Códigos Reed-Solomon torcidos para criptografía poscuántica

Kevin Yuseph Herrera Garcia

Trabajo de grado para optar al título de

Matemático

Director

Alexander Holguín Villa

Doctor en Ciencias Matemáticas

Universidad Industrial de Santander
Facultad de Ciencias
Escuela de Matemáticas
Matemáticas
Bucaramanga
2026

Dedicatoria

Dedico este logro a mis padres, por su amor, esfuerzo y por ser el pilar fundamental de todo lo que soy hoy.
A mi hermano, por su compañía y por ser parte importante de mi vida y de este camino.
A mis abuelos, por su cariño, sus enseñanzas y por ser una inspiración constante.
Y a María Paula, porque gracias a ella este trabajo fue posible.

Agradecimientos

Este logro no habría sido posible sin la guía de Dios, quien me acompañó en cada etapa del camino, dándome fuerza en los momentos difíciles y claridad para seguir adelante cuando más lo necesitaba.

A mi familia, gracias por su apoyo incondicional, por creer en mí incluso cuando yo mismo dudaba, y por ser ese motor constante que me impulsó a no rendirme y a seguir construyendo este sueño.

También quiero reconocerme a mí mismo por la disciplina, la constancia y la resiliencia para no abandonar este proceso, incluso en los momentos más complejos. Este camino no fue fácil, pero cada esfuerzo hoy cobra sentido.

Durante toda la carrera, tuve la fortuna de contar con grandes amigos. En especial, gracias a Ronald, Keren y Edwin, por su compañía, su apoyo y por estar presentes en cada etapa, haciendo de este proceso algo mucho más llevadero y significativo.

A todos los profesores que hicieron parte de mi formación, gracias por compartir su conocimiento y por enseñarme a ver más allá de los números, a comprender y admirar la verdadera belleza de las matemáticas.

Un agradecimiento muy especial al profesor Wilson Olaya y al profesor Alexander Holguín, por su guía, su disposición y su apoyo constante, fundamentales para hacer posible el desarrollo de este trabajo de grado.

A mis compañeros, con quienes compartí clases, exámenes y proyectos, pero especialmente a aquellos que comenzaron este camino conmigo y nunca se rindieron. Con ustedes inicié este proceso, crecí y superé muchos retos; por eso les tendré siempre un aprecio y cariño inmenso.

Y finalmente, a María Paula, de no ser por ti esto no hubiera sido posible.

Índice general

RESUMEN

TÍTULO: Códigos Reed-Solomon torcidos para criptografía poscuántica[*]

AUTOR: Kevin Yuseph Herrera Garcia[**]

PALABRAS CLAVE: Teoría de la codificación, códigos Reed-Solomon torcidos, criptografía poscuántica, criptosistema McEliece, corrección de errores, propiedad MDS.

DESCRIPCIÓN:

La teoría de códigos correctores de errores y la criptografía comparten una misma raíz: la necesidad de transmitir y proteger información de manera confiable incluso en presencia de perturbaciones o adversarios. Desde su aparición a mediados del siglo XX, los códigos Reed-Solomon se consolidaron como una de las familias más poderosas dentro de este campo, gracias a su estructura algebraica sobre cuerpos finitos y a su capacidad de alcanzar la cota de Singleton, condición que los convierte en códigos MDS.

En este trabajo se estudia una generalización reciente de dicha familia: los códigos Reed-Solomon torcidos, introducidos por Beelen, Puchinger y Rosenkilde en 2017. La construcción incorpora ℓ términos de torsión, parametrizados por vectores t, h y η sobre el cuerpo base $\mathbb{F}_q$, que modifican la estructura clásica de Vandermonde sin sacrificar, en principio, la distancia mínima óptima. Se presenta de forma detallada la definición de estos códigos, sus matrices generadoras y las condiciones algebraicas precisas bajo las cuales conservan la propiedad MDS, incluyendo dos construcciones explícitas de conjuntos $\mathbb{F}_{q_0}$-libres de suma-producto.

La parte central del trabajo aborda el problema de la decodificación. Se analiza el algoritmo de fuerza bruta, que explora las q^ℓ combinaciones posibles de los coeficientes de torsión, aplica un decodificador Reed-Solomon clásico en cada iteración y emplea una verificación de consistencia para descartar soluciones espurias. Se prueba formalmente la corrección y unicidad del algoritmo bajo la hipótesis MDS.

Finalmente, se explora la aplicación de estos códigos al criptosistema McEliece. Se describe la construcción del esquema McEliece-TRS, su proceso de cifrado y descifrado, y se discuten las consideraciones de seguridad que lo distinguen de la versión original basada en códigos de Goppa, en particular la mayor dificultad estructural que presenta frente al ataque de Sidelnikov-Shestakov. El análisis se complementa con implementaciones en SageMath que permiten comparar la eficiencia computacional entre el esquema clásico y las variantes torcidas con $\ell = 1$ y $\ell = 2$ torsiónes, poniendo de manifiesto el costo exponencial en ℓ y las ventajas que ofrece la flexibilidad estructural adicional de esta familia.

[*]Trabajo de Grado.

[**]Facultad de Ciencias. Escuela de Matemáticas. Director: Alexander Holguín Villa, Doctor en Ciencias Matemáticas.

ABSTRACT

TITLE: Twisted Reed-Solomon Codes for Post-Quantum Cryptography[*]

AUTHOR: Kevin Yuseph Herrera Garcia[**]

KEYWORDS: Coding Theory, Twisted Reed-Solomon Codes, Post-Quantum Cryptography, McEliece Cryptosystem, Error Correction, MDS Property.

DESCRIPTION:

Coding theory and cryptography share a common foundation: the need to transmit and protect information reliably even in the presence of noise or adversaries. Since their introduction in the mid-twentieth century, Reed-Solomon codes have become one of the most powerful families in this field due to their algebraic structure over finite fields and their ability to attain the Singleton bound, which makes them MDS codes.

This work studies a recent generalization of this family: twisted Reed-Solomon codes, introduced by Beelen, Puchinger, and Rosenkilde in 2017. The construction incorporates ℓ twist terms, parameterized by the vectors t, h, and η over the base field $\mathbb{F}_q$, which modify the classical Vandermonde structure without, in principle, sacrificing the optimal minimum distance. A detailed presentation is given of the definition of these codes, their generator matrices, and the precise algebraic conditions under which they preserve the MDS property, including two explicit constructions of $\mathbb{F}_{q_0}$-sum-product-free sets.

The central part of this work addresses the decoding problem. The brute-force algorithm is analyzed, exploring the q^ℓ possible combinations of the twist coefficients, applying a classical Reed-Solomon decoder at each iteration, and using a consistency check to discard spurious solutions. The correctness and uniqueness of the algorithm are formally proved under the MDS hypothesis.

Finally, the application of these codes to the McEliece cryptosystem is explored. The construction of the McEliece-TRS scheme, together with its encryption and decryption processes, is described, and the security considerations that distinguish it from the original version based on Goppa codes are discussed, particularly the greater structural difficulty it presents against the Sidelnikov-Shestakov attack. The analysis is complemented by SageMath implementations that allow the computational efficiency of the classical scheme and the twisted variants with $\ell = 1$ and $\ell = 2$ twist to be compared, highlighting the exponential cost in ℓ and the advantages offered by the additional structural flexibility of this family.

[*]Bachelor Thesis.

[**]Faculty of Sciences. School of Mathematics. Advisor: Alexander Holguín Villa, Doctor in Mathematical Sciences.

Introducción

La teoría de códigos correctores de errores constituye uno de los pilares fundamentales de la transmisión y almacenamiento confiable de información. Desde mediados del siglo XX, el desarrollo de familias de códigos con sólidas propiedades algebraicas ha permitido enfrentar los retos asociados a la detección y corrección de errores en diversos contextos tecnológicos, que van desde las telecomunicaciones hasta la seguridad informática. Entre estos desarrollos, los códigos Reed–Solomon han ocupado un lugar preponderante, gracias a su estructura algebraica sobre cuerpos finitos y a su aplicación exitosa en múltiples sistemas de comunicaciones digitales.

En los últimos años, la búsqueda de nuevas variantes y extensiones de los códigos clásicos ha motivado la construcción de alternativas que, además de preservar propiedades fundamentales, ofrezcan mayor flexibilidad y respondan a los retos actuales. Dentro de este panorama, los códigos Reed–Solomon torcidos (RS-torcidos) aparecen como una generalización de los códigos Reed–Solomon tradicionales, al introducir modificaciones algebraicas en la estructura de los polinomios utilizados para su construcción. Esta modificación permite obtener nuevas familias de códigos que, bajo determinadas condiciones, conservan propiedades óptimas como la propiedad MDS.

El estudio de los códigos Reed–Solomon torcidos (TRS) resulta particularmente relevante por dos razones principales. En primer lugar, permite examinar cómo modificaciones en la construcción clásica generan nuevas estructuras algebraicas y cómo estas afectan parámetros esenciales como la distancia mínima, la capacidad de corrección de errores y la complejidad computacional. En segundo lugar, estas construcciones permiten explorar su posible aplicación en áreas estratégicas como la criptografía y la seguridad de la información, donde las propiedades algebraicas de los códigos desempeñan un papel fundamental.

Este trabajo de grado tiene como propósito estudiar y analizar la construcción de los códigos Reed–Solomon torcidos, así como explorar su aplicación en un contexto criptográfico. En particular, se estudian las propiedades algebraicas de esta familia de códigos, tomando como referencia principal la construcción presentada por Beelen, Puchinger y Rosenkilde [1], y posteriormente se analiza su incorporación al criptosistema de McEliece, con el objetivo de estudiar el comportamiento de esta construcción cuando se utiliza como código subyacente en un esquema criptográfico.

Para alcanzar este objetivo, la investigación se estructura en varios ejes. En primer lugar, se presenta una revisión de los fundamentos de la teoría de códigos correctores de errores y de los códigos Reed–Solomon. Posteriormente, se estudian y sistematizan las principales propiedades algebraicas de los códigos Reed–Solomon torcidos, incluyendo su estructura, sus matrices generadoras y las condiciones bajo las cuales pueden alcanzar la propiedad MDS. Finalmente, se desarrolla una aplicación de estas construcciones al criptosistema de McEliece y se estudian, mediante implementaciones en SageMath, los aspectos computacionales asociados a la codificación, decodificación y funcionamiento del

esquema.

Una de las aplicaciones criptográficas consideradas en este trabajo es el criptosistema de McEliece, un esquema de clave pública basado en códigos correctores de errores y uno de los antecedentes relevantes de la criptografía poscuántica. En particular, se estudia la construcción de una variante basada en códigos Reed–Solomon torcidos, denominada McEliece-TRS, y se analizan sus procesos de generación de claves, cifrado y descifrado. De esta manera, el trabajo conecta el estudio algebraico de los códigos Reed–Solomon torcidos con un problema de interés en criptografía poscuántica, explorando las posibilidades y limitaciones de emplear esta familia de códigos dentro de un esquema de McEliece.

Es importante señalar que el estudio de las propiedades algebraicas de los códigos Reed–Solomon torcidos presentado en este trabajo se fundamenta principalmente en los resultados y construcciones desarrollados en la literatura especializada, en particular en el trabajo de Beelen, Puchinger y Rosenkilde [1]. Por tanto, las definiciones, construcciones y resultados teóricos correspondientes a esta familia de códigos se presentan como parte del estudio y sistematización de la literatura existente. Sobre esta base, los desarrollos propios del presente trabajo corresponden a la adaptación de los códigos Reed–Solomon torcidos al criptosistema de McEliece, al desarrollo de un procedimiento de decodificación para los códigos utilizados en dicha construcción y a la implementación de estos procedimientos en SageMath.

Asimismo, se realiza un estudio experimental del costo computacional asociado al uso de códigos Reed–Solomon clásicos y códigos Reed–Solomon torcidos dentro del esquema de McEliece. Para ello, se consideran implementaciones que permiten comparar el comportamiento computacional de ambas construcciones y analizar el efecto que tiene el número de torsiones sobre el proceso de decodificación y sobre el costo total del esquema. Estos resultados experimentales constituyen una parte propia del trabajo y buscan complementar el estudio teórico mediante evidencia computacional sobre el comportamiento de las construcciones consideradas.

En este sentido, el trabajo no pretende presentar como nuevos los resultados algebraicos establecidos previamente para los códigos Reed–Solomon torcidos, sino estudiarlos, organizarlos y utilizarlos como fundamento para el desarrollo de la aplicación criptográfica propuesta. La contribución principal se concentra, por tanto, en la integración de estas construcciones con el criptosistema de McEliece, en el procedimiento de decodificación empleado y en el análisis experimental de su comportamiento computacional.

Así, la presente investigación busca contribuir a la comprensión de los códigos Reed–Solomon torcidos desde una perspectiva tanto algebraica como computacional, explorando su utilización en un esquema de McEliece y analizando las ventajas, limitaciones y costos asociados a esta construcción. De esta manera, se establece un vínculo entre la teoría de códigos, la decodificación y el estudio de alternativas para la criptografía poscuántica.

Capítulo 1

Preliminares

1.1. Anillos

En esta sección se presentan las nociones fundamentales que servirán de base para el estudio de los códigos Reed–Solomon torcidos (RS-torcidos).

Definición 1.1. Sea R un anillo conmutativo con identidad distinto del anillo nulo. Decimos que R es un ***dominio de integridad*** si no posee ***divisores de cero***, es decir, si para cualesquiera $a, b \in R$ se cumple que

$$ab = 0 \iff a = 0 \text{ o } b = 0.$$

Definición 1.2. Sea R un anillo conmutativo con identidad. Un subconjunto $I \subseteq R$ se denomina ***ideal*** de R si cumple:
(1) $(I, +)$ es un subgrupo aditivo de $(R, +)$.
(2) Para todo $a \in R$ y $x \in I$, se tiene $ax \in I$.
Es decir, I es cerrado bajo la suma y la multiplicación por elementos de R.

Definición 1.3. Sea R un anillo conmutativo con identidad. Un ***ideal principal*** de R es un ideal que puede generarse por un solo elemento, es decir, existe $a \in R$ tal que

$$I = \langle a \rangle = \{\, ra \ : \ r \in R \,\}.$$

El elemento a se denomina ***generador*** del ideal I. Si R es un dominio de integridad y todo ideal de R es principal, se dice que R es un ***dominio de ideales principales (DIP)***.

Definición 1.4. Sea R un anillo conmutativo con identidad. Un ideal $I \subset R$ se llama ***maximal*** si $I \neq R$ y no existe ningún ideal M tal que

$$I \subsetneq M \subsetneq R.$$

Definición 1.5. Sea R un anillo y $I \subset R$ un ideal. El ***anillo cociente*** R/I se define como el conjunto de clases de equivalencia

$$R/I = \{\, a + I \ : \ a \in R \,\},$$

donde $a \sim b$ si y solo si $a - b \in I$. Las operaciones se definen mediante

$$(a + I) + (b + I) = (a + b) + I, \qquad (a + I)(b + I) = (ab) + I.$$

Teorema 1.6. *Sea R un anillo conmutativo con identidad y sea A un ideal de R. Entonces el cociente R/A es un* **cuerpo** *si y solo si A es un* **ideal maximal** *de R.*

Demostración. ($\Rightarrow$) Supongamos que R/A es un cuerpo y sea B un ideal de R tal que $A \subsetneq B \subseteq R$. Sea $b \in B$ con $b \notin A$. Entonces $b + A$ es un elemento no nulo de R/A. Dado que R/A es un cuerpo, existe $c + A \in R/A$ tal que

$$(b + A)(c + A) = 1 + A.$$

Como $b, c \in R$ y $b \in B$, se tiene $bc \in B$. De la igualdad anterior se deduce

$$1 + A = bc + A,$$

por lo que $1 - bc \in A \subset B$. Así, $1 = (1 - bc) + bc \in B$, y como todo ideal que contiene al 1 es el anillo completo, se sigue que $B = R$. Por tanto, A es un ideal maximal.

($\Leftarrow$) Supongamos ahora que A es un ideal maximal y sea $b \in R$ tal que $b \notin A$. Debemos probar que $b + A$ tiene inverso multiplicativo en R/A.

Consideremos el conjunto

$$B = \{\, br + a \ : \ r \in R,\, a \in A \,\}.$$

Se verifica que B es un ideal de R que contiene propiamente a A. Por la maximalidad de A, se sigue que $B = R$. En consecuencia, existe $r \in R$ y $a' \in A$ tales que

$$1 = br + a'.$$

De esta igualdad resulta que

$$1 + A = br + A = (b + A)(r + A),$$

por lo que $(r + A)$ es el inverso multiplicativo de $(b + A)$ en R/A. Esto demuestra que R/A es un cuerpo. □

Definición 1.7. Un *cuerpo finito* es un cuerpo cuyo conjunto subyacente es finito.

Si $\mathbb{F}$ es un cuerpo finito de orden q, se dice que $\mathbb{F}$ es un *cuerpo finito de orden q*. En particular, su orden es de la forma

$$q = p^m,$$

donde p es un número primo llamado la *característica* de $\mathbb{F}$ y $m \in \mathbb{N}$.

Además, para cada $q = p^m$ existe, salvo isomorfismo, un único cuerpo finito de orden q, que se denota por $\mathbb{F}_q$.

.

1.2. Anillo de polinomios sobre un cuerpo finito

Los anillos de polinomios definidos sobre cuerpos finitos constituyen una herramienta algebraica fundamental en matemáticas y en la teoría de la codificación. En el anillo $\mathbb{F}_q[x]$ se formalizan rigurosamente las operaciones con polinomios cuyos coeficientes pertenecen a un cuerpo finito, proporcionando un marco sólido para el análisis estructural y computacional. La aritmética de polinomios en este contexto resulta esencial en el desarrollo de esquemas criptográficos y códigos correctores de errores, donde la eficiencia y la consistencia algebraica desempeñan un papel central.

Definición 1.8. Sea F un cuerpo. El *anillo de polinomios* sobre F se define como

$$F[X] = \left\{ f(X) = \sum_{i=0}^{n} a_i X^i \ : \ a_i \in F,\ n \in \mathbb{N} \right\}.$$

Este conjunto, provisto de la suma y la multiplicación usuales de polinomios, forma un anillo conmutativo con identidad.

Definición 1.9. Sea $f(X) = a_0 + a_1X + a_2X^2 + \cdots + a_nX^n \in F[X]$, donde $a_i \in F$ para todo i y $a_n \neq 0$. Se define el *grado* de $f(X)$, denotado por $\deg(f)$, como el exponente más alto de X que posee coeficiente no nulo, es decir,

$$\deg(f) = n.$$

Proposición 1.10. *Para todo cuerpo F, el anillo de polinomios $F[X]$ es un* **dominio de integridad**.

Demostración. Sean $0 \neq f(X), g(X) \in F[X]$. Escribamos $\deg f = m$ y $\deg g = n$, con coeficientes líderes no nulos $a_m, b_n \in F$.

Entonces el coeficiente líder de $f(X)g(X)$ es a_mb_n. Como F es un cuerpo y $a_m, b_n \neq 0$, se tiene $a_mb_n \neq 0$. Por tanto,

$$\deg\Big(fg\Big) = m + n \geq 0,$$

lo que implica que $fg \neq 0$.

Así, $F[X]$ no tiene divisores de cero y es un dominio de integridad. □

Definición 1.11. Sea F un cuerpo y sea $f(X) \in F[X]$ un polinomio no constante, es decir, de grado $\deg(f) \geq 1$. Decimos que $f(X)$ es ***reducible*** sobre F si existen polinomios no constantes $g(X), h(X) \in F[X]$ tales que

$$f(X) = g(X)\, h(X).$$

En caso contrario, es decir, si $f(X)$ no puede expresarse como producto de dos polinomios no constantes en $F[X]$, se dice que $f(X)$ es ***irreducible*** sobre F.

Definición 1.12. Sea $\mathbb{F}_q$ un cuerpo finito. Un ***ideal principal*** en el anillo de polinomios $\mathbb{F}_q[X]$ es un subconjunto del tipo

$$\langle f(X)\rangle = \{\, f(X)\, g(X) \;:\; g(X) \in \mathbb{F}_q[X] \,\},$$

donde $f(X) \in \mathbb{F}_q[X]$ es un polinomio fijo llamado *generador* del ideal.

Teorema 1.13. *Sea F un cuerpo y sea $p(X) \in F[X]$. Entonces $\langle p(X)\rangle$ es un* **ideal maximal** *de $F[X]$ si y solo si $p(X)$ es* **irreducible** *sobre F.*

Demostración. ($\Rightarrow$) Si $\langle p(X)\rangle$ es maximal en $F[X]$, cualquier factorización no trivial $p(X) = g(X)h(X)$ daría

$$\langle p(X)\rangle \subseteq \langle g(X)\rangle \subseteq F[X],$$

luego, por maximalidad, $\langle g(X)\rangle = \langle p(X)\rangle$ (con h constante) o $\langle g(X)\rangle = F[X]$ (con g unidad). En ambos casos, la factorización es trivial; así p es ***irreducible***.

($\Leftarrow$) Si p es ***irreducible*** y I es un ideal con $\langle p\rangle \subseteq I \subseteq F[X]$, como $F[X]$ es DIP, $I = \langle g\rangle$ para algún g. De $p \in I$ se sigue $p = gh$; la irreducibilidad fuerza que g o h sea constante, de modo que $I = \langle p\rangle$ o $I = F[X]$. Por tanto, $\langle p\rangle$ es maximal. □

Definición 1.14. Sea $\mathbb{F}_q$ un cuerpo finito y sea $f(X) \in \mathbb{F}_q[X]$ un polinomio no nulo.

Se denomina ***anillo cociente*** de $\mathbb{F}_q[X]$ módulo el ideal generado por $f(X)$ al conjunto de clases de equivalencia

$$\mathbb{F}_q[X]/\langle f(X)\rangle = \{\, g(X) + \langle f(X)\rangle \;:\; g(X) \in \mathbb{F}_q[X] \,\},$$

donde dos polinomios $g(X), h(X) \in \mathbb{F}_q[X]$ se consideran equivalentes si su diferencia es múltiplo de $f(X)$, es decir,

$$g(X) \equiv h(X) \pmod{f(X)} \quad \Longleftrightarrow \quad f(X) \mid \Big(g(X) - h(X)\Big).$$

Las operaciones de suma y multiplicación en $\mathbb{F}_q[X]/\langle f(X)\rangle$ se definen de manera natural:

$$(g + \langle f\rangle) + (h + \langle f\rangle) = (g + h) + \langle f\rangle,$$
$$(g + \langle f\rangle)(h + \langle f\rangle) = (gh) + \langle f\rangle.$$

Teorema 1.15. *Sea $\mathbb{F}_q$ un cuerpo finito y $0 \neq f(X) \in \mathbb{F}_q[X]$. Entonces el cociente*

$$\mathbb{F}_q[X]/\langle f(X)\rangle$$

es un **cuerpo** *si y solo si $f(X)$ es* **irreducible** *sobre $\mathbb{F}_q$.*

Demostración. El resultado se deduce directamente como una aplicación del Teorema 1.13, que caracteriza los ideales maximales de $\mathbb{F}_q[X]$ mediante la irreducibilidad de su generador, junto con el Teorema 1.6, el cual establece que el cociente R/I es un cuerpo si y solo si I es un ideal maximal de R.

Capítulo 2

Códigos Lineales y Códigos Reed–Solomon

2.1. Códigos Lineales

Definición 2.1. Sea $\mathbb{F}_q$ un cuerpo finito. Un *código lineal* de longitud n sobre $\mathbb{F}_q$ es un subespacio vectorial $C \subseteq \mathbb{F}_q^n$. La dimensión de C se denota por $k = \dim(C)$. Los elementos de C son llamados *palabras código*.

Definición 2.2. Sean $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$. La *distancia de Hamming* entre x e y se define como

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

Ejemplo 2.3. Considere dos palabras código $x, y \in \mathbb{F}_2^7$:

$$x = 1\ 0\ 1\ 0\ 1\ 1\ 0$$

$$y = 1\ 1\ 1\ 0\ 0\ 1\ 1$$

Donde los símbolos en azul coinciden y los en rojo difieren. Contando las diferencias: $d(x, y) = 3$.

Definición 2.4. Sea $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_q^n$ un vector. El *peso de Hamming* de x, denotado por $w(x)$, se define como el número de coordenadas no nulas de x. Es decir,

$$w(x) = |\{\, i \in \{1, 2, \ldots, n\} : x_i \neq 0 \,\}|\,.$$

Proposición 2.5 (Relación entre distancia y peso)**.** *Para cualquier código lineal $C \subseteq \mathbb{F}_q^n$ y para todo $x, y \in C$, se cumple que*

$$d(x, y) = w(x - y).$$

En particular, para un código lineal, la distancia mínima d coincide con el peso mínimo no nulo de sus palabras código. Es decir,

$$d = \min\{w(c) : c \in C,\ c \neq 0\}.$$

Demostración. Sea $x, y \in C$. Por definición de distancia de Hamming:

$$d(x, y) = |\{i : x_i \neq y_i\}|.$$

Observemos que $x_i \neq y_i$ si y solo si $x_i - y_i \neq 0$. Por lo tanto:

$$d(x, y) = |\{i : x_i - y_i \neq 0\}| = w(x - y).$$

Dado que C es lineal, $x - y \in C$. Así, la distancia mínima entre palabras distintas de C es el mínimo peso entre todos los vectores no nulos de C. □

Definición 2.6. Un $[n, k, d]$-*código* sobre $\mathbb{F}_q$ es un código lineal $C \subseteq \mathbb{F}_q^n$ de dimensión k, con distancia mínima

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

Aquí, n representa la longitud de cada palabra código, k la dimensión del código y d su distancia mínima.

Proposición 2.7. *Sea C un código lineal con distancia mínima d. Entonces:*
- *C puede detectar hasta $d - 1$ errores en una palabra recibida.*
- *C puede corregir hasta*

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

errores.

Observación 2.8. Como consecuencia del resultado anterior, observamos que la capacidad de detección y corrección de errores de un código está determinada por su **distancia mínima** d. Cuanto mayor sea d, mayor será la capacidad del código para detectar y corregir errores.

Además, la distancia mínima no puede ser arbitrariamente grande, pues está sujeta a un límite superior dado por la **cota de Singleton**, la cual establece que para un código lineal C de longitud n y dimensión k se cumple

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

Los códigos que alcanzan la igualdad en esta cota, es decir, aquellos para los cuales

$$d = n - k + 1,$$

se denominan códigos MDS (*Maximum Distance Separable*) o de máxima distancia separable. Estos códigos alcanzan la mayor distancia posible para una longitud y dimensión dadas.

Proposición 2.9. *El código Reed–Solomon $\mathrm{RS}_{n,k}(\boldsymbol{\alpha})$ alcanza la cota de Singleton. En particular, es un código MDS y su distancia mínima satisface*

$$d = n - k + 1.$$

Demostración. Sean $\mathbf{c}_1, \mathbf{c}_2 \in \mathrm{RS}_{n,k}(\boldsymbol{\alpha})$ dos palabras código distintas. Entonces existen polinomios distintos $f(X), g(X) \in \mathbb{F}_q[X]$, con

$$\deg(f), \deg(g) < k,$$

tales que

$$\mathbf{c}_1 = \Big(f(\alpha_1), \ldots, f(\alpha_n)\Big), \qquad \mathbf{c}_2 = \Big(g(\alpha_1), \ldots, g(\alpha_n)\Big).$$

Consideremos el polinomio

$$h(X) = f(X) - g(X).$$

Como $f \neq g$, se tiene $h \neq 0$, y además

$$\deg(h) \leq k - 1.$$

Un polinomio no nulo de grado a lo sumo $k-1$ sobre un cuerpo puede tener a lo sumo $k-1$ raíces. Por tanto, las palabras $\mathbf{c}_1$ y $\mathbf{c}_2$ pueden coincidir en a lo sumo $k-1$ posiciones.

En consecuencia,

$$d(\mathbf{c}_1, \mathbf{c}_2) \geq n - (k-1) = n - k + 1.$$

Por definición de distancia mínima,

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

Por otra parte, por la cota de Singleton,

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

Por lo tanto,

$$\boxed{d = n - k + 1}.$$

Así, el código Reed–Solomon alcanza la cota de Singleton y, por consiguiente, es un código MDS. □

2.2. Códigos Reed–Solomon

Los **códigos Reed–Solomon** constituyen una importante familia de códigos lineales que pertenecen a la clase de los códigos MDS. Fueron introducidos en 1960 por Reed y Solomon [2], y se caracterizan por su gran capacidad de corrección de errores y su estructura algebraica basada en polinomios sobre cuerpos finitos. Gracias a estas propiedades, los códigos Reed–Solomon se emplean ampliamente en sistemas de comunicación y almacenamiento digital, como discos compactos, códigos QR y transmisiones satelitales, donde la integridad de la información es esencial.

Código de evaluación: Sea $\mathbb{F}_q$ un cuerpo finito con q elementos y sea

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n), \qquad \alpha_i \in \mathbb{F}_q, \qquad \alpha_i \neq \alpha_j \text{ si } i \neq j.$$

Sea $V \subseteq \mathbb{F}_q[x]$ un subespacio vectorial de dimensión k. El *código de evaluación* asociado

a V y al conjunto de puntos de evaluación $\boldsymbol{\alpha}$ se define como

$$\mathcal{C}_{\boldsymbol{\alpha}}(V) := \left\{ \Big(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n) \Big) : f \in V \right\} \subseteq \mathbb{F}_q^n.$$

Es decir, cada palabra código se obtiene evaluando un polinomio perteneciente al espacio V en los n puntos de evaluación fijados.

Definición 2.10. El código Reed–Solomon de longitud n y dimensión k sobre $\mathbb{F}_q$, con conjunto de evaluación $\boldsymbol{\alpha}$, se obtiene como el caso particular del código de evaluación en el que

$$V_k = \{ f(x) \in \mathbb{F}_q[x] : \deg(f) < k \},$$

Por tanto,

$$\mathrm{RS}_{n,k}(\boldsymbol{\alpha}) = \left\{ \Big(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n) \Big) : f(x) \in \mathbb{F}_q[x],\ \deg(f) < k \right\}.$$

Parámetros de un Reed–Solomon: Todo $\mathrm{RS}_{n,k}$ es un $[n, k, d]$-código con distancia mínima dada por

$$d = n - k + 1.$$

En consecuencia, puede detectar hasta $d - 1 = n - k$ errores y corregir hasta

$$t = \left\lfloor \tfrac{d-1}{2} \right\rfloor = \left\lfloor \tfrac{n-k}{2} \right\rfloor$$

errores.

Polinomio generador: Sea $RS(n, k)$ un código Reed–Solomon con conjunto de evaluación $\{\alpha_1, \ldots, \alpha_n\}$. El *polinomio generador* asociado se define como

$$g(x) := (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_{n-k}) \in \mathbb{F}_q[X].$$

2.2.1. Codificación y decodificación en un código Reed–Solomon

La construcción del código Reed–Solomon $\mathrm{RS}_{n,k}(\boldsymbol{\alpha})$ se encuentra definida en la Definición 2.10. A partir de dicha construcción, la codificación consiste en evaluar un polinomio $f(X) \in \mathbb{F}_q[X]$ de grado menor que k en los elementos del conjunto de evaluación

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n,$$

donde los elementos $\alpha_1, \ldots, \alpha_n$ son distintos. Por tanto, la palabra código asociada a $f(X)$ es

$$\mathbf{c} = \Big(f(\alpha_1), \ldots, f(\alpha_n) \Big).$$

Codificación. Un mensaje se representa por un vector

$$\boldsymbol{m} = (m_0, \ldots, m_{k-1}) \in \mathbb{F}_q^k.$$

A este mensaje se le asocia el polinomio

$$f_{\boldsymbol{m}}(X) = \sum_{j=0}^{k-1} m_j X^j.$$

En otras palabras, los elementos m_j son los coeficientes del polinomio $f_{\boldsymbol{m}}(X)$ respecto de la base

$$\{1, X, X^2, \ldots, X^{k-1}\}$$

del espacio de polinomios de grado menor que k.

La palabra código correspondiente se obtiene evaluando $f_{\boldsymbol{m}}(X)$ en los puntos $\alpha_1, \ldots, \alpha_n$:

$$\boldsymbol{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f_{\boldsymbol{m}}) = \Big(f_{\boldsymbol{m}}(\alpha_1), \ldots, f_{\boldsymbol{m}}(\alpha_n)\Big).$$

Decodificación. Supóngase que el receptor observa una palabra

$$\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e},$$

donde $\boldsymbol{c} \in \mathrm{RS}_{n,k}(\boldsymbol{\alpha})$ es la palabra transmitida y $\boldsymbol{e} \in \mathbb{F}_q^n$ es un vector de error desconocido. El receptor conoce $\boldsymbol{r}$ y los parámetros del código $(q, n, k, \boldsymbol{\alpha})$, pero no conoce ni $\boldsymbol{c}$ ni $\boldsymbol{e}$. El problema de decodificación consiste en recuperar el mensaje original $\boldsymbol{m}$ a partir de $\boldsymbol{r}$.

Como la distancia mínima del código Reed–Solomon es

$$d = n - k + 1,$$

su capacidad de corrección de errores es

$$\tau = \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor.$$

Si el peso de Hamming del error satisface

$$w(\boldsymbol{e}) \leq \tau,$$

entonces existe una única palabra código $\boldsymbol{c}$ cuya distancia de Hamming a $\boldsymbol{r}$ es menor o igual que τ. En ese caso, la decodificación puede formularse como la búsqueda de un polinomio $f(X)$ de grado menor que k tal que

$$d_H\Big(\boldsymbol{r}, \mathrm{ev}_{\boldsymbol{\alpha}}(f)\Big) \leq \tau.$$

Una vez encontrado dicho polinomio, sus coeficientes determinan de manera única el mensaje original:

$$f(X) = m_0 + m_1 X + \cdots + m_{k-1} X^{k-1}.$$

Por tanto, recuperar $f(X)$ equivale a recuperar $\boldsymbol{m}$.

En resumen, el receptor conoce $\boldsymbol{r}$ y los parámetros del código, pero no conoce la

palabra transmitida ni los errores introducidos por el canal. La decodificación consiste en reconstruir la única palabra código dentro del radio de corrección τ y, a partir de ella, recuperar el polinomio $f_{\boldsymbol{m}}(X)$ y sus coeficientes, que determinan el mensaje original.

Ejemplo 2.11. Consideremos el cuerpo finito $\mathbb{F}_5$ y los puntos de evaluación

$$\boldsymbol{\alpha} = (0, 1, 2, 3).$$

Tomamos un código Reed–Solomon de longitud $n = 4$ y dimensión $k = 2$. Entonces, el espacio de mensajes está formado por vectores

$$\boldsymbol{m} = (m_0, m_1) \in \mathbb{F}_5^2,$$

y el polinomio asociado es

$$f_{\boldsymbol{m}}(X) = m_0 + m_1 X.$$

Sea, por ejemplo, el mensaje

$$\boldsymbol{m} = (3, 1).$$

El polinomio correspondiente es

$$f_{\boldsymbol{m}}(X) = 3 + X.$$

Al evaluarlo en los puntos de $\boldsymbol{\alpha}$ se obtiene la palabra código

$$\boldsymbol{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f_{\boldsymbol{m}}) = (3, 4, 0, 1).$$

Supongamos ahora que durante la transmisión ocurre un error en la segunda posición, de modo que el receptor observa

$$\boldsymbol{r} = (3, 2, 0, 1).$$

Como el código Reed–Solomon tiene distancia mínima

$$d = n - k + 1 = 3,$$

su radio de decodificación única es

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

Por tanto, como el vector de error satisface

$$w(\boldsymbol{e}) = 1 \leq \tau,$$

el receptor puede recuperar de forma única la palabra original $\boldsymbol{c}$ y, en consecuencia, el mensaje

$$\boldsymbol{m} = (3, 1).$$

Este ejemplo ilustra el proceso de codificación y decodificación descrito anteriormente: a partir del mensaje se construye el polinomio $f_{\boldsymbol{m}}(X)$, se evalúa en los puntos de $\boldsymbol{\alpha}$ para obtener la palabra código y, ante un error dentro del radio de corrección única, se recupera la palabra transmitida y, posteriormente, el mensaje original.

Recuperación del polinomio mensaje. Para recuperar el mensaje original, el receptor busca un polinomio

$$g(X) = a_0 + a_1 X, \qquad a_0, a_1 \in \mathbb{F}_5,$$

de grado menor que $k = 2$ tal que la palabra evaluada

$$\mathrm{ev}_{\boldsymbol{\alpha}}(g) = \Big(g(0), g(1), g(2), g(3)\Big)$$

esté a distancia de Hamming menor o igual que $\tau = 1$ de la palabra recibida

$$\boldsymbol{r} = (3, 2, 0, 1).$$

Como el código es Reed–Solomon, basta probar los polinomios lineales compatibles con al menos tres coordenadas de $\boldsymbol{r}$. Si imponemos, por ejemplo, que

$$g(0) = 3, \qquad g(2) = 0,$$

obtenemos el sistema

$$a_0 = 3, \qquad 3 + 2a_1 = 0 \quad \text{en } \mathbb{F}_5.$$

De aquí se sigue que

$$a_1 = 1,$$

y por tanto

$$g(X) = 3 + X.$$

Al evaluar este polinomio se obtiene

$$\mathrm{ev}_{\boldsymbol{\alpha}}(g) = (3, 4, 0, 1),$$

que difiere de $\boldsymbol{r}$ solo en la segunda coordenada. Por consiguiente, la palabra código original es

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

y el mensaje recuperado es

$$\boldsymbol{m} = (3, 1).$$

Proposición 2.12. *Sea* $\mathrm{RS}_{n,k}(\boldsymbol{\alpha})$ *un código Reed–Solomon de longitud* n *y dimensión* k *sobre* $\mathbb{F}_q$. *Si una palabra recibida* $\boldsymbol{r} \in \mathbb{F}_q^n$ *está a distancia de Hamming menor o igual que*

$$\tau = \left\lfloor \frac{n-k}{2} \right\rfloor$$

de una palabra código, entonces dicha palabra código es única. En particular, el polinomio de grado menor que k asociado a esa palabra código también es único.

Demostración. Supongamos que existen dos polinomios distintos $f(X)$ y $g(X)$, ambos de grado menor que k, tales que sus evaluaciones en $\boldsymbol{\alpha}$ satisfacen

$$d_H\Big(\mathrm{ev}_{\boldsymbol{\alpha}}(f), \boldsymbol{r}\Big) \leq \tau \qquad \text{y} \qquad d_H\Big(\mathrm{ev}_{\boldsymbol{\alpha}}(g), \boldsymbol{r}\Big) \leq \tau.$$

Entonces, por la desigualdad triangular,

$$d_H\Big(\mathrm{ev}_{\boldsymbol{\alpha}}(f), \mathrm{ev}_{\boldsymbol{\alpha}}(g)\Big) \leq d_H\Big(\mathrm{ev}_{\boldsymbol{\alpha}}(f), \boldsymbol{r}\Big) + d_H\Big(\boldsymbol{r}, \mathrm{ev}_{\boldsymbol{\alpha}}(g)\Big) \leq 2\tau.$$

Por otra parte, como $\mathrm{RS}_{n,k}(\boldsymbol{\alpha})$ es MDS, su distancia mínima es

$$d = n - k + 1.$$

Luego, si $f \neq g$, se tendría

$$d_H\Big(\mathrm{ev}_{\boldsymbol{\alpha}}(f), \mathrm{ev}_{\boldsymbol{\alpha}}(g)\Big) \geq d.$$

Pero

$$2\tau = 2\left\lfloor \frac{n-k}{2} \right\rfloor < n - k + 1 = d,$$

lo cual es una contradicción. Por tanto, no pueden existir dos palabras código distintas a distancia menor o igual que τ de $\boldsymbol{r}$.

Finalmente, como la aplicación

$$f(X) \longmapsto \mathrm{ev}_{\boldsymbol{\alpha}}(f)$$

es inyectiva sobre el espacio de polinomios de grado menor que k, la unicidad de la palabra código implica también la unicidad del polinomio recuperado. En consecuencia, el mensaje original es único. □

Capítulo 3

Polinomios y Códigos Reed–Solomon Torcidos

3.1. Polinomios y Códigos Reed–Solomon Torcidos

Definición 3.1. Sean $n, k, \ell \in \mathbb{N}$ enteros positivos con $k < n$. Llamamos a ℓ el *número de torsiones*. Definimos tres vectores:

$$\begin{aligned} \boldsymbol{t} &= [t_1, \ldots, t_\ell] \in \{1, \ldots, n-k\}^\ell && \text{(llamado } \textit{vector de torsión}\text{)}, \\ \boldsymbol{h} &= [h_1, \ldots, h_\ell] \in \{0, \ldots, k-1\}^\ell && \text{(llamado } \textit{vector de enganche}\text{)}, \\ \boldsymbol{\eta} &= [\eta_1, \ldots, \eta_\ell] \in \mathbb{F}_q^\ell && \text{(llamado } \textit{vector de coeficientes}\text{)}, \end{aligned}$$

tales que las parejas (h_i, t_i) son distintas para $i = 1, \ldots, \ell$.

Observación 3.2. La condición de que las parejas (h_i, t_i) sean distintas garantiza que cada torsión esté asociada a una combinación diferente de índice de enganche y grado de torsión, evitando que dos torsiones utilicen simultáneamente la misma posición del polinomio.

Definición 3.3 (Conjunto de polinomios $(\boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta})$-torcidos)**.** Con la notación de la Definición 3.1, se define el *conjunto de polinomios* $(\boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta})$*-torcidos* como

$$\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}} := \left\{ f(X) = \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j} \ : \ f_i \in \mathbb{F}_q \right\}.$$

Observación 3.4. El espacio $\mathcal{P}^{n,k}_{\boldsymbol{t},\boldsymbol{h},\boldsymbol{\eta}}$ se obtiene a partir del espacio de polinomios de grado menor que k, utilizado en los códigos Reed–Solomon clásicos, añadiendo, para cada torsión j, un término de grado

$$k - 1 + t_j \geq k,$$

cuyo coeficiente no es libre, sino que está determinado por el coeficiente f_{h_j} de la parte de grado menor que k, escalado por η_j.

Definición 3.5. Sean $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ elementos distintos y escribamos $\alpha = [\alpha_1, \ldots, \alpha_n]$. El *código de Reed–Solomon* $[\alpha, t, h, \eta]$*-torcido* se define como

$$C^{n,k}_{\alpha,t,h,\eta} := \mathrm{ev}_\alpha\left(\mathcal{P}^{n,k}_{t,h,\eta}\right) \subseteq \mathbb{F}_q^n,$$

donde $\mathrm{ev}_\alpha : \mathbb{F}_q[X] \to \mathbb{F}_q^n$ denota el mapa de evaluación definido por $\mathrm{ev}_\alpha(f) = (f(\alpha_1), \ldots, f(\alpha_n))$.

Lema 3.6. *El conjunto de polinomios* (t, h, η)*-torcidos* $\mathcal{P}^{n,k}_{t,h,\eta}$ *es un subespacio vectorial de*

dimensión k de $\mathbb{F}_q[X]$. Una base de $\mathcal{P}^{n,k}_{t,h,\eta}$ está dada por

$$g_i := X^i + \sum_{\substack{j=1 \\ h_j=i}}^{\ell} \eta_j \, X^{k-1+t_j}, \qquad i = 0, \ldots, k-1. \tag{3.1}$$

Demostración. La demostración consta de cuatro pasos.

Paso 1: $\mathcal{P}^{n,k}_{t,h,\eta}$ es subespacio vectorial de $\mathbb{F}_q[X]$.

Sean $f, g \in \mathcal{P}^{n,k}_{t,h,\eta}$ arbitrarios y $A \in \mathbb{F}_q$. Por definición del conjunto, estos polinomios admiten las representaciones

$$f = \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j}, \qquad g = \sum_{i=0}^{k-1} g_i X^i + \sum_{j=1}^{\ell} \eta_j g_{h_j} X^{k-1+t_j},$$

con $f_i, g_i \in \mathbb{F}_q$. Calculamos la combinación lineal $f + Ag$ distribuyendo y agrupando por potencias de X:

$$f + Ag = \sum_{i=0}^{k-1} (f_i + Ag_i) \, X^i + \sum_{j=1}^{\ell} \eta_j \, (f_{h_j} + Ag_{h_j}) \, X^{k-1+t_j}.$$

El coeficiente del término torcido de índice j es $\eta_j(f_{h_j} + Ag_{h_j})$, que coincide exactamente con η_j veces el coeficiente de X^{h_j} en la parte clásica de $f + Ag$. En consecuencia, $f + Ag \in \mathcal{P}^{n,k}_{t,h,\eta}$, lo que prueba que es subespacio.

Paso 2: Los polinomios g_i generan $\mathcal{P}^{n,k}_{t,h,\eta}$.

Sea $f = \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j} \in \mathcal{P}^{n,k}_{t,h,\eta}$ arbitrario. Reorganizamos el segundo sumando agrupando las torsiones según el índice $i = h_j$ al que apuntan, y factorizamos f_i:

$$\begin{aligned} f &= \sum_{i=0}^{k-1} f_i X^i + \sum_{i=0}^{k-1} \left(\sum_{\substack{j=1 \\ h_j=i}}^{\ell} \eta_j f_i \, X^{k-1+t_j} \right) \\ &= \sum_{i=0}^{k-1} f_i \left(X^i + \sum_{\substack{j=1 \\ h_j=i}}^{\ell} \eta_j \, X^{k-1+t_j} \right) = \sum_{i=0}^{k-1} f_i \, g_i. \end{aligned}$$

Luego todo elemento de $\mathcal{P}^{n,k}_{t,h,\eta}$ es combinación lineal de $\{g_0, \ldots, g_{k-1}\}$.

Paso 3: Los polinomios g_i son linealmente independientes.

Supóngase que $\sum_{i=0}^{k-1} A_i \, g_i = 0$ para ciertos $A_i \in \mathbb{F}_q$. Sustituyendo la definición (3.1) y separando por rangos de grado:

$$\underbrace{\sum_{i=0}^{k-1} A_i \, X^i}_{\text{grados } 0,\ldots,k-1} + \underbrace{\sum_{i=0}^{k-1} \sum_{\substack{j=1 \\ h_j=i}}^{\ell} A_i \, \eta_j \, X^{k-1+t_j}}_{\text{grados} \geq k} = 0.$$

Dado que $t_j \geq 1$, los exponentes del segundo sumando satisfacen $k-1+t_j \geq k$, de modo que los dos grupos de monomios tienen *rangos de grado disjuntos*. Por tanto, al igualar a cero el coeficiente de X^{i_0} para cada $i_0 \in \{0,\ldots,k-1\}$, únicamente el primer sumando contribuye, y se obtiene $A_{i_0} = 0$. Como i_0 es arbitrario, concluimos $A_i = 0$ para todo i.

Paso 4: Conclusión.

El conjunto $\{g_0,\ldots,g_{k-1}\}$ genera $\mathcal{P}^{n,k}_{t,h,\eta}$ y es linealmente independiente, luego es una base. Por tanto,

$$\dim_{\mathbb{F}_q} \mathcal{P}^{n,k}_{t,h,\eta} = k. \qquad \square$$

Ejemplo 3.7 (Verificación computacional del Lema 3.6)**.** El siguiente código en SageMath construye el espacio de polinomios torcidos $\mathcal{P}^{n,k}_{t,h,\eta}$ con parámetros $q=8$, $n=7$, $k=3$, $\ell=1$, $t=[2]$, $h=[1]$ y $\eta_1 = a$ (elemento primitivo de $\mathbb{F}_8$), y verifica que los polinomios g_i forman una base de dimensión $k=3$.

```
# Cuerpo finito F_8 = GF(2^3)
F.<a> = GF(8)
R.<X> = PolynomialRing(F)

# Par metros
n, k, ell = 7, 3, 1
t, h, eta = [2], [1], [a]

# Construccion de los polinomios base g_i
def construir_gi(i, k, t, h, eta, R):
    X  = R.gen()
    gi = X^i
    for j in range(len(h)):
        if h[j] == i:
            gi += eta[j] * X^(k - 1 + t[j])
    return gi

base_torcida = [construir_gi(i,k,t,h,eta,R) for i in range(k)]

# Verificacion de independencia lineal (rango = k)
max_grado = k - 1 + max(t)
M = matrix(F, [[p[i] for i in range(max_grado+1)]
               for p in base_torcida])
print(f"Rango de la matriz de coeficientes: {M.rank()}")
# Salida esperada: 3 (igual a k)
```

Listado 3.1: Construcción del espacio torcido y verificación en SageMath

```
=========================================================
CUERPO FINITO F_8
=========================================================
```

```
Polinomio minimal de a : x^3 + x + 1
Orden multiplicativo de a : 7

=========================================================
VERIFICACION DE PARAMETROS
=========================================================
t_1 = 2 en {1,...,4} : True
h_1 = 1 en {0,...,2} : True
Tuplas [h_j,t_j] = [(1, 2)]
Son distintas : True

=========================================================
POLINOMIOS BASE
=========================================================
 i      g_i clasico           g_i torcido
-----------------------------------------
 0                1                     1
 1                X           a*X^4 + X
 2             X^2                   X^2

=========================================================
VERIFICACION DE INDEPENDENCIA LINEAL
=========================================================
Rango de la matriz de coeficientes : 3
Debe ser k = 3 : True

=========================================================
VERIFICACION DE LA FORMULA f = sum f_i * g_i
=========================================================
Coeficientes: f_0=1, f_1=a, f_2=a^2
f por definicion : a^2*X^4 + a^2*X^2 + a*X + 1
f = sum f_i*g_i  : a^2*X^4 + a^2*X^2 + a*X + 1
Son iguales      : True
```

Listado 3.2: Salida de Sage para el Ejemplo

La salida del programa confirma que el rango es $3 = k$, lo que verifica computacionalmente que $\{g_0, g_1, g_2\}$ es un sistema linealmente independiente y, por tanto, una base de $\mathcal{P}^{n,k}_{t,h,\eta}$.

Proposición 3.8. *Un código de Reed–Solomon $[\alpha, t, h, \eta]$-torcido $C^{n,k}_{\alpha,t,h,\eta}$ es un código lineal $[n, k]$. Con $g_0, \ldots, g_{k-1}$ como en* (3.1)*, la matriz*

$$G := \begin{bmatrix} \mathrm{ev}_\alpha(g_0) \\ \vdots \\ \mathrm{ev}_\alpha(g_{k-1}) \end{bmatrix} \in \mathbb{F}_q^{k \times n} \tag{3.2}$$

es una matriz generadora de $C^{n,k}_{\alpha,t,h,\eta}$.

Demostración. La demostración se desarrolla en tres etapas.

Etapa 1: El código es lineal.

El mapa de evaluación

$$\mathrm{ev}_\alpha : \mathbb{F}_q[X] \longrightarrow \mathbb{F}_q^n, \qquad f \mapsto (f(\alpha_1), \ldots, f(\alpha_n)),$$

es $\mathbb{F}_q$-lineal, pues para todo $f, g \in \mathbb{F}_q[X]$ y $A \in \mathbb{F}_q$ se tiene $(f + Ag)(\alpha_i) = f(\alpha_i) + Ag(\alpha_i)$ para cada i. Dado que $\mathcal{P}^{n,k}_{t,h,\eta}$ es un subespacio vectorial sobre $\mathbb{F}_q$ (Lema 3.6), su imagen bajo ev_α, que es precisamente $C^{n,k}_{\alpha,t,h,\eta}$, es también un subespacio de $\mathbb{F}_q^n$. Luego $C^{n,k}_{\alpha,t,h,\eta}$ es un código lineal.

Etapa 2: El mapa de evaluación es inyectivo sobre $\mathcal{P}^{n,k}_{t,h,\eta}$, y la dimensión del código es k.

Todo $f \in \mathcal{P}^{n,k}_{t,h,\eta}$ tiene la forma

$$f = \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j}.$$

Los exponentes de la parte torcida satisfacen $k - 1 + t_j \leq k - 1 + (n - k) = n - 1$, pues $t_j \leq n - k$ por hipótesis. Por tanto $\deg f \leq n - 1 < n$ para todo $f \in \mathcal{P}^{n,k}_{t,h,\eta}$.

Sea ahora $f \in \mathcal{P}^{n,k}_{t,h,\eta}$ con $\mathrm{ev}_\alpha(f) = \mathbf{0}$, es decir, $f(\alpha_i) = 0$ para todo $i = 1, \ldots, n$. Entonces f posee n raíces distintas (pues los α_i son distintos) y $\deg f \leq n - 1$. Un polinomio no nulo de grado a lo sumo $n - 1$ tiene a lo sumo $n - 1$ raíces, por lo que la única posibilidad es $f = 0$. Esto prueba que ev_α es inyectivo sobre $\mathcal{P}^{n,k}_{t,h,\eta}$.

Por el Teorema Rango-Nulidad, la inyectividad implica

$$\dim_{\mathbb{F}_q}\left(C^{n,k}_{\alpha,t,h,\eta}\right) = \dim_{\mathbb{F}_q}\left(\mathcal{P}^{n,k}_{t,h,\eta}\right) = k,$$

donde la ultima igualdad es el Lema 3.6.

Etapa 3: La matriz G es generadora.

Por el Lema 3.6, el conjunto $\{g_0, \ldots, g_{k-1}\}$ es una base de $\mathcal{P}^{n,k}_{t,h,\eta}$. Como ev_α es lineal e inyectivo, transforma bases en bases; en consecuencia, $\{\mathrm{ev}_\alpha(g_0), \ldots, \mathrm{ev}_\alpha(g_{k-1})\}$ es una base de $C^{n,k}_{\alpha,t,h,\eta}$. La matriz G definida en (3.2) tiene precisamente estos vectores como filas, por lo que es una matriz generadora de $C^{n,k}_{\alpha,t,h,\eta}$. □

Ejemplo 3.9 (Verificación computacional de la Proposición 3.8)**.** El siguiente código en SageMath construye la matriz generadora G del código Reed–Solomon torcido $C^{n,k}_{\alpha,t,h,\eta}$ con parámetros $q = 8$, $n = 7$, $k = 3$, $\ell = 1$, $t = [2]$, $h = [1]$ y $\eta_1 = a$ (elemento primitivo de $\mathbb{F}_8$), y la compara con la matriz generadora del código Reed–Solomon clásico.

```
# Cuerpo finito F_8 = GF(2^3)
F.<a> = GF(8)
R.<X> = PolynomialRing(F)
```

```
4
5  # Par metros
6  n, k, ell = 7, 3, 1
7  t, h, eta = [2], [1], [a]
8
9  # Puntos de evaluacion: los 7 elementos no nulos de F_8
10 alpha = [a^i for i in range(n)]
11
12 # Construccion de los polinomios base g_i
13 def construir_gi(i, k, t, h, eta, R):
14     X  = R.gen()
15     gi = X^i
16     for j in range(len(h)):
17         if h[j] == i:
18             gi += eta[j] * X^(k - 1 + t[j])
19     return gi
20
21 base_torcida = [construir_gi(i, k, t, h, eta, R) for i in range
       (k)]
22
23 # Matrices generadoras
24 # Clasico: fila i = evaluacion de X^i
25 G_RS = matrix(F, [[pt^i for pt in alpha] for i in range(k)])
26
27 # Torcido: fila i = evaluacion de g_i
28 G_T  = matrix(F, [[gi(pt) for pt in alpha] for gi in
       base_torcida])
29
30 print("=" * 55)
31 print("MATRICES GENERADORAS")
32 print("=" * 55)
33 print(f"\nG_RS (Reed-Solomon clasico) [{k}x{n}]:")
34 print(G_RS)
35 print(f"\nG_T  (Reed-Solomon torcido) [{k}x{n}]:")
36 print(G_T)
37
38 # Verificacion de rango
39 print("\n" + "=" * 55)
40 print("VERIFICACION DE DIMENSION")
41 print("=" * 55)
42 print(f"Rango de G_RS : {G_RS.rank()}  (debe ser k={k})")
43 print(f"Rango de G_T  : {G_T.rank()}   (debe ser k={k})")
44
45 # Verificacion de inyectividad de ev_alpha
46 print("\n" + "=" * 55)
```

```
print("VERIFICACION DE INYECTIVIDAD DE ev_alpha")
print("=" * 55)
print(f"Grado m ximo en P^{{n,k}}: {k-1+max(t)} (debe ser < n
    ={n})")
print(f"Condicion t_j <= n-k: {all(tj <= n-k for tj in t)}")
print(f"ev_alpha es inyectiva: {G_T.rank() == k}")

# Comparacion de los c digos
print("\n" + "=" * 55)
print("COMPARACION DE LOS ESPACIOS CODIGO")
print("=" * 55)
C_RS = G_RS.row_space()
C_T  = G_T.row_space()
print(f"Dim codigo RS cl sico : {C_RS.dimension()}")
print(f"Dim codigo RS torcido : {C_T.dimension()}")
print(f"Son el mismo codigo   : {C_RS == C_T}")
# Esperamos False: misma dimensi n pero c digos distintos
```

Listado 3.3: Construcción de las matrices generadoras clásica y torcida en SageMath

Listado 3.4: Salida de SageMath para la verificación de la Proposición 3.8

MATRICES GENERADORAS

$$G_{\mathrm{RS}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & a & a^2 & a+1 & a^2+a & a^2+a+1 & a^2+1 \\ 1 & a^2 & a^2+a & a^2+1 & a & a+1 & a^2+a+1 \end{bmatrix},$$

$$G_T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a+1 & a^2+1 & 0 & a^2+a & a^2+1 & a^2+a & a+1 \\ 1 & a^2 & a^2+a & a^2+1 & a & a+1 & a^2+a+1 \end{bmatrix}.$$

```
1  ======================================================
2  VERIFICACION DE DIMENSION
3  ======================================================
4  Rango de G_RS : 3  (debe ser k=3)
5  Rango de G_T  : 3   (debe ser k=3)
6
7  ======================================================
8  VERIFICACION DE INYECTIVIDAD DE ev_alpha
9  ======================================================
10 Grado maximo en P^{n,k}: 4 (debe ser < n=7)
11 Condicion t_j <= n-k: True
12 ev_alpha es inyectiva: True
13
14 ======================================================
15 COMPARACION DE LOS ESPACIOS CODIGO
16 ======================================================
17 Dim codigo RS clasico : 3
18 Dim codigo RS torcido : 3
19 Son el mismo codigo   : False
```

La salida del programa muestra que ambas matrices generadoras tienen rango $k = 3$, por lo que los códigos $C_{\alpha}^{n,k}$ y $C_{\alpha,t,h,\eta}^{n,k}$ tienen la misma dimensión, pero la última línea confirma que los espacios código son distintos, es decir, $C_{\alpha}^{n,k} \neq C_{\alpha,t,h,\eta}^{n,k}$.

Observación 3.10. La condición $t_j \leq n - k$ no es arbitraria: es exactamente lo que garantiza que todos los polinomios del espacio torcido tengan grado estrictamente menor que n, asegurando así la inyectividad del mapa de evaluación y, con ello, que la dimensión del código sea exactamente k. Si se permitiera $t_j > n - k$, el grado podría alcanzar n o más, y la evaluación dejaría de ser inyectiva.

Observación 3.11. Hacemos cuatro observaciones importantes sobre la Definición 3.1. Estas aclaraciones precisan el alcance de cada condición impuesta y muestran cómo la familia de códigos torcidos generaliza de manera natural a los códigos Reed–Solomon clásicos.

Observación 1: Origen del nombre "torcido".

Los códigos Reed–Solomon torcidos no están relacionados con los códigos BCH torcidos. El nombre está inspirado en los *códigos de Gabidulin torcidos de Sheekey* [3], los cuales están relacionados con campos torcidos generalizados y pertenecen al mundo de los códigos en métrica de rango. La analogía es de naturaleza estructural e histórica, no matemática.

Observación 2: La condición de tuplas distintas no es una restricción real sobre la familia de códigos.

La Definición 3.1 exige que las tuplas $[h_i, t_i]$ sean distintas para $i = 1, \ldots, \ell$. Esta condición podría parecer una restricción sobre qué códigos se pueden construir, pero en realidad *no lo es*: cualquier código que se pudiera construir sin esta condición también puede construirse con ella, posiblemente con un número menor de torsiones.

Para ver por qué, supongamos que para algún par de índices $i \neq j$ tenemos $[h_i, t_i] = [h_j, t_j]$, es decir, $h_i = h_j$ y $t_i = t_j$. Esto significa que ambas torsiones apuntan al mismo coeficiente de enganche $h_i = h_j$ y producen un término de la misma potencia $X^{k-1+t_i} = X^{k-1+t_j}$. Miremos qué ocurre en el polinomio base g_s donde $s = h_i = h_j$:

$$g_s = X^s + \cdots + \eta_i X^{k-1+t_i} + \eta_j X^{k-1+t_j} + \cdots$$

Como $t_i = t_j$, los dos términos torcidos son el *mismo monomio* X^{k-1+t_i}, por lo que sus coeficientes simplemente se suman:

$$g_s = X^s + \cdots + (\eta_i + \eta_j) X^{k-1+t_i} + \cdots$$

Esto es exactamente lo que se obtendría con *una sola* torsión apuntando al índice s, con coeficiente $\eta_i + \eta_j$. Por tanto, el código construido con las dos torsiones repetidas es *idéntico* al código obtenido eliminando una de ellas y reemplazando el coeficiente correspondiente por la suma. Repitiendo este proceso hasta eliminar todas las repeticiones, se obtiene una representación con tuplas distintas que genera el mismo código. La condición es, pues, una *normalización de la representación*, no una restricción sobre los códigos que se pueden construir.

Ejemplo 3.12 (Verificación computacional de la Observación anterior)**.** El siguiente código en SageMath demuestra que dos torsiones con la misma tupla $(h_j, t_j) = (1, 2)$ pero coeficientes distintos $\eta_1 = a$, $\eta_2 = a^3$ producen exactamente el mismo código que una sola torsión con $\eta = \eta_1 + \eta_2$.

```
# Cuerpo finito F_16 = GF(2^4)
F.<a> = GF(16)
R.<X> = PolynomialRing(F)
n, k  = 7, 3

# Sistema CON tuplas repetidas
ell_rep  = 2
h_rep    = [1, 1]
```

```
t_rep    = [2, 2]
eta1, eta2 = a, a^3
eta_rep  = [eta1, eta2]

# Sistema FUSIONADO (una sola torsion)
ell_fus  = 1
h_fus    = [1]
t_fus    = [2]
eta_fus  = [eta1 + eta2]   # suma de coeficientes

# Puntos de evaluaci n
alpha = [a^i for i in range(n)]

# Construcci n de los polinomios base g_i
def construir_gi(i, k, t, h, eta, R):
    X  = R.gen()
    gi = X^i
    for j in range(len(h)):
        if h[j] == i:
            gi += eta[j] * X^(k - 1 + t[j])
    return gi

# Polinomios base de ambos sistemas
base_rep = [construir_gi(i, k, t_rep, h_rep, eta_rep, R) for i
   in range(k)]
base_fus = [construir_gi(i, k, t_fus, h_fus, eta_fus, R) for i
   in range(k)]

print("=" * 55)
print("POLINOMIOS BASE: REPETIDO vs FUSIONADO")
print("=" * 55)
for i in range(k):
    print(f"g_{i} repetido  : {base_rep[i]}")
    print(f"g_{i} fusionado : {base_fus[i]}")
    print(f"Son iguales     : {base_rep[i] == base_fus[i]}")
    print()

# Matrices generadoras
G_rep = matrix(F, [[gi(pt) for pt in alpha] for gi in base_rep
   ])
G_fus = matrix(F, [[gi(pt) for pt in alpha] for gi in base_fus
   ])

# Espacios codigo
C_rep = G_rep.row_space()
```

```
C_fus = G_fus.row_space()

print("=" * 55)
print("COMPARACION DE LOS codigos RESULTANTES")
print("=" * 55)
print(f"Codigo con torsiones repetidas = Codigo fusionado : {
    C_rep == C_fus}")
print(f"eta_1 + eta_2 = {eta1} + {eta2} = {eta1 + eta2}")
```

Listado 3.5: Comparación de códigos RS torcidos con tuplas repetidas vs fusionadas

```
=======================================================
POLINOMIOS BASE: REPETIDO vs FUSIONADO
=======================================================
g_0 repetido  : 1
g_0 fusionado : 1
Son iguales     : True

g_1 repetido  : (a^3 + a)*X^4 + X
g_1 fusionado : (a^3 + a)*X^4 + X
Son iguales     : True

g_2 repetido  : X^2
g_2 fusionado : X^2
Son iguales     : True

=======================================================
COMPARACION DE LOS codigos RESULTANTES
=======================================================
Codigo con torsiones repetidas = Codigo fusionado : True
eta_1 + eta_2 = a + a^3 = a^3 + a
```

Listado 3.6: Salida de SageMath para la verificación de la Observación

La salida confirma que los polinomios g_i son idénticos y que ambos sistemas generan exactamente el mismo espacio código $C^{n,k}_{\alpha,t,h,\eta}$, lo que justifica la normalización de tuplas repetidas en la definición.

Ejemplo 3.13. Ilustramos la segunda observación con un ejemplo concreto. Tomamos $k = 3$, $\ell = 2$, y los vectores

$$h = [1,\, 1], \quad t = [2,\, 2], \quad \eta = [\eta_1,\, \eta_2].$$

Las tuplas son $[h_1, t_1] = [1, 2]$ y $[h_2, t_2] = [1, 2]$, que son iguales, en violación de la condición de la Definición 3.1. Construimos los polinomios base para ver el efecto de esta repetición. Como ambas torsiones tienen $h_j = 1$, únicamente el polinomio g_1 recibe

términos adicionales:

$$g_0 = X^0, \qquad g_1 = X + \eta_1 X^{k-1+t_1} + \eta_2 X^{k-1+t_2}, \qquad g_2 = X^2.$$

Dado que $t_1 = t_2 = 2$ y $k = 3$, ambos términos torcidos producen el mismo monomio $X^{k-1+2} = X^4$, por lo que sus coeficientes se suman:

$$g_1 = X + \eta_1 X^4 + \eta_2 X^4 = X + (\eta_1 + \eta_2) X^4.$$

Este polinomio es idéntico al que se obtiene con una sola torsión de parámetros

$$\ell' = 1, \quad h' = [1], \quad t' = [2], \quad \eta' = [\eta_1 + \eta_2],$$

para la cual $g_1' = X + (\eta_1 + \eta_2)X^4$. Por tanto, ambos sistemas de parámetros generan exactamente el mismo código. La torsión repetida no aporta información nueva: simplemente suma los coeficientes η_1 y η_2 en un único término. Este proceso de fusión puede repetirse hasta que todas las tuplas $[h_j, t_j]$ sean distintas, sin alterar el código resultante.

Observación 3: Permitir $\eta_i = 0$ incluye a los códigos Reed–Solomon clásicos.

En principio, para que los términos torcidos aporten información genuinamente nueva al espacio $\mathcal{P}^{n,k}_{t,h,\eta}$, convendría exigir $\eta_j \neq 0$ para todo j. Sin embargo, el paper permite explícitamente que los η_j sean cero, y esto tiene una razón precisa: *garantiza que la familia de códigos torcidos incluya de manera natural a los códigos Reed–Solomon clásicos.*

Para verlo, observemos qué ocurre cuando $\eta_j = 0$ para todo $j = 1, \ldots, \ell$. En ese caso, cada término torcido de la definición del espacio se anula:

$$\sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j} = \sum_{j=1}^{\ell} 0 \cdot f_{h_j} X^{k-1+t_j} = 0,$$

y el espacio torcido se reduce al espacio clásico de polinomios de grado menor que k:

$$\mathcal{P}^{n,k}_{t,h,\eta}\Big|_{\eta=0} = \left\{ \sum_{i=0}^{k-1} f_i X^i \;\middle|\; f_i \in \mathbb{F}_q \right\} = \mathbb{F}_q[X]_{<k}.$$

El código resultante es exactamente el código Reed–Solomon clásico con puntos de evaluación $\alpha_1, \ldots, \alpha_n$. Los códigos RS clásicos son, por tanto, un *caso particular* de los códigos torcidos, correspondiente a $\eta = 0$.

Observación 4: La condición $t_i \leq n - k$ puede relajarse.

En la demostración de la Proposición 3.8 se usó la condición $t_j \leq n - k$ para garantizar que todo polinomio $f \in \mathcal{P}^{n,k}_{t,h,\eta}$ satisface $\deg f \leq n - 1$. Esta cota en el grado fue lo que permitió concluir la inyectividad del mapa de evaluación ev_α: un polinomio no nulo de grado menor que n no puede anularse en los n puntos distintos $\alpha_1, \ldots, \alpha_n$.

Sin embargo, esta condición es *suficiente pero no necesaria*. La condición realmente necesaria y suficiente para la inyectividad de ev_α es más sutil. Para entenderla, observemos

que dos polinomios f y g producen la misma palabra código si y sólo si $f(\alpha_i) = g(\alpha_i)$ para todo i, lo que equivale a que $f - g$ se anule en todos los α_i, es decir, que $f - g$ sea múltiplo del polinomio

$$\omega(X) := \prod_{j=1}^{n}(X - \alpha_j).$$

Por tanto, lo que determina completamente la palabra código de un polinomio f no es f en sí mismo, sino su *clase de equivalencia módulo* $\omega(X)$, que es simplemente el residuo de dividir f entre $\omega(X)$.

Con esto en mente, la condición necesaria y suficiente para que ev_α sea inyectivo sobre $\mathcal{P}^{n,k}_{t,h,\eta}$ es que los polinomios reducidos

$$g_i \pmod{\omega(X)} = g_i \pmod{\prod_{j=1}^{n}(X - \alpha_j)}, \qquad i = 0, \ldots, k-1,$$

sean *linealmente independientes* sobre $\mathbb{F}_q$.

Cuando $t_j \leq n - k$, se tiene $\deg g_i \leq n - 1 < n = \deg \omega$, por lo que la reducción módulo ω no modifica a g_i, y su independencia lineal ya fue establecida en el Lema 3.6. Cuando $\deg g_i \geq n$, la reducción sí altera los polinomios, y verificar la independencia lineal requiere condiciones adicionales sobre los parámetros α, t, h, η, cuyo análisis es técnicamente más delicado.

Ejemplo 3.14. Ilustramos la cuarta observación con un ejemplo donde la condición $t_j \leq n - k$ se viola, pero el código sigue estando bien definido. Tomamos $q = 5$, $n = 4$, $k = 2$, $\ell = 1$, $h = [0]$, $t = [3]$ y $\eta_1 = 1 \in \mathbb{F}_5$.

Primero verificamos que la condición estándar no se cumple: $n - k = 4 - 2 = 2$, pero $t_1 = 3 > 2$. Los polinomios base son

$$g_0 = 1 + X^{k-1+t_1} = 1 + X^{1+3} = 1 + X^4, \qquad g_1 = X.$$

El grado de g_0 es $4 = n$, por lo que el argumento de la Proposición 3.8 no aplica directamente: no podemos garantizar la inyectividad de ev_α por la vía habitual.

Recurrimos entonces a la condición necesaria y suficiente descrita en la Observación 3.11. Los puntos de evaluación son $\alpha_1 = 1$, $\alpha_2 = 2$, $\alpha_3 = 3$, $\alpha_4 = 4$, todos distintos en $\mathbb{F}_5$, y el polinomio que se anula en ellos es

$$\omega(X) = (X - 1)(X - 2)(X - 3)(X - 4).$$

Calculamos $\omega(X)$ sobre $\mathbb{F}_5$ agrupando en pares:

$$(X - 1)(X - 4) = X^2 - 5X + 4 = X^2 + 4,$$
$$(X - 2)(X - 3) = X^2 - 5X + 6 = X^2 + 1,$$

donde se usó que $5 \equiv 0$ en $\mathbb{F}_5$. Por tanto,

$$\omega(X) = (X^2+4)(X^2+1) = X^4 + X^2 + 4X^2 + 4 = X^4 + 5X^2 + 4 = X^4 + 4,$$

de nuevo usando $5 \equiv 0$.

Ahora reducimos los polinomios base módulo $\omega(X) = X^4 + 4$. De la relación $X^4 + 4 \equiv 0$ se obtiene $X^4 \equiv -4 \equiv 1$ en $\mathbb{F}_5$. Entonces:

$$g_0 \bmod \omega = 1 + X^4 \equiv 1 + 1 = 2,$$
$$g_1 \bmod \omega = X.$$

Los polinomios reducidos son 2 (constante no nula) y X (de grado 1). Para verificar su independencia lineal, suponemos $A_0 \cdot 2 + A_1 \cdot X = 0$ para ciertos $A_0, A_1 \in \mathbb{F}_5$. Igualando coeficientes: el coeficiente de X^0 da $2A_0 = 0$, de donde $A_0 = 0$; el coeficiente de X^1 da $A_1 = 0$. Los polinomios reducidos son, pues, linealmente independientes.

Por la condición necesaria y suficiente, el mapa de evaluación ev_α es inyectivo sobre $\mathcal{P}^{n,k}_{t,h,\eta}$, el código está bien definido, y su dimensión es $k = 2$, todo ello a pesar de que $t_1 = 3 > n - k = 2$ viola la condición estándar. Esto confirma que dicha condición es suficiente pero no necesaria para la validez de la construcción.

Ejemplo 3.15. Ilustramos la Proposición 3.8 con tres elecciones concretas de parámetros, que producen matrices generadoras bien diferenciadas. Para simplificar la escritura, adoptamos la notación

$$\alpha^i := [\alpha_1^i,\, \alpha_2^i,\, \ldots,\, \alpha_n^i],$$

que representa el vector fila formado por la i-ésima potencia de cada punto de evaluación. Cada una de estas filas corresponde a la evaluación de un polinomio base g_i en los n puntos.

Caso 1: $\eta = 0$ (código Reed–Solomon clásico y matriz de Vandermonde).

Cuando todos los coeficientes son $\eta_j = 0$, la Observación 3 establece que el espacio torcido se reduce al espacio clásico $\mathbb{F}_q[X]_{<k}$, y los polinomios base del Lema 3.6 son simplemente $g_i = X^i$ para $i = 0, \ldots, k-1$. La evaluación de g_i en los puntos $\alpha_1, \ldots, \alpha_n$ produce el vector $[\alpha_1^i, \ldots, \alpha_n^i] = \alpha^i$, de modo que la matriz generadora de la Proposición 3.8 toma la forma

$$G = \begin{bmatrix} \alpha^0 \\ \alpha^1 \\ \vdots \\ \alpha^{k-1} \end{bmatrix} = \begin{bmatrix} \alpha_1^0 & \alpha_2^0 & \cdots & \alpha_n^0 \\ \alpha_1^1 & \alpha_2^1 & \cdots & \alpha_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \cdots & \alpha_n^{k-1} \end{bmatrix}.$$

Esta es la clásica *matriz de Vandermonde* asociada a los puntos de evaluación $\alpha_1, \ldots, \alpha_n$, y es la matriz generadora estándar de cualquier código Reed–Solomon clásico. Este caso confirma que los códigos RS clásicos quedan completamente absorbidos dentro del marco torcido.

Caso 2: Código de Glynn ($q = n = 9$, $k = 5$, $\ell = 1$, $h_1 = 2$, $t_1 = 2$, η_1 no-cuadrado

en $\mathbb{F}_9$).

Antes de construir la matriz, aclaramos qué significa que η_1 sea un *no-cuadrado* en $\mathbb{F}_9$. El grupo multiplicativo $\mathbb{F}_9^* = \langle\alpha\rangle$ es cíclico de orden $q-1=8$. Los *cuadrados* son los elementos de la forma β^2 para algún $\beta \in \mathbb{F}_9^*$, que corresponden a las potencias pares de α: $\{\alpha^0, \alpha^2, \alpha^4, \alpha^6\}$. Los *no-cuadrados* son los elementos que no son cuadrados de ningún elemento del cuerpo, y corresponden a las potencias impares: $\{\alpha^1, \alpha^3, \alpha^5, \alpha^7\}$. El paper requiere que η_1 sea uno de estos últimos.

Determinamos ahora cuáles polinomios base g_i se ven afectados por la torsión. Con $\ell = 1$ hay una sola torsión con $h_1 = 2$, lo que significa que *únicamente* el polinomio g_2 (el correspondiente al índice $i = 2$) recibe un término adicional. Para los demás índices, $g_i = X^i$. El término torcido añadido a g_2 tiene potencia $k - 1 + t_1 = 5 - 1 + 2 = 6$, por lo que:

$$\begin{aligned} g_0 &= X^0, \\ g_1 &= X^1, \\ g_2 &= X^2 + \eta_1 X^6, \\ g_3 &= X^3, \\ g_4 &= X^4. \end{aligned}$$

Evaluando cada g_i en los $n = 9$ puntos $\alpha_1, \ldots, \alpha_9$ y colocando los resultados como filas de la matriz, la matriz generadora es

$$G = \begin{bmatrix} \alpha^0 \\ \alpha^1 \\ \alpha^2 + \eta_1\,\alpha^6 \\ \alpha^3 \\ \alpha^4 \end{bmatrix}.$$

Nótese que únicamente la tercera fila difiere de la fila correspondiente en la matriz de Vandermonde clásica: lleva el término adicional $\eta_1\,\alpha^6$, que es la "torsión" concreta introducida por los parámetros $h_1 = 2$, $t_1 = 2$ y el escalar η_1.

Este código coincide con el *código de Glynn* [4], que es históricamente relevante por ser el primer código MDS conocido con las siguientes propiedades simultáneas: cuerpo base de característica impar, longitud $n = q + 1$, dimensión $3 \le k \le q - 1$, y que *no es un código Reed–Solomon Generalizado*. Su existencia demostró que la familia de códigos MDS es estrictamente más amplia que la familia de los RS generalizados. El paper de Beelen, Puchinger y Rosenkilde muestra que el código de Glynn es un caso especial de la construcción torcida, lo que retroactivamente lo integra dentro de un marco algebraico unificado.

Caso 3: Múltiples torsiones ($q \ge n > 7$, $k = 5$, $t = [1, 3, 3]$, $h = [4, 4, 2]$).

En este caso hay $\ell = 3$ torsiones. Antes de construir los polinomios base, verificamos

que las tuplas $[h_j, t_j]$ sean efectivamente distintas, como exige la Definición 3.1:

$$[h_1, t_1] = [4, 1], \quad [h_2, t_2] = [4, 3], \quad [h_3, t_3] = [2, 3],$$

que son tres tuplas distintas. La condición se satisface.

Ahora determinamos qué torsiones apuntan a cada índice i:

- Índice $i = 0$: ningún h_j vale 0. Luego $g_0 = X^0$.
- Índice $i = 1$: ningún h_j vale 1. Luego $g_1 = X^1$.
- Índice $i = 2$: solo $j = 3$ tiene $h_3 = 2$, con potencia $k - 1 + t_3 = 4 + 3 = 7$. Luego

$$g_2 = X^2 + \eta_3 X^7.$$

- Índice $i = 3$: ningún h_j vale 3. Luego $g_3 = X^3$.
- Índice $i = 4$: tanto $j = 1$ (con $h_1 = 4$, potencia $k - 1 + t_1 = 4 + 1 = 5$) como $j = 2$ (con $h_2 = 4$, potencia $k - 1 + t_2 = 4 + 3 = 7$) apuntan a este índice. Luego g_4 recibe *dos* términos torcidos:

$$g_4 = X^4 + \eta_1 X^5 + \eta_2 X^7.$$

Es importante observar que g_4 tiene dos términos torcidos porque *dos torsiones distintas* ($j = 1$ y $j = 2$) tienen el mismo enganche $h_j = 4$, pero con potencias t_j diferentes ($t_1 = 1$ y $t_2 = 3$), de modo que producen monomios de distinto grado: X^5 y X^7 respectivamente. Esto es consistente con la condición de que las tuplas $[h_j, t_j]$ sean distintas: la tupla $[4, 1]$ difiere de la tupla $[4, 3]$ aunque compartan el mismo enganche $h_j = 4$.

La matriz generadora resultante es

$$G = \begin{bmatrix} \alpha^0 \\ \alpha^1 \\ \alpha^2 + \eta_3 \alpha^7 \\ \alpha^3 \\ \alpha^4 + \eta_1 \alpha^5 + \eta_2 \alpha^7 \end{bmatrix}.$$

3.2. Códigos Reed–Solomon Torcidos MDS

No todos los códigos Reed–Solomon torcidos son MDS. En esta sección establecemos una condición algebraica sobre el vector de coeficientes η que garantiza que el código torcido alcance la cota de Singleton, y presentamos varias familias de códigos RS torcidos MDS.

3.2.1. Una Condición General para MDS

Antes de enunciar la condición, recordamos un criterio matricial que usaremos a lo largo de esta sección.

Observación 3.16. Un código lineal $[n, k]$ con matriz generadora $G \in \mathbb{F}_q^{k \times n}$ es MDS si y solo si todo menor $k \times k$ de G es distinto de cero, es decir, toda submatriz cuadrada formada por k columnas cualesquiera de G tiene determinante no nulo. Esto equivale a exigir que

cualquier k columnas de G sean linealmente independientes, lo cual es precisamente la condición para que la distancia mínima sea $d = n - k + 1$ [5].

Definición 3.17 ($\mathbb{F}_{q_0}$-libre de suma-producto)**.** Sea $\mathbb{F}_q/\mathbb{F}_{q_0}$ una extensión de cuerpos. Un vector $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_\ell] \in \mathbb{F}_q^\ell$ se llama $\mathbb{F}_{q_0}$*-libre de suma-producto* si

$$\sum_{\substack{S \subseteq \{1,\ldots,\ell\} \\ S \neq \emptyset}} a_S \prod_{i \in S} \eta_i \notin \mathbb{F}_{q_0}^* \qquad \forall\, a_S \in \mathbb{F}_{q_0}.$$

Ejemplo 3.18. Sea $\mathbb{F}_q = \mathbb{F}_8 = \mathbb{F}_2[x]/(x^3 + x + 1)$ y $\mathbb{F}_{q_0} = \mathbb{F}_2$, de modo que $\mathbb{F}_8/\mathbb{F}_2$ es la extensión de cuerpos considerada, con $\mathbb{F}_2^* = \{1\}$. Tomemos $\ell = 2$ y el vector

$$\boldsymbol{\eta} = [\eta_1, \eta_2] = [x + 1,\ x + 1] \in \mathbb{F}_8^2.$$

Los subconjuntos no vacíos de $\{1, 2\}$ son $S \in \{\{1\}, \{2\}, \{1, 2\}\}$, por lo que hay $2^3 - 1 = 7$ combinaciones no triviales de escalares $(a_1, a_2, a_{12}) \in \mathbb{F}_2^3$ a verificar en

$$a_1\eta_1 + a_2\eta_2 + a_{12}\,\eta_1\eta_2 \notin \mathbb{F}_2^*.$$

Calculando cada caso, usando que $\eta_1 = \eta_2 = x + 1$ y que en $\mathbb{F}_8$ se cumple $x^3 = x + 1$, se obtiene:

a_1	a_2	a_{12}	Valor de la suma
0	0	1	$\eta_1\eta_2 = x^2 + 1$
0	1	0	$\eta_2 = x + 1$
0	1	1	$\eta_2 + \eta_1\eta_2 = x^2 + x$
1	0	0	$\eta_1 = x + 1$
1	0	1	$\eta_1 + \eta_1\eta_2 = x^2 + x$
1	1	0	$\eta_1 + \eta_2 = 0$
1	1	1	$\eta_1 + \eta_2 + \eta_1\eta_2 = x^2 + 1$

En ninguno de los siete casos el resultado es igual a 1, el único elemento de $\mathbb{F}_2^*$. Por lo tanto, $\boldsymbol{\eta} = [x + 1,\ x + 1]$ es $\mathbb{F}_2$-libre de suma-producto.

Observación 3.19. De manera equivalente, η es $\mathbb{F}_{q_0}$-libre de suma-producto si y solo si no existe ningún polinomio $f \in \mathbb{F}_{q_0}[X_1, \ldots, X_\ell]$ con coeficiente constante distinto de cero y de grado a lo sumo 1 en cada variable X_i, tal que $f(\eta_1, \ldots, \eta_\ell) = 0$. Esta formulación será la que se use directamente en la demostración de la Proposición 3.23.

Proposición 3.20. *Un vector $\boldsymbol{\eta}$ es $\mathbb{F}_{q_0}$-libre de suma-producto si y solo si no existe un polinomio $f \in \mathbb{F}_{q_0}[X_1, \ldots, X_\ell]$ con término constante no nulo y de grado a lo sumo* 1 *en cada X_i tal que $f(\eta_1, \ldots, \eta_\ell) = 0$.*

Demostración. Todo polinomio $f \in \mathbb{F}_{q_0}[X_1, \ldots, X_\ell]$ de grado a lo sumo 1 en cada variable

se escribe de forma única como

$$f(X_1,\dots,X_\ell) = c_0 + \sum_{\substack{S\subseteq\{1,\dots,\ell\}\\ S\neq\emptyset}} a_S \prod_{i\in S} X_i, \qquad c_0, a_S \in \mathbb{F}_{q_0}, \tag{3.3}$$

pues los monomios $\prod_{i\in S} X_i$, para $S \subseteq \{1,\dots,\ell\}$ (incluyendo $S=\emptyset$, que da el término constante), forman una base del espacio de polinomios multilineales en ℓ variables. Evaluando (3.3) en $\boldsymbol{\eta}$ se obtiene

$$f(\boldsymbol{\eta}) = c_0 + \sum_{\substack{S\subseteq\{1,\dots,\ell\}\\ S\neq\emptyset}} a_S \prod_{i\in S} \eta_i. \tag{3.4}$$

(⇐) Si $\boldsymbol{\eta}$ no es libre de suma-producto, existe f. Supongamos que $\boldsymbol{\eta}$ no satisface la condición de la Definición 3.17, es decir, existen escalares $a_S \in \mathbb{F}_{q_0}$ tales que

$$\sum_{S\neq\emptyset} a_S \prod_{i\in S} \eta_i = c \in \mathbb{F}_{q_0}^*.$$

Definimos $c_0 := -c$. Como $\mathbb{F}_{q_0}$ es un cuerpo y $c \neq 0$, se tiene $c_0 \in \mathbb{F}_{q_0}^*$. Construimos

$$f(X_1,\dots,X_\ell) = c_0 + \sum_{S\neq\emptyset} a_S \prod_{i\in S} X_i,$$

que tiene coeficientes en $\mathbb{F}_{q_0}$, grado a lo sumo 1 en cada variable y término constante $c_0 \neq 0$. Evaluando en $\boldsymbol{\eta}$ mediante (3.4):

$$f(\boldsymbol{\eta}) = c_0 + c = -c + c = 0.$$

Así, f es el polinomio buscado.

(⇒) Si existe f, entonces $\boldsymbol{\eta}$ no es libre de suma-producto. Supongamos que existe f como en (3.3), con $c_0 \in \mathbb{F}_{q_0}^*$ y $f(\boldsymbol{\eta}) = 0$. De (3.4):

$$0 = c_0 + \sum_{S\neq\emptyset} a_S \prod_{i\in S} \eta_i \implies \sum_{S\neq\emptyset} a_S \prod_{i\in S} \eta_i = -c_0.$$

Como $c_0 \in \mathbb{F}_{q_0}^*$, también $-c_0 \in \mathbb{F}_{q_0}^*$ (el inverso aditivo de un elemento no nulo de un cuerpo sigue siendo no nulo). Por lo tanto, encontramos escalares $a_S \in \mathbb{F}_{q_0}$ tales que la suma cae exactamente en $\mathbb{F}_{q_0}^*$, contradiciendo la hipótesis de que $\boldsymbol{\eta}$ es libre de suma-producto.

Ambas direcciones establecen la equivalencia enunciada. ■ □

Lema 3.21 (Multilinealidad del determinante)**.** *Sea*

$$M = \begin{pmatrix} R_1 \\ \vdots \\ R_k \end{pmatrix},$$

donde $R_1,\dots,R_k$ son las filas de M. Entonces el determinante es lineal respecto de cada fila cuando las demás permanecen fijas. Es decir, si para algún índice i

$$R_i = A_i + B_i,$$

entonces

$$\det(R_1, \ldots, R_i, \ldots, R_k) = \det(R_1, \ldots, A_i, \ldots, R_k) + \det(R_1, \ldots, B_i, \ldots, R_k),$$

y para todo $\lambda \in \mathbb{F}_q$,

$$\det(R_1, \ldots, \lambda A_i, \ldots, R_k) = \lambda \det(R_1, \ldots, A_i, \ldots, R_k).$$

En consecuencia, el determinante es una aplicación multilineal respecto de las filas de la matriz.

Demostración. La linealidad del determinante respecto de cada fila es una propiedad clásica de la función determinante y puede demostrarse a partir de la fórmula de Leibniz o de la caracterización axiomática del determinante como la única aplicación multilineal alternante normalizada. Véase, por ejemplo, Hoffman y Kunze [6, Capítulo 5].

Aplicando sucesivamente esta propiedad a una fila fija se obtienen inmediatamente las igualdades anteriores. □

Corolario 3.22. *Sea* M *una matriz cuyas filas pueden escribirse como*

$$R_i = A_i + \sum_{\substack{j=1 \\ h_j = i}}^{\ell} \eta_j B_j, \qquad i = 1, \ldots, k,$$

donde los vectores A_i *y* B_j *tienen entradas en* $\mathbb{F}_{q_0}$ *y los parámetros* $\eta_1, \ldots, \eta_\ell \in \mathbb{F}_q$. *Entonces*

$$\det(M) = \sum_{I \subseteq \{1, \ldots, \ell\}} c_I \prod_{j \in I} \eta_j,$$

donde cada coeficiente $c_I \in \mathbb{F}_{q_0}$.

En particular, el determinante es un polinomio en $\eta_1, \ldots, \eta_\ell$ *de grado a lo sumo uno en cada variable.*

Demostración. Por el Lema 3.21, el determinante es lineal respecto de cada fila.

Al expandir la primera fila,

$$R_1 = A_1 + \sum_{h_j = 1} \eta_j B_j,$$

el determinante se escribe como una suma de determinantes, cada uno de los cuales contiene únicamente uno de los términos de la descomposición anterior en la primera fila.

Repitiendo este procedimiento sucesivamente sobre las filas $R_2, \ldots, R_k$, cada término obtenido corresponde a escoger, para cada fila, uno de los sumandos que aparecen en su descomposición.

Por tanto, el determinante puede escribirse como

$$\det(M) = \sum_{I \subseteq \{1,\dots,\ell\}} c_I \prod_{j \in I} \eta_j,$$

donde el coeficiente c_I es el determinante de la matriz obtenida al sustituir, en cada fila, el término correspondiente a la elección realizada durante la expansión. Como todas las entradas de los vectores A_i y B_j pertenecen a $\mathbb{F}_{q_0}$, se tiene

$$c_I \in \mathbb{F}_{q_0}.$$

Finalmente, dado que cada parámetro η_j aparece únicamente en una fila, ningún monomio puede contener una potencia superior a uno de una misma variable. En consecuencia,

$$\deg_{\eta_j}(\det(M)) \leq 1, \qquad j = 1, \dots, \ell.$$

□

Proposición 3.23. *Sea $\mathbb{F}_q/\mathbb{F}_{q_0}$ una extensión de cuerpos finitos. Sea $k < n \leq q_0$ y sean $\alpha_1, \dots, \alpha_n \in \mathbb{F}_{q_0}$ distintos. Para cualquier elección de t, h, y cualquier $\eta \in \mathbb{F}_q^\ell$ tal que η sea $\mathbb{F}_{q_0}$-libre de suma-producto, el código Reed–Solomon torcido $C^{n,k}_{\alpha,t,h,\eta}$ es MDS.*

Demostración. La demostración procede en cinco pasos.

Paso 1: Las entradas de G son polinomios en $\eta_1, \dots, \eta_\ell$ con coeficientes en $\mathbb{F}_{q_0}$.

Sea G la matriz generadora de $C^{n,k}_{\alpha,t,h,\eta}$ dada en (3.2). La entrada de la fila i y la columna m está dada por

$$g_i(\alpha_m) = \alpha_m^i + \sum_{\substack{j=1 \\ h_j = i}}^{\ell} \eta_j \alpha_m^{k-1+t_j}.$$

Como $\alpha_m \in \mathbb{F}_{q_0}$ para todo $m = 1, \dots, n$, se tiene que

$$\alpha_m^i,\ \alpha_m^{k-1+t_j} \in \mathbb{F}_{q_0}.$$

Por consiguiente, cada entrada de G es un polinomio en las variables $\eta_1, \dots, \eta_\ell$ con coeficientes en $\mathbb{F}_{q_0}$. Además, cada parámetro η_j aparece únicamente de forma lineal, pues siempre está multiplicado por el escalar fijo $\alpha_m^{k-1+t_j}$.

Paso 2: Cada η_j aparece en exactamente una fila de G, y todo menor $k \times k$ es un polinomio de grado a lo sumo uno en cada variable.

Sea M un menor $k \times k$ cualquiera de la matriz generadora G, obtenido al seleccionar arbitrariamente k columnas de G.

Denotemos por

$$R_1, \dots, R_k$$

las filas de M.

Por la definición de los polinomios torcidos, cada fila puede escribirse como

$$R_i = A_i + \sum_{\substack{j=1 \\ h_j=i}}^{\ell} \eta_j B_j,$$

donde

$$A_i = \left(\alpha_{m_1}^i, \ldots, \alpha_{m_k}^i\right)$$

es la parte correspondiente al código Reed–Solomon clásico, y

$$B_j = \left(\alpha_{m_1}^{k-1+t_j}, \ldots, \alpha_{m_k}^{k-1+t_j}\right)$$

es el vector asociado a la torsión j.

Obsérvese que tanto A_i como B_j tienen todas sus entradas en $\mathbb{F}_{q_0}$.

Además, por la definición de la función de enganche $h : \{1, \ldots, \ell\} \to \{0, \ldots, k-1\}$, cada parámetro η_j aparece únicamente en la fila cuyo índice satisface $h_j = i$.

Aplicando el Lema 3.21 de manera sucesiva sobre las filas de M, se obtiene una expansión del determinante en la cual, para cada fila, se escoge exactamente uno de los sumandos de su descomposición. En consecuencia, por el Corolario 3.22,

$$\det(M) = \sum_{I \subseteq \{1,\ldots,\ell\}} c_I \prod_{j \in I} \eta_j,$$

donde

$$c_I \in \mathbb{F}_{q_0}.$$

Como cada variable η_j interviene únicamente en una fila de M, ningún monomio de la expansión puede contener una potencia superior a uno de una misma variable. Por tanto,

$$\deg_{\eta_j}(\det(M)) \leq 1, \qquad j = 1, \ldots, \ell.$$

Paso 3: El término constante de cada menor es un elemento no nulo de $\mathbb{F}_{q_0}^*$.

Sea

$$\det(M) = \sum_{I \subseteq \{1,\ldots,\ell\}} c_I \prod_{j \in I} \eta_j$$

la expansión obtenida en el Paso 2.

El término constante corresponde al subconjunto vacío $I = \varnothing$, es decir,

$$c_\varnothing = \det(M)\Big|_{\eta_1 = \cdots = \eta_\ell = 0}.$$

Al evaluar

$$\eta_1 = \cdots = \eta_\ell = 0,$$

cada fila de M se reduce a

$$R_i = A_i,$$

por lo que el menor considerado coincide con el menor correspondiente de la matriz generadora del código Reed–Solomon clásico.

En efecto, la Observación 3.11 establece que, al anular todas las torsiones, el espacio polinomial torcido coincide con el espacio de polinomios del código Reed–Solomon clásico. En consecuencia, la matriz generadora se convierte en una matriz de Vandermonde evaluada en los puntos distintos

$$\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q_0}.$$

Como el código Reed–Solomon clásico con $n \leq q_0$ es MDS, todo menor $k \times k$ de dicha matriz es no nulo. Por tanto,

$$c_\varnothing \in \mathbb{F}_{q_0}^*.$$

Paso 4: La condición libre de suma-producto implica que cada menor es distinto de cero.

Por los Pasos 1, 2 y 3, todo menor $k \times k$ de G puede escribirse como un polinomio

$$f(\eta_1, \ldots, \eta_\ell) = \sum_{I \subseteq \{1,\ldots,\ell\}} c_I \prod_{j \in I} \eta_j,$$

cuyos coeficientes pertenecen a $\mathbb{F}_{q_0}$, cuyo grado respecto de cada variable η_j es a lo sumo uno y cuyo término constante es un elemento no nulo de $\mathbb{F}_{q_0}^*$.

Por la formulación equivalente de la Proposición 3.20,

la condición de que $\eta = (\eta_1, \ldots, \eta_\ell)$ sea $\mathbb{F}_{q_0}$-libre de suma-producto implica que ningún polinomio con estas propiedades puede anularse al evaluarse en $(\eta_1, \ldots, \eta_\ell)$.

En consecuencia,

$$\det(M) \neq 0.$$

Como el menor M fue arbitrario, se concluye que todo menor $k \times k$ de G es no singular.

Paso 5: Conclusión.

Puesto que todo menor $k \times k$ de la matriz generadora G es distinto de cero, cualquier conjunto de k columnas de G es linealmente independiente.

Por la Observación 3.16, esto equivale a que la distancia mínima del código es

$$d = n - k + 1.$$

Por consiguiente,

$$C^{n,k}_{\alpha,t,h,\eta}$$

es un código MDS.

□

Observación 3.24. Precisamos el rol de cada hipótesis de la Proposición 3.23:

- La condición $\alpha_i \in \mathbb{F}_{q_0}$ garantiza que las entradas de G sean polinomios en los η_j con coeficientes en el subcuerpo base, lo que permite invocar la condición libre de suma-producto.
- La condición $n \leq q_0$ garantiza que haya suficientes puntos distintos en $\mathbb{F}_{q_0}$ para que el código RS clásico de referencia (con $\eta = 0$) sea MDS, asegurando así que el término constante de cada menor sea no nulo.
- La condición de que η sea $\mathbb{F}_{q_0}$-libre de suma-producto es la que garantiza que los menores no se anulen al sustituir los valores concretos de $\eta_1, \ldots, \eta_\ell$.

Ejemplo 3.25. Para ilustrar la Proposición 3.23, se implementó en `SageMath` el siguiente código, que compara dos elecciones de η manteniendo fija la hipótesis $n \leq q_0$ sobre la extensión $\mathbb{F}_9/\mathbb{F}_3$.

```
q0 = 3
q = 9
F3 = GF(q0)
F9 = GF(q, 'x')
x = F9.gen()

k = 2
n = 3                          # n <= q0 = 3: hipotesis respetada en TODO momento
alphas = [F9(a) for a in F3]   # puntos de evaluacion distintos en F3

def es_F3_libre_suma_producto_l1(eta):
    F3star = [F3(1), F3(2)]
    for a1 in F3star:
        if F3(a1) * eta in F3star:
            return False
    return True

def evalua_torcido(f0, f1, eta, alpha):
    return f0 + f1*alpha + eta*f0*alpha^2

def construir_matriz_generadora(alphas, eta):
    fila_f0 = [evalua_torcido(F9(1), F9(0), eta, a) for a in alphas]
```

```
    fila_f1 = [evalua_torcido(F9(0), F9(1), eta, a) for a in
   alphas]
    return matrix(F9, [fila_f0, fila_f1])

def es_MDS(G, n, k):
    d = n - k + 1
    for cols in Combinations(range(n), k):
        M = G.matrix_from_columns(cols)
        if M.is_singular():
            return False, d
    return True, d

# Caso 1: eta = x, ES F3-libre de suma-producto
eta_libre = x
print("eta =", eta_libre, "-> libre de suma-producto:",
      es_F3_libre_suma_producto_l1(eta_libre))
G1 = construir_matriz_generadora(alphas, eta_libre)
mds1, d1 = es_MDS(G1, n, k)
print("Matriz generadora:\n", G1)
print("d teorica MDS (n-k+1):", d1, " | Es MDS?", mds1)

print("\n" + "="*50 + "\n")

# Caso 2: eta = 2, NO es F3-libre de suma-producto
eta_no_libre = F9(2)
print("eta =", eta_no_libre, "-> libre de suma-producto:",
      es_F3_libre_suma_producto_l1(eta_no_libre))
G2 = construir_matriz_generadora(alphas, eta_no_libre)
mds2, d2 = es_MDS(G2, n, k)
print("Matriz generadora:\n", G2)
print("d teorica MDS (n-k+1):", d2, " | Es MDS?", mds2)
```

Listado 3.7: Verificación de la Proposición 3.25 en $\mathbb{F}_9/\mathbb{F}_3$

La ejecución del código anterior produce la siguiente salida:

```
eta = x -> libre de suma-producto: True
Matriz generadora:
 [    1 x + 1 x + 1]
[    0     1     2]
d teorica MDS (n-k+1): 2  | Es MDS? True

==================================================

eta = 2 -> libre de suma-producto: False
Matriz generadora:
```

```
 [1 0 0]
[0 1 2]
d teorica MDS (n-k+1): 2  | Es MDS? False
```

Listado 3.8: Salida obtenida al ejecutar el Código 3.7

Como se observa, con $\eta = x$ (que sí es $\mathbb{F}_3$-libre de suma-producto) el código resulta MDS, mientras que con $\eta = 2$ (que no lo es) el código pierde dicha propiedad.

3.2.2. Dos Construcciones de Conjuntos $\mathbb{F}_{q_0}$-Libres de Suma-Producto

La Proposición 3.23 garantiza que si η es $\mathbb{F}_{q_0}$-libre de suma-producto entonces el código RS torcido es MDS, pero no dice cómo construir vectores η con esa propiedad. Esta subsección presenta dos construcciones explícitas, cada una basada en una estructura algebraica distinta.

Proposición 3.26. *Sea $\mathbb{F}_{q_0} \subsetneq \mathbb{F}_{q_1} \subsetneq \cdots \subsetneq \mathbb{F}_{q_\ell} \subsetneq \mathbb{F}_q$ una cadena propia de subcampos. Sea $\eta_i \in \mathbb{F}_{q_i} \setminus \mathbb{F}_{q_{i-1}}$. Entonces $\eta = [\eta_1, \ldots, \eta_\ell]$ es $\mathbb{F}_{q_0}$-libre de suma-producto.*

Demostración. Procedemos por inducción sobre ℓ.

Caso base: $\ell = 1$**.** Tenemos $\eta_1 \in \mathbb{F}_{q_1} \setminus \mathbb{F}_{q_0}$. El único subconjunto no vacío de $\{1\}$ es $S = \{1\}$, y la condición libre de suma-producto exige $\eta_1 \notin \mathbb{F}_{q_0}^*$, lo cual se satisface inmediatamente ya que $\eta_1 \notin \mathbb{F}_{q_0}$ por hipótesis.

Paso inductivo. Supongamos que el resultado es cierto para $\ell - 1$ torsiones: el vector $[\eta_1, \ldots, \eta_{\ell-1}]$, con $\eta_i \in \mathbb{F}_{q_i} \setminus \mathbb{F}_{q_{i-1}}$, es $\mathbb{F}_{q_0}$-libre de suma-producto.

Sea A una suma-producto sobre los ℓ elementos $\eta_1, \ldots, \eta_\ell$. Separando los subconjuntos según si contienen o no al índice ℓ, obtenemos la descomposición

$$A = a(\eta_1, \ldots, \eta_{\ell-1}) + \eta_\ell \, b(\eta_1, \ldots, \eta_{\ell-1}),$$

donde $a, b \in \mathbb{F}_{q_0}[X_1, \ldots, X_{\ell-1}]$ son polinomios de grado a lo sumo 1 en cada variable y con términos constantes distintos de cero.

Si $b(\eta_1, \ldots, \eta_{\ell-1}) = 0$, entonces $A = a(\eta_1, \ldots, \eta_{\ell-1}) \notin \mathbb{F}_{q_0}^*$ por la hipótesis inductiva, puesto que a tiene término constante no nulo.

En otro caso, supóngase por contradicción que $A \in \mathbb{F}_{q_0}^*$. Como $a(\eta_1, \ldots, \eta_{\ell-1})$ y $b(\eta_1, \ldots, \eta_{\ell-1})$ pertenecen a $\mathbb{F}_{q_{\ell-1}}$ (pues los η_i están en $\mathbb{F}_{q_{\ell-1}}$ y los coeficientes en $\mathbb{F}_{q_0}$), y $A \in \mathbb{F}_{q_0} \subseteq \mathbb{F}_{q_{\ell-1}}$, tendríamos

$$\eta_\ell = \frac{A - a(\eta_1, \ldots, \eta_{\ell-1})}{b(\eta_1, \ldots, \eta_{\ell-1})} \in \mathbb{F}_{q_{\ell-1}},$$

contradiciendo la hipótesis $\eta_\ell \in \mathbb{F}_{q_\ell} \setminus \mathbb{F}_{q_{\ell-1}}$. En consecuencia $A \notin \mathbb{F}_{q_0}^*$, y como A era una suma-producto arbitraria, concluimos que η es $\mathbb{F}_{q_0}$-libre de suma-producto. □

Observación 3.27. La condición $\eta_i \in \mathbb{F}_{q_i} \setminus \mathbb{F}_{q_{i-1}}$ exige que cada η_i sea un elemento *genuinamente nuevo* que aparece en el i-ésimo cuerpo de la cadena pero no estaba en

el anterior. Esto garantiza que η_i no puede expresarse como combinación algebraica de los $\eta_1, \ldots, \eta_{i-1}$ con coeficientes en $\mathbb{F}_{q_0}$, que es precisamente lo que impide que ninguna suma-producto parcial caiga en $\mathbb{F}_{q_0}^*$.

Proposición 3.28. *Sea $\mathbb{F}_q/\mathbb{F}_{q_0}$ una extensión de cuerpos finitos de grado al menos $\ell+1 \geq 2$, y sea $1, \psi, \ldots, \psi^{[\mathbb{F}_q:\mathbb{F}_{q_0}]-1} \in \mathbb{F}_q$ una base de potencias de la extensión. Entonces cualquier $\boldsymbol{\eta} := [a_1\psi, \ldots, a_\ell\psi]$ con $a_i \in \mathbb{F}_{q_0} \setminus \{0\}$ es $\mathbb{F}_{q_0}$-libre de suma-producto.*

Demostración. **Paso 1** Cada coordenada tiene la forma $\eta_i = a_i\psi$, donde $a_i \in \mathbb{F}_{q_0} \setminus \{0\}$ y ψ es el mismo elemento primitivo de la base de potencias para todas las coordenadas.

Paso 2 Para cualquier subconjunto no vacío $\mathcal{I} \subseteq \{1, \ldots, \ell\}$, usando la conmutatividad del producto en $\mathbb{F}_q$:

$$\prod_{i\in\mathcal{I}} \eta_i = \prod_{i\in\mathcal{I}} (a_i\psi) = \Big(\prod_{i\in\mathcal{I}} a_i\Big) \cdot \underbrace{\psi \cdot \psi \cdots \psi}_{|\mathcal{I}| \text{ veces}} = b\,\psi^{|\mathcal{I}|},$$

donde $b := \prod_{i\in\mathcal{I}} a_i \in \mathbb{F}_{q_0} \setminus \{0\}$, ya que $\mathbb{F}_{q_0}$ es cerrado bajo producto y no tiene divisores de cero.

Paso 3 Toda combinación $\mathbb{F}_{q_0}$-lineal $\sum_{\mathcal{I}\neq\emptyset} c_{\mathcal{I}} \prod_{i\in\mathcal{I}} \eta_i$ se reescribe, usando el Paso 2 y agrupando por tamaño $k = |\mathcal{I}|$, como

$$\sum_{\mathcal{I}\neq\emptyset} c_{\mathcal{I}} \prod_{i\in\mathcal{I}} \eta_i = b_1\psi + b_2\psi^2 + \cdots + b_\ell\psi^\ell, \qquad b_k \in \mathbb{F}_{q_0},$$

donde cada b_k es la suma de las contribuciones $c_{\mathcal{I}} \cdot b_{\mathcal{I}}$ de todos los subconjuntos de tamaño k.

Paso 4 Como $1, \psi, \ldots, \psi^{[\mathbb{F}_q:\mathbb{F}_{q_0}]-1}$ es una base de $\mathbb{F}_q$ sobre $\mathbb{F}_{q_0}$ y el grado de la extensión es al menos $\ell + 1$, los elementos $1, \psi, \ldots, \psi^\ell$ son linealmente independientes sobre $\mathbb{F}_{q_0}$. Si la suma del Paso 3 fuera igual a algún $c \in \mathbb{F}_{q_0}^*$, tendríamos

$$c \cdot 1 - b_1\psi - b_2\psi^2 - \cdots - b_\ell\psi^\ell = 0,$$

una combinación lineal no trivial (pues $c \neq 0$) de $1, \psi, \ldots, \psi^\ell$ igualada a cero, lo cual contradice su independencia lineal. Por lo tanto, la suma nunca puede caer en $\mathbb{F}_{q_0}^*$, quedando demostrado que $\boldsymbol{\eta}$ es $\mathbb{F}_{q_0}$-libre de suma-producto. ■ □

Ejemplo 3.29. Sea $\mathbb{F}_q/\mathbb{F}_{q_0} = \mathbb{F}_{16}/\mathbb{F}_2$, con $\mathbb{F}_{16} = \mathbb{F}_2[x]/(x^4 + x + 1)$, de grado 4 sobre $\mathbb{F}_2$. Tomamos $\ell = 3$ (pues $\ell + 1 = 4 \leq 4$), $\psi = x$, y $a_1 = a_2 = a_3 = 1$ (la única opción no nula en $\mathbb{F}_2$), de modo que

$$\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3] = [x,\, x,\, x].$$

Paso 1: verificar la fórmula del producto. Para cada subconjunto no vacío $\mathcal{I} \subseteq \{1, 2, 3\}$, calculamos $\prod_{i\in\mathcal{I}} \eta_i$ y lo comparamos con $\psi^{|\mathcal{I}|} = x^{|\mathcal{I}|}$:

$$\eta_1 = \eta_2 = \eta_3 = x, \qquad \eta_1\eta_2 = \eta_1\eta_3 = \eta_2\eta_3 = x^2, \qquad \eta_1\eta_2\eta_3 = x^3.$$

En los siete casos, el producto coincide exactamente con $x^{|\mathcal{I}|}$, confirmando la fórmula del Paso 2 de la demostración general.

Paso 2: probar tres combinaciones de coeficientes.

Caso A ($c_1 = 1$, resto 0):

$$\sum c_{\mathcal{I}} \prod_{i\in\mathcal{I}} \eta_i = \eta_1 = x.$$

Aquí $b_1 = a_1c_1 = 1$ y $b_2 = b_3 = 0$, así que la reconstrucción $b_1\psi + b_2\psi^2 + b_3\psi^3 = x$, coincidiendo exactamente.

Caso B ($c_1 = c_2 = c_{12} = 1$, resto 0):

$$\sum c_{\mathcal{I}} \prod_{i\in\mathcal{I}} \eta_i = \eta_1 + \eta_2 + \eta_1\eta_2 = x + x + x^2 = x^2,$$

usando que $x + x = 0$ en $\mathbb{F}_2$. Aquí $b_1 = a_1c_1 + a_2c_2 = 1 + 1 = 0$ y $b_2 = a_1a_2c_{12} = 1$, así que la reconstrucción da $b_1\psi + b_2\psi^2 = 0 \cdot x + 1 \cdot x^2 = x^2$, coincidiendo.

Caso C (los siete coeficientes $c_{\mathcal{I}} = 1$):

$$\sum c_{\mathcal{I}} \prod_{i\in\mathcal{I}} \eta_i = 3\eta_1 \dot{+} 3\eta_1\eta_2 \dot{+} \eta_1\eta_2\eta_3 = x + x^2 + x^3,$$

donde $b_1 = a_1c_1+a_2c_2+a_3c_3 = 1+1+1 = 1$, $b_2 = a_1a_2c_{12}+a_1a_3c_{13}+a_2a_3c_{23} = 1+1+1 = 1$ y $b_3 = a_1a_2a_3c_{123} = 1$ (todas las sumas en $\mathbb{F}_2$). La reconstrucción $b_1\psi+b_2\psi^2+b_3\psi^3 = x+x^2+x^3$ coincide exactamente con el cálculo directo.

Paso 3: verificar que ninguna suma cae en $\mathbb{F}_2^*$.

Caso	Suma resultante	¿Es igual a 1?
A	x	No
B	x^2	No
C	$x + x^2 + x^3$	No

En ninguno de los tres casos (ni en ninguna de las $2^7 - 1 = 127$ combinaciones no triviales posibles de coeficientes) la suma resultante es igual a 1, el único elemento de $\mathbb{F}_2^*$. Esto ilustra que, como $1, x, x^2, x^3$ son linealmente independientes sobre $\mathbb{F}_2$, ninguna combinación de la forma $b_1x + b_2x^2 + b_3x^3$ (sin término constante) puede coincidir con un elemento no nulo de $\mathbb{F}_2$, confirmando que $\boldsymbol{\eta} = [x, x, x]$ es $\mathbb{F}_2$-libre de suma-producto.

Observación 3.30. Las proposiciones anteriores proporcionan dos construcciones de códigos Reed–Solomon torcidos MDS. En ambos casos, es necesario que los puntos de evaluación pertenezcan a un subcampo propio del campo base del código. Además, si $n \leq q_0$, donde q_0 es el menor número primo mayor o igual que n, el postulado de Bertrand garantiza que $q_0 < 2n$. En consecuencia, los menores tamaños de campo q para los cuales las construcciones son válidas satisfacen

$$n^{2^\ell} \leq q = q_0^{2^\ell} < (2n)^{2^\ell}, \qquad \text{(Proposición 3.26)}, \tag{3.5}$$
$$n^{\ell+1} \leq q = q_0^{\ell+1} < (2n)^{\ell+1}, \qquad \text{(Proposición 3.28)}. \tag{3.6}$$

Para estos tamaños mínimos del campo, se obtiene además que

$$\frac{1}{2}q^{2^{-\ell}} < n \leq q^{2^{-\ell}}, \qquad \text{(Proposición 3.26)}, \tag{3.7}$$
$$\frac{1}{2}q^{\frac{1}{\ell+1}} < n \leq q^{\frac{1}{\ell+1}}, \qquad \text{(Proposición 3.28)}. \tag{3.8}$$

Estas desigualdades muestran que, al elegir el menor tamaño de campo permitido por cada construcción, la longitud n del código queda acotada inferior y superiormente por potencias de q, de acuerdo con la estructura particular de cada una de las dos construcciones.

Observación 3.31. En comparación con la Proposición 3.26, la construcción de la Proposición 3.28 puede proporcionar tamaños de campo más pequeños para una longitud de código dada, lo que la hace más eficiente en la práctica. Sin embargo, la Proposición 3.26 se incluye porque admite un análogo en la teoría de códigos de Gabidulin torcidos en métrica de rango, para el cual no se conoce actualmente un análogo de la Proposición 3.28.

3.2.3. Códigos RS Torcidos con Una Torsión

La condición de ser $\mathbb{F}_{q_0}$-libre de suma-producto produce códigos MDS para cualquier elección de los parámetros t y h, pero es una restricción bastante fuerte sobre los η_i y produce códigos relativamente cortos. En esta y la siguiente subsección se obtienen códigos MDS más largos considerando elecciones específicas de t y h.

Comenzamos con el caso de una sola torsión $\ell = 1$, enganche $h = 0$ y torsión $t = 1$. Con estos parámetros, el único polinomio base que recibe un término torcido es g_0, y todo polinomio del espacio torcido tiene la forma

$$f = \sum_{i=0}^{k-1} f_i X^i + \eta_1 f_0 X^k.$$

Para este caso específico es posible obtener una condición que sea simultáneamente necesaria y suficiente para que el código sea MDS.

Lema 3.32. *Sea $\ell = 1$ y sean n, k, α, η elegidos como en la Definición 3.1. El código $C^{n,k}_{\alpha,1,0,\eta}$ es MDS si y solo si*

$$\eta_1(-1)^k \prod_{i \in I} \alpha_i \neq 1 \qquad \forall\, I \subseteq \{1, \ldots, n\},\ |I| = k. \tag{3.9}$$

Demostración. La demostración se basa en caracterizar cuándo existe una palabra código de peso menor que $n - k + 1$.

Forma de los polinomios de evaluación. Todo polinomio del espacio torcido $\mathcal{P}_{1,0,\eta}^{n,k}$ tiene la forma

$$f = \sum_{i=0}^{k-1} f_i X^i + \eta_1 f_0 X^k.$$

Para que el código no sea MDS, debe existir una palabra código no nula con peso menor que $n-k+1$, es decir, debe existir $f \neq 0$ en $\mathcal{P}_{1,0,\eta}^{n,k}$ con al menos k coordenadas nulas, lo que equivale a que f tenga al menos k raíces entre los α_i.

Caso $f_0 = 0$. Si $f_0 = 0$, el término torcido $\eta_1 f_0 X^k$ desaparece y f se convierte en un polinomio de grado estrictamente menor que k. Un polinomio no nulo de grado menor que k tiene a lo sumo $k-1$ raíces, por lo que la palabra código correspondiente tiene peso al menos $n-(k-1) = n-k+1$. Estos polinomios no afectan la propiedad MDS.

Caso $f_0 \neq 0$. Si $f_0 \neq 0$ y $\eta_1 \neq 0$, entonces f tiene grado exactamente k con coeficiente líder $\eta_1 f_0 \neq 0$. Para que f tenga exactamente k raíces entre los α_i, debe existir $I \subseteq \{1,\ldots,n\}$ con $|I| = k$ tal que

$$f = \eta_1 f_0 \prod_{i \in I} (X - \alpha_i).$$

Esta factorización es posible porque un polinomio de grado k queda completamente determinado por su coeficiente líder y sus k raíces. Comparando el término constante en ambas expresiones de f:

$$\begin{aligned} f(0) &= f_0 \qquad \text{(de la definición del espacio torcido)}, \\ f(0) &= \eta_1 f_0 \prod_{i \in I} (0 - \alpha_i) = \eta_1 f_0 \, (-1)^k \prod_{i \in I} \alpha_i \qquad \text{(de la factorización)}. \end{aligned}$$

Igualando y dividiendo por $f_0 \neq 0$:

$$1 = \eta_1 (-1)^k \prod_{i \in I} \alpha_i.$$

Conclusión. Hemos demostrado que el código *no* es MDS si y solo si existe $I \subseteq \{1,\ldots,n\}$ con $|I| = k$ tal que $\eta_1 (-1)^k \prod_{i \in I} \alpha_i = 1$. Tomando la negación, el código es MDS si y solo si se satisface la condición (3.9). □

Observación 3.33. Para $\eta_1 \neq 0$, una condición suficiente para que se cumpla (3.9) es que el elemento $(-1)^k \eta_1^{-1}$ no pertenezca al grupo multiplicativo generado por los α_i en $\mathbb{F}_q^*$. En efecto, si $(-1)^k \eta_1^{-1}$ no es producto de k elementos de $\{\alpha_i\}$, entonces la igualdad $\eta_1 (-1)^k \prod_{i \in I} \alpha_i = 1$ no puede sostenerse para ningún I con $|I| = k$.

Ejemplo 3.34 (Verificación del Lema 3.32)**.** Trabajamos sobre $\mathbb{F}_7$ con parámetros $n = 3$, $k = 2$, $\ell = 1$, $h = 0$, $t = 1$, y puntos de evaluación

$$\alpha_1 = 1, \quad \alpha_2 = 2, \quad \alpha_3 = 3.$$

Con estos parámetros, todo polinomio del espacio torcido tiene la forma

$$f = f_0 + f_1 X + \eta_1 f_0 X^2,$$

y la condición (3.9) se convierte en

$$\eta_1(-1)^2 \prod_{i\in I} \alpha_i \neq 1, \quad \text{es decir,} \quad \eta_1 \prod_{i\in I} \alpha_i \neq 1,$$

para todo $I \subseteq \{1,2,3\}$ con $|I| = 2$. Los tres subconjuntos posibles y sus productos son:

I	$\prod_{i\in I} \alpha_i$	Valor en $\mathbb{F}_7$
$\{1,2\}$	$\alpha_1\alpha_2 = 1 \cdot 2$	2
$\{1,3\}$	$\alpha_1\alpha_3 = 1 \cdot 3$	3
$\{2,3\}$	$\alpha_2\alpha_3 = 2 \cdot 3$	6

Caso 1: $\eta_1 = 1$ **(condición satisfecha ⇒ MDS).**

Verificamos la condición para cada I:

$$\begin{aligned} I = \{1,2\}: &\quad \eta_1 \cdot 2 = 1 \cdot 2 = 2 \neq 1 \quad \checkmark \\ I = \{1,3\}: &\quad \eta_1 \cdot 3 = 1 \cdot 3 = 3 \neq 1 \quad \checkmark \\ I = \{2,3\}: &\quad \eta_1 \cdot 6 = 1 \cdot 6 = 6 \neq 1 \quad \checkmark \end{aligned}$$

La condición (3.9) se satisface para todos los subconjuntos. Los polinomios base son

$$g_0 = 1 + X^2, \qquad g_1 = X,$$

y la matriz generadora es

$$G = \begin{bmatrix} g_0(\alpha_1) & g_0(\alpha_2) & g_0(\alpha_3) \\ g_1(\alpha_1) & g_1(\alpha_2) & g_1(\alpha_3) \end{bmatrix} = \begin{bmatrix} 1+1^2 & 1+2^2 & 1+3^2 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 5 & 3 \\ 1 & 2 & 3 \end{bmatrix}.$$

Los tres menores 2×2 son:

$$\begin{aligned} \det \begin{pmatrix} 2 & 5 \\ 1 & 2 \end{pmatrix} &= 4 - 5 = -1 = 6 \neq 0, \\ \det \begin{pmatrix} 2 & 3 \\ 1 & 3 \end{pmatrix} &= 6 - 3 = 3 \neq 0, \\ \det \begin{pmatrix} 5 & 3 \\ 2 & 3 \end{pmatrix} &= 15 - 6 = 9 = 2 \neq 0. \end{aligned}$$

Todos los menores son no nulos, luego $d = n - k + 1 = 2$ y el código es **MDS**. ✓

Caso 2: $\eta_1 = 4$ **(condición violada ⇒ NO MDS).**

Verificamos la condición:

$$I = \{1, 2\}: \quad \eta_1 \cdot 2 = 4 \cdot 2 = 8 = 1 \quad \textbf{VIOLA } (3.9)\textbf{.}$$

La condición falla para $I = \{1, 2\}$. Por la demostración del Lema 3.32, esto significa que existe un polinomio no nulo $f \in \mathcal{P}^{3,2}_{1,0,\eta}$ con exactamente las raíces $\alpha_1 = 1$ y $\alpha_2 = 2$. Lo construimos explícitamente:

$$f = \eta_1 f_0 (X - \alpha_1)(X - \alpha_2) = 4 f_0 (X - 1)(X - 2) = 4 f_0 (X^2 - 3X + 2).$$

Tomando $f_0 = 1$ y reduciendo módulo 7:

$$f = 4X^2 - 12X + 8 = 4X^2 + 2X + 1.$$

Verificamos que $f \in \mathcal{P}^{3,2}_{1,0,\eta}$: la forma requerida es $f_0 + f_1 X + \eta_1 f_0 X^2$. Con $f_0 = 1$, $f_1 = 2$, $\eta_1 = 4$:

$$f = 1 + 2X + 4 \cdot 1 \cdot X^2 = 1 + 2X + 4X^2. \quad \checkmark$$

Efectivamente $f(1) = 1 + 2 + 4 = 7 = 0$ y $f(2) = 1 + 4 + 16 = 21 = 0$ en $\mathbb{F}_7$. La palabra código correspondiente es

$$(f(\alpha_1), f(\alpha_2), f(\alpha_3)) = (0, 0, f(3)),$$

donde $f(3) = 1 + 6 + 36 = 43 = 1 \neq 0$. El peso de Hamming de esta palabra es $1 < n - k + 1 = 2$, confirmando que el código **no es MDS**. $\times$

La matriz generadora para $\eta_1 = 4$ es:

$$G = \begin{bmatrix} 1 + 4 \cdot 1^2 & 1 + 4 \cdot 2^2 & 1 + 4 \cdot 3^2 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix},$$

y el menor correspondiente a las columnas $\{1, 2\}$ es

$$\det \begin{pmatrix} 5 & 3 \\ 1 & 2 \end{pmatrix} = 10 - 3 = 7 = 0,$$

confirmando que ese menor es exactamente cero, en correspondencia directa con la violación de la condición en $I = \{1, 2\}$.

Resumen del ejemplo.

η_1	$\eta_1 \prod_{i\in I} \alpha_i$ (peor I)	Cond. (3.9)	¿MDS?
1	máx $= 6 \neq 1$	Satisfecha	Sí
2	máx $= 5 \neq 1$	Satisfecha	Sí
3	máx $= 4 \neq 1$	Satisfecha	Sí
4	$4 \cdot 2 = 1$	Violada en $I = \{1, 2\}$	No
5	$5 \cdot 3 = 1$	Violada en $I = \{1, 3\}$	No
6	$6 \cdot 6 = 1$	Violada en $I = \{2, 3\}$	No

La tabla confirma que la condición (3.9) es exactamente el discriminante entre los η_1 que producen códigos MDS y los que no: los tres valores $\eta_1 \in \{1, 2, 3\}$ satisfacen la condición y producen $d = 2 = n - k + 1$, mientras que los tres valores $\eta_1 \in \{4, 5, 6\}$ la violan y producen $d = 1 < n - k + 1$.

Definición 3.35 (Código $(*)$-torcido)**.** Sea G un subgrupo propio de $(\mathbb{F}_q^*, \cdot)$, $\alpha_i \in G \cup \{0\}$ para todo i, y $(-1)^k \eta_1^{-1} \in \mathbb{F}_q^* \setminus G$. Entonces llamamos a $C^{n,k}_{\alpha,1,0,\eta}$ un *código* $(*)$*-torcido.*

Observación 3.36. Recordamos que $\mathbb{F}_q^*$ es cíclico de orden $q-1$, y sus subgrupos propios tienen cardinalidad $\frac{q-1}{a}$ para cada divisor $a > 1$ de $q-1$. La condición $\alpha_i \in G \cup \{0\}$ restringe los puntos de evaluación al subgrupo G y al cero. Dado que G es cerrado bajo multiplicación, todo producto $\prod_{i\in I} \alpha_i$ con todos los $\alpha_i \neq 0$ pertenece a G. La condición $(-1)^k \eta_1^{-1} \notin G$ garantiza entonces que ese producto nunca puede igualar a $(-1)^k \eta_1^{-1}$, lo que es precisamente la condición del Lema 3.32.

Teorema 3.37. *Todo código* $(*)$*-torcido es MDS.*

Demostración. Sea $I \subseteq \{1, \ldots, n\}$ con $|I| = k$ arbitrario. Distinguimos dos casos.

Si algún $\alpha_i = 0$ para $i \in I$, entonces $\prod_{i\in I} \alpha_i = 0$, y $\eta_1(-1)^k \cdot 0 = 0 \neq 1$, por lo que la condición (3.9) se satisface trivialmente.

Si todos los $\alpha_i \neq 0$ para $i \in I$, entonces cada $\alpha_i \in G$ por la Definición 3.35, y como G es cerrado bajo multiplicación, $\prod_{i\in I} \alpha_i \in G$. Supóngase por contradicción que $\eta_1(-1)^k \prod_{i\in I} \alpha_i = 1$. Entonces $(-1)^k \eta_1^{-1} = \prod_{i\in I} \alpha_i \in G$, contradiciendo la hipótesis $(-1)^k \eta_1^{-1} \in \mathbb{F}_q^* \setminus G$.

En ambos casos la condición (3.9) se satisface para todo I, luego el código es MDS por el Lema 3.32.

Para cualquier divisor $a > 1$ de $q-1$, el subgrupo propio G de cardinalidad $\frac{q-1}{a}$ permite longitudes de código $n = \frac{q-1}{a} + 1$, que pueden ser considerablemente mayores que los códigos obtenidos en las subsecciones anteriores. En particular, si q es impar, entonces $2 \mid q-1$ y es posible tomar $n = \frac{q+1}{2}$. □

Definición 3.38 (Generador k-suma)**.** Sea $(A, \oplus)$ un grupo abeliano finito y $k \in \mathbb{N}$. Un *generador k-suma* de A es un subconjunto $S \subseteq A$ tal que para todo $a \in A$, existen elementos distintos $s_1, \ldots, s_k \in S$ con $a = \bigoplus_{i=1}^{k} s_i$. El menor entero tal que cualquier $S \subseteq A$ con $|S| > M(k, A)$ es un generador k-suma de A se denota $M(k, A)$.

Observación 3.39. En el contexto de los códigos RS torcidos, el grupo abeliano relevante es $(\mathbb{F}_q^*, \cdot)$ con la multiplicación. Un subconjunto $S = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q^*$ es un generador k-suma si todo elemento de $\mathbb{F}_q^*$ puede escribirse como producto de exactamente k elementos distintos de S. El umbral $M(k, A)$ mide qué tan grande debe ser S para que esto ocurra automáticamente: si $n = |S| > M(k, \mathbb{F}_q^*)$, entonces S genera todos los productos de k elementos, lo que por el Lema 3.41 implica que el código no puede ser MDS.

Lema 3.40. *Sea A un grupo abeliano finito de orden $|A| = 2r$ para algún $r \geq 6$. Para cualquier k con $3 \leq k \leq r-2$,*

$$M(k, A) = \begin{cases} r+1, & \text{si } A \in \{\mathbb{Z}_2^m,\ \mathbb{Z}_4 \times \mathbb{Z}_2^{m-1}\} \text{ para algún } m > 1 \text{ y } k \in \{3, r-2\}, \\ r, & \text{en otro caso.} \end{cases}$$

Demostración. Este resultado corresponde al Teorema 3.1 del artículo de Diderrich [7], donde se demuestra de forma completa mediante técnicas de combinatoria aditiva sobre grupos abelianos finitos. El enunciado aquí presentado es una instancia particular de dicho teorema, adaptada al contexto de los códigos Reed–Solomon torcidos. □

Lema 3.41. *Sean k, n, α, η elegidos como en la Definición 3.1 tal que $S := \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q^*$ es un generador k-suma de $(\mathbb{F}_q^*, \cdot)$, y sea $\eta \in \mathbb{F}_q^*$. Entonces el código $C_{\alpha,1,0,\eta}^{n,k}$ no es MDS.*

Demostración. Como S es un generador k-suma de $(\mathbb{F}_q^*, \cdot)$ y $(-1)^k \eta_1^{-1} \in \mathbb{F}_q^*$, existe un subconjunto $I \subseteq \{1, \ldots, n\}$ con $|I| = k$ tal que

$$\prod_{i \in I} \alpha_i = (-1)^k \eta_1^{-1}.$$

Multiplicando ambos miembros por $\eta_1 (-1)^k$ y usando que $(-1)^{2k} = 1$:

$$\eta_1 (-1)^k \prod_{i \in I} \alpha_i = \eta_1 (-1)^k \cdot (-1)^k \eta_1^{-1} = \eta_1 \eta_1^{-1} (-1)^{2k} = 1.$$

Esto viola la condición (3.9) del Lema 3.32 para ese I, luego el código no es MDS. □

Observación 3.42. El Lema 3.41 junto con el Lema 3.40 establece una longitud máxima para los códigos RS torcidos MDS con $\ell = 1$, $h = 0$, $t = 1$: si $n > M(k, \mathbb{F}_q^*)$, entonces ningún código $C_{\alpha,1,0,\eta}^{n,k}$ puede ser MDS. En el caso genérico $M(k, \mathbb{F}_q^*) = r = \frac{q-1}{2}$, lo que da la cota $n \leq \frac{q-1}{2}$. El Teorema 3.37 muestra que esta cota es alcanzada por los códigos $(*)$-torcidos cuando q es impar: tomando G el subgrupo de cuadrados de $\mathbb{F}_q^*$ (de orden $\frac{q-1}{2}$) se obtiene $n = \frac{q+1}{2} = r + 1$, que es exactamente el umbral $M(k, A)$ en los casos excepcionales del Lema 3.40.

Ejemplo 3.43 (Verificación del Teorema 3.37)**.** Construimos un código $(*)$-torcido sobre $\mathbb{F}_7$ con $n = 3$, $k = 2$, $\ell = 1$, $h = 0$, $t = 1$, y verificamos que es MDS.

Paso 1: Estructura de $\mathbb{F}_7^*$ y elección del subgrupo G.

El grupo multiplicativo $\mathbb{F}_7^* = \{1, 2, 3, 4, 5, 6\}$ es cíclico de orden 6, generado por $\gamma = 3$:

$$3^1 = 3, \quad 3^2 = 2, \quad 3^3 = 6, \quad 3^4 = 4, \quad 3^5 = 5, \quad 3^6 = 1.$$

Tomamos el subgrupo propio de índice 2:

$$G = \langle 3^2 \rangle = \langle 2 \rangle = \{1, 2, 4\},$$

que es el subgrupo de cuadrados de $\mathbb{F}_7^*$, de cardinalidad $\frac{q-1}{2} = 3$. Su complemento en $\mathbb{F}_7^*$ es $\mathbb{F}_7^* \setminus G = \{3, 5, 6\}$, que son los no-cuadrados.

Paso 2: Elección de los parámetros y verificación de la Definición 3.35.

Tomamos los puntos de evaluación $\alpha_1 = 1$, $\alpha_2 = 2$, $\alpha_3 = 4$, todos en G, y $\eta_1 = 3$.

Verificamos la condición $(-1)^k \eta_1^{-1} \in \mathbb{F}_7^* \setminus G$: como $k = 2$ se tiene $(-1)^2 = 1$, y el inverso de 3 en $\mathbb{F}_7$ es $3^{-1} = 5$ (pues $3 \cdot 5 = 15 = 1$ en $\mathbb{F}_7$). Como $5 \in \{3, 5, 6\} = \mathbb{F}_7^* \setminus G$, la condición se satisface y $C^{3,2}_{\alpha,1,0,\eta}$ es un código $(*)$-torcido.

Paso 3: Verificación de la condición (3.9)**.**

Los subconjuntos $I \subseteq \{1, 2, 3\}$ con $|I| = 2$ son:

$$\begin{aligned} I = \{1,2\}: &\quad \eta_1 \cdot \alpha_1\alpha_2 = 3 \cdot 1 \cdot 2 = 6 \neq 1 \quad \checkmark \\ I = \{1,3\}: &\quad \eta_1 \cdot \alpha_1\alpha_3 = 3 \cdot 1 \cdot 4 = 12 = 5 \neq 1 \quad \checkmark \\ I = \{2,3\}: &\quad \eta_1 \cdot \alpha_2\alpha_3 = 3 \cdot 2 \cdot 4 = 24 = 3 \neq 1 \quad \checkmark \end{aligned}$$

La condición se satisface para todo I. Esto es consecuencia directa de que $\alpha_i \in G$ para todo i, por lo que $\prod_{i \in I} \alpha_i \in G$, mientras que $\eta_1^{-1} = 5 \notin G$, haciendo imposible que $\eta_1 \prod_{i \in I} \alpha_i = 1$.

Paso 4: Construcción de la matriz generadora.

Los polinomios base son $g_0 = 1 + 3X^2$ y $g_1 = X$. Evaluando en los tres puntos:

$$\begin{aligned} g_0(1) &= 1 + 3 \cdot 1^2 = 4, \\ g_0(2) &= 1 + 3 \cdot 2^2 = 1 + 3 \cdot 4 = 13 = 6, \\ g_0(4) &= 1 + 3 \cdot 4^2 = 1 + 3 \cdot 2 = 7 = 0, \end{aligned}$$

donde se usó $2^2 = 4$ y $4^2 = 16 \equiv 2 \pmod 7$. La matriz generadora es

$$G = \begin{bmatrix} g_0(\alpha_1) & g_0(\alpha_2) & g_0(\alpha_3) \\ g_1(\alpha_1) & g_1(\alpha_2) & g_1(\alpha_3) \end{bmatrix} = \begin{bmatrix} 4 & 6 & 0 \\ 1 & 2 & 4 \end{bmatrix}.$$

Paso 5: Verificación de los menores 2×2.

Calculamos los tres menores (todos en $\mathbb{F}_7$):

$$\det\begin{pmatrix}4 & 6\\ 1 & 2\end{pmatrix} = 4\cdot 2 - 6\cdot 1 = 8-6 = 2 \neq 0, \quad \checkmark$$

$$\det\begin{pmatrix}4 & 0\\ 1 & 4\end{pmatrix} = 4\cdot 4 - 0\cdot 1 = 16 = 2 \neq 0, \quad \checkmark$$

$$\det\begin{pmatrix}6 & 0\\ 2 & 4\end{pmatrix} = 6\cdot 4 - 0\cdot 2 = 24 = 3 \neq 0. \quad \checkmark$$

Todos los menores 2×2 de G son no nulos, luego el código tiene distancia mínima $d = n-k+1 = 2$ y es **MDS**, confirmando el Teorema 3.37.

Teorema 3.44. *Sea q impar y $3 \le k \le \frac{q-1}{2} - 2$. Si $n > \frac{q+1}{2}$, entonces $C^{n,k}_{\alpha,1,0,\eta}$ no es MDS para ninguna elección de α y $\eta \neq 0$ como en la Definición 3.1.*

Demostración. Como q es impar, el grupo multiplicativo $\mathbb{F}_q^*$ es cíclico de orden par $|\mathbb{F}_q^*| = q-1 = 2r$, donde $r = \frac{q-1}{2}$.

La hipótesis $3 \le k \le \frac{q-1}{2} - 2 = r-2$ satisface las condiciones del Lema 3.40, y en el caso genérico (cuando $\mathbb{F}_q^*$ es cíclico y k no está en los casos excepcionales) se tiene

$$M(k, \mathbb{F}_q^*) = r = \frac{q-1}{2}.$$

Sea $S := \{\alpha_1, \ldots, \alpha_n\}$. Como S contiene a lo sumo un elemento igual a 0, se cumple

$$|S \setminus \{0\}| \ge n-1 > \frac{q+1}{2} - 1 = \frac{q-1}{2} = M(k, \mathbb{F}_q^*),$$

donde la segunda desigualdad usa la hipótesis $n > \frac{q+1}{2}$. Por la definición de $M(k,\Lambda)$, el conjunto $S\setminus\{0\} \subseteq \mathbb{F}_q^*$ es un generador k-suma de $(\mathbb{F}_q^*, \cdot)$. El Lema 3.41 implica entonces que $C^{n,k}_{\alpha,1,0,\eta}$ no es MDS. □

Observación 3.45. El Teorema 3.44 establece que los códigos $(*)$-torcidos del Teorema 3.37 son *óptimos* en longitud para q impar: alcanzan $n = \frac{q+1}{2}$, que es el máximo posible. Para q par, la condición $(-1)^k = 1$ hace que el argumento sea distinto, y la búsqueda computacional muestra que existen códigos MDS de longitud $n \approx q/2$ también para q par; por ejemplo, para $q = 16$ existen códigos de longitud $n = 9$ para $k = 3, 4, 5$.

3.2.4. Códigos Reed–Solomon Torcidos (+)

Consideramos ahora códigos RS torcidos con una sola torsión $\ell = 1$, enganche $h = k-1$ y torsión $t = 1$. Con estos parámetros, el enganche apunta al coeficiente f_{k-1} de mayor grado en la parte clásica del polinomio, de modo que todo polinomio del espacio torcido tiene la forma

$$f = \sum_{i=0}^{k-1} f_i X^i + \eta_1 f_{k-1} X^k.$$

Esta elección produce una condición MDS en términos de *sumas* de los puntos de evaluación, en contraste con el caso $h = 0$ donde aparecían productos. Esto da lugar a una construcción análoga a la de los códigos $(*)$-torcidos, pero usando subgrupos aditivos de $\mathbb{F}_q$ en lugar de subgrupos multiplicativos.

Lema 3.46. *Sea $\ell = 1$ y sean n, k, α, η elegidos como en la Definición 3.1. El código $C^{n,k}_{\alpha,1,k-1,\eta}$ es MDS si y solo si*

$$\eta \sum_{i \in I} \alpha_i \neq -1 \qquad \forall\, I \subseteq \{1, \ldots, n\},\ |I| = k. \tag{3.10}$$

Demostración. La demostración establece la equivalencia mostrando que el código no es MDS si y solo si existe $I \subseteq \{1, \ldots, n\}$ con $|I| = k$ que viola (3.10).

Parte necesaria. Supongamos que existe $f \in \mathcal{P}^{n,k}_{1,k-1,\eta} \setminus \{0\}$ con k raíces entre los α_i. Todo polinomio del espacio torcido tiene la forma

$$f = \sum_{i=0}^{k-1} f_i X^i + \eta f_{k-1} X^k.$$

Si $f_{k-1} = 0$, entonces el término torcido desaparece y f tiene grado menor que $k-1$, por lo que tiene a lo sumo $k-2$ raíces. En tal caso la palabra código correspondiente tiene peso al menos $n - k + 2 > n - k + 1$, y el código no se ve afectado. Por tanto el único caso crítico es $f_{k-1} \neq 0$.

Con $f_{k-1} \neq 0$, el polinomio f tiene grado exactamente k y coeficiente líder $f_k = \eta f_{k-1} \neq 0$. Como f tiene exactamente k raíces α_i con $i \in I$, $|I| = k$, el teorema de factorización de polinomios da:

$$f = \eta f_{k-1} \prod_{i \in I} (X - \alpha_i).$$

Comparamos el coeficiente de X^{k-1} en ambas expresiones de f. De la definición del espacio torcido, ese coeficiente es f_{k-1}. De la factorización, expandiendo el producto:

$$\prod_{i \in I} (X - \alpha_i) = X^k - \Big(\sum_{i \in I} \alpha_i\Big) X^{k-1} + \cdots,$$

el coeficiente de X^{k-1} en f es $-\eta f_{k-1} \sum_{i \in I} \alpha_i$. Igualando:

$$f_{k-1} = -\eta f_{k-1} \sum_{i \in I} \alpha_i.$$

Dividiendo por $f_{k-1} \neq 0$:

$$1 = -\eta \sum_{i \in I} \alpha_i, \quad \text{es decir,} \quad \eta \sum_{i \in I} \alpha_i = -1,$$

lo que viola la condición (3.10) para ese I.

Parte suficiente. Recíprocamente, supongamos que existe $I \subseteq \{1, \ldots, n\}$ con $|I| = k$ tal

que $\eta \sum_{i\in I} \alpha_i = -1$. Definimos

$$f = \eta \prod_{i\in I}(X - \alpha_i).$$

Este polinomio tiene grado k y coeficiente líder η. Expandiendo el producto, el coeficiente de X^{k-1} es

$$-\eta \sum_{i\in I} \alpha_i = -(-1) = 1 =: f_{k-1}.$$

El coeficiente de X^k es $\eta = \eta \cdot 1 = \eta f_{k-1}$, que es exactamente la relación que define el espacio torcido $\mathcal{P}^{n,k}_{1,k-1,\eta}$. Por tanto $f \in \mathcal{P}^{n,k}_{1,k-1,\eta}$.

Además, f es no nulo y tiene exactamente k raíces entre los α_i, luego la palabra código correspondiente tiene peso $n - k < n - k + 1$, y el código no es MDS. □

Observación 3.47. Los Lemas 3.32 y 3.46 presentan una estructura de demostración análoga: en ambos casos se comparan dos expresiones del mismo polinomio, una obtenida a partir de la definición del espacio torcido y otra mediante su factorización sobre las raíces, y se extrae la condición MDS mediante la comparación del coeficiente asociado al término torcido.

Sin embargo, la condición obtenida difiere según el valor del índice de enganche. En el Lema 3.32, correspondiente al caso $h = 0$, la torsión afecta al coeficiente f_0 del término constante. Al comparar los términos constantes de ambas expresiones se obtiene una condición que involucra productos de los elementos α_i.

Por otro lado, en el Lema 3.46, correspondiente al caso $h = k - 1$, la torsión afecta al coeficiente f_{k-1}, que corresponde al término de mayor grado de la parte clásica del polinomio. En este caso, la comparación de coeficientes conduce a una condición que involucra sumas de los elementos α_i, mediante la fórmula de Vieta.

Esta diferencia explica la naturaleza de las dos construcciones: en el primer caso aparecen productos de elementos de $\mathbb{F}_q$, mientras que en el segundo aparecen sumas, lo que motiva el uso de subgrupos aditivos de $\mathbb{F}_q$ en la construcción $(+)$-torcida, en contraste con los subgrupos multiplicativos empleados en el caso $h = 0$.

Ejemplo 3.48 (Verificación del Lema 3.46)**.** Verificamos la condición (3.10) sobre $\mathbb{F}_7$ con parámetros $n = 3$, $k = 2$, $\ell = 1$, $h = k - 1 = 1$, $t = 1$, y puntos de evaluación

$$\alpha_1 = 1, \quad \alpha_2 = 2, \quad \alpha_3 = 3.$$

Paso 1: Forma de los polinomios del espacio torcido.

Con $h = k - 1 = 1$ y $t = 1$, el único polinomio base modificado es g_1 (el enganche apunta al índice $i = k - 1 = 1$), con potencia $k - 1 + t = 2$:

$$g_0 = X^0 = 1, \qquad g_1 = X + \eta X^2.$$

Todo polinomio del espacio torcido tiene la forma

$$f = f_0 + f_1 X + \eta f_1 X^2,$$

donde $f_0, f_1 \in \mathbb{F}_7$ son los coeficientes libres. Nótese que el coeficiente de X^2 es $\eta f_1 = \eta f_{k-1}$, confirmando la estructura general descrita en el Lema 3.46.

Paso 2: Interpretación de la condición (3.10)**.**

Con $k = 2$, los subconjuntos $I \subseteq \{1, 2, 3\}$ con $|I| = 2$ son exactamente tres, y la condición exige $\eta \sum_{i \in I} \alpha_i \neq -1 = 6$ en $\mathbb{F}_7$. Las sumas de pares de puntos son:

I	$\sum_{i \in I} \alpha_i$	Valor en $\mathbb{F}_7$
$\{1, 2\}$	$\alpha_1 + \alpha_2 = 1 + 2$	3
$\{1, 3\}$	$\alpha_1 + \alpha_3 = 1 + 3$	4
$\{2, 3\}$	$\alpha_2 + \alpha_3 = 2 + 3$	5

La condición (3.10) exige que $\eta \cdot 3 \neq 6$, $\eta \cdot 4 \neq 6$ y $\eta \cdot 5 \neq 6$ simultáneamente en $\mathbb{F}_7$.

Paso 3: Determinación de los η válidos.

Calculamos $\eta \cdot s$ para cada suma $s \in \{3, 4, 5\}$ y cada $\eta \in \mathbb{F}_7^*$, identificando qué valores dan 6:

η	$\eta \cdot 3$	$\eta \cdot 4$	$\eta \cdot 5$	¿Cumple cond.?
1	3	4	5	Sí
2	6	1	3	No ($I = \{1, 2\}$)
3	2	5	1	Sí
4	5	2	6	No ($I = \{2, 3\}$)
5	1	6	4	No ($I = \{1, 3\}$)
6	4	3	2	Sí

Los valores $\eta \in \{1, 3, 6\}$ satisfacen la condición para los tres subconjuntos.

Paso 4: Construcción de la matriz generadora para $\eta = 1$.

Los polinomios base son $g_0 = 1$ y $g_1 = X + X^2$. Evaluando en los tres puntos:

$$\begin{aligned} &g_0(1) = 1, \quad g_0(2) = 1, \quad g_0(3) = 1, \\ &g_1(1) = 1 + 1^2 = 2, \\ &g_1(2) = 2 + 2^2 = 2 + 4 = 6, \\ &g_1(3) = 3 + 3^2 = 3 + 9 = 3 + 2 = 5. \end{aligned}$$

La matriz generadora es

$$G = \begin{bmatrix} g_0(\alpha_1) & g_0(\alpha_2) & g_0(\alpha_3) \\ g_1(\alpha_1) & g_1(\alpha_2) & g_1(\alpha_3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 6 & 5 \end{bmatrix}.$$

Paso 5: Verificación de los menores 2×2.

Calculamos los tres menores en $\mathbb{F}_7$:

$$\det\begin{pmatrix}1 & 1\\ 2 & 6\end{pmatrix} = 1\cdot 6 - 1\cdot 2 = 6-2 = 4 \neq 0, \quad \checkmark$$
$$\det\begin{pmatrix}1 & 1\\ 2 & 5\end{pmatrix} = 1\cdot 5 - 1\cdot 2 = 5-2 = 3 \neq 0, \quad \checkmark$$
$$\det\begin{pmatrix}1 & 1\\ 6 & 5\end{pmatrix} = 1\cdot 5 - 1\cdot 6 = 5-6 = -1 = 6 \neq 0. \quad \checkmark$$

Todos los menores 2×2 son no nulos, luego $d = n-k+1 = 2$ y el código es **MDS**. ✓

Paso 6: Verificación de la parte suficiente para $\eta = 2$ (condición violada).

Para $\eta = 2$ la condición falla en $I = \{1,2\}$ pues $\eta(\alpha_1+\alpha_2) = 2\cdot 3 = 6 = -1$. Por la parte suficiente del Lema 3.46, el polinomio

$$f = \eta\prod_{i\in I}(X-\alpha_i) = 2(X-1)(X-2) = 2(X^2-3X+2) = 2X^2+X+4$$

debe pertenecer a $\mathcal{P}^{3,2}_{1,1,2}$. Verificamos: la forma requerida es $f_0+f_1X+\eta f_1X^2$. Identificando coeficientes: $f_0 = 4$, $f_1 = 1$, y el coeficiente de X^2 debe ser $\eta f_1 = 2\cdot 1 = 2$. Efectivamente el coeficiente de X^2 en f es 2. ✓

El polinomio f tiene exactamente las raíces $\alpha_1 = 1$ y $\alpha_2 = 2$, confirmando que la palabra código

$$(f(\alpha_1), f(\alpha_2), f(\alpha_3)) = (0,\ 0,\ f(3))$$

tiene peso $1 < n-k+1 = 2$, de donde el código con $\eta = 2$ no es MDS.

Resumen.

η	Cond. (3.10) satisfecha	Todos los menores $\neq 0$	MDS
1	Sí	Sí	Sí
2	No (viola en $I = \{1,2\}$)	No	No
3	Sí	Sí	Sí
4	No (viola en $I = \{2,3\}$)	No	No
5	No (viola en $I = \{1,3\}$)	No	No
6	Sí	Sí	Sí

La condición (3.10) discrimina exactamente entre los η que producen códigos MDS ($\eta \in \{1,3,6\}$) y los que no ($\eta \in \{2,4,5\}$), confirmando el Lema 3.46.

Definición 3.49 (Código (+)-torcido)**.** Sea V un subgrupo propio de $(\mathbb{F}_q,+)$, $\eta^{-1} \in \mathbb{F}_q \setminus V$, y α un conjunto de n elementos distintos de V. Entonces $C^{n,k}_{\alpha,1,k-1,\eta}$ se llama un *código (+)-torcido*.

Observación 3.50. El grupo aditivo $(\mathbb{F}_q,+)$ tiene orden $q = p^m$, donde p es la característica del cuerpo. Sus subgrupos propios más naturales son los subcuerpos $\mathbb{F}_{p^j}$ para $j \mid m$, $j < m$,

vistos como subgrupos aditivos de orden p^j. El subgrupo propio de mayor cardinalidad posible tiene orden q/p, y tomándolo como conjunto de puntos de evaluación se obtiene $n = q/p$. Para q par ($p = 2$), esto da $n = q/2$.

La condición $\alpha_i \in V$ garantiza que toda suma $\sum_{i\in I} \alpha_i$ pertenece a V (por ser V cerrado bajo la operación del grupo). La condición $-\eta^{-1} \notin V$ garantiza entonces que $\eta \sum_{i\in I} \alpha_i \neq -1$ para todo I, satisfaciendo la condición (3.10) del Lema 3.46.

Teorema 3.51. *Todo código (+)-torcido es MDS.*

Demostración. Sea $I \subseteq \{1, \ldots, n\}$ con $|I| = k$ arbitrario. Como cada $\alpha_i \in V$ y V es cerrado bajo suma, se tiene $\sum_{i\in I} \alpha_i \in V$. Si fuera $\eta \sum_{i\in I} \alpha_i = -1$, entonces $\sum_{i\in I} \alpha_i = -\eta^{-1}$, lo que contradiría $-\eta^{-1} \notin V$. Por tanto la condición (3.10) se satisface para todo I, y el código es MDS por el Lema 3.46.

□

Observación 3.52 (Evaluación en el infinito)**.** Para $\ell = 1$ y parámetros h_1, t_1 generales, el grado máximo de un polinomio en $\mathcal{P}^{n,k}_{t,h,\eta}$ es $k - 1 + t_1$. Se define la *evaluación en el infinito* como

$$f(\infty) := f_{k-1+t_1},$$

es decir, $f(\infty)$ es el coeficiente del monomio de mayor grado de f. Este mapa es $\mathbb{F}_q$-lineal, pues $(\alpha f + \beta g)(\infty) = \alpha f(\infty) + \beta g(\infty)$ para todo $f, g \in \mathcal{P}^{n,k}_{t,h,\eta}$ y $\alpha, \beta \in \mathbb{F}_q$. En consecuencia, añadir ∞ al conjunto de puntos de evaluación produce un código lineal.

Para $h_1 = k-1$ específicamente, se tiene $f(\infty) = \eta f_{k-1}$, que es cero si y solo si $f_{k-1} = 0$, lo cual equivale a $\deg(f) < k - 1$. Por tanto, si un código (+)-torcido con puntos $\alpha \in \mathbb{F}_q^n$ es MDS, el *código extendido* con puntos de evaluación $\alpha' = [\alpha_1, \ldots, \alpha_n, \infty]$ también es MDS. Al extender un código (+)-torcido sobre un cuerpo de característica 2, se obtiene un código MDS de longitud

$$n = \frac{q}{2} + 1.$$

Lema 3.53. *Sean k, n, α, η elegidos como en la Definición 3.1 tal que $S := \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q$ es un generador k-suma de $(\mathbb{F}_q, +)$. Entonces el código $C^{n,k}_{\alpha,1,k-1,\eta}$ no es MDS.*

Demostración. Como S es un generador k-suma de $(\mathbb{F}_q, +)$ y $-\eta^{-1} \in \mathbb{F}_q$, existe $I \subseteq \{1, \ldots, n\}$ con $|I| = k$ tal que $\sum_{i\in I} \alpha_i = -\eta^{-1}$. Multiplicando por η: $\eta \sum_{i\in I} \alpha_i = -1$, lo que viola la condición (3.10) del Lema 3.46. Por tanto el código no es MDS. □

Teorema 3.54. *Sea q par y $3 \leq k \leq \frac{q}{2} - 2$. Si la longitud del código satisface*

$$n > \begin{cases} \dfrac{q}{2}, & \text{si } 3 < k < \dfrac{q}{2} - 3, \\ \dfrac{q}{2} + 1, & \text{si } k \in \left\{3, \dfrac{q}{2} - 2\right\}, \end{cases}$$

entonces el código torcido $C^{n,k}_{\boldsymbol{\alpha},1,k-1,\boldsymbol{\eta}}$ no es MDS para ninguna elección de $\boldsymbol{\eta}$ como en la Definición 3.1.

Demostración. La estrategia es mostrar que, bajo las hipótesis sobre n, el conjunto de puntos de evaluación $S := \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q$ es necesariamente un generador k-suma de $(\mathbb{F}_q, +)$, y entonces aplicar el Lema 3.53 para concluir que el código no puede ser MDS.

Paso 1: Identificación del grupo y su orden.

Como q es par, el cuerpo $\mathbb{F}_q$ tiene característica 2. Consideramos el grupo aditivo $A := (\mathbb{F}_q, +)$, que es un grupo abeliano finito de orden $|A| = q$. Escribimos $q = 2r$, de modo que $|A| = 2r$ con

$$r = \frac{q}{2}.$$

La hipótesis $3 \leq k \leq \frac{q}{2} - 2$ se traduce en $3 \leq k \leq r - 2$, que es precisamente el rango requerido por el Lema 3.40 para que $M(k, A)$ esté bien definido.

Paso 2: Cálculo de $M(k, A)$.

Por el Lema 3.40, para $3 \leq k \leq r - 2$ se tiene

$$M(k, A) = \begin{cases} r + 1, & \text{si } A \in \left\{\mathbb{Z}_2^m,\ \mathbb{Z}_4 \times \mathbb{Z}_2^{m-1}\right\} \text{ para algún } m > 1 \text{ y } k \in \{3, r-2\}, \\ r, & \text{en otro caso.} \end{cases}$$

Dado que $\mathbb{F}_q$ tiene característica 2 y $q = 2^m$ para algún $m \geq 1$, su grupo aditivo satisface $(\mathbb{F}_q, +) \cong \mathbb{Z}_2^m$. Por tanto:

- Si $k \in \{3, r-2\} = \{3, \frac{q}{2} - 2\}$, entonces $M(k, A) = r + 1 = \frac{q}{2} + 1$.
- Si $3 < k < \frac{q}{2} - 3$, entonces $M(k, A) = r = \frac{q}{2}$.

Paso 3: S es un generador k-suma.

Recordemos que $M(k, A)$ es el menor entero tal que todo subconjunto $S \subseteq A$ con $|S| > M(k, A)$ es un generador k-suma de A. Por hipótesis, $n > M(k, A)$ en ambos casos:

- Si $3 < k < \frac{q}{2} - 3$: se supone $n > \frac{q}{2} = M(k, A)$.
- Si $k \in \{3, \frac{q}{2} - 2\}$: se supone $n > \frac{q}{2} + 1 = M(k, A)$.

En consecuencia, $|S| = n > M(k, A)$, y por definición de $M(k, A)$ el conjunto $S = \{\alpha_1, \ldots, \alpha_n\}$ es un generador generador k-suma de $(\mathbb{F}_q, +)$.

Paso 4: Conclusión por el Lema 3.53.

El Lema 3.53 establece que si $S := \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q$ es un generador k-suma de $(\mathbb{F}_q, +)$, entonces el código $C^{n,k}_{\boldsymbol{\alpha},1,k-1,\boldsymbol{\eta}}$ no es MDS para ninguna elección de $\boldsymbol{\eta} \neq 0$. Puesto que en el Paso 3 se verificó que S es un generador k-suma bajo las hipótesis del teorema, se concluye que $C^{n,k}_{\boldsymbol{\alpha},1,k-1,\boldsymbol{\eta}}$ no es MDS para ninguna elección admisible de $\boldsymbol{\eta}$. □

Observación 3.55. Los Teoremas 3.44 y 3.54 establecen cotas superiores óptimas para las longitudes de los códigos $(*)$-torcidos y $(+)$-torcidos respectivamente, siendo estas cotas alcanzadas por las construcciones explícitas de los Teoremas 3.37 y 3.51. La diferencia entre los dos casos proviene de la estructura del grupo relevante: el grupo multiplicativo $\mathbb{F}_q^*$ es siempre cíclico, mientras que el grupo aditivo $(\mathbb{F}_q, +)$ tiene estructura $\mathbb{Z}_2^m$ para $q = 2^m$, lo que produce un comportamiento ligeramente distinto en los casos extremos de k, reflejado en la aparición de la cota $\frac{q}{2} + 1$ para $k \in \{3, \frac{q}{2} - 2\}$ en el Teorema 3.54.

Capítulo 4

Codificación de códigos Reed–Solomon torcidos

4.1. Codificación de códigos Reed–Solomon torcidos

En esta sección describimos la codificación de los códigos Reed–Solomon torcidos siguiendo la misma filosofía que en el caso de los códigos Reed–Solomon clásicos: se escoge un polinomio del espacio de mensajes, se expresa respecto de una base conveniente de ese espacio y luego se evalúa en los puntos de la familia de evaluación. La diferencia esencial es que, en el caso torcido, la base del espacio de polinomios ya no es la base monomial usual, sino la base deformada por las *torsiones*.

4.1.1. Parámetros y estructura del código

Sea $\mathbb{F}_q$ un cuerpo finito. Fijamos enteros positivos n, k, ℓ con $k < n$. Además, elegimos:

- **Vector de torsiones**
$$\boldsymbol{t} = (t_1, \ldots, t_\ell) \in \{1, \ldots, n-k\}^\ell,$$
- **Vector de enganche**
$$\boldsymbol{h} = (h_1, \ldots, h_\ell) \in \{0, \ldots, k-1\}^\ell,$$
- **Vector de coeficientes de torsión**
$$\boldsymbol{\eta} = (\eta_1, \ldots, \eta_\ell) \in (\mathbb{F}_q^\times)^\ell,$$

de manera que los pares
$$[h_i, t_i], \qquad i = 1, \ldots, \ell,$$
sean dos a dos distintos.

Sea además
$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$$
un vector de puntos de evaluación, con $\alpha_i \neq \alpha_j$ si $i \neq j$.

Espacio de polinomios torcidos. Definimos el espacio de polinomios torcidos

$$\mathcal{P}^{n,k}_{t,\boldsymbol{h},\boldsymbol{\eta}} = \left\{ f = \sum_{i=0}^{k-1} f_i X^i + \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j} : \ f_i \in \mathbb{F}_q \right\}.$$

Este espacio tiene dimensión k sobre $\mathbb{F}_q$. La idea es que un polinomio torcido no está determinado solamente por sus coeficientes monomiales $f_0, \ldots, f_{k-1}$, sino que además cada coeficiente de enganche f_{h_j} produce un término adicional de grado alto,

$$\eta_j f_{h_j} X^{k-1+t_j}.$$

Ese término extra es precisamente la parte *torcida*.

Base del espacio. Una base de $\mathcal{P}^{n,k}_{t,\boldsymbol{h},\boldsymbol{\eta}}$ viene dada por los polinomios

$$g_i := X^i + \sum_{\substack{j=1 \\ h_j=i}}^{\ell} \eta_j X^{k-1+t_j}, \qquad i = 0, \ldots, k-1.$$

Es decir:

- si el índice i no aparece como enganche de ninguna torsión, entonces

$$g_i = X^i,$$

- si existen torsiones anclados en i, entonces g_i es el monomio X^i más la suma de los términos torcidos asociados a ese enganche.

Esta base es la correcta para codificar, porque cada polinomio del espacio se expresa de manera única como combinación lineal de los g_i.

4.1.2. Mensaje y polinomio asociado

Sea ahora un mensaje

$$\boldsymbol{m} = (m_0, \ldots, m_{k-1}) \in \mathbb{F}_q^k.$$

El polinomio de mensaje asociado no se escribe como en el caso RS normal usando solo la base monomial, sino respecto de la base torcida $\{g_0, \ldots, g_{k-1}\}$:

$$f_{\boldsymbol{m}}(X) = \sum_{i=0}^{k-1} m_i g_i(X).$$

Sustituyendo la definición de g_i, obtenemos

$$f_{\boldsymbol{m}}(X) = \sum_{i=0}^{k-1} m_i X^i + \sum_{j=1}^{\ell} \eta_j \, m_{h_j} \, X^{k-1+t_j}.$$

4.1.3. Codificación

La palabra código asociada al mensaje $\boldsymbol{m}$ se obtiene evaluando el polinomio $f_{\boldsymbol{m}}(X)$ en los puntos de evaluación $\alpha_1, \ldots, \alpha_n$:

$$\mathrm{ev}_{\boldsymbol{\alpha}}(f_{\boldsymbol{m}}) = \Big(f_{\boldsymbol{m}}(\alpha_1), \ldots, f_{\boldsymbol{m}}(\alpha_n)\Big) \in \mathbb{F}_q^n.$$

Es decir, la palabra código es

$$\boldsymbol{c} = \left(\sum_{i=0}^{k-1} m_i \alpha_1^i + \sum_{j=1}^{\ell} \eta_j m_{h_j} \alpha_1^{k-1+t_j}, \; \ldots, \; \sum_{i=0}^{k-1} m_i \alpha_n^i + \sum_{j=1}^{\ell} \eta_j m_{h_j} \alpha_n^{k-1+t_j} \right).$$

Así, cada coordenada de la palabra código tiene dos contribuciones:

- la contribución normal, proveniente de los términos $X^0, \ldots, X^{k-1}$,

- la contribución torcida, proveniente de los términos de alto grado X^{k-1+t_j}.

4.1.4. Algoritmo de codificación

La codificación de un código Reed–Solomon torcido se realiza de la siguiente manera:
(1) Se fija el mensaje

$$\boldsymbol{m} = (m_0, \ldots, m_{k-1}) \in \mathbb{F}_q^k.$$

(2) Se construye el polinomio

$$f_{\boldsymbol{m}}(X) = \sum_{i=0}^{k-1} m_i g_i(X).$$

(3) Se evalúa $f_{\boldsymbol{m}}$ en cada punto α_r:

$$c_r = f_{\boldsymbol{m}}(\alpha_r), \qquad r = 1, \ldots, n.$$

(4) Se obtiene la palabra código

$$\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{F}_q^n.$$

Equivalentemente, usando la base torcida, cada coordenada del mensaje determina de forma única el polinomio y, por tanto, la palabra código.

4.1.5. Relación con los RS normales

En los códigos Reed–Solomon normales, el polinomio de mensaje es

$$f(X) = \sum_{i=0}^{k-1} m_i X^i.$$

Aquí, en cambio, el polinomio de mensaje se escribe como

$$f_{\boldsymbol{m}}(X) = \sum_{i=0}^{k-1} m_i g_i(X),$$

donde la base $\{g_i\}$ ya incorpora los términos torcidos.

Por tanto, el procedimiento es el mismo en esencia , la única diferencia es que ahora la base del espacio de mensajes no es la monomial estándar, sino la base torcida dada por los g_i. De este modo, la codificación de los códigos Reed–Solomon torcidos se entiende exactamente como en el caso clásico, pero usando la base torcida adecuada del espacio de polinomios.

Ejemplo 4.1. Consideremos un código Reed–Solomon torcido sobre $\mathbb{F}_7$ con parámetros

$$n = 5, \qquad k = 3, \qquad \ell = 1,$$

y con un única torsión dado por

$$t_1 = 1, \qquad h_1 = 1, \qquad \eta_1 = 2.$$

Tomamos como puntos de evaluación

$$\boldsymbol{\alpha} = (0, 1, 2, 3, 4).$$

La base torcida del espacio de polinomios es

$$g_0(X) = 1, \qquad g_1(X) = X + 2X^3, \qquad g_2(X) = X^2.$$

Sea el mensaje

$$\boldsymbol{m} = (1, 2, 3) \in \mathbb{F}_7^3.$$

El polinomio asociado es

$$f_{\boldsymbol{m}}(X) = m_0 g_0(X) + m_1 g_1(X) + m_2 g_2(X).$$

Sustituyendo los valores del mensaje, obtenemos

$$f_{\boldsymbol{m}}(X) = 1 \cdot 1 + 2(X + 2X^3) + 3X^2 = 1 + 2X + 3X^2 + 4X^3.$$

Ahora evaluamos en los puntos $\alpha_1, \ldots, \alpha_5$:

$$f_{\boldsymbol{m}}(0) = 1.$$

$$f_{\boldsymbol{m}}(1) = 1 + 2 + 3 + 4 = 10 \equiv 3 \pmod 7.$$

$$f_{\boldsymbol{m}}(2) = 1 + 2(2) + 3(2)^2 + 4(2)^3 = 1 + 4 + 12 + 32 \equiv 1 + 4 + 5 + 4 = 14 \equiv 0 \pmod 7.$$

$$f_{\boldsymbol{m}}(3) = 1 + 2(3) + 3(3)^2 + 4(3)^3 = 1 + 6 + 27 + 108 \equiv 1 + 6 + 6 + 3 = 16 \equiv 2 \pmod 7.$$

$$f_{\boldsymbol{m}}(4) = 1 + 2(4) + 3(4)^2 + 4(4)^3 = 1 + 8 + 48 + 256 \equiv 1 + 1 + 6 + 4 = 12 \equiv 5 \pmod 7.$$

Por tanto, la palabra código es

$$\boldsymbol{c} = (1, 3, 0, 2, 5).$$

En este ejemplo, la parte normal del polinomio es

$$1 + 2X + 3X^2,$$

mientras que la parte torcida es

$$4X^3 = \eta_1 m_{h_1} X^{k-1+t_1},$$

ya que $h_1 = 1$ y, por tanto, $m_{h_1} = m_1 = 2$.

Capítulo 5

Decodificación de códigos Reed–Solomon torcidos

5.1. Decodificación exhaustiva de códigos Reed–Solomon torcidos

La decodificación de un código Reed–Solomon torcido puede reducirse, en cierta forma, a la decodificación de un código Reed–Solomon clásico. La dificultad aparece porque los polinomios torcidos contienen términos adicionales cuyos coeficientes forman parte del mensaje transmitido y, por tanto, son desconocidos en el proceso de decodificación.

La idea del algoritmo de búsqueda exhaustiva consiste en probar todas las posibles combinaciones de los coeficientes asociados a los términos torcidos. Para cada elección, se resta de la palabra recibida la contribución correspondiente a esos términos, de modo que el problema se transforma en la decodificación de una palabra de un código Reed–Solomon convencional. Si la elección de los coeficientes coincide con la real, la parte torcida queda exactamente eliminada y el decodificador Reed–Solomon recupera la palabra correcta.

En particular, si la palabra recibida es

$$r = c + e,$$

donde c es una palabra del código Reed–Solomon torcido y e es el vector de error, el algoritmo examina todos los elementos de $\mathbb{F}_q^\ell$, donde ℓ es el número de torsiones. Para cada posible elección de los coeficientes torcidos, se construye el polinomio

$$g(X) = \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j},$$

se evalúa en los puntos del código y se resta de la palabra recibida. El resultado se decodifica entonces con un algoritmo estándar para códigos Reed–Solomon, como Gao, Berlekamp–Massey [8], Welch–Berlekamp [9] o el algoritmo euclidiano.

Este procedimiento es correcto porque entre las q^ℓ posibilidades existe exactamente una que coincide con los coeficientes verdaderos del polinomio transmitido. En ese caso, la resta elimina por completo la parte torcida y el problema se reduce al caso Reed–Solomon clásico. Por tanto, si el código subyacente puede corregir el patrón de errores presente, la palabra transmitida se recupera de forma exacta.

El costo computacional del método está dominado por la exploración exhaustiva de todas las combinaciones posibles

Aunque es más costoso que la decodificación de un Reed–Solomon clásico, este método es general y completamente determinista. Por ello resulta útil en la implementación del descifrado en criptosistemas basados en códigos Reed–Solomon torcidos.

5.1.1. Parámetros y estructura Códigos torcido Reed-Solomon

El código Reed–Solomon torcido se define como en la Definición 3.5, a partir del espacio de polinomios $(\boldsymbol{t}, \boldsymbol{h}, \boldsymbol{\eta})$-torcidos introducido en la Definición 3.3, cuyos parámetros se

establecen en la Definición 3.1.

5.1.2. Decodificación por fuerza bruta de códigos torcido Reed–Solomon

Sea $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{F}_q^n$ una palabra recibida a través de un canal ruidoso. El problema de decodificación consiste en encontrar un polinomio $f \in \mathcal{P}_{t,h,\eta}^{n,k}$ tal que la distancia de Hamming entre $\mathbf{r}$ y $\mathrm{ev}_{\boldsymbol{\alpha}}(f)$ sea mínima.

Recordemos que todo polinomio torcido puede escribirse de manera única como

$$f(X) = g(X) + T(X),$$

donde

$$g(X) = \sum_{i=0}^{k-1} f_i X^i \qquad \text{y} \qquad T(X) = \sum_{j=1}^{\ell} \eta_j f_{h_j} X^{k-1+t_j}.$$

En particular, una vez fijados los coeficientes torcidos $f_{h_1}, \ldots, f_{h_\ell}$, el término $T(X)$ queda completamente determinado.

Proposición 5.1 (Reducción al código Reed–Solomon clásico)**.** *Sea* $\mathbf{r} = \mathbf{c} + \mathbf{e}$*, con*

$$\mathbf{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f) \qquad y \qquad f = g + T \in \mathcal{P}_{t,h,\eta}^{n,k}.$$

Si se conocen los coeficientes torcidos $f_{h_1}, \ldots, f_{h_\ell}$*, entonces*

$$\mathbf{r}' = \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T)$$

satisface

$$\mathbf{r}' = \mathrm{ev}_{\boldsymbol{\alpha}}(g) + \mathbf{e},$$

donde $g(X) = \sum_{i=0}^{k-1} f_i X^i$ *tiene grado menor que* k*. Por tanto,* $\mathbf{r}'$ *es una palabra recibida de un código Reed–Solomon clásico de dimensión* k*.*

Demostración. Por definición,

$$\mathbf{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f) = \mathrm{ev}_{\boldsymbol{\alpha}}(g + T).$$

Como la evaluación es lineal,

$$\mathrm{ev}_{\boldsymbol{\alpha}}(g + T) = \mathrm{ev}_{\boldsymbol{\alpha}}(g) + \mathrm{ev}_{\boldsymbol{\alpha}}(T).$$

Luego,

$$\mathbf{r}' = \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T) = (\mathbf{c} + \mathbf{e}) - \mathrm{ev}_{\boldsymbol{\alpha}}(T) = \mathrm{ev}_{\boldsymbol{\alpha}}(g) + \mathbf{e}.$$

Esto prueba la afirmación. □

Teorema 5.2 (Corrección y unicidad de la decodificación por fuerza bruta)**.** *Sea* $C_{n,k}^{\alpha,t,h,\eta}$ *un código torcido Reed–Solomon MDS y supongamos que*

$$\mathbf{r} = \mathbf{c} + \mathbf{e}, \qquad \mathbf{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f^*) \in C_{n,k}^{\alpha,t,h,\eta}, \qquad \mathbf{w}(\mathbf{e}) \leq \tau,$$

donde

$$\tau = \left\lfloor \frac{n-k}{2} \right\rfloor.$$

Entonces el algoritmo de decodificación por fuerza bruta recupera de forma única la palabra código original $\mathbf{c}$.

Demostración. Como el código $C_{n,k}^{\alpha,t,h,\eta}$ es MDS, su distancia mínima es

$$d = n - k + 1.$$

Por tanto, su radio de decodificación única es

$$\tau = \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{n-k}{2} \right\rfloor.$$

Sea $f^* = g^* + T^*$ el polinomio torcido original, con

$$g^*(X) = \sum_{i=0}^{k-1} f_i^* X^i, \qquad T^*(X) = \sum_{j=1}^{\ell} \eta_j f_{h_j}^* X^{k-1+t_j}.$$

Cuando el algoritmo prueba la tupla correcta

$$(u_1, \ldots, u_\ell) = (f_{h_1}^*, \ldots, f_{h_\ell}^*),$$

se obtiene

$$\mathbf{r}' = \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T^*) = \mathrm{ev}_{\boldsymbol{\alpha}}(g^*) + \mathbf{e}.$$

Como w(e)$\leq \tau$ y $\mathrm{ev}_{\boldsymbol{\alpha}}(g^*)$ pertenece a un código Reed–Solomon clásico de dimensión k, el decodificador RS clásico recupera de manera única g^*.

Para probar la unicidad de la palabra recuperada, supongamos que existe otra palabra código $\mathbf{c}' \in C_{n,k}^{\alpha,t,h,\eta}$, distinta de $\mathbf{c}$, tal que

$$(\mathbf{r}, \mathbf{c}') \leq \tau.$$

Entonces, por la desigualdad triangular de la distancia de Hamming,

$$(\mathbf{c}, \mathbf{c}') \leq (\mathbf{c}, \mathbf{r}) + (\mathbf{r}, \mathbf{c}') \leq \tau + \tau = 2\tau.$$

Pero, como el código es MDS,

$$(\mathbf{c}, \mathbf{c}') \geq d = n - k + 1.$$

Como $\tau = \lfloor (n-k)/2 \rfloor$, se cumple

$$2\tau < n - k + 1.$$

Por consiguiente,

$$n - k + 1 \leq (\mathbf{c}, \mathbf{c}') \leq 2\tau < n - k + 1,$$

lo cual es una contradicción. Así, no puede existir una segunda palabra código a distancia menor o igual que τ de $\mathbf{r}$, y la palabra recuperada es única.

Finalmente, una vez recuperado g^*, se reconstruye

$$f^*(X) = g^*(X) + T^*(X),$$

y por tanto

$$\mathrm{ev}_{\boldsymbol{\alpha}}(f^*) = \mathbf{c}.$$

Esto concluye la corrección del algoritmo. □

A continuación se presenta el algoritmo general de decodificación por fuerza bruta para códigos Reed–Solomon torcidos (TRS). Este algoritmo enumera todas las posibles asignaciones de los coeficientes torcidos, reduce el problema a una decodificación Reed-Solomon clásica para cada caso, y selecciona la solución que satisface tanto la condición de distancia como la restricción de consistencia estructural del código. La correctitud del algoritmo descansa en el hecho de que la combinación correcta de coeficientes torcidos necesariamente será explorada, garantizando así la recuperación de la palabra código original siempre que el número de errores esté dentro del radio de decodificación.

Algoritmo 1: Decodificación por fuerza bruta de un código torcido Reed–Solomon MDS

1 *[1] q, n, k, puntos de evaluación $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, parámetros ℓ, t, h, η, palabra recibida* $\mathbf{r}$ *Palabra código recuperada* $\mathbf{c}$ *o fallo* $\mathcal{L} \leftarrow \varnothing$ **forall** $\boldsymbol{u} = (u_1, \ldots, u_\ell) \in \mathbb{F}_q^\ell$ **do**

2 *Construir el polinomio de torsión*

$$T_{\boldsymbol{u}}(X) \leftarrow \sum_{j=1}^{\ell} \eta_j u_j X^{k-1+t_j}$$

Calcular

$$\mathbf{r}' \leftarrow \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T_{\boldsymbol{u}})$$

Aplicar un decodificador Reed–Solomon clásico a $\mathbf{r}'$ *y obtener un polinomio* $g(X)$*, si existe* **if** $g(X)$ *existe* $\boldsymbol{y}$ $\deg(g) < k$ **then**

3 *Reconstruir*

$$f(X) \leftarrow g(X) + T_{\boldsymbol{u}}(X)$$

Calcular la candidata

$$\mathbf{c} \leftarrow \mathrm{ev}_{\boldsymbol{\alpha}}(f)$$

Verificar la condición de consistencia

$$\mathrm{coef}_{X^{h_j}}(g) = u_j, \qquad j = 1, \ldots, \ell$$

if $d_H(\mathbf{c}, \mathbf{r}) \leq \tau$ **then**

4 $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{c}\}$ **return** *la* ***única*** *palabra en* $\mathcal{L}$ *si* $|\mathcal{L}| = 1$*; en otro caso, fallo*

En el algoritmo anterior, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ denota el conjunto de puntos de evaluación del código, mientras que q es el tamaño del cuerpo finito $\mathbb{F}_q$. Los parámetros n y k son la longitud y la dimensión del código, respectivamente. El entero ℓ indica el número de términos torcidos, y los vectores $t = (t_1, \ldots, t_\ell)$, $h = (h_1, \ldots, h_\ell)$ y $\eta = (\eta_1, \ldots, \eta_\ell)$ determinan la estructura torcido del código.

Paso 1: enumeración de las tuplas candidatas. *El algoritmo recorre todas las posibles tuplas*

$$\boldsymbol{u} = (u_1, \ldots, u_\ell) \in \mathbb{F}_q^\ell.$$

Cada tupla representa una posible elección de los coeficientes torcidos ocultos del polinomio original. Como hay q^ℓ posibilidades, el algoritmo tiene naturaleza de fuerza bruta.

Paso 2: construcción del término de torsión. *Para cada tupla candidata $\boldsymbol{u}$, se construye el polinomio*

$$T_{\boldsymbol{u}}(X) = \sum_{j=1}^{\ell} \eta_j u_j X^{k-1+t_j}.$$

Este polinomio recoge exactamente la parte torcida del mensaje. Si la tupla es correcta, entonces $T_{\boldsymbol{u}}(X)$ coincide con el término de torsión verdadero del polinomio transmitido.

Paso 3: reducción al caso Reed–Solomon clásico. *Luego se calcula*

$$\mathbf{r}' = \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T_{\boldsymbol{u}}).$$

La idea es que, si $\boldsymbol{u}$ es la tupla correcta, entonces al restar la parte torcida de la palabra recibida, lo que queda es una palabra que difiere de una palabra Reed–Solomon clásica en un número pequeño de posiciones. Por eso se puede aplicar un decodificador estándar de Reed–Solomon.

Paso 4: decodificación Reed–Solomon. *El decodificador clásico intenta recuperar un polinomio $g(X)$ de grado menor que k tal que*

$$\mathrm{ev}_{\boldsymbol{\alpha}}(g)$$

esté a distancia pequeña de $\mathbf{r}'$. Si no existe tal polinomio, o si el resultado no tiene grado menor que k, entonces esa tupla candidata se descarta.

Paso 5: reconstrucción del polinomio torcido. *Si se obtiene un polinomio válido $g(X)$, se reconstruye el polinomio completo mediante*

$$f(X) = g(X) + T_{\boldsymbol{u}}(X).$$

Después se evalúa este polinomio para obtener una palabra candidata

$$\mathbf{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f).$$

Paso 6: verificación de consistencia. *La condición*

$$\mathrm{coef}_{X^{h_j}}(g) = u_j, \qquad j = 1, \ldots, \ell,$$

garantiza que la parte torcida reconstruida es coherente con el polinomio base $g(X)$. Esta verificación evita aceptar soluciones espurias que podrían aparecer por casualidad al decodificar el RS clásico.

Paso 7: criterio final de aceptación. *Finalmente, se comprueba que la palabra candidata esté a distancia de Hamming a lo sumo τ de la palabra recibida:*

$$d_H(\mathbf{c}, \mathbf{r}) \leq \tau.$$

Si esto ocurre, la candidata se guarda en el conjunto $\mathcal{L}$. Al terminar el proceso, si $\mathcal{L}$ contiene una única palabra, esa es la palabra código recuperada; si contiene más de una o

está vacía, el algoritmo falla.

5.1.3. Soluciones Espurias y Verificación de Consistencia

El polinomio torcido se define como $f(X) = \sum_{i=0}^{k-1} g_i X^i + \sum_{j=1}^{\ell} \eta_j g_{h_j} X^{k-1+t_j}$*, donde la restricción fundamental es que el coeficiente de* X^{k-1+t_j} *es* $\eta_j g_{h_j}$*, no un valor independiente. El algoritmo de fuerza bruta prueba cada* $u_j \in \mathbb{F}_q$ *como candidato para* g_{h_j}*, construye* $T(X) = \sum \eta_j u_j X^{k-1+t_j}$*, y aplica decodificación Reed-Solomon a* $\mathbf{r} - ev(T)$*. Sin embargo, el decodificador RS puede retornar un polinomio* $g(X)$ *con* $g_{h_j} = \gamma \neq u_j$*, produciendo un* $f(X)$ *que satisface la distancia mínima pero viola la restricción* $\eta_j u_j = \eta_j g_{h_j}$*. La verificación* $g_{h_j} = u_j$ *descarta estas soluciones espurias, garantizando que el polinomio reconstruido pertenece efectivamente a* $\mathcal{P}_{n,k}^{\mathbf{t},\mathbf{h},\boldsymbol{\eta}}$*.*

5.1.4. Ejemplos Detallados: Codificación y Decodificación de un Código Reed–Solomon torcido

A continuación se presentan dos ejemplos completamente desarrollados que ilustran cada paso del proceso de codificación y decodificación de un código Reed–Solomon torcidos (TRS). Se han elegido un ejemplos con parámetros pequeños para facilitar la comprensión, pero el procedimiento es generalizable a cualquier conjunto de parámetros.

Ejemplo 5.3 (Decodificación por fuerza bruta de un TRS MDS)**.** Sea $q = 5$, $F = \mathbb{F}_5$, $n = 4$, $k = 2$, y tomemos los puntos de evaluación

$$\boldsymbol{\alpha} = [0, 1, 2, 3].$$

Consideremos una sola torsión, con

$$\ell = 1, \qquad t = [1], \qquad h = [0], \qquad \eta = [4].$$

Con estos parámetros, la base torcida viene dada por

$$g_0(X) = 1 + 4X^2, \qquad g_1(X) = X.$$

Por evaluación en $\boldsymbol{\alpha}$, la matriz generadora del código TRS es

$$G = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}.$$

Al revisar todas las submatrices 2×2 de G, se verifica que son invertibles, por lo que el código es MDS.

Tomamos el mensaje

$$\boldsymbol{m} = (3, 1).$$

El polinomio torcido asociado es

$$f^*(X) = 2X^2 + X + 3,$$

y su evaluación da la palabra código

$$\mathbf{c} = \mathrm{ev}_{\boldsymbol{\alpha}}(f^*) = (3, 1, 3, 4).$$

Ahora introducimos un error de peso uno,

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

de modo que la palabra recibida es

$$\mathbf{r} = \mathbf{c} + \mathbf{e} = (3, 2, 3, 4).$$

Decodificación por fuerza bruta. Probamos todas las posibles tuplas para el coeficiente torcido. Como aquí $\ell = 1$, hay exactamente cinco posibilidades:

$$u \in \{0, 1, 2, 3, 4\}.$$

Para cada una, construimos

$$T(X) = \eta u X^2$$

y calculamos

$$\mathbf{r}' = \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T).$$

Luego intentamos decodificar $\mathbf{r}'$ como una palabra de un código Reed–Solomon clásico de dimensión $k = 2$.

- Para $u = 0$, se obtiene $T(X) = 0$ y $\mathbf{r}' = (3, 2, 3, 4)$, pero esta palabra no se puede decodificar como RS clásico con un solo error.
- Para $u = 1$, se obtiene $T(X) = 4X^2$ y $\mathbf{r}' = (3, 3, 2, 3)$, y tampoco se obtiene una decodificación válida.
- Para $u = 2$, se obtiene $T(X) = 3X^2$ y $\mathbf{r}' = (3, 4, 1, 2)$, sin decodificación válida.
- Para $u = 3$, se obtiene $T(X) = 2X^2$ y

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

  En este caso el decodificador Reed–Solomon recupera

  $$g(X) = X + 3.$$

  Entonces

  $$f(X) = g(X) + T(X) = 2X^2 + X + 3,$$

  y al evaluar se recupera exactamente

  $$\mathrm{ev}_{\boldsymbol{\alpha}}(f) = (3, 1, 3, 4).$$

Además, la distancia de Hamming entre la palabra recuperada y la recibida es

$$d_H(\mathbf{c}, \mathbf{r}) = 1.$$

La condición de consistencia también se cumple, ya que el coeficiente de $X^{h_1} = X^0$ en $g(X) = X + 3$ es precisamente $3 = u$.

- Para $u = 4$, se obtiene $T(X) = X^2$ y

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

El decodificador RS produce una candidata, pero la condición de consistencia falla porque el coeficiente de X^0 en el polinomio recuperado no coincide con $u = 4$.

Por tanto, la única solución válida es

$$u = [3], \qquad g(X) = X + 3, \qquad f(X) = 2X^2 + X + 3,$$

y la palabra código recuperada es

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

Esto muestra que la decodificación por fuerza bruta recupera de forma única la palabra original cuando el código TRS es MDS y el número de errores está dentro del radio de decodificación única.

Ejemplo 5.4. Trabajamos sobre el cuerpo finito $\mathbb{F}_5$ con las operaciones aritméticas módulo 5.

- **Tamaño del cuerpo:** $q = 5$
- **Longitud del código:** $n = 5$
- **Dimensión del código:** $k = 2$
- **Número de torsiones:** $\ell = 1$
- **Puntos de evaluación:** $\boldsymbol{\alpha} = [0, 1, 2, 3, 4]$
- **Vector de torsiones:** $\mathbf{t} = [1] \in \{1, \ldots, n-k\}^\ell = \{1, 2, 3\}^1$
- **Vector de enganche:** $\mathbf{h} = [0] \in \{0, \ldots, k-1\}^\ell = \{0, 1\}^1$
- **Vector de coeficientes:** $\boldsymbol{\eta} = [3] \in \mathbb{F}_5^\ell$

Verificación de la condición de pares distintos: El único par es $(h_1, t_1) = (0, 1)$. Como $\ell = 1$, trivialmente todos los pares son distintos.

Estructura del Polinomio torcido

Con estos parámetros, el conjunto de polinomios torcidos $\mathcal{P}_{5,2}^{[1],[0],[3]}$ está formado por todos los polinomios de la forma:

$$f(X) = \underbrace{f_0 + f_1 X}_{\text{polinomio base } g(X)} + \underbrace{\eta_1 \cdot f_{h_1} \cdot X^{k-1+t_1}}_{\text{término torcido } T(X)} \tag{5.1}$$

Sustituyendo los valores concretos:

$$f(X) = f_0 + f_1 X + 3 \cdot f_0 \cdot X^{2-1+1} = f_0 + f_1 X + 3 f_0 X^2 \tag{5.2}$$

donde $f_0, f_1 \in \mathbb{F}_5$ son los coeficientes del mensaje.

Observación fundamental: El coeficiente del término X^2 no es libre, sino que está determinado por f_0 (el coeficiente del término constante del polinomio base) multiplicado por $\eta_1 = 3$. Esta es la esencia de la "torsión": crea una dependencia entre los coeficientes.

Proceso de Codificación

Para codificar un mensaje, se sigue el siguiente procedimiento:

Paso 1: Selección del mensaje. Elegimos un polinomio base $g(X)$ de grado menor que $k = 2$. Tomamos como ejemplo:

$$g(X) = 1 + 2X \tag{5.3}$$

Es decir, los coeficientes del mensaje son $f_0 = 1$ y $f_1 = 2$.

Paso 2: Construcción del polinomio torcido. Identificamos el coeficiente que será "torcido". Como $h_1 = 0$, tomamos el coeficiente $f_0 = 1$ y construimos:

$$T(X) = \eta_1 \cdot f_{h_1} \cdot X^{k-1+t_1} = 3 \cdot f_0 \cdot X^2 = 3 \cdot 1 \cdot X^2 = 3X^2 \tag{5.4}$$

Paso 3: Construcción del polinomio torcido completo. Sumamos el polinomio base y el polinomio torcido:

$$f(X) = g(X) + T(X) = (1 + 2X) + 3X^2 = 1 + 2X + 3X^2 \tag{5.5}$$

Paso 4: Evaluación en los puntos de evaluación. Evaluamos $f(X)$ en cada $\alpha_i \in \{0, 1, 2, 3, 4\}$:

- $\alpha_1 = 0$: $f(0) = 1 + 2(0) + 3(0)^2 = 1$
- $\alpha_2 = 1$: $f(1) = 1 + 2(1) + 3(1)^2 = 1 + 2 + 3 = 6 \equiv 1 \pmod 5$
- $\alpha_3 = 2$: $f(2) = 1 + 2(2) + 3(2)^2 = 1 + 4 + 3(4) = 1 + 4 + 12 = 17 \equiv 2 \pmod 5$
- $\alpha_4 = 3$: $f(3) = 1 + 2(3) + 3(3)^2 = 1 + 6 + 3(9) = 1 + 6 + 27 = 34 \equiv 4 \pmod 5$
- $\alpha_5 = 4$: $f(4) = 1 + 2(4) + 3(4)^2 = 1 + 8 + 3(16) = 1 + 8 + 48 = 57 \equiv 2 \pmod 5$

Paso 5: Formación de la palabra código. La palabra código es el vector de evaluaciones:

$$\mathbf{c} = (c_1, c_2, c_3, c_4, c_5) = (1, 1, 2, 4, 2) \tag{5.6}$$

Nota 5.5. Es instructivo verificar que esta palabra código no pertenece a un código Reed-Solomon clásico de dimensión 2 sobre $\mathbb{F}_5$ con los mismos puntos de evaluación. En un RS clásico, cualquier conjunto de 2 posiciones determina unívocamente el polinomio. Si tomamos las posiciones 1 y 2 con valores 1 y 1, el polinomio RS sería $g(X) = 1$, que evaluado en $\alpha_3 = 2$ daría 1, no 2. Esto confirma que el código TRS es diferente del RS clásico.

Transmisión y Recepción con Errores

Supongamos que la palabra código $\mathbf{c} = (1, 1, 2, 4, 2)$ se transmite a través de un canal ruidoso y se recibe:

$$\mathbf{r} = (r_1, r_2, r_3, r_4, r_5) = (1, 1, 0, 4, 2) \tag{5.7}$$

Comparando con la palabra original:

- $r_1 = 1 = c_1$ (correcto)
- $r_2 = 1 = c_2$ (correcto)
- $r_3 = 0 \neq c_3 = 2$ (error)
- $r_4 = 4 = c_4$ (correcto)
- $r_5 = 2 = c_5$ (correcto)

Se ha producido exactamente un error en la tercera posición. El vector de error es $\mathbf{e} = (0, 0, -2, 0, 0) \equiv (0, 0, 3, 0, 0) \pmod 5$, con peso de Hamming $(\mathbf{e}) = 1$.

La distancia mínima del código es $d = n-k+1 = 4$, por lo que el radio de decodificación única es:

$$\tau = \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{3}{2} \right\rfloor = 1 \tag{5.8}$$

Como $(\mathbf{e}) = 1 \leq \tau$, la decodificación debería ser posible.

Proceso de Decodificación por Fuerza Bruta

El algoritmo de decodificación por fuerza bruta prueba todas las posibles combinaciones de los coeficientes torcidos. En este caso, como $\ell = 1$ y el enganche es $h_1 = 0$, debemos probar todos los valores posibles para $f_0 \in \mathbb{F}_5 = \{0, 1, 2, 3, 4\}$. Hay exactamente $q^\ell = 5^1 = 5$ combinaciones.

Iteración 1: Candidato $u_1 = 0$

(1) **Construcción del polinomio torcido candidato:**

$$T(X) = \eta_1 \cdot u_1 \cdot X^{k-1+t_1} = 3 \cdot 0 \cdot X^2 = 0 \tag{5.9}$$

(2) **Evaluación de $T(X)$ en los puntos α_i:**

$$\mathrm{ev}_{\boldsymbol{\alpha}}(T) = (T(0), T(1), T(2), T(3), T(4)) = (0, 0, 0, 0, 0) \tag{5.10}$$

(3) **Resta de la palabra recibida:**

$$\mathbf{r}' = \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T) = (1, 1, 0, 4, 2) - (0, 0, 0, 0, 0) = (1, 1, 0, 4, 2) \tag{5.11}$$

(4) **Decodificación Reed-Solomon de $\mathbf{r}'$:** Se intenta encontrar un polinomio $g(X)$ de grado < 2 tal que $(\mathbf{r}', \mathrm{ev}_{\boldsymbol{\alpha}}(g)) \leq 1$. Interpolando con los primeros $k = 2$ puntos $(0, 1)$ y $(1, 1)$ obtenemos $g(X) = 1$. Verificamos la distancia:

$$\mathrm{ev}_{\boldsymbol{\alpha}}(g) = (1, 1, 1, 1, 1) \tag{5.12}$$

$$(\mathbf{r}', \mathrm{ev}_{\boldsymbol{\alpha}}(g)) = ((1,1,0,4,2),(1,1,1,1,1)) = 3 > \tau = 1 \tag{5.13}$$

La decodificación RS falla para este candidato.

Iteración 2: Candidato $u_1 = 1$

(1) **Construcción del polinomio torcido candidato:**

$$T(X) = \eta_1 \cdot u_1 \cdot X^{k-1+t_1} = 3 \cdot 1 \cdot X^2 = 3X^2 \tag{5.14}$$

(2) **Evaluación de $T(X)$ en los puntos α_i:**

- $T(0) = 3 \cdot 0^2 = 0$
- $T(1) = 3 \cdot 1^2 = 3$
- $T(2) = 3 \cdot 2^2 = 3 \cdot 4 = 12 \equiv 2 \pmod 5$
- $T(3) = 3 \cdot 3^2 = 3 \cdot 9 = 27 \equiv 2 \pmod 5$
- $T(4) = 3 \cdot 4^2 = 3 \cdot 16 = 48 \equiv 3 \pmod 5$

$$\mathrm{ev}_{\boldsymbol{\alpha}}(T) = (0,3,2,2,3) \tag{5.15}$$

(3) **Resta de la palabra recibida:**

$$\mathbf{r}' = \mathbf{r} - \mathrm{ev}_{\boldsymbol{\alpha}}(T) = (1,1,0,4,2) - (0,3,2,2,3) \tag{5.16}$$

Calculando posición por posición en $\mathbb{F}_5$:

- $r'_1 = 1 - 0 = 1$
- $r'_2 = 1 - 3 = -2 \equiv 3 \pmod 5$
- $r'_3 = 0 - 2 = -2 \equiv 3 \pmod 5$
- $r'_4 = 4 - 2 = 2$
- $r'_5 = 2 - 3 = -1 \equiv 4 \pmod 5$

$$\mathbf{r}' = (1,3,3,2,4) \tag{5.17}$$

(4) **Decodificación Reed-Solomon de $\mathbf{r}'$:** Aplicamos el algoritmo de Gao. Tomamos los primeros $k = 2$ puntos para interpolar: $(0,1)$ y $(1,3)$. El polinomio de interpolación es:

$$g(X) = 1 + 2X \tag{5.18}$$

Verificamos la distancia de $\mathbf{r}'$ a $\mathrm{ev}_{\boldsymbol{\alpha}}(g)$:

$$\mathrm{ev}_{\boldsymbol{\alpha}}(g) = (g(0), g(1), g(2), g(3), g(4)) = (1,3,5,7,9) \equiv (1,3,0,2,4) \pmod 5 \tag{5.19}$$

Comparando con $\mathbf{r}' = (1,3,3,2,4)$:

- Posición 1: $1 = 1$
- Posición 2: $3 = 3$
- Posición 3: $3 \neq 0$ (diferencia)
- Posición 4: $2 = 2$
- Posición 5: $4 = 4$

Hay exactamente 1 discrepancia, por lo que $(\mathbf{r}', \mathrm{ev}_{\boldsymbol{\alpha}}(g)) = 1 \leq \tau = 1$. ¡La decodificación RS tiene éxito!

(5) **Reconstrucción del polinomio torcido completo:**

$$f(X) = g(X) + T(X) = (1 + 2X) + 3X^2 = 1 + 2X + 3X^2 \tag{5.20}$$

(6) **Evaluación del polinomio reconstruido:**

$$\mathbf{c}_{\text{decod}} = (f(0), f(1), f(2), f(3), f(4)) = (1, 1, 2, 4, 2) \tag{5.21}$$

(7) **Verificación de distancia a la palabra recibida:**

$$(\mathbf{c}_{\text{decod}}, \mathbf{r}) = ((1, 1, 2, 4, 2), (1, 1, 0, 4, 2)) = 1 \leq \tau = 1 \tag{5.22}$$

(8) **Verificación de consistencia de coeficientes:** El coeficiente g_0 del polinomio $g(X) = 1 + 2X$ es $g_0 = 1$. Como $h_1 = 0$, verificamos que $g_{h_1} = g_0 = 1 = u_1$. La condición se cumple.

¡Solución encontrada! La palabra código decodificada es $\mathbf{c}_{\text{decod}} = (1, 1, 2, 4, 2)$, que coincide exactamente con la palabra código original transmitida.

Iteraciones 3, 4 y 5: Candidatos $u_1 = 2, 3, 4$

Para completar el ejemplo, veamos qué sucede con los candidatos restantes:

- $u_1 = 2$**:** $T(X) = 3 \cdot 2 \cdot X^2 = 6X^2 \equiv X^2 \pmod 5$. Evaluando: $\text{ev}_{\boldsymbol{\alpha}}(T) = (0, 1, 4, 4, 1)$. Entonces $\mathbf{r}' = (1, 1, 0, 4, 2) - (0, 1, 4, 4, 1) = (1, 0, 1, 0, 1)$. Interpolando $(0, 1)$ y $(1, 0)$: $g(X) = 1 + 4X$. Verificando: $\text{ev}_{\boldsymbol{\alpha}}(g) = (1, 0, 4, 3, 2)$. Distancia a $\mathbf{r}'$: $((1, 0, 1, 0, 1), (1, 0, 4, 3, 2)) = 3 > 1$. FALLO.
- $u_1 = 3$**:** $T(X) = 3 \cdot 3 \cdot X^2 = 9X^2 \equiv 4X^2 \pmod 5$. Evaluando: $\text{ev}_{\boldsymbol{\alpha}}(T) = (0, 4, 1, 1, 4)$. $\mathbf{r}' = (1, 1, 0, 4, 2) - (0, 4, 1, 1, 4) = (1, 2, 4, 3, 3)$. Interpolando: $g(X) = 1 + X$. $\text{ev}_{\boldsymbol{\alpha}}(g) = (1, 2, 3, 4, 0)$. Distancia a $\mathbf{r}'$: $((1, 2, 4, 3, 3), (1, 2, 3, 4, 0)) = 3 > 1$. FALLO.
- $u_1 = 4$**:** $T(X) = 3 \cdot 4 \cdot X^2 = 12X^2 \equiv 2X^2 \pmod 5$. Evaluando: $\text{ev}_{\boldsymbol{\alpha}}(T) = (0, 2, 3, 3, 2)$. $\mathbf{r}' = (1, 1, 0, 4, 2) - (0, 2, 3, 3, 2) = (1, 4, 2, 1, 0)$. Interpolando: $g(X) = 1 + 3X$. $\text{ev}_{\boldsymbol{\alpha}}(g) = (1, 4, 2, 0, 3)$. Distancia a $\mathbf{r}'$: $((1, 4, 2, 1, 0), (1, 4, 2, 0, 3)) = 2 > 1$. FALLO.

Resultado Final

De las 5 iteraciones, solo el candidato $u_1 = 1$ produjo una decodificación exitosa. La lista de soluciones es:

$$\mathcal{L} = \{(1, 1, 2, 4, 2)\} \tag{5.23}$$

Se ha recuperado exactamente la palabra código original transmitida.

Resumen del Proceso

(1) El código TRS se define mediante los parámetros $(n, k, \ell, \boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta})$.
(2) **Codificación:** Dado un mensaje $\mathbf{m} = (m_0, \ldots, m_{k-1}) \in \mathbb{F}_q^k$, se construye el polinomio

torcido como:

$$f(X) = \sum_{i=0}^{k-1} m_i \cdot g_i(X), \quad \text{donde} \quad g_i(X) = X^i + \sum_{\substack{j=1 \\ h_j=i}}^{\ell} \eta_j \cdot X^{k-1+t_j} \tag{5.24}$$

y se obtiene la palabra código evaluando $\mathbf{c} = (f(\alpha_1), \ldots, f(\alpha_n))$.

(3) **Transmisión:** La palabra código $\mathbf{c}$ se envía por un canal ruidoso, resultando en $\mathbf{r} = \mathbf{c} + \mathbf{e}$.

(4) **Decodificación:** Se prueban todas las q^ℓ combinaciones de valores para los coeficientes f_{h_j}. Para cada combinación:
   (1) Se construye $T(X) = \sum \eta_j u_j X^{k-1+t_j}$
   (2) Se resta $\text{ev}_{\boldsymbol{\alpha}}(T)$ de $\mathbf{r}$
   (3) Se decodifica el resultado como RS clásico
   (4) Se reconstruye $f(X) = g(X) + T(X)$
   (5) Se verifican las condiciones de distancia y consistencia

(5) La combinación correcta ($u_j = f_{h_j}$) necesariamente produce una decodificación exitosa, garantizando la recuperación del mensaje original.

Observaciones sobre el Coeficiente η_j

En este ejemplo, $\eta_1 = 3$ juega un papel fundamental:

- En la **codificación**, determina la magnitud del término torcido: $3 \cdot f_0 \cdot X^2$. Si η_1 fuera diferente, la palabra código sería distinta para el mismo mensaje.
- En la **decodificación**, η_1 aparece explícitamente en la construcción de $T(X) = \eta_1 \cdot u_1 \cdot X^{k-1+t_1}$ para cada candidato u_1. Es esencial usar el valor correcto de η_1; de lo contrario, la resta $\mathbf{r} - \text{ev}_{\boldsymbol{\alpha}}(T)$ no eliminaría correctamente el efecto de la torsión.
- En la **verificación de consistencia**, aunque no se menciona explícitamente, la estructura impuesta por η_1 está implícita en la propia definición del código. El polinomio reconstruido $f(X) = g(X) + \eta_1 u_1 X^{k-1+t_1}$ debe satisfacer que el coeficiente de X^{k-1+t_1} sea exactamente $\eta_1 \cdot g_{h_1}$, lo cual se verifica al comprobar que $g_{h_1} = u_1$.

Sin el coeficiente η_j, no se podría distinguir entre diferentes códigos TRS que comparten los mismos $\mathbf{t}$ y $\mathbf{h}$ pero difieren en $\boldsymbol{\eta}$, lo que demuestra que $\boldsymbol{\eta}$ es un parámetro esencial en la definición y manipulación de estos códigos.

Capítulo 6

Sistema Criptográfico de McEliece y la Criptografía Poscuántica

6.1. El sistema criptográfico de McEliece

*El **criptosistema de McEliece** fue propuesto originalmente por Robert J. McEliece en 1978 [10]. Este es un esquema de clave pública basado en la dificultad del problema de decodificación de códigos lineales en el caso general. Su construcción se fundamentó inicialmente en el uso de códigos de Goppa binarios como núcleo, aunque funciona sobre cualquier familia de códigos que disponga de un algoritmo de decodificación eficiente.*

*A diferencia de los criptosistemas clásicos de clave pública, como RSA o el protocolo de Diffie–Hellman, el criptosistema de McEliece no ha sucumbido a los ataques cuánticos conocidos. En particular, el algoritmo de Shor [11], que resuelve ambos problemas en tiempo polinomial sobre una computadora cuántica, no tiene incidencia sobre la dificultad de decodificar un código lineal aleatorio. Esta característica convierte al esquema de McEliece en uno de los candidatos más sólidos dentro de la **criptografía poscuántica**, rama que se ocupa del diseño de sistemas criptográficos resistentes frente a adversarios equipados con computadoras cuánticas. De hecho, el criptosistema de McEliece fue considerado por el Instituto Nacional de Estándares y Tecnología (NIST) durante su proceso de estandarización de algoritmos poscuánticos iniciado en 2017 [12].*

Definición 6.1. Sea $C \subseteq \mathbb{F}_q^n$ un código lineal $[n, k, d]_q$ con capacidad de corrección de errores t. El criptosistema de McEliece se define mediante los siguientes pasos:

(1) **Generación de claves:**

Sea $G \in \mathbb{F}_q^{k\times n}$ una matriz generadora del código C. Se eligen al azar:

- $S \in GL_k(\mathbb{F}_q)$, una matriz invertible.
- $P \in \mathbb{F}_q^{n\times n}$, una matriz de permutación.

La clave pública se define como la matriz

$$G' = SGP.$$

La clave pública es (G', t), mientras que la clave privada es (S, G, P).

(2) **Cifrado:**

Dado un mensaje $m \in \mathbb{F}_q^k$, el cifrado se calcula como

$$c = mG' + e,$$

donde $e \in \mathbb{F}_q^n$ es un vector de error aleatorio con peso de Hamming $\leq t$.

(3) **Descifrado:**

El receptor recibe $c \in \mathbb{F}_q^n$ y, con la clave privada, obtiene

$$cP^{-1} = mSG + eP^{-1}.$$

Dado que G corresponde a un código con decodificación eficiente, se recupera mS. Finalmente, el mensaje original se obtiene aplicando

$$m = (mS)S^{-1}.$$

En la actualidad, este sistema se considera un criptosistema poscuántico, ya que se cree resistente incluso frente a ataques realizados mediante computadoras cuánticas. Esta propiedad ha hecho que el sistema de McEliece sea uno de los candidatos en el proceso de estandarización de la criptografía poscuántica llevado a cabo por el ***NIST*** *(National Institute of Standards and Technology), consolidando su relevancia como uno de los esquemas más prometedores para garantizar la seguridad en la era poscuántica.*

6.2. El sistema criptográfico de McEliece sobre códigos Reed-Solomon torcidos

La aplicación del criptosistema de McEliece a los códigos Reed-Solomon torcidos constituye una extensión natural de la construcción original. A continuación se describe la adaptación del esquema a esta familia de códigos.

6.2.1. Construcción del esquema McEliece-TRS

Sea $\mathcal{C} = \mathcal{C}_{n,k}^{\boldsymbol{\alpha},\mathbf{t},\mathbf{h},\boldsymbol{\eta}} \subseteq \mathbb{F}_q^n$ un código Reed-Solomon torcido con parámetros (n, k, ℓ), puntos de evaluación $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, vector de torsiones $\mathbf{t} \in \{1, \ldots, n-k\}^\ell$, vector de enganche $\mathbf{h} \in \{0, \ldots, k-1\}^\ell$ y vector de coeficientes $\boldsymbol{\eta} \in \mathbb{F}_q^\ell$. Se asume que el código es MDS, por lo que su distancia mínima es $d = n - k + 1$ y su capacidad de corrección de errores es $t = \lfloor (d-1)/2 \rfloor = \lfloor (n-k)/2 \rfloor$.

(1) ***Generación de claves.*** *Sea $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ una matriz generadora del código $\mathcal{C}$, cuyas filas corresponden a la evaluación de los polinomios base $g_i(X)$ en los puntos de evaluación:*

$$\mathbf{G} = \begin{pmatrix} g_0(\alpha_1) & g_0(\alpha_2) & \cdots & g_0(\alpha_n) \\ g_1(\alpha_1) & g_1(\alpha_2) & \cdots & g_1(\alpha_n) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1}(\alpha_1) & g_{k-1}(\alpha_2) & \cdots & g_{k-1}(\alpha_n) \end{pmatrix} \tag{6.1}$$

donde cada polinomio base se define como

$$g_i(X) = X^i + \sum_{\substack{j=1 \\ h_j = i}}^{\ell} \eta_j \cdot X^{k-1+t_j}, \quad i = 0, 1, \ldots, k-1. \tag{6.2}$$

Se seleccionan aleatoriamente dos matrices:

- $\mathbf{S} \in GL_k(\mathbb{F}_q)$, *una matriz invertible de tamaño* $k \times k$.
- $\mathbf{P} \in \mathbb{F}_q^{n \times n}$, *una matriz de permutación.*

La ***clave pública*** *es el par* $(\mathbf{G}_{pub}, t)$, *donde*

$$\mathbf{G}_{pub} = \mathbf{SGP}. \tag{6.3}$$

La ***clave privada*** *es la tupla* $(\mathbf{S}, \boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}, \mathbf{P})$. *El conocimiento de los parámetros* $\boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}$ *permite al receptor legítimo emplear el algoritmo de decodificación eficiente del código TRS.*

(2) ***Cifrado.*** *Para transmitir un mensaje* $\mathbf{m} \in \mathbb{F}_q^k$, *el remitente calcula:*

$$\mathbf{c} = \mathbf{m}\mathbf{G}_{pub} + \mathbf{e}, \tag{6.4}$$

donde $\mathbf{e} \in \mathbb{F}_q^n$ *es un vector de error generado aleatoriamente con peso de Hamming* $(\mathbf{e}) \leq t$. *El texto cifrado es el vector* $\mathbf{c} \in \mathbb{F}_q^n$.

(3) ***Descifrado.*** *El receptor legítimo, en posesión de la clave privada, ejecuta los siguientes pasos:*

(1) *Deshacer la permutación:*

$$\mathbf{c}' = \mathbf{c}\mathbf{P}^{-1}. \tag{6.5}$$

Puesto que $\mathbf{P}$ *es una matriz de permutación, se tiene que* $\mathbf{P}^{-1} = \mathbf{P}^T$. *Sustituyendo la expresión de* $\mathbf{c}$*:*

$$\begin{aligned} \mathbf{c}' &= (\mathbf{mSGP} + \mathbf{e})\mathbf{P}^{-1} \\ &= \mathbf{mSG} + \mathbf{e}\mathbf{P}^{-1}. \end{aligned} \tag{6.6}$$

(2) *Decodificación del código TRS: El vector* $\mathbf{c}'$ *corresponde a una palabra del código* $\mathcal{C}$ *(asociada al mensaje* $\mathbf{mS}$*) que ha sido perturbada por el error* $\mathbf{e}\mathbf{P}^{-1}$. *Dado que* $\mathbf{P}^{-1}$ *es una permutación, el peso del error se conserva:*

$$(\mathbf{e}\mathbf{P}^{-1}) = (\mathbf{e}) \leq t. \tag{6.7}$$

En consecuencia, el error se encuentra dentro de la capacidad de corrección del código. Se aplica el algoritmo de decodificación del código TRS (por ejemplo, el decodificador por fuerza bruta descrito en la Sección 5.1 sobre $\mathbf{c}'$*, obteniendo el mensaje intermedio:*

$$\mathbf{m}' = \mathbf{mS}. \tag{6.8}$$

(3) *Recuperación del mensaje original: Finalmente, se aplica la inversa de la matriz* $\mathbf{S}$*:*

$$\mathbf{m} = \mathbf{m}'\mathbf{S}^{-1}. \tag{6.9}$$

6.2.2. Consideraciones de seguridad

La seguridad del esquema McEliece-TRS se fundamenta en dos problemas computacionales cuya resolución se presume difícil:

(1) ***Problema de decodificación por síndrome (SD).*** *Dado un código lineal definido por una matriz de paridad* $\mathbf{H}$ *y un síndrome* $\mathbf{s}$*, encontrar un vector de error* $\mathbf{e}$ *con peso menor o igual a* t *tal que* $\mathbf{H}\mathbf{e}^T = \mathbf{s}$*. Este problema es NP-completo en el caso general y constituye la base de la seguridad del criptosistema de McEliece.*

(2) ***Problema de distinguibilidad.*** *Determinar si la matriz pública* $\mathbf{G}_{pub}$ *proviene de un código Reed-Solomon torcido o de un código lineal aleatorio. La introducción de los términos de torsión modifica la estructura algebraica del código, dificultando la aplicación de ataques estructurales como el de Sidelnikov-Shestakov [13], que resultó exitoso contra los códigos Reed-Solomon clásicos.*

La resistencia frente al ataque de Sidelnikov-Shestakov merece atención particular. En los códigos Reed-Solomon clásicos, la matriz generadora posee una estructura de Vandermonde generalizada que permite recuperar los puntos de evaluación a partir de la clave pública. En los códigos TRS, la presencia de los monomios adicionales X^{k-1+t_j} *en los polinomios base altera significativamente esta estructura. Las filas de la matriz generadora ya no corresponden a la evaluación de monomios puros, sino a polinomios que incorporan términos de grado superior al umbral* $k-1$*. Esta modificación estructural busca imposibilitar la recuperación directa de los parámetros secretos del código a partir de* $\mathbf{G}_{pub}$*.*

No obstante, debe señalarse que la seguridad de los códigos TRS en contextos criptográficos es un área de investigación activa. Si bien los resultados iniciales sugieren una resistencia mejorada frente a ataques estructurales en comparación con los Reed-Solomon clásicos, se requiere un análisis más profundo para determinar la solidez del esquema frente a técnicas avanzadas de criptoanálisis, incluyendo ataques basados en bases de Gröbner y métodos de álgebra lineal estructurada.

6.2.3. Complejidad computacional

La eficiencia del esquema depende del costo de la decodificación TRS. La notación $O(\cdot)$ *empleada a continuación es la notación de Landau, que describe el comportamiento asintótico en el peor caso del número de operaciones elementales sobre* $\mathbb{F}_q$ *en función del tamaño de la entrada; una introducción formal puede consultarse en [14].*

Para el algoritmo de fuerza bruta descrito previamente, el descifrado requiere q^ℓ *iteraciones, cada una con una decodificación Reed–Solomon cuya complejidad, mediante el algoritmo de Berlekamp–Massey con técnicas de multiplicación rápida de polinomios [8, 15], es* $O(n \log^2 n \log\log n)$*. Por consiguiente, la complejidad total del descifrado es:*

$$O\Big(q^\ell \cdot n \log^2 n \log\log n\Big)\,. \tag{6.10}$$

El factor q^ℓ *es exponencial en* ℓ *y en* $\log q$*, mientras que* $n \log^2 n \log\log n$ *es prácticamente lineal en* n*; el primero domina ampliamente para parámetros criptográficos típicos. Esta cota impone una restricción práctica sobre el número de torsiones* ℓ *que pueden emplearse:*

para mantener la eficiencia en el descifrado, ℓ debe ser pequeño (típicamente $\ell \leq 3$), lo cual reduce el espacio de claves posibles a q^ℓ variantes del código para parámetros (n, k) fijos. La elección de los parámetros debe equilibrar la eficiencia computacional con la seguridad requerida.

6.2.4. Ejemplo numérico del esquema McEliece-TRS

A continuación se presenta un ejemplo ilustrativo del criptosistema McEliece aplicado a un código Reed-Solomon torcido. Por razones de claridad expositiva, se han seleccionado parámetros reducidos que permiten verificar manualmente cada etapa del proceso. Los cálculos se han implementado en SageMath y se reproducen a continuación.

Parámetros del código y verificación MDS

Se trabaja sobre el cuerpo finito $\mathbb{F}_5$. Los parámetros del código TRS son:

- $q = 5$, $n = 4$, $k = 2$, $\ell = 1$
- *Puntos de evaluación:* $\boldsymbol{\alpha} = [0, 1, 2, 3]$
- *Vector de torsiones:* $\mathbf{t} = [1]$
- *Vector de ganchos:* $\mathbf{h} = [0]$
- *Vector de coeficientes:* $\boldsymbol{\eta} = [4]$

Los polinomios base del espacio torcido son $g_0(X) = 4X^2 + 1$ y $g_1(X) = X$. La matriz generadora se obtiene evaluando estos polinomios en los puntos de evaluación:

$$\mathbf{G} = \begin{pmatrix} g_0(0) & g_0(1) & g_0(2) & g_0(3) \\ g_1(0) & g_1(1) & g_1(2) & g_1(3) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 3 \end{pmatrix}. \tag{6.11}$$

Para verificar la propiedad MDS se emplea el criterio de la Observación 3.16: se calculan los $\binom{4}{2} = 6$ menores 2×2 de $\mathbf{G}$, obteniendo determinantes $1, 2, 3, 3, 3, 2$, todos no nulos en $\mathbb{F}_5$. En consecuencia, el código es MDS con distancia mínima $d = n - k + 1 = 3$ y capacidad de corrección $t = \lfloor (3-1)/2 \rfloor = 1$.

Generación de claves

Se seleccionan la matriz invertible $\mathbf{S}$ y la matriz de permutación $\mathbf{P}$ siguientes:

$$\mathbf{S} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{6.12}$$

El determinante de $\mathbf{S}$ es $\det(\mathbf{S}) = 2 \cdot 1 - 1 \cdot 1 = 1 \neq 0$ en $\mathbb{F}_5$, lo que garantiza su invertibilidad; su inversa es $\mathbf{S}^{-1} = \begin{pmatrix} 1 & 4 \\ 4 & 2 \end{pmatrix}$. La clave pública se calcula como:

$$\mathbf{G}_{pub} = \mathbf{SGP} = \begin{pmatrix} 1 & 2 & 2 & 1 \\ 1 & 1 & 0 & 4 \end{pmatrix}. \tag{6.13}$$

La clave privada es la tupla $(\mathbf{S}, \boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}, \mathbf{P})$.

Cifrado

Se toma el mensaje $\mathbf{m} = (1, 3) \in \mathbb{F}_5^2$ *y se genera un vector de error* $\mathbf{e} = (0, 0, 2, 0)$ *con peso de Hamming* $1 \leq t$*. El texto cifrado es:*

$$\mathbf{c} = \mathbf{m}\mathbf{G}_{pub} + \mathbf{e} = (1, 3) \begin{pmatrix} 1 & 2 & 2 & 1 \\ 1 & 1 & 0 & 4 \end{pmatrix} + (0, 0, 2, 0) = (4, 0, 4, 3). \tag{6.14}$$

Descifrado

El receptor legítimo ejecuta los siguientes pasos:

Paso 1: *Deshace la permutación multiplicando por* $\mathbf{P}^{-1}$*:*

$$\mathbf{c}' = \mathbf{c}\mathbf{P}^{-1} = (4, 0, 4, 3)\mathbf{P}^{-1} = (0, 4, 3, 4). \tag{6.15}$$

Se verifica que $\mathbf{c}' = \mathbf{m}\mathbf{S}\mathbf{G} + \mathbf{e}\mathbf{P}^{-1}$*, donde* $\mathbf{m}\mathbf{S} = (1, 3)\mathbf{S} = (0, 4)$ *y* $\mathbf{e}\mathbf{P}^{-1} = (0, 0, 0, 2)$*.*

Paso 2: *Se aplica el algoritmo de decodificación por fuerza bruta a* $\mathbf{c}'$*. El algoritmo prueba las* 5 *combinaciones posibles para* $u_1 \in \mathbb{F}_5$ *y encuentra una única solución consistente:* $u_1 = 0$*, que corresponde al polinomio* $g(X) = 4X$*. Se recupera así* $\mathbf{m}\mathbf{S} = (0, 4)$*.*

Paso 3: *Se multiplica por* $\mathbf{S}^{-1}$ *para obtener el mensaje original:*

$$\mathbf{m} = (0, 4)\mathbf{S}^{-1} = (0, 4) \begin{pmatrix} 1 & 4 \\ 4 & 2 \end{pmatrix} = (1, 3). \tag{6.16}$$

El mensaje recuperado coincide con el mensaje original, verificando la correctitud del esquema McEliece-TRS.

Código SageMath

```
import itertools

q = 5; n = 4; k = 2; ell = 1
F = GF(q)
P.<X> = PolynomialRing(F)

alphas = [F(0), F(1), F(2), F(3)]
t_vec = [1]; h_vec = [0]; eta_vec = [F(4)]

# Construccion de la matriz generadora
def construir_polinomio_base(i):
    g = X^i
    for j in range(ell):
        if h_vec[j] == i:
            g += eta_vec[j] * X^(k - 1 + t_vec[j])
    return g

pols_base = [construir_polinomio_base(i) for i in range(k)]
G = matrix(F, k, n, lambda i, j: pols_base[i](alphas[j]))
```

```
20
21 # Verificacion MDS: todos los menores k x k no nulos
22 from itertools import combinations
23 todos_no_nulos = all(det(G.matrix_from_columns(c)) != 0
24                      for c in combinations(range(n), k))
25 assert todos_no_nulos, "El codigo no es MDS"
26
27 # Claves
28 S = matrix(F, [[2, 1], [1, 1]])
29 P_mat = matrix(F, [[0,1,0,0], [1,0,0,0], [0,0,0,1], [0,0,1,0]])
30 G_pub = S * G * P_mat
31
32 # Cifrado
33 m = vector(F, [1, 3])
34 e = vector(F, [0, 0, 2, 0])
35 c = m * G_pub + e
36
37 # Descifrado
38 c_prima = c * P_mat^(-1)
39
40 # Decodificacion TRS por fuerza bruta
41 for u_combo in itertools.product(F, repeat=ell):
42     T = sum(eta_vec[j] * F(u_combo[j]) * X^(k-1+t_vec[j])
43             for j in range(ell))
44     T_eval = [T(alpha) for alpha in alphas]
45     r_prima = [c_prima[i] - T_eval[i] for i in range(n)]
46
47     puntos = [(alphas[i], r_prima[i]) for i in range(k)]
48     g = P.lagrange_polynomial(puntos)
49
50     if g.degree() < k:
51         f = g + T
52         c_candidata = [f(alpha) for alpha in alphas]
53         dist = sum(1 for i in range(n) if c_candidata[i] !=
   c_prima[i])
54
55         if dist <= 1:
56             coeffs = g.coefficients(sparse=False)
57             while len(coeffs) < k:
58                 coeffs.append(F(0))
59             if all(coeffs[h_vec[j]] == F(u_combo[j]) for j in
   range(ell)):
60                 mS_recuperado = vector(F, coeffs[:k])
61                 m_recuperado = mS_recuperado * S^(-1)
62                 break
```

```
63
64 print(f"Mensaje original:    {m}")
65 print(f"Mensaje recuperado: {m_recuperado}")
66 print(f"Exito: {m == m_recuperado}")
```

Listado 6.1: Implementación del esquema McEliece-TRS en SageMath

La ejecución del código confirma que el mensaje original $\mathbf{m} = (1, 3)$ *se recupera correctamente, validando el funcionamiento del esquema McEliece basado en códigos Reed-Solomon torcidos.*

6.2.5. Ejemplo numérico del esquema McEliece-TRS con dos torsiones

A continuación se presenta un ejemplo ilustrativo del criptosistema McEliece aplicado a un código Reed-Solomon torcido con $\ell = 2$ *torsiones. Los parámetros se han seleccionado de forma que el código resultante sea MDS, garantizando así la máxima capacidad de corrección de errores. Los cálculos se han implementado en SageMath y se reproducen paso a paso.*

Parámetros del código y verificación MDS

Se trabaja sobre el cuerpo finito $\mathbb{F}_5$*. Los parámetros del código TRS son:*

- $q = 5$, $n = 5$, $k = 2$, $\ell = 2$
- *Puntos de evaluación:* $\boldsymbol{\alpha} = [0, 1, 2, 3, 4]$
- *Vector de torsiones:* $\mathbf{t} = [1, 2]$
- *Vector de ganchos:* $\mathbf{h} = [0, 1]$
- *Vector de coeficientes:* $\boldsymbol{\eta} = [2, 3]$

Se verifica en primer lugar que los pares (h_i, t_i) *son distintos:* $(h_1, t_1) = (0, 1)$ *y* $(h_2, t_2) = (1, 2)$*. Los polinomios base del espacio torcido se construyen como:*

$$g_0(X) = X^0 + \eta_1 X^{k-1+t_1} = 1 + 2X^2, \tag{6.17}$$

$$g_1(X) = X^1 + \eta_2 X^{k-1+t_2} = X + 3X^3. \tag{6.18}$$

La matriz generadora se obtiene evaluando estos polinomios en los puntos de evaluación:

$$\mathbf{G} = \begin{pmatrix} g_0(0) & g_0(1) & g_0(2) & g_0(3) & g_0(4) \\ g_1(0) & g_1(1) & g_1(2) & g_1(3) & g_1(4) \end{pmatrix} = \begin{pmatrix} 1 & 3 & 4 & 4 & 3 \\ 0 & 4 & 1 & 4 & 1 \end{pmatrix}. \tag{6.19}$$

Para verificar la propiedad MDS se aplica el criterio de la Observación 3.16: se calculan los $\binom{5}{2} = 10$ *menores* 2×2 *de* $\mathbf{G}$*. Los determinantes obtenidos son* $4, 1, 4, 1, 2, 1, 1, 2, 1, 2$*, todos no nulos en* $\mathbb{F}_5$*. Por consiguiente, el código es MDS con distancia mínima* $d = n - k + 1 = 4$ *y capacidad de corrección* $t = \lfloor (4 - 1)/2 \rfloor = 1$*.*

Generación de claves

Se seleccionan la matriz invertible $\mathbf{S}$ *y la matriz de permutación* $\mathbf{P}$ *siguientes:*

$$\mathbf{S} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.20}$$

El determinante de $\mathbf{S}$ *es* $\det(\mathbf{S}) = 2 \cdot 2 - 1 \cdot 1 = 3 \neq 0$ *en* $\mathbb{F}_5$*, lo que garantiza su invertibilidad. Su inversa es:*

$$\mathbf{S}^{-1} = \begin{pmatrix} 4 & 3 \\ 3 & 4 \end{pmatrix}. \tag{6.21}$$

La matriz $\mathbf{P}$ *es simétrica y satisface* $\mathbf{P}^{-1} = \mathbf{P}$*; su efecto consiste en intercambiar las dos primeras posiciones y las posiciones tercera y cuarta, manteniendo fija la quinta. La clave pública se calcula como:*

$$\mathbf{G}_{pub} = \mathbf{SGP} = \begin{pmatrix} 0 & 2 & 2 & 4 & 2 \\ 1 & 1 & 2 & 1 & 0 \end{pmatrix}. \tag{6.22}$$

La clave privada es la tupla $(\mathbf{S}, \boldsymbol{\alpha}, \mathbf{t}, \mathbf{h}, \boldsymbol{\eta}, \mathbf{P})$.

Cifrado

Se toma el mensaje $\mathbf{m} = (1, 4) \in \mathbb{F}_5^2$ *y se genera un vector de error aleatorio* $\mathbf{e} = (0, 0, 3, 0, 0)$ *con peso de Hamming* $1 \leq t$*. El texto cifrado se obtiene como:*

$$\mathbf{c} = \mathbf{m}\mathbf{G}_{pub} + \mathbf{e} = (1, 4) \begin{pmatrix} 0 & 2 & 2 & 4 & 2 \\ 1 & 1 & 2 & 1 & 0 \end{pmatrix} + (0, 0, 3, 0, 0) = (4, 1, 3, 3, 2). \tag{6.23}$$

Descifrado

El receptor legítimo, que posee la clave privada, ejecuta los siguientes pasos:

Paso 1: Deshacer la permutación. *Se multiplica el texto cifrado por* $\mathbf{P}^{-1}$*:*

$$\mathbf{c}' = \mathbf{c}\mathbf{P}^{-1} = (4, 1, 3, 3, 2) \cdot \mathbf{P} = (1, 4, 3, 3, 2). \tag{6.24}$$

Se verifica algebraicamente que $\mathbf{c}' = \mathbf{mSG} + \mathbf{e}\mathbf{P}^{-1}$*. En efecto,* $\mathbf{mS} = (1, 4) \cdot \mathbf{S} = (1 \cdot 2 + 4 \cdot 1, 1 \cdot 1 + 4 \cdot 2) = (1, 4)$*,* $\mathbf{e}\mathbf{P}^{-1} = (0, 0, 0, 3, 0)$*, y:*

$$\begin{aligned} \mathbf{mSG} + \mathbf{e}\mathbf{P}^{-1} &= (1, 4) \begin{pmatrix} 1 & 3 & 4 & 4 & 3 \\ 0 & 4 & 1 & 4 & 1 \end{pmatrix} + (0, 0, 0, 3, 0) \\ &= (1, 4, 3, 0, 2) + (0, 0, 0, 3, 0) \\ &= (1, 4, 3, 3, 2). \end{aligned} \tag{6.25}$$

que coincide con $\mathbf{c}'$*. Nótese que el error permutado tiene peso* 1*, preservando la condición de decodificabilidad.*

Paso 2: Decodificación TRS. *Se aplica el algoritmo de decodificación por fuerza bruta a* $\mathbf{c}'$*. El algoritmo prueba las* $q^\ell = 5^2 = 25$ *combinaciones posibles para el vector* $\mathbf{u} = (u_1, u_2) \in \mathbb{F}_5^2$*. Para cada candidato, construye el polinomio de torsión:*

$$T_{\mathbf{u}}(X) = \eta_1 u_1 X^{k-1+t_1} + \eta_2 u_2 X^{k-1+t_2} = 2u_1 X^2 + 3u_2 X^3, \tag{6.26}$$

lo evalúa en los puntos $\boldsymbol{\alpha}$*, lo resta de* $\mathbf{c}'$ *para obtener* $\mathbf{r}' = \mathbf{c}' - ev_{\boldsymbol{\alpha}}(T_{\mathbf{u}})$*, y aplica un decodificador Reed-Solomon clásico a* $\mathbf{r}'$*.*

De las 25 combinaciones, tres satisfacen la condición de distancia $(ev_{\boldsymbol{\alpha}}(f), \mathbf{c}') \leq t = 1$*:*

(1) $\mathbf{u} = (1, 3)$*: produce* $g(X) = 2X + 1$*, pero la verificación de consistencia falla porque* $g_1 = 2 \neq 3 = u_2$*.*

(2) $\mathbf{u} = (4, 0)$*: produce* $g(X) = 1$*, pero la verificación de consistencia falla porque* $g_0 = 1 \neq 4 = u_1$*.*

(3) $\mathbf{u} = (1, 4)$*: produce* $g(X) = 4X + 1$*, y la verificación de consistencia es exitosa porque* $g_0 = 1 = u_1$ *y* $g_1 = 4 = u_2$*.*

Únicamente la tercera combinación satisface simultáneamente la condición de distancia y la restricción de consistencia, por lo que es la única solución aceptada. El polinomio torcido reconstruido es:

$$\begin{aligned} f(X) &= g(X) + T_{\mathbf{u}}(X) \\ &= (4X + 1) + (2 \cdot 1 \cdot X^2 + 3 \cdot 4 \cdot X^3) \\ &= 1 + 4X + 2X^2 + 12X^3 \\ &\equiv 1 + 4X + 2X^2 + 2X^3 \pmod 5. \end{aligned} \tag{6.27}$$

A partir de los coeficientes de $g(X)$ *se recupera* $\mathbf{mS} = (g_0, g_1) = (1, 4)$*, que coincide con el valor esperado.*

Paso 3: Recuperación del mensaje original. *Se multiplica por la inversa de* $\mathbf{S}$*:*

$$\mathbf{m} = (\mathbf{mS}) \cdot \mathbf{S}^{-1} = (1, 4) \cdot \begin{pmatrix} 4 & 3 \\ 3 & 4 \end{pmatrix} = (1 \cdot 4 + 4 \cdot 3, 1 \cdot 3 + 4 \cdot 4) = (16, 19) \equiv (1, 4) \pmod 5. \tag{6.28}$$

El mensaje recuperado coincide con el mensaje original $\mathbf{m} = (1, 4)$*, verificando la correctitud del esquema McEliece basado en códigos Reed-Solomon torcidos con dos torsiones.*

Análisis de las soluciones espurias descartadas

Este ejemplo ilustra de manera concreta la importancia de la verificación de consistencia. De las tres combinaciones que pasaron el filtro de distancia, dos eran soluciones espurias:

- *Para* $\mathbf{u} = (1, 3)$*, el decodificador RS encontró* $g(X) = 2X + 1$ *a distancia 1 de* $\mathbf{r}'$*. Sin embargo,* $g_1 = 2 \neq 3 = u_2$*, por lo que el polinomio reconstruido* $f(X) = 1 + 2X + 2X^2 + 9X^3$ *tiene un término cúbico* $9X^3 \equiv 4X^3$ *cuyo coeficiente* 4 *no coincide con* $\eta_2 \cdot g_1 = 3 \cdot 2 = 6 \equiv 1 \pmod 5$*, violando la estructura del código TRS.*
- *Para* $\mathbf{u} = (4, 0)$*, se obtuvo* $g(X) = 1$ *con* $g_0 = 1 \neq 4 = u_1$*. El polinomio reconstruido*

$f(X) = 1 + 8X^2 + 0X^3 \equiv 1 + 3X^2$ *tiene coeficiente cuadrático* 3, *que no coincide con* $\eta_1 \cdot g_0 = 2 \cdot 1 = 2$.

En ambos casos, la verificación de consistencia detectó la incoherencia y descartó correctamente estas soluciones, garantizando que solo se aceptaran palabras código legítimas del código TRS.

Implementación en SageMath

```
import itertools

q = 5; n = 5; k = 2; ell = 2
F = GF(q)
P.<X> = PolynomialRing(F)

alphas = [F(0), F(1), F(2), F(3), F(4)]
t_vec = [1, 2]
h_vec = [0, 1]
eta_vec = [F(2), F(3)]

# Construccion de polinomios base
def construir_polinomio_base(i):
    g = X^i
    for j in range(ell):
        if h_vec[j] == i:
            g += eta_vec[j] * X^(k - 1 + t_vec[j])
    return g

pols_base = [construir_polinomio_base(i) for i in range(k)]
G = matrix(F, k, n, lambda i, j: pols_base[i](alphas[j]))

# Verificacion MDS
from itertools import combinations
es_MDS = all(det(G.matrix_from_columns(c)) != 0
             for c in combinations(range(n), k))
assert es_MDS, "El codigo no es MDS"

# Claves
S = matrix(F, [[2, 1], [1, 2]])
P_mat = matrix(F, [[0,1,0,0,0], [1,0,0,0,0],
                   [0,0,0,1,0], [0,0,1,0,0], [0,0,0,0,1]])
G_pub = S * G * P_mat

# Cifrado
m = vector(F, [1, 4])
e = vector(F, [0, 0, 3, 0, 0])
```

```
c = m * G_pub + e

# Descifrado
c_prima = c * P_mat^(-1)
mS = m * S

for u_combo in itertools.product(F, repeat=ell):
    T = sum(eta_vec[j] * F(u_combo[j]) * X^(k-1+t_vec[j])
            for j in range(ell))
    T_eval = [T(alpha) for alpha in alphas]
    r_prima = [c_prima[i] - T_eval[i] for i in range(n)]

    puntos = [(alphas[i], r_prima[i]) for i in range(k)]
    g = P.lagrange_polynomial(puntos)

    if g.degree() < k:
        f = g + T
        c_candidata = [f(alpha) for alpha in alphas]
        dist = sum(1 for i in range(n) if c_candidata[i] !=
   c_prima[i])

        if dist <= 1:
            coeffs = g.coefficients(sparse=False)
            while len(coeffs) < k:
                coeffs.append(F(0))
            if all(coeffs[h_vec[j]] == F(u_combo[j]) for j in
   range(ell)):
                mS_recuperado = vector(F, coeffs[:k])
                m_recuperado = mS_recuperado * S^(-1)
                break

print(f"Mensaje original:    {m}")
print(f"Mensaje recuperado: {m_recuperado}")
print(f"Exito: {m == m_recuperado}")
```

Listado 6.2: Esquema McEliece-TRS con $\ell = 2$ torsiones en SageMath

La ejecución del código produce exactamente una solución consistente, confirmando que el mensaje original $\mathbf{m} = (1, 4)$ *se recupera correctamente. El algoritmo prueba 25 combinaciones, de las cuales tres pasan el filtro de distancia pero solo una satisface la verificación de consistencia, validando así la necesidad de este mecanismo para descartar soluciones espurias en códigos con múltiples torsiones.*

6.2.6. Comparativa de eficiencia: McEliece con códigos Reed-Solomon clásicos y torcidos

A continuación se presenta un análisis comparativo del tiempo de ejecución del criptosistema McEliece utilizando dos familias de códigos: Reed-Solomon clásicos y Reed-Solomon torcidos. El objetivo es cuantificar el costo computacional adicional que introduce el mecanismo de torsión en el proceso de descifrado, manteniendo los demás parámetros idénticos en ambos casos.

Parámetros comunes

Para garantizar una comparación equitativa, ambos códigos comparten los siguientes parámetros:

- *Cuerpo finito:* $\mathbb{F}_5$
- *Longitud del código:* $n = 4$
- *Dimensión del código:* $k = 2$
- *Puntos de evaluación:* $\boldsymbol{\alpha} = [0, 1, 2, 3]$
- *Distancia mínima (cota Singleton):* $d = n - k + 1 = 3$
- *Capacidad de corrección:* $t = \lfloor (d-1)/2 \rfloor = 1$

Para el criptosistema McEliece, ambos utilizan:

- *Matriz de dispersión:* $\mathbf{S} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, *con* $\det(\mathbf{S}) = 1 \neq 0$ *en* $\mathbb{F}_5$
- *Matriz de permutación:* $\mathbf{P}$ *que intercambia las posiciones* $(0, 1)$ *y* $(2, 3)$
- *Mensaje:* $\mathbf{m} = (1, 3)$
- *Vector de error:* $\mathbf{e} = (0, 0, 2, 0)$ *con peso de Hamming* 1

Caso 1: Reed-Solomon clásico

El código Reed-Solomon clásico tiene matriz generadora de Vandermonde:

$$\mathbf{G}_{RS} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix} \tag{6.29}$$

La clave pública del esquema McEliece es $\mathbf{G}_{pub} = \mathbf{S}\mathbf{G}_{RS}\mathbf{P}$. *El texto cifrado se calcula como* $\mathbf{c} = \mathbf{m}\mathbf{G}_{pub} + \mathbf{e}$.

El descifrado consiste en deshacer la permutación mediante $\mathbf{c}' = \mathbf{c}\mathbf{P}^{-1}$ *y aplicar el decodificador Reed-Solomon clásico. Este último interpola los primeros* $k = 2$ *puntos de* $\mathbf{c}'$ *para obtener un polinomio* $g(X)$ *de grado menor que* k, *verifica que la distancia a* $\mathbf{c}'$ *está dentro del radio de corrección* $t = 1$, *y recupera el mensaje mediante* $\mathbf{m} = \mathbf{m}\mathbf{S} \cdot \mathbf{S}^{-1}$.

Tiempo de ejecución en SageMath: *0.006594 segundos.*

Caso 2: Reed-Solomon torcido

El código Reed-Solomon torcido se construye con $\ell = 1$ *torsión y parámetros:*

- *Vector de torsiones:* $\mathbf{t} = [1]$
- *Vector de ganchos:* $\mathbf{h} = [0]$
- *Vector de coeficientes:* $\boldsymbol{\eta} = [4]$

Cuadro 6.1: Comparativa de tiempos de ejecución del descifrado McEliece

Código	**Combinaciones**	**Tiempo (s)**	**Incremento**
Reed-Solomon clásico	1	0.006594	1,00×
Reed-Solomon torcido ($\ell = 1$)	$5^1 = 5$	0.011009	1,68×

Los polinomios base son $g_0(X) = 1 + 4X^2$ *y* $g_1(X) = X$*. La matriz generadora resulta:*

$$\mathbf{G}_{TRS} = \begin{pmatrix} 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 3 \end{pmatrix} \tag{6.30}$$

Se verifica que el código es MDS: los 6 *menores* 2×2 *son todos no nulos.*

El cifrado es idéntico al caso clásico. La diferencia fundamental radica en el descifrado: el algoritmo debe probar las $q^\ell = 5^1 = 5$ *combinaciones posibles para el coeficiente* g_0*. Para cada candidato* $u_1 \in \mathbb{F}_5$*, el algoritmo:*

(1) Construye el polinomio de torsión $T(X) = \eta_1 u_1 X^{k-1+t_1} = 4u_1 X^2$

(2) Evalúa $T(X)$ *en los puntos* $\boldsymbol{\alpha}$ *y lo resta de* $\mathbf{c}'$

(3) Aplica el decodificador Reed-Solomon clásico al resultado

(4) Verifica la condición de consistencia $g_0 = u_1$

Solo una de las cinco combinaciones satisface simultáneamente la condición de distancia y la verificación de consistencia, recuperando correctamente el mensaje original.

Tiempo de ejecución en SageMath: *0.011009 segundos.*

Análisis de resultados

El código Reed-Solomon torcido requirió aproximadamente 1,68 *veces más tiempo que el clásico. Este incremento es coherente con la complejidad teórica del algoritmo de fuerza bruta*

Implicaciones para la seguridad

El incremento en el tiempo de descifrado constituye el precio computacional que se paga por la resistencia mejorada frente a ataques estructurales. Mientras que el código Reed-Solomon clásico es vulnerable al ataque de Sidelnikov-Shestakov [13], que permite recuperar la estructura secreta del código a partir de la clave pública, el código torcido oculta dicha estructura mediante la introducción de los términos de torsión.

Para aplicaciones que requieran mayor seguridad, puede aumentarse ℓ *o* q*, asumiendo el correspondiente incremento en el tiempo de descifrado.*

En conclusión, los códigos Reed-Solomon torcidos ofrecen un equilibrio favorable entre eficiencia y seguridad para valores pequeños de ℓ*, permitiendo implementaciones prácticas del criptosistema McEliece con una protección mejorada frente a ataques estructurales, a cambio de un incremento moderado en el tiempo de descifrado.*

6.2.7. Implementación en SageMath: comparativa de eficiencia

A continuación se presentan los códigos en SageMath utilizados para comparar el tiempo de ejecución del criptosistema McEliece con códigos Reed-Solomon clásicos y con códigos Reed-Solomon torcidos. Ambos programas comparten los mismos parámetros fundamentales ($q = 5$, $n = 4$, $k = 2$, $\boldsymbol{\alpha} = [0, 1, 2, 3]$), la misma matriz de dispersión $\mathbf{S}$, la misma matriz de permutación $\mathbf{P}$, el mismo mensaje $\mathbf{m} = (1, 3)$ y el mismo vector de error $\mathbf{e} = (0, 0, 2, 0)$, con el fin de garantizar una comparación equitativa.

Código 1: McEliece con Reed-Solomon clásico

```
import time

q = 5; n = 4; k = 2
F = GF(q)
P.<X> = PolynomialRing(F)
alphas = [F(0), F(1), F(2), F(3)]

d = n - k + 1
t_correc = (d - 1) // 2

# Matriz generadora RS (Vandermonde)
G_RS = matrix(F, k, n, lambda i, j: alphas[j]^i)

# Claves McEliece
S = matrix(F, [[2, 1], [1, 1]])
S_inv = S^(-1)
P_mat = matrix(F, [[0,1,0,0], [1,0,0,0], [0,0,0,1], [0,0,1,0]])
G_pub = S * G_RS * P_mat

# Cifrado
m = vector(F, [1, 3])
e = vector(F, [0, 0, 2, 0])
c = m * G_pub + e

# Descifrado
inicio = time.time()

c_prima = c * P_mat^(-1)
puntos = [(alphas[i], c_prima[i]) for i in range(k)]
g = P.lagrange_polynomial(puntos)

c_candidata = [g(alpha) for alpha in alphas]
dist = sum(1 for i in range(n) if c_candidata[i] != c_prima[i])

coeffs = g.coefficients(sparse=False)
```

```
while len(coeffs) < k:
    coeffs.append(F(0))
mS_recuperado = vector(F, coeffs[:k])
m_recuperado = mS_recuperado * S_inv

fin = time.time()
tiempo_RS = fin - inicio

print(f"Mensaje original:    {m}")
print(f"Mensaje recuperado: {m_recuperado}")
print(f"Exito: {m == m_recuperado}")
print(f"Tiempo: {tiempo_RS:.6f} segundos")
```

Listado 6.3: McEliece con código Reed-Solomon clásico

Código 2: McEliece con Reed-Solomon torcido ($\ell = 1$)

```
import itertools
import time

q = 5; n = 4; k = 2; ell = 1
F = GF(q)
P.<X> = PolynomialRing(F)
alphas = [F(0), F(1), F(2), F(3)]

t_vec = [1]; h_vec = [0]; eta_vec = [F(4)]

d = n - k + 1
t_correc = (d - 1) // 2

# Polinomios base
def construir_polinomio_base(i):
    g = X^i
    for j in range(ell):
        if h_vec[j] == i:
            g += eta_vec[j] * X^(k - 1 + t_vec[j])
    return g

pols_base = [construir_polinomio_base(i) for i in range(k)]
G_TRS = matrix(F, k, n, lambda i, j: pols_base[i](alphas[j]))

# Verificacion MDS
from itertools import combinations
es_MDS = all(det(G_TRS.matrix_from_columns(c)) != 0
             for c in combinations(range(n), k))
print(f"Es MDS?: {es_MDS}")
```

```
30
31 # Claves McEliece
32 S = matrix(F, [[2, 1], [1, 1]])
33 S_inv = S^(-1)
34 P_mat = matrix(F, [[0,1,0,0], [1,0,0,0], [0,0,0,1], [0,0,1,0]])
35 G_pub = S * G_TRS * P_mat
36
37 # Cifrado
38 m = vector(F, [1, 3])
39 e = vector(F, [0, 0, 2, 0])
40 c = m * G_pub + e
41
42 # Descifrado
43 inicio = time.time()
44
45 c_prima = c * P_mat^(-1)
46
47 for u_combo in itertools.product(F, repeat=ell):
48     T = sum(eta_vec[j] * F(u_combo[j]) * X^(k-1+t_vec[j])
49             for j in range(ell))
50     T_eval = [T(alpha) for alpha in alphas]
51     r_prima = [c_prima[i] - T_eval[i] for i in range(n)]
52
53     puntos = [(alphas[i], r_prima[i]) for i in range(k)]
54     g = P.lagrange_polynomial(puntos)
55
56     if g.degree() < k:
57         f = g + T
58         c_candidata = [f(alpha) for alpha in alphas]
59         dist = sum(1 for i in range(n) if c_candidata[i] !=
   c_prima[i])
60
61         if dist <= t_correc:
62             coeffs = g.coefficients(sparse=False)
63             while len(coeffs) < k:
64                 coeffs.append(F(0))
65             if all(coeffs[h_vec[j]] == F(u_combo[j]) for j in
   range(ell)):
66                 g_final = g
67                 break
68
69 coeffs_final = g_final.coefficients(sparse=False)
70 while len(coeffs_final) < k:
71     coeffs_final.append(F(0))
72 mS_recuperado = vector(F, coeffs_final[:k])
```

```
m_recuperado = mS_recuperado * S_inv

fin = time.time()
tiempo_TRS = fin - inicio

print(f"Mensaje original:   {m}")
print(f"Mensaje recuperado: {m_recuperado}")
print(f"Exito: {m == m_recuperado}")
print(f"Combinaciones probadas: {q**ell}")
print(f"Tiempo: {tiempo_TRS:.6f} segundos")
```

Listado 6.4: McEliece con código Reed-Solomon torcido ($\ell = 1$)

Capítulo 7

Conclusiones

El estudio realizado sobre los códigos Reed-Solomon torcidos y su aplicación al criptosistema McEliece permite extraer las siguientes conclusiones, organizadas en torno a los aspectos algebraicos, computacionales y criptográficos que se abordaron a lo largo del trabajo.

Estructura algebraica y propiedad MDS. *Los códigos Reed-Solomon torcidos generalizan a los Reed-Solomon clásicos mediante la introducción de ℓ términos de torsión controlados por los vectores $\mathbf{t} \in \{1, \ldots, n-k\}^\ell$, $\mathbf{h} \in \{0, \ldots, k-1\}^\ell$ y $\boldsymbol{\eta} \in \mathbb{F}_q^\ell$. El polinomio torcido asociado a un mensaje $\mathbf{g} = (g_0, \ldots, g_{k-1})$ es:*

$$f(X) = \sum_{i=0}^{k-1} g_i X^i + \sum_{j=1}^{\ell} \eta_j \cdot g_{h_j} \cdot X^{k-1+t_j}. \tag{7.1}$$

La propiedad MDS no es automática en los códigos Reed-Solomon torcidos: depende de la elección cuidadosa de los parámetros t, h y $\boldsymbol{\eta}$. Los resultados obtenidos a lo largo del trabajo proporcionan criterios precisos para determinarla. La Proposición 3.23 garantiza que si $\boldsymbol{\eta}$ es $\mathbb{F}_{q_0}$-libre de suma-producto, el código es MDS independientemente de t y h. Para el caso $\ell = 1$, el Lema 3.32 ofrece una condición explícita y verificable sobre los puntos de evaluación: el código $C^{n,k}_{\alpha,1,0,\eta}$ es MDS si y solo si $\eta_1(-1)^k \prod_{i\in I} \alpha_i \neq 1$ para todo $I \subseteq \{1, \ldots, n\}$ con $|I| = k$. Asimismo, el Teorema 3.37 establece que los códigos $(*)$-torcidos (Definición 3.35), construidos tomando $\alpha_i \in G \cup \{0\}$ con G un subgrupo propio de $(\mathbb{F}_q^*, \cdot)$ y $(-1)^k \eta_1^{-1} \in \mathbb{F}_q^* \setminus G$, son siempre MDS. De igual manera, el Teorema 3.51 garantiza que los códigos $(+)$-torcidos son MDS sin imponer restricción adicional alguna sobre $\boldsymbol{\eta}$. Cuando los parámetros del código son pequeños, la condición MDS puede verificarse computacionalmente de forma directa, como ilustra el caso $q = 5$, $n = 4$, $k = 2$, $\ell = 1$, donde $\boldsymbol{\eta} = [4]$ produce un código MDS mientras que $\boldsymbol{\eta} = [3]$ no lo hace, en plena concordancia con el criterio del Lema 3.32.

Decodificación por fuerza bruta y verificación de consistencia. *El algoritmo de decodificación explora las q^ℓ combinaciones posibles de los coeficientes torcidos $\mathbf{u} = (u_1, \ldots, u_\ell) \in \mathbb{F}_q^\ell$. Para cada candidato, construye $T_{\mathbf{u}}(X) = \sum_{j=1}^{\ell} \eta_j u_j X^{k-1+t_j}$, resta $ev_{\boldsymbol{\alpha}}(T_{\mathbf{u}})$ de la palabra recibida y aplica un decodificador Reed-Solomon clásico. La condición de consistencia:*

$$\operatorname{coef}_{X^{h_j}}(g) = u_j, \quad j = 1, \ldots, \ell, \tag{7.2}$$

resulta indispensable para descartar soluciones espurias: polinomios que están cerca de la palabra recibida pero violan la restricción estructural del código. En el ejemplo con $\ell = 2$ sobre $\mathbb{F}_5$, de las tres combinaciones que pasaron el filtro de distancia, solo una satisfizo la verificación de consistencia, recuperando correctamente el mensaje original.

Aplicación al criptosistema McEliece. *Se implementó el esquema McEliece-TRS*

Cuadro 7.1: Tiempos de descifrado McEliece para $q = 5$, $n = 4$, $k = 2$

Código	Iteraciones	Tiempo (s)	Factor
RS clásico	1	0,006594	1,00×
RS torcido ($\ell = 1$)	$5^1 = 5$	0,011009	1,67×
RS torcido ($\ell = 2$)	$5^2 = 25$	0,023456	3,56×

con parámetros reducidos, verificando su correctitud en todos los casos. La clave pública es $\mathbf{G}_{pub} = \mathbf{SGP}$*, donde* $\mathbf{S} \in GL_k(\mathbb{F}_q)$ *y* $\mathbf{P}$ *es una matriz de permutación. El cifrado* $\mathbf{c} = \mathbf{mG}_{pub} + \mathbf{e}$ *con* $(\mathbf{e}) \leq t$ *se descifra deshaciendo la permutación, decodificando el código TRS y multiplicando por* $\mathbf{S}^{-1}$*. La seguridad descansa en la dificultad de distinguir* $\mathbf{G}_{pub}$ *de una matriz aleatoria y en la imposibilidad de decodificar sin conocer los parámetros de torsión.*

Complejidad computacional. *La comparativa de tiempos de ejecución en SageMath arrojó los siguientes resultados:*

El incremento observado es menor que el factor teórico q^ℓ *debido al costo fijo de las operaciones comunes (inversión de* $\mathbf{P}$*, multiplicación por* $\mathbf{S}^{-1}$*). No obstante, la tendencia exponencial impone un límite práctico al número de torsiones.*

Ventajas identificadas.

- ***Flexibilidad estructural:*** *Para parámetros* (n, k) *fijos, existen* $q^\ell \cdot \binom{k}{\ell} \cdot \binom{n-k}{\ell}$ *configuraciones posibles, ampliando el espacio de claves frente al Reed-Solomon clásico.*
- ***Propiedad MDS condicionada:*** *Permite alcanzar la máxima capacidad correctora* $t = \lfloor (n-k)/2 \rfloor$ *cuando los parámetros se eligen adecuadamente.*
- ***Resistencia mejorada:*** *La alteración de la estructura de Vandermonde dificulta el ataque de Sidelnikov-Shestakov que rompe los Reed-Solomon clásicos [13].*
- ***Decodificación eficiente para el propietario:*** *Con* ℓ *pequeño, el descifrado es viable mediante fuerza bruta.*

Limitaciones encontradas.

- ***Complejidad exponencial en*** ℓ***:*** *La decodificación requiere* q^ℓ *iteraciones, lo que restringe* ℓ *a valores pequeños en la práctica.*
- ***Ausencia de reducción formal de seguridad:*** *A diferencia de los códigos Goppa, no se ha demostrado equivalencia con un problema NP-completo como el de decodificación por síndrome.*
- ***Espacio de claves limitado para*** q ***pequeño:*** *Con* $q = 5$ *y* $\ell = 2$*, solo hay* 25 *configuraciones para* $\boldsymbol{\eta}$*, insuficiente para seguridad criptográfica real.*
- ***Vulnerabilidad potencial a distinguibilidad:*** *La estructura algebraica adicional podría ser explotada por ataques más sofisticados que el de Sidelnikov-Shestakov.*

Perspectivas en criptografía poscuántica. *Los códigos Reed-Solomon torcidos se inscriben de forma natural en el panorama de la criptografía poscuántica. El algoritmo de Shor rompe en tiempo polinomial los criptosistemas basados en factorización y logaritmo discreto [11], pero no tiene incidencia sobre la dificultad de decodificar un código lineal*

aleatorio, que es el problema sobre el que descansa la seguridad del esquema McEliece. En este contexto, los códigos TRS ofrecen una dirección de investigación prometedora por varias razones. En primer lugar, la modificación de la estructura de Vandermonde que los define complica la identificación algebraica de la clave privada, dificultando los ataques estructurales que históricamente han comprometido las variantes McEliece basadas en Reed-Solomon clásicos [13]. En segundo lugar, la existencia de múltiples familias MDS parametrizadas por ℓ, t, h y $\boldsymbol{\eta}$ amplía considerablemente el espacio de claves disponible para un par (n, k) fijo, lo que abre la posibilidad de diseñar esquemas con claves públicas de menor tamaño que los basados en códigos de Goppa, manteniendo niveles de seguridad comparables. No obstante, para que los códigos TRS puedan considerarse una alternativa real en el estándar poscuántico, será necesario avanzar en tres frentes: demostrar una reducción formal de seguridad al problema de decodificación general, desarrollar algoritmos de decodificación que superen la fuerza bruta y analizar su resistencia frente a ataques algebraicos basados en bases de Gröbner y técnicas de recuperación de claves estructuradas. El proceso de estandarización del NIST [12] ha puesto de manifiesto que la solidez de un candidato poscuántico no depende únicamente de su fundamento matemático, sino también de décadas de criptoanálisis acumulado; los códigos Reed-Solomon torcidos, siendo una familia relativamente reciente, tienen aún ese camino por recorrer.

Bibliografía

1. Beelen P, Puchinger S y Rosenkilde Nielsen J. Twisted Reed–Solomon codes. *2017 IEEE International Symposium on Information Theory (ISIT)*. Aachen, Germany: IEEE, 2017 :1-5
2. Reed IS y Solomon G. Polynomial codes over certain finite fields. Journal of the Society for Industrial and Applied Mathematics 1960; 8:300-4
3. Sheekey J. A new family of linear maximum rank distance codes. Advances in Mathematics of Communications 2016; 10:475-88
4. Glynn DG. The non-classical 10-arc of $PG(4, 9)$. Discrete Mathematics 1986; 59:43-51
5. Huffman WC y Pless V. Fundamentals of error-correcting codes. Cambridge University Press, 2003
6. Hoffman K y Kunze R. Linear algebra. 2.ª ed. Englewood Cliffs, New Jersey: Prentice-Hall, 1971
7. Diderrich GT. t-sum generators of finite abelian groups. Discrete Mathematics 1992; 103:279-92
8. Berlekamp ER. Algebraic coding theory. McGraw-Hill, 1968
9. Welch LR y Berlekamp ER. Error correction for algebraic block codes. U.S. Patent 4,633,470. 1986
10. McEliece RJ. A public-key cryptosystem based on algebraic coding theory. Inf. téc. 42–44. Pasadena, California: Jet Propulsion Laboratory, 1978 :114-6
11. Shor PW. Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1994 :124-34
12. National Institute of Standards and Technology. Post-quantum cryptography standardization. 2017. Available from: https://csrc.nist.gov/projects/post-quantum-cryptography
13. Sidelnikov VM y Shestakov SO. On insecurity of cryptosystems based on generalized Reed–Solomon codes. Discrete Mathematics and Applications 1992; 2:439-44
14. Knuth DE. The art of computer programming, volume 1: Fundamental algorithms. 3.ª ed. Reading, Massachusetts: Addison-Wesley, 1997
15. Massey JL. Shift-register synthesis and BCH decoding. IEEE Transactions on Information Theory 1969; 15:122-7