2016. október 7., péntek 14:25

Binary codes to "q"-ary codes because of correcting any given number of errors
⇒ Wireless communication needs it ($W \leq N$)

$GF(q)$ where "q" must be prime → mod q

Reed-Solomon (RS) codes: $C(n,k)$ over $GF(q)$; $n = q-1$; $\alpha_0, \alpha_1, \dots \alpha_{n-1} \in GF(q)$: $\alpha_i \neq 0 \ \forall i = 0 \dots n-1$

$\underline{u} = (u_0, u_1, \dots u_{k-1}) \xrightarrow{X} u(x) = u_0 + u_1 x + u_2 x^2 + \dots u_{k-1} x^{k-1}$ → $\deg(u(x)) = k-1$

$$\left.\begin{aligned} c_0 &= u(x)\big|_{x=\alpha_0} = u_0 + u_1\alpha_1 + u_2\alpha_1^2 + \dots u_{k-1}\alpha_1^{k-1} \\ c_1 &= u(x)\big|_{x=\alpha_1} = u_0 + u_1\alpha_2 + u_2\alpha_2^2 + \dots u_{k-1}\alpha_2^{k-1} \\ &\vdots \\ c_{n-1} &= u(x)\big|_{x=\alpha_{n-1}} = u_0 + u_1\alpha_{n-1} + u_2\alpha_{n-1}^2 + \dots u_{k-1}\alpha_{n-1}^{k-1} \end{aligned}\right\} \quad (c_0, c_1, \dots c_{n-1}) = (u_0, u_1, \dots u_{k-1})\, \underline{\underline{G}}_{k\times n}$$

$$\underline{\underline{G}}_{k\times n} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_0 & \alpha_1 & \dots & \alpha_{n-1} \\ \alpha_0^2 & \alpha_1^2 & \dots & \alpha_{n-1}^2 \\ \vdots & \vdots & & \vdots \\ \alpha_0^{k-1} & \alpha_1^{k-1} & \dots & \alpha_{n-1}^{k-1} \end{pmatrix}$$

⇒ It is an MDS code!

$d_{min} = w_{min} = n - \#\text{ of zero components} \geq n - (k-1) = n - k + 1$

$d_{min} \leq n - k + 1$

$d_{min} = n - k + 1$

Standard implementation: $\alpha \in GF(q)$ $\quad \alpha_0 = \alpha^0 = 1, \alpha_1 = \alpha^1 = \alpha, \alpha_2 = \alpha^2, \dots \alpha_{n-1} = \alpha^{n-1}$

$ord(\alpha) = q-1$

$$\underline{\underline{G}}_{k\times n} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{k-1} & \alpha^{2(k-1)} & \dots & \alpha^{(k-1)(n-1)} \end{pmatrix}$$

optimal code: $d_{min} = n - k + 1$

Example: Correcting every double errors

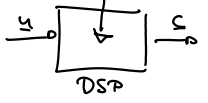
$t = 2$

q is prime; $n = q-1$

$t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor = 2 = \left\lfloor \frac{n-k+1-1}{2} \right\rfloor$ ⇒ $n - k = 2t = 4$

q	n	k	
1	0	–	invalid codes
2	1	–	
3	2	–	
5	4	–	
7	6	2	← we pick that code because it has less noise. But that code means a higher loss on datarate at the same time
11	10	6	
13	12	8	

$$\underline{\underline{G}}_{2\times 6} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 6 & 4 & 5 \end{pmatrix}$$



$\underline{\underline{H}}\underline{c}^T = \underline{0}^T \quad \forall \underline{c} \in C$

$\underline{c} = (c_0, c_1, \dots c_{n-1}) \xrightarrow{X} c(x) = c_0 + c_1 x + c_2 x^2 + \dots c_{n-1} x^{n-1}$

$$\Downarrow \quad c(x)\big|_{x=\alpha^i} = 0 \quad : i = 1 \dots n-k$$

$$c(\alpha) = c_0 + c_1\alpha + c_2\alpha^2 + \dots c_{n-1}\alpha^{n-1} = 0$$
$$c(\alpha^2) = c_0 + c_1\alpha^2 + c_2\alpha^4 + \dots c_{n-1}\alpha^{(n-1)\cdot 2} = 0$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$c(\alpha^{n-k}) = c_0 + c_1\alpha^{n-k} + c_2\alpha^{2(n-k)} + \dots c_{n-1}\alpha^{(n-1)(n-k)}$$

$$\underline{\underline{H}}_{(n-k)\times n} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-k} & \alpha^{2(n-k)} & \dots & \alpha^{(k-1)(n-k)} \end{pmatrix}$$

$$\underline{\underline{G}}_{k\times n} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{k-1} & \alpha^{2(k-1)} & \dots & \alpha^{(k-1)(n-1)} \end{pmatrix}$$

$$\underline{\underline{G}}^T \underline{\underline{H}} = \underline{\underline{0}}$$

$$\left(\underline{\underline{H}}\,\underline{c}^T\right)_i = \sum_{i=1}^{n-1} \alpha^{li} c_i = \sum_{i=0}^{n-1} \alpha^{li} \sum_{j=0}^{k-1} \alpha^{ij} u_j = \sum_{j=0}^{k-1} u_j \underbrace{\sum_{i=0}^{n-1} \alpha^{i(j+l)}}_{\text{quotient} = \alpha^{j+1}} = \sum_{j=0}^{k-1} u_j \frac{\overbrace{\alpha^{n(j+l)}}^{=1} - 1}{\alpha^{j+l}+1} = 0$$

$$n = q-1 \qquad \alpha^{n(j+l)} = (\alpha^n)^{j+l} = (\alpha^{q-1})^{j+l} = 1^{j+l} = 1$$

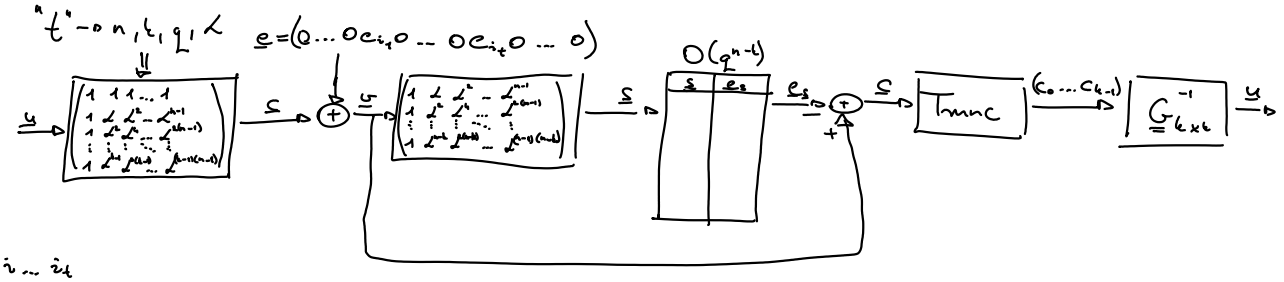
$$d_{min} = n-k+1$$
$$t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor$$

$t \to 2t$ # of linearly independent columns in $\underline{\underline{H}}$

? ?

$$\underline{\underline{H}}_{(n-k)\times n} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-k} & \alpha^{2(n-k)} & \dots & \alpha^{(n-k)(n-1)} \end{pmatrix}$$

"t" - n, k, q, α $\quad \underline{e} = (0 \dots 0 c_{i_1} 0 \dots 0 c_{i_t} 0 \dots 0)$



$i \dots i_t$

$e_{i_j} \in \{0, 1, \dots q-1\}$

$$\underbrace{(u_0 \dots u_{k-1})}_{?} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \alpha^{2(k-1)} & \dots & \alpha^{(k-1)(n-1)} \end{pmatrix} - \underbrace{(c_0 \dots c_{n-1})}_{\text{known}}$$

known, $k \times k$

# of errors $= t <$ # of equations

$\Downarrow$

overdetermined

$\Downarrow$

we drop a few $\to k \times k$

$$(u_0 \dots u_{k-1}) \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \alpha^{2(k-1)} & \dots & \alpha^{(k-1)(n-1)} \end{pmatrix} = (c_0 \dots c_{k-1})$$

$\Rightarrow$ we can correct

given # of errors

=

Optimal!

MDS

det ≠ 0

↓

invert