Skew Reed-Solomon Codes: Construction and Decoding

José Gómez-Torrecillas, F. J. Lobillo and Gabriel Navarro
CITIC, University of Granada

Skew Reed-Solomon Codes

L denotes a field, $\sigma \in \mathrm{Aut}(L)$ a field automorphism of order n, and L^σ, the invariant subfield. The skew polynomial ring $R = L[x;\sigma]$ consists in polynomials in x with coefficients on the left, the usual sum and multiplication defined by the rule

$$xa = \sigma(a)x \text{ for all } a \in L.$$

The center of R is $Z(R) = L^\sigma[x^n]$, so $x^n - 1$ is a central element and $R(x^n-1)$ is a twosided ideal. Let $\mathcal{R} = R/R(x^n-1)$ and let $\mathfrak{v} : \mathcal{R} \to L^n$ be the canonical isomorphism of vector spaces over L.

Definition. A skew cyclic code over L is $\mathcal{C} = \mathfrak{v}(I)$ where I is a left ideal of $\mathcal{R}$. Since R is principal, each code has the form $\mathcal{C} = \mathfrak{v}(\mathcal{R}g)$ for some $g \in R$ right divisor of $x^n - 1$.

Theorem. *The ring $\mathcal{R}$ is isomorphic to the matrix ring $\mathcal{M}_n(L^\sigma)$. As a consequence, for each $k \leq n$ there exists an SCCC of dimension k. Moreover, each SCCC can be generated as left ideal by an idempotent.*

Let $\alpha \in L$ such that $\{\alpha, \sigma(\alpha), \ldots, \sigma^{n-1}(\alpha)\}$ is a normal basis of the field extension $L^\sigma \subseteq L$. Let $\beta = \alpha^{-1}\sigma(\alpha)$. Then

$$x^n - 1 = \left[x - \beta, x - \sigma(\beta), \ldots, x - \sigma^{n-1}(\beta)\right]_\ell$$

Proposition. *For each $\{i_1, \ldots i_l\} \subseteq \{0, \ldots, n-1\}$, $g = \left[x - \sigma^{i_1}(\beta), \ldots, x - \sigma^{i_l}(\beta)\right]_\ell$ is a right divisor of $x^n - 1$. In particular the skew code $\mathcal{C} = \mathfrak{v}(\mathcal{R}g)$ has dimension $k = n - l$. If there are $\delta - 1$ consecutive indexes in $\{i_1, \ldots i_l\}$, then $\mathrm{d}(\mathcal{C}) \geq \delta$.*

Definition. A skew Reed-Solomon code is an SCCC $\mathcal{C} = \mathfrak{v}(\mathcal{R}g)$ generated by $g = \left[x - \sigma^r(\beta), \ldots, x - \sigma^{r+\delta-2}(\beta)\right]_\ell$. It is an MDS $[n, n-\delta+1, \delta]$–code with respect to the Hamming distance.

Remark. This section has been presented having a general field L as a base field. Hence it can be instanced to linear block codes, setting $L = \mathbb{F}_q$, or to convolutional codes, setting $L = \mathbb{F}_q(t)$. Despite the fact that the Hamming distance is not the best one to measure the correction capability of a convolutional code, the main difference in the two cases is the structure of $\mathrm{Aut}(L)$. Whilst $\mathrm{Aut}(\mathbb{F}_q)$ is a cyclic group, $\mathrm{Aut}(\mathbb{F}_q(t))$ is non commutative.

In order to decode skew Reed-Solomon codes, we fix the following setting. Let $\mathcal{C} = \mathfrak{v}(\mathcal{R}g)$ a skew Reed-Solomon code with $g = \left[x - \beta, \ldots, x - \sigma^{\delta-2}(\beta)\right]_\ell$. There is not loose of generality in assuming $\mathcal{C}$ is narrow sense. Set $\tau = \left\lfloor \frac{\delta-1}{2} \right\rfloor$. Assume $c \in \mathcal{C}$ is transmitted and $y = c + e$ is received, where $e = \mathfrak{v}(e_1 x^{k_1} + \cdots + e_\nu x^{k_\nu})$ and $\nu \leq \tau$.

For each $0 \leq i \leq n-1$, the i-th syndrome s_i of the received word $y = (y_0, \ldots, y_{n-1})$ is defined to be the remainder of the left division of $\sum_{j=0}^{n-1} y_j x^j$ by $x - \sigma^i(\beta)$. Whenever $0 \leq i \leq 2\tau - 1$, the right evaluations on c are zero, and it follows that

$$s_i = \sigma^i(\alpha^{-1}) \sum_{j=1}^{\nu} e_j \sigma^{k_j+i}(\alpha).$$

Therefore $\sigma^i(\alpha)S_i = \sum_{j=1}^{\nu} e_j \sigma^{k_j+i}(\alpha)$ and we call $S = \sum_{i=0}^{2\tau-1} \sigma^i(\alpha) s_i x^i$ the *syndrome polynomial* of y.

PGZ Decoding

The knowledge of the error positions allows the knowledge of the error values because the error values $(e_1, \ldots, e_\nu)$ are the unique solution of the linear system

$$X\,(\Sigma^\nu)^T = (\alpha s_0, \sigma(\alpha)s_1, \cdots, \sigma^{\nu-1}(\alpha)s_{\nu-1}).$$

where

$$\Sigma^\nu = \begin{pmatrix} \sigma^{k_1}(\alpha) & \sigma^{k_2}(\alpha) & \cdots & \sigma^{k_\nu}(\alpha) \\ \sigma^{k_1+1}(\alpha) & \sigma^{k_2+1}(\alpha) & \cdots & \sigma^{k_\nu+1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{k_1+\nu-1}(\alpha) & \sigma^{k_2+\nu-1}(\alpha) & \cdots & \sigma^{k_\nu+\nu-1}(\alpha) \end{pmatrix}_{\nu\times\nu}.$$

For each $1 \leq r \leq \tau$, let

$$S^r = \begin{pmatrix} s_0\alpha & \sigma^{-1}(s_1)\alpha & \ldots & \sigma^{-r+1}(s_{r-1})\alpha \\ s_1\sigma(\alpha) & \sigma^{-1}(s_2)\sigma(\alpha) & \ldots & \sigma^{-r+1}(s_r)\sigma(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ s_\tau\sigma^\tau(\alpha) & \sigma^{-1}(s_{\tau+1})\sigma^\tau(\alpha) & \ldots & \sigma^{-r+1}(s_{\tau+r-1})\sigma^\tau(\alpha) \end{pmatrix}_{(\tau+1)\times r}$$

Algorithm. `PGZ decoding algorithm`

Input: A received transmission $y = (y_0, \ldots, y_{n-1}) \in L^n$ with no more than τ errors.
Output: The error $e = (e_0, \ldots, e_{n-1})$ such that $y - e \in \mathcal{C}$

1: **for** $0 \leq i \leq 2\tau - 1$ **do**
2: $\quad s_i \leftarrow \sum_{j=0}^{n-1} y_j N_j(\sigma^i(\beta))$
3: **if** $s_i = 0$ for all $0 \leq i \leq 2\tau - 1$ **then**
4: $\quad$ **return** 0.
5: $S^\tau \leftarrow \left(\sigma^{-j}(s_{i+j})\sigma^i(\alpha)\right)_{0\leq i\leq\tau, 0\leq j\leq\tau-1}$
6: Compute $\mathrm{rcef}(S^\tau) = \left(\begin{array}{c|c} I_\mu & \\ \hline a_0 \cdots a_{\mu-1} & 0_{(\tau+1)\times(\tau-\mu)} \\ \hline H' & \end{array}\right)$.
7: $\rho = (\rho_0, \ldots, \rho_\mu) \leftarrow (-a_0, \ldots, -a_{\mu-1}, 1)$ and $\rho_N \leftarrow (\rho_0, \ldots, \rho_\mu, 0, \ldots, 0)N$.
8: $\{k_1, \ldots, k_\nu\} \leftarrow$ zero coordinates of ρ_N
9: **if** $\mu \neq \nu$ **then**
10: $\quad M_\rho \leftarrow \begin{pmatrix} \rho_0 & \rho_1 & \ldots & \rho_\mu & 0 & \ldots & 0 \\ 0 & \sigma(\rho_0) & \ldots & \sigma(\rho_{\mu-1}) & \sigma(\rho_\mu) & \ldots & 0 \\ & \ddots & & & \ddots & & \\ 0 & \ldots & 0 & \sigma^{n-\mu-1}(\rho_0) & \ldots & \ldots & \sigma^{n-\mu-1}(\rho_\mu) \end{pmatrix}_{(n-\mu)\times n}$
11: $\quad N_\rho \leftarrow M_\rho N$
12: $\quad H_\rho \leftarrow \mathrm{rref}(N_\rho)$
13: $\quad H'$ gets the matrix obtained removing all rows of H_ρ different from ε_i for any i.
14: $\quad \{k_1, \ldots, k_\nu\} \leftarrow$ zero column coordinates of H'
15: Find $(x_1, \ldots, x_\nu)$ such that $(x_1, \ldots, x_\nu)(\Sigma^\nu)^T = (\alpha s_0, \sigma(\alpha)s_1, \ldots, \sigma^{\mu-1}(\alpha)s_{\mu-1})$
16: **return** $(e_0, \ldots, e_{n-1})$ with $e_i = x_i$ for $i \in \{k_1, \ldots, k_\nu\}$, and zero otherwise.

Sugiyama Decoding

We define the *error locator* polynomial as

$$\lambda = \left[1 - \sigma^{k_1}(\beta)x, 1 - \sigma^{k_2}(\beta)x, \ldots, 1 - \sigma^{k_\nu}(\beta)x\right]_r.$$

Consequently, for any $1 \leq j \leq \nu$, $\lambda = (1 - \sigma^{k_j}(\beta)x)p_j$ for some polynomial $p_j \in R$ with $\deg p_j = \nu - 1$. We define the *error evaluator* polynomial as

$$\omega = \sum_{j=1}^{\nu} e_j \sigma^{k_j}(\alpha) p_j.$$

Proposition. *$1 - \sigma^d(\beta)x$ left divides λ if and only if $x - \sigma^{d-1}(\beta^{-1})$ left divides λ if and only if $d \in \{k_1, \ldots, k_\nu\}$.*

Therefore, once λ is known, the error positions can be located. Moreover, once we also know ω, we may compute the values $e_1, e_2, \ldots, e_\nu$ by an easy interpolation and determine completely the error e.

As for classical cyclic block codes, syndrome, error locator and error evaluator polynomials satisfy a key equation.

Theorem. *The error locator and the error evaluator satisfy the* non-commutative key equation

$$S\lambda + x^{2\tau}u = \omega, \tag{NKE}$$

where $u \in R$ is of degree less than ν. The non-commutative key equation is a right multiple of the equation

$$x^{2\tau}u_l + Sv_l = r_l, \tag{Bez}$$

where u_l, v_l and r_l are the Bezout coefficients returned by the Right Extended Euclidean Algorithm with input $x^{2\tau}$ and S, and l is the index determined by the conditions $\deg r_{l-1} \geq \tau$ and $\deg r_l < \tau$. In particular, $\lambda = v_l g$ and $\omega = r_l g$ for some $g \in R$. Moreover $\deg g = 0$ if and only if $(\omega, \lambda)_r = 1$.

The key equation for classical cyclic block codes can be solved using the Extended Euclidean Algorithm because the error locator and the error evaluator polynomial are coprime. This is no longer true in this new skew setting since

$$(\omega, \lambda)_r = 1 \iff \begin{vmatrix} e_1 & e_2 & \ldots & e_\nu \\ \sigma(e_1) & \sigma(e_2) & \ldots & \sigma(e_\nu) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{\nu-1}(e_1) & \sigma^{\nu-1}(e_2) & \ldots & \sigma^{\nu-1}(e_\nu) \end{vmatrix} \neq 0,$$

which happens with low probability. However, a solution of (NKE) can be lifted from a solution of (Bez) thanks to the following theorem.

Theorem. *Let $q, p, s \in R$ such that $x^{2\tau}q + Sp = s$, $qg = u$, $pg = \lambda$ and $sg = \omega$ for some $g \in R$. Let $T = \{t_1, t_2, \cdots, t_m\} \subset \{0, 1, \ldots, n-1\}$ be the set of indices verifying that $\sigma^{j-1}(\beta^{-1})$ is a left root of p if and only if $j \in T$. Then $m = \deg p$ if and only if g is a constant.*

Bibliography

[1] J. Gómez-Torrecillas, F. J. Lobillo and G. Navarro, A new perspective of cyclicity in convolutional codes, IEEE Trans. Inform. Theory 62 (2016), 2702–2706.
[2] J. Gómez-Torrecillas, F. J. Lobillo and G. Navarro, A Sugiyama-like decoding algorithm for convolutional codes, Submitted. `https://arxiv.org/abs/1607.07187`
[3] J. Gómez-Torrecillas, F. J. Lobillo and G. Navarro, Peterson-Gorenstein-Zierler algorithm for skew BCH codes, Submitted.

Research partially supported by grants MTM2013-41992-P and MTM2016-78364-P from Ministerio de Economía y Competitividad of the Spanish Government and from FEDER.