JP Journal of Algebra, Number Theory and Applications

https://pphmjopenaccess.com
https://doi.org/10.17654/0972555526002 **P-ISSN: 0972-5555**
Volume 65, Number 1, January 2026, Pages 21-34 **E-ISSN: 3049-2084**

LIST DECODING OF GOPPA CODES IN STEGANOGRAPHY

Mohamed Bouye[1], Tariq Alraqad[2], Hicham Saber[2] and Abdelkader Moumen[2,*]

[1]Department of Mathematics
King Khalid University
P.O. Box 9004, Abha
Saudi Arabia

[2]Department of Mathematics
College of Science
University of Ha'il
Ha'il 55473, Saudi Arabia
e-mail: mo.abdelkader@uoh.edu.sa

Abstract

Matrix encoding (or syndrome coding) represents a general coding theory process applied on steganographic schemes to reduce distortion during embedding and improve embedding efficiency and security.

Received: September 5, 2025; Accepted: October 29, 2025
2020 Mathematics Subject Classification: 94A24, 11T71.
Keywords and phrases: steganography, error-correcting codes, Goppa codes, list-decoding, digital communications, finite fields.
*Corresponding author
Communicated by K. K. Azad

How to cite this article: Mohamed Bouye, Tariq Alraqad, Hicham Saber and Abdelkader Moumen, List decoding of Goppa codes in steganography, JP Journal of Algebra, Number Theory and Applications 65(1) (2026), 21-34. https://doi.org/10.17654/0972555526002

Published Online: November 17, 2025

The disruption of statistic properties of the cover object is not considerable when a smaller number of embedding changes occur, and the steganographic security of schemes utilizing matrix embedding seems to be better. As a result, embedding efficiency, which is the number of bits integrated on the number of changes introduced into the cover-object, has become a very important parameter in the field of steganography. In this article, we propose a new steganographic scheme by investigating the possibility of remotely transferred hidden information and subsequently increasing the embedding efficiency, and extending steganography construction and error correcting code. In this approach, we utilize the list-decoding algorithm up to the Johnson radius for binary Goppa codes to hide a message in a cover image. We evaluate the embedding efficiency of our approach, and then use it to compare our method with the previous works and the theoretical upper bound.

1. Introduction

Steganography (also known as “covered writing”) is considered as both art and science of the field of hidden communications. This notion is often combined with cryptography (also known as “secret writing”), in order to achieve a major goal which is keeping information secure from unauthorized access by a third party. While in cryptography, the opposing party can see and intercept the transmitted information [13], the communication in steganography must be kept secret. In order to achieve this, secret messages are embedded into other ones, seemingly inoffensive messages (the covers). Nowadays, standard covers are computer files, primarily image, video and audio files; more generally any electronic document containing irrelevant or redundant information is a good candidate to cover the hiding secrets. Error-correcting codes are applied in order to detect and correct the errors or erasures taking place while transmitting in data. For more information on the relations between steganographic systems and error-correcting codes, see [1, 7, 8].

Linear codes are commonly used in steganography, but the non-linear case is also considered [14]. The investigation of parity check matrix plays a

vital role in designing better steganographic protocols. Westfeld in his work [2], which is considered among the first studies that uses the notion of matrix encoding, presented an algorithm (F5) that utilizes matrix encoding to reduce modification of the quantized DCT coefficients. Since then, reducing the capacity of embedding gained much more importance for steganographers [3-6]. The approach in F5 is based on modifying at most one bit from a nonzero coefficient to hide b bits. For instance, using a matrix encoding technique, among seven coefficients, at most one coefficient is modified in order to hide three bits. This $(7, 3)$ steganographic protocol is based on Hamming code [7, 4]. Therefore, the image distortion is reduced, however the embedding capacity decreased. But now by using an $(a, b, 1)$ code, where $a = 2^b - 1,$ it is not necessary to modify all coefficients.

Modified matrix encoding (MME), introduced in [4], uses $(a, b, 2)$ code, where two coefficients can be changed in each group. The key idea of the matrix encoding technique is that "reducing the number of modifications of the DCT coefficients leads to reducing the image's distortion" [2]. For the realization of matrix encoding, many efficient codes have been used later on. See for example BCH error-correcting code [3], product perfect codes [8], and Reed-Solomon (RS) [5]. Munuera [6] studied error-correcting codes, and steganographic systems, it is proved in [7] that one can find a correspondence relationship between perfect error correcting codes and the MLE (maximum length embeddable) codes. Moreover, the work of Sudan [9] in 1997, brought the first list-decoding algorithm in order to produce Reed-Solomon codes that have a lower rate, but still positive. Due to the fact that the correction radius of the codes generated by the Sudan's algorithm is larger compared to the ones achieved while using unambiguous decoding algorithms, this was considered as a very important milestone in investigating list-decoding, which was before at the level of theoretical study. For more information regarding "capacity" of list-decoding see [10] and references therein. Afterwards, in [11], Guruswami and Sudan added a multiplicity condition in the interpolation procedure to improve the previous algorithm. This helped increasing the correction radius of Sudan's algorithm for Reed-Solomon

codes of any rate [11]. The number of errors that can be list-decoded by this algorithm is the Johnson radius.

In this work, we introduce a steganographic technique based on list-decoding of binary Goppa codes. Our approach utilizes the list-decoding of alternant codes combined with a remarkable special case of the classical Goppa codes. For applications, we focus on binary Goppa codes $(q = 2)$, which permits to obtain an effective decoding algorithm (Algorithm 2). This algorithm allows us to list-decode, up to the q-ary Johnson bound, any alternant code, which is better when it is compared to the error correction capacities of other algorithms. Algorithm 3 illustrates an efficient steganography protocol that is obtained by combining Algorithms 1 and 2.

Section 2 is devoted to review the standard steganograpic application of coding theory, and we recall the list-decoding technique in Section 3. Section 4 presents our steganographic scheme and also presents the results experimentally obtained on some classical binary Goppa codes with a comparison with the F5 scheme and the BCH scheme. The conclusion of this article is given in Section 5.

2. Error-correcting Codes in Steganography

Suppose that a and b are positive integers with $b \leq a$, and that S is a finite set. A (a, b) embedding/retrieval steganographic protocol over S is defined by a pair of maps $Emb : S^b \times S^a \to S^a$ and $Ret : S^a \to S^b$ such that for all $s \in S^b$ and $v \in S^a$, we have $Ret(Emb(s, v)) = s$. The map Emb, (resp. Ret) is called the embedding (resp. the retrieval) map. The radius ρ of the protocol is defined to be the maximum of a set $\{d(v, Emb(s, v)); s \in S^b, v \in S^a\}$, where d denotes the Hamming distance. We use (a, b, ρ) to abbreviate such protocol. The embedding map of (a, b, ρ) protocol enables us to hide a message of length b within a string cover of length a, by modifying at most ρ symbols of the cover [6].

Let a and b be positive integers with $b \leq a$, and q be a prime power. An $[a, a-b]$ linear code C over the finite field $\mathbb{F}_q$ is a linear subspace of $\mathbb{F}_q^a$, with co-dimension b. We define the covering radius ρ for C by $\rho = \max_{u \in \mathbb{F}_q^a} \{d(u, C)\}$, where $d(u, C)$ stands for the minimum Hamming distance from u to C. The support of $u = (u_1, u_2, ..., u_a) \in \mathbb{F}_q^a$ is the set $Supp(u) = \{i \,|\, u_i \neq 0\}$.

Suppose that C is an $[a, a-b]$ linear code over $S = \mathbb{F}_q$, and denote its parity check matrix by H. The syndrome of a vector $u \in S^a$ is defined by $r(u) = H \times u^T$. The set of elements in $\mathbb{F}_q^a$ with the same syndrome as u is the coset $C + u$, and a leader of the coset $C + u$ is a vector $l_{r(u)} \in C + u$ having minimum weight. Note that a leader is not necessarily unique. The retrieval map of a steganographic protocol of type (a, b, ρ) is the syndrome map $Ret : \mathbb{F}_q^n \to \mathbb{F}_q^k$, defined by $Ret(u) = H \times u^T$. The term linear is used to emphasize the fact that the retrieval map Ret is linear. We apply the next algorithm to calculate $Emb(m, u)$ for a linear (a, b, ρ) protocol [6]:

Algorithm 1: Coset steganographic algorithm.

Needed: an algorithm to decode a coset: input a syndrome v, output: a coset leader l_v

Input: a message m of size b with a cover u of size a.

Output: a steganographic cover $Emb(m, u)$ of m having a distortion $d(u, u')$ as minimal as possible.

1: Calculate $v := Ret(u) - m$,

2: put $c := u - l_v$,

3: **return** $Emb(m, u) := c$.

---

3. Binary Goppa Codes

Let a, b be natural numbers with $b \le a$, and let q be a prime power. It is known that if H is a matrix of type $(a-b)\times a$ with coefficients in $\mathbb{F}_q$ and having rank $a-b$, then $C=\{c\in\mathbb{F}_q^n \mid Hc^T=0\}$, is an $[a, b]$ linear code over $\mathbb{F}_q$, with length equals to a and dimension equals to b, and parity check matrix H. A $b\times a$ matrix M is said to be *generator* of C if its rows form a basis of C. A linear code of length a and dimension b over $\mathbb{F}_q$ will be denoted by $[a, b]_q$.

3.1. Definition of the Goppa codes

Consider a polynomial $g(x)=\sum d_i x^i \in \mathbb{F}_q[x]$ and let $L=\{\beta_1, \beta_2, ..., \beta_n\}$, where $\beta_1, \beta_2, ..., \beta_n$ are elements of $\mathbb{F}_q$ that are not roots of $g(x)$. The Goppa code $\Gamma(L, g)$ is defined as the set of all vectors $w=(w_1, w_2, ..., w_n)\in\mathbb{F}_q^n$ satisfying

$$\sum_{i=0}^{n}\frac{w_i}{x-\beta_i}=0 \bmod g(x).$$

It is known that [12] a parity check matrix for $\Gamma(L, g)$ is given by

$$H=\begin{pmatrix} \frac{1}{g(\beta_1)} & \frac{1}{g(\beta_2)} & \cdots & \frac{1}{g(\beta_n)} \\ \frac{\beta_1}{g(\beta_1)} & \frac{\beta_2}{g(\beta_2)} & \cdots & \frac{\beta_n}{g(\beta_n)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\beta_1^{r-1}}{g(\beta_1)} & \frac{\beta_2^{r-1}}{g(\beta_2)} & \cdots & \frac{\beta_n^{r-1}}{g(\beta_n)} \end{pmatrix}.$$

The code $\Gamma(L, g)$ is called a *Goppa code of degree* $r=\deg(g)$, and it is said to be *irreducible* provided that $g(x)$ is irreducible over $\mathbb{F}_q$.

Additionally, the Goppa code is called *maximal* if $L = \{\beta_1, \beta_2, ..., \beta_n\}$ consists of all of $\mathbb{F}_q$ except the roots of $g(x)$.

3.2. List decoding to binary Goppa codes

Some of the direct applications of this algorithm are binary Goppa codes defined with a square-free polynomial *g*. Indeed, since we have

$$\Gamma_2(L, g) = \Gamma(L, g^2)$$

both codes benefit at least from the distance of $\Gamma(L, g^2)$ and the dimension of $\Gamma_2(L, g)$ and [10-12].

Algorithm 2: List-decoding of binary Goppa codes

Needed: $L = \{\beta_1, \beta_2, ..., \beta_n\}$. A square-free Goppa polynomial g. The $C = \Gamma_2(L, g)$ associated Goppa code. The corresponding map of evaluation *eu*. The Johnson radius of *C*

Input: The received word $x \in \mathbb{F}_q^n$

Output: the codewords at distance inferior or equal to the Johnson radius of *x*

1: identification of $\Gamma_2(L, g)$ with $\Gamma(L, g^2)$

2: Taking the Generalised Reed-Solomon code $GRS(n, k)$ above $\Gamma_2(L, g)$

3: Finding the codewords at a distance less than *x* by list decoding of alternant codes up to the Johnson radius

---

4. Steganography based on Goppa Codes

In this section, we consider the suggested method for hiding information into the spatial domain of an image of gray scale. This technique is built on the division of the cover into equally sized blocks. One can represent a binary data block (such as LSB values of cover data) $\{u_0, u_1, ..., u_{n-1}\}$ over

$\mathbb{F}_2$ by an element of $\mathbb{F}_2[X]$ as follows: $u(X) = u_0 + u_1X + \cdots + u_{n-1}X^{n-1}$. The message *m* when embedded into *u*, the cover data, generates *s*, the stego data, that we represent by $s(X) = s_0 + s_1X + \cdots + s_{n-1}X^{n-1}$. Thus, the message *m* and the stego data *s* are related by the formula:

$$m = s \times H^T. \tag{1}$$

Also this formula is used to extract *m* from *s*. When we hide the message into the cover, the bits of the cover data might be changed from one to zero and vice versa. If $e(X)$ represents the flipping pattern that depicts the positions of the changed bits [2], then the modification of the stego data occurs in the following way:

$$r(X) = u(X) + e(X). \tag{2}$$

Therefore, combining equation (1) and equation (2) leads to

$$m - u \times H^T = e \times H^T. \tag{3}$$

We refer to the quantity in equation (3) as syndrome and denote that by *S*; that is

$$S = m - u \times H^T = e \times H^T. \tag{4}$$

Our main objective, from the steganographic perspective, is decreasing the distortion by finding $e(X)$ that satisfies equation (4) and having a minimal number of flips. This process is called the syndrome coding. Applying error-correcting code to hide the data gives a solution of the equation (4) that depends on the vector *e*. This provides the required positions of the bits that need to be changed in the vector $u(X)$ to hide the message *m* within the vector $u(X)$. Equation (2) is used to calculate the stego vector $r(X)$. While we apply equation (1) to extract the hidden message *m* from $r(X)$. One way to solve this problem, for our code, is designing an algorithm that decodes any syndrome or a good proportion of them at least. That is why we have chosen the Goppa codes. The maximum

distance to a given linear code is called the covering radius, and it is usually hard to be determined exactly. As a matter of fact, it is an old and difficult open problem for Goppa codes. However, we are not interested in its exact value. For $q = 2$, let $e_{\inf}(n, d) = \lceil n - \sqrt{n(n-d)} \rceil - 1$, with d being the minimum distance of the code, and n is the length. This is the number of errors that Algorithm 2 can list-decode, it is called the Johnson radius and is only a lower bound for the covering radius. In fact, by considering a proper value of the alphabet's size q, one can improve this bound to

$$e_q(n, d) = \left\lfloor \theta_q \left(n - \sqrt{n\left(n - \frac{d}{\theta_q} \right)} \right) \right\rfloor - 1$$

with $\theta_q = 1 - \frac{1}{q}$. Refer to ([15], Chapter 3) for more information on bounds that relate list-decoding radius to minimum distance.

Algorithm 3: Proposed Embedding Scheme

Needed : a list decoding algorithm *dec* up to the Johnson radius. For a given u it generates a boolean value $\mathrm{dec}1(u)$ together with a vector $\mathrm{dec}2(u)$: "true", $c \in C$ satisfying $d(c, u) \geq t$ when succeeding, and "false", u otherwise.

Input: a message m of size k and a cover u of size n.

Output: a steganographic cover $u' = e(m, u)$ of m having distortion $d(u, u')$ as small as the Johnson radius.

1: The computation of $v := r(u) - m$,

2: The computation of x such that $r(x) = v$

3: Decoding x by using Algorithm 2. Setting $c \in C$ the output of the decoding algorithm.

4: Letting $e = x - c$ the error vector

5: return $e(m, u) = u + e$

4.1. Performance and comparison

In this section we treat the efficiency of the embedding offered by the Goppa codes to steganography. The efficiency of the embedding is defined as the number of embedded bits divided by the number of modified bits. For the steganographic protocol (n, k, ρ), the ratios $\alpha = \frac{k}{n}$ and $e = \frac{k}{\rho}$ are called the relative capacity and the embedding efficiency, respectively.

Table 1 presents the experimental results of simulations on several different binary Goppa codes up to the binary Johnson bound. These results were achieved by testing, for each code, 100000 random messages and random covers. In these tables we used the following notations:

- k : the code's co-dimension, (meaning the length of the steganographic message),
- n : the code's length (meaning the steganographic cover's length),
- r_{code} : the degree of the generator polynomial
- ρ_a : the average of the number of changed symbols,
- $\rho_{\max}$: the maximum number of changed symbols,
- α: the relative capacity,
- e: the embedding efficiency (meaning the number of embedded bits per unit bit of distortion).

Table 1. New sets of parameters for steganographic scheme via list-decoding algorithm of binary Goppa codes

N	k	r_{code}	ρ_a	$\rho_{\max}$	e	α
16	12	3	3	5	2.4	0.75
32	30	6	4	9	3.333	0.937
64	48	10	5	10	4.8	0.75
64	60	8	7	13	4.615	0.937
128	105	18	9	18	5.833	0.820

128	126	15	11	22	5.727	0.984
256	208	45	15	30	6.933	0.812
256	248	35	18	36	6.888	0.968
512	315	55	19	38	8.289	0.615
512	405	45	25	50	8.1	0.791
1024	500	60	27	53	9.433	0.488
1024	600	50	37	74	8.108	0.585
2048	1100	100	53	105	10.476	0.537
2048	1320	120	64	128	10.312	0.644

Table 2 gives a comparison between the theoretical parameters of steganographic protocols based on Hamming codes (F5 [2]), steganographic scheme via list-decoding algorithm of Goppa codes, and 2 errors correcting BCH codes [3]. Here the third value represents the maximum number of modification given by our decoding algorithm.

Table 2. Comparison between our steganographic scheme with the binary Goppa codes and Hamming, BCH steganographic scheme

BCH [3]	Hamming [2]	Binary Goppa codes
(15, 8, 3)	(15, 4, 1)	(16, 12, 5)
(31, 10, 3)	(31, 5, 1)	(32, 30, 9)
(63, 12, 3)	(63, 6, 1)	(64, 48, 10)
(127, 14, 3)	(127, 7, 1)	(128, 105, 18)
(255, 16, 3)	(255, 8, 1)	(256, 208, 30)
(511, 18, 3)	(511, 9, 1)	(512, 315, 38)
(1023, 20, 3)	(1023, 10, 1)	(1024, 500, 53)

We illustrate the embedding efficiency as a function of α in Figure 1, for BCH codes, the binary Golay code, the binary Hamming code, and our method (list-decoding of binary Goppa code in steganography). If we fix α, then by using the sphere bound from [7], one can obtain the following upper bound on e:

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

where $\mathcal{H}^{-1}(\alpha)$ stands for the inverse of the function $\mathcal{H}(x) = -x\log_2 x - (1-x)\log_2(1-x)$ known as the binary entropy function on the interval $\left[0, \frac{1}{2}\right]$. Note that since the covering radius ρ provides an upper bound on the average number of embedding changes ρ_a, as defined in this paper, we see that we have a lower bound on the embedding efficiency $\frac{k}{\rho_a}$ as usually defined in steganographic literature. From our point of view, it is more convenient to work with the simpler invariant ρ. The situation is similar to the setting of the classical coding theory where we mostly use the minimum distance d to find bounds on the error probability instead of the average distance. In fact, $\frac{k}{\rho_a} - e$ is small, and when the length increases, it tends to 0.

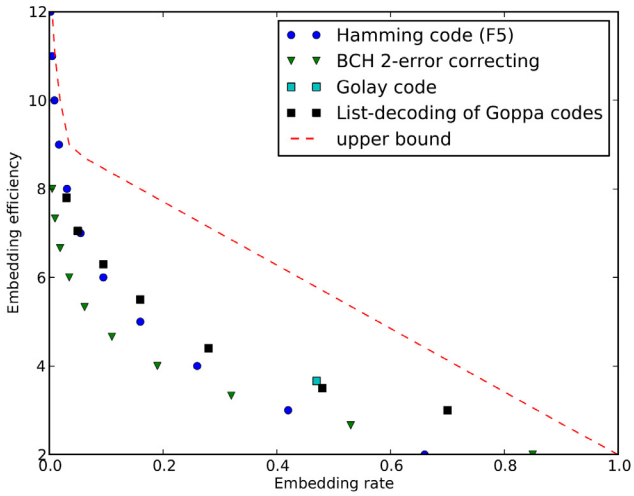

Figure 1. F5 steganographic scheme, Golay, BCH steganographic scheme and our method.

5. Conclusion

In this work, we propose a new construction of a steganographic protocol

that extends steganography construction and the error correcting code. Our method is based on using the list-decoding introduced in [9-12], implemented on Goppa codes for embedding the message in the cover image, where the function that recovers the message is based on syndrome coding. We have shown that Goppa codes are very good candidates for constructing efficient steganographic schemes. In steganography applications, the achievements of algebraic-geometric codes are, in general, remarkably better than a simple list-decoding technique in terms of performance (higher embedding efficiency). A future work will be to investigate more into geometric codes.

References

[1] R. Crandall, Some notes on steganography, 1998. Available at http://os.inf.tu-dresden.de/westfeld/crandall.pdf.

[2] A. Westfeld, High capacity despite better steganalysis (F5 – A steganographic algorithm), Lecture Notes in Comput. Sci., Vol. 2137, Springer-Verlag, 2001, pp. 289-302.

[3] R. Zhang, V. Sanchev and H. J. Kim, Fast BCH syndrome coding for steganography, S. Katzenbeisser and A.-R. Sadeghi, eds., Information Hiding 2009, IH'2009, LNCS 5806, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 48-58.

[4] Y. Kim, Z. Duric and D. Richards, Modified matrix encoding technique for minimal distortion steganography, J. L. Camenisch, C. S. Collberg, N. F. Johnson and P. Sallee, eds., IH 2006, LNCS, Vol. 4437, 2007, pp. 314-327.

[5] C. Fontaine and F. Galand, How Reed-Solomon codes can improve steganographic schemes, EURASIP Journal on Information Security 2009 (2009), Article ID 274845. doi:10.1155/2009/274845.

[6] C. Munuera, Steganography and error-correcting codes, Signal Processing 87 (2007), 1528-1533.

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

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

[9] Madhu Sudan, Decoding of Reed-Solomon codes beyond the error-correction bound, Journal of Complexity 13(1) (1997), 180-193.

[10] Peter Elias, Error-correcting codes for list decoding, Information Theory, IEEE Transactions on 37(1) (1991), 5-12.

[11] V. Guruswami and M. Sudan, Improved decoding of Reed-Solomon and algebraic geometry codes, Information Theory, IEEE Transactions on 45(6) (1999), 1757-1767.

[12] D. Augot, M. Barbier and A. Couvreur, List-decoding of binary Goppa codes up to the binary Johnson bound, IEEE, ITW'11, 2011.

[13] H. J. Highland, Data encryption: a non-mathematical approach, Comput. Secur. 16 (1997), 369-386.

[14] H. Jouhari and E. M. Souidi, A novel embedding scheme based on Walsh Hadamard transform, Journal of Theoretical and Applied Information Technology 32 (2011), 55-60.

[15] V. Guruswami, List Decoding of Error-Correcting Codes, Winning Thesis of the 2002 ACM Doctoral Dissertation Competition, Volume 3282 of Lectures Notes in Computer Science, Springer, 2004.