5. Reed-Solomon Codes

Generalized Reed-Solomon Codes

- Let $\alpha_1, \alpha_2, \ldots, \alpha_n$, $n < q$, be distinct nonzero elements of $\mathbb{F}_q$, and let $v_1, v_2, \ldots, v_n$ be *nonzero* elements of $\mathbb{F}_q$ (not necessarily distinct). A *generalized Reed-Solomon (GRS)* code is a linear $[n, k, d]$ code $\mathcal{C}_{\mathrm{GRS}}$ over $\mathbb{F}_q$, with PCM

$$H_{\mathrm{GRS}} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \ldots & \alpha_n^{n-k-1} \end{pmatrix} \begin{pmatrix} v_1 & & 0 \\ & v_2 & \\ 0 & & \ddots \\ & & & v_n \end{pmatrix}.$$

α_j: *column locators* (distinct), v_j: *column multipliers* ($\neq 0$)

Theorem

$\mathcal{C}_{\mathrm{GRS}}$ is an MDS code, namely, $d = n - k + 1$.

Proof. Any subset of $r = n - k$ distinct columns of the left part of H_{GRS} has the form of a Vandermonde matrix defined by distinct elements, which is nonsingular. Hence, $d \geq n - k + 1$. By Singleton's bound, $d = n - k + 1$. $\square$

$$X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_r \\ x_1^2 & x_2^2 & \cdots & x_r^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{r-1} & x_2^{r-1} & \cdots & x_r^{r-1} \end{bmatrix}$$

$$|X| = \prod_{i<j}(x_j - x_i)$$

About column multipliers

Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, and define

$$M_{n-k}(\boldsymbol{\alpha}) = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \ldots & \alpha_n^{n-k-1} \end{pmatrix}, \; D(\mathbf{v}) = \begin{pmatrix} v_1 & & & 0 \\ & v_2 & & \\ 0 & & \ddots & \\ & & & v_n \end{pmatrix}.$$

- We have $H_{\mathrm{GRS}} = M_{n-k}(\boldsymbol{\alpha})D(\mathbf{v})$. Consider the code $\mathcal{C}'_{\mathrm{GRS}}$ with PCM $H'_{\mathrm{GRS}} = M_{n-k}(\boldsymbol{\alpha})$.
- Clearly, $H_{\mathrm{GRS}}\mathbf{c}^T = 0 \;\Leftrightarrow\; H'_{\mathrm{GRS}}(D(\mathbf{v})\mathbf{c}^T)) = 0$: the codewords of $\mathcal{C}'_{\mathrm{GRS}}$ are the same as the codewords of $\mathcal{C}_{\mathrm{GRS}}$, but with the value in coordinate j multiplied by v_j, $1 \leq j \leq n$.
- $\mathcal{C}'_{\mathrm{GRS}}$ has the same parameters $[n, k, d]$ as $\mathcal{C}_{\mathrm{GRS}}$ (d is preserved since all v_j are nonzero). Column multipliers seem to make no difference (???).
- However, column multipliers do make a *big* difference on the properties of *sub-field sub-codes* of GRS codes. Also, certain choices of multipliers (and locators) have advantages when implementing encoders/decoders.

Duals of GRS codes

Theorem

The dual of a GRS code is a GRS code.

Proof. We show that G_{GRS} can have the form $G = M_k(\boldsymbol{\alpha})D(\mathbf{v}')$ for an appropriate choice of column multipliers v'_j (but same column locators as H). Typical rows of such G, and of H, have the form

$$G_i = [v'_1\alpha_1^i,\, v'_2\alpha_2^i,\, \ldots,\, v'_n\alpha_n^i],\ 0 \le i \le k-1,$$
$$H_j = [v_1\alpha_1^j,\, v_2\alpha_2^j,\, \ldots,\, v_n\alpha_n^j],\ 0 \le j \le n-k-1\,.$$

We have

$$G_i \cdot H_j^T = \sum_{\ell=1}^{n} v_\ell v'_\ell \alpha_\ell^{i+j},\ \ 0 \le i \le k-1,\ 0 \le j \le n-k-1,$$

with $0 \le i+j \le n-2$. Therefore, $GH^T = 0$ if and only if

$$\sum_{\ell=1}^{n} v_\ell v'_\ell \alpha_\ell^t = 0, \quad 0 \le t \le n-2\,.$$

These equations can be written in matrix form as $M_{n-1}(\boldsymbol{\alpha})D(\mathbf{v})(\mathbf{v}')^T = 0$. Now, $M_{n-1}(\boldsymbol{\alpha})D(\mathbf{v})$ is the PCM of an $[n, 1, n]$ GRS code, which has nonzero codewords. Taking $\mathbf{v}'$ to be such a codeword, the equations are satisfied. This codeword has weight n, hence all v'_j are nonzero. □

Distinguished Classes of GRS Codes

- *Primitive GRS codes:* $n = q-1$ and $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} = F^*$; usually $\alpha_i = \alpha^{i-1}$ for a primitive $\alpha \in \mathbb{F}$.
- *Normalized GRS codes:* $v_j = 1$ for all $1 \leq j \leq n$.
- *Narrow-sense GRS codes:* $v_j = \alpha_j$ for all $1 \leq j \leq n$.
- Allowing one $\alpha_i = 0$ (column $[1\,0\,\ldots\,0]^T$, not in narrow sense GRS): *(singly) extended GRS code* $\implies n \leq q$
- Allowing one $\alpha_i = \infty$ (column $[0\,\ldots\,0\,1]^T$, not in narrow sense GRS): *(doubly) extended GRS code* $\implies n \leq q+1$

Example. Let $v_1, v_2, \ldots, v_n$ be the column multipliers of a primitive GRS code. We can verify that the dual GRS code has column multipliers α_j / v_j

$$\implies \quad (\text{normalized primitive GRS})^{\perp} = (\text{narrow-sense primitive GRS}).$$

GRS Encoding as Polynomial Evaluation

- For $\mathbf{u} = (u_0\, u_1\, \ldots\, u_{k-1})$, let $u(x) = u_0 + u_1 x + u_2 x^2 + \cdots + u_{k-1}x^{k-1}$. Then,

$$\mathbf{c} = \mathbf{u}\, G_{\text{GRS}} = (u_0\, u_1 \ldots u_{k-1}) \cdot \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \ldots & \alpha_n^{k-1} \end{pmatrix} \begin{pmatrix} v_1' & & & 0 \\ & v_2' & & \\ 0 & & \ddots & \\ & & & v_n' \end{pmatrix}$$

$$= [\, v_1' u(\alpha_1) \;\; v_2' u(\alpha_2) \;\; \ldots \;\; v_n' u(\alpha_n)\,]$$

- Minimum distance now follows from the fact that a polynomial of degree $\leq k-1$ cannot have more than $k-1$ roots in $\mathbb{F}_q \implies \text{wt}(\mathbf{c}) \geq n-k+1$.
- Decoding as *noisy interpolation*: reconstruct $u(x)$ from $(k+2t)$ noisy evaluations $u(\alpha_1)+e_1, u(\alpha_2)+e_2, \ldots, u(\alpha_{k+2t})+e_{k+2t}$, possible if at most t evaluations are corrupted.

Refresher: shortening a linear code

Given an $[n, k, d]$ code, we can obtain an $[n-\ell, k-\ell, d]$ code, $1 \leq \ell \leq k$, by

1. selecting all the codewords that start with ℓ zeros,
2. deleting the first ℓ coordinates.

If the code is systematic, this can be visualized as follows

$$\mathbf{u}\,G = \mathbf{u}\ \left(I_{k\times k} | A_{k\times(n-k)}\right)$$

$$= [\underbrace{0, 0, \ldots, 0}_{\ell}, u_{k-\ell-1}, \ldots, u_0] \left(\begin{array}{c|c|c} I_\ell & \mathbf{0}_{\ell\times k-\ell} & A^U_{\ell\times(n-k)} \\ \hline \mathbf{0}_{(k-\ell)\times k} & I_{k-\ell} & A^L_{(k-\ell)\times(n-k)} \end{array}\right)$$

Generator matrix of the shortened code

Shortening is equivalent to setting the first ℓ message symbols to zero and then ignoring them.
In terms of the systematic generator matrix, it is equivalent to taking the lower-right $(k-\ell)\times(n-\ell)$ corner of the original matrix.

Conventional Reed-Solomon Codes

- *Conventional Reed-Solomon (RS)* code $\mathcal{C}_{\mathrm{RS}}$: GRS code with $n|(q-1)$, $\alpha \in \mathbb{F}^*$ with $\mathcal{O}(\alpha) = n$,

$$\begin{aligned} \alpha_j &= \alpha^{j-1}, \quad 1 \le j \le n, \\ v_j &= \alpha^{b(j-1)}, \quad 1 \le j \le n, \; b \in \mathbb{Z}. \end{aligned}$$

  - Commonly, $n = q-1$: *primitive code*.
  - Code can be shortened to any length $n' \le n$.
    - Two ways to get shorter codes: choose $n|(q-1), \; n < q-1$, or shorten by setting message digits to zero (or do both).

- *Canonical PCM* of a RS code is given by

$$H_{\mathrm{RS}} = \begin{pmatrix} 1 & \alpha^b & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & \alpha^{b+r-1} & \alpha^{2(b+r-1)} & \cdots & \alpha^{(n-1)(b+r-1)} \end{pmatrix}$$

#rows $= r = n-k = d-1$

Conventional Reed-Solomon Codes

$$H_{\mathrm{RS}} = \begin{pmatrix} 1 & \alpha^{b} & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & \alpha^{b+r-1} & \alpha^{2(b+r-1)} & \cdots & \alpha^{(n-1)(b+r-1)} \end{pmatrix}$$

#rows $= r = n-k = d-1$

- Associate $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}] \in \mathbb{F}^n$ with $c(x) = \sum_{\ell=0}^{n-1} c_i x^i \in \mathbb{F}[x]$.
- $\mathbf{c} \in \mathcal{C}_{\mathrm{RS}} \iff H_{\mathrm{RS}} \mathbf{c}^T = \mathbf{0}$.
- For a typical row $\bar{\mathbf{h}}_i$ of H_{RS}, $\bar{\mathbf{h}}_i \mathbf{c}^T = \sum_{j=0}^{n-1} \left(\alpha^{b+i}\right)^j c_j = c(\alpha^{b+i})$.
  Therefore, $\mathbf{c} \in \mathcal{C}_{\mathrm{RS}} \iff c(\alpha^{\ell}) = 0, \quad \ell = b, b+1, \ldots, b+r-1$.
- $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+r-1}$: *roots* of $\mathcal{C}_{\mathrm{RS}}$.
- $g(x) = (x-\alpha^b)(x-\alpha^{b+1})\cdots(x-\alpha^{b+r-1})$:
  generator polynomial of $\mathcal{C}_{\mathrm{RS}}$.
  $\deg(g) = r = n-k$

RS Codes as Cyclic codes (another polynomial characterization)

- $\mathbf{c} \in \mathcal{C}_{\rm RS} \iff c(\alpha^\ell) = 0, \;\; \ell = b, b+1, \ldots, b+r-1$
- $g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdots (x - \alpha^{b+r-1}) \qquad (\deg(g) = r)$

  Therefore, $\mathbf{c} \in \mathcal{C}_{\rm RS} \iff g(x) | c(x)$ and

  $$\mathcal{C}_{\rm RS} = \{\, u(x)g(x) \;:\; \deg(u) < k \,\} \subseteq \mathbb{F}_q[x]_n$$

  Every root of $g(x)$ is also a root of $x^n - 1 \implies g(x) \,|\, x^n - 1$.
- $\mathcal{C}_{\rm RS}$ is the *ideal* generated by $g(x)$ in the ring $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$.
- RS codes are *cyclic*: $c(x) \in \mathcal{C}_{\rm RS} \implies xc(x) \mod (x^n - 1) \in \mathcal{C}_{\rm RS}$, or

  $$\mathbf{c} = [\, c_0\, c_1 \,\ldots\, c_{n-1} \,] \in \mathcal{C}_{\rm RS} \implies [\, c_{n-1}\, c_0\, c_1 \,\ldots\, c_{n-2} \,] \in \mathcal{C}_{\rm RS}$$

- Distinguished RS codes
  - Primitive RS: $n = q - 1$, α primitive element of $\mathbb{F}_q$
  - Narrow-sense RS: $b = 1$ (*common choice*)
  - Normalized RS: $b = 0$
- Cyclic property is not preserved if we shorten the code, but the other properties are.

Encoding RS codes

- We saw the *polynomial evaluation* interpretation of GRS encoding

$$\mathbf{c} = \mathbf{u}G_{\mathrm{GRS}} = \mathbf{u} \cdot \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_n^{k-1} \end{pmatrix} \begin{pmatrix} v_1' & & & 0 \\ & v_2' & & \\ 0 & & \ddots & \\ & & & v_n' \end{pmatrix}$$

$$= [\, v_1' u(\alpha_1)\; v_2' u(\alpha_2)\; \dots\; v_n' u(\alpha_n) \,]$$ *non-systematic*

- In the *polynomial ideal* interpretation of RS codes: $u(x) \mapsto u(x)g(x)$, corresponds to a non-systematic generator matrix

$$G = \begin{pmatrix} g_0 & g_1 & \dots & g_{n-k} & & & 0 \\ & g_0 & g_1 & \dots & g_{n-k} & & \\ 0 & & \ddots & \ddots & \dots & \ddots & \\ & & & g_0 & g_1 & \dots & g_{n-k} \end{pmatrix} \qquad (g_{n-k} = 1)$$

How about a systematic encoding?

Systematic Encoding of RS Codes

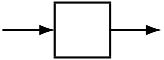
- For $u(x) \in \mathbb{F}_q[x]_k$, let $r_u(x)$ be the unique polynomial in $\mathbb{F}_q[x]_{n-k}$ such that
$$r_u(x) \equiv x^{n-k}u(x) \mod g(x)$$

- Let $c(x) = x^{n-k}u(x) - r_u(x)$.
Clearly, $g(x) \mid c(x)$, and $\deg(c(x)) \leq n-1$, so
$$c(x) \in \mathcal{C}_{\mathrm{RS}}$$

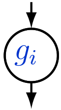
- The mapping $\mathcal{E}_{\mathrm{RS}} : u(x) \mapsto c(x) = x^{n-k}u(x) - r_u(x)$ is a *linear, systematic* encoding for $\mathcal{C}_{\mathrm{RS}}$

$$\begin{array}{rcccccccl}
[& u_{k-1} & u_{k-2} & \ldots & u_0 & 0 & 0 & \ldots & 0 \;] \\
-[& 0 & 0 & \ldots & 0 & r_{n-k-1} & r_{n-k-2} & \ldots & r_0 \;] \\
\hline
[& c_{n-1} & c_{n-2} & \ldots & c_{n-k} & c_{n-k-1} & c_{n-k-2} & \ldots & c_0 \;] \\
 & \longleftarrow & & k & \longrightarrow & \longleftarrow & n-k & & \longrightarrow
\end{array}$$

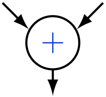
Circuit elements for a systematic encoder



1 clock cycle
delay unit

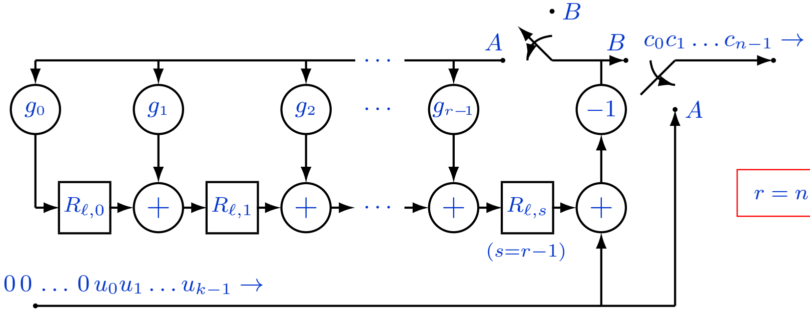


multiply
by g_i



add

Systematic Encoding Circuit



$\boxed{r = n - k}$

Switches:

- at A for k cycles
- at B for $r=n-k$ cycles

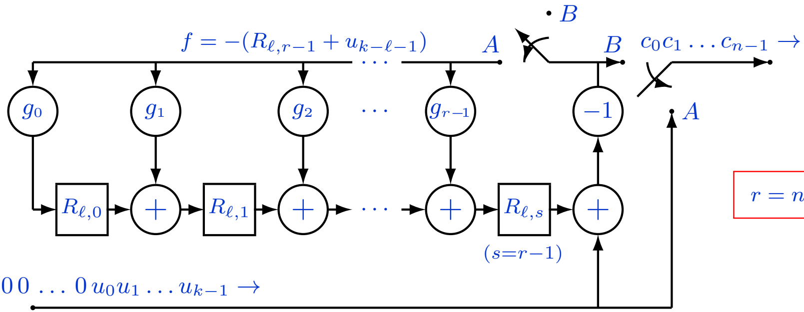
Register contents:

$$R_\ell(x) = \sum_{i=0}^{r-1} R_{\ell,i} x^i, \quad 1 \le \ell \le k,$$

with initial condition

$$R_0(x) = 0$$

Systematic Encoding Circuit



$r = n - k$

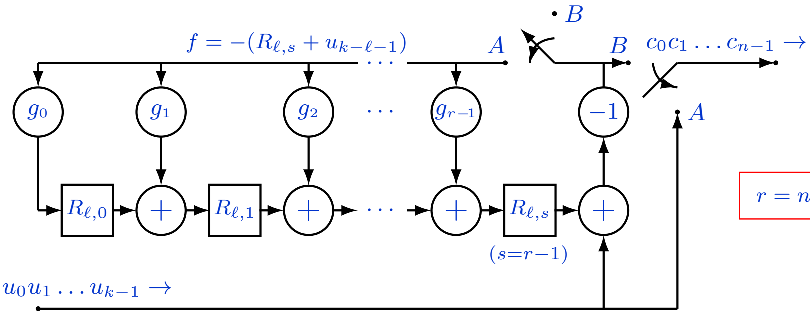
$$g(x) = x^r + g_{r-1}x^{r-1} + g_{r-2}x^{r-2} + \cdots + g_1 x + g_0 \triangleq x^r + \bar{g}(x)$$

Notice: $\bar{g}(x) \equiv -x^r \bmod g(x)$.

One step while switches are at A:

$$\begin{aligned} R_{\ell+1}(x) &= xR_\ell(x) - R_{\ell,r-1}x^r + \bar{g}(x)f \\ &= xR_\ell(x) \underbrace{-R_{\ell,r-1}x^r - \bar{g}(x)R_{\ell,r-1}}_{-R_{\ell,r-1}g(x)} \underbrace{-\bar{g}(x)u_{k-\ell-1}}_{x^r u_{k-\ell-1}} \\ &\equiv \left(xR_\ell(x) + x^r u_{k-\ell-1} \right) \bmod g(x) \end{aligned}$$

Systematic Encoding Circuit



$\boxed{r = n - k}$

Switches:

- at A for k cycles
- at B for $r{=}n{-}k$ cycles

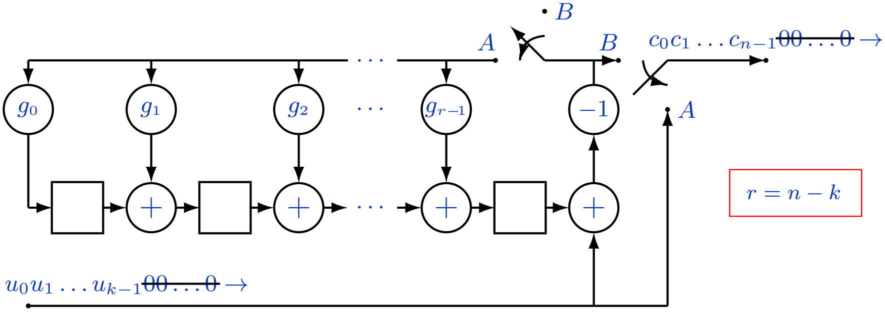
Register contents: $R_0(x) = 0$

$$\begin{aligned} R_{\ell+1}(x) &= xR_\ell(x) + x^r u_{k-\ell-1} \\ &= x^2 R_{\ell-1}(x) + x^r \left(x u_{k-\ell} + u_{k-\ell-1}\right) \\ &= x^r \sum_{i=1}^{\ell+1} u_{k-i} x^{\ell+1-i} \mod g(x) \end{aligned}$$

$$\ell = 0, 1, \ldots, k-1 \, .$$

$$\boxed{R_k(x) = x^r \sum_{i=1}^{k} u_{k-i} x^{k-i} \bmod g(x) = x^r u(x) \bmod g(x) \, .}$$

Shortened RS codes: Encoding Circuit



$\boxed{r = n - k}$

The "conceptual" zeros are never stored or manipulated. They do not participate in any computation.

Constant multipliers

Assume $q = 2^m$. Multiplying by a constant $g_i \in \mathrm{GF}(2^m)$ is a linear transformation over $\mathrm{GF}(2)$.

- If elements are represented as m-vectors over $\mathrm{GF}(2)$, the transformation can be implemented via multiplication by an $m \times m$ matrix with entries in $\mathrm{GF}(2)$, i.e., computing m XOR sums, each over a subset of the m input bits.

  Example: Multiply generic $\beta : [\beta_0\, \beta_1\, \beta_2\, \beta_3]$ by α^8 in $\mathrm{GF}(2^4)$.

$$\alpha^8 \beta \quad \longleftrightarrow \quad \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_2 \\ \beta_1 + \beta_2 + \beta_3 \\ \beta_0 + \beta_2 + \beta_3 \\ \beta_1 + \beta_3 \end{bmatrix}$$

- We have r such multipliers in the encoder, all sharing the same input. If we have $g_i = g_j$ for some $i \neq j$, the output from the g_i multiplier can be re-used, and fed to the adder in the j-th stage of the register (eliminating the g_j multiplier). This would save hardware resources.

Palindromic generator polynomial

$$g(x) = x^r + g_{r-1}x^{r-1} + \cdots + g_1 x + g_0,$$

with $g_0 \neq 0$. Reversed:

$$\overleftarrow{g}(x) = g_0 x^r + g_1 x^{r-1} + \cdots + g_{r-1}x + 1$$

We have $\overleftarrow{g}(x) = x^r g(x^{-1})$, so, β is a root of $g(x)$ iff β^{-1} is a root of $\overleftarrow{g}(x)$.
Can we make $g(x) = \overleftarrow{g}(x)$ (*palindromic*)? This would make $g_0 = 1$, $g_1 = g_{r-1}$, $g_2 = g_{r-2}$, ...

Yes, if the set of roots is closed under inversion. Assume $q = 2^m$.

If r is even, choose $b = \frac{q}{2} - \frac{r}{2}$.

If r is odd, choose $b = -\frac{r-1}{2}$

(equivalently, $b = q - 1 - \frac{r-1}{2}$).