BCAM

Post-Quantum Cryptography

Decoding Alternant Codes

S. Xambó

UPC

14/9/2018

- Notations and conventions: Finite fields, Relative control matrices.
- Alternant codes. Special families: RS, GRS, BCH, Goppa.
- The PGZ decoders (PGZa and PGZb). Notice on the BMS decoder.
- Philosophy: Balance between mathematical theory and effective computations.
- Main reference: R. Farré, N. Sayols, S. Xambó-Descamps: "On the PGZ decoding of alternant codes". *Computational and Applied Mathematics*, 2018 (in press).
- For the computational environment, see https://mat-web.upc.edu/people/sebastia.xambo/PyECC.html.

It will be the computational support of the second edition of *Block error-correcting codes* (SX, Springer, 2003) planned for Spring 2019.

K: A finite field and $q = |K|$ (so $K \simeq F_q$)

F: a finite extension of K and $m = [F/K]$ (so $F \simeq F_{q^m}$).

$F = K[X]/(f)$, $f = X^m + f_1 X^{m-1} + \cdots + f_{m-1}X + f_m$ irreducible$/K$.

As a K-vector space, $F = \langle 1, \alpha, \ldots, \alpha^{m-1} \rangle_K$, where $\alpha = [X]_f$ (polynomial expressions of degree $< m$ in α). Multiplication is carried out as the multiplication of polynomial expressions, but with the reduction rule $\alpha^m = -(f_1\alpha^{m-1} + \cdots + f_{m-1}\alpha + f_m)$.

If $a = a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1} \in F$, we will write $[a]_K = [a_{m-1}, \ldots, a_1, a_0] \in K^m$ (or simply $[a]$). The map $F \to K^m$, $a \mapsto [a]$, is a K-linear isomorphism.

Gauss formula. The number of monic irreducible polynomials $f \in K[X]$ of degree m is

$$N_q(m) = \frac{1}{m} \sum_{d|m} \mu(d) q^{m/d} = \frac{q^m}{m} + \cdots$$

m	f	$N_2(m)$
2	X^2+X+1	1
3	X^3+X+1	2
4	X^4+X+1	3
5	X^5+X^2+1	6
6	X^6+X+1	9
7	X^7+X+1	18
8	$X^8+X^4+X^3+X+1$	30
9	X^9+X+1	56
10	$X^{10}+X^3+1$	99
11	$X^{11}+X^2+1$	186
12	$X^{12}+X^3+1$	335
13	$X^{13}+X^4+X^3+X+1$	630
14	$X^{14}+X^5+1$	1161
15	$X^{15}+X+1$	2182
16	$X^{16}+X^5+X^3+X+1$	4080

$H \in F(r,n)$: an $r \times n$ matrix with entries in F (*control matrix*).

For $y \in F^n$, $s_H(y) = yH^T \in F^r$ (*syndrome* of y)

$C = C_K(H) = \{x \in K^n \mid s_H(x) = 0\} = C_F(H) \cap K^n$

(*linear code*/K associated to H)

'Blow' H relative to K by replacing each of its entries h_{ij} by the column vector $[h_{ij}]^T$. If we let $[H] = [H]_K \in K(rm, n)$ be the resulting matrix, then we have

$$C_K(H) = C_K([H]), \quad k = \dim(C_K(H)) = n - \mathrm{rank}([H]).$$

In the special case $F = K$ (equivalent to $m = 1$), $[H] = H$ and $k = n - \mathrm{rank}(H)$.

Solving the homogeneous linear system $x[H]^T = 0$ provides a *generating matrix* G of C. Its rows form a linear basis of C.

```
K = Zn(2); [KX,X] = polynomial_ring(K)
f = X**5 + X**2 + 1
[F,x] = extension(K,f,'x')

H = matrix(geometric_series(x**3,11))
bH = blow(H) =>
[[0 0 0 1 0 1 0 1 1 0 1]
 [0 1 1 1 1 1 0 1 1 1 0]
 [0 0 0 0 1 1 0 0 1 0 0]
 [0 0 1 1 1 1 1 0 1 1 1]
 [1 0 0 0 0 1 1 0 0 1 0]] :: Matrix[K]

rank(bH) => 5
```

So $k = 6$.

```
[[1 0 1 1 1 1 0 0 0 0 0]
 [0 1 0 1 1 1 1 0 0 0 0]
 [0 0 1 0 1 1 1 1 0 0 0]
 [0 0 0 1 0 1 1 1 1 0 0]
 [0 0 0 0 1 0 1 1 1 1 0]
 [0 0 0 0 0 1 0 1 1 1 1]] :: Matrix[K]
           * *     *
```

Remark. Since the sum of the columns marked with $*$ is zero, the minimum weight of C is at most 3, and actually it is 3 because any two columns are distinct.

$u \in K^k$: *information vector*.

$x = uG \in C$: *code vector* ('sent vector')

$e \in K^n$: *error vector*. The number of non-zero entries of e is denoted $|e|$ (*weight* of e).

$y = x + e$: 'received vector'.

$s = s_H(y) = s_H(e) \in F^r$: *syndrome vector*

The decoding problem is to find an algorithm (*decoder*) that takes s as input and delivers x (hence also u). If this can be accomplished for all e such that $|e| \leqslant t$, we say that the decoder *corrects up to t errors*.

Recall: For general linear codes, the decoding problem is NP-complete (Berlekamp-McEliece-van Tilborg, 1978).

Let $\alpha_1, \ldots, \alpha_n$ and $h_1, \ldots, h_n$ be elements of F such that $h_i \neq 0$ for all i and $\alpha_i \neq \alpha_j$ for all $i \neq j$. Consider the matrix

$$H = V_r(\alpha_1, \ldots, \alpha_n)\mathrm{diag}(h_1, \ldots, h_n) \in F(r, n), \tag{1}$$

that is,

$$H = \begin{pmatrix} h_1 & \ldots & h_n \\ h_1\alpha_1 & \ldots & h_n\alpha_n \\ \vdots & & \vdots \\ h_1\alpha_1^{r-1} & \ldots & h_n\alpha_n^{r-1} \end{pmatrix} \tag{2}$$

We say that H is the *alternant control matrix* of order r associated with the vectors

$$\boldsymbol{h} = (h_1, \ldots, h_n) \quad \text{and} \quad \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n).$$

To make explicit that the entries of $\boldsymbol{h}$ and $\boldsymbol{\alpha}$ (and hence of H) lie in F, we will say that H is defined over F.

The codes $A_K(\boldsymbol{h}, \boldsymbol{\alpha}, r) = C_K(H)$ defined by the control matrix H are called *alternant codes*.

Proposition (Alternant bounds)

If $C = A_K(\boldsymbol{h}, \boldsymbol{\alpha}, r)$, then

$$n - r \geqslant \dim C \geqslant n - rm$$

and

$$d \geqslant r + 1$$

(*minimum distance alternant bound*).

Given a list or vector $\boldsymbol{\alpha}$ of distinct non-zero elements $\alpha_1, \ldots, \alpha_n \in K$, the Reed–Solomon code

$$C = \mathrm{RS}(\boldsymbol{\alpha}, k) \subseteq K^n$$

is the subspace of K^n generated by the rows of the Vandermonde matrix $V_k(\alpha_1, \ldots, \alpha_n)$. It turns out that

$$\mathrm{RS}(\boldsymbol{\alpha}, k) = A_K(\boldsymbol{h}, \boldsymbol{\alpha}, n-k),$$

where $\boldsymbol{h} = (h_1, \ldots, h_n)$ is given by

$$h_i = 1/\prod_{j \neq i}(\alpha_j - \alpha_i). \tag{3}$$

Note that in this case $F = K$, hence $m = 1$, and that the alternant bounds are sharp. Indeed, we have $r = n - k$, hence $k = n - r$, while $n - k + 1 \geqslant d$ (by the Singleton bound) and $d \geqslant r + 1 = n - k + 1$ by the minimum distance alternant bound. In other words, C is MDS (maximum distance separable).

The vector $\boldsymbol{h}$ in the definition of the code $\mathrm{RS}([\alpha_1, \dots, \alpha_n], k)$ as an alternant code is obtained from $\boldsymbol{\alpha}$ by the formula (3). If we allow that $\boldsymbol{h}$ can be chosen possibly unrelated to $\boldsymbol{\alpha}$, but still with components in K, the resulting codes $A_K(\boldsymbol{h}, \boldsymbol{\alpha}, n-k)$ are called *Generalized Reed–Solomon* (GRS) codes, and we will write $\mathrm{GRS}(\boldsymbol{h}, \boldsymbol{\alpha}, k)$ to denote them. An argument as above shows that such codes have type $[n, k, n-k+1]$.

Notice that the code $A_K(\boldsymbol{h}, \boldsymbol{\alpha}, r)$ is the intersection of the GRS code $A_F(\boldsymbol{h}, \boldsymbol{\alpha}, r)$ with K^n.

These codes are denoted $\mathrm{BCH}(\alpha, d, l)$, where $\alpha \in F$ and $d > 0$, $l \geqslant 0$ are integers (called the *designed minimum distance* and the *offset*, respectively).

When $l = 1$, we simply write $\mathrm{BCH}(\alpha, d)$ and say that the it is a *strict* BCH code. The good news here is that

$$\mathrm{BCH}(\alpha, d, l) = A_K(\boldsymbol{h}, \boldsymbol{\alpha}, d-1), \tag{4}$$

where $\boldsymbol{h} = (1, \alpha^l, \alpha^{2l}, \ldots, \alpha^{(n-1)l})$, $\boldsymbol{\alpha} = (1, \alpha, \alpha^2, \ldots, \alpha^{(n-1)})$, $n = \mathrm{period}(\alpha)$.

Let $g \in F[T]$ be a polynomial of degree $r > 0$ and let $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_n \in F$ be distinct non-zero elements such that $g(\alpha_i) \neq 0$ for all i.

The *classical Goppa code* over K associated with g and $\boldsymbol{\alpha}$, which will be denoted $\Gamma(g, \boldsymbol{\alpha})$, can be defined as $A_K(\boldsymbol{h}, \boldsymbol{\alpha}, r)$, where $\boldsymbol{h}$ is the vector $(1/g(\alpha_1), \ldots, 1/g(\alpha_n))$. Thus the minimum distance of $\Gamma(g, \boldsymbol{\alpha})$ is $\geqslant r + 1$ and its dimension k satisfies $n - rm \leqslant k \leqslant n - r$.

The minimum distance bound can be improved to $d \geqslant 2r + 1$ in the case that $K = \mathbb{F}_2$ and the roots of g are distinct.

Let $C = A_K(\boldsymbol{h}, \boldsymbol{\alpha}, r)$ be an alternant code. Let $t = \lfloor r/2 \rfloor$, that is, the highest integer t such that $2t \leqslant r$. For reasons that will become apparent below, t is called the *error-correction capacity* of C.

Let $x \in C$ (*sent vector*) and $e \in F^n$ (*error vector*, or *error pattern*). Let $y = x + e$ (*received vector*). The goal of a decoder is to obtain x from y and H when $l : |e| \leqslant t$. Henceforth we will assume that $l > 0$.

If $e_m \neq 0$, we say that m is an *error position*.

Let $\{m_1, \dots, m_l\}$ be the error positions and $\{e_{m_1}, \dots, e_{m_l}\}$ the corresponding *error values*.

The *error locators* $\eta_1, \dots, \eta_l$ are defined by $\eta_k = \alpha_{m_k}$. Since $\alpha_1, \dots, \alpha_n$ are distinct, the knowledge of the η_k is equivalent to that of the error positions.

The monic polynonial $L(z)$ whose roots are the error locators is called the *error-locator polynomial*. Notice that

$$L(z) = \prod_{i=1}^{l}(z - \eta_i) = z^l + a_1 z^{l-1} + a_2 z^{l-2} + \cdots + a_l, \qquad (5)$$

where $a_j = (-1)^j \sigma_j$, $\sigma_j = \sigma_j(\eta_1, ..., \eta_l)$ the j-th elementary symmetric polynomial in the η_i $(0 \leq j \leq l)$.

We will write $\boldsymbol{a}_l = (a_l, ..., a_1)$.

Recall that the *syndrome* of y is the vector $s = yH^T$, say $s = (s_0, \ldots, s_{r-1})$.

Since $xH^T = 0$, we have $s = eH^T$.

Using the definitions, we easily find that

$$s_j = \sum_{i=0}^{n-1} e_i h_i \alpha_i^j = \sum_{k=1}^{l} h_{m_k} e_{m_k} \alpha_{m_k}^j = \sum_{k=1}^{l} h_{m_k} e_{m_k} \eta_k^j \tag{6}$$

Now we will use the following notations:

$$A_l = \begin{pmatrix} s_0 & s_1 & \dots & s_{l-1} \\ s_1 & s_2 & \dots & s_l \\ \vdots & \vdots & \ddots & \vdots \\ s_{l-1} & s_l & \dots & s_{2l-2} \end{pmatrix} \tag{7}$$

$$\boldsymbol{b}_l = (s_l, \dots, s_{2l-1}). \tag{8}$$

Next proposition establishes the key relation for computing the error-locator polynomial.

Recall that $\boldsymbol{a}_l = (a_l, ..., a_1)$ (see Equation (5)).

Proposition

$$\boldsymbol{a}_l A_l + \boldsymbol{b}_l = 0. \tag{9}$$

Proof

Substituting z by η_i in the identity

$$\prod_{i=1}^{l}(z-\eta_i) = z^l + a_1 z^{l-1} + \ldots + a_l$$

we obtain the relations

$$\eta_i^l + a_1\eta_i^{l-1} + \cdots + a_l = 0,$$

where $i = 1, \ldots, l$. Multiplying by $h_{m_i} e_{m_i} \eta_i^j$ and adding with respect to i, we obtain (using (6)) the relations

$$s_{j+l} + a_1 s_{j+l-1} + \cdots + a_l s_j = 0,$$

where $j = 0, \ldots, l-1$, and these relations are equivalent to the stated matrix relation. □

Remark

In the Equation (9), the matrix A_l turns out to be non-singular and hence it determines $\boldsymbol{a}_l$ (hence also $L(z)$) uniquely, namely

$$\boldsymbol{a}_l = -\boldsymbol{b}_l A_l^{-1}.$$

In next section we are going to establish this fact as a corollary of Eq. (11), whose main outcome is *a fast solution* of Equation (9).

Consider the matrix

$$S = \left(\begin{array}{ccccccc} s_0 & s_1 & \cdots & s_{l-1} & s_l & \cdots & s_t \\ s_1 & s_2 & \cdots & s_l & s_{l+1} & \cdots & s_{t+1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ s_{l-1} & s_l & \cdots & s_{2l-2} & s_{2l-1} & \cdots & s_{t+l-1} \\ \hline \vdots & \vdots & & \vdots & \vdots & & \vdots \\ s_{t-1} & s_t & \cdots & s_{t+l-2} & s_{t+l-1} & \cdots & s_{2t-1} \end{array}\right). \qquad (10)$$

Note that $2t-1 \leqslant r-1$, so that all components are well defined. Note also that the $l \times l$ submatrix at the upper left corner is the matrix A_l defined by Equation (7) an that the column $(s_l, s_{l+1}, \ldots, s_{2l-1})^T$ to its right is the vector $\boldsymbol{b}_l$ defined by Eq. (8).

In next Theorem we use the following notation: $V_s = V_s(\eta_1, \ldots, \eta_l)$. Thus the i-th row of V_s, for $0 \leqslant i \leqslant s-1$, is the vector $(\eta_1^i, \ldots, \eta_l^i)$. We also write $D = \mathrm{diag}(h_{m_1}e_{m_1}, \ldots, h_{m_l}e_{m_l})$.

Theorem

$$S = V_t D V_{t+1}^T. \tag{11}$$

Proof

Let $0 \leqslant i \leqslant t-1$ and $0 \leqslant j \leqslant t$. Then the j-th column of DV_{t+1}^T is the column vector $(h_{m_1} e_{m_1} \eta_1^j, \ldots, h_{m_l} e_{m_l} \eta_l^j)^T$. It follows that the element in row i column j of $V_t D V_{t+1}^T$ is $h_{m_1} e_{m_1} \eta_1^{i+j} + \cdots + h_{m_l} e_{m_l} \eta_l^{i+j} = s_{i+j}$ (by Equation (6)). □

Corollary

The rank of S is l and the matrix A_l is non-singular.

Proof

Since D has rank l, the rank of S is at most l. On the other hand, the theorem shows that $A_l = V_l D V_l^T$ and therefore

$$\det(A_l) = \det(V_l)^2 \det(D) \neq 0.$$

Note that $\det(V_l)$ is the Vandermonde determinant of $\eta_1, \ldots, \eta_l$, which is non-zero because the error locators are distinct. □

Corollary

The Gauss-Jordan algorithm applied to the matrix S returns a matrix that has the form

$$\left(\begin{array}{cccccc} 1 & 0 & \cdots & 0 & -a_l & * \\ 0 & 1 & \cdots & 0 & -a_{l-1} & * \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_1 & * \\ \hline \vdots & \vdots & & \vdots & \vdots & \vdots \end{array}\right) \tag{12}$$

where $*$ denotes unneeded values *(if any)* and the vertical dots below the horizontal line denote that all its elements *(if any)* are zero. This matrix gives at the same time l, the number of errors, and the coefficients of the error-locator polynomial. □

In the descriptions that follow, *Error* means "a suitable decoding-error message" and the function GJ(S) returns the values $-a_l, \ldots, -a_1$ of the matrix (12) as a column vector (this is a conveniently modified form of the Gauss-Jordan procedure).

1. Get the syndrome vector, $s = (s_0, ..., s_{r-1}) = yH^T$. If $s = 0$, return y.
2. Form the matrix S as in the Equation (10).
3. Set $\boldsymbol{a} = -\mathrm{GJ}(S)$ (Equation (12)). After this we have $a_1, ..., a_l$, hence also the error-locator polynomial L.
4. Find the elements α_j that are roots of the polynomial L. If the number of these roots is $< l$, return *Error*. Otherwise let $\eta_1, ..., \eta_l$ be the error-locators corresponding to the roots and set $M = \{m_1, \ldots, m_l\}$, where $\eta_i = \alpha_{m_i}$.
5. Solve for $e_{m_1}, ..., e_{m_l}$ the following system of linear equations:
$$h_{m_1} e_{m_1} \eta_1^j + h_{m_2} e_{m_2} \eta_2^j + ... + h_{m_l} e_{m_l} \eta_l^j = s_j \ (0 \leqslant j \leqslant l-1).$$
If any of the values of e_m is not in K, return *Error*. Otherwise return $y - e$.

Theorem

The algorithm PGZa corrects up to t errors. □

Remark

The equations in Step 5 are equivalent to the matrix equation

$$\begin{pmatrix} h_{m_1} & h_{m_2} & \dots & h_{m_l} \\ h_{m_1}\eta_1 & h_{m_2}\eta_2 & \dots & h_{m_l}\eta_l \\ h_{m_1}\eta_1^2 & h_{m_2}\eta_2^2 & \dots & h_{m_l}\eta_l^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_{m_1}{\eta_1}^{l-1} & h_{m_2}{\eta_2}^{l-1} & \dots & h_{m_l}{\eta_l}^{l-1} \end{pmatrix} \begin{pmatrix} e_{m_1} \\ e_{m_2} \\ e_{m_3} \\ \vdots \\ e_{m_l} \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{l-1} \end{pmatrix}$$

There is al alternative to step 5 of the PGZa algorithm. Let

- $\sigma(z) = s_0 + s_1 z + \cdots + s_{r-1} z^{r-1}$ (*syndromy polynomial*).
- $\widetilde{L}(z) = 1 + a_1 z + \cdots + a_l z^l$ (its roots are $1/\eta_1, \ldots, 1/\eta_l$).
- $E(z) = \widetilde{L}(z)\sigma(z) \mod z^r$ (*error-evaluator*)

Theorem

For any $m \in \{m_1, \ldots, m_l\}$

$$e_m = -\frac{\alpha_m\, E(1/\alpha_m)}{h_m \widetilde{L}'(1/\alpha_m)}, \tag{13}$$

where $\widetilde{L}'(z)$ denotes the derivative of $\widetilde{L}(z)$.

In this decoding algorithm, the *error-locating polynomial* is defined by

$$E(z) = -\sum_{i=1}^{l} h_{m_i} e_{m_i} \eta_i^r \prod_{j \neq i} (z - \eta_j).$$

It satisfies the Forney's formula

$$e_{m_k} = -\frac{E(\eta_k)}{h_{m_k} \eta_k^r L'(\eta_k}.$$

So we could handle error-location and error-evaluation as soon as we knew how to find $L(z)$ and $E(z)$ from the syndrome.

Notice that $\deg(E(z)) < l$.

Theorem

Let

$$S(z) = s_0 z^{r-1} + \cdots + s_{r-1}$$

Then

$$E(z) \equiv L(z)S(z) \mod z^r.$$

Equivalently, there exists a polynomial $M(z)$ such that

$$E(z) = L(z)S(z) + M(z)z^r.$$

Compute the sequence $r_0 = z^r$, $r_1 = S(z)$, , . . . , r_j of the Euclidean-algorithm remainders until $\deg(r_j) < t$. Let $q_2, \ldots, q_j$ be the corresponding quotients (thus $r_i = r_{i-2} - q_i r_{i-1}$.

Define $v_0 = 0$, $v_1 = 1$, and $v_i = v_{i-2} - q_i v_{i-1}$.

Output $\{E(z), L(z)\} = \{v_j, r_j\}$.

This solves the key equation with $M(z) = u_j$, where $u_0 = 1$, $u_1 = 0$, . . . , $u_i = u_{i-2} - q_i u_{i-1}$ $(i = 2, \ldots, j)$.