Universidad de Oviedo
Departamento de Matemáticas

Doctorado en Matemáticas y Estadística

Descodificación de códigos grupo

Fabián Ricardo Molina Gómez

Oviedo, 2024

Universidad de Oviedo

RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: Descodificación de códigos grupo	Inglés: Decoding group codes

2.- Autor	
Nombre: Fabián Ricardo Molina Gómez	
Programa de Doctorado: Doctorado en Matemáticas y Estadística	
Órgano responsable: Centro Internacional de Postgrado	

RESUMEN (en español)

En esta tesis se diseñan algoritmos de descodificación para códigos grupo, centrando la atención en álgebras de grupo semisimples. Presentamos un algoritmo general de descodificación inspirado en el bien conocido algoritmo de descodificación por síndrome para códigos lineales y que utiliza la descomposición de un álgebra de grupo semisimple KG como suma directa de ideales biláteros minimales. También mostramos que, si G es abeliano, el algoritmo se puede modificar para hacerlo más simple y eficiente. Luego, utilizamos la descomposición de KG como suma de dos ideales biláteros, uno de ellos el código grupo, para diseñar dos algoritmos de descodificación. Uno de ellos generaliza el algoritmo de descodificación de Meggitt y el otro, mejora del algoritmo de descodificación general. El algoritmo de descodificación por permutación y su versión para códigos grupo también se explora en la tesis. Finalmente, definimos la noción de código grupo LDOI y presentamos la versión del algoritmo de descodificación *Bit Flipping con una única iteración* para su implementación en algunos códigos grupo.

RESUMEN (en Inglés)

In this work we design decoding algorithms for group codes. We focus on semisimple group algebras. We present a general decoding algorithm inspired by the well-known syndrome decoding algorithm for linear codes and that uses the decomposition of a semisimple group algebra KG as a direct sum of minimal two-sided ideals. In case of an abelian group G, the algorithm can be modified to be simpler and more efficient. Using the decomposition of a semisimple group algebra as the sum of an arbitrary two-sided ideal and its orthogonal ideal, two new decoding algorithms are presented. One of them generalizes the Meggitt decoding algorithm and the other one improves the general decoding algorithm. A permutation decoding algorithm for group codes is also explored in the thesis. Finally, we define the notion of LDOI group code and present a version of the one iteration Bit Flipping decoding algorithm to be implemented in some group codes.

SR. PRESIDENTE DE LA COMISIÓN ACADÉMICA DEL PROGRAMA DE DOCTORADO EN MATEMÁTICAS Y ESTADÍSTICA

Universidad de Oviedo
Departamento de Matemáticas

Doctorado en Matemáticas y Estadística

Descodificación de códigos grupo

Fabián Ricardo Molina Gómez

Directores:
Consuelo Martínez López y Alejandro Piñera Nicolás

Oviedo, 2024

Esta investigación ha sido financiada parcialmente por la beca de Formación del Personal Investigador (FPI) MCIU-19-PRE2018-085335 asociada al proyecto de investigación: "Estructuras Algebraicas y Aplicaciones. Codificación y Criptografía" (MTM 2017-83506-C2-2-P del Ministerio de Ciencia e Innovación y Universidades) y por el proyecto de investigación "Estructuras algebraicas y su relevante papel en la información" (PID2021-123461NB-C22 del Ministerio de Ciencia e Innovación).

Dedicatoria y agradecimientos

Dedico esta tesis a la memoria de mi madre Doris Gómez y de, quien me adoptó como su hijo, Joaquín Barreto. Les debo todo lo que soy.

Quiero agradecer a Dios por poner a las personas y bendiciones en los momentos indicados. Me gustaría expresar mi agradecimiento a la profesora Consuelo Martínez por darme la oportunidad de hacer este trabajo, por su orientación académica, y por su calidad humana. También, agradezco a mi esposa Jessica Vargas y a mi hijo Santiago Molina por su amor y el sacrificio que hicieron al acompañarme incondicionalmente en esta etapa de mi vida. Ofrezco un agradecimiento especial al profesor Santos González por su apoyo a nivel personal y profesional. Asimismo, gracias a Ignacio Fernández y a mi codirector Alejando Piñera por sus consejos y sugerencias. De igual manera, agradezco a todas aquellas personas de la Universidad de Oviedo que hicieron parte de este proceso. Finalmente, deseo agradecer al Ministerio de Ciencia, Innovación y Universidades del Gobierno de España por financiar mis estudios de doctorado, participación en congresos y estancias de investigación.

Resumen

En esta tesis se diseñan algoritmos de descodificación para códigos grupo, centrando la atención en álgebras de grupo semisimples. Presentamos un algoritmo general de descodificación inspirado en el bien conocido algoritmo de descodificación por síndrome para códigos lineales y que utiliza la descomposición de un álgebra de grupo semisimple $\mathbb{K}G$ como suma directa de ideales biláteros minimales. También mostramos que, si G es abeliano, el algoritmo se puede modificar para hacerlo más simple y eficiente. Luego, utilizamos la descomposición de $\mathbb{K}G$ como suma de dos ideales biláteros, uno de ellos el código grupo, para diseñar dos algoritmos de descodificación. Uno de ellos generaliza el algoritmo de descodificación de Meggitt y el otro, mejora del algoritmo de descodificación general. El algoritmo de descodificación por permutación y su versión para códigos grupo también se explora en la tesis. Finalmente, definimos la noción de código grupo LDOI y presentamos la versión del algoritmo de descodificación *Bit Flipping con una única iteración* para su implementación en algunos códigos grupo.

Abstract

In this work we design decoding algorithms for group codes. We focus on semisimple group algebras. We present a general decoding algorithm inspired by the well-known syndrome decoding algorithm for linear codes and that uses the decomposition of a semisimple group algebra $\mathbb{K}G$ as a direct sum of minimal two-sided ideals. In case of an abelian group G, the algorithm can be modified to be simpler and more efficient. Using the decomposition of a semisimple group algebra as the sum of an arbitrary two-sided ideal and its orthogonal ideal, two new decoding algorithms are presented. One of them generalizes the Meggitt decoding algorithm and the other one improves the general decoding algorithm. A permutation decoding algorithm for group codes is also explored in the thesis. Finally, we define the notion of LDOI group code and present a version of the *one iteration Bit Flipping* decoding algorithm to be implemented in some group codes.

Índice general

Introducción

Los códigos correctores de errores (en lo sucesivo usaremos códigos como sinónimo de códigos correctores de errores) son fundamentales para garantizar la fiabilidad de la información enviada a través de un canal con ruido. Para lograrlo, se necesitan algoritmos de descodificación que permitan corregir los errores que se producen durante la transmisión. Este proceso se realiza de manera eficiente con ciertos tipos de códigos existentes.

Por otro lado, es posible desarrollar sistemas criptográficos (criptosistemas) fundamentados en códigos que, en principio, son resistentes a la computación cuántica. De hecho, la criptografía basada en códigos se perfila como uno de los enfoques criptográficos post-cuánticos más prometedores. En este contexto, un "buen" código es crucial para el diseño de un criptosistema de clave pública. El primer esquema de este tipo fue propuesto en 1978 por R. J. McEliece ([45]). Esta propuesta utiliza códigos Goppa (clásicos) que se ocultan bajo un código lineal, aparentemente arbitrario, que es la clave pública. Para descifrar el mensaje, se necesita descodificarlo con un código lineal, que se sabe, es un problema complejo. La propuesta de McEliece sigue sin romperse hasta el día de hoy, y aunque los procesos de cifrado y descifrado son simples, el principal problema que presenta es el tamaño de la clave pública.

Se han planteado diversas propuestas para mitigar este problema, sugiriendo el reemplazo de los códigos Goppa por códigos más estructurados, como los códigos Reed-Solomon generalizados ([2, 4, 5, 50]) o códigos Reed-Muller binarios ([59]). Sin embargo, se ha demostrado que tales propuestas son vulnerables ante ciertos ataques ([12, 48, 58, 63, 64]). Otra alternativa para resolver el problema, se basa en el uso de variantes de los códigos Goppa ([20, 37, 51]). El principal inconveniente de esta alternativa radica en que, en algunas de las propuestas presentadas, es posible atacar el criptosistema al reducir los grados y el número de variables del sistema algebraico a resolver. De este modo, las nuevas propuestas han sido rotas parcial o totalmente (ver [13] y [19]).

Esta tesis se enmarca en el contexto de los códigos grupo, que pueden ser identificados con ideales biláteros de un álgebra de grupo $\mathbb{K}G$. Por tal razón, y asumiendo un orden fijo para los elementos en G, a veces escribiremos los elementos de $\mathbb{K}G$ como $n-$tuplas de coeficientes en el cuerpo $\mathbb{K}$. Formalmente, un $(n, k, d)-$código lineal $\mathfrak{C}$ sobre $\mathbb{K}$ es un $G-$código si existe un $\mathbb{K}-$isomorfismo $\vartheta : \mathbb{K}^n \to \mathbb{K}G$ tal que $\vartheta(\mathfrak{C})$ es un ideal bilátero de $\mathbb{K}G$. Los códigos lineales que son códigos grupo fueron caracterizados por Bernal et al. ([8]). Dicha caracterización se dio en términos del grupo de automorfismos permutación del código y, a pesar de su interés teórico, no es muy útil para aplicaciones. De manera similar, sustituyendo "ideal bilátero" por "ideal a izquierda" se puede hablar de códigos grupo a izquierda.

Se dice que un código lineal $\mathfrak{C}$, es código grupo si existe un grupo G tal que $\mathfrak{C}$ es un $G-$código. Además, un grupo código $\mathfrak{C}$ es abeliano si existe un grupo abeliano A tal que $\mathfrak{C}$ es un $A-$código. Bernal et al. también demostraron que si G es descomponible como producto de dos subgrupos abelianos, entonces todo $G-$código es código grupo abeliano. En una serie de trabajos posteriores (ver [24, 25, 26]) se aborda el estudio de condiciones suficientes sobre la longitud de un $G-$código para que sea código grupo abeliano, y se demuestra que si $\mathfrak{C}$ es un $G-$código que es código abeliano sobre $\mathbb{K}$, también, lo es sobre cualquier subcuerpo de $\mathbb{K}$. En [28], se prueba que todo $G-$código de dimensión menor o igual a 3, es código grupo abeliano.

Es importante tener en cuenta que un código grupo puede ser identificado como $G-$código para diversos grupos, y algunos pueden ser abelianos y otros no abelianos. En [25], usando el grupo $G = S_4$ y el cuerpo $\mathbb{K} = \mathbb{F}_5$, se construyen códigos grupo no abelianos, es decir, que no pueden ser identificados como $A-$códigos para ningún grupo abeliano A. En [26], se prueba la existencia de códigos grupo no abelianos en el caso no semisimple construyendo $G-$códigos sobre $\mathbb{K} = \mathbb{F}_2$ y $\mathbb{K} = \mathbb{F}_3$ y manteniendo $G = S_4$. En [27], se demuestra la existencia de códigos grupo no abelianos de longitud 24 sobre cualquier cuerpo finito.

En algunos casos, los códigos grupo no abelianos tienen peores parámetros que los construidos con grupos abelianos. A pesar de ello, en [26] se construyó un código grupo no abeliano de longitud 24, dimensión 12 y distancia mínima 6. Tal código es óptimo, en el sentido que cualquier código lineal con la misma longitud y dimensión tiene una distancia mínima menor o igual a 6 y, además, tal distancia mínima no se puede ser alcanzada con un código grupo abeliano. Este resultado justifica el interés de considerar códigos grupo no abelianos.

La dificultad para distinguir los códigos grupo entre códigos lineales y el hecho de poder utilizar grupos abelianos y no abelianos parecen, en principio, propiedades favorables para utilizar estos códigos en el diseño de un criptosistema de tipo McEliece. Sin embargo, también es esencial contar con algoritmos de descodificación eficientes.

Si bien hay algunas propuestas de algoritmos de descodificación para códigos grupo abelianos o códigos grupo a izquierda, no hemos encontrado en la literatura algoritmos de descodificación para códigos grupo. En [10], se propone un algoritmo de descodificación parcial por permutación para códigos grupo abelianos en el caso semisimple. En [18], los autores sugieren un algoritmo para códigos grupo a izquierda con capacidad correctora igual a 1. En [17] se realiza una descripción constructiva de las condiciones necesarias para la implementación de un algoritmo de descodificación por mayoría, también, para códigos grupo a izquierda. El objetivo central de esta tesis es el diseño de algoritmos de descodificación para códigos grupo.

Nos restringimos al caso semisimple dado que la propiedad principal que utilizamos en el diseño de los algoritmos presentados, es que para todo ideal bilátero I de $\mathbb{K}G$, existe un ideal bilátero I^+ tal que $\mathbb{K}G = I \oplus I^+$. Por lo tanto, $x \in I$ si y solo si $xy = 0$ para todo $y \in I^+$. Esto no se cumple para ideales a izquierda en cualquier caso, ni para ideales biláteros de álgebras de grupo no semisimples. Los algoritmos presentados en este trabajo incluyen ejemplos ilustrativos en los cuales se usó el software GAP ([65]) para hacer los cálculos.

Con el propósito de facilitar la lectura de la tesis, incluimos en el Capítulo 1, las definiciones fundamentales y principales resultados conocidos de códigos lineales, álgebras de grupo semisimples y códigos grupo.

En el Capítulo 2, diseñamos un algoritmo general de descodificación para códigos grupo. Este algoritmo está inspirado en el bien conocido algoritmo de descodificación por síndrome para códigos lineales y utiliza la descomposición de un álgebra de grupo semisimple $\mathbb{K}G$ como suma directa de ideales biláteros minimales. También mostramos que si G es abeliano, el algoritmo se puede modificar para hacerlo más simple y eficiente.

En el Capítulo 3, utilizamos la descomposición de $\mathbb{K}G$ como suma de dos ideales biláteros, uno de ellos el código grupo, para diseñar dos algoritmos de descodificación. El primero generaliza el algoritmo de descodificación de Meggitt para códigos cíclicos, mientras que el otro, es una mejora del algoritmo de descodificación expuesto en el capítulo anterior.

El objetivo del Capítulo 4 es considerar la descodificación por permutación en códigos grupo. Para ello usamos la descomposición semisimple del álgebra de grupo $\mathbb{K}G$ con el fin de obtener resultados que justifican su implementación. Además de esto, generalizamos el algoritmo de descodificación por permutación en códigos grupo.

En el Capítulo 5, definimos la familia de códigos grupo con idempotente ortogonal de peso muy pequeño (LDOI). Tales códigos tienen una fuerte conexión con los códigos LDPC (Low-Density Parity-Check). Luego, presentamos la versión del algoritmo de descodificación *Bit Flipping con una única iteración* para algunos códigos grupo LDOI. Terminamos este capítulo, exponiendo algunas condiciones para códigos grupo LDOI que aseguran que, en esos casos, se puede aplicar tal algoritmo. Este trabajo finaliza con las conclusiones en las que se resumen los objetivos cumplidos en la tesis y los resultados obtenidos en la misma.

Capítulo 1

Preliminares

Este capítulo tiene por objeto presentar conceptos y resultados preliminares que se utilizarán en capítulos posteriores. Esto se lleva a cabo en tres secciones. La primera está dedicada a exponer aspectos generales de la teoría de códigos lineales. La segunda, con un enfoque más algebraico, muestra las definiciones y propiedades de los anillos semisimples y las álgebras de grupo. Mientras que la tercera ilustra algunos resultados familiares en códigos grupo.

1.1. Códigos lineales

En esta sección, A denotará un conjunto finito no vacío con cardinal $|A| = q$. Las definiciones y resultados de esta sección, se pueden consultar en [29] y [40].

Definición 1.1.1. *Se dice que $\mathfrak{C}$ es un código de longitud n sobre el alfabeto A (o simplemente que $\mathfrak{C}$ es un código de A^n) si $\mathfrak{C}$ es un subconjunto del producto cartesiano A^n. En caso de que $q = 2$, se dice que $\mathfrak{C}$ es un código binario de longitud n.*

En este contexto, los elementos de A^n se denominan *palabras* y si $x = (x_1, \ldots, x_n)$ es una palabra de A^n, entonces x_i se denomina *símbolo* de x en *la $i-$ésima posición*. Los elementos de un código se denominan *palabras código*. Ahora presentamos la definición de dos parámetros muy importantes.

Definición 1.1.2. *Dado $\mathfrak{C}$ un código de A^n, la dimensión combinatoria de $\mathfrak{C}$ es el número k dado por*

$$k := \log_q(|\mathfrak{C}|).$$

Definición 1.1.3. *Sean $x = (x_1, \ldots, x_n)$ e $y = (y_1, \ldots, y_n)$ dos palabras de A^n. La distancia de Hamming entre x e y, denotada como $d_H(x, y)$, se define como el cardinal*

$$d_H(x, y) := |\{i \in \{1, \ldots, n\} : x_i \neq y_i\}|.$$

La distancia mínima de un código $\mathfrak{C}$ de A^n, denotada por d, se define por

$$d := \text{mín}\{d_H(x, y) : x, y \in \mathfrak{C}, x \neq y\}.$$

Si $\mathfrak{C}$ tiene dimensión combinatoria k y distancia mínima d, se dice que $\mathfrak{C}$ es un $(n, k, d)-$código sobre (el alfabeto) A.

Ahora supongamos que se transmite un mensaje a través de un canal (virtual), el cual se puede presentar como un elemento de un código $\mathfrak{C}$. Si durante el proceso de transmisión de dicho mensaje se producen algunas modificaciones en sus símbolos, tales modificaciones se denominan *errores*, y el proceso de recuperación del mensaje original se denomina *descodificación*. Para llevar a cabo tal proceso es necesario detectar y corregir los errores producidos durante la transmisión.

En este punto, supondremos que la probabilidad de que uno de los símbolos de la palabra código enviada sea modificada es igual para cualquiera de estos. También vamos a considerar que la probabilidad de recibir un símbolo incorrecto es menor a $1/2$. Por lo tanto, si $r \in A^n$ es la palabra recibida, la descodificación se hace buscando la palabra código $c \in \mathfrak{C}$ con menor distancia $d_H(r, c)$. Este proceso se denomina *descodificación por distancia mínima*.

Una forma de descodificar por distancia mínima, consiste en tomar cada palabra $x \in A^n$ y calcular la distancia entre x y todas las palabras código de $\mathfrak{C}$, de manera que se pueda determinar cuál es la palabra código $c = c(x)$ con la menor distancia (de Hamming) a x. Con esto, se elabora una lista con los elementos $x \in A^n$ y su correspondiente palabra código $c(x)$ "más cercana". Esta lista se construye antes de empezar el proceso de descodificación y tiene q^n entradas. Sin embargo, la generación de tal lista puede ser un proceso computacionalmente costoso, especialmente para códigos con longitudes no demasiado pequeñas. A lo largo de este trabajo iremos mostrando otros métodos de descodificación equivalentes al anterior, pero más eficientes.

Adicionalmente, el mayor número de errores que pueden ser corregidos por un código $\mathfrak{C}$ se denomina *capacidad correctora* de $\mathfrak{C}$.

Teorema 1.1.4. *Si $\mathfrak{C}$ es un código sobre el alfabeto A con distancia mínima d, entonces su capacidad correctora es*

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Dados n y d números enteros positivos, denotamos por $M_q(n, d)$ el número máximo de palabras que puede tener un código de A^n con distancia mínima

d. En la literatura se pueden encontrar muchas cotas superiores para el valor de $M_q(n,d)$ (ver Capítulo 2 de [29]). Una de las más conocida es la dada por Singleton, que da origen a los códigos MDS.

Teorema 1.1.5 (Cota de Singleton)**.** *Si $\mathfrak{C}$ es un código de A^n con distancia mínima d, entonces*

$$M_q(n,d) \leq q^{n-d+1}.$$

Definición 1.1.6. *Un código $\mathfrak{C}$ de A^n con distancia mínima d es MDS si su cardinal es $|\mathfrak{C}| = q^{n-d+1}$, o equivalentemente, si su dimensión combinatoria es igual a $n-d+1$.*

Es decir, el número de palabras de cualquier código de A^n con distancia mínima d, es menor o igual al número de palabras de un código MDS con los mismos parámetros.

Ahora vamos a introducir la noción de equivalencia (por permutación) entre códigos de A^n. Para ello, denotamos a S_n como el grupo simétrico de grado n. Notemos que S_n actúa sobre A^n mediante

$$\sigma(x_1,\ldots,x_n) := (x_{\sigma(1)},\ldots,x_{\sigma(n)}),$$

donde $\sigma \in S_n$ y $(x_1,\ldots,x_n) \in A^n$. Se dice que dos códigos $\mathfrak{C}$ y $\mathfrak{C}^*$ de A^n son *equivalentes* si existe $\sigma \in S_n$ tal que $\mathfrak{C}^* = \sigma(\mathfrak{C})$. Con esto, si $\mathfrak{C}$ es un código de A^n, entonces el conjunto dado por

$$\mathrm{PAut}(\mathfrak{C}) := \{\sigma \in S_n : \sigma(\mathfrak{C}) = \mathfrak{C}\},$$

es un subgrupo de S_n y se denomina el *grupo de automorfismos permutación de* $\mathfrak{C}$. Aunque no es sencillo de calcular, más adelante veremos cómo se puede usar para distinguir un código grupo entre códigos lineales de igual longitud.

Definición 1.1.7. *Sean $\mathbb{K}$ un cuerpo finito y $\mathfrak{C}$ un código de $\mathbb{K}^n$. Se dice que $\mathfrak{C}$ es un código lineal de dimensión k (o simplemente $(n,k)-$código lineal sobre $\mathbb{K}$) si $\mathfrak{C}$ es un subespacio vectorial de dimensión k de $\mathbb{K}^n$.*

En lo que resta de la sección $\mathbb{K}$ será un cuerpo finito con q elementos. Si $\mathfrak{C}$ es un $(n,k)-$código lineal sobre $\mathbb{K}$, entonces la dimensión de $\mathfrak{C}$ como subespacio vectorial coincide con su dimensión combinatoria. Denotaremos por $\mathrm{Supp}(x)$ al soporte de $x \in \mathbb{K}^n$, es decir,

$$\mathrm{Supp}(x) := \{i \in \{1,\ldots,n\} : x_i \neq 0\}.$$

Se define *el peso de x* como el cardinal

$$\mathrm{wt}(x) := |\mathrm{Supp}(x)|,$$

y, por lo tanto, la distancia $d_H(x, y)$ entre dos *palabras* $x, y \in \mathbb{K}^n$ coincide con $\mathrm{wt}(x-y)$. Si d es la distancia mínima de $\mathfrak{C}$, se dice que $\mathfrak{C}$ es un $(n, k, d)-$*código lineal sobre* $\mathbb{K}$,

$$d = \mín\{\mathrm{wt}(x) : x \in \mathfrak{C}, x \neq 0\},$$

y la cota de Singleton puede expresarse como

$$d \leq n - k + 1.$$

Definición 1.1.8. *Dado* $\mathfrak{C}$ *un* $(n, k)-$*código lineal sobre* $\mathbb{K}$*, se dice que* $\mathcal{G} \in M_{k\times n}(\mathbb{K})$ *es una matriz generadora de* $\mathfrak{C}$ *si sus filas forman una base de* $\mathfrak{C}$ *sobre* $\mathbb{K}$*.*

Definición 1.1.9. *Dado* $\mathfrak{C}$ *un* $(n, k)-$*código lineal sobre* $\mathbb{K}$*, se dice que*

$$I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$$

es un conjunto de información de $\mathfrak{C}$ *si al proyectar* $\mathfrak{C}$ *en las posiciones* $i_1, \ldots, i_k$*, se obtiene un espacio vectorial de dimensión* k*. En tal caso, los elementos de* I *se denominan posiciones de información de* $\mathfrak{C}$ *y el complemento* $I' = \{1, \ldots, n\}\backslash I$ *de* I *se denomina conjunto de posiciones de control de* $\mathfrak{C}$*. Además, si* $x = (x_1, \ldots, x_n) \in \mathbb{K}^n$*, entonces para cada* $i \in I$*,* x_i *se denomina símbolo de información de* x*.*

De la definición anterior, si $\mathcal{G}$ es una matriz generadora de $\mathfrak{C}$, entonces todo conjunto de k posiciones que se corresponda con columnas $\mathbb{K}-$linealmente independientes de $\mathcal{G}$, es un conjunto de información de $\mathfrak{C}$ y viceversa.

Ahora, consideremos "$\cdot$" el producto habitual de matrices. Si $x = (x_1, \ldots, x_n)$ e $y = (y_1, \ldots, y_n)$ son elementos de $\mathbb{K}^n$, tenemos

$$x \cdot y^T := \sum_{i=1}^{n} x_i y_i.$$

Definición 1.1.10. *Dado* $\mathfrak{C}$ *un* $(n, k)-$*código lineal sobre* $\mathbb{K}$*, el código dual de* $\mathfrak{C}$*, denotado por* $\mathfrak{C}^\perp$*, se define como el subespacio de* $\mathbb{K}^n$ *dado por*

$$\mathfrak{C}^\perp := \{x \in \mathbb{K}^n : x \cdot y^T = 0 \text{ para todo } y \in \mathfrak{C}\}.$$

De la definición anterior, si $\mathfrak{C}$ tiene dimensión k, entonces $\mathfrak{C}^\perp$ es un código lineal de $\mathbb{K}^n$ con dimensión $n - k$. Además, si $\mathcal{H} \in M_{(n-k)\times n}(\mathbb{K})$ es una matriz generadora de $\mathfrak{C}^\perp$, entonces

$$\mathfrak{C} = \{x \in \mathbb{K}^n : \mathcal{H} \cdot x^T = 0\}.$$

Cualquier matriz generadora de $\mathfrak{C}^{\perp}$ se denomina *matriz de control* de $\mathfrak{C}$. Adicionalmente, $(\mathfrak{C}^{\perp})^{\perp} = \mathfrak{C}$ y, por lo tanto, para toda matriz generadora $\mathcal{G}$ de $\mathfrak{C}$ tenemos que

$$\mathfrak{C}^{\perp} = \{x \in \mathbb{K}^n : x \cdot \mathcal{G}^T = 0\}.$$

De lo anterior, todo conjunto de información de $\mathfrak{C}$ es un conjunto de posiciones de control de $\mathfrak{C}^{\perp}$ y viceversa. Otras propiedades útiles que se pueden extraer de la matriz de control de un código lineal, se presentan en el siguiente resultado.

Proposición 1.1.11. *Si $\mathfrak{C}$ es un $(n,k)-$código lineal sobre $\mathbb{K}$ y $\mathcal{H}$ es una matriz de control de $\mathfrak{C}$, entonces $\mathcal{H}$ tiene rango $n-k$ y la distancia mínima de $\mathfrak{C}$ es igual al menor número de columnas de $\mathcal{H}$ que son $\mathbb{K}-$linealmente dependientes.*

Describiremos a continuación el algoritmo de descodificación por distancia mínima para códigos lineales. Para ello, vamos a denotar a $c \in \mathfrak{C}$ como la palabra enviada y a $r \in \mathbb{K}^n$ como la palabra recibida. Una forma de establecer los errores producidos durante el proceso de transmisión, es hallando el vector $e \in \mathbb{K}^n$ tal que $r = c + e$. Si $\mathfrak{C}$ tiene capacidad correctora t y suponemos que el número de errores producidos es menor o igual a t, entonces $\mathrm{wt}(e) \leq t$ y c es la palabra código con menor distancia a r. Así, e tiene peso mínimo y descodificar consiste en encontrar la palabra $e \in \mathbb{K}^n$ de peso mínimo tal que $r - e \in \mathfrak{C}$. La siguiente noción nos permite hacerlo siempre que $\mathrm{wt}(e) \leq t$.

Definición 1.1.12. *Dados $\mathfrak{C}$ un $(n,k)-$código lineal sobre $\mathbb{K}$ y $\mathcal{H}$ una matriz de control de $\mathfrak{C}$, el síndrome $x \in \mathbb{K}^n$, denotado por Syn(x), se define como*

$$Syn(x) := \mathcal{H} \cdot x^T.$$

De la definición anterior, tenemos que $x \in \mathfrak{C}$ si y solo si $\mathrm{Syn}(x) = 0$. Por lo tanto, si $r = c + e$ donde $c \in \mathfrak{C}$, entonces $\mathrm{Syn}(r) = \mathrm{Syn}(e)$. En consecuencia, la descodificación por síndrome equivale a descodificar por distancia mínima y consiste en encontrar una palabra e de peso mínimo cuyo síndrome sea igual al síndrome de r. Para abordar este problema es necesario señalar que, para todo par $x, y \in \mathbb{K}^n$, se tiene que $x + \mathfrak{C} = y + \mathfrak{C}$ si y solo si $\mathrm{Syn}(x) = \mathrm{Syn}(y)$. Es decir, podemos considerar las clases módulo $\mathfrak{C}$ y dentro de cada una de ellas, escoger un elemento de peso mínimo. Es muy fácil probar que si dicho elemento tiene peso menor o igual a t, entonces es el único elemento dentro de cada clase que tiene tal propiedad. El elemento de cada clase con peso menor o igual a t se denominan *líder de clase* y así, con la notación anterior, el problema de descodificar r, se traduce a encontrar el líder de la clase $r + \mathfrak{C}$.

Una manera de llevar a cabo este tipo de descodificación, consiste en la elaboración de una lista en la cual se recojan los líderes de clase con su respectivo

síndrome. La lista se construye antes de empezar el proceso. Una vez que se recibe la palabra $r \in \mathbb{K}^n$, se calcula su síndrome y se verifica si es el síndrome de alguno de los líderes de clase de la lista. El líder de clase cuyo síndrome es igual al síndrome de r es el vector que indica el error. Sin embargo, si el síndrome de r no se encuentra en la lista, no es posible descodificar r, puesto que, el número de errores ocurridos durante la transmisión excede la capacidad correctora del código.

Esta lista tiene a lo sumo q^{n-k} entradas y aunque es más pequeña que la mencionada para códigos (no necesariamente lineales), de nuevo, tiene la desventaja que para un valor moderado de n, construirla puede resultar bastante complejo. Por lo que, el método de descodificación por síndrome no siempre es eficiente. En la actualidad, se conocen varios tipos de códigos con algoritmos de descodificación equivalentes, pero mucho mas eficientes al anterior. Estos códigos se definen usando diferentes estructuras algebraicas (grupos, módulos, anillos, etc.) y permiten utilizar sus respectivas propiedades para obtener tales algoritmos de descodificación. Un ejemplo de códigos lineales con buenas propiedades para descodificación son los códigos cíclicos. Estos códigos fueron introducidos por E. Prange en [53] y [54], y posteriormente estudiados en [52].

Definición 1.1.13. *Un $(n,k)-$código lineal $\mathfrak{C}$ sobre $\mathbb{K}$ es cíclico si cualquier palabra $(c_1, c_2, \ldots, c_n)$ satisface:*

$$(c_1, \ldots, c_{n-1}, c_n) \in \mathfrak{C} \text{ si y solo si } (c_n, c_1, \ldots, c_{n-1}) \in \mathfrak{C}.$$

Se sabe, que un código cíclico de longitud n se puede identificar con un ideal del anillo $\mathbb{K}[X]/\langle X^n - 1\rangle$ y que está generado por un único polinomio mónico divisor de $X^n - 1$ (ver [29] y [40]). Por tanto, en este contexto es pertinente denotar las posiciones entre 0 y $n-1$, en lugar de 1 y n. Con esto, la palabra $a = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{K}^n$ se identifica con el polinomio $a(X) = a_0 + a_1X + \cdots + a_{n-1}X^{n-1}$ y todas las operaciones se hacen módulo $\langle X^n - 1\rangle$.

También se puede utilizar el algoritmo de la división para redefinir la noción de síndrome. En efecto, si $\mathfrak{C}$ es un código cíclico de $\mathbb{K}^n$ generado por $g(X)$, se define el *R−síndrome* de la palabra $a(X) \in \mathbb{K}[X]/\langle X^n - 1\rangle$, denotado por R$-Syn(a(X))$, como el resto de dividir $a(X)$ por $g(X)$. Con esto, se tiene que $a(X) \in \mathfrak{C}$ si y solo si R$-$Syn$(a(X)) = 0$. De igual forma, si $r(X) = c(X)+e(X)$, donde $c(X) \in \mathfrak{C}$, entonces R$-$Syn$(r(X))$ = R$-$Syn$(e(X))$. Además, dados $a(X), b(X) \in \mathbb{K}[X]/\langle X^n - 1\rangle$ distintos y con peso menor o igual a t, tenemos que R$-$Syn$(a(X)) \neq$ R$-$Syn$(b(X))$ y la descodificación por R$-$síndrome equivale a descodificar por síndrome.

Para descodificar en códigos cíclicos usando la noción de R−síndrome, es necesario elaborar la lista con los correspondientes líderes de clase y sus respectivos R−síndromes. Sin embargo, se puede usar la estructura algebraica de estos códigos para reducir la lista de líderes de clase. En efecto, si $\mathfrak{C}$ es un código cíclico de $\mathbb{K}^n$ y $\mathbf{c}(X) \in \mathfrak{C}$, entonces $X^m\mathbf{c}(X) \in \mathfrak{C}$ para todo $m \in \{0, 1, \ldots, n-1\}$. Con lo cual, si $\mathbf{r}(X) = \mathbf{c}(X)+\mathbf{e}(X)$, entonces $X^m\mathbf{r}(X) = X^m\mathbf{c}(X)+X^m\mathbf{e}(X)$ y, en consecuencia, R−Syn$(X^m\mathbf{r}(X))$ = R−Syn$(X^m\mathbf{e}(X))$. Esto implica que la lista de líderes de clase con un peso menor o igual a t se puede reducir a aquellos elementos cuyos soportes contienen una posición específica $i_0 \in \{0, 1, \ldots, n-1\}$.

Esta nueva lista se denomina *lista reducida de R−síndromes* y sus elementos *representantes*. Si el número de errores ocurridos es a lo sumo t, entonces existe $m \in \{0, 1, \ldots, n-1\}$ tal que el conjunto de posiciones de $X^m\mathbf{r}(X)$, donde ocurrieron los errores, contiene la posición i_0 y, por lo tanto, R−Syn$(X^m\mathbf{r}(X))$ debe estar en la lista reducida de R−síndromes.

Con lo anterior, el algoritmo de descodificación queda descrito de la siguiente manera: recibida la palabra $r(X)$, se van calculando los R−síndromes de $r(X), Xr(X), \ldots, X^nr(X)$ hasta que uno de estos aparezca en la lista reducida de R−síndromes. Si R−Syn$(X^mr(X))$ es R−síndrome de un representante $e'(X)$ en tal lista, entonces el error es $e(X) = X^{n-m}e'(X)$ y el algoritmo termina. En caso de que ninguno de los R−síndromes de $r(X), Xr(X), \ldots, X^nr(X)$ aparezcan en la lista, el número de errores producidos excede la capacidad correctora y así no se puede descodificar $r(X)$. Este método de descodificación fue introducido por Meggitt en [46] y [47]. Veamos un ejemplo.

Ejemplo 1.1.14 ([29])**.** *Consideremos el* $(15, 7, 5)$*−código cíclico* $\mathfrak{C}$ *sobre* $\mathbb{K} = \mathbb{F}_2$ *generado por* $g(X) = 1 + X^4 + X^6 + X^7 + X^8$*. La capacidad correctora en este caso es* $t = 2$ *y todas las operaciones las realizamos módulo* $\langle X^{15} - 1\rangle$*. Si fijamos la posición* $i_0 = 14$*, la lista reducida de R−síndromes se presenta en la Tabla I.*

Si $r(X) = 1 + X^4 + X^7 + X^9 + X^{10} + X^{12}$*, vemos que*

$$R{-}Syn(r(X)) = X + X^2 + X^6 + X^7,$$

y

$$R{-}Syn(Xr(X)) = 1 + X^2 + X^3 + X^4 + X^6$$

no son R−síndromes de la lista anterior. Sin embargo,

$$R{-}Syn(X^2r(X)) = X + X^3 + X^4 + X^5 + X^7,$$

tiene como representante a $e'(X) = X^4 + X^{14}$. *Por tanto, el error es*

$$e(X) = X^{13}(X^4 + X^{14}) = X^2 + X^{12}$$

y, en consecuencia,

$$c(X) = 1 + X^2 + X^4 + X^7 + X^9 + X^{10}$$

es la palabra código enviada. La lista con líderes de clase tendría alrededor de $2^8 = 256$ *entradas, mientras que la lista reducida de* $R-$*síndromes solo tiene 15. Esto último nos permite ver que el algoritmo de descodificación de Meggitt es más eficiente que la descodificación por síndrome.*

Tabla I: Lista reducida de $R-$*síndromes del Ejemplo 1.1.14.*

Representante	$R-$*síndrome*
X^{14}	X^7
$1 + X^{14}$	$1 + X^4 + X^6$
$X + X^{14}$	$1 + X + X^4 + X^5 + X^6 + X^7$
$X^2 + X^{14}$	$1 + X^2 + X^5 + X^6$
$X^3 + X^{14}$	$1 + X^2 + X^3 + X^4 + X^7$
$X^4 + X^{14}$	$X + X^3 + X^4 + X^5 + X^7$
$X^5 + X^{14}$	$X^2 + X^3 + X^5 + X^6 + X^7$
$X^6 + X^{14}$	$X^3 + X^5 + X^6$
$X^7 + X^{14}$	$1 + X^7$
$X^8 + X^{14}$	$X + X^7$
$X^9 + X^{14}$	$X^2 + X^7$
$X^{10} + X^{14}$	$X^3 + X^7$
$X^{11} + X^{14}$	$X^4 + X^7$
$X^{12} + X^{14}$	$X^5 + X^7$
$X^{13} + X^{14}$	$X^6 + X^7$

1.2. Anillos semisimples y álgebras de grupo

Las definiciones y resultados que presentaremos a continuación pueden ser consultados en [14].

Un $R-$módulo (a izquierda) M es *irreducible* si $RM \neq 0$ y los únicos submódulos que tiene son los triviales, es decir, M y $\{0\}$. Se dice que M es *noetheriano (a izquierda)* si toda colección no vacía de submódulos a izquierda de M tiene elemento maximal. Notemos que, por ejemplo, todo espacio vectorial de dimensión finita es un módulo noetheriano.

Un $R-$módulo M es *completamente reducible* si para todo $N \leq_R M$, existe $N' \leq_R M$ tal que $M = N \oplus N'$. Es bién sabido que M es completamente reducible si y solo si es suma directa interna de submódulos irreducibles.

Como todo anillo R es un $R-$módulo, entonces los $R-$submódulos de R corresponden a los ideales a izquierda de R y, por tanto, un ideal a izquierda I de R es irreducible si y solo si los únicos ideales a izquierda que contiene son I y $\{0\}$, es decir, si y solo si es minimal.

Definición 1.2.1. *Un anillo R con elemento unidad, denotado por 1_R, es* noetheriano *si es noetheriano como $R-$módulo.*

Se puede probar que en un anillo noetheriano, todo ideal a izquierda minimal no-nilpotente está generado por un elemento idempotente, y a su vez es sumando directo de todo ideal a izquierda que lo contiene.

Definición 1.2.2. *Un anillo R es semisimple si es noetheriano y completamente reducible como $R-$módulo.*

Dado un anillo R, se dice que el par $x, y \in R$ son *ortogonales* si $xy = 0$. Un elemento $z \in R$ es *central* si conmuta con todo elemento de R. Un elemento $e \in R$ es *idempotente* si $e^2 = e$ y un idempotente $e \in R$ es *primitivo* si no puede ser escrito como suma de dos idempotentes ortogonales no nulos.

Si R es un anillo semisimple, podemos escribir

$$R = L_1 \oplus \cdots \oplus L_s,$$

donde L_j es un ideal a izquierda minimal de R para todo $j \in \{1, \ldots, r\}$.

Además, si $1_R = e_1 + \cdots + e_s$, con $e_j \in L_j$, entonces L_j está generado por el elemento e_j. Por la estructura semisimple de R, tenemos que $e_i e_j = 0$ para todo $i \neq j$ y $e_1, \ldots, e_s$ son idempotentes de R.

Como un álgebra sobre un cuerpo es *semisimple* si lo es como anillo, entonces los resultados que mostraremos para anillos semisimples también son válidos para álgebras semisimples.

Proposición 1.2.3. *Sean R un anillo semisimple y L un ideal a izquierda minimal de R. Entonces la suma B_L de todos los ideales a izquierda minimales de R isomorfos a L es un anillo simple, es decir, el anillo B_L es noetheriano y sus únicos ideales biláteros son los triviales. Además, B_L está generado por un elemento idempotente central primitivo.*

De la primera parte de la proposición anterior, se obtiene que un anillo semisimple R es suma directa interna de todos los anillos B_L.

Definición 1.2.4. *Sean R un anillo semisimple y L un ideal a izquierda minimal de R. El anillo B_L que se obtiene a partir de la suma de todos los ideales a izquierda minimales isomorfos a L, se denomina componente simple de R asociada a L.*

Así, el número de componentes simples en un anillo semisimple R es igual al número de ideales a izquierda minimales no isomorfos. En virtud de la segunda parte de la proposición anterior, se deduce que las componentes simples son ideales biláteros de R. Sin embargo, no todo ideal bilátero de R es una componente simple.

Proposición 1.2.5. *Todo ideal bilátero de un anillo semisimple, es suma de cierto número de componentes simples y está generado por un elemento idempotente central.*

El estudio de las nociones anteriores en el contexto de las álgebras de grupo, resulta interesante e importante para este trabajo.

Definición 1.2.6. *Sean $\mathbb{K}$ un cuerpo y $G = \{g_1 = 1_G, \ldots, g_n\}$ un grupo finito (multiplicativo). El álgebra de grupo de G sobre $\mathbb{K}$, denotada por $\mathbb{K}G$, es el conjunto de todas las combinaciones $\mathbb{K}-$lineales de la forma*

$$\mathbf{z} = \sum_{i=1}^{n} \alpha_i g_i, \tag{1.1}$$

junto con las operaciones dadas por

$$\left(\sum_{i=1}^{n} \alpha_i g_i\right) + \left(\sum_{j=1}^{n} \beta_j g_j\right) := \sum_{l=1}^{n} (\alpha_l + \beta_l) g_l,$$

$$\left(\sum_{i=1}^{n} \alpha_i g_i\right)\left(\sum_{j=1}^{n} \beta_j g_j\right) := \sum_{l=1}^{n} \left(\sum_{g_l = g_i g_j} \alpha_i \beta_j\right) g_l.$$

El álgebra de grupo $\mathbb{K}G$ tiene estructura de anillo con las operaciones anteriormente mencionadas y de espacio vectorial sobre $\mathbb{K}$ con el producto por escalares dado por

$$\alpha \sum_{i=1}^{n} \alpha_i g_i := \sum_{i=1}^{n} (\alpha \alpha_i) g_i, \qquad \alpha \in \mathbb{K}.$$

Observemos que el elemento $g_i \in G$ se identifica con $1_{\mathbb{K}} g_i$ y, por lo tanto, $G \subseteq \mathbb{K}G$. En consecuencia, los elementos de G forman una base finita de $\mathbb{K}G$ sobre $\mathbb{K}$ y así, $\mathbb{K}G$ es un anillo noetheriano.

En el resto de la sección conservaremos la notación para $\mathbb{K}$ y G de la definición anterior.

Definición 1.2.7. *Dado* $\mathbf{z} = \sum_{i=1}^{n} \alpha_i g_i \in \mathbb{K}G$, *se define el soporte de* $\mathbf{z}$ *como el conjunto*

$$Supp(\mathbf{z}) := \{g_i \in G : \alpha_i \neq 0\}.$$

El peso de $\mathbf{z}$, *denotado por* $wt(\mathbf{z})$, *se define como el cardinal de* $Supp(\mathbf{z})$. *Es decir,*

$$wt(\mathbf{z}) := |Supp(\mathbf{z})|.$$

Otras nociones de mucho interés y que se relacionan con el producto de $\mathbb{K}G$ son las siguientes.

Definición 1.2.8. *Dados dos elementos* $\mathbf{z}_1 = \sum_{i=1}^{n} \alpha_i g_i$ *y* $\mathbf{z}_2 = \sum_{j=1}^{n} \beta_j g_j$ *de* $\mathbb{K}G$, *el producto escalar* $\cdot$ *de* $\mathbf{z}_1$ *y* $\mathbf{z}_2$ *se define como*

$$\mathbf{z}_1 \cdot \mathbf{z}_2 := \sum_{l=1}^{n} \alpha_l \beta_l.$$

Además, dado un subconjunto $M \neq \emptyset$ *de* $\mathbb{K}G$ *se define*

$$M^{\perp} := \{\mathbf{z} \in \mathbb{K}G : \mathbf{z} \cdot \mathbf{z}' = 0 \text{ para todo } \mathbf{z}' \in M\}. \tag{1.2}$$

Definición 1.2.9. *Se define la aplicación* $\varphi : \mathbb{K}G \to \mathbb{K}G$ *mediante*

$$\varphi\left(\sum_{i=1}^{n} \alpha_i g_i\right) := \sum_{i=1}^{n} \alpha_i g_i^{-1}.$$

Notemos que el producto $\cdot$ es una aplicación bilineal sobre $\mathbb{K}G$ y φ es un anti-automorfismo de orden 2. Los resultados que se recogen a continuación permiten relacionar ambos conceptos (ver [61]). Recordemos que, si R es un anillo e I es un ideal a izquierda de R, el conjunto

$$\mathrm{Ann}_r(I) := \{y \in R : xy = 0 \text{ para todo } x \in I\}$$

es un ideal a derecha de R denominado *el anulador a derecha de* I.

Proposición 1.2.10 ([61])**.** *Si* $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{K}G$, *entonces*

$$\mathbf{z}_1 \cdot \varphi(\mathbf{z}_2) = 1_{\mathbb{F}G} \cdot (\mathbf{z}_1 \mathbf{z}_2).$$

Proposición 1.2.11 ([61]). *Si I es un ideal a izquierda $\mathbb{K}G$, entonces*

$$\varphi(I^{\perp}) = Ann_r(I).$$

La proposición anterior también se cumple si se intercambia la palabra "izquierda" por "derecha" y viceversa. Además, indica que si I es un ideal a izquierda (derecha, bilátero) de $\mathbb{K}G$, entonces $I^{\perp}$ también será un ideal a izquierda (derecha, bilátero) de $\mathbb{K}G$. A continuación, presentamos un resultado bien conocido en el estudio de álgebras de grupo.

Teorema 1.2.12 (Teorema de Maschke). *Sean $\mathbb{K}$ un cuerpo y G un grupo finito. Si la característica de $\mathbb{K}$ no divide al orden de G, entonces el álgebra de grupo $\mathbb{K}G$ es semisimple.*

Es posible conocer el número de componentes simples de $\mathbb{K}G$ es algunos casos concretos, como podemos apreciar en el siguiente resultado.

Teorema 1.2.13 ([14]). *Sean G un grupo finito y $\mathbb{K}$ un cuerpo algebraicamente cerrado cuya característica no divide al orden de G. Entonces el número de ideales a izquierda minimales de $\mathbb{K}G$ y, por tanto, el número de componentes simples de $\mathbb{K}G$ es igual al número de clases de conjugación de G.*

También es posible usar teoría de representaciones y teoría de caracteres para calcular los idempotentes centrales primitivos de algunas álgebras de grupo semisimples. Para ello, dado un grupo finito G, se dice que un cuerpo $\mathbb{K}$ *descompone a* G si el álgebra de grupo $\mathbb{K}G$ se puede escribir de la forma

$$\mathbb{K}G \cong M_{n_1}(\mathbb{K}) \oplus \cdots \oplus M_{n_s}(\mathbb{K}).$$

Teorema 1.2.14 ([14]). *Sean G un grupo finito y $\mathbb{K}$ un cuerpo de característica cero que descompone a G. Si L es un ideal a izquierda minimal de $\mathbb{K}G$ y χ es el caracter (irreducible) asociado a la $\mathbb{K}-$representación de G en L, entonces la componente simple de $\mathbb{K}G$ asociada a L está generada por el idempotente central primitivo:*

$$e = \frac{\chi(1)}{|G|} \sum_{i=1}^{n} \chi(g_i^{-1}) g_i.$$

Además, si $\mathbb{K}$ es algebraicamente cerrado, entonces dicha componente tiene dimensión igual a $\chi(1)^2$.

En el caso de cuerpos con característica distinta de cero, existen programas de software, como GAP ([65]), que son ampliamente conocidos y permiten calcular los idempotentes centrales primitivos para una gran cantidad de grupos finitos.

1.3. Códigos grupo

De aquí en adelante, $\mathbb{K}$ es un cuerpo finito, $G = \{g_1 = 1_G, \ldots, g_n\}$ es un grupo finito y E es la base canónica del $\mathbb{K}-$espacio vectorial $\mathbb{K}^n$.

Definición 1.3.1. *Un $(n,k)-$código lineal $\mathfrak{C}$ sobre $\mathbb{K}$ es un $G-$código si existe una biyección $\vartheta : E \to G$ tal que su extensión por linealidad a un $\mathbb{K}-$isomorfismo $\vartheta : \mathbb{K}^n \to \mathbb{K}G$, satisface que $\vartheta(\mathfrak{C})$ es un ideal bilátero de $\mathbb{K}G$.*

De manera análoga se definen los $G-$códigos a izquierda y los $G-$códigos a derecha sustituyendo "ideal bilátero" por "ideal a izquierda" e "ideal a derecha" respectivamente. Notemos que si $\mathfrak{C}$ es $G-$código a izquierda, entonces es $G-$código a derecha. En efecto, si tomamos la restricción φ a G de la aplicación dada por la Definición 1.2.8, entonces $\varphi \circ \vartheta$ es una biyección de E en G y $(\varphi \circ \vartheta)(\mathfrak{C})$ es un ideal a derecha de $\mathbb{K}G$. Lo anterior no quiere decir que si $\mathfrak{C}$ es $G-$código a izquierda, entonces $\mathfrak{C}$ es $G-$código. Por ejemplo, el álgebra de grupo $\mathbb{F}_5 A_4$ tiene un ideal a izquierda de dimensión 4, pero todos sus ideales biláteros tienen dimensiones menores a 4 o mayores a 8 ([7]).

Definición 1.3.2. *Un código lineal $\mathfrak{C}$ es código grupo (a izquierda) si existe un grupo G tal que $\mathfrak{C}$ es $G-$código (a izquierda).*

El ejemplo más conocido de códigos grupo son los códigos cíclicos. Es decir, cualquier código cíclico $\mathfrak{C}$ se puede ver como un código grupo. En efecto, el anillo $\mathbb{K}[X]/\langle X^n - 1\rangle$ es isomorfo a $\mathbb{K}G$ siendo G un grupo cíclico de orden n y como mencionamos anteriormente, si $\mathfrak{C}$ es un código cíclico de $\mathbb{K}^n$, entonces $\mathfrak{C}$ se puede identificar con un ideal del anillo $\mathbb{K}[X]/\langle X^n - 1\rangle$. Recíprocamente, si $G = \langle g\rangle$ es un grupo cíclico de orden n y $\mathfrak{C}$ es un $G-$código, entonces existe un $\mathbb{K}-$isomorfismo $\vartheta : \mathbb{K}^n \to \mathbb{K}G$ tal que $\vartheta(\mathfrak{C})$ es un ideal de $\mathbb{K}G$ y, por lo tanto, $\mathfrak{C}$ es un código cíclico. En efecto, si $(c_0, c_1, \ldots, c_{n-1}) \in \mathfrak{C}$, entonces

$$\begin{aligned}(c_{n-1}, c_0, \ldots, c_{n-2}) &= \vartheta^{-1}(c_{n-1} + c_0 g + \cdots + c_{n-2} g^{n-1}) \\ &= \vartheta^{-1}(g(c_0 + c_1 g + \ldots, +c_{n-1} g^{n-1})) \in \vartheta^{-1}(\vartheta(\mathfrak{C})) = \mathfrak{C},\end{aligned}$$

puesto que $\vartheta(\mathfrak{C})$ es un ideal de $\mathbb{K}G$. Existen una amplia variedad de códigos lineales que son códigos grupo (ver [33]). Algunos de los más conocidos son los códigos de Golay ([11]), los códigos generalizados de residuos cuadráticos ([61]), los códigos extendidos por paridad de los códigos Reed-Solomon en el sentido estricto ([55, 62]), y los códigos Reed-Muller generalizados ([1, 6, 16, 31, 32, 36, 49]).

En [8] se dan condiciones necesarias y suficientes para distinguir códigos grupo y códigos grupo a izquierda de entre otros códigos lineales. Esta caracterización se hace de manera intrínseca a partir del grupo de automorfismos permutación.

Teorema 1.3.3 ([8]). *Si $\mathfrak{C}$ es un código lineal de longitud n y G es un grupo finito de orden n, entonces*

1. *$\mathfrak{C}$ es $G-$código a izquierda si y solo si G es isomorfo a un subgrupo transitivo de S_n contenido en $PAut(\mathfrak{C})$.*

2. *$\mathfrak{C}$ es $G-$código si y solo si G es isomorfo a un subgrupo transitivo H de S_n tal que $H \cup C_{S_n}(H) \subseteq PAut(\mathfrak{C})$, donde $C_{S_n}(H)$ denota el centralizador de H en S_n.*

En el contexto de los códigos grupo, utilizaremos el $\mathbb{K}-$isomorfismo $\vartheta : \mathbb{K}^n \to \mathbb{K}G$ de la Definición 1.3.2 para identificar las palabras de $\mathbb{K}^n$ como elementos de $\mathbb{K}G$. Con esto, los elementos de $\mathbb{K}G$ algunas veces se escribirán como combinaciones $\mathbb{K}-$lineales de G y otras veces como $n-$tuplas de coeficientes en $\mathbb{K}$ con el orden fijado en G. Además, dado un $G-$código $\mathfrak{C} \subseteq \mathbb{K}^n$, trabajaremos indistintamente con $\mathfrak{C}$ o con $\vartheta(\mathfrak{C})$. Es decir, $\mathfrak{C}$ será considerado como un ideal bilátero de $\mathbb{K}G$. Notemos que bajo esta suposición, el conjunto $\mathfrak{C}^\perp$ dado por la expresión (1.2) coincide con el código dual de $\mathfrak{C}$ dado por la Definición 1.1.10 y gracias a la Proposición 1.2.11 también es un ideal bilátero de $\mathbb{K}G$. De hecho, tenemos la siguiente caracterización.

Proposición 1.3.4. *Si G es un grupo finito, entonces el código lineal $\mathfrak{C}$ es $G-$código si y solo si $\mathfrak{C}^\perp$ es $G-$código.*

Notemos que un código grupo se puede obtener utilizando grupos distintos y es posible que algunos de estos grupos sean abelianos y otros no.

Definición 1.3.5. *Sean G un grupo finito y $\mathfrak{C}$ un $G-$código. Se dice que $\mathfrak{C}$ es código grupo abeliano si existe un grupo abeliano A tal que $\mathfrak{C}$ es un $A-$código.*

Durante un tiempo estuvo planteada la pregunta: ¿Todo código grupo se puede realizar como código grupo abeliano? En [8] se encontraron condiciones para el grupo no abeliano G que garantizan que cualquier $G-$código es código grupo abeliano.

Teorema 1.3.6 ([8]). *Si G un grupo finito, y existen A y B subgrupos abelianos de G tales que $G = AB$, entonces todo $G-$código es código grupo abeliano.*

En trabajos sucesivos ([23, 24, 25]) se utilizó tal resultado para demostrar que todo $G-$código de longitud menor a 128 y distinta de 24, 48, 54, 60, 64, 72, 96, 108 y 120, es código grupo abeliano. Además, se estudió el comportamiento de los códigos grupo abelianos por extensión de escalares.

Proposición 1.3.7 ([25]). *Sean $\mathbb{K}$ un cuerpo finito y G un grupo finito. Si $\mathfrak{C}$ es un $G-$código que es código grupo abeliano sobre $\mathbb{K}$ y $\mathbb{F}$ es un subcuerpo de $\mathbb{K}$, entonces $\mathfrak{C}$ es un código grupo abeliano sobre $\mathbb{F}$.*

Proposición 1.3.8 ([25]). *Sean $\mathbb{K}$ un cuerpo finito y G un grupo finito. Supongamos que la característica de $\mathbb{K}$ no divide al orden de G y que $\mathbb{K}$ descompone a G. Si $\mathfrak{C}$ es un $G-$código que es código grupo abeliano sobre $\mathbb{K}$ y $\mathbb{E}$ es una extensión de $\mathbb{K}$, entonces $\mathfrak{C}$ es un código grupo abeliano sobre $\mathbb{E}$.*

Una condición suficiente para la dimensión de un código grupo abeliano puede encontrarse en [28].

Teorema 1.3.9 ([28]). *Sean $\mathbb{K}$ un cuerpo finito y G un grupo finito. Si $\mathfrak{C}$ es un $G-$código sobre $\mathbb{K}$ y su dimensión es $k \leq 3$, entonces $\mathfrak{C}$ es un código grupo abeliano.*

De la Definición 1.3.5, se dice que un código grupo $\mathfrak{C}$ es *no abeliano* si no existe un grupo abeliano G tal que $\mathfrak{C}$ es un $G-$código. Con esto, surge de manera natural la pregunta: ¿Existen tales códigos? Este fue el problema básico tratado en la tesis doctoral de C. García ([22]) en donde probó el siguiente resultado.

Teorema 1.3.10 ([25]). *Si $\mathfrak{C}$ es un código grupo no abeliano de longitud n, entonces $n \geq 24$.*

En los trabajos [23, 24, 25] también se construyeron S_4-códigos sobre el cuerpo $\mathbb{K} = \mathbb{F}_5$ que son códigos grupo no abelianos. La demostración se basa en el hecho que la distribución de pesos de los códigos grupo construidos no coinciden con la distribución de pesos de ningún código grupo abeliano de longitud 24.

Después, en [26], y considerando el mismo grupo, se estudió el caso no semisimple. En concreto, se construyeron códigos grupo no abelianos sobre $\mathbb{K} = \mathbb{F}_2$ y $\mathbb{K} = \mathbb{F}_3$. El argumento utilizado para $\mathbb{F}_3$ fue similar al usado para $\mathbb{F}_5$. El caso de $\mathbb{F}_2$ fue más complicado porque para todo código grupo sobre $\mathbb{F}_2$ de longitud 24, existe un código grupo abeliano con la misma longitud y distribución de pesos. Sin embargo, los autores lograron encontrar un código grupo que no es equivalente (por permutación) a ningún código grupo abeliano. Es interesante mencionar que los códigos grupo no abelianos obtenidos con el grupo S_4 resultaron tener peores parámetros que códigos grupo abelianos de la misma longitud. Por tanto, parece natural preguntarse si merece la pena trabajar con grupos no abelianos. En el mismo trabajo, considerando el grupo $G = \mathrm{SL}(2, \mathbb{F}_3)$, también se pudo construir un $G-$código no abeliano de dimensión 6 y distancia mínima 10, que es óptimo en el sentido que tal distancia es la máxima alcanzada por cualquier código lineal binario de longitud 24 de dimensión 6 y, además, no puede ser alcanzada utilizando un código grupo abeliano de longitud 24 y dimensión 6.

Finalmente, en [27], se usaron los grupos $G = S_4$ y $G = \mathrm{SL}(2, \mathbb{F}_3)$, para probar la existencia de códigos grupo no abelianos sobre $\mathbb{F}_p$ para cada número primo $p \geq 3$ y $p \geq 5$, respectivamente. Considerando cualquiera de los códigos grupo sobre $\mathbb{F}_2$ hallados en [26] y el Teorema 1.3.7, se obtiene el siguiente resultado.

Teorema 1.3.11 ([27])**.** *Existen códigos grupo no abelianos de longitud 24 sobre cualquier cuerpo finito.*

Todos estos resultados justifican el hecho de considerar códigos grupos con grupos no abelianos.

Capítulo 2

Descodificación por el conjunto de $\mathbb{K}G-$síndromes

Consideremos el álgebra de grupo $\mathbb{K}G$ con G un grupo finito de orden n y $\mathbb{K}$ un cuerpo finito. También vamos a fijar un orden en los elementos de G, es decir, fijamos una base $B = \{g_1 = 1_G, g_2, \ldots, g_n\}$ de $\mathbb{K}G$. Por tanto, los elementos de $\mathbb{K}G$ se pueden expresar como $n-$tuplas (o $n-$secuencias) de elementos de $\mathbb{K}$, respecto a la base fijada B.

En adelante, asumiremos que la característica de $\mathbb{K}$ no divide a n. Entonces, por el Teorema de Maschke (ver Teorema 1.2.12), el álgebra de grupo $\mathbb{K}G$ es semisimple y, por la Proposición 1.2.3, $\mathbb{K}G$ puede escribirse como suma de s ideales biláteros minimales. Estos ideales son las componentes simples de $\mathbb{K}G$ y están generadas por elementos idempotentes centrales primitivos de $\mathbb{K}G$ que denotaremos por $e_1, \ldots, e_s$.

Además, por la Proposición 1.2.5, cualquier ideal bilátero de $\mathbb{K}G$ está generado por un idempotente central y es suma directa de algunas componentes simples de $\mathbb{K}G$. Con lo anterior, si $\mathfrak{C}$ es un código grupo, entonces $\mathfrak{C}$ puede ser identificado con un ideal bilátero de $\mathbb{K}G$ y, por lo tanto, si reindexamos los idempotentes centrales primitivos de manera que $\{e_1, \ldots, e_m\}$, donde $m < s$, sea el conjunto de aquellos que son ortogonales a $\mathfrak{C}$, es decir que son ortogonales a los elementos de $\mathfrak{C}$, el código (grupo) $\mathfrak{C}$ está generado por el idempotente central $e_0 = e_{m+1} + \cdots + e_s$, y se denota $\mathfrak{C} = \langle e_0 \rangle$. En consecuencia, un elemento $\mathbf{z} \in \mathbb{K}G$ es una palabra código si y solo si existe $\mathbf{z}' \in \mathbb{K}G$ tal que $\mathbf{z} = \mathbf{z}'e_0$ y así $\mathbf{z} \in \mathbb{K}G$ es una palabra código si y solo si $\mathbf{z}e_h = 0$ para todo $h \in \{1, \ldots, m\}$.

A lo largo de este capítulo vamos a suponer que el código grupo $\mathfrak{C}$ tiene distancia mínima d y capacidad correctora igual a t.

2.1. Caso general

Definición 2.1.1. *Dado* $\mathbf{z} \in \mathbb{K}G$*, el* $\mathbb{K}G-$*síndrome de* $\mathbf{z}$ *asociado al idempotente central primitivo* $e_h \in \mathbb{K}G$ *se define como el elemento de* $\mathbb{K}G$ *dado por*

$$S_h(\mathbf{z}) := \mathbf{z}e_h.$$

Además, $\{S_1(\mathbf{z}), \ldots, S_m(\mathbf{z})\}$ *se denomina conjunto de* $\mathbb{K}G-$*síndromes de* $\mathbf{z}$*.*

Ahora bien, si se envía una palabra $\mathbf{c} \in \mathfrak{C}$ y durante la transmisión, se producen errores y los escribimos como el elemento $\mathbf{e} \in \mathbb{K}G$, entonces la palabra recibida será $\mathbf{r} = \mathbf{c}+\mathbf{e}$ y así, $S_h(\mathbf{r}) = (\mathbf{c}+\mathbf{e})e_h = \mathbf{e}e_h$ para todo $h \in \{1, \ldots, m\}$.

Lo anterior implica que si $\mathbf{x}$ denota una ideterminada en $\mathbb{K}G$, entonces *descodificar en* $\mathfrak{C}$ *por el conjunto de* $\mathbb{K}G-$*síndromes* consiste en encontrar una solución, con peso menor o igual a t, del sistema de ecuaciones

$$\begin{aligned} \mathbf{x}e_1 &= S_1(\mathbf{r}), \\ &\vdots \\ \mathbf{x}e_m &= S_m(\mathbf{r}). \end{aligned} \tag{2.1}$$

Para poder encontrar, de manera correcta, la palabra código enviada es necesario que esta solución sea única. Lo anterior ocurrirá, si el número de errores producidos es menor o igual que t.

Teorema 2.1.2. *Dado un código grupo* $\mathfrak{C}$ *que corrige hasta* t *errores, si* $\{S_1(\mathbf{r}), \ldots, S_m(\mathbf{r})\}$ *es el conjunto de* $\mathbb{K}G-$*síndromes de la palabra recibida* $\mathbf{r} \in \mathbb{K}G$*, entonces existe a lo sumo un elemento* $\mathbf{e} \in \mathbb{K}G$*, con peso* $w \leq t$*, que es solución del sistema (2.1).*

Demostración. Si $\mathbf{e}, \mathbf{e}' \in \mathbb{K}G$ son dos soluciones distintas del sistema (2.1) con pesos $w, w' \leq t$, respectivamente, entonces el peso de $\mathbf{e}-\mathbf{e}'$ es menor o igual a $w + w' \leq 2t \leq d-1$ y además $\mathbf{e}e_h = \mathbf{e}'e_h = S_h(\mathbf{r})$ para todo $h \in \{1, \ldots, m\}$. Con lo anterior, concluimos $(\mathbf{e}-\mathbf{e}')e_h = 0$, para todo $h \in \{1, \ldots, m\}$, es decir, $\mathbf{e}-\mathbf{e}' \in \mathfrak{C}$ y por ello tiene peso mayor o igual a d. Lo anterior contradice el hecho que $\mathbf{e}-\mathbf{e}'$ tiene peso menor o igual a $d-1$. □

Como la descodificación en códigos grupo de dimensión 1 es trivial, dado que las palabras código son múltiplos escalares del idempotente central que genera el código grupo, en lo que sigue, consideraremos solo códigos grupo con dimensión $k \geq 2$.

Con la notación anterior, $\mathbf{r} \in \mathfrak{C}$ si y solo si $S_h(\mathbf{r}) = \mathbf{r}e_h = 0$ para todo $h \in \{1, \ldots, m\}$. En tal caso no hay errores ($\mathbf{e} = 0$).

Para descodificar por el conjunto de $\mathbb{K}G-$síndromes, calculamos en primer lugar los productos $g_i e_h$, para todo par $i \in \{1, \ldots, n\}$ y $h \in \{1, \ldots, m\}$. Con esto, se define $C_{g_i}^h \in M_{n\times 1}(\mathbb{K})$ como el vector columna de los coeficientes de $g_i e_h$ con respecto a la base $B = G$, es decir, al orden fijado en G.

Si definimos la matriz

$$\mathcal{C}(g_{i_1}, \ldots, g_{i_b}) := \begin{pmatrix} C_{g_{i_1}}^1 & \cdots & C_{g_{i_b}}^1 \\ \vdots & & \vdots \\ C_{g_{i_1}}^m & \cdots & C_{g_{i_b}}^m \end{pmatrix},$$

tenemos el siguiente resultado.

Proposición 2.1.3. *Si $b < d$ y $g_{i_1}, \ldots, g_{i_b}$ son b elementos distintos de G, entonces la matriz $\mathcal{C}(g_{i_1}, \ldots, g_{i_b}) \in M_{nm\times b}(\mathbb{K})$, tiene rango igual a b.*

Demostración. Supongamos que $\mathcal{C}(g_{i_1}, \ldots, g_{i_b})$ tiene rango estrictamente menor que b. Entonces, existen $\nu_1, \ldots, \nu_b \in \mathbb{K}$, no todos iguales a cero, tales que $\nu_1 C_{g_{i_1}}^h + \cdots + \nu_b C_{g_{i_b}}^h = 0$. Con lo anterior, $(\nu_1 g_{i_1} + \cdots + \nu_b g_{i_b})e_h = 0$ para todo $h \in \{1, \ldots, m\}$ y, por tanto, $\mathbf{z} = \nu_1 g_{i_1} + \cdots + \nu_b g_{i_b} \in \mathfrak{C}$. Esto contradice el hecho que la distancia mínima de $\mathfrak{C}$ sea d, puesto que, $\mathbf{z}$ es distinto de cero y tiene peso menor o igual a $b < d$. □

Ahora explicaremos los detalles del algoritmo de descodificación por el conjunto de $\mathbb{K}G-$síndromes. Para ello, supongamos que recibimos una palabra $\mathbf{r} \in \mathbb{K}G$. En primer lugar, debemos decidir si $\mathbf{r}$ es una palabra código o no. En el segundo caso, necesitaremos recuperar la palabra código enviada. Para hacerlo, calculamos $\{S_1(\mathbf{r}), \ldots, S_m(\mathbf{r})\}$ el conjunto de $\mathbb{K}G-$síndromes. Luego, para cada $h \in \{1, \ldots, m\}$, definimos el vector columna S^h de los coeficientes de $S_h(\mathbf{r})$, con respecto a B. El objetivo del algoritmo de descodificación es encontrar un elemento $\mathbf{e} = \alpha_1 g_{i_1} + \cdots + \alpha_w g_{i_w}$, con peso $w \leq t$, tal que $\mathbf{e}e_h = S_h$ para todo $h \in \{1, \ldots, m\}$.

Como inicialmente no conocemos el valor de $w = \mathrm{wt}(\mathbf{e})$, vamos a considerar un $t-$subconjunto (ordenado) de elementos $g_{i_1}, \ldots, g_{i_t} \in G$ y verificamos si existe algún elemento $\mathbf{e} = \alpha_1 g_{i_1} + \cdots + \alpha_t g_{i_t}$ que sea solución del sistema (2.1).

Lo anterior ocurre si y solo si el sistema lineal

$$\begin{aligned} X_1 C^1_{g_{i_1}} + \cdots + X_t C^1_{g_{i_t}} &= S^1 \\ &\vdots \\ X_1 C^m_{g_{i_1}} + \cdots + X_t C^m_{g_{i_t}} &= S^m \end{aligned} \tag{2.2}$$

admite la solución $X_i = \alpha_i$ para todo $i \in \{1, \ldots, t\}$. Por el Teorema 2.1.2, sabemos que si este sistema tiene alguna solución, entonces es única. Esto equivale a que la matriz $\mathcal{C}(g_{i_1}, \ldots, g_{i_t})$ y la matriz extendida

$$\mathcal{M}(g_{i_1}, \ldots, g_{i_t}) := \left(\begin{array}{ccc|c} C^1_{g_{i_1}} & \cdots & C^1_{g_{i_t}} & S^1 \\ \vdots & & \vdots & \vdots \\ C^m_{g_{i_1}} & \cdots & C^m_{g_{i_t}} & S^m \end{array} \right),$$

tengan el mismo rango. Pero según la Proposición 2.1.3, el rango de $\mathcal{C}(g_{i_1}, \ldots, g_{i_t})$ es igual a t y, por lo tanto, solo necesitamos verificar que el rango de $\mathcal{M}(g_{i_1}, \ldots, g_{i_t})$ sea igual a t.

Observemos que el número de elementos $\alpha_1, \ldots, \alpha_w$ no nulos en la (única) solución del sistema (2.1) indica el número de errores producidos durante la transmisión y sus valores nos proporcionan las magnitudes de los errores.

En resumen, el algoritmo busca $t-$subconjuntos de G para los cuales, el correspondiente sistema lineal (2.2) es compatible determinado. Es decir, busca $t-$subconjuntos en G que contengan todas las posiciones de error. Suponiendo que el error producido es $\beta_1 g_{i_1} + \cdots + \beta_w g_{i_w}$, con $w = t$, entonces $\{g_{i_1}, \ldots, g_{i_t}\}$ es el único $t-$subconjunto que puede ser hallado por el algoritmo. Sin embargo, si $w < t$, el algoritmo puede encontrar varios $t-$subconjuntos que contienen las posiciones de error $\{g_{i_1}, \ldots, g_{i_w}\}$ y para los cuales el sistema (2.2) tiene solución única. Para cualquiera de los posibles $t-$subconjuntos que hacen que el sistema (2.2) sea compatible determinado, los coeficientes de g_j, para todo $j \neq i_1, \ldots, i_w$, son siempre iguales a cero en la solución.

2.1.1. Algoritmo de descodificación por el conjunto de $\mathbb{K}G-$síndromes (Algoritmo SSD)

A continuación presentamos la descripción del algoritmo de descodificación. Para ello, supongamos que $\mathbf{r} \in \mathbb{K}G$ es la palabra recibida.

Paso 1.

Se calcula el conjunto $\{S_1(\mathbf{r}), \ldots, S_m(\mathbf{r})\}$ de $\mathbb{K}G-$síndromes de $\mathbf{r}$. Si $S_h(\mathbf{r}) = 0$ para todo $h \in \{1, \ldots, m\}$, entonces no hay errores y $\mathbf{r}$ es la palabra código enviada. En ese caso, el algoritmo termina. De lo contrario, se procede al Paso 2.

Paso 2.

Se selecciona aleatoriamente un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G. Se considera la matriz $\mathcal{M}(g_{i_1}, \ldots, g_{i_t})$ y se calcula su rango.

a. Si el rango es igual a t, entonces se halla la (única) solución del sistema lineal (2.2). Si $\alpha_{j_1}, \ldots, \alpha_{j_w}$ es el conjunto de elementos no nulos de la solución anterior, entonces el error es $\mathbf{e} = \alpha_{j_1} g_{i_{j_1}} + \cdots + \alpha_{j_w} g_{i_{j_w}}$ y el algoritmo termina.

b. De lo contrario, se descarta el $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$. Se selecciona al azar otro $t-$subconjunto de G y se repite el Paso 2.

El algoritmo termina cuando es posible encontrar un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G tal que

$$\text{Rank}(\mathcal{M}(g_{i_1}, \ldots, g_{i_t})) = t, \tag{2.3}$$

o cuando todos los $t-$subconjuntos de G han sido considerados y ninguno satisface (2.3). En este último caso, se concluye que el número de errores producidos durante la transmisión es mayor a t y, por tanto, la palabra $\mathbf{r}$ no se puede descodificar.

Ejemplo 2.1.4. *Sean $G = S_4$ y $\mathbb{K} = \mathbb{F}_5$. Vamos a ordernar los elementos de la base $B = G$ como*

$$\begin{array}{lll}
g_1 = (1), & g_2 = (3,4), & g_3 = (2,3), \\
g_4 = (2,3,4), & g_5 = (2,4,3), & g_6 = (2,4), \\
g_7 = (1,2), & g_8 = (1,2)(3,4), & g_9 = (1,2,3), \\
g_{10} = (1,2,3,4), & g_{11} = (1,2,4,3), & g_{12} = (1,2,4), \\
g_{13} = (1,3,2), & g_{14} = (1,3,4,2), & g_{15} = (1,3), \\
g_{16} = (1,3,4), & g_{17} = (1,3)(2,4), & g_{18} = (1,3,2,4), \\
g_{19} = (1,4,3,2), & g_{20} = (1,4,2), & g_{21} = (1,4,3), \\
g_{22} = (1,4), & g_{23} = (1,4,2,3), & g_{24} = (1,4)(2,3).
\end{array}$$

El álgebra de grupo $\mathbb{F}_5 S_4$ se descompone como suma directa de cinco componentes simples de dimensiones 1,1,4,9 y 9 generadas por los idempotentes centrales primitivos $e_1, \ldots, e_5 \in \mathbb{F}_5 S_4$. De manera concreta,

$$\mathbb{F}_5 S_4 = \langle e_1 \rangle \oplus \langle e_2 \rangle \oplus \langle e_3 \rangle \oplus \langle e_4 \rangle \oplus \langle e_5 \rangle.$$

donde

$$e_1 = (4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4),$$
$$e_2 = (4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4),$$
$$e_3 = (1,0,0,2,2,0,0,1,2,0,0,2,2,0,0,2,1,0,0,2,2,0,0,1),$$
$$e_4 = (1,3,3,0,0,3,3,3,0,2,2,0,0,2,3,0,3,2,2,0,0,3,2,3),$$
$$e_5 = (1,2,2,0,0,2,2,3,0,3,3,0,0,3,2,0,3,3,3,0,0,2,3,3).$$

Si tomamos $\mathfrak{C} = \langle e_5 \rangle$, *entonces el código grupo* $\mathfrak{C}$ *tiene parámetros* $n = 24$, $k = 9$ *y* $d = 8$. *La capacidad correctora es* $t = 3$.

Si

$$\mathbf{r} = (2,1,4,1,1,4,0,0,1,1,3,4,1,3,4,4,1,1,1,4,4,0,4,2),$$

es la palabra recibida, entonces el proceso de descodificación es el siguiente:

Paso 1. *Calculamos* S_h *para* $h = 1,2,3,4$. *En este caso,*

$$S^1 = (4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4)^T,$$
$$S^2 = (1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1)^T,$$
$$S^3 = (0,1,2,0,0,2,1,0,0,2,2,0,0,2,2,0,0,1,2,0,0,2,1,0)^T,$$
$$S^4 = (1,3,0,2,4,0,0,1,1,0,0,3,3,0,0,4,4,4,0,1,2,0,3,4)^T.$$

Como no son iguales a cero, vamos al Paso 2.

Paso 2. *Consideramos los* $3-$*subconjuntos de* G. *Si empezamos con* $\{g_1, g_2, g_3\}$, *vemos que la matriz*

$$\mathcal{M}(g_1,g_2,g_3) = \left(\begin{array}{ccc|c} C^1_{g_1} & C^1_{g_2} & C^1_{g_3} & S^1 \\ C^2_{g_1} & C^2_{g_2} & C^2_{g_3} & S^2 \\ C^3_{g_1} & C^3_{g_2} & C^3_{g_3} & S^3 \\ C^4_{g_1} & C^4_{g_2} & C^4_{g_3} & S^4 \end{array}\right) \in M_{96\times 4}(\mathbb{F}_5),$$

tiene rango igual a 4, donde $C^1_{g_1} = C^1_{g_2} = C^1_{g_3}$, $C^2_{g_2} = C^2_{g_3}$ *y*

$$C^1_{g_3} = (4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4)^T,$$
$$C^2_{g_1} = (4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4)^T,$$
$$C^2_{g_3} = (1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1)^T,$$

$$C^3_{g_1} = (1,0,0,2,2,0,0,1,2,0,0,2,2,0,0,2,1,0,0,2,2,0,0,1)^T,$$
$$C^3_{g_2} = (0,1,2,0,0,2,1,0,0,2,2,0,0,2,2,0,0,1,2,0,0,2,1,0)^T,$$
$$C^3_{g_3} = (0,2,1,0,0,2,2,0,0,2,1,0,0,1,2,0,0,2,2,0,0,1,2,0)^T,$$
$$C^4_{g_1} = (1,3,3,0,0,3,3,3,0,2,2,0,0,2,3,0,3,2,2,0,0,3,2,3)^T,$$
$$C^4_{g_2} = (3,1,0,3,3,0,3,3,2,0,0,2,2,0,0,3,2,3,0,2,3,0,3,2)^T,$$
$$C^4_{g_3} = (3,0,1,3,3,0,0,2,3,0,3,2,3,3,0,2,2,0,0,2,2,3,0,3)^T.$$

Así que descartamos el $3-$*subconjunto* $\{g_1, g_2, g_3\}$. *Después de repetir este paso iteradamente, tomamos el* $3-$*subconjunto* $\{g_4, g_7, g_{18}\}$, *y vemos que la matriz*

$$\mathcal{M}(g_4, g_7, g_{18}) = \left(\begin{array}{ccc|c} C^1_{g_4} & C^1_{g_7} & C^1_{g_{18}} & S^1 \\ C^2_{g_4} & C^2_{g_7} & C^2_{g_{18}} & S^2 \\ C^3_{g_4} & C^3_{g_7} & C^3_{g_{18}} & S^3 \\ C^4_{g_4} & C^4_{g_7} & C^4_{g_{18}} & S^4 \end{array}\right),$$

tiene rango igual a 3, donde $C^1_{g_4} = C^1_{g_7} = C^1_{g_{18}}$, $C^2_{g_7} = C^2_{g_{18}}$, $C^3_{g_7} = C^3_{g_{18}}$ *y*

$$C^1_{g_{18}} = (4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4)^T,$$
$$C^2_{g_4} = (4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4)^T,$$
$$C^2_{g_{18}} = (1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1,1,4,4,1)^T,$$
$$C^3_{g_{18}} = (0,1,2,0,0,2,1,0,0,2,2,0,0,2,2,0,0,1,2,0,0,2,1,0)^T,$$
$$C^3_{g_4} = (2,0,0,1,2,0,0,2,2,0,0,1,1,0,0,2,2,0,0,2,1,0,0,2)^T,$$
$$C^4_{g_4} = (0,3,3,1,0,3,2,0,0,3,2,3,3,3,2,0,0,2,2,0,3,2,3,0)^T,$$
$$C^4_{g_7} = (3,3,0,2,2,0,1,3,3,0,0,3,3,0,0,2,2,3,0,3,2,0,3,2)^T,$$
$$C^4_{g_{18}} = (2,3,0,2,3,0,3,2,2,0,0,3,3,0,0,3,3,1,0,2,2,0,3,3)^T.$$

La solución del sistema

$$X_1 C^1_{g_4} + X_2 C^1_{g_7} + X_3 C^1_{g_{18}} = S^1$$

$$X_1 C^2_{g_4} + X_2 C^2_{g_7} + X_3 C^2_{g_{18}} = S^2$$

$$X_1 C^3_{g_4} + X_2 C^3_{g_7} + X_3 C^3_{g_{18}} = S^3$$

$$X_1 C^4_{g_4} + X_2 C^4_{g_7} + X_3 C^4_{g_{18}} = S^4$$

es $X_1 = 0$, $X_2 = 4$ *y* $X_3 = 2$.

En consecuencia,

$$\mathbf{e} = 4g_7 + 2g_{18} = (0,0,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0)$$

y así

$$\mathbf{c} = (2,1,4,1,1,4,1,0,1,1,3,4,1,3,4,4,1,4,1,4,4,0,4,2)$$

es la palabra código enviada.

Como hemos dicho previamente, hay varios $3-$*subconjuntos de* S_4 *que pueden ser usados para hallar el error. De hecho, todos aquellos que contienen las posiciones* g_7 *y* g_{18} *para un total de 22. Por ejemplo, si tomamos* $\{g_7, g_{12}, g_{18}\}$*, la matriz* $\mathcal{M}(g_7, g_{12}, g_{18})$ *tiene rango 3 y obtenemos la solución* $X_1 = 4$*,* $X_2 = 0$ *y* $X_3 = 2$*, que de nuevo producen el error* $\mathbf{e} = 2g_{13} + 3g_{17}$*.*

2.1.2. Análisis de la complejidad.

Ahora vamos a contar el número de operaciones que realiza el algoritmo anterior para recuperar una palabra después de que se haya producido un error. Para ello, vamos a considerar como *operación* con orden de complejidad $\mathcal{O}(1)$ las siguientes:

- Productos elementos de $\mathbb{K}$ o evaluación de una aplicación $f : \mathbb{K} \to \mathbb{K}$.
- Productos elementos de G o evaluación de una aplicación $f : G \to G$.
- Sumas o comparación de elementos (iguales o distintos) de $\mathbb{K}G$ de peso 1. En particular, elementos de $\mathbb{K}$.

Por lo tanto, si consideraramos los elementos de $\mathbb{K}G$ como $n-$tuplas, entonces sumar o comparar elementos de $\mathbb{K}G$, o evaluar una función de permutación $f : \mathbb{K}G \to \mathbb{K}G$ tiene orden de complejidad $\mathcal{O}(n)$.

Antes de iniciar el algoritmo, se calculan todos los productos $g_i e_h$ para cada par $i \in \{1, \ldots, n\}$ y $h \in \{1, \ldots, m\}$. Este proceso requiere un total de mn^2 operaciones.

En el paso 1, para calcular el conjunto de $\mathbb{K}G-$síndromes hay que hacer $3mn^2$ operaciones en total. Verificar que cada $S^h(\mathbf{r})$ sea igual a cero, requiere mn operaciones. Con lo cual, este paso tiene orden de complejidad $\mathcal{O}(mn^2)$.

Si alguno de los $\mathbb{K}G-$síndromes es distinto de cero, al tomar $\{g_{i_1}, \ldots, g_{i_t}\}$, calcular (por eliminación gaussiana) el rango de $\mathcal{M}^+(g_{i_1}, \ldots, g_{i_t})$ tiene orden de complejidad $\mathcal{O}(mn \times t^2)$. Como el algoritmo realiza la búsqueda en todos los $t-$subconjuntos de G, entonces el orden de complejidad del Paso 2 es

$$\mathcal{O}\left(mnt^2 \times \binom{n}{t}\right). \tag{2.4}$$

Además, observemos que si el error $\mathbf{e} \in \mathbb{K}G$ tiene un peso $w < t$, entonces el número de $t-$subconjuntos de G que contienen a $\mathrm{Supp}(\mathbf{e})$, el soporte de $\mathbf{e}$, es

$$\binom{n-w}{t-w}.$$

Por consiguiente, el número medio de $t-$subconjuntos de G que contienen a $\mathrm{Supp}(\mathbf{e})$ es

$$\binom{n}{t} \div \binom{n-w}{t-w}$$

y, por lo tanto, la probabilidad de tomar aleatoriamente un $t-$subconjunto de G que contenga a $\mathrm{Supp}(\mathbf{e})$ es

$$\binom{n-w}{t-w} \div \binom{n}{t}.$$

Así, cuanto más pequeño sea w, la probabilidad de encontrar tal $t-$subconjunto será mayor y así el algoritmo será más eficiente.

2.2. Un caso especial

En esta sección presentamos una variación del algoritmo anterior cuando el grupo G es abeliano y las componentes simples de $\mathbb{K}G$ tienen dimensión 1. Esto ocurre cuando $\mathbb{K}$ posee una $u-$raíz primitiva de la unidad, siendo $u = \exp(G)$ (ver [14]). En tal caso, hay exactamente $m = n - k$ idempotentes centrales primitivos que son ortogonales al código grupo y que denotaremos $e_1, \ldots, e_{n-k}$.

Antes de comenzar el algoritmo de descodificación para tal caso, calculamos los escalares $\lambda_{g_i}^h \in \mathbb{K}$ tales que $g_i e_h = \lambda_{g_i}^h e_h$, para todo par $i \in \{1, \ldots, n\}$ y $h \in \{1, \ldots, n-k\}$. Notemos que esto se debe a que las componentes simples $\langle e_1 \rangle, \ldots, \langle e_{n-k} \rangle$ tienen dimensión 1. Para cada $g_i \in G$, definimos el vector columna

$$A_{g_i} := (\lambda_{g_i}^1, \ldots, \lambda_{g_i}^{n-k})^T.$$

También definimos la matriz

$$\mathcal{A}(g_{i_1},\ldots,g_{i_b}) := \begin{pmatrix} A_{g_{i_1}} & \cdots & A_{g_{i_b}} \end{pmatrix}.$$

Proposición 2.2.1. *Si $b < d$ y $g_{i_1},\ldots,g_{i_b}$ son b elementos distintos de G, entonces la matriz*

$$\mathcal{A}(g_{i_1},\ldots,g_{i_b}) = \begin{pmatrix} \lambda^1_{g_{i_1}} & \cdots & \lambda^1_{g_{i_b}} \\ \vdots & & \vdots \\ \lambda^{n-k}_{g_{i_1}} & \cdots & \lambda^{n-k}_{g_{i_b}} \end{pmatrix} \in M_{(n-k)\times b}(\mathbb{K})$$

tiene rango igual a b.

Demostración. Es análoga a la Proposición 2.1.3. □

Una vez se recibe la palabra $\mathbf{r} \in \mathbb{K}G$, el algoritmo calcula su conjunto $\mathbb{K}G-$síndromes $\{S_1(\mathbf{r}),\ldots,S_{n-k}(\mathbf{r})\}$ y luego, se hallan los elementos $\mu_h \in \mathbb{K}$ tales que $S_h(\mathbf{r}) = \mu^h e_h$ para $h = 1,\ldots,n-k$. Lo anterior nos permite definir el vector columna

$$S := (\mu^1,\ldots,\mu^{n-k})^T.$$

El algoritmo de descodificación buscará $t-$subconjuntos de G que contengan todas las posiciones de error.

Teniendo en cuenta que los sistemas de ecuaciones lineales (2.2) y

$$X_1 A_{g_{i_1}} + \cdots + X_t A_{g_{i_t}} = S. \tag{2.5}$$

son equivalentes, el objetivo será encontrar t elementos distintos $g_{i_1},\ldots,g_{i_t}$ de G tales que el correspondiente sistema lineal (2.5) tenga solución única. Esto sucede si y solo si la matriz extendida

$$\mathcal{E}(g_{i_1},\ldots,g_{i_t}) := \left(\begin{array}{ccc|c} \lambda^1_{g_{i_1}} & \cdots & \lambda^1_{g_{i_t}} & \mu^1 \\ \vdots & & \vdots & \vdots \\ \lambda^{n-k}_{g_{i_1}} & \cdots & \lambda^{n-k}_{g_{i_t}} & \mu^{n-k} \end{array}\right)$$

tiene rango igual al de la matriz $\mathcal{A}(g_{i_1},\ldots,g_{i_t})$. Pero, según la Proposición 2.2.1, el rango de esta última matriz es igual a t. Con lo cual, basta encontrar un $t-$subconjunto $\{g_{i_1},\ldots,g_{i_t}\}$ de G tal que la correspondiente matriz $\mathcal{E}(g_{i_1},\ldots,g_{i_t})$ tenga rango igual a t. Ahora bien, si $\alpha_1,\ldots,\alpha_t \in \mathbb{K}$ es la solución del sistema asociado a la matriz $\mathcal{E}(g_{i_1},\ldots,g_{i_t})$, entonces el error es $\mathbf{e} = \alpha_1 g_{i_1} + \cdots + \alpha_t g_{i_t}$.

Algoritmo de descodificación

A continuación presentamos explícitamente el algoritmo de descodificación para este caso.

Paso 1.

Recibida la palabra $\mathbf{r} \in \mathbb{K}G$, se calcula el conjunto $\{S_1(\mathbf{r}), \ldots, S_{n-k}(\mathbf{r})\}$ de sus $\mathbb{K}G-$síndromes. Si $S_h(\mathbf{r}) = 0$ para todo $h \in \{1, \ldots, n-k\}$, entonces no hay errores y el algoritmo finaliza. De lo contrario, se calcula S y se continúa al Paso 2.

Paso 2.

Se selecciona aleatoriamente un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G. Se considera la matriz $\mathcal{E}(g_{i_1}, \ldots, g_{i_t})$ y se calcula su rango.

a. Si el rango es igual a t, se halla la (única) solución del sistema lineal (2.5). Si $\alpha_{j_1}, \ldots, \alpha_{j_w}$ es el conjunto de elementos no nulos de la solución anterior, entonces el error es $\mathbf{e} = \alpha_{j_1} g_{i_{j_1}} + \cdots + \alpha_{j_w} g_{i_{j_w}}$ y el algoritmo termina.

b. De lo contrario, se descarta el $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$, se selecciona un nuevo $t-$subconjunto aleatoriamente y se repite el Paso 2.

El algoritmo termina cuando se encuentra un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G tal que

$$\text{Rank}(\mathcal{E}(g_{i_1}, \ldots, g_{i_t})) = t, \tag{2.6}$$

o cuando todos los $t-$subconjuntos de G han sido considerados y ninguno satisface (2.6). En este último caso, la palabra $\mathbf{r}$ no se puede descodificada, puesto que, el número de errores producidos durante la transmisión es mayor que la capacidad correctora t.

Observemos que si G es abeliano y las componentes simples de $\mathbb{K}G$ tienen dimensión 1, la variante del Algoritmo SSD es más eficiente, puesto que calcula rangos de matrices con $n-k$ filas en lugar de matrices con $n(n-k)$ filas.

2.3. Caso abeliano

El algoritmo visto en la sección anterior demostró ser más eficiente cuando se trata de grupos abelianos. Sin embargo, el cuerpo $\mathbb{K}$ tiene que ser lo suficiente grande para contener una $u-$raíz primitiva de la unidad donde $u = \exp(G)$.

Por lo que nuestro objetivo ahora, será ajustar el algoritmo general para usarlo en cuerpos finitos arbitrarios. En esta sección, G es un grupo abeliano cuyo orden n no es divisible por la característica del cuerpo finito $\mathbb{K}$.

El siguiente resultado, probablemente conocido, es esencial para lo que sigue. Como no pudimos encontrar una referencia para este, incluimos una demostración.

Proposición 2.3.1. *Sean $\mathbb{K}$ un cuerpo finito y $\mathbb{E}$ una extensión finita de $\mathbb{K}$. Si $\mathfrak{C}$ es un código lineal sobre $\mathbb{K}$, entonces el código lineal $\widetilde{\mathfrak{C}} = \mathfrak{C} \otimes_{\mathbb{K}} \mathbb{E}$ tiene los mismos parámetros de $\mathfrak{C}$.*

Demostración. Sabemos que si V es un espacio vectorial sobre $\mathbb{K}$, entonces $V \otimes_{\mathbb{K}} \mathbb{E}$ también es un espacio vectorial sobre $\mathbb{E}$, donde el producto del vector $v \otimes \gamma \in V \otimes_{\mathbb{K}} \mathbb{E}$ por el escalar $\alpha \in \mathbb{E}$ está dado por

$$\alpha(v \otimes \gamma) := v \otimes (\alpha\gamma).$$

También sabemos que si $\{v_i\}_{i\in I}$ es una base de V sobre $\mathbb{K}$, entonces $\{v_i \otimes 1\}_{i\in I}$ es una base de $V \otimes_{\mathbb{K}} \mathbb{E}$ sobre $\mathbb{E}$. Así $\mathfrak{C}$ y $\widetilde{\mathfrak{C}}$ tienen la misma longitud y la misma dimensión, y además, tienen en común una matriz de control $\mathcal{H}$ (cuyas entradas son elementos de $\mathbb{K}$). Si denotamos $\widetilde{d}$ la distancia mínima de $\widetilde{\mathfrak{C}}$ y d la distancia mínima de $\mathfrak{C}$, entonces en virtud de la segunda parte de la Proposición 1.1.11 tenemos que

$$\begin{aligned} d &= \text{mín}\{b \mid \text{Existen } b \text{ columnas de } \mathcal{H} \text{ linealmente } \mathbb{K}-\text{dependientes}\} \\ &= \text{mín}\{\widetilde{b} \mid \text{Existen } \widetilde{b} \text{ columnas de } \mathcal{H} \text{ linealmente } \mathbb{E}-\text{dependientes}\} = \widetilde{d}. \end{aligned}$$

□

De ahora en adelante, consideramos el cuerpo finito $\mathbb{E}$ como la extensión más pequeña de $\mathbb{K}$ tal que las componentes simples de $\mathbb{E}G$ tienen dimensión 1. Entonces, $\mathbb{E}G$ es semisimple y sus n componentes simples son generadas por idempotentes centrales primitivos que denotaremos por $f_1, \ldots, f_n$.

Recordemos que si $\{e_1, \ldots, e_s\}$ es el conjunto de todos los idempotentes centrales primitivos de $\mathbb{K}G$, entonces cada e_i es suma de aquellos idempotentes $f_j \in \{f_1, \ldots, f_n\}$ tales que $e_i f_j \neq 0$. En efecto, como $\{f_1, \ldots, f_n\}$ es una base de $\mathbb{E}G$ sobre $\mathbb{E}$, entonces existen $\gamma_1^i, \ldots, \gamma_n^i \in \mathbb{E}$ tales que $e_i = \gamma_1^i f_1 + \cdots + \gamma_n^i f_n$. Por tanto, $e_i = e_i^2 = (\gamma_1^i)^2 f_1 + \cdots + (\gamma_n^i)^2 f_n$ y así $(\gamma_j^i)^2 = \gamma_j^i$. De lo anterior, $\gamma_j^i = 0$ o $\gamma_j^i = 1$ para todo $j \in \{1, \ldots, n\}$ y $\gamma_j^i = 1$ si y solo si $e_i f_j \neq 0$.

Si $\mathfrak{C}$ es el código grupo generado por $e_0 \in \mathbb{K}G$ y $\widetilde{\mathfrak{C}} = \mathfrak{C} \otimes_{\mathbb{K}} \mathbb{E}$ es el código grupo en $\mathbb{E}G$ generado por $e_0 \in \mathbb{E}G$, entonces por la Proposición 2.3.1 tenemos que $\widetilde{\mathfrak{C}}$ corrige el mismo número de errores que $\mathfrak{C}$. Ahora vamos a reindexar los idempotentes $f_1, \ldots, f_n$ de tal manera que $e_0 = f_{n-k+1} + \cdots + f_n$.

Recibida la palabra $\mathbf{r} \in \mathbb{K}G$, la podemos considerar como un elemento de $\mathbb{E}G$ y la vamos a descodificar usando el código $\widetilde{\mathfrak{C}}$. Para ello, calculamos el conjunto de $\mathbb{E}G-$síndromes de $\mathbf{r}$ denotado por $\{\widetilde{S}_1(\mathbf{r}), \ldots, \widetilde{S}_{n-k}(\mathbf{r})\}$, donde

$$\widetilde{S}_j(\mathbf{r}) = \mathbf{r}f_j, \qquad j = 1, \ldots, n-k.$$

Como el error $\mathbf{e}$ tiene coeficientes en $\mathbb{K}$, el algoritmo descodifica correctamente gracias al siguiente resultado.

Proposición 2.3.2. *Adoptando la notación anterior, si $\mathbf{e} \in \mathbb{K}G$ y $\widetilde{\mathbf{e}} \in \mathbb{E}G$ tienen peso menor o igual a t y además*

$$\mathbf{e}e_h = S_h(\mathbf{r}), \qquad h = 1, \ldots, m, \tag{2.7}$$

y

$$\widetilde{\mathbf{e}}f_j = \widetilde{S}_j(\mathbf{r}), \qquad j = 1, \ldots, n-k, \tag{2.8}$$

entonces $\widetilde{\mathbf{e}} = \mathbf{e}$ y, por lo tanto, $\widetilde{\mathbf{e}} \in \mathbb{K}G$.

Demostración. Para cada $e_i \in \mathbb{K}G$, denotemos a $J(i)$ como el conjunto de aquellos $j \in \{1, \ldots, n\}$ tales que $e_i f_j \neq 0$. Entonces, $e_i = \sum_{j \in J(i)} f_j$ y

$$S_h(\mathbf{r}) = \mathbf{r}e_h = \mathbf{r}\left(\sum_{j \in J(h)} f_j\right) = \sum_{j \in J(h)} \widetilde{S}_j(\mathbf{r}),$$

para todo $h \in \{1, \ldots, m\}$. Por la hipótesis del Teorema 2.1.2, $\mathbf{e}$ y $\widetilde{\mathbf{e}}$ son los únicos elementos de $\mathbb{K}G$ y $\mathbb{E}G$ que satisfacen (2.7) y (2.8), respectivamente. Además, (2.7) implica que

$$\mathbf{e}\left(\sum_{j \in J(h)} f_j\right) = \mathbf{e}e_h = S_h(\mathbf{r}) = \sum_{j \in J(h)} \widetilde{S}_j(\mathbf{r}),$$

para todo $h \in \{1, \ldots, m\}$. Así,

$$\sum_{j \in J(h)} [\mathbf{e}f_j - \widetilde{S}_j(\mathbf{r})] = 0$$

y dado que $\mathbf{e}f_j - \widetilde{S}_j(\mathbf{r}) \in \langle f_j \rangle$ para todo $j \in J(h)$, entonces $\mathbf{e}f_j - \widetilde{S}_j(\mathbf{r}) = 0$ para todo $j \in J(h)$. Esto implica que $\mathbf{e}f_j = \widetilde{S}_j(\mathbf{r})$ para todo $j \in J(h)$ y, por consiguiente, $\mathbf{e} \in \mathbb{K}G \subseteq \mathbb{E}G$ satisface (2.8). Por unicidad, tenemos que $\widetilde{\mathbf{e}} = \mathbf{e}$ y, en consecuencia, $\widetilde{\mathbf{e}} \in \mathbb{K}G$. □

Este resultado asegura que si usamos $\widetilde{\mathfrak{C}}$ para descodificar a $\mathbf{r} \in \mathbb{K}G$ y obtenemos una palabra código $\mathbf{c} \in \widetilde{\mathfrak{C}} \setminus \mathfrak{C}$, entonces el número de errores producidos durante la transmisión es mayor que la capacidad correctora de $\mathfrak{C}$.

Así, para descodificar en un código grupo abeliano (en el caso semisimple), basta extender el cuerpo $\mathbb{K}$ a un cuerpo $\mathbb{E}$ que contenga una $u-$raíz primitiva de la unidad siendo $u = \exp(G)$. A continuación veremos un "pequeño" ejemplo.

Ejemplo 2.3.3. *Sean $\mathbb{K} = \mathbb{F}_2$ y $G = C_3 \times C_3 \times C_3 = \langle a, b, c \rangle$. Aquí $\mathbb{F}_2(C_3 \times C_3 \times C_3)$ es suma de 14 componentes simples. Una tiene dimensión 1 y las demás tienen dimensión 2. Si en G fijamos el siguiente orden*

$$\{1_G, a, b, c, a^2, ab, ac, b^2, bc, c^2, a^2b, a^2c, ab^2, abc, ac^2, b^2c, bc^2,$$
$$a^2b^2, a^2bc, a^2c^2, ab^2c, abc^2, b^2c^2, a^2b^2c, a^2bc^2, ab^2c^2, a^2b^2c^2\},$$

entonces las componentes simples de $\mathbb{F}_2(C_3 \times C_3 \times C_3)$ son generadas, respectivamente, por los siguientes idempotentes centrales primitivos:

$$e_1 = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),$$
$$e_2 = (0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0),$$
$$e_3 = (0,1,1,1,1,1,1,0,1,0,1,0,1,0,1,1,1,1,1,1,1,0,1,0,0,0,1,1,1),$$
$$e_4 = (0,1,1,1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,0,1,0,1,1),$$
$$e_5 = (0,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,1,1,1,1,1,1,0,1),$$
$$e_6 = (0,1,1,1,0,1,1,1,1,1,1,0,0,1,0,1,1,1,1,1,1,1,0,1,1,0,1,1,1,0,0,1),$$
$$e_7 = (0,1,1,1,0,1,1,0,1,1,1,1,0,1,1,1,1,0,1,1,1,1,0,1,1,1,0,1,0,1,1,0),$$
$$e_8 = (0,1,1,0,1,1,1,1,1,0,1,1,1,1,0,1,1,0,1,1,1,1,0,1,1,1,0,1,0,1,0,1),$$
$$e_9 = (0,1,1,0,1,1,1,1,0,0,1,1,1,1,1,1,0,1,1,1,1,1,1,0,0,1,1,1,1,0,1,0),$$
$$e_{10} = (0,1,1,0,0,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,0,1,1,1,1),$$
$$e_{11} = (0,0,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,0,0,1,0,0,1,1),$$
$$e_{12} = (0,0,1,1,1,0,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,0,1,1,0,0),$$
$$e_{13} = (0,0,1,1,0,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1),$$
$$e_{14} = (0,0,0,1,1,0,0,1,1,0,1,1,0,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1).$$

La extensión $\mathbb{E}$ más pequeña de $\mathbb{F}_2$ tal que las dimensiones de todas las componentes simples de $\mathbb{E}G$ son iguales a 1, es $\mathbb{E} = \mathbb{F}_4 = \mathbb{F}_2(\omega)$ donde $\omega^2 = \omega + 1$. En este caso, los idempotentes centrales primitivos de $\mathbb{F}_4(C_3 \times C_3 \times C_3)$ son:

$$f_1 = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),$$

$f_2 = (1, \omega, \omega, \omega, \omega^2, \omega^2, \omega^2, \omega^2, \omega^2, \omega^2, 1, 1, 1, 1, 1, 1, 1, \omega, \omega, \omega, \omega, \omega, \omega, \omega^2, \omega^2, \omega^2, 1),$

$f_3 = (1, \omega^2, \omega^2, \omega^2, \omega, \omega, \omega, \omega, \omega, \omega, 1, 1, 1, 1, 1, 1, 1, \omega^2, \omega^2, \omega^2, \omega^2, \omega^2, \omega^2, \omega, \omega, \omega, 1),$

$f_4 = (1, \omega, \omega, \omega^2, \omega^2, \omega^2, 1, \omega^2, 1, \omega, 1, \omega, 1, \omega, \omega^2, \omega, \omega^2, \omega, \omega^2, 1, \omega^2, 1, 1, 1, \omega, \omega, \omega^2),$

$f_5 = (1, \omega^2, \omega^2, \omega, \omega, \omega, 1, \omega, 1, \omega^2, 1, \omega^2, 1, \omega^2, \omega, \omega^2, \omega, \omega^2, \omega, 1, \omega, 1, 1, 1, \omega^2, \omega^2, \omega),$

$f_6 = (1, \omega, \omega^2, \omega, \omega^2, 1, \omega^2, \omega, 1, \omega^2, \omega, 1, \omega^2, \omega, 1, \omega^2, \omega, 1, \omega^2, \omega, 1, \omega^2, 1, \omega, 1, \omega, \omega^2),$

$f_7 = (1, \omega^2, \omega, \omega^2, \omega, 1, \omega, \omega^2, 1, \omega, \omega^2, 1, \omega, \omega^2, 1, \omega, \omega^2, 1, \omega, \omega^2, 1, \omega, 1, \omega^2, 1, \omega^2, \omega),$

$f_8 = (1, \omega, \omega^2, \omega^2, \omega^2, 1, 1, \omega, \omega, \omega, \omega, \omega, \omega^2, \omega^2, \omega^2, 1, 1, 1, 1, 1, 1, \omega, \omega, \omega^2, \omega^2, \omega^2, 1, \omega),$

$f_9 = (1, \omega^2, \omega, \omega, \omega, 1, 1, \omega^2, \omega^2, \omega^2, \omega^2, \omega^2, \omega, \omega, \omega, 1, 1, 1, 1, 1, 1, \omega^2, \omega^2, \omega, \omega, \omega, 1, \omega^2),$

$f_{10} = (1, \omega, \omega, 1, \omega^2, \omega^2, \omega, \omega^2, \omega, 1, 1, \omega^2, 1, \omega^2, \omega, \omega^2, \omega, \omega, 1, \omega^2, 1, \omega^2, \omega^2, \omega, 1, 1, \omega),$

$f_{11} = (1, \omega^2, \omega^2, 1, \omega, \omega, \omega^2, \omega, \omega^2, 1, 1, \omega, 1, \omega, \omega^2, \omega, \omega^2, \omega^2, 1, \omega, 1, \omega, \omega, \omega^2, 1, 1, \omega^2),$

$f_{12} = (1, \omega, \omega^2, 1, \omega^2, 1, \omega, \omega, \omega^2, 1, \omega, \omega^2, \omega^2, 1, \omega, \omega, \omega^2, 1, \omega, \omega^2, \omega^2, 1, \omega, 1, \omega, \omega^2, 1),$

$f_{13} = (1, \omega^2, \omega, 1, \omega, 1, \omega^2, \omega^2, \omega, 1, \omega^2, \omega, \omega, 1, \omega^2, \omega^2, \omega, 1, \omega^2, \omega, \omega, 1, \omega^2, 1, \omega^2, \omega, 1),$

$f_{14} = (1, \omega, 1, \omega, \omega^2, \omega, \omega^2, 1, \omega, \omega^2, \omega^2, 1, \omega, \omega^2, 1, \omega, \omega^2, \omega^2, 1, \omega, \omega^2, 1, \omega^2, 1, \omega, 1, \omega),$

$f_{15} = (1, \omega^2, 1, \omega^2, \omega, \omega^2, \omega, 1, \omega^2, \omega, \omega, 1, \omega^2, \omega, 1, \omega^2, \omega, \omega, 1, \omega^2, \omega, 1, \omega, 1, \omega^2, 1, \omega^2),$

$f_{16} = (1, \omega, 1, \omega^2, \omega^2, \omega, 1, 1, \omega^2, \omega, \omega^2, \omega, \omega, 1, \omega^2, \omega^2, \omega, \omega^2, \omega, 1, 1, \omega^2, \omega, \omega, 1, \omega^2, 1),$

$f_{17} = (1, \omega^2, 1, \omega, \omega, \omega^2, 1, 1, \omega, \omega^2, \omega, \omega^2, \omega^2, 1, \omega, \omega, \omega^2, \omega, \omega^2, 1, 1, \omega, \omega^2, \omega^2, 1, \omega, 1),$

$f_{18} = (1, \omega, 1, 1, \omega^2, \omega, \omega, 1, 1, 1, \omega^2, \omega^2, \omega, \omega, \omega, 1, 1, \omega^2, \omega^2, \omega^2, \omega, \omega, 1, \omega^2, \omega^2, \omega, \omega^2),$

$f_{19} = (1, \omega^2, 1, 1, \omega, \omega^2, \omega^2, 1, 1, 1, \omega, \omega, \omega^2, \omega^2, \omega^2, 1, 1, \omega, \omega, \omega, \omega^2, \omega^2, 1, \omega, \omega, \omega^2, \omega),$

$f_{20} = (1, 1, \omega, \omega, 1, \omega, \omega, \omega^2, \omega^2, \omega^2, \omega, \omega, \omega^2, \omega^2, \omega^2, 1, 1, \omega^2, \omega^2, \omega^2, 1, 1, \omega, 1, 1, \omega, \omega),$

$f_{21} = (1, 1, \omega^2, \omega^2, 1, \omega^2, \omega^2, \omega, \omega, \omega, \omega^2, \omega^2, \omega, \omega, \omega, 1, 1, \omega, \omega, \omega, 1, 1, \omega^2, 1, 1, \omega^2, \omega^2),$

$f_{22} = (1, 1, \omega, \omega^2, 1, \omega, \omega^2, \omega^2, 1, \omega, \omega, \omega^2, \omega^2, 1, \omega, \omega, \omega^2, \omega^2, 1, \omega, \omega, \omega^2, 1, \omega, \omega^2, 1, 1),$

$f_{23} = (1, 1, \omega^2, \omega, 1, \omega^2, \omega, \omega, 1, \omega^2, \omega^2, \omega, \omega, 1, \omega^2, \omega^2, \omega, \omega, 1, \omega^2, \omega^2, \omega, 1, \omega^2, \omega, 1, 1),$

$f_{24} = (1, 1, \omega, 1, 1, \omega, 1, \omega^2, \omega, 1, \omega, 1, \omega^2, \omega, 1, \omega^2, \omega, \omega^2, \omega, 1, \omega^2, \omega, \omega^2, \omega^2, \omega, \omega^2, \omega^2),$

$f_{25} = (1, 1, \omega^2, 1, 1, \omega^2, 1, \omega, \omega^2, 1, \omega^2, 1, \omega, \omega^2, 1, \omega, \omega^2, \omega, \omega^2, 1, \omega, \omega^2, \omega, \omega, \omega^2, \omega, \omega),$

$f_{26} = (1, 1, 1, \omega, 1, 1, \omega, 1, \omega, \omega^2, 1, \omega, 1, \omega, \omega^2, \omega, \omega^2, 1, \omega, \omega^2, \omega, \omega^2, \omega^2, \omega, \omega^2, \omega^2, \omega^2),$

$f_{27} = (1, 1, 1, \omega^2, 1, 1, \omega^2, 1, \omega^2, \omega, 1, \omega^2, 1, \omega^2, \omega, \omega^2, \omega, 1, \omega^2, \omega, \omega^2, \omega, \omega, \omega^2, \omega, \omega, \omega).$

Aquí $e_1 = f_1$, $e_2 = f_2 + f_3$, $e_3 = f_4 + f_5$, $e_4 = f_6 + f_7$, $e_5 = f_8 + f_9$, $e_6 = f_{10} + f_{11}$, $e_7 = f_{12} + f_{13}$, $e_8 = f_{14} + f_{15}$, $e_9 = f_{16} + f_{17}$, $e_{10} = f_{18} + f_{19}$, $e_{11} = f_{20} + f_{21}$, $e_{12} = f_{22} + f_{23}$, $e_{13} = f_{24} + f_{25}$, $e_{14} = f_{26} + f_{27}$.

Consideremos $\mathfrak{C} = \langle e_0 \rangle$, *donde* $e_0 = e_9 + e_{10} + e_{11} + e_{12} + e_{13} + e_{14}$. *Entonces los parámetros de* $\mathfrak{C}$ *son* $n = 27$, $k = 12$ *y* $d = 6$. *Su capacidad correctora es* $t = 2$. *Con lo anterior,*

$$\widetilde{\mathfrak{C}} = \langle f_{16} + f_{17} + f_{18} + f_{19} + f_{20} + f_{21} + f_{22} + f_{23} + f_{24} + f_{25} + f_{26} + f_{27} \rangle,$$

tiene los mismos parámetros de $\mathfrak{C}$. *Vamos a descodificar en* $\widetilde{\mathfrak{C}}$. *Para ello, calculamos que:*

$$\widetilde{A}_{1_G} = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)^T,$$
$$\widetilde{A}_a = (1,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega)^T,$$
$$\widetilde{A}_b = (1,\omega^2,\omega,\omega^2,\omega,\omega,\omega^2,\omega,\omega^2,\omega^2,\omega,\omega,\omega^2,1,1)^T,$$
$$\widetilde{A}_c = (1,\omega^2,\omega,\omega,\omega^2,\omega^2,\omega,\omega,\omega^2,1,1,1,1,\omega^2,\omega)^T,$$
$$\widetilde{A}_{a^2} = (1,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2,\omega,\omega^2)^T,$$
$$\widetilde{A}_{ab} = (1,\omega,\omega^2,\omega,\omega^2,1,1,1,1,\omega,\omega^2,1,1,\omega^2,\omega)^T,$$
$$\widetilde{A}_{ac} = (1,\omega,\omega^2,1,1,\omega,\omega^2,1,1,\omega^2,\omega,\omega^2,\omega,\omega,\omega^2)^T,$$
$$\widetilde{A}_{b^2} = (1,\omega,\omega^2,\omega,\omega^2,\omega^2,\omega,\omega^2,\omega,\omega,\omega^2,\omega^2,\omega,1,1)^T,$$
$$\widetilde{A}_{bc} = (1,\omega,\omega^2,1,1,1,1,\omega^2,\omega,\omega^2,\omega,\omega,\omega^2,\omega^2,\omega)^T,$$
$$\widetilde{A}_{c^2} = (1,\omega,\omega^2,\omega^2,\omega,\omega,\omega^2,\omega^2,\omega,1,1,1,1,\omega,\omega^2)^T,$$
$$\widetilde{A}_{a^2b} = (1,1,1,1,1,\omega^2,\omega,\omega^2,\omega,1,1,\omega^2,\omega,\omega,\omega^2)^T,$$
$$\widetilde{A}_{a^2c} = (1,1,1,\omega^2,\omega,1,1,\omega^2,\omega,\omega,\omega^2,\omega,\omega^2,1,1)^T,$$
$$\widetilde{A}_{ab^2} = (1,1,1,1,1,\omega,\omega^2,\omega,\omega^2,1,1,\omega,\omega^2,\omega^2,\omega)^T,$$
$$\widetilde{A}_{abc} = (1,1,1,\omega^2,\omega,\omega^2,\omega,\omega,\omega^2,\omega,\omega^2,1,1,\omega,\omega^2)^T,$$
$$\widetilde{A}_{ac^2} = (1,1,1,\omega,\omega^2,1,1,\omega,\omega^2,\omega^2,\omega,\omega^2,\omega,1,1)^T,$$
$$\widetilde{A}_{b^2c} = (1,1,1,\omega^2,\omega,\omega,\omega^2,1,1,\omega,\omega^2,\omega^2,\omega,\omega^2,\omega)^T,$$
$$\widetilde{A}_{bc^2} = (1,1,1,\omega,\omega^2,\omega^2,\omega,1,1,\omega^2,\omega,\omega,\omega^2,\omega,\omega^2)^T,$$
$$\widetilde{A}_{a^2b^2} = (1,\omega^2,\omega,\omega^2,\omega,1,1,1,1,\omega^2,\omega,1,1,\omega,\omega^2)^T,$$
$$\widetilde{A}_{a^2bc} = (1,\omega^2,\omega,\omega,\omega^2,\omega,\omega^2,1,1,1,1,\omega^2,\omega,1,1)^T,$$
$$\widetilde{A}_{a^2c^c} = (1,\omega^2,\omega,1,1,\omega^2,\omega,1,1,\omega,\omega^2,\omega,\omega^2,\omega^2,\omega)^T,$$
$$\widetilde{A}_{ab^2c} = (1,\omega^2,\omega,\omega,\omega^2,1,1,\omega^2,\omega,1,1,\omega,\omega^2,\omega,\omega^2)^T,$$
$$\widetilde{A}_{abc^2} = (1,\omega^2,\omega,1,1,\omega,\omega^2,\omega^2,\omega,\omega,\omega^2,1,1,1,1)^T,$$
$$\widetilde{A}_{b^2c^2} = (1,\omega^2,\omega,1,1,1,1,\omega,\omega^2,\omega,\omega^2,\omega^2,\omega,\omega,\omega^2)^T,$$

$$\widetilde{A}_{a^2b^2c} = (1,\omega,\omega^2,1,1,\omega^2,\omega,\omega,\omega^2,\omega^2,\omega,1,1,1,1)^T,$$
$$\widetilde{A}_{a^2bc^2} = (1,\omega,\omega^2,\omega^2,\omega,1,1,\omega,\omega^2,1,1,\omega^2,\omega,\omega^2,\omega)^T,$$
$$\widetilde{A}_{ab^2c^2} = (1,\omega,\omega^2,\omega^2,\omega,\omega^2,\omega,1,1,1,1,\omega,\omega^2,1,1)^T,$$
$$\widetilde{A}_{a^2b^2c^2} = (1,1,1,\omega,\omega^2,\omega,\omega^2,\omega^2,\omega,\omega^2,\omega,1,1,\omega^2,\omega)^T.$$

Si

$$\mathbf{r} = (1,0,0,1,0,1,1,0,1,0,0,1,1,1,1,0,1,1,0,0,0,0,1,0,1,0,1),$$

entonces

Paso 1. *Tenemos que* $\widetilde{S}_j(\mathbf{r}) \neq 0$ *para todo* $j \in \{1,\ldots,15\} \setminus \{1,10,11,12,13\}$ *y*

$$\widetilde{S} = (0,\omega,\omega^2,\omega^2,\omega,1,1,\omega,\omega^2,0,0,0,0,\omega^2,\omega)^T.$$

Paso 2. *Si tomamos el* $2-$*subconjunto* $\{a^2b, a^2bc\}$*, entonces*

$$\widetilde{\mathcal{E}}(a^2b, a^2bc) = \left(\begin{array}{cc|c} \widetilde{A}_{a^2b} & \widetilde{A}_{a^2bc} & \widetilde{S} \end{array} \right)$$

tiene rango igual a 2. Ahora la solución del sistema lineal

$$X_1\widetilde{A}_{a^2b} + X_2\widetilde{A}_{a^2bc} = \widetilde{S}$$

es $X_1 = X_2 = 1$.

Por lo tanto, el error es

$$\mathbf{e} = a^2b + a^2bc = (0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0).$$

En este ejemplo, el algoritmo general busca $2-$*subconjuntos de* G *cuyas matrices extendidas tienen rango* 2*. Estas matrices tienen* $27 \times 8 = 216$ *filas. Sin embargo, la variante del algoritmo general para el caso abeliano calcula el rango de matrices con* 15 *filas. Claramente, esta nueva versión es más eficiente.*

Capítulo 3

Otros algoritmos de descodificación

En este capítulo presentamos dos algoritmos de descodificación para códigos grupo en el caso semisimple. El primero generaliza el algoritmo de Meggitt para códigos cíclicos y el segundo, aunque muy similar, es una mejora del algoritmo diseñado en el capítulo anterior.

3.1. Consideraciones previas

Como en el Capítulo 2, asumiremos que $G = \{g_1, \ldots, g_n\}$ es un grupo finito de orden n y que $\mathbb{K}$ es un cuerpo finito cuya característica no divide a n. Esto implica que $\mathbb{K}G$ puede escribirse como la suma de s ideales biláteros minimales, cada uno de ellos, generado por un idempotente central primitivo. Si $\mathfrak{C}$ es un código grupo, está generado por un idempotente central $e_0 \in \mathbb{K}G$ que es suma de $m < s$ idempotentes centrales primitivos. Denotaremos n, k y d, la longitud, la dimensión y la distancia mínima de $\mathfrak{C}$, respectivamente.

Con la notación anterior, $\mathfrak{C}^+$ denota el complemento directo de $\mathfrak{C}$. Es decir, $\mathfrak{C}^+$ es un ideal bilátero de dimensión $n-k$ tal que $\mathbb{K}G = \mathfrak{C} \oplus \mathfrak{C}^+$. El ideal $\mathfrak{C}^+$ está generado por el idempotente central $e_0^+ = 1 - e_0$, el cual, es ortogonal a $\mathfrak{C}$. Por tanto, $\mathbf{z} \in \mathbb{K}G$ es una palabra código si y solo si es de la forma $\mathbf{z} = \mathbf{z}'e_0$, para algún $\mathbf{z}' \in \mathbb{K}G$ y, en consecuencia, $\mathbf{z} \in \mathfrak{C}$ si y solo si $\mathbf{z}e_0^+ = 0$.

Definición 3.1.1. *Dado $\mathbf{z} \in \mathbb{K}G$, el $\mathbb{K}G-$síndrome suma (si no hay confusión, el $\mathbb{K}G-$síndrome) de $\mathbf{z}$ es el elemento de $\mathbb{K}G$ dado por*

$$S^+(\mathbf{z}) := \mathbf{z}e_0^+.$$

Si $\mathbf{c} \in \mathfrak{C}$ es la palabra enviada, $\mathbf{e} \in \mathbb{K}G$ es el error que se produce durante la transmisión y $\mathbf{r} = \mathbf{c} + \mathbf{e}$ es la palabra recibida, entonces

$$S^+(\mathbf{r}) = (\mathbf{c} + \mathbf{e})e_0^+ = \mathbf{e}e_0^+.$$

Notemos que no hay errores ($\mathbf{e} = 0$) si y solo si $S^+(\mathbf{r}) = 0$. Bajo la suposición que t es la capacidad correctora de $\mathfrak{C}$ y de que durante el proceso de transmisión han ocurrido a lo sumo t errores, *descodificar por $\mathbb{K}G-$síndrome* consiste en encontrar una solución particular de la ecuación $\mathbf{x}e_0^+ = S^+(\mathbf{r})$. La ecuación anterior se denomina *ecuación clave* y si tiene una solución con peso menor o igual a t, entonces esta solución es única (ver Teorema 2.1.2 del Capítulo 2).

Teorema 3.1.2. *Dado un código grupo $\mathfrak{C}$ que corrige hasta t errores, si $\mathbf{r}$ es una palabra recibida y $S^+(\mathbf{r})$ es su $\mathbb{K}G-$síndrome, entonces existe a lo sumo un elemento en $\mathbb{K}G$ con peso $w \leq t$ que es solución de la ecuación clave $\mathbf{x}e_0^+ = S^+(\mathbf{r})$.*

Demostración. En efecto, si $\mathbf{e}, \mathbf{e}' \in \mathbb{K}G$ son dos soluciones distintas de la ecuación clave con pesos $w, w' \leq t$, respectivamente, entonces $\mathbf{e}e_0^+ = \mathbf{e}'e_0^+ = S^+(\mathbf{r})$. Así, $(\mathbf{e} - \mathbf{e}')e_0^+ = 0$ y, por lo tanto, $\mathbf{e} - \mathbf{e}' \neq 0$ es un elemento de $\mathfrak{C}$. Esto es una contradicción, puesto que el peso mínimo de $\mathfrak{C}$ es $d \geq 2t + 1$ y $\mathbf{e} - \mathbf{e}'$ tiene un peso menor o igual a $w + w' \leq 2t$. □

De hecho, si $e_1, \ldots, e_m \in \mathbb{K}G$ son los idempotentes centrales primitivos de $\mathbb{K}G$ que son ortogonales a $\mathfrak{C}$, $\mathbf{z} \in \mathbb{K}G$ y $\{S_1(\mathbf{z}), \ldots, S_m(\mathbf{z})\}$ es el conjunto de $\mathbb{K}G-$síndromes de $\mathbf{z}$ de la Definición 2.1.1 y $\mathbf{u} \in \mathbb{K}G$ es solución del sistema

$$\begin{aligned} \mathbf{x}e_1 &= S_1(\mathbf{z}), \\ &\vdots \\ \mathbf{x}e_m &= S_m(\mathbf{z}), \end{aligned} \tag{3.1}$$

entonces $\mathbf{u}$ es solución del sistema $\mathbf{x}e_0^+ = S^+(\mathbf{z})$. En efecto,

$$\mathbf{u}e_0^+ = \mathbf{u}(e_1 + \cdots + e_m) = S_1(\mathbf{z}) + \cdots + S_m(\mathbf{z}) = S^+(\mathbf{z}),$$

puesto que $e_0^+ = e_1 + \cdots + e_m$. Pero también se cumple el recíproco. Es decir, si $\mathbf{u}e_0^+ = S^+(\mathbf{z})$, entonces $\mathbf{u}(e_1 + \cdots + e_m) = \mathbf{z}(e_1 + \cdots + e_m)$ y obtenemos que

$$\sum_{h=1}^{m} (\mathbf{u} - \mathbf{z})e_h = 0.$$

Como $\mathbb{K}G$ es semisimple, se sigue que $(\mathbf{u} - \mathbf{z})e_h = 0$ para todo $h \in \{1, \ldots, m\}$ y así $\mathbf{u}$ satisface el conjunto de ecuaciones dado por (3.1).

Lo anterior nos quiere decir que cualquier algoritmo de descodificación que use el $\mathbb{K}G-$síndrome (Definición 3.1.1) es equivalente al algoritmo de descodificación del Capítulo 2. A continuación presentaremos dos algoritmos de descodificación de este tipo.

3.2. Generalización del algoritmo de Meggitt

Los códigos cíclicos fueron introducidos por E. Prange en [53]. Constituyen una familia de códigos lineales cuya estructura algebraica permite obtener muchas propiedades y una gran variedad de métodos de descodificación (ver [39] y [54]).

Como cada código cíclico de longitud n se puede identificar con un $G-$código (es decir, es equivalente a un $G-$código), donde G es un grupo cíclico de orden n, los códigos grupo generalizan de manera natural los códigos cíclicos. En [46] y [47], J. E. Meggitt presentó un algoritmo de descodificación para códigos cíclicos (ver final Sección 1.1). A continuación mostraremos cómo extender tal algoritmo a códigos grupo.

Para ello, observemos que si $\mathbf{z}, \mathbf{z}' \in \mathbb{K}G$ son elementos distintos y tienen pesos respectivos $w, w' \leq t$, entonces $S^+(\mathbf{z}) \neq S^+(\mathbf{z}')$. En efecto, si $S^+(\mathbf{z}) = S^+(\mathbf{z}')$, entonces $(\mathbf{z}-\mathbf{z}')e_0^+ = 0$ y, como en la demostración del Teorema 3.1.2, obtenemos una contradicción. Por otro lado, notemos que todo elemento no nulo $\alpha_1 g_1 + \cdots + \alpha_n g_n \in \mathbb{K}G$ puede ser expresado de la forma

$$g_j \left(\alpha_j g_1 + \sum_{g_i \neq 1_G} \alpha_i' g_i \right),$$

donde $g_1 = 1_G$, $g_j \in G$ y α_j es un elemento no nulo de $\mathbb{K}$.

Definición 3.2.1. *Sea $\mathcal{T}$ el conjunto de todos los elementos de $\mathbb{K}G$ que tienen peso menor o igual a t y cuyo soporte contiene la posición 1_G. Es decir,*

$$\mathcal{T} = \{\mathbf{z} \in \mathbb{K}G : wt(\mathbf{z}) \leq t \ y \ 1_G \in Supp(\mathbf{z})\}.$$

Los elementos de $\mathcal{T}$ se denominan representantes y, para cada $u \in \{1, \ldots, n\}$, el conjunto $\mathcal{R}_u$ de todos los pares $(\mathbf{z}, S^+(\mathbf{z}))$ tales que $\mathbf{z} \in \mathcal{T}$ y $wt(S^+(\mathbf{z})) = u$, se denomina lista reducida de $\mathbb{K}G-$síndromes con peso u.

Supongamos que $\mathbf{r} = \mathbf{c} + \mathbf{e}$, donde $\mathbf{c} \in \mathfrak{C}$ y $\mathbf{e} \in \mathbb{K}G$. Si $\mathrm{wt}(\mathbf{e}) \leq t$, entonces existen $g_j \in G$ y $\mathbf{e}' \in \mathcal{T}$ tales que $\mathbf{e} = g_j \mathbf{e}'$ y así, $g_i S^+(\mathbf{r}) = \mathbf{e}' e_0^+$ donde $g_i = g_j^{-1}$. Además, si $v = \mathrm{wt}(S^+(\mathbf{r}))$, entonces $(\mathbf{e}', g_i S^+(\mathbf{r})) \in \mathcal{R}_v$. Recíprocamente, si

$g_i \in G$, $\mathbf{e}' \in \mathcal{T}$ y $g_i S^+(\mathbf{r}) = \mathbf{e}' e_0^+$, entonces $S^+(\mathbf{r}) = (g_i^{-1}\mathbf{e}')e_0^+$ y $\mathrm{wt}(g_i^{-1}\mathbf{e}') \leq t$. En virtud del Teorema 3.1.2, tenemos que $g_i^{-1}\mathbf{e}'$ es el único elemento de $\mathbb{K}G$ que satisface la ecuación clave $\mathbf{x}e_0^+ = S^+(\mathbf{r})$. La elección de $g_i \in G$ y $\mathbf{e}' \in \mathcal{T}$ no necesariamente es única. Sin embargo, $\mathbf{e} = g_i^{-1}\mathbf{e}'$ es el error producido al enviar $\mathbf{c} \in \mathfrak{C}$.

3.2.1. Algoritmo generalizado de Meggitt (Algoritmo GMD)

Para cada $u \in \{1, \ldots, n\}$, se calcula la lista $\mathcal{R}_u$. Una vez se ha recibido la palabra $\mathbf{r} \in \mathbb{K}G$, se realizan los siguientes pasos.

Paso 1.

Se calcula $S^+(\mathbf{r})$, el $\mathbb{K}G-$síndrome de $\mathbf{r}$. Si $S^+(\mathbf{r}) = 0$, entonces no hay errores y el algoritmo finaliza. De lo contrario, se considera la lista $\mathcal{R}_v$, donde $v = \mathrm{wt}(S^+(\mathbf{r}))$ y se procede al siguiente paso.

Paso 2.

Se toma aleatoriamente $g_i \in G$ y se calcula $S_{g_i}(\mathbf{r})^+ := g_i S^+(\mathbf{r})$.

a. Si $S_{g_i}^+(\mathbf{r})$ es el $\mathbb{K}G-$síndrome de algún representante $\mathbf{e}'$ en $\mathcal{R}_v$, es decir $(\mathbf{e}', S_{g_i}^+(\mathbf{r})) \in \mathcal{R}_v$, entonces el error es $\mathbf{e} = g_i^{-1}\mathbf{e}'$ y el algoritmo termina.

b. De lo contrario, se descarta el elemento g_i y se repite el Paso 2 con otro elemento de G.

El algoritmo termina cuando encuentra un elemento $g_i \in G$ tal que

$$S_{g_i}^+(\mathbf{r}) \textit{ es el } \mathbb{K}G-\textit{síndrome de algún representante en } \mathcal{R}_v \tag{3.2}$$

o cuando se han considerado todos los elementos de G y ninguno satisface (3.2). En el último caso, el número de errores es mayor a t y el código grupo no puede corregir la palabra $\mathbf{r}$.

Ejemplo 3.2.2. *Sean $\mathbb{K} = \mathbb{F}_2$ y $G = C_7 \times C_7$, donde $G = \langle a, b \rangle$. Si en $\mathbb{F}_2(C_7 \times C_7)$ fijamos la base*

$$\begin{aligned}\{&1, a, b, a^2, ab, b^2, a^3, a^2b, ab^2, b^3, a^4, a^3b, a^2b^2, ab^3, b^4,\\ &a^5, a^4b, a^3b^2, a^2b^3, ab^4, b^5, a^6, a^5b, a^4b^2, a^3b^3, a^2b^4, ab^5,\\ &b^6, a^6b, a^5b^2, a^4b^3, a^3b^4, a^2b^5, ab^6, a^6b^2, a^5b^3, a^4b^4, a^3b^5,\\ &a^2b^6, a^6b^3, a^5b^4, a^4b^5, a^3b^6, a^6b^4, a^5b^5, a^4b^6, a^6b^5, a^5b^6, a^6b^6\},\end{aligned}$$

entonces las componentes simples de $\mathbb{F}_2(C_7 \times C_7)$ *están generadas por los siguientes idempotentes centrales primitivos:*

$$e_1 = (111111111111111111111111111111111111111111111111),$$
$$e_2 = (111111111011101111010111010011010010100010010000),$$
$$e_3 = (111111011110111010111001011100101100101001000100 0),$$
$$e_4 = (111111000011111000000000000011111111111111000110),$$
$$e_5 = (111101010010011001110011101011010001001001111 1101),$$
$$e_6 = (111101001011001011100010111000101110010110011101 1),$$
$$e_7 = (110110010111100001011011100100101111100010111010 1),$$
$$e_8 = (110110001110100011101010011111010001110100110111 0),$$
$$e_9 = (110100100110010100101100101100101101011101101111 1),$$
$$e_{10} = (110100011111010010011001110111010010011111001001 1),$$
$$e_{11} = (101011110000101101110111001000101101110100110111 0),$$
$$e_{12} = (101011101000111110100100111011010000111101001101 1),$$
$$e_{13} = (101001111001011110010101110000101111001011101010 1),$$
$$e_{14} = (101001100101001101001110100111010011010110111011 1),$$
$$e_{15} = (100010110101110100111101001111010011101011100110 0),$$
$$e_{16} = (100010101101110111001110010100101110111111010001 0),$$
$$e_{17} = (100000111100000111111111111111111100000000011100 1).$$

Supongamos que $\mathfrak{C} = \langle e_0 \rangle$, *donde* $e_0 = e_2 + e_{13} + e_{14} + e_{15} + e_{16} + e_{17}$, *es decir,*

$$e_0 = (011111000011111000000000100000101111101011101010 1).$$

Entonces $n = 49$, $k = 18$ *y* $d = 12$. *Si*

$$\mathbf{r} = (000100010111001101100111101110001010101100111010 0),$$

entonces, descodificando con este algoritmo tenemos que:

Paso 1. $S^+(\mathbf{r}) = (111110111111011111011111011001011000101001100111 0) \neq \mathbf{0}$. *Como* $wt(S^+(\mathbf{r})) = 33$, *consideramos* $\mathcal{R}_{33}$ *y vamos al*

Paso 2. *Si tomamos* $a^3b \in G$, *entonces*

$$S^{+}_{a^3b}(\mathbf{r}) = (111011011101010111001111000111110010111011011110),$$

tiene representante $\mathbf{e}' = 1 + a^2b^2 + a^4b^5 + a^6b^4 + a^5b^6$. *Esto implica que el error es:*

$$\begin{aligned}
\mathbf{e} &= (a^3b)^{-1}\mathbf{e}' \\
&= a^4b^6(1 + a^2b^2 + a^4b^5 + a^6b^4 + a^5b^6) \\
&= ab^4 + a^3b^3 + a^6b + a^2b^5 + a^4b^6 \\
&= (000000000000000000010000100010001000000000001000).
\end{aligned}$$

3.2.2. Análisis de complejidad

Antes de aplicar el Algoritmo GMD, se considera el conjunto $\mathcal{T}$. Tal conjunto tiene tamaño

$$\sum_{j=0}^{t-1} \binom{n-1}{j} (q-1)^{j+1}. \tag{3.3}$$

Luego, se calcula el $\mathbb{K}G-$síndrome de todos los elementos de $\mathcal{T}$ y, al mismo tiempo, se construye la lista $\mathcal{R}_u$ para cada $u \in \{1, \ldots, n\}$. La cantidad de operaciones que requiere este proceso es el producto de nt multiplicado por (3.3). Así, el orden de complejidad para obtener las listas $\mathcal{R}_1, \ldots, \mathcal{R}_n$ es

$$\mathcal{O}\left(nt \times (q-1)^t \times \binom{n-1}{t-1}\right). \tag{3.4}$$

El peor caso de este algoritmo podría ocurrir cuando los $\mathbb{K}G-$síndromes de todos los elementos de $\mathcal{T}$ tienen el mismo peso v. En ese caso, solo hay una lista reducida no vacía $\mathcal{R}_v$ que tiene tamaño (3.3).

Antes de iniciar el proceso de descodificación, se ordena la lista $\mathcal{R}_v$. Esto se hace con el fin de obtener una descodificación más eficiente. Si utilizamos un algoritmo de ordenamiento eficiente (por ejemplo, Mergesort que tiene un orden de complejidad $\mathcal{O}(N \log N)$, donde N es el tamaño de la lista a ordenar y log se calcula en base 2), el reordenamiento de $\mathcal{R}_v$ tiene un orden de complejidad

$$\mathcal{O}\left((q-1)^t \times \binom{n-1}{t-1} \times \log\left[(q-1)^t \times \binom{n-1}{t-1}\right]\right). \tag{3.5}$$

En el paso 1, calcular $S^{+}(\mathbf{r})$ requiere $3n^2$ operaciones. Verificar si $S^{+}(\mathbf{r}) = 0$ ó $S^{+}(\mathbf{r}) \neq 0$ requiere n operaciones. Con lo cual, tenemos orden de complejidad $\mathcal{O}(n^2)$.

En el paso 2, se aplica el algoritmo de búsqueda binaria que tiene orden de complejidad $\mathcal{O}(\log|\mathcal{R}_v|)$. La comparación de n componentes por cada uno de los $G-$múltiplos de $S^+(\mathbf{r})$ en la lista reducida $\mathcal{R}_v$ tiene orden de complejidad

$$\mathcal{O}\left(n^2 \times \log\left[(q-1)^t \times \binom{n-1}{t-1}\right]\right). \tag{3.6}$$

3.3. Descodificación por $\mathbb{K}G-$síndrome

El segundo algoritmo propuesto en este capítulo se diseña en dos fases. La primera también está inspirada en el conocido algoritmo de descodificación por síndrome para códigos lineales y sustituye el conjunto de $\mathbb{K}G-$síndromes del Algoritmo SSD por un único $\mathbb{K}G-$síndrome.

Antes de iniciar el proceso de *descodificación por* $\mathbb{K}G-$*síndrome*, calculamos $g_i e_0^+$ para $i \in \{1,\ldots,n\}$. Con esto, definimos $C_{g_i}^+ \in M_{n\times 1}(\mathbb{K})$ como el vector columna de coeficientes de $g_i e_0^+$ con respecto a la base $B(=G)$. Si definimos

$$\mathcal{C}^+(g_{i_1},\ldots,g_{i_b}) := \begin{pmatrix} C_{g_{i_1}}^+ & \cdots & C_{g_{i_b}}^+ \end{pmatrix},$$

tenemos el siguiente resultado (ver Proposición 2.1.3 del Capítulo 2).

Proposición 3.3.1. *Si $b < d$ y $g_{i_1},\ldots,g_{i_b}$ son elementos distintos de G, entonces la matriz*

$$\mathcal{C}^+(g_{i_1},\ldots,g_{i_b}) \in M_{n\times b}(\mathbb{K}),$$

tiene rango igual a b.

Demostración. Si el rango de $\mathcal{C}^+(g_{i_1},\ldots,g_{i_b})$ es menor a b, entonces hay b elementos $\nu_1,\ldots,\nu_b \in \mathbb{K}$, no todos nulos tales que $\nu_1 C_{g_{i_1}}^+ + \cdots + \nu_b C_{g_{i_b}}^+ = 0$. Así, $(\nu_1 g_{i_1} + \cdots + \nu_b g_{i_b})e_0^+ = 0$ y en consecuencia $\mathbf{z} = \nu_1 g_{i_1} + \cdots + \nu_b g_{i_b} \in \mathfrak{C}$. Este hecho es una contradicción, puesto que $\mathbf{z} \neq 0$ tiene un peso menor o igual que b y la distancia mínima de $\mathfrak{C}$ es igual a d. □

Cuando se recibe una palabra $\mathbf{r} \in \mathbb{K}G$, este nuevo algoritmo calcula $S^+(\mathbf{r})$ y el vector columna de sus coeficientes (con respecto a B). Tal vector será denotado por V^+. Entonces, el propósito del algoritmo es encontrar $\mathbf{e} = \alpha_1 g_{i_1} + \cdots + \alpha_w g_{i_w}$ con peso $w \leq t$ tal que $S^+(\mathbf{r}) = \mathbf{e}e_0^+$. Como no sabemos el número de errores ni tampoco sus posiciones, el algoritmo buscará un $t-$subconjunto de G que contenga todas las posiciones de error. Es decir, busca un elemento $\beta_1 g_{i_1} + \cdots + \beta_t g_{i_t} \in \mathbb{K}G$ tal que, $\beta_i = \alpha_i$ si g_{i_j} es una posición de error y $\beta_i = 0$ en caso contrario.

Con lo anterior, el algoritmo busca $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G tal que el sistema lineal de la forma

$$X_1 C^+_{g_{i_1}} + \cdots + X_t C^+_{g_{i_t}} = V^+. \tag{3.7}$$

que admita una solución. Por el Teorema 3.1.2, cualquier sistema de este tipo tiene una solución única o no tiene soluciones. Así, el algoritmo buscará un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G tal que (3.7) sea compatible determinado y, por lo tanto, que tenga una solución única. Como sabemos, esto sucede si y solo si la matriz $\mathcal{C}(g_{i_1}, \ldots, g_{i_t})$ y la matriz extendida

$$\mathcal{M}^+(g_{i_1}, \ldots, g_{i_t}) := \left(\begin{array}{ccc|c} C^+_{g_{i_1}} & \cdots & C^+_{g_{i_t}} & V^+ \end{array} \right),$$

tienen rango igual a t. Por la Proposición 3.3.1, basta verificar que $\mathcal{M}(g_{i_1}, \ldots, g_{i_t})$ tiene rango igual a t.

3.3.1. Algoritmo de descodificación por $\mathbb{K}G-$síndrome (Algoritmo SD)

Paso 1.

Recibida la palabra $\mathbf{r} \in \mathbb{K}G$, se calcula su $\mathbb{K}G-$síndrome. Si $S^+(\mathbf{r}) = 0$, entonces no hay errores y el algoritmo finaliza. De lo contrario, se continúa al siguiente paso.

Paso 2.

Se selecciona aleatoriamente un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G. Se considera la matriz $\mathcal{M}^+(g_{i_1}, \ldots, g_{i_t})$ y se calcula su rango.

a. Si el rango es igual a t, se halla la (única) solución del sistema (3.7). Si $\beta_{j_1}, \ldots, \beta_{j_w}$ son los elementos no nulos de la solución anterior, entonces el error es $\mathbf{e} = \beta_{j_1} g_{i_{j_1}} + \cdots + \beta_{j_w} g_{i_{j_w}}$ y el algoritmo termina.

b. De lo contrario, se descarta el $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$, se considera aleatoriamente otro $t-$subconjunto de G y se repite el Paso 2.

El algoritmo termina cuando encuentra un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G tal que

$$\text{Rank}(\mathcal{M}^+(g_{i_1}, \ldots, g_{i_t})) = t, \tag{3.8}$$

o cuando todos los $t-$subconjuntos de G han sido considerados y ninguno satisface (3.8). En tal caso, el número de errores producidos durante la transmisión es mayor a t, y la palabra recibida $\mathbf{r}$ no puede ser corregida por el código grupo.

Ejemplo 3.3.2. *Consideremos el código grupo* $\mathfrak{C}$ *de* $\mathbb{F}_5\mathcal{S}_4$ *del Ejemplo 2.1.4. Recordemos que* $\mathfrak{C}$ *está generado por*

$$e_0 = e_5 = (1,2,2,0,0,2,2,3,0,3,3,0,0,3,2,0,3,3,3,0,0,2,3,3).$$

Previo al proceso de descodificación, calculamos que

$$\begin{aligned}
C^+_{g_1} &= (0,3,3,0,0,3,3,2,0,2,2,0,0,2,3,0,2,2,2,0,0,3,2,2),\\
C^+_{g_2} &= (3,0,0,3,3,0,2,3,2,0,0,2,2,0,0,3,2,2,0,2,3,0,2,2),\\
C^+_{g_3} &= (3,0,0,3,3,0,0,2,3,0,2,2,3,2,0,2,2,0,0,2,2,2,0,3),\\
C^+_{g_4} &= (0,3,3,0,0,3,2,0,0,3,2,2,2,3,2,0,0,2,2,0,2,2,3,0),\\
C^+_{g_5} &= (0,3,3,0,0,3,2,0,2,2,3,0,0,2,2,2,0,3,3,2,0,2,2,0),\\
C^+_{g_6} &= (3,0,0,3,3,0,0,2,2,2,0,3,2,0,2,2,3,0,2,3,2,0,0,2),\\
C^+_{g_7} &= (3,2,0,2,2,0,0,3,3,0,0,3,3,0,0,2,2,2,0,3,2,0,2,2),\\
C^+_{g_8} &= (2,3,2,0,0,2,3,0,0,3,3,0,0,3,2,0,2,2,3,0,0,2,2,2),\\
C^+_{g_9} &= (0,2,3,0,2,2,3,0,0,3,3,0,0,2,3,2,0,2,2,2,0,2,3,0),\\
C^+_{g_{10}} &= (2,0,0,3,2,2,0,3,3,0,0,3,2,0,2,3,2,0,2,2,2,0,0,3),\\
C^+_{g_{11}} &= (2,0,2,2,3,0,0,3,3,0,0,3,2,2,0,2,3,0,0,2,3,2,0,2),\\
C^+_{g_{12}} &= (0,2,2,2,0,3,3,0,0,3,3,0,2,2,2,0,0,3,2,0,2,3,2,0),\\
C^+_{g_{13}} &= (0,2,3,2,0,2,3,0,0,2,2,2,0,3,3,0,0,3,3,0,2,2,2,0),\\
C^+_{g_{14}} &= (2,0,2,3,2,0,0,3,2,0,2,2,3,0,0,3,3,0,0,3,2,2,0,2),\\
C^+_{g_{15}} &= (3,0,0,2,2,2,0,2,3,2,0,2,3,0,0,3,3,0,2,2,3,0,0,2),\\
C^+_{g_{16}} &= (0,3,2,0,2,2,2,0,2,3,2,0,0,3,3,0,0,3,2,2,0,3,2,0),\\
C^+_{g_{17}} &= (2,2,2,0,0,3,2,2,0,2,3,0,0,3,3,0,0,3,2,0,0,2,3,2),\\
C^+_{g_{18}} &= (2,2,0,2,3,0,2,2,2,0,0,3,3,0,0,3,3,0,0,2,2,0,2,3),\\
C^+_{g_{19}} &= (2,0,0,2,3,2,0,3,2,2,0,2,3,0,2,2,2,0,0,3,3,0,0,3),\\
C^+_{g_{20}} &= (0,2,2,0,2,3,3,0,2,2,2,0,0,3,2,2,0,2,3,0,0,3,3,0),\\
C^+_{g_{21}} &= (0,3,2,2,0,2,2,0,0,2,3,2,2,2,3,0,0,2,3,0,0,3,3,0),\\
C^+_{g_{22}} &= (3,0,2,2,2,0,0,2,2,0,2,3,2,2,0,3,2,0,0,3,3,0,0,3),\\
C^+_{g_{23}} &= (2,2,0,3,2,0,2,2,3,0,0,2,2,0,0,2,3,2,0,3,3,0,0,3),\\
C^+_{g_{24}} &= (2,2,3,0,0,2,2,2,0,3,2,0,0,2,2,0,2,3,3,0,0,3,3,0).
\end{aligned}$$

Si

$$\mathbf{r} = (3,2,0,0,1,2,0,4,0,4,1,1,1,4,2,1,4,0,2,3,0,0,3,2),$$

para recuperar la palabra código enviada, seguimos los pasos:

Paso 1. *Calculamos el síndrome* $S^+(\mathbf{r})$*. En este caso,*

$$S^+(\mathbf{r}) = (1,0,2,4,0,3,2,1,0,0,3,4,0,0,0,0,0,0,2,0,4,0,3,1).$$

Paso 2. *Tomando el* $3-$*subconjunto* $\{g_4, g_{13}, g_{17}\}$*, vemos que la matriz*

$$\mathcal{M}^+(g_4, g_{13}, g_{17}) = \left(\begin{array}{ccc|c} C^+_{g_4} & C^+_{g_{13}} & C^+_{g_{17}} & V^+ \end{array} \right)$$

tiene rango igual a 3, y $X_1 = 0$*,* $X_2 = 2$ *y* $X_3 = 3$ *es la solución del sistema*

$$X_1 C^+_{g_4} + X_2 C^+_{g_{13}} + X_3 C^+_{g_{17}} = V^+,$$

y en consecuencia

$$\mathbf{e} = 2g_{13} + 3g_{17} = (0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,3,0,0,0,0,0,0,0,0)$$

Así, la palabra código enviada es

$$\mathbf{c} = (3,2,0,0,1,2,0,4,0,4,1,1,4,4,2,1,1,0,2,3,0,0,3,2).$$

3.3.2. Análisis de la complejidad

Antes de iniciar la descodificación con el Algoritmo SD, se calculan todos los productos $g_i e_0^+$ para $i \in \{1, \ldots, n\}$. Tal proceso requiere n^2 operaciones en total.

En el primer paso, se calcula $S^+(\mathbf{r})$ con una cantidad de $3n^2$ operaciones necesarias.

En caso que $S^+(\mathbf{r}) \neq 0$, se selecciona un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G. El orden de complejidad para calcular el rango de $\mathcal{M}^+(g_{i_1}, \ldots, g_{i_t})$ es $\mathcal{O}(nt^2)$. Dado que el algoritmo realiza la búsqueda en todos los $t-$subconjuntos de G, entonces el orden de complejidad del paso 2 es

$$\mathcal{O}\left(nt^2 \times \binom{n}{t}\right). \tag{3.9}$$

Del mismo modo, si el error tiene peso $w < t$, la probabilidad de encontrar un $t-$subconjunto de G que contenga a su soporte aumenta cuando w disminuye. En estos casos, el algoritmo también será más eficiente.

3.4. Otro punto de vista

Aunque el Algoritmo SD y la versión para el caso abeliano del Algoritmo SSD tienen el mismo orden de complejidad, este último es más eficiente dado que trabaja con matrices más pequeñas. Por tal razón, presentamos una versión mejorada del algoritmo de descodificación por $\mathbb{K}G-$síndrome.

Conservando las notaciones de la sección anterior, $\alpha_1 g_1 + \cdots + \alpha_n g_n \in \mathbb{K}G$ es una solución de $\mathbf{x}e_0^+ = S^+(\mathbf{r})$ si y solo si

$$\alpha_1 C_{g_1}^+ + \cdots + \alpha_n C_{g_n}^+ = V^+. \tag{3.10}$$

Por el Teorema 3.1.2, sabemos que si existe $\mathbf{e} = \alpha_1 g_1 + \cdots + \alpha_n g_n$ con peso $w \leq t$ que es solución de $\mathbf{x}e_0^+ = S^+(\mathbf{r})$, entonces $\mathbf{e}$ es el único elemento de $\mathbb{K}G$ que satisface estas dos condiciones. Esto ocurre si y solo si hay exactamente w escalares $\alpha_{i_1}, \ldots, \alpha_{i_w} \in \{\alpha_1, \ldots, \alpha_n\}$ no nulos tales que

$$\alpha_{i_1} C_{g_{i_1}}^+ + \cdots + \alpha_{i_w} C_{g_{i_w}}^+ = V^+.$$

Por tanto, nuestro objetivo es encontrar una solución $X = (X_1, \ldots, X_n)$, con peso $w \leq t$, del sistema

$$X_1 C_{g_1}^+ + \cdots + X_n C_{g_n}^+ = V^+. \tag{3.11}$$

Proposición 3.4.1. *Las matrices*

$$\mathcal{C}^+(g_1, \ldots, g_n) = \begin{pmatrix} C_{g_1}^+ & \cdots & C_{g_n}^+ \end{pmatrix} \in M_{n\times n}(\mathbb{K}),$$

y

$$\mathcal{M}^+(g_1, \ldots, g_n) = \left(\begin{array}{ccc|c} C_{g_1}^+ & \cdots & C_{g_n}^+ & V^+ \end{array}\right) \in M_{n\times(n+1)}(\mathbb{K}),$$

tienen rango igual a $n - k$.

Demostración. En efecto, cada elemento de $\mathfrak{C}^+$ es de la forma

$$(\mu_1 g_1 + \cdots + \mu_n g_n)e_0^+ = \mu_1 g_1 e_0^+ \cdots + \mu_n g_n e_0^+,$$

donde $\mu_i \in \mathbb{K}$ para todo $i \in \{1 \ldots, n\}$. Entonces $\{g_1 e_0^+, \ldots, g_n e_0^+\}$ genera a $\mathfrak{C}^+$ y como $\mathfrak{C}^+$ tiene dimensión $n-k$, entonces el rango de la matriz $\mathcal{C}^+(g_1, \ldots, g_n)$ es exactamente $n - k$. La matriz $\mathcal{M}^+(g_1, \ldots, g_n)$ también tiene rango $n - k$ puesto que V^+ es el vector columna de coeficientes de $S^+(\mathbf{r})$, que es un elemento de $\mathfrak{C}^+$. □

Antes de continuar, recordaremos la siguiente noción.

Definición 3.4.2. *Sean $\mathcal{U}$ y $\widetilde{\mathcal{U}}^R$ dos matrices del mismo tamaño. Se dice que $\widetilde{\mathcal{U}}^R$ es la matriz escalonada reducida (por filas) de $\mathcal{U}$ si es posible obtener $\widetilde{\mathcal{U}}^R$ a partir de $\mathcal{U}$ mediante operaciones elementales de filas y $\widetilde{\mathcal{U}}^R$ satisface las siguientes condiciones:*

1. *La primera entrada no nula (pivote) en cada fila no nula, es igual a 1.*
2. *Si el pivote en la fila i está en la columna $\gamma(i)$, entonces $\gamma(i) < \gamma(i+1)$, para toda $i \geq 1$.*
3. *Si una columna contiene un pivote, este pivote es el único elemento distinto de cero en la columna.*
4. *Cualquier fila nula está en la parte inferior de la matriz.*

Si $\widetilde{\mathcal{M}}^R(g_1, \ldots, g_n)$ es la matriz escalonada reducida de $\mathcal{M}^+(g_1, \ldots, g_n)$, entonces para $i \in \{1, \ldots, n\}$, denotamos la $i-$ésima columna de $\widetilde{\mathcal{M}}^R(g_1, \ldots, g_n)$ por $\widetilde{C}^R_{g_i}$ y su $(n+1)-$ésima columna como $\widetilde{V}^R$.

Para $g_{i_1}, \ldots, g_{i_b} \in G$, definimos la matriz

$$\widetilde{\mathcal{C}}^R(g_{i_1}, \ldots, g_{i_b}) := \begin{pmatrix} \widetilde{C}^R_{g_{i_1}} & \cdots & \widetilde{C}^R_{g_{i_b}} \end{pmatrix}$$

Proposición 3.4.3. *Si $b < d$ y $g_{i_1}, \ldots, g_{i_b}$ son b elementos distintos de G, entonces*

$$\widetilde{\mathcal{C}}^R(g_{i_1}, \ldots, g_{i_b}) \in M_{n \times b}(\mathbb{K}),$$

tiene rango igual a b.

Demostración. Supongamos que $\mu_1 \widetilde{C}^R_{g_{i_1}} + \cdots + \mu_b \widetilde{C}^R_{g_{i_b}} = 0$, donde $\mu_h \in \mathbb{K}$ para todo $h \in \{1, \ldots, b\}$. Como $\widetilde{\mathcal{M}}^R(g_1, \ldots, g_n)$ es la matriz escalonada reducida de $\mathcal{M}^+(g_1, \ldots, g_n)$, existe $\mathcal{U} \in M_{n \times n}(\mathbb{K})$ invertible tal que

$$\mathcal{U} \cdot \mathcal{M}^+(g_1, \ldots, g_n) = \widetilde{\mathcal{M}}^R(g_1, \ldots, g_n).$$

Con lo anterior, $\mathcal{U} \cdot C^+_{g_i} = \widetilde{C}^R_{g_i}$ para todo $i \in \{1, \ldots, n\}$ y en consecuencia

$$\mathcal{U} \cdot (\mu_1 C^+_{g_{i_1}} + \cdots + \mu_b C^+_{g_{i_b}}) = \mathbf{0}.$$

Así, $\mu_1 C^+_{g_{i_1}} + \cdots + \mu_b C^+_{g_{i_b}} = \mathbf{0}$ y como $C^+_{g_{i_1}}, \ldots, C^+_{g_{i_b}}$ son $\mathbb{K}-$linealmente independientes puesto que $\mathcal{C}^+(g_{i_1}, \ldots, g_{i_b})$ tiene rango b (ver Proposición 3.3.1), entonces $\mu_h = 0$ para todo $h \in \{1, \ldots, b\}$. Lo anterior implica que $\widetilde{C}^R_{g_{i_1}}, \ldots, \widetilde{C}^R_{g_{i_b}}$ son $\mathbb{K}-$linealmente independientes y, por lo tanto, $\widetilde{\mathcal{C}}^R(g_{i_1}, \ldots, g_{i_b})$ tiene rango b. □

Como las últimas k filas de $\widetilde{\mathcal{M}}^R(g_1,\ldots,g_n)$ son nulas, podemos omitirlas y denotar por $\mathcal{M}^R(g_1,\ldots,g_n)$, a la matriz resultante. De la misma forma obtenemos $\widetilde{C}^R_{g_i}$, $\widetilde{V}^R$ y $\widetilde{\mathcal{C}}^R(g_{i_1},\ldots,g_{i_b})$ a partir de $C^R_{g_i}$, V^R y $\mathcal{C}^R(g_{i_1},\ldots,g_{i_b})$.

Ejemplo 3.4.4. *En el Ejemplo 3.3.2, tenemos que* $\mathcal{C}^R(g_1,\ldots,g_{24})$ *es igual a*

$$\begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&1&3&0&1\\
0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&4&0&1\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&1&4&0&0\\
0&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&1&4&4&1\\
0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&1&0&4&0&1&0&4&0&1\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&1&1&0&4&0&1&1&3&4&2\\
0&0&0&0&0&0&1&0&0&0&0&4&0&0&4&0&0&1&0&4&0&1&1&4\\
0&0&0&0&0&0&0&1&0&0&0&4&0&0&0&4&0&1&0&4&4&2&1&4\\
0&0&0&0&0&0&0&0&1&0&0&4&0&0&4&0&0&1&0&0&0&1&0&4\\
0&0&0&0&0&0&0&0&0&1&0&4&0&0&0&4&0&1&0&0&0&1&0&4\\
0&0&0&0&0&0&0&0&0&0&1&4&0&0&0&0&0&0&0&0&4&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&4&0&0&0&0&4&0&1&1&4\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&4&0&0&0&4&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&4&0&0&0&0&4&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&4&4&1&1&4
\end{pmatrix}.$$

Así, los pivotes están en las posiciones $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $(5,5)$, $(6,6)$, $(7,7)$, $(8,8)$, $(9,9)$, $(10,10)$, $(11,11)$, $(12,13)$, $(13,14)$, $(14,17)$ *y* $(15,19)$.

Corolario 3.4.5. *Si* $b<d$ *y* $g_{i_1},\ldots,g_{i_b}$ *son* b *elementos distintos de* G*, entonces*

$$\mathcal{C}^R(g_{i_1},\ldots,g_{i_b})\in M_{(n-k)\times b}(\mathbb{K}),$$

tiene rango b*.*

Para mejorar el Algoritmo SD, notemos que una solución $X=(X_1,\ldots,X_n)$ de (3.11) es equivalente a encontrar una para el sistema

$$X_1C^R_{g_1}+\cdots+X_nC^R_{g_n}=V^R. \tag{3.12}$$

Los sistemas lineales (3.11) y (3.12) tienen el mismo número de indeterminadas, pero el número de ecuaciones en (3.12) es menor que el número de ecuaciones en (3.11). Además, si los pivotes de $\widetilde{\mathcal{C}}^R(g_1,\ldots,g_n)$ están en las posiciones $(1,l_1),\ldots,(n-k,l_{n-k})$ y $L=\{l_1,\ldots,l_{n-k}\}\subseteq\{1,\ldots,n\}$, entonces hay dos casos a considerar:

- Si V^R tiene $w\leq t$ componentes $\beta_{j_1},\ldots,\beta_{j_w}\in\mathbb{K}$ no nulas en las posiciones $l_{j_1},\ldots,l_{j_w}\in L$, entonces $X_{l_j}=\beta_j$, para cada $j\in\{j_1,\ldots,j_w\}$ y $X_i=0$ para todo $i\in\{1,\ldots,n\}\setminus\{l_{j_1},\ldots,l_{j_w}\}$ y, por consiguiente, el error es $\mathbf{e}=\beta_{j_1}g_{l_{j_1}}+\cdots+\beta_{j_w}g_{l_{j_w}}$.

- De lo contrario, para encontrar la solución X con peso $w \leq t$, usamos el hecho que X es solución (3.11) si y solo si el sistema lineal (3.7) tiene una solución (única) $X_{i_j} = \alpha_{i_j}$, para todo $j \in \{1, \ldots, t\}$ donde hay exactamente w elementos no nulos y el $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ no está contenido en $G_{\mathcal{L}} = \{g_{l_1}, \ldots, g_{l_{n-k}}\}$. Esto es equivalente a que el sistema lineal

$$X_{i_1} C^R_{g_{i_1}} + \cdots + X_{i_t} C^R_{g_{i_t}} = V^R \tag{3.13}$$

tiene una solución única $X_{i_j} = \alpha_{i_j}$, para todo $j \in \{1, \ldots, t\}$. Lo anterior sucede si y solo si la matriz $\mathcal{C}^R(g_{i_1}, \ldots, g_{i_t})$ y la matriz extendida

$$\mathcal{M}^R(g_{i_1}, \ldots, g_{i_t}) := \left(\begin{array}{ccc|c} C^R_{g_{i_1}} & \cdots & C^R_{g_{i_t}} & V^R \end{array} \right),$$

tienen rango igual a t. Por el Corolario 3.4.5, solo necesitamos comprobar que el rango de $\mathcal{M}^R(g_{i_1}, \ldots, g_{i_t})$ es igual a t. Por lo tanto, el algoritmo mejorado busca un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\}$ de G no contenido en $G_{\mathcal{L}}$ (denotamos a $\mathcal{J}$ como el conjunto de todos aquellos $t-$subconjuntos) tal que la correspondiente matriz $\mathcal{M}^R(g_{i_1}, \ldots, g_{i_t})$ tenga un rango igual a t.

Por eficiencia, antes de descodificar, calculamos las matrices $\mathcal{C}^R(g_1, \ldots, g_n)$ y $\mathcal{U}$ de la Proposición 3.4.3. Esto se hace, calculando para la matriz

$$\left(\begin{array}{ccc|c} C^+_{g_1} & \cdots & C^+_{g_n} & \mathcal{I}_{n \times n} \end{array} \right),$$

su matriz escalonada reducida:

$$\left(\begin{array}{ccc|c} \widetilde{C}^R_{g_1} & \cdots & \widetilde{C}^R_{g_n} & \mathcal{U} \end{array} \right).$$

Una vez se ha recibido la palabra $\mathbf{r} \in \mathbb{K}G$, solo necesitamos calcular V^R. Para ello, calculamos el vector columna V^+ de los coeficientes de su $\mathbb{K}G-$síndrome $S^+(\mathbf{r})$ y el producto $\widetilde{V}^R = \mathcal{U} \cdot V^+$.

3.4.1. Algoritmo mejorado de descodificación por $\mathbb{K}G-$síndrome (Algoritmo ISD)

Paso 1.

Una vez se ha recibido la palabra $\mathbf{r} \in \mathbb{K}G$, se calcula su $\mathbb{K}G-$síndrome. Si $S^+(\mathbf{r}) = 0$, entonces no hay errores y el algoritmo finaliza. De lo contrario, se calcula V^R y se continúa con el Paso 2.

Paso 2.

Si V^R tiene $w \leq t$ componentes $\beta_{j_1}, \ldots, \beta_{j_w} \in \mathbb{K}$ distintas de cero en las posiciones $l_{j_1}, \ldots, l_{j_w} \in L$, entonces $\mathbf{e} = \beta_{j_1} g_{l_{j_1}} + \cdots + \beta_{j_w} g_{l_{j_w}}$ es el error y el algoritmo finaliza. De lo contrario, se considera la matriz $\mathcal{M}^R(g_1, \ldots, g_n)$ y se procede al Paso 3.

Paso 3.

Se selecciona aleatoriamente un $t-$subconjunto $\{g_{i_1}, \ldots, g_{i_t}\} \in \mathcal{J}$. Se considera $\mathcal{M}^R(g_{i_1}, \ldots, g_{i_t})$ y se calcula su rango.

a. Si el rango es igual a t, se halla la solución del sistema lineal (3.13). Si $\alpha_{i_1}, \ldots, \alpha_{i_w}$ son los elementos no nulos de tal solución, entonces $\mathbf{e} = \alpha_{i_1} g_{i_{j_1}} + \cdots + \alpha_{i_w} g_{i_{j_w}}$ y el algoritmo finaliza.

b. De lo contrario, se descarta $\{g_{i_1}, \ldots, g_{i_t}\}$ y se repite el Paso 3 con otro elemento de $\mathcal{J}$.

El algoritmo finaliza cuando encuentra $\{g_{i_1}, \ldots, g_{i_t}\} \in \mathcal{J}$ tal que

$$\text{Rank}(\mathcal{M}^R(g_{i_1}, \ldots, g_{i_t})) = t \tag{3.14}$$

o cuando todos los elementos de $\mathcal{J}$ han sido considerados y ninguno satisface la propiedad (3.14). En tal caso, el error no puede ser corregido.

El Algoritmo ISD mejora el Algoritmo SD, puesto que, si todas las posiciones de error pertenecen a G_L, no es necesario hacer la búsqueda en todos los $t-$subconjuntos de G. En caso contrario, igual sigue siendo más eficiente, puesto que, calcula el rango de matrices con $n - k$ filas en lugar de matrices con n filas y porque el conjunto $\mathcal{J}$ es más pequeño que el conjunto de todos los $t-$subconjuntos de G.

Ejemplo 3.4.6. *En el ejemplo 3.3.2, tenemos que (ver Ejemplo 3.4.4)*

$$L = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 17, 19\}.$$

Descodificando la misma palabra

$$\mathbf{r} = (3, 2, 0, 0, 1, 2, 0, 4, 0, 4, 1, 1, 1, 4, 2, 1, 4, 0, 2, 3, 0, 0, 3, 2),$$

en el Paso 2 del Algoritmo ISD, calculamos que

$$V^R = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 3, 0)^T$$

tiene solo 2 coordenadas no nulas. Estas son $l_{12} = 13$ *y* $l_{14} = 17$ *con magnitudes* $\beta_{l_{12}} = 2$ *y* $\beta_{l_{14}} = 3$. *Así,* $\mathbf{e} = 2g_{13} + 3g_{17}$ *(como lo habíamos calculado antes).*

Ahora bien, si

$$\mathbf{r} = (3, 2, 0, 0, 1, 1, 0, 4, 0, 4, 1, 2, 4, 4, 2, 1, 1, 0, 2, 3, 0, 0, 3, 2),$$

entonces, descodificando con el mismo algoritmo, tenemos que $S^+(\mathbf{r}) \neq 0$. *Por consiguiente, calculamos*

$$V^R = (1, 1, 1, 1, 1, 0, 4, 4, 4, 4, 4, 0, 0, 0, 0)^T.$$

Como el número de componentes no nulas en V^R *es mayor que* $t = 3$, *entonces al menos una de las posiciones de error no pertenece a* $G_{\mathcal{L}}$. *Por lo tanto, vamos al paso 3 y obtenemos que las posiciones de error son* g_6 *y* g_{12}.

En las Tablas II y III podemos apreciar como el Algoritmo ISD es más eficiente que el Algoritmo SD. Si bien el Algoritmo GMD es más eficiente que el Algoritmo ISD, requiere más precálculos.

Aun así, el algoritmo GMD se muestra más eficiente en el caso de un cuerpo binario (ver Tabla II).

Tabla II: Número aproximado de operaciones para el Ejemplo 3.2.2.

Algoritmo	Precálculos	Descodificación
SSD	$2,6 \times 10^4$	$2,2 \times 10^{10}$
SSD (Caso Abeliano)	$7,9 \times 10^4$	$1,2 \times 10^9$
GMD	$5,2 \times 10^7$	$3,7 \times 10^4$
SD	$2,4 \times 10^3$	$2,0 \times 10^9$
ISD	$1,8 \times 10^5$	$1,1 \times 10^9$

Tabla III: Número aproximado de operaciones para el Ejemplo 3.3.2.

Algoritmo	Precálculos	Descodificación
SSD	$2,9 \times 10^3$	$2,4 \times 10^6$
GMD	$4,0 \times 10^5$	$2,3 \times 10^3$
SD	$5,8 \times 10^2$	$4,9 \times 10^5$
ISD	$2,1 \times 10^4$	$2,3 \times 10^5$

3.4.2. Análisis de la complejidad

En el Algoritmo ISD, además de calcular los vectores columna C_{g_i}, se calculan las matrices $\mathcal{C}^R(g_1, \ldots, g_n)$ y $\mathcal{U}$. Este proceso (por eliminación de Gauss-Jordan) tiene orden de complejidad $\mathcal{O}(n^3)$.

El primer paso coincide con el paso 1 del Algoritmo SD. Si $S^+(\mathbf{r}) \neq 0$, el algoritmo calcula el vector $V^R = \mathcal{A} \cdot V^+$. Esto requiere un total de n^2 operaciones.

El paso 2 es inmediato si V^R tiene $w \leq t$ componentes no nulas.

De lo contrario, similar al paso 2 del Algoritmo SD, el Algoritmo ISD realiza la búsqueda en todos los elementos de $\mathcal{J}$ calculando rangos de matrices con $n-k$ filas. Dado que $\mathcal{J}$ tiene tamaño

$$\binom{n}{t} - \binom{n-k}{t}$$

y

$$\mathcal{O}\left(\binom{n}{t} - \binom{n-k}{t}\right) = \mathcal{O}\left(\binom{n}{t-1}\right),$$

el orden de complejidad del Paso 3 es

$$\mathcal{O}\left(nt^2 \times \binom{n}{t-1}\right). \tag{3.15}$$

Además, en el paso 3, si el error $\mathbf{e} \in \mathbb{K}G$ tiene peso $w < t$, la probabilidad de tomar aleatoriamente un elemento de $\mathcal{J}$ que contenga a $\mathrm{Supp}(\mathbf{e})$ es

$$\binom{n-w}{t-w} \div \left[\binom{n}{t} - \binom{n-k}{t}\right].$$

Así, cuanto menor sea w, la probabilidad de encontrar dicho $t-$subconjunto es mayor que la probabilidad considerada para el Algoritmo SD.

Capítulo 4

Descodificación por permutación

4.1. Consideraciones previas

A continuación presentamos algunas definiciones y resultados básicos de la técnica de descodificación por permutación en códigos lineales (ver [29] y [40]). Para ello, vamos a considerar un cuerpo finito $\mathbb{K}$ y $\mathfrak{C}$ un código lineal sobre $\mathbb{K}$ con longitud n, dimensión k y capacidad correctora t. Recordemos que $t < n - k$, en virtud de la cota de Singleton. También vamos a considerar " $\cdot$ " como el producto usual de matrices.

Recordemos algunas definiciones del Capítulo 1. Se dice que $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ es *un conjunto de información* de $\mathfrak{C}$ si al proyectar $\mathfrak{C}$ en las posiciones $i_1, \ldots, i_k$, se obtiene un espacio vectorial de dimensión k. En este caso, los elementos de I se denominan *posiciones de información* y su complemento $I' = \{1, \ldots, n\} \setminus I$ se denomina *conjunto de posiciones de control* de $\mathfrak{C}$. Dado un conjunto de información, los símbolos de una palabra que están ubicados en las posiciones de información se denominan *símbolos de información*. Además, explicamos que todo conjunto de información de $\mathfrak{C}$ es un conjunto de posiciones de control de su código dual y viceversa. El grupo simétrico S_n actúa sobre $\mathbb{K}^n$ mediante

$$\sigma(x_1, \ldots, x_n) = (x_{\sigma(1)}, \ldots, x_{\sigma(n)}).$$

El código $\mathfrak{C}^* \subseteq \mathbb{K}^n$ es *equivalente (por permutación)* a $\mathfrak{C}$ si existe $\sigma \in S_n$ tal que $\mathfrak{C}^* = \sigma(\mathfrak{C})$. Con esto, el conjunto definido por

$$\mathrm{PAut}(\mathfrak{C}) := \{\sigma \in S_n : \sigma(\mathfrak{C}) = \mathfrak{C}\},$$

es un subgrupo de S_n llamado *el grupo de automorfismos permutación de* $\mathfrak{C}$.

En lo que resta de esta sección, fijamos a I como un conjunto de información de $\mathfrak{C}$ e I' denotará su correspondiente conjunto de posiciones de control.

Definición 4.1.1 ([34]). *Dado un número entero positivo $s \leq t$, se dice que $\mathcal{P} \subseteq PAut(\mathfrak{C})$ es un $s-$PD-conjunto parcial de $\mathfrak{C}$ con respecto a I si para todo $S \subseteq \{1, \ldots, n\}$ con cardinal $|S| = s$, existe $\sigma \in \mathcal{P}$ tal que $\sigma(S) \cap I = \emptyset$. En particular, si $s = t$, se dice que $\mathcal{P}$ es un PD-conjunto de $\mathfrak{C}$ con respecto a I.*

En [35] encontramos condiciones para determinar $s-$PD-conjuntos parciales de códigos lineales.

Proposición 4.1.2 ([35]). *Sean $\mathcal{P}$ un subgrupo de $PAut(\mathfrak{C})$ y*

$$m = \max_{\mathcal{O}} \left\{ \frac{|\mathcal{O} \cap I|}{|\mathcal{O}|} \right\},$$

donde $\mathcal{O}$ recorre todas las órbitas de $\{1, \ldots, n\}$ bajo la acción de $\mathcal{P}$. If $s = \min\{\lceil 1/m \rceil - 1, t\}$, entonces $\mathcal{P}$ es un $s-$PD-conjunto parcial de $\mathfrak{C}$ con respecto a I.

Dado un conjunto de información, la idea del algoritmo de descodificación por permutación consiste en aplicar los elementos de un PD-conjunto parcial a las posiciones de error hasta que queden por fuera del conjunto de información ([40]). Tal idea se justifica con el siguiente resultado.

Teorema 4.1.3 ([40]). *Sea $\mathcal{H}$ una matriz de control de $\mathfrak{C}$ cuyas columnas ubicadas en las posiciones de I' forman la matriz identidad $\mathcal{I}_{n-k}$. Si $\mathbf{c} \in \mathfrak{C}$ es la palabra enviada y $\mathbf{r} = \mathbf{c} + \mathbf{e}$ es la palabra recibida, donde $wt(\mathbf{e}) \leq t$, entonces los símbolos de información de $\mathbf{r}$ son correctos si y solo si $wt(\mathcal{H} \cdot \mathbf{r}^T) \leq t$.*

Es decir, si $\mathrm{wt}(\mathbf{e}) \leq t$, entonces los errores de $\mathbf{r} \in \mathbb{K}^n$ quedan por fuera de I si y solo si $\mathrm{wt}(\mathcal{H} \cdot \mathbf{r}^T) \leq t$.

Algoritmo de descodificación parcial por permutación (Algoritmo PPD)

Con la notación del Teorema 4.1.3, si $s \leq t$, $I' = \{l_1, \ldots, l_{n-k}\}$ y $\mathcal{P}$ es un $s-$PD-conjunto parcial de $\mathfrak{C}$ con respecto a I, *el algoritmo de descodificación parcial por permutación* dado en [40] permite corregir errores de peso menor o igual a s y se describe de la siguiente manera:

Recibida $\mathbf{r} \in \mathbb{K}^n$, se toma $\sigma \in \mathcal{P}$ y se calcula $\mathrm{Syn}(\sigma(\mathbf{r})) := \mathcal{H} \cdot \sigma(\mathbf{r})^T$. Suponga que

$$\mathrm{Syn}(\sigma(\mathbf{r})) = (e''_1, \ldots, e''_{n-k}).$$

- Si $\mathrm{wt}(\mathrm{Syn}(\sigma(\mathbf{r}))) \leq s$, se define $\mathbf{e}' := (e'_i) \in \mathbb{K}^n$ mediante

$$e'_i = \begin{cases} e''_j, \text{ si } i = l_j \text{ para algún } j \in \{1, \ldots, n-k\}, \\ 0, \text{ en caso contrario.} \end{cases}$$

  Entonces, el error es $\mathbf{e} = \sigma^{-1}(\mathbf{e}')$ y el algoritmo termina.

- De lo contrario, se descarta el elemento σ y se repite de nuevo el proceso con otro elemento de $\mathcal{P}$.

El algoritmo termina cuando se encuentra un elemento $\sigma \in \mathcal{P}$ tal que

$$\mathrm{wt}\left(\mathrm{Syn}(\sigma(\mathbf{r}))\right) \leq s, \tag{4.1}$$

o cuando todos los elementos de $\mathcal{P}$ han sido verificados y ninguno satisface (4.1). En este último caso, el algoritmo no puede descodificar la palabra recibida, puesto que, la cantidad de errores ocurridos supera la capacidad correctora de $\mathfrak{C}$.

Si $s = t$, se denomina *algoritmo de descodificación por permutación* (Algoritmo PD). El orden de complejidad del Algoritmo PPD es

$$\mathcal{O}(\rho \times n^3), \tag{4.2}$$

donde $\rho = |\mathcal{P}|$. En efecto, si cada permutación $\sigma \in \mathcal{P}$ implica n operaciones para aplicarla al vector $\mathbf{r}$, y luego se realiza la multiplicación de la matriz $\mathcal{H} \in M(\mathbb{K})_{(n-k)\times n}$ por cada $\sigma(\mathbf{r})$, entonces la mayor cantidad de operaciones que debemos hacer es $\rho \cdot n \cdot n(n-k)$.

Lo anterior indica que el Algoritmo PPD será más eficiente cuanto menor sea el tamaño del PD-conjunto parcial. También cabe notar que este algoritmo se puede usar si existe un método eficiente para encontrar un conjunto de información I de $\mathfrak{C}$, una matriz de control de paridad que cumpla la hipótesis del Teorema 4.1.3 y un $s-$PD-conjunto parcial de $\mathfrak{C}$ con respecto a I. Uno de los objetivos de este capítulo es usar propiedades de grupos y álgebras de grupo semisimples para determinar si es posible establece tal método en códigos grupo.

4.2. Implementación en códigos grupo

De nuevo, $G = \{g_1 = 1_G, \ldots, g_n\}$ denotará un grupo finito y $\mathbb{K}G$ el álgebra de grupo de G sobre $\mathbb{K}$. A partir de esta sección, asumiremos que $\mathfrak{C}$ es un

$G-$código sobre $\mathbb{K}$, es decir, $\mathfrak{C}$ puede ser identificado con un ideal bilátero de $\mathbb{K}G$. Cada elemento $\sigma \in S_n$ define una aplicación lineal sobre $\mathbb{K}G$, denotada también σ, y dada por

$$\sigma\left(\sum_{i=1}^{n} \alpha_i g_i\right) = \sum_{i=1}^{n} \alpha_i g_{\sigma(i)}.$$

Como la base de $\mathbb{K}G$ es fija, $\mathrm{PAut}(\mathfrak{C})$ se define de manera análoga al caso lineal, y si $I \subseteq \{1, \ldots, n\}$ es un conjunto de información de $\mathfrak{C}$, entonces se puede trabajar indistintamente con $G_I = \{g_i : i \in I\}$ como conjunto de información de $\mathfrak{C}$. Del mismo modo, el conjunto $G_{I'} = \{g_i : i \in I'\} = G \setminus G_I$ también se puede considerar conjunto de posiciones de control de $\mathfrak{C}$.

En el resto del capítulo, asumiremos que la característica de $\mathbb{K}$ no divide al orden de G y, por tanto, como en el Capítulo 3, $\mathfrak{C} = \langle e_0 \rangle$ está generado por un idempotente central $e_0 \in \mathbb{K}G$. Lo anterior también implica que existe un ideal bilátero $\mathfrak{C}^+$, generado por un idempotente central $e_0^+ \in \mathbb{K}G$, tal que $\mathbb{K}G = \mathfrak{C} \oplus \mathfrak{C}^+$. En consecuencia, $\mathbf{z} \in \mathfrak{C}$ si y solo si existe $\mathbf{z}' \in \mathbb{K}G$ tal que $\mathbf{z} = \mathbf{z}'e_0$. Esto es equivalente a que $\mathbf{z}e_0^+ = 0$. Recordemos que el $\mathbb{K}G-$síndrome de $\mathbf{z} \in \mathbb{K}G$ es el elemento de $\mathbb{K}G$ dado por $S^+(\mathbf{z}) = \mathbf{z}e_0^+$ y que, para cada $g_i \in G$, $C_{g_i}^+$ es el vector columna de los coeficientes de $g_i e_0^+$ con respecto al orden fijado en la base B.

En el Capítulo 3 vimos que si $\mathbf{r} \in \mathbb{K}G$ es la palabra recibida de enviar una palabra código, la *ecuación clave* $\mathbf{x}e_0 = S^+(\mathbf{r})$ se puede escribir de manera equivalente como el sistema de ecuaciones lineales

$$\mathcal{C}^+(g_1, \ldots, g_n) \cdot X^T = V^+,$$

siendo

$$\mathcal{C}^+(g_1, \ldots, g_n) = \begin{pmatrix} C_{g_1}^+ & \ldots & C_{g_n}^+ \end{pmatrix} \in M_{n\times n}(\mathbb{K}),$$

y V^+ el vector columna de coeficientes de $S^+(\mathbf{r})$. De donde obtenemos que

$$\mathfrak{C} = \{\mathbf{z} \in \mathbb{K}G : \mathcal{C}^+(g_1, \ldots, g_n) \cdot Z^T = 0\}. \tag{4.3}$$

La expresión (4.3) y la Proposición 3.4.1, nos hacen pensar que $\mathcal{C}^+(g_1, \ldots, g_n)$ se comporta como una matriz de control del código grupo $\mathfrak{C}$. En efecto, tenemos el siguiente resultado.

Proposición 4.2.1. *Con las notaciones anteriores, las filas de $\mathcal{C}^+(g_1, \ldots, g_n)$ son vectores del código dual $\mathfrak{C}^\perp$ de $\mathfrak{C}$.*

Demostración. Supongamos que

$$e_0^+ = \sum_{i=1}^{n} \varepsilon_0^+(g_i) g_i,$$

donde $\varepsilon_0^+(g_i) \in \mathbb{K}$. Entonces,

$$g_j e_0^+ = \sum_{i=1}^{n} \varepsilon_0^+(g_i) g_j g_i = \sum_{h=1}^{n} \varepsilon_0^+(g_j^{-1} g_h) g_h$$

y así

$$\mathcal{C}^+(g_1, \ldots, g_n) = \begin{pmatrix} \varepsilon_0^+(g_1^{-1}g_1) & \varepsilon_0^+(g_2^{-1}g_1) & \ldots & \varepsilon_0^+(g_n^{-1}g_1) \\ \varepsilon_0^+(g_1^{-1}g_2) & \varepsilon_0^+(g_2^{-1}g_2) & \ldots & \varepsilon_0^+(g_n^{-1}g_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_0^+(g_1^{-1}g_n) & \varepsilon_0^+(g_2^{-1}g_n) & \ldots & \varepsilon_0^+(g_n^{-1}g_n) \end{pmatrix}.$$

Por la proposición 1.2.11, tenemos que $\varphi(\mathfrak{C}^+) = \mathfrak{C}^\perp$, donde $\varphi : \mathbb{K}G \to \mathbb{K}G$ es el anti-automorfismo de anillos dado por

$$\varphi\left(\sum_{i=1}^{n} \alpha_i g_i\right) = \sum_{i=1}^{n} \alpha_i g_i^{-1}.$$

Como el código dual de un código grupo $\mathfrak{C}$, también es un código grupo (ver Proposición 1.1.10), entonces por la Proposición 1.2.5 tenemos que $\mathfrak{C}^\perp$ es un ideal bilátero de $\mathbb{K}G$. Como $\mathbb{K}G$ es semisimple, entonces $\mathfrak{C}^\perp$ está generado por un idempotente central que será denotado por $e_0^\perp$ y

$$e_0^\perp = \varphi(e_0^+) = \sum_{i=1}^{n} \varepsilon_0^+(g_i) g_i^{-1}.$$

Pero

$$g_j e_0^\perp = \sum_{i=1}^{n} \varepsilon_0^+(g_i) g_j g_i^{-1} = \sum_{h=1}^{n} \varepsilon_0^+(g_h^{-1} g_j) g_h,$$

y por lo tanto, las filas de $\mathcal{C}^+(g_1, \ldots, g_n)$ son vectores de $\mathfrak{C}^\perp$. □

Dado que los coeficientes de las filas de la matriz $\mathcal{C}^+(g_1, \ldots, g_n)$ están ordenados de la misma forma que la base B de $\mathbb{K}G$, tenemos el siguiente resultado para el cual debemos recordar la Definición 3.4.2 y la notación posterior a ella.

Corolario 4.2.2. *Sea $\widetilde{\mathcal{C}}^R(g_1,\ldots,g_n)$ la matriz escalonada reducida (por filas) de $\mathcal{C}^+(g_1,\ldots,g_n)$. Supongamos que los pivotes de $\widetilde{\mathcal{C}}^R(g_1,\ldots,g_n)$ están en las posiciones $(1,l_1),\ldots,(n-k,l_{n-k})$. Si $L=\{l_1,\ldots,l_{n-k}\}$ y $G_L=\{g_l\in G : l\in L\}$, entonces $G\setminus G_L$ es un conjunto de información de $\mathfrak{C}$ y $\mathcal{C}^R(g_1,\ldots,g_n)$ es una matriz de control de $\mathfrak{C}$ que tiene la matriz identidad $\mathcal{I}_{n-k}$ en las posiciones indicadas por los elementos de G_L.*

Notemos que calcular tal matriz tiene orden de complejidad $\mathcal{O}(n^3)$. Con lo cual, el corolario anterior nos da un método eficiente de obtener un conjunto de información y una matriz de control para el código grupo $\mathfrak{C}$ que satisfacen las hipótesis del Teorema 4.1.3. Solo resta encontrar PD-conjuntos parciales respecto al conjunto de posiciones de control hallado en el Corolario 4.2.2. Así podremos implementar el Algoritmo PPD en códigos grupo.

4.3. PD-conjuntos en códigos grupo

Conservando la notación de la sección anterior, los siguientes resultados nos permiten construir subgrupos de $\mathrm{PAut}(\mathfrak{C})$ que pueden ser usados como $s-$PD-conjuntos parciales de $\mathfrak{C}$ con respecto a cualquier conjunto de información del código grupo $\mathfrak{C}$.

Proposición 4.3.1 ([7])**.** *Sea I un conjunto de información de $\mathfrak{C}$. Si $\Theta=\{\theta_i\}_{i=1}^n$, donde $\theta_i(z):=g_iz$ para todo $z\in\mathbb{K}G$ y $g_i\in G$, entonces Θ es un subgrupo de $PAut(\mathfrak{C})$ y es un $s-$PD-conjunto parcial de $\mathfrak{C}$ con respecto a I, donde $s=\min\{\lceil n/k\rceil-1,t\}$.*

Demostración. El conjunto Θ tiene estructura de grupo y es isomorfo a G. Como $\mathfrak{C}$ es un ideal bilátero de $\mathbb{K}G$, entonces $\theta_i(\mathfrak{C})=g_i\mathfrak{C}=\mathfrak{C}$ y así, $\theta_i\in\mathrm{PAut}(\mathfrak{C})$. Ahora bien, la acción de Θ sobre G tiene una única órbita, que es G. Si aplicamos la Proposición 4.1.2 con $m=k/n$, podemos completar la prueba. □

El resultado anterior nos dice que si $tk<n$, es posible implementar el Algoritmo PD en códigos grupo tomando a Θ como PD-conjunto.

Ejemplo 4.3.2. *Sean $\mathbb{K}=\mathbb{F}_3$ y $G=D_{13}=\langle a,b : a^2=b^{13}=1, aba^{-1}=b^{-1}\rangle$. Si fijamos el orden en la base de $\mathbb{F}_3D_{13}$ como*

$$\{1_G,a,b,ab,b^2,ab^2,b^3,ab^3,b^4,ab^4,b^5,ab^5,b^6,ab^6,\\ b^7,ab^7,b^8,ab^8,b^9,ab^9,b^{10},ab^{10},b^{11},ab^{11},b^{12},ab^{12}\}$$

y suponemos que $\mathfrak{C}=\langle e_0\rangle$, donde

$$e_0=(0,0,0,0,2,0,0,0,0,0,0,2,0,2,0,2,0,2,0,0,0,0,0,0,2,0,0,0),$$

entonces $n = 26$, $k = 12$ y $t = 2$. Por el Corolario 4.3.1, Θ es un PD-conjunto de $\mathfrak{C}$ con respecto a cualquier conjunto de información de $\mathfrak{C}$.

Observemos que en el ejemplo anterior, $\lceil n/k \rceil - 1 = 2 = t$. Sin embargo, esto no ocurre siempre (ver Ejemplo 2.1.4). Por lo tanto, nuestro objetivo es buscar subconjuntos de $\mathrm{PAut}(\mathfrak{C})$ que se puedan utilizar como posibles PD-conjuntos de $\mathfrak{C}$ con respecto al conjunto de información encontrado en el Corolario 4.2.2. De hecho, veremos que tales resultados pueden ser aplicados considerando cualquier conjunto de información de $\mathfrak{C}$.

Probablemente el siguiente resultado es conocido. Incluimos una demostración para completitud del trabajo y compresión del lector.

Proposición 4.3.3. *Si $\psi \in Aut(G)$, entonces ψ se extiende a un automorfismo del álgebra $\mathbb{K}G$.*

Demostración. Si $\psi \in \mathrm{Aut}(G)$, entonces ψ se extiende a un automorfismo lineal de $\mathbb{K}G$, que también es un automorfismo de anillos, definiendo

$$\psi\left(\sum_{i=1}^{n} \alpha_i g_i\right) := \sum_{i=1}^{n} \alpha_i \psi(g_i).$$

En efecto,

$$\psi\left(\left(\sum_{i=1}^{n} \alpha_i g_i\right)\left(\sum_{j=1}^{n} \beta_j g_j\right)\right) = \psi\left(\sum_{l=1}^{n}\left(\sum_{g_l = g_i g_j} \alpha_i \beta_j\right) g_l\right) =$$

$$\sum_{l=1}^{n}\left(\sum_{g_l = g_i g_j} \alpha_i \beta_j\right)\psi(g_l) = \sum_{l=1}^{n}\left(\sum_{\psi(g_l) = \psi(g_i)\psi(g_j)} \alpha_i \beta_j\right)\psi(g_l) =$$

$$\left(\sum_{i=1}^{n} \alpha_i \psi(g_i)\right)\left(\sum_{j=1}^{n} \beta_j \psi(g_j)\right) = \psi\left(\sum_{i=1}^{n} \alpha_i g_i\right)\psi\left(\sum_{j=1}^{n} \beta_j g_j\right).$$

□

Los automorfismos lineales de $\mathbb{K}G$ que nos interesan considerar son aquellos cuya restricción a G definen una biyección de G a G. Con lo cual, existe un correspondecia biyectiva entre las permutaciones del grupo G y el conjunto de tales automorfismos.

Teorema 4.3.4. *Si $\lambda \in Aut(G)$ y denotamos también λ al correspondiente automorfismo inducido en el álgebra $\mathbb{K}G$, entonces $\lambda \in PAut(\mathfrak{C})$ si y solo si $\lambda(e_0) = e_0$.*

Demostración. Como λ es un automorfismo de anillos, la imagen de un idempotente es un idempotente. Si $\lambda \in \mathrm{PAut}(\mathfrak{C})$, entonces $\lambda(\mathfrak{C}) = \mathfrak{C}$. Dado que e_0 (respectivamente $\lambda(e_0)$) es la identidad de $\mathfrak{C} = \langle e_0 \rangle$ (respectivamente $\lambda(\mathfrak{C}) = \langle \lambda(e_0) \rangle$), entonces $\lambda(e_0) = e_0$. Recíprocamente, si $\lambda(e_0) = e_0$ y $\mathbf{c} \in \mathfrak{C}$, entonces $\mathbf{c} = \mathbf{z}e_0$ para algún $\mathbf{z} \in \mathbb{F}G$ y así $\lambda(\mathbf{c}) = \lambda(\mathbf{z}e_0) = \lambda(\mathbf{z})\lambda(e_0) = \lambda(\mathbf{z})e_0$. Con lo anterior, $\lambda(\mathbf{c}) \in \mathfrak{C}$ y, en consecuencia, $\lambda \in \mathrm{PAut}(\mathfrak{C})$. □

Denotaremos por $\mathrm{RPAut}(\mathfrak{C})$ al subgrupo de $\mathrm{PAut}(\mathfrak{C})$ de aquellos elementos que son inducidos por automorfismos del grupo G.

Corolario 4.3.5. *El conjunto $RPAut(\mathfrak{C}) = \{\lambda \in Aut(G) : \lambda(e_0) = e_0\}$ es un grupo contenido en $PAut(\mathfrak{C})$.*

Demostración. En efecto, $\mathrm{RPAut}(\mathfrak{C})$ es el estabilizador de e_0 bajo la acción de $\mathrm{Aut}(G)$. □

Notemos que el automorfismo identidad $id : \mathbb{K}G \to \mathbb{K}G$ es el único elemento de Θ que pertenece a las extensiones lineales de los elementos de $\mathrm{Aut}(G)$. Por tanto, si Ψ es un subgrupo de extensiones lineales de los elementos de $\mathrm{Aut}(G)$, entonces $\Theta \cap \Psi = \{id\}$ y, por consiguiente, los elementos de $\Theta\Psi$ se pueden escribir de manera única. Este hecho se usará para formular la Definición 4.5.1, que veremos en la Sección 4.5. Adicionalmente, $\Psi\Theta = \Theta\Psi$. En efecto, para cada $\theta_i \in \Theta$ y $\psi \in \Psi$, tenemos que $(\psi\theta_i)(g_l) = \psi(g_i g_l) = \psi(g_i)\psi(g_l) = (\theta_j\psi)(g_l)$, donde $j \in \{1, \ldots, n\}$ es el índice tal que $g_j = \psi(g_i)$. Como Θ y $\mathrm{RPAut}(\mathfrak{C})$ son grupos contenidos en $\mathrm{RPAut}(\mathfrak{C})$, tenemos el siguiente resultado.

Corolario 4.3.6. *Si Λ es un subgrupo de $RPAut(\mathfrak{C})$, entonces $\Phi = \langle \Theta, \Lambda \rangle = \Theta\Lambda$ es un subgrupo de $PAut(\mathfrak{C})$.*

El siguiente teorema nos dice cuando el grupo Φ, del Corolario anterior, contiene un PD-conjunto de $\mathfrak{C}$ con respecto a cualquier conjunto de información.

Teorema 4.3.7. *Sean I un conjunto de información de $\mathfrak{C}$ y Λ un subgrupo de $RPAut(\mathfrak{C})$. Supongamos que $G_{I'} = \{g_i : i \in I'\}$ y $\sim$ es la relación de equivalencia sobre $\Phi = \langle \Theta, \Lambda \rangle$, dada por*

$$\sigma_1 \sim \sigma_2 \text{ si y solo si } \sigma_1(G_{I'}) = \sigma_2(G_{I'}).$$

Si $\mathcal{P}$ es un sistema completo de representantes del conjunto cociente $\Phi/\sim$, y para cada $\sigma \in \mathcal{P}$, $\mathcal{J}_\sigma$ denota el conjunto de todos los $t-$subconjuntos de $\sigma(G_{I'})$, entonces $\mathcal{P}$ es un PD-conjunto de $\mathfrak{C}$ con respecto a I si y solo si

$$\left|\bigcup_{\sigma\in\mathcal{P}} \mathcal{J}_\sigma\right| = \binom{n}{t}. \tag{4.4}$$

Demostración. Sea $\mathcal{J}$ el conjunto de todos los $t-$subconjuntos de G. Si

$$\mathcal{J} = \bigcup_{\sigma\in\mathcal{P}} \mathcal{J}_\sigma,$$

entonces para todo $t-$subconjunto T de G, existe $\sigma \in \mathcal{P}$ tal que $T \subseteq \sigma(G_{I'})$. En consecuencia, $\sigma^{-1}(T) \cap (G \setminus G_{I'}) = \emptyset$ y así $\mathcal{P}$ es un PD-conjunto de $\mathfrak{C}$ con respecto a I. Recíprocamente, si para todo $t-$subconjunto T de G, existe $\sigma \in \mathcal{P}$ tal que $\sigma(T) \cap (G \setminus G_{I'}) = \emptyset$, entonces $T \subseteq \sigma^{-1}(G_{I'})$ y, por lo tanto, $T \in \mathcal{J}_{\sigma^{-1}}$. De lo anterior,

$$\mathcal{J} \subseteq \bigcup_{\sigma\in\mathcal{P}} \mathcal{J}_\sigma.$$

Como tenemos trivialmente que

$$\bigcup_{\sigma\in\mathcal{P}} \mathcal{J}_\sigma \subseteq \mathcal{J},$$

luego

$$\mathcal{J} = \bigcup_{\sigma\in\mathcal{P}} \mathcal{J}_\sigma,$$

y con esto se completa la prueba. □

De acuerdo al Corolario 4.2.2, $G \setminus G_L$ es un conjunto de información para cualquier $G-$código. Este conjunto de información es el único conocido para cualquier grupo arbitrario G. En particular, el resultado anterior se cumple para $G \setminus G_L$.

Observemos que si Λ un subgrupo de $\mathrm{RPAut}(\mathfrak{C})$ y $v = |\Lambda|$, entonces calcular $\mathcal{P}$ tiene orden de complejidad

$$\mathcal{O}(v^2 \times n^3). \tag{4.5}$$

En efecto, el grupo $\Phi = \langle\Theta, \Lambda\rangle$ tiene nv elementos. Por cada $\sigma \in \Phi$ calculamos $\sigma(G_L)$ el cual tiene orden $n - k$.Por consiguiente, el número de operaciones necesarias para comparar cada par de elementos (distintos) de Φ es

$$(n-k) \times \frac{nv(nv-1)}{2}.$$

Adicionalmente, si $\rho = |\mathcal{P}|$, para verificar la Condición (4.4) del mismo teorema, por cada elemento $\sigma \in \mathcal{P}$, hay que considerar

$$\binom{n-k}{t}$$

$t-$subconjuntos de $\sigma(G_L)$. Entonces verificar si el conjunto $\mathcal{P}$, del teorema anterior, es un PD-conjunto tiene orden de complejidad

$$\mathcal{O}\left(\rho \times \binom{n-k}{t}\right) \tag{4.6}$$

A continuación veremos con un ejemplo que la elección $\Lambda = \mathrm{RPAut}(\mathfrak{C})$ algunas veces resulta muy útil.

Ejemplo 4.3.8. *Consideremos de nuevo el álgebra de grupo $\mathbb{F}_3 D_{13}$ con la base fijada en el Ejemplo 4.3.2. Para este caso, tomemos $\mathfrak{C} = \langle e_0 \rangle$, donde*

$$e_0 = (1,0,0,0,1,0,0,0,0,0,0,1,0,1,0,1,0,1,0,0,0,0,0,0,1,0,0,0).$$

Entonces, $n = 26$, $k = 14$ y $t = 2$. Aquí,

$$G_L = \{1, a, b, ab, b^2, ab^2, b^3, ab^3, b^4, ab^4, b^5, ab^5\},$$

y Θ no es un PD-conjunto de $\mathfrak{C}$ con respecto a $D_{13} \setminus G_L$. Sin embargo, tenemos que $\Lambda = RPAut(\mathfrak{C})$ es el conjunto de las extensiones lineales $\lambda_{(u,v)} : \mathbb{F}_3 D_{13} \to \mathbb{F}_3 D_{13}$ de los automorfismos de grupo

$$\begin{array}{rccc} \lambda_{(u,v)} : & G & \to & G \\ & a & \mapsto & ab^u \\ & b & \mapsto & b^v \end{array}$$

donde $u = 0, 1, \ldots, 12$ y $v = 1, 3, 4, 9, 10, 12$. Con esta elección, cualquier conjunto $\mathcal{P}$, que sea construido como se menciona en el teorema anterior, es un PD-conjunto de $\mathfrak{C}$ con respecto a $D_{13} \setminus G_L$. Como $|\Phi| = 2028$ y todas las clases de equivalencia de tales construcciones tienen 2 elementos, entonces $|\mathcal{P}| = 1014$ y, por lo tanto, la descodificación es más eficiente si usamos $\mathcal{P}$ en lugar de Φ.

Es importante mencionar que para grupos cíclicos, diédricos, cuaternios, alternados o simétricos, el grupo $\Lambda = \mathrm{RPAut}(\mathfrak{C})$ se puede calcular fácilmente dado que su grupo de automorfismo es bien conocido y tiene un tamaño menor que n^2. Sin embargo, verificar si $\lambda(e_0) = e_0$ para cada $\lambda \in \mathrm{Aut}(G)$ tiene orden de complejidad $\mathcal{O}(\mu \times n)$ donde $\mu = |\mathrm{Aut}(G)|$. Esto implica que en ocasiones, el cálculo de $\mathrm{RPAut}(\mathfrak{C})$ puede ser bastante complejo (por ejemplo, si $\mathrm{Aut}(G)$ es

muy grande o desconocido). En otros casos, el tamaño de RPAut($\mathfrak{C}$) podría ser muy grande y, como consecuencia, la descodificación por permutación no es tan eficiente. Por lo tanto, proponemos algunas otras opciones para $\Lambda \leq \mathrm{RPAut}(\mathfrak{C})$ que pueden dar lugar a PD-conjuntos. Para la descodificación, se escogerá Λ lo más pequeño posible de tal modo que el conjunto $\mathcal{P}$, inducido por el Teorema 4.3.7, sea un PD-conjunto.

Proposición 4.3.9. *Sea G un grupo no abeliano. Si $\Lambda = Inn(G)$ es el subgrupo de automorfismos internos de G, entonces $\Lambda \subseteq RPAut(\mathfrak{C})$.*

Demostración. Si $\lambda_i \in \mathrm{Inn}(G)$ es la acción por conjugación dada por $g_i \in G$ y $\mathfrak{C}$ está generado por el idempotente central $e_0 \in \mathbb{K}G$, entonces la extensión lineal de λ_i a $\mathbb{K}G$ satisface que $\lambda_i(e_0) = g_i^{-1} e_0 g_i = g_i^{-1} g_i e_0 = e_0$. □

Si G es un grupo no abeliano, dados $g_i, g_j \in G$, definimos $\sigma_{(i,j)}(\mathbf{z}) := g_i \mathbf{z} g_j$ para todo $\mathbf{z} \in \mathbb{K}G$ y así

$$\Sigma = \{\sigma_{(i,j)} : g_i, g_j \in G\},$$

es un grupo de orden $|\Sigma| = |G|^2/|Z(G)|$ que está contenido en PAut($\mathfrak{C}$). En ese caso, podríamos usar Σ para encontrar un PD-conjunto de $\mathfrak{C}$ con respecto a $I = G \setminus G_L$. Sin embargo, Σ coincide con $\Phi = \langle \Lambda, \Theta \rangle$ tomando $\Lambda = \mathrm{Inn}(G)$. En efecto, si $\theta_i \in \Theta$ y $\lambda_j \in \mathrm{Inn}(G)$, entonces $(\theta_i \lambda_j)(g_l) = g_i g_j^{-1} g_l g_j = \sigma_{(i^*,j)}(g_l)$ siendo $i^* \in \{1, \ldots, n\}$ el índice tal que $g_{i^*} = g_i g_j^{-1}$.

Ahora veremos un par de resultados para el caso abeliano. Notemos que si G es un grupo abeliano y $q \in \mathbb{Z}^+$ es potencia de un número primo p, el cual no divide al orden de G, entonces $g \mapsto g^q$ es un automorfismo de G. Dicho automorfismo se extiende a una aplicación lineal biyectiva $\tau_q : \mathbb{F}_q G \to \mathbb{F}_q G$, dada por $\tau_q(\mathbf{z}) = \mathbf{z}^q$ para todo $\mathbf{z} \in \mathbb{F}_q G$, que resulta ser el automorfismo de Frobenius del anillo $\mathbb{F}_q G$.

Proposición 4.3.10. *Si G es abeliano y $\mathbb{K} = \mathbb{F}_q$, entonces el automorfismo de Frobenius $\tau_q : \mathbb{K}G \to \mathbb{K}G$ pertenece a $RPAut(\mathfrak{C})$ y $\Lambda = \langle \tau_q \rangle$ es un subgrupo de $RPAut(\mathfrak{C})$.*

Demostración. Si $\mathfrak{C} = \langle e_0 \rangle$, entonces $\tau_q(e_0) = e_0^q = e_0$. □

Para los siguientes resultados, notemos que si $G = G_1 \times G_2$, donde G_1 y G_2 son grupos, entonces $\mathbb{K}G = \mathbb{K}G_1 \otimes \mathbb{K}G_2$. Este hecho también es válido para el producto directo de $m > 2$ grupos. Además, si J_1 y J_2 son ideales biláteros de $\mathbb{K}G_1$ y $\mathbb{K}G_2$, respectivamente, entonces $J_1 \otimes J_2$ es un ideal bilátero de $\mathbb{K}G$. Sin embargo, no todo ideal bilátero minimal de $\mathbb{K}G$ es de la forma anterior. Por tal razón, vamos a ilustrar un resultado para códigos ligados a ideales biláteros

que se obtienen como una suma directa interna de productos tensoriales de ideales biláteros minimales de $\mathbb{K}G_1$ y $\mathbb{K}G_2$.

Proposición 4.3.11. *Sean $\mathbb{K} = \mathbb{F}_q$ y $G = G_1 \times G_2$, donde G_1 y G_2 son grupos cíclicos. Si $\mathfrak{C}$ es un $G-$código sobre $\mathbb{K}$ el cual es suma (directa interna) del producto tensorial de ideales biláteros minimales de $\mathbb{K}G_1$ y $\mathbb{K}G_2$, entonces el automorfismo lineal $\lambda_1 : \mathbb{K}G_1 \otimes \mathbb{K}G_2 \to \mathbb{K}G_1 \otimes \mathbb{K}G_2$ dado por*

$$\sum_{\mathbf{z}_1,\mathbf{z}_2} (\mathbf{z}_1 \otimes \mathbf{z}_2) \mapsto \sum_{\mathbf{z}_1,\mathbf{z}_2} (\mathbf{z}_1^q \otimes \mathbf{z}_2),$$

donde $\mathbf{z}_1 \in \mathbb{K}G_1$ e $\mathbf{z}_2 \in \mathbb{K}G_2$, es un automorfismo del anillo $\mathbb{K}G$ y pertenece a RPAut($\mathfrak{C}$). En consecuencia, el grupo $\Lambda = \langle \lambda_1 \rangle$ está contenido en RPAut($\mathfrak{C}$).

Demostración. En efecto, λ_1 es la extensión lineal a $\mathbb{K}G$ del automorfismo de G dado por $(g, h) \mapsto (g^q, h)$ y por la Proposición 4.3.3, es un automorfismo del anillo $\mathbb{K}G$. Ahora bien, si $\mathfrak{C}$ está generado por e_0, la hipótesis implica que

$$e_0 = \sum_{i,j} (e'_i \otimes e''_j),$$

donde $e'_i \in \mathbb{K}G_1$ y $e''_j \in \mathbb{K}G_2$ son idempotentes (centrales) primitivos. Entonces,

$$\lambda_1(e_0) = \lambda_1 \left(\sum_{i,j} (e'_i \otimes e''_j) \right) = \sum_{i,j} \left((e'_i)^q \otimes e''_j \right) = \sum_{i,j} (e'_i \otimes e''_j) = e_0.$$

□

Ejemplo 4.3.12. *Sean $\mathbb{K} = \mathbb{F}_2$ y $G = \mathcal{C}_7 \times \mathcal{C}_7$, donde $G = \langle a, b \rangle$. Tomamos como base de $\mathbb{F}_2(\mathcal{C}_7 \times \mathcal{C}_7)$ a*

$$\{1, a, b, a^2, ab, b^2, a^3, a^2b, ab^2, b^3, a^4, a^3b, a^2b^2, ab^3, b^4, a^5, a^4b, a^3b^2, a^2b^3, ab^4,$$
$$b^5, a^6, a^5b, a^4b^2, a^3b^3, a^2b^4, ab^5, b^6, a^6b, a^5b^2, a^4b^3, a^3b^4, a^2b^5, ab^6, a^6b^2,$$
$$a^5b^3, a^4b^4, a^3b^5, a^2b^6, a^6b^3, a^5b^4, a^4b^5, a^3b^6, a^6b^4, a^5b^5, a^4b^6, a^6b^5, a^5b^6, a^6b^6\}.$$

Por otro lado, calculamos que $e_1 = (1, 1, 1, 1, 1, 1, 1)$, $e_2 = (1, 1, 1, 0, 1, 0, 0)$ y $e_3 = (1, 0, 0, 1, 0, 1, 1)$ son los idempotentes (centrales) primitivos de $\mathbb{F}_2\mathcal{C}_7$ con el orden $\{1, g, \ldots, g^6\}$ para la base $\mathcal{C}_7 = \langle g \rangle$.

Consideremos $\mathfrak{C} = \langle e_0 \rangle$ donde

$$e_0 = (e_1 \otimes e_2) + (e_2 \otimes e_1) + (e_3 \otimes e_1) + (e_3 \otimes e_2).$$

Entonces

$$e_0 = (0,0,1,0,0,1,1,0,0,1,0,0,0,1,1,1,0,0,1,0,1,1,0,0,0,\\ 0,1,1,0,0,1,0,1,1,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0).$$

Los parámetros de $\mathfrak{C}$ *son* $n = 49$, $k = 18$ *y* $t = 5$. *Un conjunto de información de este código grupo es* $(\mathcal{C}_7 \times \mathcal{C}_7) \setminus G_{\mathcal{L}}$, *donde*

$$G_L = \{1, a, b, a^2, ab, b^2, a^3, a^2b, ab^2, b^3, a^4, a^3b, a^2b^2, ab^3, b^4, a^5, a^4b,\\ a^3b^2, a^2b^3, ab^4, b^5, a^6, a^5b, a^4b^2, a^3b^3, a^2b^4, ab^5, b^6, a^6b, a^5b^2, a^2b^5\}.$$

Si tomamos $\Lambda = \langle \tau_q \rangle$, *entonces* $\Phi = \langle \Theta, \Lambda \rangle$ *no satisface la Condición (4.4). Sin embargo, si consideramos* $\Lambda = \langle \lambda_1 \rangle$ *en lugar de* $\langle \tau_q \rangle$, *entonces* $\mathcal{P} = \langle \Theta, \Lambda \rangle$ *cumple tal condición y, por lo tanto, es un PD-conjunto de* $\mathfrak{C}$ *con respecto a* $G \setminus G_L$.

También es posible hacer la construcción analóga para $\Lambda = \langle \lambda_2 \rangle$ donde

$$\lambda_2 : \mathbb{F}G_1 \otimes \mathbb{F}G_2 \to \mathbb{F}G_1 \otimes \mathbb{F}G_2$$
$$\sum_{\mathbf{z}_1, \mathbf{z}_2} (\mathbf{z}_1 \otimes \mathbf{z}_2) \mapsto \sum_{\mathbf{z}_1, \mathbf{z}_2} (\mathbf{z}_1 \otimes \mathbf{z}_2^q),$$

para todo par $\mathbf{z}_1 \in \mathbb{F}G_1$ y $\mathbf{z}_2 \in \mathbb{F}G_2$. Notemos que $\langle \tau_q \rangle \leq \langle \lambda_1, \lambda_2 \rangle$. Todo lo anterior se puede generalizar, si G es un producto directo de $m > 2$ grupos (no necesariamente cíclicos) pero tomando subgrupos específicos de automorfismos de G como se muestra a continuación.

Proposición 4.3.13. *Sea* $G = G_1 \times \cdots \times G_m$ *donde* G_i *es un grupo arbitrario para todo* $i \in \{1, \ldots, m\}$. *Supongamos que* J *es un conjunto finito no vacío y*

$$\mathfrak{C} = \sum_{j \in J} \left[\mathfrak{C}_1^{(j)} \otimes \cdots \otimes \mathfrak{C}_m^{(j)} \right],$$

donde $\mathfrak{C}_i^{(j)}$ *es un ideal bilátero minimal de* $\mathbb{K}G_i$ *para cada par* $i \in \{1, \ldots, m\}$ *y* $j \in J$. *Si* Λ_i *es un subgrupo de automorfismos del algebra* $\mathbb{K}G_i$ *que pertenecen a* $RPAut\left(\mathfrak{C}_i^{(j)}\right)$ *para todo* $j \in J$, *entonces para cada* $\lambda \in \Lambda_i$, *el automorfismo lineal* $\widehat{\lambda} : \mathbb{K}G \to \mathbb{K}G$ *dado por:*

$$\sum_{\mathbf{z}_1, \ldots, \mathbf{z}_m} [\mathbf{z}_1 \otimes \cdots \otimes \mathbf{z}_i \otimes \cdots \otimes \mathbf{z}_m] \mapsto \sum_{\mathbf{z}_1, \ldots, \mathbf{z}_m} [\mathbf{z}_1 \otimes \cdots \otimes \lambda(\mathbf{z}_i) \otimes \cdots \otimes \mathbf{z}_m]$$

donde $\mathbf{z}_h \in \mathbb{K}G_h$ *para todo* $h \in \{1, \ldots, m\}$, *es un automorfismo del anillo* $\mathbb{K}G$ *y pertenece a* $RPAut(\mathfrak{C})$. *Por tanto,* $\widehat{\Lambda}_i = \{\widehat{\lambda} : \mathbb{K}G \to \mathbb{K}G : \lambda \in \Lambda_i\}$ *es un subgrupo de* $RPAut(\mathfrak{C})$.

Con el teorema anterior, podemos hacer muchas elecciones para Λ. Por ejemplo, $\Lambda = \widehat{\Lambda}_i$, $\Lambda = \langle \widehat{\Lambda}_i, \widehat{\Lambda}_{i'} \rangle_{i \neq i'}$ y así sucesivamente con hasta m subgrupos diferentes Λ_i.

4.4. Nuestro trabajo versus otros estudios previos

En primer lugar, debemos señalar que en la literatura no existen estudios de descodificación por permutación en códigos grupo no abelianos. Sin embargo, la descodificación por permutación en códigos cíclicos ([39]) y códigos abelianos (ver [10] y [15]) sí ha sido considerada anteriormente.

En efecto, sean G un grupo cíclico de orden n impar y $\mathfrak{C}$ un $G-$código binario de dimensión k que corrige hasta t errores. MacWilliams demostró el siguiente resultado en [39].

Teorema 4.4.1 ([39]). *Si I es un conjunto de información de $\mathfrak{C}$ con k posiciones consecutivas de $\{1,\ldots,n\}$, entonces Θ es un PD-conjunto de $\mathfrak{C}$ con respecto a I si y sólo si $tk < n$.*

En el mismo trabajo, se usó el *automorfismo de Frobenius* $\tau_2 : \mathbb{F}_2G \to \mathbb{F}_2G$ para caracterizar cuando $\mathcal{P} = \langle \Theta, \tau_2 \rangle$ es un PD-conjunto de un código cíclico binario de longitud impar n. Los resultados de MacWilliams son válidos en códigos cíclicos sobre cualquier cuerpo finito $\mathbb{K} = \mathbb{F}_q$ con la condición que $\mathbb{K}G$ sea semisimple y son equivalentes al método propuesto en este capítulo.

En [3], A. Benyamin-Seeyar et al. incluyeron el término *descodificable por permutación en u pasos*, donde $u \in \{1,\ldots,m-1\}$ y m es el orden de τ_2, para referirse a aquellos códigos cíclicos para los cuales

$$\mathcal{P} = \{\theta^i \tau_2^j : i = 1,\ldots,n; j = 0\ldots,u-1\}$$

es PD-conjunto con respecto a cualquier conjunto de información con k posiciones consecutivas de $\{1,\ldots,n\}$. Algunos resultados que dan caracterizaciones de códigos cíclicos binarios descodificables por permutación en 2 pasos fueron dados por [3] y se resumen en [30]. Aunque todas estas condiciones son muy fáciles de comprobar, sólo resultan útiles para casos muy específicos de códigos cíclicos binarios. Tampoco se conocen resultados similares para códigos cíclicos sobre cuerpos finitos de característica distinta de 2.

En la misma dirección, mencionamos el trabajo de Chabanne ([15]) para códigos bicíclicos semisimples sobre $\mathbb{F}_2$ que, en nuestra notación, son $G-$códigos binarios siendo G el producto directo de dos grupos cíclicos de orden impar. Chabanne en su trabajo consideró el grupo $\mathcal{P} = \langle \Theta, \tau_2 \rangle$ y explicó como puede usarse para descodificar parcialmente en estos $G-$códigos. Tal método también se puede generalizar a códigos grupo abelianos sobre cualquier cuerpo $\mathbb{K} = \mathbb{F}_q$

para el caso semisimple, pero no permite la correcion de todos los errores dentro de la capacidad correctora como si lo hace nuestro método.

En [9], Bernal et al. estudiaron un conjunto de información para códigos grupo abelianos en el caso semisimple. En [10] y también en el caso semisimple, establecieron condiciones que garantizan que el grupo Θ de multiplicaciones a la izquierda de G contiene un PD-conjunto parcial de un $G-$código (abeliano) con respecto al conjunto de información estudiado previamente. En el mismo trabajo, dado un $G-$código $\mathfrak{C}$ de $\mathbb{F}_q G$, se dieron condiciones para que el grupo $\mathcal{P} = \langle \Theta, \tau_q \rangle$ sea un $s-$PD-conjunto parcial, donde $s = 2, 3$, de $\mathfrak{C}$ con respecto al mismo conjunto de información. Podríamos aplicar nuestro método a códigos grupo abelianos que no necesariamente satisfacen tales condiciones.

4.5. Generalizando el algoritmo en códigos grupo

Ahora veremos cómo podemos generalizar el algoritmo de descodificación por permutación en aquellos códigos grupo en los cuales el grupo $\mathrm{RPAut}(\mathfrak{C})$ del Corolario 4.3.5 es difícil de calcular (por ejemplo, cuando $\mathrm{Aut}(G)$ es desconocido o sensiblemente mayor que n^2) o en aquellos códigos grupo en los cuales el conjunto $\mathcal{P}$, del Teorema 4.3.7, no es un PD-conjunto.

Para ello, retomaremos la notación de las secciones 4.2 y 4.3. Supongamos que $\mathbf{r} = \mathbf{c} + \mathbf{e}$ es la palabra recibida, donde $\mathbf{c} \in \mathfrak{C}$ y $\mathbf{e} \in \mathbb{K}G$. Por la Proposición 4.3.3, si $\psi \in \mathrm{Aut}(G)$, entonces ψ se extiende a un automorfismo del álgebra $\mathbb{K}G$, también denotado por ψ , y por lo tanto, $\psi(\mathfrak{C}) = \langle \psi(e_0) \rangle$ también es un código grupo equivalente a $\mathfrak{C}$. Dado que $\psi(S^+(\mathbf{r})) = \psi(\mathbf{r}e_0^+) = \psi(\mathbf{r})\psi(e_0^+)$, y $\psi(e_0^+)$ es el idempotente que es ortogonal a $\psi(\mathfrak{C})$, entonces $\psi(S^+(\mathbf{r}))$ es el $\mathbb{K}G-$síndrome de $\psi(\mathbf{r})$ con respecto a $\psi(\mathfrak{C})$, y

$$\psi(S^+(\mathbf{r})) = \psi(\mathbf{c})\psi(e_0^+) + \psi(\mathbf{e})\psi(e_0^+) = \psi(\mathbf{e})\psi(e_0^+).$$

Además, si $\mathrm{wt}(\mathbf{e}) \leq t$, entonces $\mathrm{wt}(\psi(\mathbf{e})) = \mathrm{wt}(\mathbf{e}) \leq t$ y, por el Teorema 3.1.2, $\psi(\mathbf{e})$ es el único elemento de $\mathbb{K}G$, con peso menor o igual a t, que satisface la ecuación

$$\overline{\mathbf{x}}\psi(e_0^+) = \psi(S^+(\mathbf{r})). \tag{4.7}$$

En conclusión, recibida la palabra $\mathbf{r} \in \mathbb{K}G$, si $\mathbf{e}' \in \mathbb{K}G$ es solución de la ecuación (4.7) y tiene peso $\mathrm{wt}(\mathbf{e}') \leq t$, entonces el error producido es $\mathbf{e} = \psi^{-1}(\mathbf{e}')$.

Por consiguiente, para descodificar usaremos los códigos grupo que son obtenidos al aplicar a $\mathfrak{C}$ las extensiones lineales de automorfismos de G.

Conservando la notación anterior, si $C_{g_i}^+(\psi)$ es el vector columna de los coeficientes de $g_i\psi(e_0^+)$ para cada $g_i \in \{1,\ldots,n\}$,

$$\mathcal{C}_\psi^+(g_1,\ldots,g_n) := \begin{pmatrix} C_{g_1}^+(\psi) & \ldots & C_{g_n}^+(\psi) \end{pmatrix} \in M_{n\times n}(\mathbb{K}),$$

y $V^+(\psi)$ es el vector columna de los coeficientes de $\psi(S^+(\mathbf{r}))$, entonces resolver la ecuación (4.7) se reduce, en términos lineales, a resolver el sistema lineal de ecuaciones

$$\mathcal{C}_\psi^+(g_1,\ldots,g_n)\cdot \overline{X}^T = V_\psi^+. \tag{4.8}$$

En virtud de la Proposición 3.4.1, la matriz $\mathcal{C}_\psi^+(g_1,\ldots,g_n)$ y la matriz extendida

$$\mathcal{M}_\psi^+(g_1,\ldots,g_n) := \left(\mathcal{C}_\psi^+(g_1,\ldots,g_n) \mid V_\psi^+\right),$$

tienen rango igual a $n-k$ y, en consecuencua, podemos calcular su matriz escalonada reducida, que denotamos $\widetilde{\mathcal{C}}_\psi^R(g_1,\ldots,g_n)$ y $\widetilde{\mathcal{M}}_\psi^R(g_1,\ldots,g_n)$ respectivamente. De acuerdo al hecho anterior, existe una matriz invertible $\mathcal{U}_\psi \in M_{n\times n}(\mathbb{K})$ tal que

$$\mathcal{U}_\psi\cdot\mathcal{C}_\psi^+(g_1,\ldots,g_n) = \widetilde{\mathcal{C}}_\psi^R(g_1,\ldots,g_n)$$

y

$$\widetilde{\mathcal{M}}_\psi^R(g_1,\ldots,g_n) = \left(\mathcal{U}_\psi\cdot\mathcal{C}_\psi^+(g_1,\ldots,g_n) \mid \mathcal{U}_\psi\cdot V_\psi^+\right).$$

Así, el sistema lineal de ecuaciones

$$\widetilde{\mathcal{C}}_\psi^R(g_1,\ldots,g_n)\cdot\overline{X}^T = \mathcal{U}_\psi\cdot V_\psi^+$$

tiene las mismas soluciones que el sistema (4.8). Como $\psi(\mathbf{r})$ es una solución de (4.8), entonces $\widetilde{\mathcal{C}}_\psi^R(g_1,\ldots,g_n)\cdot\psi(\mathbf{r})^T = \mathcal{U}_\psi\cdot V_\psi^+$ y podemos definir

$$\widetilde{V}_\psi^R := \mathcal{U}_\psi\cdot V_\psi^+.$$

De donde, $\widetilde{V}_\psi^R = \widetilde{\mathcal{C}}_\psi^R(g_1,\ldots,g_n)\cdot\psi(\mathbf{r})^T$ y sus últimas k filas son también nulas. Si eliminamos las últimas k filas nulas $\widetilde{\mathcal{C}}_\psi^R(g_1,\ldots,g_n)$ y $\widetilde{V}_\psi^R$, y denotamos las matrices resultantes por $\mathcal{C}_\psi^R(g_1,\ldots,g_n)$ y V_ψ^R respectivamente, entonces resolver la ecuación (4.8), es equivalente a resolver

$$\mathcal{C}_\psi^R(g_1,\ldots,g_n)\cdot\overline{X}^T = V_\psi^R. \tag{4.9}$$

Si los pivotes de $\widetilde{\mathcal{C}}_\psi^R(g_1,\ldots,g_n)$ están en las posiciones $(1,l_1^\psi),\ldots,(n-k,l_{n-k}^\psi)$ y $L(\psi)=\{l_1^\psi,\ldots,l_{n-k}^\psi\}$, entonces $G_{L(\psi)} = \{g_l \in G : l \in L(\psi)\}$ es un conjunto de

posiciones de control de $\psi(\mathfrak{C})$ y, en consecuencia, $G \backslash G_{L(\psi)}$ es un conjunto de información de $\psi(\mathfrak{C})$ y $\mathcal{C}_{\psi}^{R}(g_1, \ldots, g_n)$ es una matriz de control de $\psi(\mathfrak{C})$ que tiene la matriz identidad $\mathcal{I}_{n-k}$ en las posiciones indicadas por los elementos de $G_{L(\psi)}$.

Con lo cual, podemos aplicar un método de descodifcación por permutación en $\psi(\mathfrak{C})$ para resolver la ecuación (4.9). Si logramos hallar una solución de la ecuación (4.7), con peso menor o igual a t, es posible descodificar $\mathbf{r} \in \mathbb{K}G$. Por el Teorema 4.1.3, si $\mathrm{wt}(\mathbf{e}) \leq t$, entonces las posiciones de información de $\mathbf{r}$ no tienen errores si y solo si $\mathrm{wt}\left(\mathcal{C}_{\psi}^{R}(g_1, \ldots, g_n) \cdot (\psi(\mathbf{r}))^T\right) \leq t$. Por tanto, la idea del algoritmo generalizado de descodificación por permutación consiste en ir aplicando la técnica de descodificación por permutación en cada uno de los códigos $\psi(\mathfrak{C})$ hasta encontrar el error o hasta determinar que el número de errores producidos es superior a la capacidad correctora. Vamos a considerar el grupo Θ de la Proposición 4.3.1, dado que, $\Theta \subseteq \mathrm{PAut}(\psi(\mathfrak{C}))$ para todo $\psi \in \mathrm{Aut}(G)$ (independientemente si G es abeliano o no).

Definición 4.5.1. *Sean $\Psi \leq Aut(G)$ y $\Omega = \langle \Theta, \Psi \rangle$. Decimos que $\mathcal{Q} \subseteq \Omega$ es un GPD-conjunto de $\mathfrak{C}$, si para todo $t-$subconjunto T de G, existe $\sigma \in \mathcal{Q}$, donde $\sigma = \theta_i \psi$, $\theta_i \in \Theta$ y $\psi \in \Psi$, tales que $\sigma(T) \cap (G \setminus G_{L(\psi)}) = \emptyset$.*

Dado que la expresión $\sigma = \theta_i \psi$, donde $\theta_i \in \Theta$ y $\psi \in \Psi$, es única, la noción de GPD-conjunto está bien definida. Además, en presencia de un GPD-conjunto, el algoritmo generalizado de descodificación por permutación funciona adecuadamente. Podemos reformular la caracterización para los GPD-conjuntos de la siguiente manera.

Teorema 4.5.2. *Sean $\Psi \leq Aut(G)$ y $\sim$ la relación de equivalencia sobre $\Omega = \langle \Theta, \Psi \rangle$ dada por*

$$\sigma_1 \sim \sigma_2 \text{ si y solo si } \sigma_1(G_{L(\psi)}) = \sigma_2(G_{L(\psi')}),$$

siendo $\sigma_1 = \theta_i \psi'$ y $\sigma_2 = \theta_j \psi'$. Si $\mathcal{Q}$ es un sistema completo de representantes del conjunto cociente $\Omega / \sim$, y para cada $\sigma \in \mathcal{Q}$, donde $\sigma = \theta_i \psi$, $\theta_i \in \Theta$ y $\psi \in \Psi$, $\mathcal{J}_\sigma$ denota el conjunto de todos los $t-$subconjuntos de $\sigma(G_{L(\psi)})$, entonces $\mathcal{Q}$ es un GPD-conjunto de $\mathfrak{C}$ si y solo si

$$\left| \bigcup_{\sigma \in \mathcal{Q}} \mathcal{J}_\sigma \right| = \binom{n}{t}. \tag{4.10}$$

Demostración. Es análoga a la prueba del Teorema 4.3.7. □

Ejemplo 4.5.3. *Consideremos* $G = C_3 \times C_3 \times C_3 = \langle a, b, c \rangle$ *y* $\mathbb{K} = \mathbb{F}_7$. *En la base vamos a fijar el siguiente orden:*

$$\{1_G, a, b, c, a^2, ab, ac, b^2, bc, c^2, a^2b, a^2c, ab^2, abc, ac^2, b^2c, bc^2,$$
$$a^2b^2, a^2bc, a^2c^2, ab^2c, abc^2, b^2c^2, a^2b^2c, a^2bc^2, ab^2c^2, a^2b^2c^2\}.$$

Si $\mathfrak{C} = \langle e_0 \rangle$, *donde*

$$e_0 = (2, 6, 3, 6, 0, 3, 1, 6, 3, 0, 3, 5, 6, 5, 5, 6, 3, 6, 2, 1, 3, 2, 6, 4, 0, 4, 0),$$

entonces $n = 27$, $k = 5$ *y* $t = 7$. *Calculamos que* $RPAut(\mathfrak{C}) = \langle \lambda \rangle$, *donde*

$$\begin{array}{rccc} \lambda : G & \to & G \\ a & \mapsto & c \\ b & \mapsto & b \\ c & \mapsto & a \end{array}.$$

De lo anterior, el conjunto $\Phi = \langle \Theta, \Lambda \rangle$, *donde* $\Lambda = RPAut(\mathfrak{C})$, *no es un PD−conjunto de* $\mathfrak{C}$ *con respecto a* $G \setminus G_L$. *Sin embargo, tomando* $\Psi = \langle \psi \rangle$, *donde*

$$\begin{array}{rccc} \psi : G & \to & G \\ a & \mapsto & a \\ b & \mapsto & b^2 \\ c & \mapsto & c, \end{array}$$

implica que el conjunto $\mathcal{Q} = \Theta\Psi$ *satisface el Teorema 4.5.2 y, por lo tanto, es un GPD−conjunto de* $\mathfrak{C}$.

4.5.1. Algoritmo generalizado de descodificación por permutación (Algoritmo GPD)

Supongamos que $\Psi \leq \mathrm{Aut}(G)$ y $\Omega = \langle \Theta, \Psi \rangle$. Si $\mathcal{Q} \subseteq \Omega$ es un GPD-conjunto de $\mathfrak{C}$, consideramos $\mathcal{C}^R_\psi(g_1, \ldots, g_1)$ y $L(\psi) = \{l_1(\psi), \ldots, l_{n-k}(\psi)\}$, donde ψ recorre el conjunto de todos los elementos $\psi \in \Psi$ tales que $\theta_i\psi \in \mathcal{Q}$ para algún $\theta_i \in \Theta$. El *algoritmo generalizado de descodificación por permutación* es el siguiente:

Recibida la palabra $\mathbf{r} \in \mathbb{K}G$, se toma aleatoriamente $\sigma \in \mathcal{Q}$, donde $\sigma = \theta_i\psi$, $\theta_i \in \Theta$ y $\psi \in \Psi$. Se calcula

$$\mathrm{Syn}(\sigma(\mathbf{r}))^\psi := \mathcal{C}^R_\psi(g_1, \ldots, g_1) \cdot \sigma(\mathbf{r})^T.$$

Suponga que $\mathrm{Syn}(\sigma(\mathbf{r}))^\psi = (e''_1, \ldots, e''_{n-k})$.

- Si $\mathrm{wt}(\mathrm{Syn}(\sigma(\mathbf{r})))^{\psi} \leq t$, se define $\mathbf{e}' := (e'_i) \in \mathbb{K}^n$ donde

$$e'_i = \begin{cases} e''_j, \text{ si } i = l_j(\psi) \text{ para algún } j \in \{1, \ldots, n-k\}, \\ 0, \text{ en caso contrario.} \end{cases}$$

  Entonces, el error es $\mathbf{e} = \sigma^{-1}(\mathbf{e}')$ y el algoritmo termina.

- De lo contrario, se descarta σ, y se repite de nuevo el proceso con otro elemento de $\mathcal{Q}$.

El algoritmo termina cuando se encuentra $\sigma = \theta_i\psi \in \mathcal{Q}$, con $\theta_i \in \Theta$ y $\psi \in \Psi$, tal que

$$\mathrm{wt}\left(\mathrm{Syn}(\sigma(\mathbf{r}))^{\psi}\right) \leq t, \tag{4.11}$$

o cuando todos los elementos de $\mathcal{Q}$, han sido verificados y ninguno satisface (4.11). En este último caso, la cantidad de errores ocurridos supera la capacidad correctora del código y, por consiguiente, el algoritmo no puede descodificar $\mathbf{r}$.

Ejemplo 4.5.4. *Tomemos el código grupo $\mathfrak{C}$ de $\mathbb{F}_7(C_3 \times C_3 \times C_3)$ en el Ejemplo 4.5.3. Para descodificar usaremos el GPD-conjunto $\mathcal{Q} = \Theta\Psi$ definido en el mismo ejemplo. Si*

$$\mathbf{r} = (3, 1, 6, 1, 0, 5, 1, 2, 5, 0, 6, 5, 3, 4, 5, 3, 6, 2, 1, 1, 5, 1, 0, 2, 3, 2, 3),$$

entonces tomando θ_3, la multiplicación a izquierda por $b \in C_3 \times C_3 \times C_3$, y $\sigma = \theta_3\psi$, tenemos que

$$Syn(\sigma(\mathbf{r}))^{\psi} = (5, 0, 5, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 4, 0, 5, 0, 2),$$

y por lo tanto,

$$\begin{aligned} \mathbf{e}' &= (5, 0, 5, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 4, 0, 5, 0, 0, 2, 0, 0, 0, 0) \\ &= 5 + 5b + 5ac + abc + 4a^2b^2 + 5a^2c^2 + 2b^2c^2. \end{aligned}$$

Así,

$$\begin{aligned} \mathbf{e} &= \sigma^{-1}(5 + 5b + 5ac + abc + 4a^2b^2 + 5a^2c^2 + 2b^2c^2) \\ &= \psi^{-1}\theta_3^{-1}(5 + 5b + 5ac + abc + 4a^2b^2 + 5a^2c^2 + 2b^2c^2) \\ &= \psi^{-1}(b^2(5 + 5b + 5ac + abc + 4a^2b^2 + 5a^2c^2 + 2b^2c^2)) \\ &= \psi^{-1}(5b^2 + 5 + 5ab^2c + ac + 4a^2b + 5a^2b^2c^2 + 2bc^2) \\ &= 5b + 5 + 5abc + ac + 4a^2b^2 + 5a^2bc^2 + 2b^2c^2 \\ &= 5 + 5b + ac + 5abc + 4a^2b^2 + 2b^2c^2 + 5a^2bc^2 \\ &= (5, 0, 5, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 4, 0, 0, 0, 0, 2, 0, 5, 0, 0). \end{aligned}$$

4.5.2. Análisis de complejidad

Dado $\Psi \subseteq \mathrm{Aut}(G)$, previo al algoritmo, debemos calcular $\mathcal{C}_{\psi}^{R}(g_1, \ldots, g_1)$ para cada $\psi \in \Psi$. Tal cálculo tiene orden de complejidad $\mathcal{O}(\gamma \times n^3)$, donde $\gamma = |\Psi|$. De lo anterior, obtenemos un total de γ conjuntos de posiciones de control $G_{L(\psi)}$.

En consecuencia, calcular el conjunto $\mathcal{Q}$ del Teorema 4.5.2 tiene orden de complejidad

$$\mathcal{O}(\gamma^2 \times n^3). \tag{4.12}$$

Si suponemos que $|\mathcal{Q}| = \varrho$, entonces comprobar si $\mathcal{Q}$ es un GPD-conjunto tiene orden de complejidad

$$\mathcal{O}\left(\varrho \times \binom{n-k}{t}\right). \tag{4.13}$$

Las expresiones (4.12) y (4.13) se demuestran de la misma forma como lo hicimos con las expresiones (4.5) y (4.6), respestivamente.

Finalmente, si $\mathcal{Q}$ es un GPD-conjunto y razonamos de forma análoga para obtener la expresión (4.2), deducimos que el orden de complejidad del Algoritmo GPD es $\mathcal{O}(\varrho \times n^3)$.

Capítulo 5

Códigos grupo LDOI

Los códigos LDPC (Low-Density Parity-Check) son un tipo de códigos lineales binarios que se caracterizan por tener una matriz de control con una cantidad muy pequeña de entradas no nulas (unos). Fueron definidos por R. Gallager en [21] y posteriormente estudiados en [38] y [60]. Un código LDPC se construye garantizando que posee una matriz de control cuyas columnas tienen exactamente el mismo número v de entradas no nulas. La elección de v y la estructura específica de tal matriz tienen un impacto significativo en el rendimiento y complejidad de los distintos algoritmos de descodificación que han sido diseñados.

Uno de los algoritmos de descodificación más conocidos y eficientes para este tipo de códigos es el llamado *Bit Flipping* (BF). Tal algoritmo se aplica de forma iterada hasta corregir los bits erróneos (posiciones de error) en el mensaje recibido. En cada iteración, el algoritmo verifica ciertas restricciones de paridad y corrige los bits que no las satisfacen. Luego, se vuelve a comprobar si cumplen otras restricciones de paridad. Si no es así, el algoritmo se repite iteradamente hasta que se cumplan todas las restricciones o se alcance un límite predefinido de iteraciones ([21, 38, 60]). La ventaja de usar el algoritmo BF es que es simple y eficiente.

El propósito de este capítulo es hacer un análisis similar al hecho en [56] y [57], en códigos LDPC, para diseñar e implementar un algoritmo análogo en códigos grupo.

5.1. Definiciones y Propiedades

En esta sección, el cuerpo base será siempre $\mathbb{K} = \mathbb{F}_2$ y presentaremos la familia de códigos grupo LDOI (Low-Density Orthogonal Idempotent). Estos

se pueden considerar como los análogos de los códigos LDPC en el contexto de los códigos grupo. De ahora en adelante, G es un grupo finito de orden n impar. Con esto, el álgebra de grupo $\mathbb{F}_2G$ es semisimple y todo código grupo tiene un sumando directo. Tanto el código como su sumando directo estan generados por idempotentes centrales de $\mathbb{F}_2G$. Siguiendo el Capítulo 3, $\mathfrak{C}$ denota un código grupo de $\mathbb{F}_2G$ y $e_0^+ \in \mathbb{F}_2G$ el idempotente que es ortogonal a $\mathfrak{C}$.

Definición 5.1.1. *Se dice que $\mathfrak{C}$ es un código grupo $v-$LDOI (con idempotente ortogonal de peso pequeño v) si $v = \mathrm{wt}(e_0^+)$ es sensiblemente menor que n.*

De acuerdo con la definición anterior, si $C_{g_i}^+$ es el vector columna de los coeficientes de $g_i e_0^+$, entonces la matriz (ver Capítulo 3)

$$\mathcal{C}^+(g_1, \ldots, g_n) = \begin{pmatrix} C_{g_1}^+ & \ldots & C_{g_n}^+ \end{pmatrix},$$

tiene el mismo número v de entradas no nulas en cada columna. En efecto, los coeficientes de las columnas de tal matriz son permutaciones de los coeficientes de e_0^+ correspondientes, a la multiplicación a izquierda de $g_i \in G$. Como $\mathcal{C}^+(g_1, \ldots, g_n)$ se comporta como una matriz de control de $\mathfrak{C}$ (ver Sección 4.2), al suponer que v es mucho más pequeño que n, tenemos que $\mathfrak{C}$ se asemeja a un código LDPC.

Suponiendo que $\mathfrak{C}$ es un código grupo $v-$LDOI generado por el idempotente $e_0 \in \mathbb{F}_2G$, tenemos que $e_0 = 1 + e_0^+$. Si $v = 1$, entonces $e_0^+ = 1$, y por lo tanto $\mathfrak{C} = \{0\}$. Con lo cual, de aquí en adelante imponemos que $v \neq 1$. Además, si d es la distancia mínima de $\mathfrak{C}$, entonces

$$d \leq \mathrm{wt}(e_0) \leq \mathrm{wt}\left(e_0^+\right) + \mathrm{wt}\,(1) = v + 1. \tag{5.1}$$

Notemos que $\mathrm{wt}(e_0) = \mathrm{wt}(e_0^+) + 1$, si $1 \notin \mathrm{Supp}(e_0^+)$.

En lo que sigue, buscaremos condiciones en el código grupo que permitan obtener la igualdad $d = v + 1$, así como otras propiedades útiles para la decodificación. Tales propiedades están ligadas a algunas características de su matriz de adyacencia.

Definición 5.1.2. *La matriz de adyacencia de $\mathfrak{C}$ es la matriz $\mathcal{D} \in M_{n\times n}(\mathbb{Z})$ cuya $(i,j)-$ésima entrada $d_{i,j}$ está dada por*

$$d_{i,j} := \begin{cases} 0 & \text{si } i = j, \\ \left|Supp(C_{g_i}^+) \cap Supp(C_{g_j}^+)\right| & \text{si } i \neq j. \end{cases}$$

Si $d_{i,j} \in \{0,1\}$ para todo par $i, j \in \{1, \ldots, n\}$, se dice que $\mathcal{D}$ es binaria.

Notemos que el cálculo de la matriz de adyacencia $\mathcal{D}$ de un código grupo de $\mathbb{F}_2G$ tiene orden de complejidad $\mathcal{O}(vn^2)$. Dicha matriz es simétrica y las entradas de cada fila se obtienen mediante permutaciones de las entradas de su primera fila. En efecto, sea $d_{i,j}$ el elemento en la fila i y columna j de la matriz $\mathcal{D}$. Entonces, existen $\mathbf{z} \in \mathbb{F}_2G$ con peso $\mathrm{wt}(\mathbf{z}) = d_{i,j}$ y $\mathbf{z}_i, \mathbf{z}_j \in \mathbb{F}_2G$ con soportes disjuntos tales que

$$\begin{aligned} g_i e_0^+ &= \mathbf{z} + \mathbf{z}_i \text{ y } \mathrm{Supp}(\mathbf{z}) \cap \mathrm{Supp}(\mathbf{z}_i) = \emptyset. \\ g_j e_0^+ &= \mathbf{z} + \mathbf{z}_j \text{ y } \mathrm{Supp}(\mathbf{z}) \cap \mathrm{Supp}(\mathbf{z}_j) = \emptyset. \end{aligned}$$

De donde se obtiene que

$$\begin{aligned} e_0^+ &= g_i^{-1}\mathbf{z} + g_i^{-1}\mathbf{z}_i \text{ y } \mathrm{Supp}(g_i^{-1}\mathbf{z}) \cap \mathrm{Supp}(g_i^{-1}\mathbf{z}_i) = \emptyset. \\ g_i^{-1}(g_j e_0^+) &= g_i^{-1}\mathbf{z} + g_i^{-1}\mathbf{z}_j \text{ y } \mathrm{Supp}(g_i^{-1}\mathbf{z}) \cap \mathrm{Supp}(g_i^{-1}\mathbf{z}_j) = \emptyset. \end{aligned}$$

Además, $g_i^{-1}\mathbf{z}_i$ y $g_i^{-1}\mathbf{z}_j$ tienen soportes disjuntos. Si $h \in \{1, \ldots, n\}$ es el índice tal que $g_h = g_i^{-1}g_j$, entonces $d_{1,h} = \mathrm{wt}(g_i^{-1}\mathbf{z}) = \mathrm{wt}(\mathbf{z}) = d_{i,j}$.

Ejemplo 5.1.3. *Supongamos que* $\mathbb{K} = \mathbb{F}_2$ *y* $G = C_7 \times C_3 = \langle a, b \rangle$. *Si fijamos la base*

$$\{1, a, b, a^2, ab, b^2, a^3, a^2b, ab^2, a^4, a^3b, a^2b^2, a^5, a^4b, a^3b^2, a^6, a^5b, a^4b^2, a^6b, a^5b^2, a^6b^2\}$$

y consideramos el código grupo $\mathfrak{C}$ *cuyo idempotente ortogonal es*

$$e_0^+ = 1 + b + b^2 + a^3b + a^3b^2 + a^5b + a^6b + a^5b^2 + a^6b^2,$$

entonces

$$\begin{aligned} C_1^+ &= (1,0,1,0,0,1,0,0,0,0,1,0,0,0,1,0,1,0,1,1,1)^T, \\ C_a^+ &= (0,1,1,0,1,1,0,0,1,0,0,0,0,1,0,0,0,1,1,0,1)^T, \\ C_b^+ &= (1,0,1,0,0,1,1,0,0,0,0,0,1,0,1,1,0,0,0,1,1)^T, \\ C_{a^2}^+ &= (0,0,1,1,1,1,0,1,1,0,0,1,0,0,0,0,1,0,0,1,0)^T, \\ C_{ab}^+ &= (1,1,0,0,1,1,0,0,1,1,0,0,0,0,0,1,0,1,0,0,1)^T, \\ C_{b^2}^+ &= (1,0,1,0,0,1,1,0,0,0,1,0,1,0,0,1,1,0,1,0,0)^T, \\ C_{a^3}^+ &= (0,0,0,0,1,0,1,1,1,0,1,1,0,0,1,0,0,0,1,0,1)^T, \\ C_{a^2b}^+ &= (1,1,0,1,0,1,0,1,1,0,0,1,1,0,0,0,0,0,0,1,0)^T, \\ C_{ab^2}^+ &= (1,1,1,0,1,0,0,0,1,1,0,0,0,1,0,1,0,0,1,0,0)^T, \end{aligned}$$

$$C^+_{a^4} = (0,0,1,0,0,1,0,1,0,1,1,1,0,1,1,0,0,1,0,0,0)^T,$$
$$C^+_{a^3b} = (0,1,0,1,0,0,1,0,1,0,1,1,0,0,1,1,0,0,0,0,1)^T,$$
$$C^+_{a^2b^2} = (1,1,1,1,1,0,0,1,0,0,0,1,1,0,0,0,1,0,0,0,0)^T,$$
$$C^+_{a^5} = (0,0,0,0,1,0,0,0,1,0,1,0,1,1,1,0,1,1,0,1,0)^T,$$
$$C^+_{a^4b} = (1,0,0,1,0,1,1,0,0,1,0,1,0,1,1,0,0,1,0,0,0)^T,$$
$$C^+_{a^3b^2} = (0,1,0,1,1,0,1,1,0,0,1,0,0,0,1,1,0,0,1,0,0)^T,$$
$$C^+_{a^6} = (0,0,0,0,0,0,0,1,0,0,0,1,0,1,0,1,1,1,1,1,1)^T,$$
$$C^+_{a^5b} = (0,1,0,0,0,0,1,0,1,1,0,0,1,0,1,0,1,1,0,1,0)^T,$$
$$C^+_{a^4b^2} = (1,0,1,1,0,0,1,1,0,1,1,0,0,1,0,0,0,1,0,0,0)^T,$$
$$C^+_{a^6b} = (0,0,0,1,0,0,0,0,0,1,0,1,1,0,0,1,0,1,1,1,1),$$
$$C^+_{a^5b^2} = (0,1,0,0,1,0,1,0,0,1,1,0,1,1,0,0,1,0,0,1,0)^T,$$
$$C^+_{a^6b^2} = (0,0,0,1,0,0,0,1,0,1,0,0,1,1,0,1,1,0,1,0,1)^T.$$

De donde, la matriz de adyacencia de $\mathfrak{C}$ *es*

$$\mathcal{D} = \begin{pmatrix}
0 & 4 & 6 & 4 & 3 & 6 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 3 & 3 & 3 \\
4 & 0 & 3 & 4 & 6 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 3 & 3 & 3 \\
6 & 3 & 0 & 3 & 4 & 6 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 \\
4 & 4 & 3 & 0 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 4 & 3 & 3 & 3 & 3 & 3 \\
3 & 6 & 4 & 3 & 0 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 \\
6 & 3 & 6 & 3 & 3 & 0 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 \\
4 & 4 & 3 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 4 & 6 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 \\
3 & 6 & 3 & 3 & 6 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 \\
4 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 6 & 3 & 3 & 3 \\
3 & 3 & 4 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 4 & 3 & 3 \\
3 & 3 & 3 & 6 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 \\
4 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 6 & 3 & 3 & 6 & 3 \\
3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 6 & 4 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 0 & 3 & 3 & 4 & 3 & 4 & 4 \\
4 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 0 & 3 & 3 & 6 & 3 & 6 \\
3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 0 & 3 & 4 & 6 & 3 \\
3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 6 & 4 & 3 & 3 & 0 & 3 & 4 & 4 \\
3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 6 & 4 & 3 & 0 & 3 & 6 \\
3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 6 & 3 & 4 & 3 & 6 & 4 & 3 & 0 & 4 \\
3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 6 & 3 & 4 & 6 & 4 & 0
\end{pmatrix}.$$

Ahora bien, si $\mathbf{z} = g_{i_1} + \cdots + g_{i_u}$ tiene peso u, entonces para calcular $\mathbf{z}e_0^+$, es necesario sumar las columnas de la matriz

$$\mathcal{C}^+(g_{i_1}, \ldots, g_{i_u}) = \begin{pmatrix} C^+_{g_{i_1}} & \ldots & C^+_{g_{i_u}} \end{pmatrix}.$$

La matriz anterior tiene uv entradas distintas de cero. Para cada $l \in \{1, \ldots, n\}$, denotamos a γ_l como el peso de la $l-$ésima fila de $\mathcal{C}^+(g_{i_1}, \ldots, g_{i_u})$. Entonces

$$uv = \sum_{l=1}^{n} \gamma_l.$$

Como $g_{i_l} \in \mathrm{Supp}(\mathbf{z}e_0^+)$ si y solo si γ_l es impar, para cada $l \in \{1, \ldots, n\}$, definimos

$$\rho_l := \begin{cases} \gamma_l & \text{si } \gamma_l \text{ es par,} \\ \gamma_l - 1 & \text{si } \gamma_l \text{ es impar.} \end{cases}$$

Entonces,

$$\mathrm{wt}(\mathbf{z}e_0^+) = \sum_{l=1}^{n} \gamma_l - \sum_{l=1}^{n} \rho_l = uv - \sum_{l=1}^{n} \rho_l.$$

Además, ρ_l es par y en la fila l de $\mathcal{C}^+(g_{i_1}, \ldots, g_{i_u})$ hay al menos $\rho_l/2$ parejas de columnas que tienen un 1 en esa posición. Por otra parte $d_{i,j} \in \mathcal{D}$, con $i \neq j$, representa el número de posiciones l en las que ambas columnas, la $i-$ésima y la $j-$ésima, tienen simultaneamente un 1. Por lo tanto,

$$\sum_{l=1}^{n} \left(\frac{\rho_l}{2}\right) \leq \sum_{i<j} d_{i,j},$$

donde la suma del lado derecho recorre todos los índices i, j tales que $g_i, g_j \in \mathrm{Supp}(\mathbf{z})$, y por consiguiente,

$$\mathrm{wt}(\mathbf{z}e_0^+) \geq uv - 2 \cdot \sum_{i<j} d_{i,j}.$$

Proposición 5.1.4. *Sean $\mathfrak{C}$ un código grupo $v-$LDOI y $\mathcal{D}$ su matriz de adyacencia. Supongamos que $\mathbf{z} \in \mathbb{F}_2 G$ tiene peso u. Entonces $\mathbf{z} \in \mathfrak{C}$ si y solo si*

$$\sum_{i<j} d_{i,j} \geq \frac{uv}{2}, \tag{5.2}$$

donde la suma recorre todos los índices i, j tales que $g_i, g_j \in$ Supp$(\mathbf{z})$.

En el resto de esta sección vamos a suponer que $\mathfrak{C} = \langle e_0 \rangle$ es un código grupo $v-$LDOI. Como las columnas y las filas de la matriz

$$\mathcal{C}^+(g_1, \ldots, g_n) = \begin{pmatrix} C^+_{g_1} & \ldots & C^+_{g_n} \end{pmatrix},$$

están en correspondencia biyectiva (ver Proposición 4.2.1) bajo la aplicación $\varphi : \mathbb{K}G \to \mathbb{K}G$, dada por

$$\varphi\left(\sum_{i=1}^{n} \alpha_i g_i\right) := \sum_{i=1}^{n} \alpha_i g_i^{-1},$$

entonces tales filas y columnas tienen el mismo peso v, y para cada $g_i \in G$, definimos $\mathcal{C}^{(i)}$ como la submatriz de $\mathcal{C}^+(g_1, \ldots, g_n)$ formada al suprimir las filas que tienen 0 en la posición i. También, definimos $C^{(i)}_j$ como la $j-$ésima columna de $\mathcal{C}^{(i)}$.

Si $\mathcal{D}$ es la matriz de adyacencia de $\mathfrak{C}$, entonces $\mathrm{wt}\left(C^{(i)}_j\right) = d_{i,j}$ para cada $j \neq i$. Además, si R_l denota la $l-$ésima fila de $\mathcal{C}^+(g_1, \ldots, g_n)$, entonces R_l tiene exactamente v entradas no nulas y

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} \mathrm{wt}\left(C^{(i)}_j\right) = \sum_{l \in \mathrm{Supp}\left(C^+_{g_i}\right)} \left|\mathrm{Supp}\left(R_l\right) \setminus \{i\}\right|. \tag{5.3}$$

Ejemplo 5.1.5. *En el ejemplo anterior, se verifica que*

$$Supp(C^+_{a^3b}) = \{2, 4, 7, 9, 11, 12, 15, 16, 21\}.$$

Con esto,

$$\mathcal{C}^{(11)} = \begin{pmatrix} 0&1&0&0&1&0&0&1&1&0&1&1&0&0&1&0&1&0&0&1&0 \\ 0&0&0&1&0&0&0&1&0&0&1&1&0&1&1&0&0&1&1&0&1 \\ 0&0&1&0&0&1&1&0&0&0&1&0&0&1&1&0&1&1&0&1&0 \\ 0&1&0&1&1&0&1&1&1&0&1&0&1&0&0&0&1&0&0&0&0 \\ 1&0&0&0&0&1&1&0&0&1&1&0&1&0&1&0&0&1&0&1&0 \\ 0&0&0&1&0&0&1&1&0&1&1&1&0&1&0&1&0&0&1&0&0 \\ 1&0&1&0&0&0&1&0&0&1&1&0&1&1&1&0&1&0&0&0&0 \\ 0&0&1&0&1&1&0&0&1&0&1&0&0&0&1&1&0&0&1&0&1 \\ 1&1&1&0&1&0&1&0&0&0&1&0&0&0&0&1&0&0&1&0&1 \end{pmatrix}.$$

En particular se verifica que $wt\left(C^{(11)}_2\right) = 3 = d_{11,2}$. *Además,*

$$\sum_{\substack{j=1 \\ j\neq i}}^{21} wt\left(C^{(11)}_j\right) = \sum_{l \in Supp(C^+_{a^3b})} |Supp\left(R_l\right) \setminus \{11\}| = 72$$

es la cantidad de entradas no nulas en la matriz $\mathcal{C}^{(11)}$ sin contar los elementos de undécima columna.

Proposición 5.1.6. *Si $\mathcal{D}$ es la matriz de adyacencia de $\mathfrak{C}$, entonces la suma de los elementos de cada fila de $\mathcal{D}$ es igual a $v(v-1)$.*

Demostración. Basta hacer la prueba para la primera fila de $\mathcal{D}$. En efecto,

$$\begin{aligned}\sum_{j=1}^{n} d_{1,j} = d_{1,1} + \sum_{j=2}^{n} d_{1,j} &= 0 + \sum_{j=2}^{n} \mathrm{wt}\left(C_j^{(1)}\right) \\ &= \sum_{l \in \mathrm{Supp}\left(C_{g_1}^{+}\right)} |\mathrm{Supp}\,(R_l) \setminus \{1\}| = \sum_{l \in \mathrm{Supp}\left(C_{g_i}^{+}\right)} (v-1) = v(v-1),\end{aligned}$$

puesto que, $\left|\mathrm{Supp}\left(C_{g_1}^{+}\right)\right| = v$. □

Teorema 5.1.7. *Sean $\mathfrak{C}$ un código grupo $v-$LDOI y $\mathcal{D}$ su matriz de adyacencia. Si $\mathcal{D}$ es binaria, entonces*

$$v(v-1) < n.$$

Demostración. Si $\mathcal{D}$ es binaria, entonces

$$\sum_{j=1}^{n} d_{1,j} \leq n-1.$$

Usando la proposición anterior tenemos que $v(v-1) \leq n-1$. □

La cota anterior es una condición (necesaria) que debemos tener en cuenta para construir códigos grupo $v-$LDOI con matriz de adyacencia binaria. Más adelante haremos algunas construcciones explícitas de códigos grupo con tal característica.

La siguiente noción es importante para lo que resta de esta sección y en la implementación del Algoritmo 1-BF en algunos códigos grupo $v-$LDOI.

Definición 5.1.8. *Si $\mathcal{D}$ es la matriz de adyacencia de $\mathfrak{C}$, los elementos de cualquiera de sus filas son los mismos. Con lo cual, al reordenar de mayor a menor, obtenemos el conjunto*

$$d_1 \geq d_2 \geq \cdots \geq d_n = 0.$$

Para cada $\ell \in \{1, \ldots, n\}$, definimos

$$\mu_{\mathcal{D}}(\ell) := \sum_{i=1}^{\ell} d_i.$$

Proposición 5.1.9. *Sea $\mathcal{D}$ la matriz de adyacencia de $\mathfrak{C}$. Si $\mathcal{D}$ es binaria, entonces $\mu_{\mathcal{D}}(\ell) = \ell$ para todo $\ell \leq v(v-1)$.*

Demostración. Por la Proposición 5.1.6, la suma de todas las entradas de cualquier fila de $\mathcal{D}$ es igual $v(v-1)$. De este modo,

$$d_1 = \cdots = d_{v(v-1)} = 1,$$

y, en consecuencia, para todo $\ell \leq v(v-1)$, se cumple que

$$\mu_{\mathcal{D}}(\ell) = \sum_{i=1}^{\ell} d_i = \underbrace{1 + \cdots + 1}_{\ell\text{-veces}} = \ell.$$

□

Ahora bién, si $\mathcal{D}$ es la matriz de adyacencia de $\mathfrak{C}$, entonces para cada $U = \{g_{i_1}, \ldots, g_{i_u}\} \subseteq G$, se sigue que

$$\begin{aligned}
\sum_{j=2}^{u} d_{i_1,i_j} &\leq \sum_{i=1}^{u-1} d_i = \mu_{\mathcal{D}}(u-1),\\
\sum_{j=3}^{u} d_{i_2,i_j} &\leq \sum_{i=1}^{u-2} d_i = \mu_{\mathcal{D}}(u-2),\\
&\vdots\\
d_{i_{u-1},i_u} &\leq d_1 = \mu_{\mathcal{D}}(1).
\end{aligned}$$

Así,

$$\sum_{\substack{g_i,g_j \in U\\ i<j}} d_{i,j} \leq \sum_{\ell=1}^{u-1} \mu_{\mathcal{D}}(\ell). \tag{5.4}$$

Teorema 5.1.10. *Sean $\mathfrak{C}$ un código grupo $v-LDOI$ y d su distancia mínima. Si la matriz de adyacencia de $\mathfrak{C}$ es binaria, entonces*

$$d = v + 1.$$

Demostración. En virtud de (5.1), tenemos que $d \leq v+1$. Para probar que $d \geq v+1$, supongamos que $\mathbf{z} \in \mathfrak{C}$ tiene soporte U y peso d. Sea $\mathcal{D}$ la matriz de adyacencia de $\mathfrak{C}$. Como $d - 1 \leq v \leq v(v-1)$, entonces por (5.4) y las Proposiciones 5.1.4 y 5.1.9, se deduce que

$$\frac{dv}{2} \leq \sum_{\substack{g_i,g_j \in U\\ i<j}} d_{i,j} \leq \sum_{\ell=1}^{d-1} \mu_{\mathcal{D}}(\ell) = \sum_{\ell=1}^{d-1} \ell = \frac{d(d-1)}{2}.$$

Lo anterior, nos permite obtener la desigualdad deseada. □

5.2. Descodificación para códigos grupo LDOI

En esta sección presentamos y justificamos el diseño e implementación de un algoritmo de descodificación para algunos códigos grupo LDOI.

Supongamos que se recibe la palabra $\mathbf{r} \in \mathbb{F}_2 G$. Dado que el $\mathbb{F}_2 G-$síndrome de $\mathbf{r}$ es igual a la suma de los elementos del conjunto $\{g_i e_0^+; g_i \in \mathrm{Supp}(\mathbf{r})\}$, entonces intentaremos determinar qué posiciones de la matriz $C^+(g_1, \ldots, g_n)$ influyen para que $S^+(\mathbf{r})$ sea distinto de cero.

Definición 5.2.1. *Con las notaciones anteriores, el contador de la $i-$ésima posición de $\mathbf{r}$, denotado $\gamma_i(\mathbf{r})$, se define por*

$$\gamma_i(\mathbf{r}) := \left| Supp(S^+(\mathbf{r})) \cap Supp(g_i e_0^+) \right|.$$

Si $\mathbf{e} \in \mathbb{F}_2 G$ es el error producido al recibir $\mathbf{r} \in \mathbb{F}_2 G$, entonces $\gamma_i(\mathbf{r}) = \gamma_i(\mathbf{e})$ y tal cantidad indica el número de veces que en el sistema

$$X_1 C_{g_1}^+ + \cdots + X_n C_{g_n}^+ = V^+,$$

aparece un "1" en la $i-$ésima posición de las ecuaciones en las que el término independiente es igual a 1.

Supongamos de ahora en adelante que $\mathfrak{C} = \langle e_0 \rangle$ es un código grupo $v-$LDOI. Si $\mathbf{e} \in \mathbb{F}_2 G$ es el error producido al recibir $\mathbf{r} \in \mathbb{F}_2 G$, $W = \mathrm{Supp}(\mathbf{e})$ y $g_i \in W$, entonces

$$\gamma_i(\mathbf{e}) = v - \mathrm{wt}\left(\sum_{g_j \in W \setminus \{g_i\}} C_j^{(i)} \right). \tag{5.5}$$

Por otro lado, si $g_i \notin W$, entonces

$$\gamma_i(\mathbf{e}) = \mathrm{wt}\left(\sum_{g_j \in W} C_j^{(i)} \right). \tag{5.6}$$

En efecto, si $g_i \in W$, entonces para calcular $\gamma_i(\mathbf{e})$ es suficiente tomar v y restar el cardinal del conjunto

$$L = \{l \in \mathrm{Supp}(C_{g_i}^+) : g_l \notin \mathrm{Supp}(S^+(\mathbf{e}))\}.$$

Como en la $i-$ésima columna de $\mathcal{C}^{(i)}$ solo hay "1", nos fijamos solo en aquellas filas de $\mathcal{C}^{(i)}$ tales que la suma de sus entradas, distintas de la $i-$ésima, sea igual a 1. Lo anterior es equivalente a sumar solamente las columnas de $\mathcal{C}^{(i)}$,

distintas de $i-$ésima, indexadas por las posiciones donde ocurrieron los errores (las demás no se consideran, dado que en tales posiciones no hay errores y, por lo tanto, no aportan ningún valor a la suma). Dicha suma es igual a

$$\sum_{g_j \in W \setminus \{g_i\}} C_j^{(i)}. \tag{5.7}$$

y tendrá tantas entradas no nulas como el cardinal de L. Así, para calcular $\gamma_i(\mathbf{e})$, tomamos v y restamos el peso de la suma (5.7).

Por otro lado, si $g_i \notin W$, para calcular $\gamma_i(\mathbf{e})$ es suficiente considerar el cardinal del conjunto

$$L' = \{l \in \mathrm{Supp}(C_{g_i}^+) : g_l \in \mathrm{Supp}(S^+(\mathbf{e}))\}.$$

Para hacer esto, nos fijamos en aquellas filas de $\mathcal{C}^{(i)}$ tales que la suma de sus entradas sea igual a 1. Esto es lo mismo que sumar aquellas columnas de $\mathcal{C}^{(i)}$ indexadas por las posiciones donde ocurrieron los errores. Esto es,

$$\sum_{g_j \in W} C_j^{(i)}. \tag{5.8}$$

En la suma anterior habrá tantas entradas no nulas como el cardinal de L'. Así, para calcular $\gamma_i(\mathbf{e})$, basta con hallar el peso de (5.8).

Proposición 5.2.2. *Sean $\mathcal{D}$ la matriz de adyacencia de $\mathfrak{C}$ y $\mathbf{e} \in \mathbb{F}_2 G$ el error producido al recibir $\mathbf{r} \in \mathbb{F}_2 G$. Supongamos que $\mathbf{e}$ tiene peso w. Si $g_i \in Supp(\mathbf{e})$, entonces*

$$\gamma_i(\mathbf{e}) \geq v - \mu_{\mathcal{D}}(w-1).$$

Por otro lado, si $g_i \notin Supp(\mathbf{e})$, entonces

$$\gamma_i(\mathbf{e}) \leq \mu_{\mathcal{D}}(w).$$

Demostración. Supongamos que $W = \mathrm{Supp}(\mathbf{e})$. Si $g_i \in W$, entonces

$$\begin{aligned}\gamma_i(\mathbf{e}) = v - \mathrm{wt}\left(\sum_{g_j \in W \setminus \{g_i\}} C_j^{(i)}\right) &\geq v - \sum_{g_j \in W \setminus \{g_i\}} \mathrm{wt}\left(C_j^{(i)}\right) \\ &= v - \sum_{g_j \in W \setminus \{g_i\}} d_{i,j} \geq v - \sum_{j=1}^{w-1} d_j = v - \mu_{\mathcal{D}}(w-1).\end{aligned}$$

Si $g_i \notin W$, entonces

$$\gamma_i(\mathbf{e}) = \mathrm{wt}\left(\sum_{g_j \in W} C_j^{(i)}\right) \leq \sum_{g_j \in W} \mathrm{wt}\left(C_j^{(i)}\right) = \sum_{g_j \in W} d_{i,j} \leq \sum_{j=1}^{w} d_j = \mu_{\mathcal{D}}(w).$$

□

Como consecuencia directa, tenemos el siguiente teorema, el cual establece el criterio que será usado para implementar un algoritmo de descodificación en aquellos códigos grupo LDOI cuya matriz de adyacencia es binaria.

Teorema 5.2.3. *Dado $\mathfrak{C}$ un código grupo $v-$LDOI con matriz de adyacencia binaria y $\mathbf{e} \in \mathbb{F}_2 G$ el error producido al recibir $\mathbf{r} \in \mathbb{F}_2 G$. Si $\mathbf{e} \in \mathbb{F}_2 G$ tiene peso $w \leq t$ y*

$$b = \begin{cases} v/2 + 1 & \text{si } v \text{ es par,} \\ \lceil v/2 \rceil & \text{si } v \text{ es impar,} \end{cases}$$

entonces $g_i \in Supp(\mathbf{e})$ si y solo si $\gamma_i(\mathbf{r}) \geq b$.

Demostración. Si d es la mínima distancia de $\mathfrak{C}$, entonces por la hipótesis y el Teorema 5.1.10, se deduce que la capacidad correctora de $\mathfrak{C}$ es

$$t = \lfloor (d-1)/2 \rfloor = \lfloor v/2 \rfloor.$$

En virtud de la Proposición 5.1.9, $\mu(\ell) = \ell$ para todo $\ell \leq v(v-1)$. Dado que

$$w \leq \lfloor v/2 \rfloor < v \leq v(v-1),$$

la proposición anterior implica que si $g_i \in \mathrm{Supp}(\mathbf{e})$, entonces

$$\gamma_i(\mathbf{e}) \geq v - \mu_{\mathcal{D}}(w-1) = v - (w-1).$$

Mientras que si $g_i \notin \mathrm{Supp}(\mathbf{e})$, entonces

$$\gamma_i(\mathbf{e}) \leq \mu_{\mathcal{D}}(w) = w.$$

Como el mayor número de errores que podemos corregir es $\lfloor v/2 \rfloor$. En particular, para $w = \lfloor v/2 \rfloor$ se cumple que si $g_i \in \mathrm{Supp}(\mathbf{e})$, entonces

$$\gamma_i(\mathbf{e}) \geq v - \mu(w-1) = v - (\lfloor v/2 \rfloor - 1) = v - \lfloor v/2 \rfloor + 1.$$

Mientras que si $g_i \notin \mathrm{Supp}(\mathbf{e})$, entonces

$$\gamma_i(\mathbf{e}) \leq \mu(w) = \lfloor v/2 \rfloor.$$

Supongamos que v es par, entonces $\lfloor v/2 \rfloor = v/2$.

Si $g_i \in \text{Supp}(\mathbf{e})$, tenemos que

$$\gamma_i(\mathbf{r}) = \gamma_i(\mathbf{e}) \geq v/2 + 1.$$

Si $g_i \notin \text{Supp}(\mathbf{e})$, entonces

$$\gamma_i(\mathbf{r}) = \gamma_i(\mathbf{e}) \leq v/2 < v/2 + 1.$$

Si v es impar, la prueba es análoga. □

5.2.1. Algoritmo Bit Flipping con una única iteración (Algoritmo 1-BF)

Supongamos que $\mathfrak{C}$ es un código grupo $v-$LDOI cuya matriz de adyacencia es binaria y que

$$b = \begin{cases} v/2 + 1 & \text{si } v \text{ es par} \\ \lceil v/2 \rceil & \text{si } v \text{ es impar.} \end{cases}$$

El algoritmo de descodificación por *Bit Flipping con una única iteración* se describe de la siguiente forma:

Paso 1.

Recibida la palabra $\mathbf{r} \in \mathbb{F}_2 G$, se calcula $S^+(\mathbf{r})$. Si $S^+(\mathbf{r}) = 0$, no hay errores y el algoritmo termina. En caso contrario, se procede al siguiente paso.

Paso 2.

Se calcula el vector de contadores $\gamma(\mathbf{r}) = (\gamma_1(\mathbf{r}), \ldots, \gamma_n(\mathbf{r}))$. Se considera el conjunto

$$W = \{g_i \in G : \gamma_i(\mathbf{r}) \geq b\}$$

y su cardinal $w = |W|$.

1. Si $w \leq t$ y $W = \{g_{i_1}, \ldots, g_{i_w}\}$, entonces el error es $\mathbf{e} = g_{i_1} + \cdots + g_{i_w}$ y el algoritmo termina.

2. De lo contrario, la palabra $\mathbf{r}$ no puede ser descodificada, puesto que, el número de errores producidos durante la transmisión es mayor a t.

Ejemplo 5.2.4. *Supongamos que* $G = C_{31} = \langle g \rangle$ *y* $\mathbb{K} = \mathbb{F}_2$. *Ordenamos los elementos de* G *mediante* $\{1, g, \ldots, g^{30}\}$. *Si* $\mathfrak{C}$ *es el código grupo que tiene idempotente ortogonal*

$$e_0^+ = (0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0),$$

entonces $\mathfrak{C}$ *es un código grupo* $5-LDOI$ *y tiene parametros* $n = 31$ *y* $k = 15$. *Dado que la matriz de adyacencia de* $\mathfrak{C}$ *es binaria, entonces* $d = 6$, *y por lo tanto, su capacidad correctora es* $t = 2$.

Para ver cómo funciona el Algoritmo 1-BF, notemos que $b = 3$ *y consideremos*

$$\mathbf{r} = (0,1,1,1,0,0,1,0,0,1,1,0,0,0,1,0,0,1,1,1,1,1,0,1,1,0,1,1,1,0,1).$$

Descodificando tenemos:

Paso 1. *El* $\mathbb{F}_2G-$*síndrome de* $\mathbf{r}$ *es*

$$S^+(\mathbf{r}) = (1,0,0,1,1,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,1,0,0,0,0,1,0,0,1,0).$$

Paso 2. *El vector de contadores de* $\mathbf{r}$ *es*

$$\gamma(\mathbf{r}) = (1,1,2,0,2,1,0,1,1,2,2,2,1,1,2,1,0,2,2,1,1,1,1,4,0,1,1,0,4,1,1).$$

Vemos que $W = \{g^{23}, g^{28}\}$ *y así el error es* $\mathbf{e} = g^{23} + g^{28}$.

5.2.2. Análisis de Complejidad

El precálculo de la matriz $\mathcal{C}(g_1, \ldots, g_n)$ tiene orden de complejidad $\mathcal{O}(nv)$. En efecto, para el cálculo de cada columna C_{g_i} hay que hacer v productos, puesto que, e_0^+ tiene peso v.

Como $\mathbf{r}$ puede tener peso n, entonces el cálculo de $S^+(\mathbf{r})$ implica que el paso 1 tiene orden de complejidad $\mathcal{O}(nv)$. En el paso 2, para calcular cada contador $\gamma_i(\mathbf{r})$ hay que hacer v comprobaciones, puesto que, cada columna tiene un peso v. En consecuencia, tenemos orden de complejidad $\mathcal{O}(nv)$.

Ahora bien, por el Teorema 5.1.7, se deduce que $n \geq v^2$. Con lo cual, el orden de complejidad del Algoritmo 1-BF está acotado por $\mathcal{O}(n\sqrt{n})$.

5.3. Códigos grupo LDOI abelianos

Es bien sabido ([9]) que si $G = \langle h_1 \rangle \times \cdots \times \langle h_m \rangle$ es abeliano y $\mathbb{K}$ es un cuerpo finito cuya característica y el orden de G son relativamente primos, entonces existe una correspondencia biyectiva entre los códigos grupo de $\mathbb{K}G$ y las uniones (disjuntas) de clases ciclotómicas de $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m}$. Es decir, toda unión de clases ciclotómicas define un código grupo abeliano y viceversa. Usaremos la noción de clases 2−ciclotómicas para definir códigos grupo abelianos cuyo idempotente ortogonal tiene peso pequeño (LDOI).

Definición 5.3.1. *La clase 2−ciclotómica de* $(z_1, \ldots, z_m) \in \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m}$ *denotada por* $Q(z_1, \ldots, z_m)$*, se define como el conjunto dado por*

$$Q(z_1, \ldots, z_m) := \{(z_1, \ldots, z_m), (2z_1, \ldots, 2z_m), \ldots, (2^{s-1}z_1, \ldots, 2^{s-1}z_m)\},$$

siendo s *es el entero positivo más pequeño tal que* $2^s z_i = z_i \mod n_i$*, para todo* $i \in \{1, \ldots, m\}$*.*

Las clases 2−ciclotómicas determinan una partición de $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m}$ ([9]). Si consideramos un grupo abeliano G y el cuerpo binario $\mathbb{F}_2$, las clases 2−ciclotómicas permiten construir los idempotentes (centrales) de $\mathbb{F}_2 G$. Para ello, notemos que si $f \in \mathbb{F}_2 G$ es un idempotente cuyo soporte es U, entonces

$$f = \sum_{g_j \in U} g_j$$

y, por lo tanto,

$$f = f^2 = \left(\sum_{g_j \in U} g_j \right)^2 = \sum_{g_j \in U} g_j^2.$$

Es decir, si $g_i \in U$, entonces $g_i^2 \in U$.

Proposición 5.3.2. *Sea*

$$G = \langle h_1 \rangle \times \cdots \times \langle h_m \rangle \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_m},$$

donde n_i *es un número impar para todo* $i \in \{1, \ldots, m\}$*. Si* $\mathcal{F}$ *es una unión (disjunta) de clases 2−ciclotómicas de* $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m}$ *y*

$$f = \sum_{Q(z_1, \ldots, z_m) \in \mathcal{F}} h_1^{z_1} \cdots h_m^{z_m},$$

entonces f *es un idempotente de* $\mathbb{F}_2 G$ *y la aplicación* $\mathcal{F} \mapsto f$ *es biyectiva.*

Demostración. En efecto,

$$f^2 = \left(\sum_{Q(z_1,\ldots,z_m)\in\mathcal{F}} h_1^{z_1} \cdots h_m^{z_m} \right)^2 = \sum_{Q(z_1,\ldots,z_m)\in\mathcal{F}} h_1^{2z_1} \cdots h_m^{2z_m}.$$

Como $(2z_1, \ldots, 2z_m)$ pertenece a la clase $2-$ciclotómica de $(z_1, \ldots, z_m)$, entonces $Q(z_1, \ldots, z_m) = Q(2z_1, \ldots, 2z_m)$ para todo $Q(z_1, \ldots, z_m) \in \mathcal{F}$ y la suma anterior es igual a f.

Ahora bién, si $f \in \mathbb{F}_2G$ es un idempotente y

$$\mathcal{F} = \{(z_1, \ldots, z_m) \in \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m} : h_1^{z_1} \cdot \cdots \cdot h_m^{z_m} \in \mathrm{Supp}(f)\}$$

entonces $(2^r z_1, \ldots, 2^r z_m) \in \mathcal{F}$ para todo $(z_1, \ldots, z_m) \in \mathcal{F}$ y $r \in \mathbb{Z}^+$. Lo anterior implica que $\mathcal{F}$ es la unión de clases $2-$ciclotómicas. □

Con la notación del resultado anterior, afirmamos que toda unión de clases $2-$ciclotómicas induce un idempotente de $\mathbb{F}_2G$ y viceversa. La unión vacía de clases $2-$ciclotómicas induce el idempotente $0 \in \mathbb{F}_2G$ y $\{(0, \ldots, 0)\}$ se denomina *la clase* $2-$*ciclotómica nula* e induce el idempotente $1 \in \mathbb{F}_2G$.

Dado $v < n$, queremos saber si es posible construir un código grupo $v-$LDOI de $\mathbb{F}_2G$ (siendo G abeliano). Para ello, es necesario notar que el idempotente $e_0^+ \in \mathbb{F}_2G$ que es ortogonal a $\mathfrak{C}$, está inducido por la unión $\mathcal{F} \neq \emptyset$ de una familia de clases $2-$ciclotómicas cuyo cardinal es $|\mathcal{F}| = v$. Es decir, podemos garantizar la existencia de un código grupo $v-$LDOI si existe una clase $2-$ciclotómica de tamaño menor o (como máximo) igual a v. Si s el menor entero positivo tal que $2^s z_i = z_i \mod n_i$, para todo $i \in \{1, \ldots, m\}$, entonces $v \geq s$. Resumiendo, dado un grupo abeliano G con orden n. Si $v < n$, es posible encontrar un $G-$código $v-$LDOI, cuyo idempotente ortogonal $e_0^+ \in \mathbb{F}_2G$ está inducido por la unión $\mathcal{F} \neq \emptyset$ de una familia de clases $2-$ciclotómicas, cuando $|\mathcal{F}| = v \geq s$.

Nuestro objetivo ahora será hacer algunas construcciones de códigos grupo que cumplan la condición anterior. Para ello, debemos tener en cuenta que el orden de 2 en el grupo multiplicativo $\mathbb{Z}_3^*$ es 2. Con lo cual, no es posible encontrar dos $2-$subconjuntos distintos en $\mathbb{Z}_3^*$.

Definición 5.3.3. *Sean $p > 3$ un número primo y s el orden de 2 en el grupo multiplicativo $\mathbb{Z}_p^*$. Decimos que p satisface la Propiedad (P), si no existen dos pares (x, x') e (y, y') distintos, con $x, x', y, y' \in \{0, 1, \ldots, s-1\}$, $x \neq x'$ e $y \neq y'$, tales que*

$$2^x - 2^{x'} = 2^y - 2^{y'} \mod p.$$

El menor número primo que satisface la Propiedad (P) es 7.

Teorema 5.3.4. *Sean $p \geq 7$ un número primo y*

$$G = \langle h_1 \rangle \times \cdots \times \langle h_m \rangle \cong \mathbb{Z}_p \oplus \cdots \oplus \mathbb{Z}_p = \mathbb{Z}_p^m.$$

Supongamos que $\mathfrak{C}$ es un código grupo de $\mathbb{F}_2 G$ cuyo idempotente ortogonal e_0^+ está inducido por la clase $2-$ciclotómica de $(z_1, \ldots, z_m) \in \mathbb{Z}_p^m$. Si $z_l \neq 0$ para todo $l \in \{1, \ldots, m\}$ y p satisface la Propiedad (P), entonces la matriz de adyacencia de $\mathfrak{C}$ es binaria.

Demostración. Supongamos que s es el orden de 2 en el grupo multiplicativo $\mathbb{Z}_p^*$. Por hipótesis,

$$e_0^+ = \sum_{r=0}^{s-1} h_1^{2^r z_1} \cdots h_m^{2^r z_m}.$$

Si $g = h_1^{i_1} \cdots h_m^{i_m}$ y $g' = h_1^{j_1} \cdots h_m^{j_m}$ son dos elementos distintos de G, entonces

$$ge_0^+ = \sum_{r=0}^{s-1} h_1^{2^r z_1 + i_1} \cdots h_m^{2^r z_m + i_m},$$

$$g'e_0^+ = \sum_{r=0}^{s-1} h_1^{2^r z_1 + j_1} \cdots h_m^{2^r z_m + j_m}$$

y, por lo tanto, existe $\ell \in \{1, \ldots, m\}$ tal que $i_\ell \neq j_\ell$. Si los soportes de ge_0^+ y $g'e_0^+$ se intersecan en una posición, entonces para cada $l \in \{1, \ldots, m\}$, existe un par $x(l), x'(l) \in \{0, 1, \ldots, s-1\}$ tal que

$$2^{x(l)} z_l + i_l = 2^{x'(l)} z_l + j_l \mod p.$$

Para $l = \ell$ denotamos $x = x(\ell)$, $x' = x'(\ell)$, y tenemos que

$$z_\ell (2^{x'} - 2^x) = i_\ell - j_\ell \neq 0 \mod p.$$

Como $z_l \neq 0$ para toda $l \in \{1, \ldots, m\}$, en particular $z_\ell \neq 0$. Esto implica que $x \neq x'$. Si los soportes de ge_0^+ y $g'e_0^+$ se intersecan en una segunda posición, existe un par $(y, y') \neq (x, x')$, con $y, y' \in \{0, 1, \ldots, s-1\}$ distintos, tal que

$$z_\ell (2^{y'} - 2^y) = i_\ell - j_\ell \mod p.$$

Con lo cual,

$$z_\ell (2^{y'} - 2^y) = z_\ell (2^{x'} - 2^x) \mod p.$$

Lo que implica,

$$2^{y'} - 2^y = 2^{x'} - 2^x \mod p. \tag{5.9}$$

Todo lo anterior contradice la Propiedad (P). En consecuencia, los soportes de ge_0^+ y $g'e_0^+$ se intersecan como máximo en una posición y así, la matriz de adyacencia de $\mathfrak{C}$ es binaria. □

A continuación presentamos algunas generalizaciones del teorema anterior.

Teorema 5.3.5. *Sean $p \geq 7$ un número primo y*

$$G = \langle h_1 \rangle \times \cdots \times \langle h_m \rangle \cong \mathbb{Z}_p \oplus \cdots \oplus \mathbb{Z}_p = \mathbb{Z}_p^m.$$

Supongamos que $\mathfrak{C}$ es un código grupo de $\mathbb{F}_2 G$ cuyo idempotente ortogonal e_0^+ está inducido por la unión de dos clases $2-$ciclotómicas distintas cuyos representantes son $z = (z_1, \ldots, z_m)$ y $w = (w_1, \ldots, w_m)$. Si las componentes de z y w son distintas de cero, $Q(z_l) = Q(w_l)$ para cada $l \in \{1, \ldots, m\}$ y p satisface la Propiedad (P), entonces la matriz de adyacencia de $\mathfrak{C}$ es binaria.

Demostración. Supongamos que s es el orden de 2 en el grupo multiplicativo $\mathbb{Z}_p^*$. Por hipótesis,

$$e_0^+ = \left(\sum_{r=0}^{s-1} h_1^{2^r z_1} \cdots h_m^{2^r z_m} \right) + \left(\sum_{r=0}^{s-1} h_1^{2^r w_1} \cdots h_m^{2^r w_m} \right).$$

Si $g = h_1^{i_1} \cdots h_m^{i_m}$ y $g' = h_1^{j_1} \cdots h_m^{j_m}$ son dos elementos distintos de G, entonces

$$ge_0^+ = \left(\sum_{r=0}^{s-1} h_1^{2^r z_1 + i_1} \cdots h_m^{2^r z_m + i_m} \right) + \left(\sum_{r=0}^{s-1} h_1^{2^r w_1 + i_1} \cdots h_m^{2^r w_m + i_m} \right),$$

$$g'e_0^+ = \left(\sum_{r=0}^{s-1} h_1^{2^r z_1 + j_1} \cdots h_m^{2^r z_m + j_m} \right) + \left(\sum_{r=0}^{s-1} h_1^{2^r w_1 + j_1} \cdots h_m^{2^r w_m + j_m} \right),$$

y, por lo tanto, existe $\ell \in \{1, \ldots, m\}$ tal que $i_\ell \neq j_\ell$. Si los soportes de ge_0^+ y $g'e_0^+$ se intersecan en una posición, entonces para cada $l \in \{1, \ldots, m\}$, existe un par $x, x' \in \{0, 1, \ldots, s-1\}$ tal que ocurre uno de los siguientes casos:

1. $2^x z_l + i_l = 2^{x'} z_l + j_l \mod p$.
2. $2^x w_l + i_l = 2^{x'} w_l + j_l \mod p$.
3. $2^x z_l + i_l = 2^{x'} w_l + j_l \mod p$.
4. $2^x w_l + i_l = 2^{x'} z_l + j_l \mod p$.

Sin pérdida de generalidad, vamos a considerar sólo los casos 1 y 3. Como $Q(z_1, \ldots, z_m) \neq Q(w_1, \ldots, w_m)$ y $Q(z_l) = Q(w_l)$ para todo $l \in \{1, \ldots, m\}$, luego existe $r_l \in \{1, \ldots, s-1\}$ tal que $w_l = 2^{r_l} z_l$. Para $l = \ell$, ocurre uno de los siguientes casos:

i. $z_\ell(2^{x'} - 2^x) = i_\ell - j_\ell \neq 0 \mod p$ y así $x \neq x'$.

ii. $z_\ell(2^{x'+r_\ell} - 2^x) = i_\ell - j_\ell \neq 0 \mod p$, y así $x \neq x' + r_\ell$. Además, existe $x'' \in \{0, 1, \ldots, s-1\}$ tal que $2^{x''} = 2^{x'+r_\ell}$. Esto implica que $z_\ell(2^{x''} - 2^x) = i_\ell - j_\ell \neq 0 \mod p$, y por consiguiente, $x \neq x''$.

Si los soportes de ge_0^+ y $g'e_0^+$ se intersecan en una segunda posición, ocurre uno de los siguientes casos:

- Si ocurre i, existe un par $(y, y') \neq (x, x')$, con $y, y' \in \{0, 1, \ldots, s-1\}$ distintos, tal que $z_\ell(2^{y'} - 2^y) = i_\ell - j_\ell \neq 0 \mod p$, y por tanto,

$$2^{y'} - 2^y = 2^{x'} - 2^x \mod p.$$

- Si ocurre ii, existe un par $(y, y'') \neq (x, x'')$, con $y, y'' \in \{0, 1, \ldots, s-1\}$ distintos, tal que $z_\ell(2^{y''} - 2^y) = i_\ell - j_\ell \neq 0 \mod p$, y por tanto,

$$2^{y''} - 2^y = 2^{x''} - 2^x \mod p.$$

En los dos casos, se contradice la Propiedad (P). En consecuencia, los soportes de ge_0^+ y $g'e_0^+$ se intersecan como máximo en una posición. □

Corolario 5.3.6. *Sean $p \geq 7$ un número primo y*

$$G = \langle h_1 \rangle \times \cdots \times \langle h_m \rangle \cong \mathbb{Z}_p \oplus \cdots \oplus \mathbb{Z}_p = \mathbb{Z}_p^m.$$

Supongamos que $\mathfrak{C}$ es un código grupo de $\mathbb{F}_2 G$ cuyo idempotente ortogonal e_0^+ está inducido por la unión de r clases $2-$ciclotómicas distintas cuyos representantes son

$$z^{(1)} = \left(z_1^{(1)}, \ldots, z_m^{(1)}\right), \ldots, z^{(r)} = \left(z_1^{(r)}, \ldots, z_m^{(r)}\right).$$

Si todas las componentes de $z^{(1)}, \ldots, z^{(r)}$ son distintas de cero,

$$Q\left(z_l^{(1)}\right) = \cdots = Q\left(z_l^{(r)}\right)$$

para cada $l \in \{1, \ldots, m\}$, y p satisface la Propiedad (P), entonces la matriz de adyacencia de $\mathfrak{C}$ es binaria.

Demostración. Es análoga a la demostración del Teorema 5.3.5. □

En la hipótesis del corolario anterior, como e_0^+ está inducido por la unión de algunas clases $2-$ciclotómicas distintas y no nulas, entonces $1 \notin \mathrm{Supp}(e_0^+)$. Este hecho junto con las demás hipótesis del corolario implican que la matriz de adyancencia de $\mathfrak{C}$ es binaria y, por lo tanto, tenemos la igualdad en (5.1). Los resultados anteriores se pueden generalizar aún más.

Teorema 5.3.7. *Sean $p \geq 7$ un número primo,*

$$G = \langle h_1 \rangle \times \cdots \times \langle h_m \rangle \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_m},$$

donde $n_1, \ldots, n_m$ son números impares, y $L = \{l \in \{1, \ldots, m\} : n_l = p\}$. Supongamos que $L \neq \emptyset$ y que $\mathfrak{C}$ es un código grupo de $\mathbb{F}_2 G$ cuyo idempotente ortogonal e_0^+ está inducido por la clase $2-$ciclotómica de un elemento no nulo $z = (z_1, \ldots, z_m) \in \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m}$. Si $Supp(z) \subseteq L$ y p satisface la Propiedad (P), entonces la matriz de adyacencia de $\mathfrak{C}$ es binaria.

Demostración. Supongamos que $\mathrm{Supp}(z) = \{l_1, \ldots, l_\eta\}$ y que s es el orden de 2 en el grupo multiplicativo $\mathbb{Z}_p^*$. Por hipótesis,

$$e_0^+ = \sum_{r=0}^{s-1} h_{l_1}^{2^r z_{l_1}} \cdots h_{l_\eta}^{2^r z_{l_\eta}}.$$

Si $g = h_1^{i_1} \cdots h_{l_1}^{i_{l_1}} \cdots h_{l_\eta}^{i_{l_\eta}} \cdots h_m^{i_m}$ y $g' = h_1^{j_1} \cdots h_{l_1}^{j_{l_1}} \cdots h_{l_\eta}^{j_{l_\eta}} \cdots h_m^{j_m}$ son dos elementos distintos de G, entonces

$$g e_0^+ = \sum_{r=0}^{s-1} h_1^{i_1} \cdots h_{l_1}^{2^r z_{l_1} + i_{l_1}} \cdots h_{l_\eta}^{2^r z_{l_\eta} + i_{l_\eta}} \cdots h_m^{i_m},$$

$$g' e_0^+ = \sum_{r=0}^{s-1} h_1^{j_1} \cdots h_{l_1}^{2^r z_{l_1} + j_{l_1}} \cdots h_{l_\eta}^{2^r z_{l_\eta} + j_{l_\eta}} \cdots h_m^{j_m},$$

y, por lo tanto, existe $\ell \in \{1, \ldots, m\}$ tal que $i_\ell \neq j_\ell$. Hay dos casos a considerar:

1. Si $\ell \in \mathrm{Supp}(z)$, entonces los soportes de $g e_0^+$ y $g' e_0^+$ se intersecan como máximo en una posición. La prueba de ello es análoga al Teorema 5.3.4.

2. Si $\ell \notin \mathrm{Supp}(z)$, entonces los soportes de $g e_0^+$ y $g' e_0^+$ no se intersecan en ninguna posición.

Con lo anterior, la matriz de adyacencia de $\mathfrak{C}$ es binaria. □

Corolario 5.3.8. *Sean $p \geq 7$ un número primo,*

$$G = \langle h_1 \rangle \times \cdots \times \langle h_m \rangle \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_m},$$

donde $n_1, \ldots, n_m$ son números impares, y $L = \{l \in \{1, \ldots, m\} : n_l = p\}$. Supongamos que $L \neq \emptyset$ y que $\mathfrak{C}$ es un código grupo de $\mathbb{F}_2 G$ cuyo idempotente ortogonal e_0^+ está inducido por la unión de r clases $2-$ciclotómicas distintas, cuyos representantes son

$$z^{(1)} = \left(z_1^{(1)}, \ldots, z_m^{(1)}\right), \ldots, z^{(r)} = \left(z_1^{(r)}, \ldots, z_m^{(r)}\right).$$

Si

$$Supp\left(z^{(1)}\right) = \cdots = Supp\left(z^{(r)}\right) = L',$$
$$Q\left(z_l^{(1)}\right) = \cdots = Q\left(z_l^{(r)}\right)$$

para cada $l \in L'$, $L' \subseteq L$ *y* $p \geq 7$ *satisface la Propiedad* (P), *entonces la matriz de adyacencia de* $\mathfrak{C}$ *es binaria.*

Demostración. Es análoga a la prueba del Teorema 5.3.5. □

El siguiente ejemplo muestra que las condiciones de los resultados anteriores son suficientes pero no necesarias. Esto deja abierta la posibilidad de encontrar muchos más códigos grupo LDOI con matriz de adyacencia binaria.

Ejemplo 5.3.9. *Supongamos que* $C_7 \times C_7 = \langle x, y\rangle$ *y* $C_3 = \langle z\rangle$. *El grupo* C_3 *actúa (como grupo de automorfismos) sobre* $C_7 \times C_7$ *mediante*

$$zxz^{-1} = x^2 \text{ y } zxz^{-1} = y^2.$$

Sean $G = (C_7 \times C_7) \rtimes C_3$ *y* $\mathfrak{C}$ *el código grupo de* $\mathbb{F}_2 G$ *que es ortogonal a*

$$e_0^+ = x^3 + x^6 + x^5 + x^5y + x^3y^2 + x^6y^4.$$

Entonces $\mathfrak{C}$ *es* $6-LDOI$, *y tiene parámetros* $n = 147$ *y* $k = 75$. *La matriz de adyacencia de* $\mathfrak{C}$ *es binaria. Esto implica que* $d = 7$ *y que* $\mathfrak{C}$ *descodifica todos los errores con peso* $w \leq 3$ *usando el Algoritmo 1-BF con* $b = 4$.

Conclusiones y problemas abiertos

El objetivo de esta tesis es el diseño de algoritmos de descodificación para códigos grupo. Presentamos 8 algoritmos de descodificación para el caso de álgebras de grupo semisimples. La tesis se dividió en cinco capítulos. En el primero, se recuerda las nociones esenciales, tanto de códigos lineales, álgebras de grupo, como de códigos grupo, que se necesitan para el desarrollo de la tesis.

En el capítulo 2, se construye un algoritmo inspirado en el clásico algoritmo de descodificación por síndrome. En este algoritmo (Algoritmo SSD) se utiliza los idempotentes centrales primitivos ortogonales a un $G-$código. El algoritmo se basa en la búsqueda de $t-$subconjuntos de G que contengan las posiciones de error. Se considera también una versión modificada para el caso abeliano que es más eficiente. El estudio de familias de grupos o la exploración de propiedades algebraicas que permitan la detección de posiciones de error con una menor complejidad es una de las cuestiones que se pretenden considerar en un futuro próximo.

En el Capítulo 3, se diseñan dos algoritmos de descodificación en el que solo se usa el idempotente central que es ortogonal al código grupo. El primero de ellos (Algoritmo GMD), generaliza el algoritmo de descodificación de Meggitt y usa un $\mathbb{K}G-$síndrome reducido. De nuevo, el estudio de propiedades algebraicas o clases de grupo que permitan reducir la lista de $\mathbb{K}G-$síndromes es otro de los problemas abiertos para trabajar posteriormente. El segundo algoritmo (Algoritmo ISD), aunque muy similar al algoritmo de descodificación diseñado en el Capítulo 2, representa una mejora del mismo. El Algoritmo GMD requiere muchos cálculos para obtener una lista reducida de $\mathbb{K}G-$síndromes, cálculos que dependen del tamaño del cuerpo. Como eso no ocurre con el Algoritmo ISD, este algoritmo es, en general, más eficiente que el Algoritmo GMD, salvo en el caso de códigos binarios, donde el algoritmo ISD es más adecuado.

En el Capítulo 4, se describe un método que permite encontrar, de manera eficiente, un conjunto de información y una matriz de control que justifican la implementación de los algoritmos de descodificación por permutación (Algoritmo PPD y Algoritmo PD) en códigos grupo. También presentamos alternativas para calcular subconjuntos del grupo de automorfismos permutación del código grupo, que se podrían usar para descodificar parcial o totalmente. Se generaliza el algoritmo de descodificación por permutación (Algoritmo GPD) que permite su implementación en algunos códigos grupo en los que no es posible aplicar el Algoritmo PD. Queda abierta la investigación sobre encontrar conjuntos de información con propiedades que sean útiles para establecer eficientemente PD−conjuntos y GPD−conjuntos en códigos grupo, utilizando la estructura de los diferentes tipos de grupos que sean simples de comprobar.

Finalmente, en el Capítulo 5, se estudian códigos grupo binarios generados por un idempotente ortogonal de peso pequeño. Logramos encontrar propiedades útiles en este tipo de códigos y diseñar un algoritmo de descodificación (Algoritmo 1-BF) para aquellos que tienen matriz de adyacencia binaria. Aunque el Algoritmo 1-BF no se puede aplicar a cualquier código grupo, se muestran explícitamente casos en los que este algoritmo puede ser implementado. Tales construcciones se basan en ciertas condiciones fijadas para el grupo G y el idempotente que es ortogonal al $G-$código. Se obtienen así condiciones que son suficientes pero no necesarias. Se puede plantear, por tanto, la cuestión de obtener una caracterización de todos los códigos grupo LDOI (abelianos y no abelianos) que tiene matriz de adyacencia binaria o determinar otro tipo de códigos grupo que puedan ser descodificados con algoritmos similares (utilizando dos o más iteraciones). Queda también abierto el estudio de códigos grupo LDOI no binarios y el diseño de algoritmos del tipo 1-BF para descodificar en ese caso, ampliando la familia de códigos grupo donde se puede aplicar este método.

Bibliografía

[1] E. Assmus y J. Key, *Polynomial codes and finite geometries,* Handbook of coding theory, vol. II, North Holland, Amsterdam, 1998, págs. 1269-1344.

[2] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal y D. Schipani, *Enhanced public key Security for the McEliece cryptosystem*, Journal of Cryptology, vol. 29, 2016, págs. 1-27.

[3] A. Benyamin-Seeyar, S. Shiva y K. Bhargava, *Capability of the error-trappin technique in decoding cyclic codes*, IEEE Transactions on Information Theory, vol. 32, no. 2, 1986, págs. 166-180.

[4] T. Berger y P. Loidreau, *How to mask the structure of codes for a cryptographic use*, Designs, Codes and Cryptography, vol. 35, 2005, págs. 63-79.

[5] T. Berger, P. Cayrel, P. Gaborit y A. Otmani, *Reducing key length of the McEliece cryptosystem*, Progress in Cryptology – AFRICACRYPT 2009. AFRICACRYPT 2009. Lecture Notes in Computer Science, vol. 5580, Springer, Berlin, Heidelberg, 2009, págs. 77-97.

[6] S. Berman, *On the theory of group codes*, Cybernetics and Systems Analysis, vol. 3, 1967, págs. 25-31.

[7] J.J. Bernal, Códigos de grupo. Conjuntos de información. Decodificación por permutación, Tesis Doctoral, Universidad de Murcia, 2011.

[8] J.J. Bernal, Á del Río y J.J. Simón, *An intrinsical description of group codes*, Designs, Codes and Cryptography, vol. 51, no. 3, 2009, págs. 289-300.

[9] J.J. Bernal y J.J. Simón, *Information sets from defining sets in abelian codes*, IEEE Transactions on Information Theory, vol. 57, no. 12, 2011, págs. 7990-7999.

[10] J.J. Bernal, Á del Río y J.J. Simón, *Partial permutation decoding for abelian codes*, IEEE Transactions on Information Theory vol. 59, no. 8, 2012, págs. 5152-5170.

[11] F. Bernhardt, P. Landrock y O. Manz, *The extended golay codes considered as ideals*, Journal of Combinatorial Theory, Series A, vol. 55, no. 2, 1990, págs. 235–246.

[12] A. Couvreur, P. Gaborit, V. Gauthier, A. Otmani y J. Tillich. *Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes*, Designs, Codes and Cryptography, vol. 73, no. 2, 2014, págs. 641-666.

[13] A. Couvreur, I. Márquez-Corbella y R. Pellikaan, *A polynomial time attack against algebraic geometry code based public key cryptosystem*, IEEE International Symposium on Information Theory, 2014, págs. 1446-1450.

[14] R. Curtis y I. Reiner, Representation Theory of Finite Groups and Associative Algebras, John Wile and Sons, N.Y. London, 1962.

[15] H. Chabanne, *Permutation decoding of abelian codes*, IEEE Transactions On Information Theory, vol. 38, no. 6, 1992, págs.1826-1829.

[16] P. Delsarte, J. Goethals y F. MacWilliams, *On generalized Reed-Muller codes and their relatives*, Infomation and Control, vol. 16, no. 5, 1970, págs. 403-442.

[17] V. Deundyak y Y. Kosolapov, *Algorithms for majority decoding of group codes*, Modeling and Analysis of Information System, vol. 22, no. 4, 2015, págs. 464-482.

[18] M. Elía y C. García, I*deal Group codes and their syndrome decoding*, Procedings of 21st International Symposium on Mathematical Theory of Networks and Systems. Groningen, The Netherlands July, 2014, págs. 7-11.

[19] J. Faugére, L. Perret y F. de Portzamparc, *Algebraic attack against variants of McEliece with Goppa polynomial of a special form*, Advances in Cryptology-ASIACRYPT 2014, vol. 8873, 2014, págs. 21-41.

[20] H. Janwa y O. Moreno, *McEliece public cryptosystem using algebraic-geometric codes*, Design, Codes Cryptography, vol. 8, 1996, págs. 293-307.

[21] R. Gallager, *Low-density parity-check codes*, IRE Transactions on Information Theory, vol. 8, no. 1, 1962, págs. 21-28.

[22] C. García, Códigos grupo no abelianos, Tesis Doctoral, Universidad de Oviedo, 2015.

[23] C. García, S. González, V. Markov, C. Martínez y A. Nechaev, *Group codes which are not abelian group codes*, Proceddings. 3rd International Castle Meeting in Coding Theory Application, Joaquim Borges and Merce Villanueva eds, Universitat Autonoma de Barcelona, Servei de Publicacions, 2011, págs. 123-127.

[24] C. García, S. González, V. Markov, C. Martínez y A. Nechaev, *When all group codes of a non commutative group are abelian (a computational approach)?*, Journal of Mathematical Sciences, vol. 186, 2012, págs. 575-585.

[25] C. García, S. González, V. Markov, C. Martínez y A. Nechaev, *Group codes over non-abelian groups*, Journal of Algebra and Its Applications, vol. 12, no. 7, 2013.

[26] C. García, S. González, V. Markov, C. Martínez y A. Nechaev, *New examples of non-abelian group codes*, Advances in Mathematics of Communications, vol. 10, no. 1, 2016, págs. 1-10.

[27] C. García, S. González, V. Markov y C. Martínez, *Non-abelian group codes over an arbitrary finite field*, Journal of Mathematical Sciences, vol. 223, no. 5, 2017, págs. 504-507.

[28] C. García, S. González, V. Markov, O. Markova y C. Martínez , *Group codes of dimension 2 and 3 are abelian*, Finite Fields and Their Applications, vol. 55, 2019, págs. 167-176.

[29] W. Huffman y V. Pless, Fundamentals of Error-Correcting Codes, Cambridge University Press, 2003.

[30] W. Huffman, *Codes and groups*, Handbook of coding Theory, Volume II, North Holland, Amsterdam, 1998, págs. 1345-1440.

[31] T. Kasami, S. Lin y W. Peterson, *New generalizations of the Reed-Muller codes, Part I: Primitive codes*, IEEE Transactions on Information Theory, vol. 14, no. 2, 1968, págs. 189-199.

[32] T. Kasami, S. Lin y W. Peterson, *Polynomial codes*, IEEE Transactions on Information Theory, vol. 14, no. 6, 1968, págs. 807-814.

[33] A. Kelarev y P. Solé, *Error correcting codes as ideals in group rings*, Contemporary Mathematics, vol. 273, 2001, págs. 11-18.

[34] J. Key, T. McDonough y V. Mavron, *Permutation decoding for codes from finite planes*, European Journal of Combinatorics, vol 26, no. 5, 2005, págs. 665-682.

[35] J. Key, T. McDonough y V. Mavron, *Information sets and partial permutation decoding for codes from finite geometries*, Finite Fields and Their Applications, vol. 12, no. 2, 2006, págs. 232-247.

[36] P. Landbrock y O. Manz, *Classical codes as ideals in group algebras*, Designs, Codes and Cryptography, vol. 2, 1992, págs. 485-505.

[37] Z. Li, C. Xing y S. Ling Yeo. *Reducing the key size of McEliece cryptosystem from automorphism-induced Goppa codes via permutations*, IACR International Workshop on Public Key Cryptography, PKC 2019: Public-Key Cryptography - PKC 2019, vol. 11443, 2019, págs. 599-617.

[38] D. MacKay, *Good error-correcting codes based on very sparse matrices*, IEEE Transactions on Information Theory, vol. 45, no. 2, 1999, págs. 399-431.

[39] F. MacWilliams, *Permutation decoding of systematic codes*, The Bell System Technical Journal, no. 43, no. 1, 1963, págs. 485-505.

[40] F. MacWilliams y N. J. A. Sloane, The Theory of Error-Correcting Codes, New York: Elsevier/North-Holland, 1977.

[41] C. Martínez y F. Molina, *The syndromes decoding algorithm in group codes*, Finite Fields and Their Applications, vol. 89, 2023.

[42] C. Martínez y F. Molina, *Two decoding algorithms in group codes*, Libro de Actas XVII Reunión Española sobre Criptología y Seguridad de la Información, Ediciones Universidad de Cantabria, Santander-Spain, 2022, págs. 157-161.

[43] C. Martínez, F. Molina y A. Piñera-Nicolás, *Decoding algorithms in group codes*. Sometido en 2023.

[44] C. Martínez y F. Molina, *Permutation decoding in group codes*. Sometido en 2024.

[45] J. McEliece, *A public-key cryptosystem based on algebraic coding theory*, DSN Progress Report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, 1978, págs. 114-116.

[46] J. Meggitt, *Error-correcting codes for correcting bursts of errors*, IBM Journal of Research and Development, vol. 4, no. 3, 1960, págs. 329-334.

[47] J. Meggitt, *Error-correcting codes and their implementation for Data Transmission Systems*, IRE Transactions on Information Theory, vol. 7, no. 4, 1961, págs. 459-470.

[48] L. Minder y A. Shokrollah, *Cryptanalysis of the Sidelnikov cryptosystem*, Naor M. (eds) Advances in Cryptology - EUROCRYPT 2007. EUROCRYPT 2007. Lecture Notes in Computer Science, vol. 4515, Springer, Berlin, Heidelberg, 2007.

[49] E. Muller, *Application of boolean algebra to switching circuit design and to error detection*, IEEE Transactions Computers, vol. EC-3, no. 3, 1954, págs. 6-12.

[50] H. Niederreiter, *Knapsack-type cryptosystems and algebraic coding theory*, Problems of Control and Information Theory, vol. 15, no. 2, 1986, págs. 157-166.

[51] E. Persichetti, *Compact McEliece keys based on quasi-dyadic Srivastava codes*, Journal of Mathematical Cryptology, vol. 6, no. 2, 2012, págs. 149-169.

[52] W. Peterson, Error-Correcting Codes, Cambridge, MA: MIT Press, 1961.

[53] E. Prange, *Cyclic error-correcting codes in two symbols*, Technical notes issued by Air Force Cambridge Research Labs, TN-57-103, 1957.

[54] E. Prange, *Some cyclic error-correcting codes with simple decoding algorithms*, Technical notes issued by Air Force Cambridge Research Labs, TN-58-156, 1958.

[55] I. Reed y G. Solomon, *Polynomial codes over certain finite fields*, Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, 1960, págs. 300–304.

[56] P. Santini, M. Battaglioni, M. Baldi y F. Chiaraluce, *Hard-decision iterative decoding of LDPC codes with bounded error rate*, ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, págs. 1-6.

[57] P. Santini, M. Battaglioni, M. Baldi y F. Chiaraluce, *Analysis of the error correction capability of LDPC and MDPC codes under parallel Bit-Flipping decoding and application to cryptography*, IEEE Transactions on Communications, vol. 68, no. 8, 2020 págs. 4648-4660.

[58] S. Shestakov y V. Sidelnikov, *On insecurity of cryptosystems based on generalized Reed-Solomon codes*, Discrete Mathematics and Applications, vol. 2, no. 4, 1992, págs. 439-444.

[59] V. Sidelnikov, *A public-key cryptosystem based on binary Reed-Muller codes*, Discrete Mathematics and Applications, vol. 4, no. 3, 1994.

[60] R. Tanner, *A recursive approach to low complexity codes*, IEEE Transactions on Information Theory, vol. 27, no. 5, 1981, págs. 533-547.

[61] H. Ward, *Quadratic residue codes and divisibility*, Handbook of coding Theory, Volume II, North Holland, Amsterdam, 1998, págs. 827-870.

[62] S. Wicker, y V. Bhargava, *Reed-Solomon Codes and Their Applications*, IEEE Press: Piscataway, 1994.

[63] C. Wieschebrink, *Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes*, Post-Quantum Cryptography, Lecture Notes in Computer Science, vol. 6061, Springer, Berlin, 2010, págs. 61-72.

[64] C. Wieschebrink, *An attack on the modified Niederreiter encryption scheme*, Proceedings of PKC 2006, Lecture Notes in Computer Science, vol. 3958, Springer, Berlin, 2006, págs. 14-26.

[65] http://www.gap-system.org