UNIVERSIDAD
DE GRANADA

A SUGIYAMA LIKE DECODING ALGORITHM

GUILLERMO GALINDO ORTUÑO

Trabajo Fin de Grado

Doble Grado en Ingeniería Informática y Matemáticas

Tutores

José Gomez Torrecillas

Francisco Javier Lobillo Borrero

FACULTAD DE CIENCIAS

E.T.S. INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, a 21 de junio de 2020

ÍNDICE GENERAL

RESUMEN

[Tur36]

1

CONCEPTOS BÁSICOS SOBRE CÓDIGOS LINEALES

En este capítulo presentaremos los conceptos básicos sobre códigos lineales, para lo cual nos hemos basado casi por completo en el primer capítulo de [HP03].

1.1 CÓDIGOS LINEALES, MATRICES GENERADORA Y DE PARIDAD

Sea $\mathbb{F}_q^n$ el espacio vectorial de todas las n-tuplas sobre el cuerpo finito $\mathbb{F}_q$.

Definición 1 (Código lineal)**.** Si $\mathcal{C}$ es un subespacio vectorial de dimensión k de $\mathbb{F}_q^n$, diremos que $\mathcal{C}$ es un $[n, k]$ *código lineal* sobre $\mathbb{F}_q$.

Normalmente escribiremos los vectores $(a_1, a_2, \ldots, a_n)$ en $\mathbb{F}_q^n$ de la forma $a_1 a_2 \cdots a_n$ y llamaremos a los vectores en $\mathcal{C}$ *palabras código*, o simplemente *palabras*. Además, utilizaremos nombres concretos para referirnos a códigos sobre algunos de los cuerpos más comunes . A los códigos sobre $\mathbb{F}_2$ los llamaremos *códigos binarios*, los códigos sobre $\mathbb{F}_3$ los notaremos como *códigos ternarios* y a los códigos sobre F_4 los llamaremos *códigos cuaternarios*.

Dicho esto, las dos maneras más comunes de presentar un código lineal son dando una matriz generadora o una matriz de paridad.

Definición 2 (Matriz generadora)**.** Una *matriz generadora* de un $[n, k]$ código lineal $\mathcal{C}$ es cualquier matriz G de dimensiones $k \times n$ cuyas filas formen una base de $\mathcal{C}$.

Dado una matriz generadora G, para cualquier conjunto de k columnas independientes de esta, diremos que el correspondiente conjunto de coordenadas es un *conjunto de información* de C. Las restantes $r = n - k$ coordenadas las notaremos como *conjunto de redundancia*, y llamaremos a r *redundancia* de $\mathcal{C}$. Si las primeras k coordenadas forman un conjunto de información, existe una única matriz generadora para el código de la forma $[I_k|A]$ donde I_k es la matriz identidad de orden k. Diremos que una matriz generadora así está en *forma estándar*.

Definición 3. Una *matriz de paridad* de un $[n,k]$ código lineal $\mathcal{C}$ es cualquier matriz H de dimensiones $(n-k)\times n$ tal que

$$\mathcal{C} = \{x \in \mathbb{F}_q^n | Hx^T = 0\}.$$

Como un código lineal es un subespacio de un espacio vectorial, es el núcleo de alguna aplicación lineal, y por tanto, para un código lineal siempre existe alguna matriz de paridad H. Mencionemos que las filas de H son también independientes. Esto es porque, al ser H una aplicación lineal de $\mathbb{F}_q^n$ en $\mathbb{F}_q^{n-k}$, y la dimensión del núcleo de dicha aplicación es k, tenemos que la dimensión de la imagen es $n-k$ y por tanto el rango de H también.

1.2 PESOS Y DISTANCIAS

La característica que distingue un código lineal de un mero subespacio vectorial es la distancia. En realidad, un código lineal debería definirse como un subespacio vectorial de un espacio vectorial dotado de una distancia. Aunque no las tratemos aquí, hay otras distancias, como la del rango, que se usan. De esta manera, el mismo subespacio vectorial puede considerarse como dos códigos distintos.

Definición 4. Dados dos vectores $x, y \in F_q^n$, definimos la distancia *Hamming* entre ellos $d(x,y)$ como el numero de coordenadas en las que x e y difieren.

Veamos que en efecto esta es una distancia:

Proposición 1. *La función distancia $d(x,y)$ satisface las siguientes condiciones:*

(I) $d(x,y) \geq 0$ *para todo* $x, y \in \mathbb{F}_q^n$.
(II) $d(x,y) = 0$ *si y solo si* $x = y$.
(III) $d(x,y) = d(y,x)$ *for all* $x, y \in \mathbb{F}_q^n$
(IV) $d(x,z) \leq d(x,y) + d(y,z)$ *para todo* $x, y, z \in \mathbb{F}_q^n$

Demostración. Las tres primeras propiedades son obvias por la propia definición de la distancia. Veamos pues la propiedad iv).

Dados dos vectores $x, y \in \mathbb{F}_q^n$, definimos el conjunto $D(x,y) = \{i | x_i \neq y_i\}$, y denotamos el complementario por $D^c(x,y) = \{i | x_i = y_i\}$. Es claro que entonces el cardinal de $D(x,y)$ coincide con nuestra distancia.

Recordemos también algunas propiedades sobre cardinales de conjuntos. Sea un conjunto A, notemos por $|A|$ su cardinal. Entonces, para cualesquiera conjuntos A y B:

(I) $|A| \leq |A \cup B|$
(II) $|A \cup |B| \leq |A| + |B|$

(III) Si $|A| \leq |B|$, entonces $|A^c| \geq |B^c|$

Con esto, dados $x, y, z \in \mathbb{F}_q^n$, conjuntos tenemos que

$$\begin{aligned} D^c(x,z) &= \{i|x_i = z_i\} = \{i|x_i = z_i = y_i\} \cup \{i|x_i = z_i \neq y_i\} \\ \implies |D^c(x,z)| &\geq |\{i|x_i = z_i = y_i\}| = |\{i|x_i = y_i\} \cap \{i|z_i = y_i\}| \\ \implies |D(x,z)| &\leq |\{i|x_i \neq y_i\} \cup \{i|z_i \neq y_i\}| = |D(x,y) \cup D(y,z)| \\ \implies |D(x,z)| &\leq |D(x,y)| + |D(y,z)| \end{aligned}$$

quedándo demostrado el resultado. □

Ahora, la *distancia (mínima)* de un código $\mathcal{C}$ es la mínima distancia entre dos palabras distintas de dicho código. Esta propiedad será crucial a la hora de determinar el número de errores que podrá corregir un código.

Definición 5. Diremos que el peso (*de Hamming*) $wt(x)$ de un vector $x \in \mathbb{F}_q^n$ es el número de coordenadas distintas de cero de x.

Si tenemos dos vectores $x, y \in \mathbb{F}_q^n$ es inmediato comprobar que $d(x,y) = wt(x-y)$. En el siguiente mostramos la relación entre la distancia y el peso.

Proposición 2. *Si $\mathcal{C}$ es un código lineal, la distancia mínima coincide con el mínimo de los pesos de las palabras distintas de cero de C.*

Demostración. Sea $d = d(x,y)$ la distancia mínima del código $\mathcal{C}$, que se alcanza entre dos vectores $x, y \in \mathcal{C}$, y $d' = wt(z)$ el peso mínimo que se alcanza en $z \in C$.

Por ser $\mathcal{C}$ un subespacio vectorial, $0 \in \mathcal{C}$, y por tanto $d \leq d(z,0) = wt(z) = d'$. De nuevo por ser $\mathcal{C}$ un subespacio vectorial tenemos que $x - y \in \mathcal{C}$, luego $d' \leq wt(x-y) = d(x,y) = d$, de donde $d = d'$. □

Como consecuencia a este resultado, para códigos lineales, a la distancia mínima también se le llama *peso mínimo* del código. En adelante, si el peso mínimo d de un $[n,k]$ código es conocido, entonces nos referiremos al código como un $[n,k,d]$ código.

A continuación mostramos que existe una relación elemental entre el peso de una palabra y una matriz de paridad de un código lineal.

Proposición 3. *Sea $\mathcal{C}$ un código lineal con matriz de paridad H. Si $c \in \mathcal{C}$, las columnas de H que corresponde a coordenadas no nulas de c son linealmente dependientes. En el sentido contrario, si existe una dependencia lineal con coeficientes no nulos entre ω columnas de H, entonces existe una palabra en $\mathcal{C}$ de peso w cuyas coordenadas no nulas corresponden a dichas columnas.*

Demostración. Sea $c = (c_1, c_2, \ldots, c_n) \in \mathcal{C}$, $J \subset \{1, 2, \ldots, n\}$ tal que $c_j \neq 0$ para todo $j \in J$. Entonces, por ser H una matriz de paridad de $\mathcal{C}$ tenemos que

$$Hc^T = 0,$$

es decir, si $H = (h_{ij})$ con $i \in \{1, \ldots, n-k\}, j \in \{1, \ldots, n\}$,

$$Hc^T = \begin{pmatrix} \sum_{j=1}^n h_{1j}c_j \\ \sum_{j=1}^n h_{2j}c_j \\ \vdots \\ \sum_{j=1}^n h_{(n-k)j}c_j \end{pmatrix} = \sum_{j=1}^n c_j \begin{pmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{(n-k)j} \end{pmatrix} = \sum_{j \in J} c_j \begin{pmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{(n-k)j} \end{pmatrix} = 0$$

quedando demostrada la primera parte.

Por otro lado, dado $J = \{j_1, \ldots, j_w\} \subset \{1, \ldots, n\}$, supongamos que tenemos una dependencia lineal entre las columnas asociadas a J dada por $c_{j_1}h_{ij_1} + \cdots + c_{j_w}h_{ij_w}$ para cualquier $i = 1, \ldots, n-k$. Entonces, si construimos $c = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$ como sigue

$$\begin{cases} c_j = 0 & \text{si } j \notin J \\ c_j = c_{j_l} & \text{si } j = j_l \in J \end{cases},$$

es evidente que $wt(c) = w$, y que

$$Hc^T = \sum_{j \in J} c_j \begin{pmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{(n-k)j} \end{pmatrix} = 0$$

terminando la demostración de la proposición. □

Una manera de encontrar la distancia mínima d de un código lineal es examinar todas las palabras no nulas. El siguiente corolario, que es consecuencia directa de la proposición recién demostrado, muestra como utilizar una matriz de paridad para hallar d.

Corolario 1. *Un código lineal tiene peso o distancia mínima d si y solo si su matriz de paridad tiene un conjunto de d columnas linealmente dependientes pero ninguno de $d-1$ columnas linealmente dependientes.*

1.3 CODIFICAR, DECODIFICAR, Y EL TEOREMA DE SHANNON

1.3.1 *Codificar*

Sea $\mathcal{C}$ un $[n, k]$ código lineal sobre el cuerpo $\mathbb{F}_q$ con matriz generadora G. Como este es un subespacio vectorial de $\mathbb{F}_q^n$ de dimensión k, contiene q^k palabras, que están en

correspondencia uno a uno con q^k posibles mensajes. Por esto, la forma más simple es ver estos mensajes como k-tuplas x en $\mathbb{F}_q^k$. Así, lo más común es codificar un mensaje x como la palabra $c = xG$. Si G está en forma estándar, las primeras k coordenadas son los símbolos de información x; el resto de $n - k$ símbolos son los simbolos de paridad, es decir, la redundancia añadida a x con el fin de poder recuperarla si ocurre algún error. Dicho esto, la matriz G puede no estar en forma estándar. En particular, si existen índices de columnas $i_1, i_2, \ldots, i_n$ tales que la matriz $k \times k$ formada por estas columnas es la matriz identidad, entonces el mensaje se encuentra en las coordenadas $i_1, i_2, \ldots, i_n$ separado pero sin modificar, es decir, el símbolo del mensaje x_j se encuentra en la componente i_j de la palabra código. Si esto ocurre diremos que el codificador es *sistemático*.

1.3.2 *Decodificar y el Teorema de Shannon*

El proceso de decodificar, consistente en determinar qué palabra (y por tanto qué mensaje x) fue mandado al recibir un vector y, es más complejo. Encontrar algoritmos de decodificación eficientes es una área de investigación muy relevante en la teoría de códigos debido a sus aplicaciones prácticas. En general, codificar es sencillo y decodificar es complicado, especialmente si tiene un tamaño suficientemente grande.

2

INTRODUCCIÓN A EXTENSIONES DE ORE

2.1 CONCEPTOS BÁSICOS SOBRE EXTENSIONES DE ORE

En esta sección introduciremos algunos conceptos básicos necesarios para continuar. Aunque las definiciones que realizamos a continuación se pueden hacer más generales, nosotros las haremos en función a lo que necesitaremos para la siguiente sección. Dicho esto, nuestro algoritmo trabajará sobre polinomios de Ore no conmutativos, con una única indeterminada x, con coeficientes en un cuerpo cualquiera. Precisando, nuestros polinomios serán elementos de un anillo asociativo $R = \mathbb{F}[x;\sigma]$, donde

- $\mathbb{F}$ es un cuerpo cualquiera.
- $\sigma : \mathbb{F} \to \mathbb{F}$ es un automorfismo de cuerpos de orden finito digamos n.

La construcción de $R = \mathbb{F}[x;\sigma]$ sigue de la siguiente forma:

- R es un $\mathbb{F}$-espacio vectorial a la izquierda sobre la base $\{x^n : n \geq 0$. Entonces, los elementos de R son polinomios a la izquierda de la forma $a_0 + a_1x + \cdots + a_nx^n$ con $a_i \in \mathbb{F}$.
- La suma de polimios es la usual.
- El producto de R está basado en las siguientes reglas: $x^n x^m = x^{n+m}$, para $m, n \in \mathbb{N}$, y $xa = \sigma(a)x$ para $a \in \mathbb{F}$. Este producto se extiende recursivamente a R.

El grado $\deg(f)$ de un polinomio no nulo $f \in R$, al igual que su coeficiente lider se definen de la manera usual, tal que

$$f = \mathrm{lc}(f)x^{\deg(f)} + f_{\downarrow}, \text{ con } \deg(f_{\downarrow}) < \deg(f) \text{ y } \mathrm{lc}(f) \neq 0.$$

Escribimos $\deg(0) = -\infty$, con las convenciones usuales para este símbolo, y $\mathrm{lc}(0) = 0$.

Es facil comprobar que, dados $f, g \in R$:

$$\deg(fg) = \deg(f) + \deg(g), \quad \mathrm{lc}(fg) = \mathrm{lc}(f)\sigma^{\deg(f)}(\mathrm{lc}(g)).$$

Esto nos dice que R es un dominio de integridad no conmutativo.

El anillo R tiene algoritmos de división a la izquierda y derecha (veánse los algoritmos 1 y 2).

Algoritmo 1: División Euclídea a la izquierda

Entrada: $f, g \in \mathbb{F}[x;\sigma]$ con $g \neq 0$
Salida: $q, r \in \mathbb{F}[x;\sigma]$ tales que $f = qg + r$ y $\deg(r) < \deg(g)$
Inicialización: $q := 0, r := f$
while $\deg(g) \leq \deg(r)$ **do**
  $a = \mathrm{lc}(r)\sigma^{deg(r)-deg(g)}(\mathrm{lc}(g)^{-1})$
  $q := q + ax^{\deg(r)-\deg(g)}, r := r - ax^{deg(r)-deg(g)}g$

Algoritmo 2: División Euclídea a la derecha

Entrada: $f, g \in \mathbb{F}[x;\sigma]$ con $g \neq 0$
Salida: $q, r \in \mathbb{F}[x;\sigma]$ tales que $f = gq + r$ y $\deg(r) < \deg(g)$
Inicialización: $q := 0, r := f$
while $\deg(g) \leq \deg(r)$ **do**
  $a = \sigma^{-deg(g)}(\mathrm{lc}(g)^{-1}\,\mathrm{lc}(r))$
  $q := q + ax^{\deg(r)-\deg(g)}, r := r - gax^{deg(r)-deg(g)}$

Mostramos a continuación la demostración que justifica estos algoritmos.

Teorema 1. *Sea $\mathbb{F}$ un cuerpo finito de q elementos siendo q una potencia de un primo, σ un autormorfismo de $\mathbb{F}$ no nulo, y $R = \mathbb{F}[x;\sigma]$ la extensión de Ore correspondiente. Entonces, dados $f, g \in R$ exiten $q, r \in R$ únicos tales que:*

(I) $f = qg + r$.
(II) $\deg(r) < \deg(g)$.

Bajo las mismas hipótesis, existen también $q, r \in R$ únicos tales que:

(I) $f = gq + r$
(II) $\deg(r) < \deg(g)$.

Demostración. Para abreviar digamos $m = \deg(g), n = \deg(f)$. Veamos primero la prueba de la división a la izquierda. Si $m > n$, entonces no tenemos nada que probar, pues tomando $q = 0, r = f$ se cumple el resultado. Por otro lado, si $m \leq n$, sean $f = \sum_{i=0}^{n} a_i x^i$ y $g = \sum_{j=0}^{m} b_j x^j$, aplicaremos inducción sobre n. Si $n = 0$, entonces también $m = 0$, así que $f = a_0, g = b_0$, y por tanto tomamos $r = 0, q = a_0 b_0^{-1}$.

Por tanto, supongamos la afirmación cierta para todo f de grado menor que n. Sea $a = a_n \sigma^{n-m}(b_m^{-1})$. Entonces es claro que

$$\deg(ax^{n-m}g) = n,$$

$$\mathrm{lc}(ax^{n-m}g) = a_n.$$

Por tanto tenemos que

$$\deg(f - ax^{n-m}g) < n,$$

y por tanto, la hipótesis de inducción nos dice que existen q' y r' compliendo que $\deg(r') < \deg(g)$ y

$$f - ax^{n-m}g = q'g + r'.$$

Sea

$$q = ax^{n-m} + q',$$

entonces

$$f = ax^{n-m}g + q'g + r' = qg + r'.$$

Queda probar que q y r son únicos como tales. Supongamos que

$$f = q_1g + r_1 = q_2g + r_2,$$

con $\deg(r_1), \deg(r_2) < \deg(g)$. Entonces, $(q_1 - q_2)g = r_2 - r_1$, y podemos afirmar entonces que

$$\deg(q_1 - q_2) + \deg(g) = \deg((q_1 - q_2)g)$$
$$= \deg(r_2 - r_1) \leq \max(\deg(r_2), \deg(r_1)) < \deg(g).$$

Esto prueba que $\deg(q_1 - q_2) = -\infty$, demostrando que $q_1 - q_2 = 0$ y que $r_2 = r_1 = 0$ que termina la prueba de la primera parte.

La prueba de la división a la derecha es completamente análoga, tomando $a = \sigma^{-m}(a_n b_m^{-1})$, y utilizando que $\deg(f - gax^{n-m}) < n$.

□

Los polinomios r y q obtenidos como salida del algoritmo 1 los llamaremos *resto a la izquierda* y *cociente a la izquierda*, respectivamente, de la división a la izquierda de f por g. Utilizaremos la notación $r = \operatorname{lrem}(f, g)$ y $q = \operatorname{lquo}(f, g)$. Asumimos convenciones y notaciones análogas para el algoritmo de división a la derecha.

Como consecuencia del algoritmo de división a la izquierda, dado un ideal a la izquierda I de R, y cualquier polinomio no nulo $f \in I$ de grado mínimo, obtenemos que f es un generador de I. Notaremos en este caso $I = Rf$. Análogamente, cualquier ideal a la derecha de R es principal. Por tanto R es un dominio de ideales principales no conmutativo.

Dados $f, g \in R$, $Rf \subset Rg$ implica que g es un *divisor a la derecha* de f, simbólicamente $g|_r f$, o que f es *múltiplo a la izquierda* de g.

Por ser R un DIP sabemos que $Rf + Rg = Rd$ para algún $d \in R$, y es inmediato comprobar que $d|_r f$ y $d|_r g$. Además, si tenemos d' con $d'|_r f$, $d'|_r g$, entonces $Rf + Rg \subset Rd'$, luego $Rd \subset Rd'$ y por tanto $d|_r d'$. En este caso diremos que d es un máximo común divisor a la derecha de f y g, estando unívocamente determinado

salvo múltiplicación a la izquierda por una unidad de R. Utilizaremos la notación $d = (f,g)_r$. Además de aquí obtenemos directamente la **identidad de Bezout**.

Similarmente $Rf \cap Rg = Rm$ si y solo si m es un mínimo común múltiplo a la izquierda de f y g, notado por $m = [f,g]_l$. Este también es único salvo multiplicación a la izquierda por una unidad de R.

Tanto $(f,g)_r$ como $[f,g]_l$ se pueden calcular utilizando el Algoritmo Extendido de Euclides a la izquierda. La versión a la derecha de estas definiciones y propiedades puede establecerse análogamente. Más adelante utilizaremos el Algoritmo Extendido de Euclides a la derecha, que nos proporciona los coeficientes de Bezout en cada etapa del algoritmo, por esto es la que demostramos a continuación y describimos explícitamente en el algoritmo 3.

Algoritmo 3: Algoritmo extendido de Euclides a la derecha

Entrada: $f, g \in \mathbb{F}[x;\sigma]$ con $f \neq 0,\ g \neq 0$
Salida: $\{u_i, v_i, r_i\}_{i=0,\dots,h,h+1}$ tales que $r_i = fu_i + gv_i$ para todo $i, r_h = (f,g)_l$, y $u_{h+1}f = [f,g]_r$.
Inicialización:
$r_0 \leftarrow f, r_1 \leftarrow g$
$u_0 \leftarrow 1, u_1 \leftarrow 0$
$v_0 \leftarrow 0, v_1 \leftarrow 1$
$q \leftarrow 0, rem \leftarrow 0$
$i \leftarrow 1$
while $r_i \neq 0$ **do**
  $q,\ rem \leftarrow \text{rquot-rem}(r_{i-1}, r_i)$
  $r_{i+1} \leftarrow rem$
  $u_{i+1} \leftarrow u_{i-1} - u_i q$
  $v_{i+1} \leftarrow v_{i-1} - v_i q\ i \leftarrow i+1$
  return $\{u_i, v_i, r_i\}_{i=0,\dots,h,h+1}$

Antes de pasar con la demostración del algoritmo probaremos un resultado que nos será necesario para esta.

Proposición 4. *Sean $u, v, f, g \in R$ tales que $fu = gv$. Entonces, $(u,v)_l = 1 \implies [f,g]_r = fu$.*

Demostración. La identidad de Bezout nos proporciona coeficientes a, b tales que $1 = ua + vb$. Entonces, dado $h \in fR \cap gR$, veamos que $fu|_l h$. Sean u', v' tales que $h = fu' = gv'$,

$$\begin{aligned} fu' &= fu'ua + fu'vb \\ gv' &= gv'ua + gv'vb \end{aligned}.$$

□

Teorema 2. *El algoritmo 3 es correcto.*

Demostración. En primer lugar, vemos que siempre que $r_i \neq 0$ se tiene que $\deg(r_{i+1}) < \deg(r_i)$, por tanto existe $h \geq 1$ tal que $r_h \neq 0$ pero $r_{h+1} = 0$.

Para $i \leq h$ tenemos que $r_i \neq 0$, y por tanto podemos utilizar la división a la derecha de r_{i-1} entre r_i para obtener

$$r_{i-1} = r_i q_{i+1} + r_{i+1}.$$

De aquí obtenemos que los divisores a la izquierda comunes de r_{i-1} y de r_i coinciden con los divisores a la izquierda comunes de r_i y de r_{i+1}. Luego

$$r_h = (0, r_h)_l = (r_{h+1}, r_h)_l = (r_h, r_{h-1-})_l = \cdots = (r_1, r_0)_l = (g, f)_l.$$

Ahora vamos a definir $u_i, v_i \in R$ con $i = 0, 1, \ldots, h, h+1$. En primer lugar, tomamos

$$u_0 = 1,\ v_0 = 0,\ u_1 = 0,\ v_1 = 1..$$

Una vez dados $u_{i-1}, v_{i-1}, u_i, v_i$ para $1 \leq i \leq h$ definimos

$$u_{i+1} = u_{i-1} - u_i q_{i+1}, \quad v_{i+1} = v_{i-1} - v_i q_{i+1}.$$

Aplicaremos un argumento por inducción para comprobar que $r_i = f u_i + g v_i$. Es inmediato comprobar que para $i = 0, 1$ se cumple, así que supongamos que se cumple para $i-1,\ i$ y veamos que se cumple para $i+1$. En efecto

$$f u_{i+1} + g v_{i+1j} = f(u_{i-1} - u_i q_{i+1}) + g(v_{i-1} - v_i q_{i+1}) =$$

$$f u_{i-1} + g v_{i-1} - (f u_i + g v_i) q_{i+1} = r_{i-1} - r_i q_{i+1} = r_{i+1}.$$

TODO-*-*-* Veamos para concluir que $u_{h+1} f = [f, g]_r$. Observemos que $0 = r_{h+1} = f u_{h+1} + g v_{h+1}$, luego $f u_{h+1} = -g v_{h+1}$ es un múltiplo a la derecha común de f y g. Para terminar y ver que $f u_{h+1} = [f, g]_r$ demostremos primero que

$$-u_{i+1} v_i + u_i v_{i+1} = 1 \quad \forall 0 \leq i \leq h.$$

Esta igualdad es clara para $i = 0$. Si $1 \leq i \leq h$, supuesta la igualdad para $i-1$, tenemos

$$-u_{i+1} v_i + u_i v_{i+1} = -(u_{i-1} - u_i q_{i+1}) v_i + u_i (v_{i-1} - v_i q_{i+1}) = .$$

□

-*-*-*-*

Veamos ahora un lema que nos será útil posteriormente.

Lema 1. *Sean $f, g \in \mathbb{F}[x;\sigma]$ y $\{u_i, v_i, r_i\}_i = 0, \dots, h$ los coeficientes obtenido al aplicar el Algoritmo Extendido de Euclides a la derecha a f y g. Notemos $R_0 = \begin{pmatrix} u_0 & v_0 \\ u_1 & v_1 \end{pmatrix}$, $Q_i = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix}$ y $R_i = R_0 Q_1 \cdots Q_i$ para cualquier $i = 0, \dots, h$. Por tanto, para todo $i = 0, \dots, h$ se cumplen las siguientes afirmaciones:*

i) $(fg)R_i = (r_{i-1} r_i)$.
ii) $R_i = \begin{pmatrix} u_i & u_{i+1} \\ v_i & v_{i+1} \end{pmatrix}$.
iii) $f u_i + g v_i = r_i$.
iv) *R_i tiene inverso a izquierda y derecha.*
v) $(u_i, v_i)_r = 1$.
vi) $\deg f = \deg r_{i-1} + \deg v_i$.

Demostración. TODO □

Ahora, con el objetivo de poder definir la evaluación de nuestro polinomios, dado $j \geq 0$, definimos la *norma j-ésima* para cualquier $\gamma \in \mathbb{F}$ de forma recursiva como sigue:

$$N_0(\gamma) = 1$$
$$N_{j+1}(\gamma) = \gamma\sigma(N_j(\gamma)) = \gamma\sigma(\gamma)\dots\sigma^j(\gamma).$$

La noción de norma j-ésima admite también una versión para índices negativos dada por

$$N_{-j-1}(\gamma) = \gamma\sigma^{-1}(N_{-j}(\gamma)) = \gamma\sigma^{-1}(\gamma)\cdots\sigma^{-j}(\gamma).$$

La *evaluación la izquierda* de un polinomio no conmutativo $f \in R$ en un $a \in \mathbb{F}$ se define como el resto de la división a la izquierda de f por $x - a$, y de forma similar para la *evaluación a la derecha*. Estas evaluaciones nos permiten hablar de raíces a izquierda y derecha de estos polinomios.

Lema 2. *Sea $\gamma \in \mathbb{F}$ y $f = \sum_{i=0}^{n} f_i x^i \in R$. Entonces:*

(i) *El resto de la división a la izquierda de f por $x - \gamma$ es $\sum_{i=0}^{n} f_i N_i(\gamma)$.*
(ii) *El resto de la división a la derecha de f por $x - \gamma$ es $\sum_{i=0}^{n} \sigma^{-i}(f_i) N_{-i}(\gamma)$.*
(iii) *$N_j(\sigma^k(\gamma)) = \sigma^k(N_j(\gamma))$ para todo i, k.*

Demostración. Para demostrar *i)* observamos primero un caso especial de este resultado:

$$x^j - N_j(\gamma) \in R(x - \gamma)\ \forall j \geq 0. \tag{1}$$

Es evidente que el resultado es cierto para $j = 0$, asi que procedemos por inducción sobre j. Supongamos el resultado cierto para j, entonces

$$\begin{aligned} x^{j+1} - N_{j+1}(\gamma) &= x^{j+1} - \sigma(N_j(\gamma))\gamma \\ &= x^{j+1} + \sigma(N_j(\gamma))(x-\gamma) - \sigma(N_j(\gamma))x \\ &= x^{j+1} + \sigma(N_j(\gamma))(x-\gamma) - xN_j(\gamma) \\ &= \sigma(N_j(\gamma))(x-\gamma) + x(x^j - N_j(\gamma)) \in R(x-\gamma) \end{aligned}$$

Utilizando (1) tenemos entonces que

$$f - \sum_{i=0}^{n} f_i N_i(\gamma) = \sum_{i=0}^{n} f_i(x^i - N_i(\gamma)) \in R(x-\gamma)$$

y, por la unicidad del resto de la división euclídea tenemos que $r = \sum_{i=0}^{n} f_i N_i(\gamma)$.

Para la siguiente afirmación procedemos de forma similar. Veamos en primer lugar que

$$x^j - N_{-j}(\gamma) \in (x-\gamma)R \ \forall j \geq 0. \tag{2}$$

De nuevo, es obvio para $j = 0$, por tanto procedemos por inducción supuesto cierto para j.

$$\begin{aligned} x^{j+1} - N_{-j-1}(\gamma) &= x^{j+1} - \gamma\sigma^{-1}(N_j(\gamma)) \\ &= x^{j+1} + (x-\gamma)\sigma^{-1}(N_{-j}(\gamma)) - x\sigma^{-1}(N_{-j}(\gamma)) \\ &= x^{j+1} + (x-\gamma)\sigma^{-1}(N_{-j}(\gamma)) - N_{-j}(\gamma)x \\ &= (x-\gamma)\sigma^{-1}(N_{-j}(\gamma)) + (x^j - N_{-j}(\gamma))x \in (x-\gamma)R \end{aligned}$$

Así, usando (2) vemos que

$$\begin{aligned} f - \sum_{i=0}^{n} \sigma^{-i}(f_i)N_{-i}(\gamma) &= \sum_{i=0}^{n} x^i\sigma^{-i}(f_i) - N_{-i}(\gamma)\sigma^{-i}(f_i) \\ &= \sum_{i=0}^{n} (x^i - N_{-i}(\gamma))\sigma^{-i}(f_i) \in (x-\gamma)R. \end{aligned}$$

Así que por la unicidad del resto queda probado *ii)*.

Para probar *iii)*,

$$\begin{aligned} N_j(\sigma^k(\gamma)) &= \sigma^k(\gamma)\sigma^{k+1}(\gamma)\cdots\sigma^{k+j-1}(\gamma) \\ &= \sigma^k(\gamma\sigma(\gamma)\ldots\sigma^{j-1}(\gamma)) = \sigma^k(N_j(\gamma)). \end{aligned}$$

□

3

ALGORITMO PARA CÓDIGOS REED-SOLOMON SESGADOS

3.1 INTRODUCCIÓN

En este capítulo seguiremos la notación introducida en el capítulo anterior, por tanto $\mathbb{F}$ será un cuerpo cualquiera, σ un autormorfismo de $\mathbb{F}$ de orden finito n, y $R = \mathbb{F}[x;\sigma]$ el anillo de polinomio sesgados correspondiente.

En primer lugar comprobemos que el polinomio $x^n - 1$ es central en R. Efectivamente, dado $f = f_m x^m + \ldots + f_1 x + f_0 \in \mathbb{F}$, tenemos que

$$\begin{aligned}(x^n-1)f &= x^n f_m x^m - f_m x^m + \cdots + x^n f_1 x - f_1 x + x^n f_0 - f_0 \\ &= \sigma^n(f_m)x^{n+m} - f_m x^m + \cdots + \sigma^n(f_1)x^{n+1} - f_1 + \sigma^n(f_0)x^n - f_0 \\ &= f_m x^{n+m} - f_m x^m + \cdots + f_1 x^{n+1} - f_1 + f_0 x^n - f_0 \\ &= f(x^n - 1).\end{aligned}$$

Por tanto el ideal por la izquierda que genera es también un ideal por la derecha, así que podemos considerar el anillo cociente $\mathcal{R} = \mathbb{F}[x;\sigma]/\langle x^n - 1\rangle$. Entonces, cada ideal a la izquierda $I \leq \mathcal{R}$ designa un código $\mathcal{C} = \mathfrak{v}(I)$ de longitud n, donde $\mathfrak{v} : \mathcal{R} \to \mathbb{F}^n$ es el mapa de coordenadas asociado a la base $\mathcal{B} = \{1, x, \ldots, x^n\}$. Así, la longitud del código coincide con el orden de σ. De aquí en adelante suponemos establecidas estas condiciones. Llamaremos a cualquier código de este tipo *código cíclico sesgado*, o CCS para abreviar.

Recordemos la proposición TODO que nos dice que el centro de $\mathbb{F}[x;\sigma]$ es el anillo de polinomios conmutativo $\mathbb{F}^\sigma[x^n]$, donde $\mathbb{F}^\sigma$ denota el subcuerpo invariante por σ, es decir, los elementos $a \in \mathbb{F}$ tales que $\sigma(a) = a$. Para continuar necesitamos conocer mejor la estructura de nuestro anillo $\mathcal{R}$, y para demostraremos el siguiente teorema.

Teorema 3. *El anillo $\mathcal{R}$ es isomorfo al anillo de matrices $\mathcal{M}_n(\mathbb{F}^\sigma)$. Como consecuencia, para cada $k \leq n$ existe un CCS de dimensión k. Mas aún, cada CCS se puede ver como el ideal a la izquierda generado por un elemento idempotente.*

Demostración. □

Veamos a continuación un método para construir CCSs. Sabemos que todo ideal a la izquierda de $\mathcal{R}$ es principal, pues ya dijimos que los anillos de polinomios sesgados son dominios de ideales principales a la izquierda, así que todo CCS está generado por un divisor a la derecha de $x^n - 1$. Por tanto, de forma análoga a como se hace para polinomios cíclicos, necesitaremos encontrar factores a la derecha de este. El problema que nos encontramos es que, hasta donde sabemos, no existe un algoritmo de factorización completa para polinomios de Ore sobre un cuerpo cualquier. Por tanto construiremos un procedimiento específico para $x^n - 1$. Veamos primero un método para encontrar divisores lineales a la derecha.

Proposición 5. *Sea $\beta \in \mathbb{F}$, entonces $x - \beta$ divide por la derecha a $x^n - 1$ si y solo si $\beta = \sigma(c)c^{-1}$ para algún $c \in \mathbb{F}$ distinto de cero.*

Demostración. Sea $R = \mathbb{F}[x;\sigma]$. Si $x - \beta$ divide por la derecha a $x^n - 1$, entonces $R/R(x-\beta) \cong R/R(x-1)$ como $\mathcal{R}$-módulos. Por la (TODO referencia), tenemos $\beta = \sigma(c)c^{-1}$ para algún $c \in \mathbb{F}$ distinto de o. Para la ver la implicación contraria, sea $c \in \mathbb{F}$, y $\beta = \sigma(c)c^{-1} \ldots$ (TODO). □

Por el teorema 3, sabemos que $x^n - 1$ se puede descomponer como el mínimo común múltiplo por la izquierda de polinomios lineales (correspondiendo al ideal cero de $\mathcal{R}$ visto como intersección de n submódulos a la izquierda maximales (TODO)). Con el objetivo de encontrar una descomposición de este tipo de $x^n - 1$ (y por tanto divisores a la derecha no lineales (TODO)), nuestra estrategia es construir $\beta \in \mathbb{F}$ tal que

$$[x - \beta, x - \sigma(\beta), \ldots, x - \sigma^{n-1}(\beta)]_l = x^n - 1 \tag{3}$$

Por analogía con [Tur36] es claro que 3 se cumple si y solo el determinante de la matriz

$$\begin{pmatrix} 1 & \beta & \beta\sigma(\beta) & \ldots & \beta\sigma(\beta)\ldots\sigma^{n-2}(\beta) \\ 1 & \sigma(\beta) & \sigma(\beta)\sigma^2(\beta) & \ldots & \sigma(\beta)\sigma^2(\beta)\ldots\sigma^{n-1}(\beta) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma^{n-1}(\beta) & \sigma^{n-1}(\beta)\beta & \ldots & \sigma^{n-1}(\beta)\beta\ldots\sigma^{n-3}(\beta) \end{pmatrix}$$

es distinto de cero. Utilizando que $\beta = \sigma(c)c^{-1}$ por la proposición 5, esto es equivalente a que el determinante de la matriz

$$\begin{pmatrix} c & \sigma(c) & \sigma^2(c) & \ldots & \sigma^{n-1}(c) \\ \sigma(c) & \sigma(c)^2 & \sigma^3(c) & \ldots & c \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma^{n-1}(c) & c & \sigma(c) & \ldots & \sigma n-2(c) \end{pmatrix}$$

sea distinto de cero o, equivalentemente, que $\{c, \sigma(c), \ldots, \sigma^{n-1}(c)\}$ sea una base normal de la extensión de cuerpos $F^\sigma \subset \mathbb{F}$ [Tur36]. Recordamos que $F^\sigma = \mathbb{F}[t]$, donde $t \in \mathbb{F}$, que puede calcularse como se muestra en [Tur36]. En nuestro caso particular, el grupo que escogemos es el grupo cíclico $\{1, \sigma, \ldots, \sigma^{n-1}\}$, de manera que el elemento $t \in \mathbb{F}$ puede obtenerse elegiendo cualquier (... TODO). La existencia de un $c \in \mathbb{F}$ que genere dicha base está asegurada por el Teorema de la Base Normal. ...

Nuestro siguiente objetivo será proporcionar un método sistemático para contruir CCSs de una determinada distancia Hamming. Debido a la analogía con los códigos Reed-Solomon, los llamaremos *códigos Reed-Solomon sesgados* o *códigos RS sesgados* para abreviar. El siguiente resultado, que es un caso particulas de [Tur36], será importante en resultados posteriores. Mostraremos una prueba elemental de este.

Lema 3. *Sea L un cuerpo, σ un automorfismo de L de orden finito n, y $K = L^\sigma$ el subcuerpo invariante bajo σ. Sea $\{a_0, \ldots, a_{n-1}\}$ una K-base de L. Entonces, para todo $t \leq n$, y cada subconjunto $k_0 < k_1 < \ldots < k_{t-1} \subset \{0, 1, \ldots, n-1\}$*

$$\begin{vmatrix} \alpha_{k_0} & \alpha_{k_1} & \ldots & \alpha_{k_{t-1}} \\ \sigma(\alpha_{k_0}) & \sigma(\alpha_{k_1}) & \ldots & \sigma(\alpha_{k_{t-1}}) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{t-1}(\alpha_{k_0}) & \sigma^{t-1}(\alpha_{k_1}) & \ldots & \sigma^{t-1}(\alpha_{k_{t-1}}) \end{vmatrix} \neq 0.$$

Demostración. Realizaremos la prueba por inducción sobre t. El caso $t = 1$ se cumple trivialmente. Por tanto, supongamos que el lema se cumple para un cierto $t \geq 1$. Tenemos que comprobar que, para toda matriz $(t+1) \times (t+1)$

$$\Delta = \begin{pmatrix} \alpha_{k_0} & \alpha_{k_1} & \ldots & \alpha_{k_t} \\ \sigma(\alpha_{k_0}) & \sigma(\alpha_{k_1}) & \ldots & \sigma(\alpha_{k_t}) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^t(\alpha_{k_0}) & \sigma^t(\alpha_{k_1}) & \ldots & \sigma^t(\alpha_{k_t}) \end{pmatrix}.$$

el determinante $|\Delta|$ es distinto de cero. Supongamos por el contrario que $|\Delta| = 0$. Por la hipótesis de inducción tenemos que las primeras t columnas de Δ son linealmente independientes, luego existen $a_0, \ldots, a_{t-1} \in L$ tales que

$$(\alpha_{k_t}, \sigma^{t-1}(\alpha_{k_t}), \ldots, \sigma^t(\alpha_{k_t})) = \sum_{j=0}^{t-1} a_j (\alpha_{k_j}, \sigma^{t-1}(\alpha_{k_j}), \ldots, \sigma^t(\alpha_{k_j})).$$

Es decir, $a_0, \ldots, a_{t-1}$ satisfacen el sistema lineal

$$\begin{cases} \alpha_{k_t} = a_0\alpha_{k_0} + \cdots + a_{t-1}\alpha_{k_{t-1}} \\ \sigma(\alpha_{k_t}) = a_0\sigma(\alpha_{k_0}) + \cdots + a_{t-1}\sigma(\alpha_{k_{t-1}}) \\ \vdots \\ \sigma^t(\alpha_{k_t}) = a_0\sigma^t(\alpha_{k_0}) + \cdots + a_{t-1}\sigma^t(\alpha_{k_{t-1}}) \end{cases} . \tag{4}$$

Para cada $j = 0, \ldots, t-1$, restamos en (4) la ecuación $j+1$ transformada por σ^{-1} a la ecuación j. Esto produce el siguiente sistema lineal homogéneo

$$\begin{cases} 0 = (a_0 - \sigma^{-1}(a_0))\alpha_{k_0} + \cdots + (a_{t-1} - \sigma^{-1}(a_{t-1}))\alpha_{k_{t-1}} \\ 0 = (a_0 - \sigma^{-1}(a_0))\sigma(\alpha_{k_0}) + \cdots + (a_{t-1} - \sigma^{-1}(a_{t-1}))\sigma(\alpha_{k_{t-1}}) \\ \vdots \\ 0 = (a_0 - \sigma^{-1}(a_0))\sigma^{t-1}(\alpha_{k_0}) + \cdots + (a_{t-1} - \sigma^{-1}(a_{t-1}))\sigma^{t-1}(\alpha_{k_{t-1}}) \end{cases} . \tag{5}$$

La matriz de coeficientes de (5) es no singular, por la hipótesis de inducción, así que tenemos que para todo $j = 0, \ldots, t-1$, $a_j - \sigma^{-1}(a_j) = 0$, y por tanto $a_0, \ldots, a_{t-1} \in K$. Como consecuencia, la ecuación (4) establece una dependencia lineal sobre K de la K-base $\{\alpha_0, \ldots, \alpha_{n-1}\}$, creando una contradicción. Por tanto, $|\Delta| \neq 0$ y el resultado queda demostrado. □

Lema 4. *Sea $\alpha \in \mathbb{F}$ tal que $\{\alpha, \sigma(\alpha), \ldots, \sigma^{n-1}(\alpha)\}$ sea una base de $\mathbb{F}$ como $\mathbb{F}^\sigma$-espacio vectorial. Fijemos $\beta = \alpha^{-1}\sigma(\alpha)$. Para todo subconjunto $T = \{t_1 < t_2 < \cdots < t_m\} \subset \{0, 1, \ldots, n-1\}$, los polinomios*

$$g^l = [x - \sigma^{t_1}(\beta), x - \sigma^{t_2}(\beta), \ldots, x - \sigma^{t_m}(\beta)]_l$$

y

$$g^r = [x - \sigma^{t_1}(\beta^{-1}), x - \sigma^{t_2}(\beta^{-1}), \ldots, x - \sigma^{t_m}(\beta^{-1})]_r$$

tienen grado m. Además, si $x - \sigma^s(\beta)|_r g^l$ o $x - \sigma^s(\beta^{-1})|_l g^r$, entonces $s \in T$.

Demostración. Supongamos que $\deg g^l < m$, así que $g^l = \sum_{i=0}^{m-1} g_i x^i$. Como g es un múltiplo a la izquierda de $x - \sigma^{t_j}(\beta)$ para todo $1 \leq j \leq m$, por el lema 2 tenemos que

$$\sum_{i=0}^{m-1} g_i N_i(\sigma^{t_j}(\beta)) = 0 \text{ para todo } 1 \leq j \leq m \tag{6}$$

Esto es un sistema lineal homogéneo cuya matriz de coeficientes es la traspuesta de la matriz M dada por

$$\begin{pmatrix} N_0(\sigma^{t_1}(\beta)) & N_0(\sigma^{t_2}(\beta)) & \dots & N_0(\sigma^{t_m}(\beta)) \\ N_1(\sigma^{t_1}(\beta)) & N_1(\sigma^{t_2}(\beta)) & \dots & N_1(\sigma^{t_m}(\beta)) \\ \vdots & \vdots & \ddots & \vdots \\ N_{m-1}(\sigma^{t_1}(\beta)) & N_{m-1}(\sigma^{t_2}(\beta)) & \dots & N_{m-1}(\sigma^{t_m}(\beta)) \end{pmatrix}.$$

Fijémonos que $N_i(\sigma^{t_j}(\beta)) = \sigma^{t_j}(N_i(\beta)) = \sigma^{t_j}(N_i(\alpha^{-1}\sigma(\alpha))) = \sigma^{t_j}(\alpha^{-1})\sigma^{t_{j+i}}(\alpha)$ para todo $1 \leq j \leq m$ y $0 \leq i \leq m-1$. Por tanto, $|M| = 0$ si y solo si el determinante de la matriz M',

$$\begin{pmatrix} \sigma^{t_1}(\alpha) & \sigma^{t_2}(\alpha) & \cdots & \sigma^{t_m}(\alpha) \\ \sigma^{t_1+1}(\alpha) & \sigma^{t_2+1}(\alpha) & \cdots & \sigma^{t_m+1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{t_1+m-1}(\alpha) & \sigma^{t_2+m-1}(\alpha) & \cdots & \sigma^{t_m+m-1}(\alpha) \end{pmatrix},$$

o equivalentemente

$$\begin{pmatrix} \sigma^{t_1}(\alpha) & \sigma^{t_2}(\alpha) & \cdots & \sigma^{t_m}(\alpha) \\ \sigma(\sigma^{t_1}(\alpha)) & \sigma(\sigma^{t_2}(\alpha)) & \cdots & \sigma(\sigma^{t_m}(\alpha)) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{m-1}(\sigma^{t_1}(\alpha)) & \sigma^{m-1}(\sigma^{t_2+m-1}(\alpha)) & \cdots & \sigma^{m-1}(\sigma^{t_m}(\alpha)) \end{pmatrix},$$

es cero. Sin embargo, por el lema 3, $|M'| \neq 0$, luego la única solución del sistema lineal (6) es $g_0 = \cdots = g_{m-1} = 0$, siendo una contradicción. Por tanto $\deg g^l = m$. Para el otro polinomio razonamos de forma análoga. Si $\deg g^r < m$ y $g^r = \sum_{i=0}^{m-1} g_i x^i$, obtenemos el sistema lineal

$$\sum_{i=0}^{m-1} \sigma^{-i}(g_i) N_{-i}(\sigma^{t_j}(\beta^{-1})) = 0 \text{ para todo } 1 \leq j \leq m. \tag{7}$$

Vemos que $N_{-i}(\sigma^{t_j}(\beta^{-1})) = \sigma^{t_j}(\alpha^{-1})\sigma^{t_j-i+1}(\alpha)$ para $0 \leq i \leq m-1$ y $1 \leq j \leq m$. Entonces, de nuevo por el lema 3 el sistema tiene una única solución $\sigma^{-i}(g_i) = 0$ para $0 \leq i \leq m-1$, luego $g_0 = g_1 = \cdots = g_{m-1} = 0$. Como dijimos esto es una contradicción y por tanto $\deg g^r = m$. □

Con esto, ya tenemos los resultados suficientes para definir los códigos sobre los que está definido nuestro algoritmo.

Definición 6. Sean $\alpha, \beta \in \mathbb{F}$ verificando las condiciones del lema 4. Un código Reed-Solomon (RS) sesgado de distancia fijada $\delta \leq n$ es un CCS generado por $[x - \sigma^r(\beta), x - \sigma^{r+1}, \cdots, x -$ para algún $r \geq 0$.

Teorema 4. *Sea $\mathcal{C}$ un código RS sesgado de distancia fijada δ. La distancia Hamming de $\mathcal{C}$ es δ.*

Demostración. Definamos en primer lugar

$$g = [x - \sigma^r(\beta), x - \sigma^{r+1}(\beta), \ldots, x - \sigma^{r+\delta-2}(\beta)]_l$$

un generador de $\mathcal{C}$ como ideal a la izquierda de $\mathcal{R}$. Entonces, una matriz de paridad H de $\mathcal{C}$ es

$$\begin{pmatrix} N_0(\sigma^r(\beta)) & N_1(\sigma^r(\beta)) & \cdots & N_{n-1}(\sigma^r(\beta)) \\ N_0(\sigma^{r+1}(\beta)) & N_1(\sigma^{r+1}(\beta)) & \cdots & N_{n-1}(\sigma^{r+1}(\beta)) \\ \vdots & \vdots & \ddots & \vdots \\ N_0(\sigma^{r+\delta-2}(\beta)) & N_1(\sigma^{r+\delta-2}(\beta)) & \cdots & N_{n-1}(\sigma^{r+\delta-2}(\beta)) \end{pmatrix}.$$

pues sus filas dan la evaluación a la derecha de las raices de g. Entonces, por el corolario 1, nos basta con probar no existe ningún conjunto de $\delta - 1$ columnas linealmente dependientes. Para ello procedemos de forma similar a la demostración del lema 4. Como ya utilizamos antes, $N_i(\sigma^k(\beta)) = \sigma^k(N_i(\beta)) = \sigma^k(\alpha^{-1})\sigma^{i+k}(\alpha)$ para cualesquiera enteros i y k. Por tanto, dado cualquier conjunto de columnas de tamaño $\delta - 1$, podemos verlo como la matriz M

$$\begin{pmatrix} N_{k_1}(\sigma^r(\beta)) & N_{k_2}(\sigma^r(\beta)) & \cdots & N_{k_{\delta-1}}(\sigma^r(\beta)) \\ N_{k_1}(\sigma^{r+1}(\beta)) & N_{k_2}(\sigma^{r+1}(\beta)) & \cdots & N_{k_{\delta-1}}(\sigma^{r+1}(\beta)) \\ \vdots & \vdots & \ddots & \vdots \\ N_{k_1}(\sigma^{r+\delta-2}(\beta)) & N_{k_2}(\sigma^{r+\delta-2}(\beta)) & \cdots & N_{k_{\delta-1}}(\sigma^{r+\delta-2}(\beta)) \end{pmatrix},$$

con $\{k_1 < k_2 < \cdots < k_{\delta-1} \subset \{0, 1, \ldots, n-1\}$. Ahora, $|M| = 0$, si y solo $|M'| = 0$, donde M' es la matriz

$$\begin{pmatrix} \sigma^{k_1+r}(\alpha) & \sigma^{k_2+r}(\alpha) & \cdots & \sigma^{k_{\delta-1}+r}(\alpha) \\ \sigma^{k_1+r+1}(\alpha) & \sigma^{k_2+r+1}(\alpha) & \cdots & \sigma^{k_{\delta-1}+r+1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{k_1+r+\delta-2}(\alpha) & \sigma^{k_2+r+\delta-2}(\alpha) & \cdots & \sigma^{k_{\delta-1}+r+\delta-2}(\alpha) \end{pmatrix}$$

$$= \begin{pmatrix} \sigma^{k_1+r}(\alpha) & \sigma^{k_2+r}(\alpha) & \cdots & \sigma^{k_{\delta-1}+r}(\alpha) \\ \sigma(\sigma^{k_1+r}(\alpha)) & \sigma(\sigma^{k_2+r})(\alpha) & \cdots & \sigma(\sigma^{k_{\delta-1}+r}(\alpha)) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^{\delta-2}(\sigma^{k_1+r}(\alpha)) & \sigma^{\delta-2}(\sigma^{k_2+r}(\alpha)) & \cdots & \sigma^{\delta-2}(\sigma^{k_{\delta-1}+r}(\alpha)) \end{pmatrix}.$$

Por ser $\{\alpha, \sigma(\alpha), \ldots, \sigma^{n-1}(\alpha)$ una base de la extensión $\mathbb{F}^\sigma \subset \mathbb{F}$, por el lema 3, $|M'| \neq 0$, y por tanto esas columnas son linealmente independientes. □

De aquí en adelante $\mathcal{C}$ denotará un código RS sesgado de distancia fijada δ generado, como ideal a la izquierda de $\mathcal{R}$, por $g = [x - \sigma^r(\beta), x - \sigma^{r+1}(\beta), \ldots, x - \sigma^{r+\delta-2}(\beta)]_l$, para algún $r \geq 0$, donde β lo elegimos como en la definición 6. Sabemos que la distancia mínima de $\mathcal{C}$ es exactamente δ por el teorema 4. Sea $\tau = \lfloor (\sigma - 1)/2 \rfloor$ que es el máximo número de errores que nuestro código puede corregir. Por simplicidad, supondremos que $r = 0$. Esto no es una restricción, pues siempre podemos escribir $\beta' = \sigma^r(\beta)$. Entonces, $\beta' = (\alpha')^{-1}\sigma(\alpha')$, donde $\alpha' = \sigma^r(\alpha)$, y es claro que α' también proporciona una base normal. Por tanto, $g = [x - \beta', x - \sigma(\beta'), \ldots, x - \sigma^{\delta-2}(\beta')]$.

Sea $c \in \mathcal{C}$ una palabra que es transmitida a través de un canal ruidoso y el polinomio $y = c + e$ es recibido, donde $e = e_1 x^k + \cdots + e_\nu x^{k_\nu}$ con $\nu \leq \tau$. Definimos el polinomio *localizador de errores* como

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

En primer lugar mostraremos que λ determina las posiciones con un error no nulo.

Lema 5. *Para cualquier subconjunto* $\{t_1, \ldots, t_m\} \subset \{0, 1, \ldots, n-1\}$,

$$\left[1 - \sigma^{t_1}(\beta)x, \ldots, 1 - \sigma^{t_m}(\beta)x\right]_r = \left[x - \sigma^{t_1-1}(\beta^{-1}), \ldots, x - \sigma^{t_m-1}(\beta^{-1})\right]_r.$$

Demostración. Para cualquier $a \in \mathbb{F}$,

$$1 - ax = (x - \sigma^{-1}(a^{-1}))(-\sigma^{-1}(a)), \quad x - \sigma^{-1}(a^{-1}) = (1 - ax)(-\sigma^{-1}(a^{-1})).$$

Entonces, los polinomios del resultado se dividen a la izquierda mutuamente, y por tanto ambos mínimos comunes múltiplos coinciden. □

Proposición 6. *$1 - \sigma^d(\beta)x$ divide a la izquierda a λ si y solo si $x - \sigma^{d-1}(\beta^{-1})$ divide a la izquierda a λ si y solo si $d \in \{k_1, \ldots, k_\nu\}$.*

Demostración. Por el lema anterior, $1 - \sigma^d(\beta)x$ divide a la izquierda λ si y solo si $x - \sigma^{d-1}(\beta^{-1})$ divide a la izquierda a λ. Ahora, por el lema 4, $x - \sigma^{d-1}(\beta^{-1})$ es un divisor a la izquierda de λ si y solo si $d \in \{k_1, \ldots, k_\nu\}$. □

Por tanto, una vez que λ es conocido, las coordenadas del error pueden ser localizadas siguiendo la siguiente regla: $d \in 0, 1, \ldots, n-1$ es una posición de error si y solo si $\sigma^{d-1}(\beta-1)$ es una raiz a la izquierda de λ. En realidad, λ puede ser sustituido por cualquier otro polinomio en R asociado a la derecha a λ, es decir, cualquier polinomio que difiera de λ en la multiplicación a la derecha por un elemento no nulo de $\mathbb{F}$.

Para cualquier $1 \leq j \leq \nu$, $\lambda = (1 - \sigma^{k_j}(\beta)x)p_j$ para algún $p_j \in R$ con $\deg p_j = \nu - 1$. Definimos entonces el polinomio *evaluador de errores* como $\omega = \sum_{j=1}^{\nu} e_j \sigma^{k_j}(\alpha) p_j$. Así, si conocemos el polinimio localizador de errores λ y el polinomio evaluador de errores ω, podremos calcular los valores $e_1, e_2, \ldots, e_\nu$ resolviendo un sistema lineal dado por $\omega = \sum_{j=1}^{\nu} e_j \sigma^{k_j}(\alpha) p_j$. Además, es directo comprobar que $\deg \omega < \nu$, pues, como ya dijimos, $\deg(p_j) = \nu - 1$ para todo $1 \leq j \leq \nu$.

Finalmente, para cada $0 \leq i \leq n-1$, el i-ésimo síndrome S_i del polinomio recibido $y = \sum_{j=0}^{n-1} y_j x^j$ se define como el resto de la división a la derecha de y por $x - \sigma^i(\beta)$. Este S_i es la evaluación a la derecha de y en $\sigma^i(\beta)$. Siempre que $0 \leq i \leq 2\tau - 1 = \delta - 2$, las evaluaciónes a la derecha en c son cero, y por tanto

$$\begin{aligned} S_i &= \sum_{j=0}^{n-1} y_j N - j(\sigma^i(\beta)) \\ &= \sum_{j=1}^{\nu} e_j N_{k_j}(\sigma^i(\beta)) \\ &= \sum_{j=1}^{\nu} e_j \sigma^i(N_{k_j}(\beta)) \qquad . \\ &= \sum_{j=1}^{\nu} e_j \sigma^i(\alpha^{-1}) \sigma^{k_j+i}(\alpha) \\ &= \sigma^i(\alpha^{-1}) \sum_{j=1}^{\nu} e_j \sigma^{k_j+i}(\alpha) \end{aligned}$$

Por esto, $\sigma^i(\alpha) S_i = \sum_{j=1}^{\nu} e_j \sigma^{k_j+i}(\alpha)$ y llamamos al polinomio $S = \sum_{i=0}^{2\tau-1} \sigma^i(\alpha) S_i x^i$ el polinomio síndrome de y.

Teorema 5. *Los polinomios localizador de errores y evaluador de errores cumplen la ecuación clave no conmutativa:*

$$\omega = S\lambda + x^{2\tau} u.$$

donde $u \in R$ es de grado menor que ν.

Demostración. Por definición sabemos que

$$S = \sum_{i=0}^{2\tau-1} \sum_{j=1}^{\nu} e_j \sigma^{k_j+i}(\alpha) x^i,$$

y que para cualquier $1 \leq j \leq \nu$

$$\lambda = (1 - \sigma^{k_j}(\beta)x)p_j.$$

Queremos llegar a que

$$\omega = \sum_{j=1}^{\nu} e_j \sigma^{k_j}(\alpha)p_j = S\lambda + x^{2\tau u}$$

para algún u de grado menor que ν. Tomamos ahora $u = \sum_{i=0}^{2\tau-1} \sum_{j=1}^{\nu} \sigma^{-2\tau}(e_j)\sigma^{k_j}(\alpha)p_j$. Entonces

$$\begin{aligned} S\lambda + x^{2\tau}u &= \sum_{i=0}^{2\tau-1} \sum_{j=1}^{\nu} e_j \sigma^{k_j+i}(\alpha)x^i(1 - \sigma^{k_j}(\beta)x)p_j + x^{2\tau}\sigma^{-2\tau}(e_j)\sigma^{k_j}(\alpha)p_j \\ &= \sum_{j=1}^{\nu} e_j \left(\sum_{i=0}^{2\tau-1} \sigma^{k_j+i}(\alpha)x^i(1 - \sigma^{k_j}(\beta)x) + x^{2\tau}\sigma^{k_j}(\alpha) \right) p_j \end{aligned},$$

tras intercambiar las sumatorias, y sacar factor común e_j y p_j. Ahora nos centramos en la sumatoria interior.

$$\begin{aligned} &\sum_{i=0}^{2\tau-1} \sigma^{k_j+i}(\alpha)x^i(1 - \sigma^{k_j}(\beta)x) + x^{2\tau}\sigma^{k_j}(\alpha) \\ &= \sum_{i=0}^{2\tau-1} x^i\sigma^{k_j}(\alpha)(1 - \sigma^{k_j}(\beta)x) + x^{2\tau}\sigma^{k_j}(\alpha) \\ &= \sum_{i=0}^{2\tau-1} x^i\sigma^{k_j}(\alpha) - x^i\sigma^{k_j}(\alpha)\sigma^{k_j}(\alpha^{-1})\sigma^{k_j+1}(\alpha)x) + x^{2\tau}\sigma^{k_j}(\alpha). \\ &= \sum_{i=0}^{2\tau-1} x^i\sigma^{k_j}(\alpha) - x^{i+1}\sigma^{k_j}(\alpha) + x^{2\tau}\sigma^{k_j}(\alpha) \\ &= \left(\sum_{i=0}^{2\tau-1} x^i - x^{i+1} + x^{2\tau} \right) \sigma^{k_j}(\alpha) = \sigma^{k_j}(\alpha) \end{aligned}$$

Con esto, sustituyendo en la expresión anterior tenemos que

$$\begin{aligned} S\lambda + x^{2\tau}u &= \sum_{j=1}^{\nu} e_j \left(\sum_{i=0}^{2\tau-1} \sigma^{k_j+i}(\alpha)x^i(1 - \sigma^{k_j}(\beta)x) + x^{2\tau}\sigma^{k_j}(\alpha) \right) p_j \\ &= \sum_{j=1}^{\nu} e_j\sigma^{k_j}(\alpha)p_j = \omega \end{aligned}.$$

□

Ahora intentaremos resolver esta ecuación, para lo que utilizaremos el Algoritmo Euclídeo Extendido a la derecha presentado en 3. Recordemos que, para cualesquiera $f, g \in R$, cada paso i del algoritmo proporciona coeficientes $\{u_i, v_i, r_i\}$ tales que $fu_i + fv_i = r_i$, donde $(f, g)_l = r_h$ y $\deg r_{i+1} < \deg r_i$ para cualquier $0 \leq i \leq h - 1$.

Teorema 6. *La ecuación clave no conmutativa*

$$x^{2\tau}u + S\lambda = \omega \tag{8}$$

es un múltiplo a la derecha de la ecuación

$$x^{2\tau}u_I + Sv_I = r_I, \tag{9}$$

donde u_I, v_I y r_I son los coeficientes de Bezout dados por el Algoritmo Euclídeo Extendido a la derecha con entrada $x^{2\tau}$ y S, e I es el índice determinado por las condiciones $\deg r_{I-1} \geq \tau$ *y* $\deg r_I < \tau$. *En particular, $\lambda = v_I g$ y $\omega = r_I g$ para algún $g \in R$.*

Demostración. Recordemos que $\deg S < 2\tau$, $\deg \lambda \leq \nu \leq \tau$ y que $\deg \omega < \nu \leq \tau$. Entonces, $\deg u < \tau$, pues en otro caso $\deg x^{2\tau}u \geq 3\tau > \deg S\lambda$, y por tanto $\deg \omega \geq 3\lambda$ que es contradicción. Por otro lado, por el lema 1 *VI)*, $\deg v_I + \deg r_{I-1} = 2\tau$, y utilizando la hipótesis $\deg r_{I-1} \geq \tau$ tenemos que $\deg v_I \leq \tau$.

Consideremos ahora el mínimo común múltiplo a la derecha $[\lambda, v_I]_r = \lambda a = v_I b$, donde $a, b \in R$ con $\deg a \leq \deg v_I \leq \tau$ y $\deg b \leq \deg \lambda \leq \tau$. Entonces $(a, b)_r = 1$. Entonces, multiplicando 8 a la derecha por a, y 9 a la derecha por b, obtenemos

$$x^{2\tau}ua + S\lambda a = \omega a \tag{10}$$

y

$$x^{2\tau}u_I b + Sv_I b = r_I b. \tag{11}$$

Ahora, restando ambas ecuaciones obtenemos $x^{2\tau}(ua - u_I b) = \omega a - r_I b$. Por $\deg \omega < \tau, \deg a \leq \tau, \deg r_I < \tau, \deg b \leq \tau$, tenemos que $\deg(\omega a - r_I b) < 2\tau$, luego $ua = u_I b$ y $\omega a = r_I b$, pues en caso contrario el grado del término izquierdo sería mayor o igual que 2τ. De hecho $(a, b)_r = 1$ nos lleva a que $[u, u_I]_r = ua = u_I b$ y $[\omega, r_I]_r = \omega a = r_I b$, y en particular $\deg a \leq \deg r_I < \tau$.

Sea $[a, b]_l = a'a = b'b$. Por ser $[\lambda, v_I]_r$ un múltiplo a la izquierda de a y b, existe $m \in R$ tal que $[\lambda, v_I]_r = m[a, b]_l$, es decir, $\lambda a = v_I b = ma'a = mb'b$. Por esto, $\lambda = ma'$ y $v_I = mb'$ y, por minimalidad, $(\lambda, v_I)_l = m$. Podemos utilizar argumentos similares para probar que existen $m', m'' \in R$ tales que $u_I = m'b'$ y $u = m'a'$, y $r_I = m''b'$ y $\omega = m''a'$. Por el lema 1 *V)* $(u_I, v_I)_r = 1$, luego $b' = 1$. Esto completa la prueba, pues tenemos $b = bb' = aa'$, y por tanto $\lambda = v_I a'$, $\omega = r_I a'$ y $u = u_I a'$. □

Usando la notación del teorema recién demostrado, vemos que, por $\lambda = v_I a'$ y $\omega = r_I a'$, si $(\lambda, \omega)_r = 1$ entonces $\lambda = v_I$ y $\omega = r_I$. En este caso, el teorema 6 proporciona un método algorítmico para calcular ambos polinomios, el localizador de errores y el evaluador de errores. Como ya dijimos, estos dos polinomios nos permiten calcular el error al recibir un polinomio $y = c + e$, con c y e cumpliendo las hipótesis previamente mencionadas, por tanto, podemos describir un algoritmo de

decodificación para códigos RS sesgados (vease el algoritmo 4) que será válido siempre que $(\lambda, \omega)_r = 1$.

Algoritmo 4: Algoritmo de decodificación para códigos RS sesgados

Entrada: Un polinomio $y = \sum_{i=0}^{n-1} y_i x^i$ obtenido de la transmisión de una palabra código c perteneciente a un código RS sesgado C generado por $g = [\{x - \sigma^i(\beta)\}_{i=0,\dots,\delta-2}]_l$ con capacidad para corregir $\tau = \lfloor (\delta - 1)/2 \rfloor$ errores.

Salida: Una palabra código c' o un *error de ecuación clave*.

for $0 \leq i \leq 2\tau - 1$ **do**
  $S_i \leftarrow \sum_{j=0}^{n-1} y_j N_j(\sigma^i(\beta))$
$S \leftarrow \sum_{i=0}^{2\tau-1} \sigma^i(\alpha) S_i x^i$
if $S = 0$ **then**
  return y
$\{u_i, v_i, r_i\}_{i=0,\dots,l} \leftarrow \text{REEA}(x^{2\tau}, S)$
$I \leftarrow$ primera iteración en REEA tal que $\deg r_i < \tau$
$pos \leftarrow \emptyset$
for $0 \leq i \leq n - 1$ **do**
  if $\sigma^{i-1}(\beta^{-1})$ *es una raiz a la izquierda de* v_I **then**
    $pos = pos \cup \{i\}$
if $\deg v_I > \text{Cardinal}(pos)$ **then**
  return *error en ecuación clave*
for $j \in pos$ **do**
  $p_j \leftarrow \text{rquot}(v_I, 1 - \sigma^j(\beta)x)$
Resolver el sistema lineal $r_I = \sum_{j \in pos} e_j \sigma^j(\alpha) p_j$
$e \leftarrow \sum_{j \in pos} e_j x^j$
return $y - e$

BIBLIOGRAFÍA

[HP03] W. Hufiman and V. Pless. Fundamentals of error-correcting codes. 01 2003.

[Tur36] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.