{r_a}$, which is the average number of hidden bits per changed bit. So, we have the relation $D\epsilon_{\mathit{eff}} = \epsilon_r$. In general, for the same embedding rate a method is better when the average distortion is smaller. As usually, we denote by (n, k, r_a) the parameters of a steganographic protocols. the reader must be careful not to confuse with the parameters $[n, k, d]$ of a code, in particular the number of bits of a steganographic scheme is generally the co-dimension $n - k$ of a code of dimension k.

The matrix encoding technique is a well-studied method to insert a hidden message into a cover message, for example into an image *cf* [6], [14], [15], [18]. It is assumed that a strategy of insertion has been previously defined, therefore in this paper we will not discuss the security of any stegosystem, which is directly dependent on the chosen strategy. The main objective of the matrix encoding is to minimize the number of modified bits during the insertion of a given message. One of the limitations of this method is the fact that a maximum likelihood decoding algorithm is required. Unfortunately, maximum-likelihood decoding of general linear codes is NP-hard [1]. Some family of codes, such as BCH codes or Goppa codes, have a decoding algorithm up to a correction capacity t. The purpose of this paper is to transform these algorithms into maximum likelihood decoding algorithms. This decoding algorithm is a kind of exhaustive search aided by an algebraic decoding algorithm. It is derived from that presented in [3], which is used to provide a short signature based on the McEliece Public key Cryptosystem. We present specific applications to some binary Goppa codes and BCH codes and show that these codes are new candidates for practical implementation of the matrix encoding technique.

This paper is organized as follows. In Section 2, we review the basic application of coding theory in steganography. In

 ISBN: 978-1-61208-204-2

Section 3, we recall the complete decoding technique. Section 4 presents the experimental results on some classical binary Goppa and BCH codes.

II. Error-correcting Codes in Steganography

An important kind of steganographic protocols can be defined from coding theory. Error-correcting codes are commonly used for detecting and correcting errors, or erasures, in data transmission. An explicit description of the relations between error-correcting codes and steganographic systems was presented in [14], [15], [18]. The most commonly used codes in steganography are linear. The existence of a parity check matrix helps on designing good steganographic protocols. Crandall [4] introduced the matrix encoding idea to improve the embedding efficiency for steganography. F5 proposed by Westfeld [17] is the first implementation of the matrix encoding concept to reduce modification of the quantized DCT coefficients. Basically, the matrix encoding technique in F5 modifies at most 1 coefficient among n coefficients to hide k bits. For example, if we use the $[7, 4]$ Hamming code, we obtain a $(7, 3)$ steganographic *i.e.*, one can insert 3 bits into a cover of length 7 by changing one bit of the cover. Modified matrix encoding (MME) [11] uses a $(n, k, 2)$ code where one more coefficient may be changed in each group compared with the matrix encoding. Main concept of the matrix encoding technique is "the less number of modification to the DCT coefficients, the less amount of distortion in the image".

Later, several efficient codes have been proposed to realize the matrix encoding: BCH error-correcting code [19], [16], Reed-Solomon (RS) [5], product perfect codes [15]. Error-correcting codes and steganographic systems were presented by Zhang [14], Munuera, Galand [18], [10]. It is shown in [14] that there is a corresponding relation between the maximum length embeddable (MLE) codes and perfect error correcting codes.

Let n and k be positive integers, $k \leq n$, and let B be a finite set. An embedding/retrieval steganographic protocol of type (n, k) over B is a pair of maps $e : B^k \times B^n \to B^n$ and $r : B^n \to B^k$ such that $r(e(s, v)) = s$ for all $s \in B^k$ and $v \in B^n$. Maps e and r are respectively the embedding and the retrieval map. The number $\rho = max\{d(v, e(s, v)); s \in B^k, v \in B^n\}$, d being the Hamming distance, is the radius of the protocol. The embedding map of a (n, k) embedding/retrieval steganographic protocol [8], [7], [20] with radius ρ allows us to hide k information symbols into a string of n cover symbols, by changing at most ρ symbols of the cover.

A linear code of length n over the finite field $GF(q)$ is a subspace C of the $GF(q)$-linear space $GF(q)^n$. The Hamming distance $d(v, w)$ between two vectors v and w of $GF(q)^n$ is the number of distinct coefficients between v and w. The support of a vector $v = (v_1, v_2, ..., v_n) \in GF(q)^n$ is the set $Supp(v) = \{i | v_i \neq 0\}$. So, $d(v, w)$ is also the number of elements of $Supp(v - w)$. The minimum distance d of a code C is the minimum distance between any pair of codewords (*i.e.* elements of C). The covering radius ρ of the code C is defined as $\rho = max_{v \in GF(q)^n}\{d(v, C)\}$, where $d(v, C)$ means the minimum Hamming distance from vector v to the code C. The parameters $[n, k', d]$ (or $[n, k']$ as d is not known) are respectively the length, the dimension and the minimum distance of the code. In the sequel, for steganographic application, we are interested in the co-dimension $k = n - k'$ of the code, which is the size of the hidden message.

Let $B = GF(q)$. A a parity check matrix H of C is a $(n - k') \times n$ full rank matrix such that $v \in C$ if and only if $H \times v^t = 0$, where v^t means the vector v as a column vector. The syndrome of any $v \in B^n$ is the vector $r(v) = H \times v^t$. A coset $C + v$ is the set of all vectors in B^n with the same syndrome. A vector $l_{r(v)}$ of minimum weight in $C+v$ is called a coset leader. Note that this coset leader is not necessarily unique.

The matrix encoding steganographic protocol is defined as follows. The syndrome map $r : B^n \to B^k$ defined by $r(v) = H \times v^t$ is the retrieval map of the (n, k, r_a) steganographic protocol, which will be called linear to emphasize that the retrieval map r is a linear map. The embedding algorithm $e(s, v)$ requires the classical coset leader decoding algorithm, which return the coset leader of $v + C$. The embedding algorithm is described in Algorithm 1.

Algorithm 1: Coset steganographic algorithm.

Required : a coset decoding algorithm: input a syndrome u, output: a coset leader l_u
Input : a cover v of size n and a message s of size k.
Output : $v' = e(s, v)$, a steganographic cover of s with distortion $d(v, v')$ as small as possible.

1: **Compute** $u := r(v) - s$,
2: **set** $c := v - l_u$,
3: **return** $e(s, v) := c$.

The maximum weight of a coset leader is the covering radius ρ of the code, so the embedding efficiency is upper bounded by ρ: $r_a \leq \rho$, with equality if and only if the code is perfect.

III. Complete Decoding Algorithm

For practical implementation of the matrix embedding technique, the crucial point is the fact that it requires a complete decoding algorithm. In this section, we will present a more efficient decoding algorithm than those used previously, under the restriction that the chosen code must posses a non-complete) algebraic decoding algorithm. A complete decoding algorithm takes in input any word of the space and return a nearest codeword in C. It performs a maximum likelihood decoding. This problem is equivalent to be able to find an error pattern of minimal weight corresponding to any given syndrome. This problem is known to be NP-hard [1], [3]. Clearly, such an algorithm will be able to correct errors of weight greater than the error-correcting capacity t of the code. The weight of correctable errors is upper-bounded by the covering radius ρ. Unfortunately, for steganographic applications,

 ISBN: 978-1-61208-204-2

the determination of the covering radius value of a code is also a hard problem. More precisely, the determination of the covering radius of a linear code was proved Π2-hard by McLoughlin [12]. In practice, the determination of the covering radius needs the enumeration of the coset leaders (minimum weight words) of any coset of the code. Roughly speaking, it requires $\binom{n}{\rho}$ operations.

A complete decoding algorithm can be performed by an exhaustive search on codewords. It can also be performed by an exhaustive search on errors of increasing weight.

In the sequel, following the idea developed in [3] in the context of digital signature, we propose to extend any classical algebraic decoding algorithm up to the error correcting capacity t into a complete decoding algorithm. If the error is of weight $w = t + i$, this algorithm performs an exhaustive search on the first i bits, the remaining t bits are corrected by the algebraic decoder.

The principle is as follows: First, we try to decode the received word x with the algebraic algorithm. If this attempt succeeds, we return the corrected codeword. If not, we enumerate all the possible errors following their increasing weight, we add this error to the received word and try to decode it again. If the distance between x and the code C is w, this algorithm succeeds with an additional error e of weight $w - t$, so the algorithm is upper-bounded by a maximal weight of additional error $\rho - t$. Clearly, this modified decoding algorithm remains exponential in the weight of the errors, however, in practice, it is efficient to decode more than t errors (typically, until $t + 4$ for practical applications).

Algorithm 2: Complete decoding [3].

Required : a decoding algorithm dec of error capacity t. For an entry v it returns a boolean value $\mathrm{dec}\,1(v)$ and a vector $\mathrm{dec}\,2(v)$: "true" and $c \in C$ with $d(c, v) \leq t$ if it succeed, "false" and v if not.
Input : a cover v of size n and a message s of size k.
Output : $v' = e(s, v)$, a steganographic cover of s with distortion $d(v, v')$ as small as possible.

if $\mathrm{dec}_1(v)$ = true **then**
  return $\mathrm{dec}_2(v)$
end if
$i := 1$
$x := v$
while $\mathrm{dec}_1(x)$ = false **do**
  Enumerate all the errors vectors e of weight $w(e) = i$
  $x := v + e$
  if $\mathrm{dec}_1(x)$ = true **then**
    return $\mathrm{dec}_2(x)$
  end if
  $i := i + 1$
end while

It is possible to derive a non-complete polynomial decoding algorithm up to a fixed error-correction capacity $c < \rho$ by limiting the exhaustive search on the i-th first errors to whose of weight less than or equal to $\delta = c - t$.

Combining Algorithm 2 with Algorithm 1, we can derive an efficient steganographic protocol as described in Algorithm 3.

Algorithm 3: Steganographic scheme.

Required : a decoding algorithm dec of error capacity t. For an entry v it returns a boolean value $\mathrm{dec}\,1(v)$ and a vector $\mathrm{dec}\,2(v)$: "true" and $c \in C$ with $d(c, v) \leq t$ if it succeed, "false" and v if not.
Input : a cover v of size n and a message s of size k.
Output : $v' = e(s, v)$, a steganographic cover of s with distortion $d(v, v')$ as small as possible.

1: **Compute** $u := r(v) - s$,
2: **Compute** x such that $r(x) = u$
3: **Decode** x with Algorithm 2. Set $c \in C$ the output of the decoding algorithm.
4: **Set** $e = x - c$ the error vector
5: **return** $e(s, v) = v + e$

IV. Application to binary BCH codes and Goppa Codes

As a concrete example of application of our method, we tested it on binary BCH codes and binary Goppa codes, with a prescribed minimum distance of 7 or 9, *i.e.*, with a decoding algorithm of error correcting capability 3 or 4. The decoding algorithm is completed with an exhaustive search until 4 additional errors. We choose these two classes of codes because they have an algebraic decoding algorithm up to the error correcting capacity, and parameters suitable for practical applications.

From a theoretical point of view on the parameters of the corresponding stegosystem, we are able to determine the true covering radius only for codes with small length and small covering radius. The following tables present the results obtained from BCH codes and Goppa codes with constructed error-correcting capability $t = 3$ and $t = 4$. We compare these values with whose obtained from known constructions.

Table I compares the theoretical parameters of steganographic protocols based on Hamming codes (F5 [17]), 2 errors correcting BCH codes [19], [16], and 3 or 4 errors correcting BCH and Goppa codes. The third value is not the embedding efficiency in average as usual, but the upper-bound given by covering radius. This value was computed using Magma Computer Algebra system [13]. For large codes, we were not able to achieve this computation. An estimation of the true embedding efficiency will be given in the next tables. It is not easy to directly compare results with distinct values of n and k. The comparison will be clearer in Figure 1. The main interest of our method is to reach new parameter values for steganographic protocols.

Tables II and III present the experimental results of simulations on BCH and Goppa codes of minimum distance 7 and

BCH $t = 2$ [16], [19]	Hamming $t = 1$ [17]
$(15, 8, 3)$	$(15, 4, 1)$
$(31, 10, 3)$	$(31, 5, 1)$
$(63, 12, 3)$	$(63, 6, 1)$
$(127, 14, 3)$	$(127, 7, 1)$
$(255, 16, 3)$	$(255, 8, 1)$
$(511, 18, 3)$	$(511, 9, 1)$
$(1023, 20, 3)$	$(1023, 10, 1)$

BCH $t = 3$	Goppa $t = 3$	BCH $t = 4$	Goppa $t = 4$
$(15, 10, 5)$	$(15, 10, 6)$	$(15, 14, 7)$	
$(31, 15, 6)$	$(31, 15, 6)$	$(31, 20, 7)$	$(30, 19, 8)$
$(63, 18, 5)$	$(63, 18, 6)$	$(63, 24, 7)$	$(63, 24, 8)$
$(127, 21, 5)$	$(127, 21, 6)$	$(127, 28, ?)$	$(127, 28, ?)$
$(255, 24, ?)$	$(255, 24, ?)$	$(255, 32, ?)$	$(255, 32, ?)$
$(511, 27, ?)$	$(511, 27, ?)$	$(511, 36, ?)$	$(511, 36, ?)$
$(1023, 30, ?)$	$(1023, 30, ?)$	$(1023, 40, ?)$	$(1023, 40, ?)$

TABLE I: Parameters (n, k, ρ), n: length of the cover, k: length of the hidden message,ρ: covering radius. t: error-correcting capacity.
First table: known results on 2-ECC BCH codes and Hamming codes.
Second table: our results on 3-ECC and 4-ECC on binary Goppa codes and BCH codes.

code	n	k	r_a	$r_{\max}$	it_a	$it_{\max}$	ϵ_{eff}	ϵ_r
BCH	15	10	$3,3$	5	$2,15$	19	3	$0,67$
Goppa	15	12	$4,52$	6	$43,6$	527	$2,7$	$0,8$
BCH	31	15	$4,28$	5	$30,5$	227	$3,5$	$0,48$
Goppa	31	15	$4,08$	6	$16,9$	502	$3,7$	$0,48$
BCH	63	18	$4,06$	5	27	648	$4,4$	$0,28$
Goppa	63	18	$3,87$	6	$10,7$	2041		$0,28$
BCH	127	21	$3,85$	5	9	276	$5,5$	$0,16$
Goppa	127	21	$3,85$	5	$7,5$	169	$5,5$	$0,16$
BCH	255	24	$3,83$	5	8	524	$6,3$	$0,095$
Goppa	255	24	$3,83$	5	$6,9$	307	$6,3$	$0,095$
BCH	511	27	$3,83$	5	$7,5$	1027	$7,05$	$0,05$
Goppa	511	27	$3,83$	5	$6,7$	540	$7,05$	$0,05$
BCH	1023	30	$3,83$	5	$7,5$	1074	$7,8$	$0,03$
Goppa	1023	30	$3,83$	5	$6,4$	1044	$7,8$	$0,03$

TABLE II: BCH and Goppa, $\delta = 7$, $t = 3$.

code	n	k	r_a	$r_{\max}$	it_a	$it_{\max}$	ϵ_{eff}	ϵ_r
BCH	15	14	$5,93$	7	96	542	$5,9$	$0,93$
BCH	31	20	$6,06$	7	340	4643	$6,06$	$0,645$
Goppa	30	19	$5,79$	7	250	2192	$5,79$	$0,63$
BCH	63	24	$5,59$	7	159	4998	$5,6$	$0,38$
Goppa	63	24	$5,58$	7	145	4362	$5,6$	$0,38$
BCH	127	28	$5,28$	7	81	8134	$5,3$	$0,22$
Goppa	127	28	$5,3$	7	$89,3$	8534	$5,3$	$0,22$
BCH	255	32	$5,06$	6	56	1281	$6,3$	$0,12$
Goppa	255	32	$5,06$	6	$57,5$	307	$6,3$	$0,12$
BCH	511	36	$4,97$	6	$35,5$	1281	$7,2$	$0,07$
Goppa	511	36	$4,97$	6	$35,5$	540	$7,2$	$0,07$
BCH	1023	40	$4,95$	6	$27,5$	1157	$8,1$	$0,039$
Goppa	1023	40	$4,95$	6	$27,5$	1044	$8,1$	$0,039$

TABLE III: BCH and Goppa, $\delta = 9$, $t = 4$.

9 respectively. These results were obtained by testing 100000 inputs (random covers and random messages) for each code.

The different values given in these tables are:
• n: the length of the code (*i.e.*, of the length of the steganographic cover),
• k: the co-dimension of the code, (*i.e.*, the length of the steganographic message),
• r_a: the average of the number of modified symbols,
• $r_{\max}$: the maximum number of modified symbols,
• it_a: the average of the number of iterations of the decoding algorithm,
• $it_{\max}$: the maximum number of iterations of the decoding algorithm,
• ϵ_{eff}: the embedding efficiency (*i.e.*, the number of embedded bits per unit bit of distortion)
• ϵ_r: the average of embedding rate.

Goppa codes are known to be asymptotically good (in term of ratio between the minimum distance and the dimension of the codes), contrary to t BCH codes. However, for our range use, it turns out that there is no significative difference between the parameters of these two families of codes. So, it is not surprising that the experimental results are similar for these two classes of codes.

The specificity of these families of codes come only from the existence of an algebraic decoding algorithm.

An iteration of our algorithm consists essentially to decode a BCH code or a Goppa code of small error correcting capacity (until $t = 4$). These decoders are implemented in many hardware and software applications. We use the function "Decode" of the Magma Computer Algebra system, which is a not optimized implementation, but a generic implementation of a decoder for GRS / Alternant codes. Depending on the parameters of the code, the encoding map needs between 1.5 and 20 seconds. An optimized C implementation will take less than one second in any case. The retrieval map is just the computation of a syndrome, as usually for the matrix encoding.

The graph in Figure 1 represents the embedding efficiency given as a function of embedding rate. We compare our results to those obtained from previous works based on: Hamming codes (F5) [17], BCH 2-errors correcting codes [19], [16] and Golay codes [11]. These results show that 3-correcting BCH codes improves the results of existing implementations. The 4-correcting BCH give poorer results, probably because the number of changes to make is too great.

In this paper, we deliberately limited our study to the binary case. So, we limit our comparisons to other binary codes with a computationally effective implementation. Fridrich et al. [8] explain how the use of non-binary codes will increase the embedding efficiency, in particular for large payload (*i.e.*, embedding rate). A natural extension of our work will be to test ternary BCH and Goppa codes in order to compare with the results presented in [9]. However, in the ternary case, the enumeration of supplementary errors is more complex.

V. CONCLUSION

In this paper, we have presented a new method for steganography. This method is based on a complete decoding algorithm,

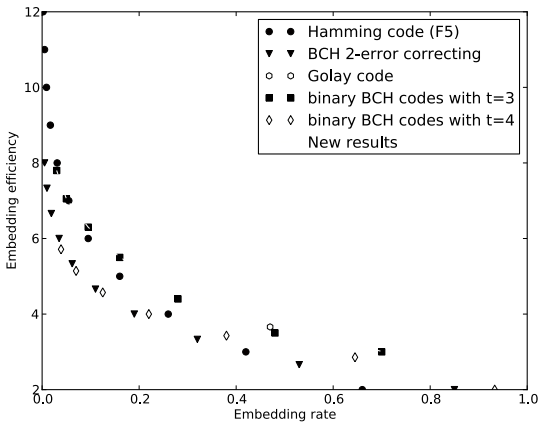

Fig. 1: Performance comparison.

which uses an exhaustive search aided by an algebraic decoding algorithm. This method is practicable for codes with small minimum distance (typically, d less than 10).

Our examples, based on Goppa codes and BCH codes, show that we are able to improve some previous results and to propose new sets of parameters for matrix encoding based on binary codes, especially for high embedding rates.

REFERENCES

[1] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems", *IEEE Transactions on Information Theory*, vol. 24 (3), pp. 384-386, 1978.

[2] J. Bierbrauer and J. Fridrich, "Constructing good covering codes for applications in steganography", in Transactions on Data Hiding and Multimedia Security III, LNCS vol. 4920 , pp. 1-22, Springer, 2008.

[3] N. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece based digital signature scheme", Asiacrypt 2001, LNCS vol. 2248, pp. 157-174. Springer, 2011.

[4] R. Crandall, "Some notes on steganography", available at http://os.inf.tu-dresden.de/ west- feld/crandall.pdf, 1998.

[5] C. Fontaine and F. Galand, "How Reed-Solomon Codes Can Improve Steganographic Schemes", EURASIP Journal on Information Security Vol. 2009, Article ID 274845, special issue "Secure Steganography in Multimedia Content" 2009.

[6] J. Fridrich, "Steganography in Digital Media: Principles, Algorithms, and Applications", Cambridge University Press, 2009.

[7] J. Fridrich and P. Lisonek, "Grid colorings in steganography", IEEE Transactions on Information Theory, vol. 53, (4), pp. 1547-1549, 2007.

[8] J. Fridrich, P. Lisonek, D. Soukal, "On steganographic embedding efficiency", Information Hiding 2006, LNCS vol. 4437, pp. 282-296, Springer, 2007.

[9] J. Fridrich and D. Soukal, "Matrix embedding for large payloads", IEEE Transactions on Information Forensics and Security, vol. 1 (3), pp. 390-395, 2006.

[10] F. Galand and G. Kabatiansky, "Information hiding by coverings", in Proceedings of IEEE Information Theory Workshop (ITW '03), pp. 151-154, Paris, France, 2003.

[11] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography", Information Hiding 2006, LNCS vol. 4437, pp. 314-327, Springer 2007.

[12] A. McLoughlin, "The complexity of computing the covering radius of a code", IEEE Transactions on Information Theory, vol. 30 (6), pp. 800-804, 1984.

[13] Magma Computer Algebra. http://magma.maths.usyd.edu.au/magma/

[14] C. Munuera, "Steganography and error-correcting codes", Signal Process. 87 (2007) pp. 1528-1533, available online at http://www.sciencedirect.com.

[15] H. Rifà-Pous and J. Rifà, "Product perfect codes and steganography", Digtal Signal Processing, vol. 19 (4), pp. 764-769, July, 2009.

[16] D. Schönfeld and A. Winkler, "Embedding with syndrome coding based on BCH codes", In: Proceedings of the 8th ACM Workshop on Multimedia and Security, pp. 214-223, 2006.

[17] A. Westfeld, "High capacity despite better steganalysis (F5 steganographic algorithm)", Information Hiding 2001, LNCS vol. 2137, pp. 289-302, Springer, 2001.

[18] W. Zhang and S. Li, "A coding problem in steganography", Designs, Codes and Cryptography, vol. 46 (1), pp. 67-81, 2008.

[19] R. Zhang, V. Sanchev and H. J. Kim, "Fast BCH Syndrome Coding for Steganography", Information Hiding 2009, LNCS vol. 5806, pp. 48- 58, 2009.

[20] X. Zhang and S.Wang, "Stego-encoding with error correction capability", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E88-A, (12), pp. 3663-3667, 2005.

 ISBN: 978-1-61208-204-2