Reed-Solomon

Práctica 3

Código -at- GitHub

Codificación de $RS(255, 223)$ usando el polinomio:

$$g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{32}) \in GF(256)[x]$$

$RS(255, 223)$ es un $[n = 255, k = 223, d = 33]_{256}-$ código sobre $GF(256)$ con α un elemento primitivo.

Obs. $deg(g) = 32 = n - k = d - 1 = 2t = 2\lfloor \frac{d-1}{2} \rfloor$, i.e. corrige hasta $t = 16$ errores. Además $g(x)$ es un factor de $x^n - 1$ por lo que se trata de un código cíclico.

Campo GF(256)

Formalmente,

$$GF(256) = \mathrm{F}_{2^8} = \mathrm{Z}_2[x]/ < f(x) > \text{ donde } deg(f) = 8$$

El campo puede ser generado con los siguientes polinomios (entre otros):

- **AES** $x^8 + x^4 + x^3 + x + 1$ con $\alpha = x + 1$ como raíz primitiva
- **Primitivo** $x^8 + x^4 + x^3 + x^2 + 1$ con $\alpha = x$ como raíz primitiva

Se representa un elemento en $GF(256)$ con un byte:

$$b_7b_6 \dots b_0 \mapsto b(x) = b_7x^7 + b_6x^6 + \dots + b_1x + b_0$$

1 Codificación

Sea $m = (m_0, m_1, \ldots, m_{k-1}) \in F_{256}^k$ el mensaje a codificar y $m(x)$ el polinomio asociado:

$$m(x) = m_{k-1}x^{k-1} + \ldots + m_1x + m_0$$

Para codificar usando el polinomio $g(x)$

1. $s(x) = m(x)x^{32} = q(x)g(x) + p(x)$ donde $deg(s) < k - 1 + n - k = 254$ y $deg(p) < 32$

$$s(x) = m_{222}x^{254} + \ldots + m_1x^{33} + m_0x^{32}$$
$$p(x) = p_{31}x^{31} + \ldots + p_1x + p_0$$

2. La palabra codificada $c(x) = s(x) - p(x) = q(x)g(x)$:

$$c(x) = m_{222}x^{254} + \ldots + m_1x^{33} + m_0x^{32} + p_31x^{31} + \ldots + p_1x + p_0$$
$$c = (p_0, p_1, \ldots, p_{31}, m_0, m_1, \ldots, m_{222}) \in F_{256}^n$$

   El polinomio codificado $c(x)$ contiene al mensaje $m(x)$ (Codificación Sistemática) y $2t$ entradas de verificación.

Ya que $c(x)|g(x)$ ent $g(\alpha^i) = 0$ syss $c(\alpha^i) = 0$, donde $i = 1..2t$. Se tiene entonces:

$$c(x) \in RS(255, 223) \Leftrightarrow c(\alpha^i) = 0, \forall i = 1..2t$$

2 Decodificación

Sea $r(x) = c(x) + e(x)$ el mensaje recibido, donde $e(x) = e_{n-1}x^{n-1} + \ldots + e_1x + e_0$ es el polinomio de errores en $r(x)$, i.e. e_i es el error en la posición i.

Obs. Si $\omega(e) > t$ entonces la capacidad de errores es excedida y no se puede decodificar $r(x)$.

2.1 Síndromes

Definimos s_i para $i = 1...2t$ como el síndrome de la posición i:

$$r(x) = q_i(x)(x - \alpha^i) + s_i(x) \Leftrightarrow s_i(x) = q_i(x)(x - \alpha^i) + r(x)$$

evaluando en α^i :

$$\begin{aligned} s_i(\alpha^i) &= q_i(x)(\alpha^i - \alpha^i) + r(\alpha^i) \\ &= r_{n-1}(\alpha^i)^{n-1} + \ldots + r_1\alpha^i + r_0 \end{aligned}$$

i.e. basta evaluar las potencias de α en $r(x)$ para encontrar los síndromes.

Por otro lado,

$$\begin{aligned} r(\alpha^i) &= c(\alpha^i) + e(\alpha^i) \\ &= e(\alpha^i) = s_i \end{aligned}$$

por lo que si todos los síndromes son cero entonces no hay errores y $r(x) = c(x)$.

2.2 Localización de errores

Asumiendo que $v \leq t$ errores ocurren, podemos escribir:

$$e(x) = Y_1 x^{e_1} + \ldots + Y_v x^{e_v}$$

donde $Y_1, \ldots, Y_v$ son los valores[1] de los errores y $e_1, \ldots, e_v$ son las posiciones.

Sabemos que $s_i = e(\alpha_i)$, por lo que se tiene:

$$\begin{aligned} s_i &= Y_1(\alpha^i)^{e_1} + \ldots + Y_v(\alpha^i)^{e_v} \\ &= Y_1 X_1^i + \ldots + Y_v X_v^i \end{aligned}$$

A los términos $X_j = \alpha^{e_j}$ se les denomina localizadores de errores.

[1] Por ser un código binario si hay un error e_i entonces $Y_{e_i} = 1$

Ya que las potencias de α son distintas, si se conocen los localizadores X_j se saben las potencias y por ende las posiciones de los errores. Para ello se define $S(x) = s_1 + s_2 x + \ldots + s_{32} x^{31}$ y $\sigma(x)$ (localizador de errores):

$$\begin{aligned}\sigma(x) &= \prod_{i=1}^{v}(1 - X_i x) \\ &= \sum_{i=0}^{v} \sigma_i x^i = 1 + \Lambda_1 x + \ldots + \Lambda_v x^v\end{aligned}$$

Obs. Las raíces de $\sigma(x)$ son los inversos de los localizadores X_j.

Sea $\omega(x)$[2] tal que $deg(\omega) < t$ se obtiene la ecuación:

$$\omega(x) \equiv \sigma(x)S(x) \mod x^{32}$$

$$\Leftrightarrow \omega(x) = \theta(x)x^{32} + \sigma(x)S(x)$$

usando el algoritmo de Euclides para x^{32} y $S(x)$ se obtienen residuos de la forma:

$$r_k(x) = a_k(x)x^{32} + b_k(x)S(x)$$

cuando $deg(r_k) < 16$ pero $deg(r_{k-1}) \geq 16$ y $deg(b_k) \leq 16$ se cumple:

$$\sigma(x) = b_k(0)^{-1} b_k(x)$$

de donde se obtienen los localizadores de errores $X_1 = \alpha^{e_1}, \ldots, X_v = \alpha^{e_v}$.

Entonces, la decodificación resulta:

$$\begin{aligned}r(x) &= c(x) + e(x) \\ c(x) &= r(x) + e(x) \\ &= r(x) + x^{e_1} + \ldots + x^{e_v}\end{aligned}$$

[2]Polinomio evaluador de errores, se usa para obtener Y_i pero en este caso no es necesario