Decodificar R-S.

Decodificar R-S.

Sea
$$w = c + e$$
la palabra recibida, con $wt(e) \leq t \underset{\wedge}{=} \left[\frac{n-k}{2}\right]$.

Decodificar R-S.

Sea

$$w = c + e$$

la palabra recibida, con $wt(e) \leq t = \left[\frac{n-k}{2}\right]$.

sabemos $d = n-k+1$

$k = -d + n + 1$

es decir: C es e-corrector

$e = \left[\frac{d-1}{2}\right]$

Decodificar R-S.

Sea

$$w = c + e$$

la palabra recibida, con $wt(e) \leq t \underset{\uparrow}{=} \left[\frac{n-k}{2}\right]$.

sabemos $d = n-k+1$

$k = -d + n + 1$

es decir: C es e-corrector

$e = \left[\frac{d-1}{2}\right]$

Buscamos un polinomio

$$Q(x,y) = Q_0(x) + y\,Q_1(x) \in \mathbb{F}_q[x,y] \setminus \{0\},$$

Decodificar R-S.

Sea
$$w = c + e$$
la palabra recibida, con $wt(e) \leq t = \left[\frac{n-k}{2}\right]$.

sabemos $d = n-k+1$
$k = -d+n+1$
es decir: C es e-corrector
$e = \left[\frac{d-1}{2}\right]$

Buscamos un polinomio
$$Q(x,y) = Q_0(x) + y\,Q_1(x) \in \mathbb{F}_q[x,y] \setminus \{0\},$$
t.q.:

1) $Q(x_i, w_i) = 0, \quad i = 1, \dots, n \quad (w_i = f(x_i) + e_i)$

2) $\deg Q_0 \leq n-1-t,$

3) $\deg Q_1 \leq n-1-t-(k-1).$

Decodificar R-S.

Sea
$$w = c + e$$
la palabra recibida, con $wt(e) \leq t = \left[\frac{n-k}{2}\right]$.

sabemos $d = n-k+1$

$k = -d+n+1$

es decir: C es e-corrector

$e = \left[\frac{d-1}{2}\right]$

Buscamos un polinomio
$$Q(x,y) = Q_0(x) + y\,Q_1(x) \in \mathbb{F}_q[x,y] \setminus \{0\},$$
t.q.:

1) $Q(x_i, w_i) = 0, \quad i = 1, \dots, n \quad (w_i = \cancel{x_i}\, f(x_i) + e_i)$

2) $\deg Q_0 \leq n-1-t$,

3) $\deg Q_1 \leq n-1-t-(k-1)$.

Esto es posible por que 1) son n ecuaciones lineales homogéneas. Siendo
$$n-1-t+1+n-1-t-(k-1)+1 \geq n+1$$
existe una solución.

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\dfrac{Q_0(x)}{Q_1(x)}$.

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

$Q(x,y)$ cálculado en $(x_i, f(x_i) + e_i)$ es 0.

$Q(x_i, f(x_i) + e_i) = 0$.

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

$Q(x, y)$ calculado en $(x_i, f(x_i) + e_i)$ es 0.

$Q(x_i, f(x_i) + e_i) = 0$.

Siendo que $e_i = 0$ para por lo menos $n-t$ i's, tenemos que $Q(x, f(x))$ tiene $n-t$ ceros o más (cuando $c_i = w_i$).

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\dfrac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

$Q(x, y)$ calculado en $(x_i; f(x_i) + e_i)$ es 0.

$Q(x_i, f(x_i) + e_i) = 0$.

Siendo que $e_i = 0$ para por lo menos $n-t$ i's, tenemos que $Q(x, f(x))$ tiene $n-t$ ceros o más (cuando $c_i = w_i$).

Ahora $\deg Q(x, f(x)) \leq n-t-1 \implies Q(x, f(x)) = 0$.

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

$Q(x,y)$ cálculado en $(x_i; f(x_i) + e_i)$ es 0.

$Q(x_i, f(x_i) + e_i) = 0$.

Siendo que $e_i = 0$ para por lo menos $n-t$ i's, tenemos que $Q(x, f(x))$ tiene $n-t$ ceros o más (cuando $c_i = w_i$).

Ahora $\deg Q(x, f(x)) \leq n-t-1 \implies Q(x, f(x)) = 0$.

Es decir $Q_0(x) + f(x) Q_1(x) = 0$, cómo queríamos $\square$

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

$Q(x,y)$ cálculado en $(x_i, f(x_i)+e_i)$ es 0.

$Q(x_i, f(x_i)+e_i) = 0$.

Siendo que $e_i = 0$ para por lo menos $n-t$ i's, tenemos que $Q(x, f(x))$ tiene $n-t$ ceros o más (cuando $c_i = w_i$).

Ahora $\deg Q(x, f(x)) \leq n-t-1 \implies Q(x, f(x)) = 0$.

Es decir $Q_0(x) + f(x)Q_1(x) = 0$, cómo queríamos $\square$

Por semplicidad, llamamos

$$\ell_0 = n-1-t, \quad \ell_1 = n-1-(k-1)-t.$$

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c+e$, $wt(e) \leq t$.

$Q(x,y)$ cálculado en $(x_i, f(x_i)+e_i)$ es 0.

$Q(x_i, f(x_i)+e_i) = 0$.

Siendo que $e_i = 0$ para por lo menos $n-t$ i's, tenemos que $Q(x, f(x))$ tiene $n-t$ ceros o más (cuando $c_i = w_i$).

Ahora $\deg Q(x, f(x)) \leq n-t-1 \implies Q(x, f(x)) = 0$.

Es decir $Q_0(x) + f(x) Q_1(x) = 0$, cómo queríamos $\square$

Por semplicidad, llamamos

$$l_0 = n-1-t, \; l_1 = n-1-(k-1)-t.$$

Siendo $Q(x,y) = Q_0(x) + y Q_1(x) = Q_1(x)\left(y + \frac{Q_0(x)}{Q_1(x)}\right)$

$$= Q_1(x)\,(y - f(x))$$

los x_i's donde hay errores son ceros de $Q_1(x)$, que por eso se llama polinomio localizador de errores.

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

$Q(x, y)$ cálculado en $(x_i, f(x_i) + e_i)$ es 0.

$Q(x_i, f(x_i) + e_i) = 0$.

Siendo que $e_i = 0$ para por lo menos $n-t$ i's, tenemos que $Q(x, f(x))$ tiene $n-t$ ceros o más (cuando $c_i = w_i$).

Ahora $\deg Q(x, f(x)) \leq n-t-1 \implies Q(x, f(x)) = 0$.

Es decir $Q_0(x) + f(x) Q_1(x) = 0$, cómo queríamos $\square$

Por semplicidad, llamamos

$$l_0 = n-1-t, \quad l_1 = n-1-(k-1)-t.$$

Siendo $Q(x,y) = Q_0(x) + y Q_1(x) = Q_1(x)\left(y + \frac{Q_0(x)}{Q_1(x)}\right)$

$$= Q_1(x)\,(y - f(x))$$

los x_i's donde hay errores son ceros de $Q_1(x)$, que por eso se llama polinomio localizador de errores.

Algoritmo

INPUT: $w = w_1 \cdots w_n$.

Teorema: Si la palabra código enviada es generada por $f(x)$ y hay menos de $\frac{d}{2}$ errores, $f(x) = -\frac{Q_0(x)}{Q_1(x)}$.

Dem: Sea $c = f(x)$ y $w = c + e$, $wt(e) \leq t$.

$Q(x,y)$ cálculado en $(x_i, f(x_i) + e_i)$ es 0.

$Q(x_i, f(x_i) + e_i) = 0$.

Siendo que $e_i = 0$ para por lo menos $n-t$ i's, tenemos que $Q(x, f(x))$ tiene $n-t$ ceros o más (cuando $c_i = w_i$).

Ahora $\deg Q(x, f(x)) \leq n-t-1 \implies Q(x, f(x)) = 0$.

Es decir $Q_0(x) + f(x) Q_1(x) = 0$, cómo queríamos $\square$

Por semplicidad, llamamos

$$l_0 = n-1-t, \; l_1 = n-1-(k-1)-t.$$

Siendo $Q(x,y) = Q_0(x) + y Q_1(x) = Q_1(x) \left(y + \frac{Q_0(x)}{Q_1(x)} \right)$

$$= Q_1(x) \left(y - f(x) \right)$$

los x_i's donde hay errores son ceros de $Q_1(x)$, que por eso se llama polinomio localizador de errores.

Algoritmo

INPUT: $w = w_1 \cdots w_n$.

① Solucionamos

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{l_0} & w_1 & w_1 x_1 & \cdots & w_1 x_1^{l_1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{l_0} & w_2 & w_2 x_2 & \cdots & w_2 x_2^{l_1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{l_0} & w_n & w_n x_n & \cdots & w_n x_n^{l_1} \end{pmatrix} \begin{pmatrix} Q_{0,0} \\ Q_{0,1} \\ \vdots \\ Q_{0,l_0} \\ Q_{1,0} \\ Q_{1,1} \\ \vdots \\ Q_{1,l_1} \end{pmatrix} = 0$$

② Sean $Q_0(x) = \sum_{j=0}^{\ell_0} Q_{0,j}\, x^j$, $\quad Q_1(x) = \sum_{j=0}^{\ell_1} Q_{1,j}\, x^j$, $\quad f(x) = -\dfrac{Q_0(x)}{Q_1(x)}$

② Sean $Q_0(x)=\sum_{j=0}^{\ell_0} Q_{0,j}\, x^j$, $\quad Q_1(x)=\sum_{j=0}^{\ell_1} Q_{1,j}\, x^j$, $\quad f(x) = -\dfrac{Q_0(x)}{Q_1(x)}$

③ Si $f(x) \in \mathbb{F}_q[x]$ OUTPUT $c = f(x_1) \cdots f(x_n)$

② Sean $Q_0(x)=\sum_{j=0}^{l_0} Q_{0,j}\, x^j, \quad Q_1(x)=\sum_{j=0}^{l_1} Q_{1,j}\, x^j, \quad f(x)=-\dfrac{Q_0(x)}{Q_1(x)}$

③ Si $f(x)\in \mathbb{F}_q[x]$ OUTPUT $c=f(x_1)\cdots f(x_n)$

si no

OUTPUT error.