Universidad de Valladolid

Facultad de Ciencias

Esteganografía

Mediante códigos correctores

Trabajo de fin de máster

Máster en Matemática

Autor:
Luisantos Bonilla Mejía

Tutor:
Carlos Munuera Gómez
Departamento de Matemática Aplicada
Universidad de Valladolid

Índice general

Introducción **3**

1 Esteganografía general **4**
1.1 Conceptos generales de la esteganografía en general 4
1.2 Esteganografía digital 5
1.2.1 Reglas de selección para la esteganografía 8
1.2.2 Parámetros para la esteganografía 12
1.3 Esteganografía LSB 14

2 Códigos lineales **16**
2.1 Códigos correctores 16
2.1.1 Códigos lineales sobre un cuerpo 20
2.1.2 Decodificación para códigos lineales 23
2.2 Códigos de Hamming 27

3 Esteganografía y códigos **30**
3.1 Esteganografía a partir de códigos 30
3.1.1 Esteganografía basado en el código de Hamming binario 32
3.2 Selección no compartida 33
3.2.1 Esquemas de papel mojado 34
3.3 Descodificación: sin distancia mínima 37
3.4 Caso ternario (± 1) 45

Bibliografía **48**

Introducción

La esteganografía es el arte de ocultar un mensaje dentro de un objeto u otro mensaje. La esteganografía no es nada nueva, pues desde la antigüedad se han guardado mensajes en poemas o pinturas, de tal forma que quién lea el poema o vea la pintura a primera vista, no le sea evidente la existencia del mensaje.

Antes de la llegada de la digitalización de la información, la esteganografía versaba en técnicas o recursos físicos limitados al ingenio artístico, y a la forma en que los seres humanos interpretamos formas o patrones. Con la llegada de la digitalización, la esteganografía adquiere nuevas técnicas, y nuevos objetos dónde guardar mensajes, incluso otros objetos (imágenes, audio y video, por ejemplo).

La característica de proteger el mensaje de personas ajenas al emisor y receptor (de la esteganografía), es lo que permite hoy en día una nueva forma de seguridad; a diferencia de la criptografía, que permite conocer el mensaje, pero no su contenido (siempre y cuando no sepamos la clave). La esteganografía impide que alguien fuera de las personas involucradas en la comunicación, conozca la existencia del mensaje. Así, la esteganografía nos puede dar una nueva capa de seguridad, permitiendo junto con la criptografía un mayor grado de seguridad en la comunicación.

Así pues, en el presente trabajo nos dedicaremos a estudiar el caso de ocultar un mensaje en una imagen (vista como mapa de bits). Para ello, el documento se divide en tres capítulos, el primero de esteganografía general, el segundo de códigos lineales y un último capitulo destinado a la esteganografía a partir de los códigos lineales.

Capítulo 1

Esteganografía general

Los procesos que usamos para ocultar un mensaje sobre algún objeto, los podemos ver como funciones que dado un mensaje y un objeto (recipiente para el mensaje), obtenemos un nuevo objeto, que además de contener en esencia el objeto original, ahora tiene nuestro mensaje de manera discreta (sin llamar la atención, salvo de quién o quienes, queremos comunicar nuestro mensaje). De igual forma, el proceso inverso se puede ver como otra función que nos permitirá saber el mensaje oculto, dado un recipiente.

De esta manera, daremos conceptos generales para la esteganografía, para luego pasar a tratar el caso donde los recipientes de nuestros mensajes serán imágenes (cadenas de ceros y unos).

1.1. Conceptos generales de la esteganografía en general

Definición 1.1.1. *Un **mensaje** es todo escrito o idea que deseamos comunicar a otra persona o personas. Al conjunto de mensajes posibles lo denotaremos por $\mathcal{M}$.*

Definición 1.1.2. *Una **cubierta** es cualquier medio en el cual podamos colocar nuestro mensaje. Al conjunto de cubiertas lo denotaremos por $\mathcal{C}$.*

Observación: La cubierta está sujeta al tipo de comunicación y al contexto de la comunicación que se usará.

Ejemplo 1.1.3. *Si la comunicación es escrita, una cubierta puede ser un poema y el mensaje se puede colocar letras por letras, cada cuatro letras del poema.*

Definición 1.1.4. *Una **clave** es la información que permite recuperar el mensaje colocado en la cubierta. Al conjunto de claves los denotaremos por $\mathcal{K}$.*

Observación: Cada cubierta tendrá una clave de acceso al mensaje que esta guarde.

Además, a la función con dominio $\mathcal{C} \times \mathcal{M} \times \mathcal{K}$ y codominio $\mathcal{C}$, que permite incrustar un mensaje en una cubierta específica la denotaremos por **emb** (por el inglés embedding). Y a la función con dominio $\mathcal{C} \times \mathcal{K}$ y codominio $\mathcal{M}$, que permite recuperar el mensaje de una cubierta específica la denotamos por **rec** (por el inglés recovering).

Observación: A la cubierta c original la llamaremos **cubierta lisa**, y a la cubierta modificada $emb(c, m, k)$ la llamaremos **estegocubierta**.

Definición 1.1.5. *Un **Esquema Esteganográfico** es una quintupla $\mathcal{S} = (\mathcal{C}, \mathcal{M}, \mathcal{K}, emb, rec)$, donde:*

1. *$\mathcal{C}$ es el conjunto de posibles cubiertas.*
2. *$\mathcal{M}$ es el conjunto de los posibles mensajes.*
3. *$\mathcal{K}$ es el conjunto de posibles claves.*
4. *$emb : \mathcal{C} \times \mathcal{M} \times \mathcal{K} \to \mathcal{C}$ es una función de incrustación.*
5. *$rec : \mathcal{C} \times \mathcal{K} \to \mathcal{M}$ es una función de recuperación.*

Verificando que $rec(emb(\boldsymbol{c}, \boldsymbol{m}, \boldsymbol{k}), \boldsymbol{k}) = \boldsymbol{m}$, para todo $\boldsymbol{m} \in \mathcal{M}, \boldsymbol{c} \in \mathcal{C}$ y $\boldsymbol{k} \in \mathcal{K}$.

1.2. Esteganografía digital

En este trabajo nos concentraremos en la comunicación digital. Por tal razón, consideramos los mensaje como **secuencias de símbolos** y las cubiertas serán **imágenes digitales**. Además, consideramos nuestro esquema esteganográfico sin especificar las claves para mayor comodidad de trabajo con las definiciones.

Así, nuestras definiciones iniciales las podemos redefinir de la siguiente manera.

Definición 1.2.1. *Sea $\mathcal{A}$ un alfabeto con q elementos.*

I. Un mensaje m de longitud k es una secuencia (o vector) de longitud k, de símbolos de $\mathcal{A}$. En otras palabras $m \in \mathcal{A}^k$, donde $m = (m_1, m_2, \ldots, m_k)$, y $m_i \in \mathcal{A}, i = 1, 2, \ldots, k$.

II. Una cubierta c de longitud n es una secuencia (o vector) de longitud n, de símbolos de $\mathcal{A}$. En otras palabras $c \in \mathcal{A}^n$, donde $c = (c_1, c_2, \ldots, c_n)$, y $c_i \in \mathcal{A}, i = 1, 2, \ldots, n$.

Así, tenemos que $\mathcal{M} = \mathcal{A}^k$ y $\mathcal{C} = \mathcal{A}^n$.

Observación: El alfabeto $\mathcal{A}$ puede ser un anillo ($\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$) o un cuerpo ($\mathbb{F}_q$, donde q es la potencia de un primo).

Definición 1.2.2. *Sea $\mathcal{A}$ un alfabeto con q elementos. Llamaremos* ***Esquema Esteganográfico digital*** *de tipo $[n,k]$ en $\mathcal{A}$, al par $\mathcal{S} = (emb, rec)$, donde:*

1. *$emb : \mathcal{A}^n \times \mathcal{A}^k \to \mathcal{A}^n$ es la función de incrustación.*
2. *$rec : \mathcal{A}^n \to \mathcal{A}^k$ es la función de recuperación.*

Verificando que $rec(emb(c,m)) = m, \forall c \in \mathcal{A}^n$ y $\forall m \in \mathcal{A}^k$.

Una consecuencia inmediata de la Definición 1.2.2 es la siguiente proposición.

Proposición 1.2.3. *Sea $\mathcal{S} = (emb, rec)$ un esquema esteganográfico de tipo $[n,k]$ en $\mathcal{A}$. Entonces*

1. *La función rec es sobreyectiva.*
2. *Para $c \in \mathcal{A}^n$ fijo, la función $emb(c,-) : \mathcal{A}^k \to \mathcal{A}^n$ es inyectiva.*

Demostración.

- Primero veamos que *rec* es sobreyectiva. Así, sea $m \in \mathcal{A}^k$ un mensaje arbitrario. Entonces por la Definición 1.2.2 tenemos que se cumple para m la siguiente igualdad:

$$rec(emb(c,m)) = m, \forall c \in \mathcal{A}^n.$$

  Así, basta tomar un $c_0 \in \mathcal{A}^n$ en particular y a $emb(c_o, m)$ como la preimagen de m por *rec*. Entonces tendremos que $rec(emb(c_o,m)) = m$, lo que significa que *rec* es sobreyectiva.

- Ahora, veamos que *emb* es inyectiva. Sea $c \in \mathcal{A}^n$ fijo. Además, sean $m_1, m_2 \in \mathcal{A}^k$ arbitrarios tales que $emb(c,m_1) = emb(c,m_2)$. Entonces, al aplicar *rec* en la igualdad anterior tendremos por la Definición 1.2.2:

$$\begin{aligned} emb(c,m_1) &= emb(c,m_2) \\ rec(emb(c,m_1)) &= rec(emb(c,m_2)) \\ m_1 &= m_2. \end{aligned}$$

  Lo que significa que *emb* es inyectiva. □

Dado un alfabeto $\mathcal{A}$ finito. Podemos definir una distancia en $\mathcal{A}^n$, con $n \in \mathbb{Z}^+$, la cual cuenta el número de símbolos diferentes entre dos elementos de $\mathcal{A}^n$. Dicha distancia recibe el nombre de **distancia de Hamming** y la definimos a continuación.

Definición 1.2.4. *Sea $\mathcal{A}$ un alfabeto de q elementos. La distancia d en $\mathcal{A}^n$ con $n \in \mathbb{Z}^+$ se define como:*

$$\forall x, y \in \mathcal{A}^n, d(x,y) = \#\{i : x_i \neq y_i, \text{ donde } x = (x_1, x_2, \ldots, x_n) \wedge y = (y_1, y_2, \ldots, y_n)\}.$$

Y recibe el nombre de ***distancia de Hamming****.*

La distancia de Hamming cumple las siguientes propiedades[1].

Proposición 1.2.5. *La distancia de Hamming cumple las siguiente propiedades:*

1. $\forall x, y \in \mathcal{A}^n, d(x,y) \geq 0$.
2. *Sean $x, y \in \mathcal{A}^n$. Entonces $d(x,y) = 0 \Leftrightarrow x = y$.*
3. $\forall x, y \in \mathcal{A}^n, d(x,y) = d(y,x)$.
4. $\forall x, y, z \in \mathcal{A}^n, d(x,y) \leq d(x,z) + d(z,y)$.

Demostración.

- Para 1 se deduce de la Definición 1.2.4, ya que d se define como la cantidad de símbolos diferentes en dos elementos de $\mathcal{A}^n$.
- Para 2. Sean $x, y \in \mathcal{A}^n$ tales que $d(x,y) = 0$, entonces por la Definición 1.2.4, tenemos que $\forall i = 1, 2, \ldots, n, x_i = y_i \Rightarrow x = y$, donde $x = (x_1, x_2, \ldots, x_n)$ y $y = (y_1, y_2, \ldots, y_n)$.

  Por otro lado, si $x = y$, donde $x, y \in \mathcal{A}^n$, entonces $\forall i = 1, 2, \ldots, n$ tenemos que $x_i = y_i \Rightarrow \#\{i : x_i \neq y_i$, donde $x = (x_1, x_2, \ldots, x_n) \wedge y = (y_1, y_2, \ldots, y_n)\} = 0 \Rightarrow d(x,y) = 0$.
- Para 3. Es inmediato de la Definición 1.2.4. Sean $x, y \in \mathcal{A}^n$ arbitrarios, entonces:

$$\begin{aligned} d(x,y) &= \#\{i : x_i \neq y_i, \text{ donde } x = (x_1, x_2, \ldots, x_n) \wedge y = (y_1, y_2, \ldots, y_n)\} \\ &= \#\{i : y_i \neq x_i, \text{ donde } y = (y_1, y_2, \ldots, y_n) \wedge x = (x_1, x_2, \ldots, x_n)\} = d(y,x). \end{aligned}$$

- Para 4. Si $n = 1$, tendremos que d solo puede devolver 0 o 1, así dados $x, y \in \mathcal{A}$ tendremos que $d(x,y) \leq 1$. Además, si tomamos $z \in \mathcal{A}$ puede cumplir dos condiciones o $x = z$ o $x \neq z$.

  Si $x = z$, entonces $d(x,z) = 0 \Rightarrow d(x,y) = 0 + d(x,y) = d(x,z) + d(z,y)$. Por otro lado si $x \neq z$, entonces $d(x,z) = 1$, además $d(z,y) \geq 0$ por 1. Así tendremos:

$$\begin{aligned} d(x,y) &\leq 1 \\ &\leq 1 + d(z,y) \\ &= d(x,z) + d(z,y). \end{aligned}$$

[1] Por este motivo recibe el nombre de «distancia».

Por lo tanto, tenemos que $\forall x, y, z \in \mathcal{A}, d(x,y) \leq d(x,z) + d(z,y)$. Finalmente, utilizando lo anterior, para demostrar la desigualdad triangular. Si $n \in \mathbb{Z}^+$, y $x, y, z \in \mathcal{A}^n$, tendremos:

$$\begin{aligned} d(x,y) &= \#\{i : x_i \neq y_i, \text{ donde } x = (x_1, x_2, \ldots, x_n) \wedge y = (y_1, y_2, \ldots, y_n)\} \\ &= \sum_{i=1}^{n} d(x_i, y_i) \\ &\leq \sum_{i=1}^{n} [d(x_i, z_i) + d(z_i, y_i)] \\ &= \sum_{i=1}^{n} d(x_i, z_i) + \sum_{i=1}^{n} d(z_i, y_i) = d(x,z) + d(z,y). \end{aligned}$$

Así, hemos probado que: $\forall x, y, z \in \mathcal{A}^n, d(x,y) \leq d(x,z) + d(z,y)$. □

Ya definida una distancia en $\mathcal{A}^n$, podemos definir la distancia de un elemento de $\mathcal{A}^n$ a un subconjunto de $\mathcal{A}^n$.

Definición 1.2.6. *Sea $x \in \mathcal{A}^n$ con $n \in \mathbb{Z}^+$, y un subconjunto $\mathcal{Y}$ de $\mathcal{A}^n$. La distancia de x a $\mathcal{Y}$ se denota por $d(x, \mathcal{Y})$ y se define como:*

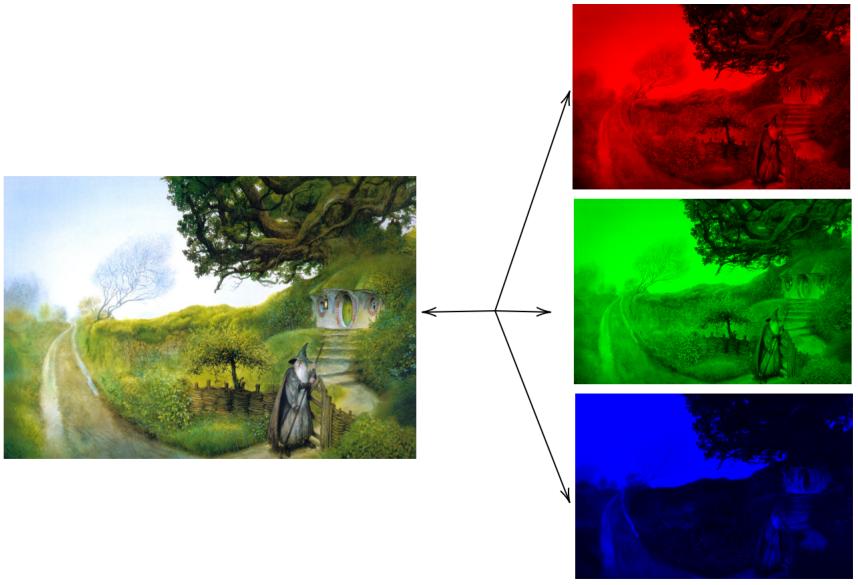
$$d(x, \mathcal{Y}) = \text{mín}\{d(x,y) : y \in \mathcal{Y}\}.$$

Como la distancia de Hamming cuenta el número de símbolos diferentes entre dos cadena de símbolos, por ejemplo $x, y \in \mathcal{A}^n$. Entonces $d(x,y)$ se puede interpretar como el número de cambios que se realizaron a x para transformarlo en y, manteniendo x fijo (de igual forma se puede analizar manteniendo y fijo). En cualquiera de los casos, esto nos permite interpretar la distancia de Hamming como el número de cambios que hacemos para agregar un mensaje m en la cubierta c.

Además, la distancia de un cadena de símbolos a un subconjunto, se interpreta como el mínimo número de cambios necesarios para agregar un mensaje a una cubierta.

1.2.1. Reglas de selección para la esteganografía

Debemos recordar que una imagen cualquiera la podemos descomponer en los tres colores básicos. Teniendo así tres copias de la misma imagen, pero con solo un tono de color, estas copias a su vez se identifican con sus escalas en grises. El proceso contrario de juntar las tres copias, nos devuelve la imagen inicial. A continuación, mostramos el proceso de descomposición en los tres colores básicos de una imagen.



De esta forma, basta que nos concentremos en el caso de tener una imagen en la escala de grises para tratar de ocultar nuestro mensaje en la imagen. Para el caso ilustrado anteriormente, nos quedaríamos con una imagen similar a la siguiente:

Cada píxel[2] puede ser representado por un número entero, con esto en mente tenemos la siguiente definición.

[2]Un píxel es la superficie (cuadrado o punto) homogénea más pequeña de las que componen una imagen, que se define por su brillo y color.

Definición 1.2.7. *Dado el conjunto de enteros consecutivos* $\{0, 1, 2, \ldots, M-1\}$, *donde* $M = 2^D$. *Llamaremos a* D *la* ***profundidad*** *de los bits usados para representar los píxeles de la imagen, y los enteros del conjunto* $\{0, 1, 2, \ldots, M-1\}$ *son la representación de los diferentes tonos de la escala de grises.*

Observación: D define el tamaño de las cadenas de ceros y unos que representan cada entero de la lista $\{0, 1, 2, \ldots, M-1\}$. Además, con $D = 8$ tenemos la escala de grises usual, y con $D = 24$ una escala para todos los colores.

Ejemplo 1.2.8. *Si* $D = 1$ *tendremos solo los colores blanco y negro, donde* 0 *el color blanco y* 1 *el color negro.*

Ejemplo 1.2.9. *Si* $D = 2$ *tendremos una escala de grises, donde* 0 *representa el color blanco,* 3 *el color negro,* 1 *el color gris claro y* 2 *el color gris oscuro.*

Teniendo en cuenta lo anterior, una forma poco práctica de ocultar el mensaje en la imagen, es considerar el mensaje como ruido y distribuirlo dentro de cada píxel sin ningún orden en particular[3]. Entonces, la idea es poder incrustar nuestro mensaje en ciertos píxeles de la imagen, de tal forma que esto no perturbe la imagen original. La selección de los píxeles debe ser en lugares donde sí haya cambio de colores, por ejemplo bordes de figuras que estén en la imagen. Así, de la imagen podemos obtener una cubierta que permita agregarle en ciertos lugares el mensaje.

Definición 1.2.10. *A la imagen* $\mathcal{I}$ *se le llamará* ***cubierta de trabajo*** *y la cubierta* $\mathbf{c}$ *que obtenemos de* $\mathcal{I}$, *con los píxeles seleccionados y codificados se llamará* ***vector de la cubierta****.*

Observación: La diferencia entre la cubierta de la definición 1.1.2 y el vector de la cubierta, es que el vector de la cubierta nos indica en qué lugares de la cubierta dejaremos el mensaje.

Definición 1.2.11. *Al proceso por el cual obtenemos* $\mathbf{c}$ *de* $\mathcal{I}$ *es llamado* ***regla de selección****.*

Observación: Debemos aclarar que $\mathbf{c}$ aun no tiene el mensaje a enviar, solo está preparado para acoger en los píxeles seleccionados de la imagen el mensaje a ocultar.

En general las reglas de selección tienen dos pasos. Primero, debemos seleccionar los píxeles que serán cambiados por nuestro mensaje[4]. Segundo, tener la cubierta de trabajo en escala de gris. Supongamos que la cubierta de trabajo es $s = (s_1, s_2, \ldots, s_n)$, con $s_i \in \{0, 1, \ldots, M-1\}$. Entonces, en función del primer paso, la cubierta c es obtenida de s

[3]Esto aunque es factible, es poco práctico, porque en algunos casos será muy obvio que se está ocultando algo.

[4]Es recomendable utilizar píxeles donde el cambio no sea tan notorio, como esquinas de alguna figura de la imagen, para que el cambio de figura y color, permita ocultar la no concordancia de los tonos vecinos al píxel cambiado.

Figura 1.1: Los píxeles (marcados con rojo) de los arbustos de la parte inferior derecha de la imagen son buenos lugares para usar. También, los píxeles (marcados con rojo) que definen la puerta.

dependiendo del tipo de esquema esteganográfico seleccionado (podemos incluso tener varios s, si los píxeles seleccionados no son vecinos).

Algo importante de denotar es que en el vector de la cubierta, sus entradas tienen píxeles que no serán modificados y píxeles que serán modificados en el proceso de incrustación del mensaje. Lo anterior dependerá del esquema seleccionado, que cambiará todos o solo algunos. Además, todas las entradas del vector de la cubierta están codificadas al alfabeto del esquema esteganográfico seleccionado.

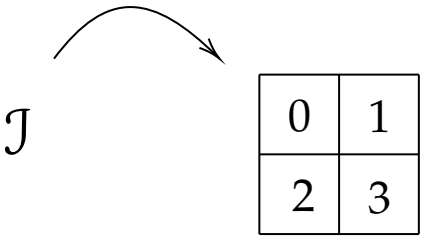


Figura 1.2: Ejemplo de como se vería una imagen $\mathcal{J}$ que solo está compuesta por cuatro tonalidades.

Así, el problema fundamental es saber cuáles píxeles podemos cambiar de tal manera que no se pierda la información original, y que no levante sospecha de que hay algo más aparte de la información. Por lo cual, el remitente debe seleccionar un mecanismo que permita la selección adecuada.

1.2.2. Parámetros para la esteganografía

La calidad de un esquema esteganográfico $\mathcal{S}$ de tipo $[n, k]$ definido sobre un alfabeto $\mathcal{A}$ con q símbolos, puede ser medida en término de ciertos parámetros, los cuales se dividen en **parámetros absolutos** y **parámetros relativos**.

- **Parámetros absolutos:**
  - La **longitud de la cubierta** n.
  - La **capacidad de incrustación** k, que es la longitud del mensaje.
  - El **radio de incrustación** ρ, que es el máximo número de cambios que se pueden hacer. ρ se define como:

$$\rho = \text{máx}\{d(c, emb(c, m)) : c \in \mathcal{A}^n, m \in \mathcal{A}^k\}.$$

  - El **promedio de cambios** R_a. Asumiendo que todos los mensajes y cubiertas tienen igual probabilidad y ya que $|\mathcal{A}| = q$, se define como:

$$R_a = \frac{1}{q^{kn}} \sum_{c \in \mathcal{A}^n, m \in \mathcal{A}^k} d(c, emb(c, m)).$$

Los esquemas esteganográficos con cubiertas de longitud n, capacidad de incrustación k y promedio de cambios R_a, se denotarán como **esquemas esteganográficos** $[n, k, R_a]$.

- **Parámetros relativos:**
  - **Carga útil relativa** $\alpha = k/n$, mide la proporción entre la capacidad de incrustación y la cantidad de símbolos de la cubierta. Lo anterior, nos dice qué tanto podemos usar de la cubierta para incrustar nuestro mensaje. A veces, es conveniente expresar esta medida en bits por símbolo de la cubierta, en este caso recibe el nombre de **carga útil relativa binaria** o **tasa de incrustación**, y se expresa como:

$$E = \frac{k}{n} \log_2(q).$$

    Observación: Usamos logaritmo base 2, ya que queremos la relación de bit por símbolo de nuestro alfabeto, así tenemos $2^E = q^{k/n}$.
  - **Tasa de cambio** $c = R_a/n$, mide la probabilidad que un símbolo en la cubierta sea cambiada durante el proceso de incrustación. La tasa de cambio también es llamada **promedio de distorsión**.

    Observación: Por definición de α y c, tenemos que $0 \leq \alpha, c \leq 1$.

- **Eficiencia de incrustación** e, es la relación entre la carga útil relativa y la tasa de cambio. Entonces, e es la cantidad de información incluida en la cubierta por un solo cambio. Consideramos también la **menor eficiencia de incrustación** $\underline{e}$.

$$e = \frac{k}{R_a}$$
$$\underline{e} = \frac{k}{\rho}.$$

  Si queremos medir en bits, entonces consideramos la **eficiencia de incrustación binaria** y la **eficiencia de incrustación binaria más baja**, que son las relaciones **tasa de incrustación/tasa de cambio** y **tasa de incrustación/radio de incrustación**, respectivamente.
  Observación: Por definición se verifica que $R_a \leq \rho \Rightarrow e \geq \underline{e}$.

En general, el rendimiento de un esquema esteganográfico se mide en términos de las relaciones de **carga útil relativa/tasa de cambio** o **tasa de incorporación/tasa de cambio**, es decir, en términos de eficiencia de incorporación.

La idea de un buen esquema esteganográfico es poder incorporar tanta información en la cubierta con el menor número de cambios posibles. El esquema que cumpla con esta propiedad lo llamaremos **adecuado** y lo definimos a continuación:

Definición 1.2.12. *El esquema $\mathcal{S} = (emb, rec)$ se llama **adecuado**, si el número de cambios producido en la cubierta por el proceso de incrustación es el mínimo posible permitido por la función de recuperación, es decir, si $d(c, emb(c,m)) = d(c, rec^{-1}(m)), \forall c \in \mathcal{A}^n, \forall m \in \mathcal{A}^k$.*

Observación: $rec^{-1}(m)$ representa la imagen inversa de m mediante la función de recuperación rec.

Proposición 1.2.13. *Sea $\mathcal{S} = (emb, rec)$ es un esquema esteganográfico del tipo $[n,k]$ sobre $\mathcal{A}$. Entonces existe un esquema esteganográfico adecuado $\mathcal{S}^* = (emb*, rec)$ del tipo $[n,k]$, tal que:*

$$R_a(\mathcal{S}^*) \leq R_a(\mathcal{S}).$$

Demostración. Dado un mensaje $m \in \mathcal{A}^k$ con $k \in \mathbb{Z}^+$, si existen $c, x \in \mathcal{A}^n$ con $n \in \mathbb{Z}^+$, tal que $rec(x) = m$ y $d(c,x) < d(c, emb(c,m))$ y $rec(x) = m$, entonces podemos definir la función emb^* como:

$$emb^*(c', m') = \begin{cases} x & si \quad (c', m') = (c, m) \\ \\ emb(c', m') & si \quad (c', m') \neq (c, m) \end{cases}$$

Además, si con emb^* (como se definió anteriormente) y el mismo mensaje m, existen nuevos c, x con la propiedad de que $d(c,x) < d(c, emb(c,m))$; entonces podemos redefinir emb^*

como se hizo anteriormente (con el nuevo x). Este proceso se terminará ya que A^n es un conjunto finito.

Al hacer el análisis anterior para cada $m \in \mathcal{A}^k$, tendremos una función incrustación emb^* que cumple que $d(c, emb^*(c,m)) = d(c, rec^{-1}(m)), \forall c \in \mathcal{A}^n, \forall m \in \mathcal{A}^k$ (porque eso hemos ido buscando, tener el menor número de cambios, cada vez que encontrábamos c, x tal que $d(c,x) < d(c, emb(c,m))$). De igual forma ya que A^k es un conjunto finito, este proceso se termina en en algún momento.

Así, por construcción tenemos $d(c, emb^*(c,m)) < d(c, emb(c,m))$, entonces considerando el esquema $\mathcal{S}^* = (emb^*, rec)$, tendremos:

$$\begin{aligned} R_a(\mathcal{S}^*) &= \frac{1}{q^{kn}} \sum_{c \in \mathcal{A}^n, m \in \mathcal{A}^k} d(c, emb^*(c,m)) \\ &< \frac{1}{q^{kn}} \sum_{c \in \mathcal{A}^n, m \in \mathcal{A}^k} d(c, emb(c,m)) = R_a(\mathcal{S}). \end{aligned}$$

Entonces, hemos construido un esquema adecuado $\mathcal{S}^*$, tal que $R_a(\mathcal{S}^*) < R_a(\mathcal{S})$. Es claro que si $\mathcal{S}$ cumple la definición 1.2.12 desde el inicio, entonces $\mathcal{S}^* = \mathcal{S} \Rightarrow R_a(\mathcal{S}^*) = R_a(\mathcal{S})$. □

1.3. Esteganografía LSB

Para finalizar el capítulo veamos un ejemplo sencillo de esteganografía, el cual recibe el nombre de **incrustación de bits menos significativos**, también llamado **esquema LSB**[5].

En esta sección usaremos el esquema LSB, que consiste en tomar el último símbolo de la codificación binaria de los píxeles de la imagen del vector de la cubierta, para guardar nuestro mensaje.

Observación: Hay más esquemas LSB, donde no necesariamente se toman los últimos bits.

Para entender mejor cómo trabaja este esquema, sean $n = 3$ y consideremos $\mathcal{A} = \mathbb{F}_2$. Consideremos $\mathcal{S}$, tal que emb deja fijo los primeros 2 símbolos y en el último agrega de forma ordenada el mensaje m. La función rec toma el último símbolo para recuperar el mensaje.

Así por ejemplo, si tuviéramos la imagen de la figura 1.2, el vector de la cubierta sería $c = (0,1,2,3)$. Al codificar cada píxel se convierte en $c = (000, 001, 010, 011)$, de aquí tendremos (por el esquema seleccionado) que $c = (00-, 00-, 01-, 01-)$. Lo cual nos dice que el vector de la cubierta tiene un espacio libre por cada píxel, para colocar el mensaje. Entonces, si queremos incrustar el mensaje 1010, tendríamos $(001, 000, 011, 010)$. En la siguiente figura, podemos ver el resumen del proceso antes descrito.

[5]Por el inglés «*Least Significant Bit embedding*».

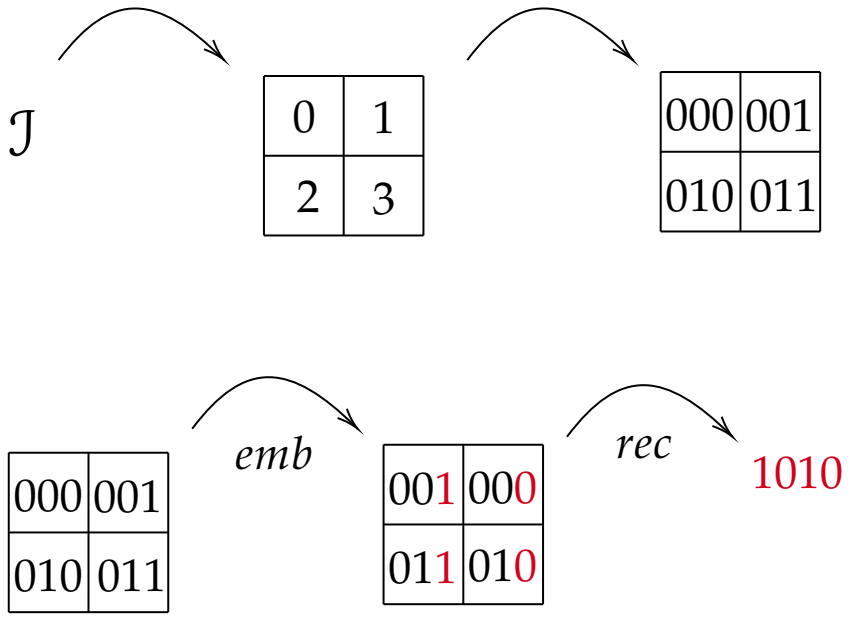


Figura 1.3: Proceso de obtención de una imagen $\mathcal{I}$ un vector de la cubierta, y del proceso de incrustación del mensaje 1010 *en el vector de la cubierta.*

De manera general, consideremos la imagen $\mathcal{I}$, con escala de gris $\mathbf{s} = (s_1, s_2, \ldots, s_n)$. Además, considere $s_i = (b_{i,0}, \ldots, b_{i,l-1})$ la codificación del píxel $i-$ésimo en $\mathbb{F}_2^l$, donde l es la longitud máxima de la cadena de ceros y unos de la expresión binaria de los píxeles[6]. Así, el mapa de bit será $\mathbf{s} = (b_{1,0}, \ldots, b_{1,l}; \ldots; b_{n,0}, \ldots, b_{n,l})$, entonces seleccionamos m píxeles, por ejemplo los primeros[7] m y podemos dejar fijos los primeros t bits de cada píxel, donde $t < l$. Con esto tendremos que el vector de cubierta es $\mathbf{c} = (b_{1,0}, \ldots, b_{1,t-1}, -, \ldots, -; \ldots; b_{m,0}, \ldots, b_{m,t-1}, -, \ldots, -)$, dejando así $l - t$ espacios para guardar nuestro mensaje m.

Este método tiene una longitud de cubierta de nl, y una capacidad de incrustación de $n(l-t)$. Esto nos da una carga útil de $\frac{n(l-t)}{nl} = \frac{l-t}{l}$. Así, la carga útil está en función de la cantidad de bits que dejamos sin usar. Esto quiere decir que si queremos más capacidad para guardar información, perderemos más bits por cada píxel (esto hará que la imagen se deforme). Por lo tanto, este método no es tan eficiente para mensajes largos.

[6]En otras palabras se cumple que $s_i = \sum_{j=0}^{l-1} b_{i,j} 2^j$.

[7]Hemos tomado los primeros m a modo de ilustración, pero en un caso real deben elegirse con cuidado teniendo en cuenta lo discutido en 1.2.1

Capítulo 2

Códigos lineales

Los mensajes son insertados como errores en la cubierta. Por lo tanto, es normal pensar en los códigos correctores, e intentar a partir de sus métodos de codificación y decodificación, crear las funciones *emb* y *rec*. Por esta razón, estudiaremos un poco de códigos correctores, para luego encontrar el vínculo entre la esteganografía y los códigos correctores.

2.1. Códigos correctores

En esta sección nos dedicaremos a mostrar los resultados principales sobre códigos lineales correctores de errores.

Definición 2.1.1. *Un **código corrector de errores de longitud** n **sobre un alfabeto** $\mathcal{A}$ es un subconjunto $\mathcal{C} \subseteq \mathcal{A}^n$. A los elementos de $\mathcal{C}$ se les llama **palabras código** o simplemente **palabras**. Además, si $\mathcal{C}$ tiene M elementos, diremos que es un **código** (n, M).*

La **distancia de un código** es uno de los parámetros más importantes que podemos conocer, ya que nos dirá qué tan bueno es nuestro código para corregir errores.

Definición 2.1.2. *La **distancia del código** $\mathcal{C}$ de $\mathcal{A}^n$ es la distancia mínima entre todas las palabras del código, tomadas de dos en dos; lo que significa:*

$$d = d(\mathcal{C}) = \text{mín}\{d(v, w) : \forall v, w \in \mathcal{C}, v \neq w\}.$$

Donde d es la distancia de Hamming en $\mathcal{A}^n$.

Definición 2.1.3. *El **soporte** de $x \in \mathcal{A}^n$ es el conjunto de coordenadas diferentes de cero de x, donde $x = (x_1, x_2, \ldots, x_n)$, lo que significa:*

$$\text{supp}(x) = \{i : x_i \neq 0\}.$$

Definición 2.1.4. *El **peso** de $x \in \mathcal{A}^n$ es el número de coordenadas diferentes de cero de x, donde $x = (x_1, x_2, \ldots, x_n)$, lo que significa:*

$$\mathrm{wt}(x) = \#\,\mathrm{supp}(x) = d(x, 0), \text{ donde } 0 \in \mathcal{A}^n.$$

wt *también es llamado **peso de Hamming**.*

Observación: Por definición de distancia de Hamming y el peso tenemos la siguiente relación: $d(x, y) = \mathrm{wt}(x - y) = \mathrm{wt}(y - x)$. Además, tanto el peso como la distancia son funciones que devuelven enteros no negativos.

El proceso de decodificación de un código $\mathcal{C}$, se puede interpretar como una función que a cada elemento de $\mathcal{A}^n$ le asigna una de las palabras del código más cercana a él. De manera más formal tenemos:

Definición 2.1.5. *Toda función* $\mathrm{dec}_{\mathcal{C}} : D_{\mathrm{dec}_{\mathcal{C}}} \subseteq \mathcal{A}^n \to \mathcal{C}$, *es llamada **método de decodificación** del código $\mathcal{C}$. Donde el conjunto $D_{\mathrm{dec}_{\mathcal{C}}}$ es el dominio de* $\mathrm{dec}_{\mathcal{C}}$.

Definición 2.1.6. *Sea* $\mathrm{dec}_{\mathcal{C}}$ *un método de decodificación del código $\mathcal{C}$. Si para cada $x \in D_{\mathrm{dec}_{\mathcal{C}}}$ se cumple que $d(x, \mathrm{dec}_{\mathcal{C}}(x)) = d(x, \mathcal{C})$, entonces* $\mathrm{dec}_{\mathcal{C}}$ *es un método de decodificación por **distancia mínima**.*

Observación: $\mathrm{dec}_{\mathcal{C}}(x)$ es una de las palabras del código más cercanas a x.

Definición 2.1.7. *Sea* $\mathrm{dec}_{\mathcal{C}}$ *el método de decodificación del código $\mathcal{C}$, con $D_{\mathrm{dec}_{\mathcal{C}}} = \mathcal{A}^n$. En este caso* $\mathrm{dec}_{\mathcal{C}}$ *es llamado método de decodificación **completo**.*

En la práctica, para la mayoría de códigos, una función dec es difícil de determinar para todos los x de $\mathcal{A}^n$. Por esta razón, basta considerar la función dec definida para un subconjunto x de $\mathcal{A}^n$. Lo anterior se traduce en que un código no puede decodificar todos los patrones de error.

Cuando mandamos un mensaje $c \in \mathcal{C}$ en un canal con ruido, la palabra c puede sufrir algunos cambios en sus coordenadas. Entonces, el receptor del mensaje tendrá un vector x, esto quiere decir que el vector error será $e = x - c$.

Definición 2.1.8. *Sea $c \in \mathcal{C}$ el mensaje enviado, y $x \in \mathcal{A}^n$ el mensaje recibido. Definimos el **patrón de error** o simplemente **error** como el vector $e = x - c \in \mathcal{A}^n$.*

En virtud de las ideas expuestas anteriormente, si se recibe $x \in \mathcal{A}^n$, tal que $x = c + e$, donde $c \in \mathcal{C}$ y $e \in \mathcal{A}^n$. Si $x \notin \mathcal{C}$, significa que han ocurrido errores ($e \neq 0$), y así decodificamos x como $\mathrm{dec}_{\mathcal{C}}(x)$. Si no se han producido demasiados errores, entonces $\mathrm{dec}_{\mathcal{C}}(x) = c$, y si $x \in \mathcal{C}$, entonces, podemos asumir que no han ocurrido errores y entonces $\mathrm{dec}_{\mathcal{C}}(x) = x$.

Así, para poder saber la cantidad máxima de errores que el código puede tolerar, tal que permita que $\mathrm{dec}_{\mathcal{C}}(x) = c$, tenemos el siguiente resultado.

Proposición 2.1.9. *Sea $x = c + e \in \mathcal{A}^n$, con $c \in \mathcal{C}$. Si $2\,\mathrm{wt}(e) < d(\mathcal{C})$, entonces $\mathrm{dec}_{\mathcal{C}}(x) = c$.*

Demostración. Sea $t = \lfloor (d-1)/2 \rfloor$, donde $d = d(\mathcal{C})$. Además, sea $x \in \mathcal{A}^n$, tal que x lo podemos escribir como $x = c + e$, con $c \in \mathcal{C}$. Si $2\,\mathrm{wt}(e) < d(\mathcal{C})$.

$$\begin{aligned} 2\,\mathrm{wt}(e) &< d(\mathcal{C}) = d \\ 2\,\mathrm{wt}(e) &\leq d-1 \\ \mathrm{wt}(e) &\leq \frac{d-1}{2}. \end{aligned}$$

Ahora, por la observación hecha en la Definición 2.1.4, tenemos:

$$d(c,x) = \mathrm{wt}(x-c) = \mathrm{wt}(c+e-c) = \mathrm{wt}(e) \leq \left\lfloor \frac{d-1}{2} \right\rfloor = t.$$

Por lo tanto, tenemos que $d(c,x) \leq t$. De hecho, es la única palabra del código que cumple dicha relación, ya que si existe $c' \in \mathcal{C}$, con $c' \neq c$, tal que $d(c',x) \leq t$, tendremos:

$$d(c,c') \leq d(c,x) + d(x,c') \leq t + t = 2t \leq d-1 < d.$$

Pero lo anterior es una contradicción, ya que d es la distancia mínima del código, y lo más cerca que pueden estar dos palabras del código es d. Por lo tanto, no puede existir otra palabra diferente a c, más cercana a x. Así, por la Definición 2.1.6 $\mathrm{dec}_{\mathcal{C}}(x) = c$. □

De la Proposición 2.1.9, observemos que $t = \lfloor (d-1)/2 \rfloor$ es la capacidad del código $\mathcal{C}$ para corregir errores.

Definición 2.1.10. *Un código $\mathcal{C}$ se dice que es t-**corrector de errores**, si $t = \lfloor (d-1)/2 \rfloor$.*

También debemos notar que la Proposición 2.1.9, no impide que podamos decodificar x por c, teniendo patrones de error con peso mayor, es decir $\mathrm{dec}_{\mathcal{C}}(x) = x$ y $2\,\mathrm{wt}(e) \geq d$.

A veces, las ubicaciones de los errores se conocen. Estos errores se denominan **borrones**. Un canal de comunicaciones en el que los únicos errores posibles son borrones se denomina **canal de borrado**. En estos canales, cuando mandamos un símbolo, el receptor recibirá o el símbolo o un mensaje de no recibido (o sea que se borró el símbolo, durante la transmisión). En estos casos la decodificación consiste simplemente en encontrar los valores que fueron borrados.

Proposición 2.1.11. *Un código de distancia mínima d corrige hasta $d-1$ borrones.*

Demostración. Sean $c, c' \in \mathcal{C}$ arbitrarias, por definición de d tenemos que $d(c,c') \geq d$, esto quiere decir que se diferencian por lo menos en d símbolos. Así, podemos borrar hasta un máximo de $d-1$ símbolos, y todavía habrá al menos un símbolo en que difieran. Luego podemos saber cuál palabra del código es, y corregir los borrones. □

Definición 2.1.12. *Sea $c \in \mathcal{A}^n$, y $t \in \mathbb{Z}_0^+$. La bola de radio t y centro c es el conjunto de palabras de A^n que están a una distancia máxima t de c, es decir:*

$$B(c,t) = \{x \in \mathcal{A}^n : d(c,x) \leq t\}.$$

De la definición anterior, tenemos la siguiente proposición relacionada con la cardinalidad de $B(c,t)$.

Proposición 2.1.13. *Sea $c \in \mathcal{A}^n$, y $t \in \mathbb{Z}_0^+$. Si $\#A = q$, entonces el cardinal de $B(c,t)$ se denota por $V_q(n,t)$ y es:*

$$V_q(n,t) = \#B(c,t) = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

Demostración. Sea $i \in \{0,1,\ldots,t\}$ arbitrario. Así, sea $x \in \mathcal{A}^n$ tal que $d(c,x) = i$, entonces hay i espacios donde puedo colocar un símbolo diferente a los que tiene c. Además, hay $q-1$ símbolos diferentes a los que tiene c en cada coordenada. Entonces tendré $(q-1)^i$ palabras diferentes a c para una elección en concreto de i espacios.

Además, puedo escoger de manera arbitraria los i espacios de n, es decir tengo $\binom{n}{i}$ maneras diferentes de elegir los i espacios. Por lo tanto, tendré $\binom{n}{i}(q-1)^i$ palabras diferentes que están a una distancia i de c. Así, tendremos:

$$\begin{aligned} V_q(n,t) = \#B(c,t) &= \sum_{i=0}^{t} \#\{x \in \mathcal{A}^n : d(c,x) = i\} \\ &= \sum_{i=0}^{t} \binom{n}{i} (q-1)^i. \end{aligned}$$

□

Con esto estamos listos para enunciar una cota para los códigos, en relación con su capacidad correctora.

Teorema 2.1.14 (Cota de Hamming para códigos correctores de errores). *Sea $\mathcal{C}$ un código t-corrector de errores (n,M) sobre un alfabeto $\mathcal{A}$ con q elementos. Entonces*

$$M \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n.$$

Demostración. De las Definiciones 2.1.2 y 2.1.10, tenemos que:

$$\forall v, w \in \mathcal{C}, d(v,w) \geq d > \lfloor (d-1)/2 \rfloor = t.$$

Por lo tanto, todas las palabras están a una distancia mayor a t, y las bolas $B(c,t)$ con $c \in \mathcal{C}$ son disjuntas. Así, tenemos que la cantidad de palabras que las bolas van a contener debe ser menor o igual que el total de palabras de $\mathcal{A}^n$. Por lo tanto:

$$M \sum_{i=0}^{t} \binom{n}{i} (q-1)^i = M \times V_q(n,t) \leq q^n.$$

□

Observación: Los códigos que cumplen la igualdad son llamados **códigos perfectos**. Además, los códigos perfectos verifican que todo $x \in \mathcal{A}^n$ pertenece a una y solo a una bola de radio t y centro una palabra del código.

Teniendo en cuenta la observación anterior, tenemos un parámetro para el código $\mathcal{C}$.

Definición 2.1.15. *Sea $\mathcal{C}$ un código t-corrector de errores (n, M) sobre un alfabeto $\mathcal{A}$ con q elementos. Definimos el **radio de recubrimiento** de $\mathcal{C}$, de la manera siguiente:*

$$\rho = \text{máx}\{d(x, \mathcal{C}) : x \in \mathcal{A}^n\}.$$

Observación: ρ es el radio de las bolas de centro una palabra del código, tal que tengamos el mínimo número de bolas que cubran $\mathcal{A}^n$.

Una consecuencia inmediata de la definición anterior y el Teorema 2.1.14, es la siguiente relación con la capacidad del código para corregir: $\rho \geq t$. Así, $\mathcal{C}$ es perfecto si, y solo si $\rho = t$.

2.1.1. Códigos lineales sobre un cuerpo

Hemos revisando algunos aspectos que nos permiten saber qué tan bueno es un código respecto a otro. En este sentido, podemos hacer uso de algunas estructuras algebraicas, para encontrar buenos códigos.

Si $\mathcal{A} = \mathbb{F}_q$ podemos tomar los códigos como subespacios lineales de $\mathcal{A}^n$ y hacer uso de toda la maquinaria del álgebra lineal para aplicar a los códigos.

Definición 2.1.16. *Un **código lineal** $\mathcal{C}$ de longitud n y dimensión k sobre $\mathbb{F}_q$ es un subespacio de dimensión k de $\mathbb{F}_q^n$. Si el código tiene distancia mínima d, entonces se dirá que es un código $[n, k, d]$.*

Al ser un subespacio vectorial, los códigos los podemos definir apropiadamente a partir de su base.

Definición 2.1.17. *Sea $\mathcal{C}$ un código lineal $[n, k, d]$, y sea $\{g_1, g_2, \ldots, g_k\}$ una base de $\mathcal{C}$. Llamaremos **matriz generadora** a la matriz G que tiene como filas la base de $\mathcal{C}$, es decir:*

$$G = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{pmatrix}.$$

Observación: De las dos definiciones anteriores tenemos:

1. La matriz G es de tamaño $k \times n$.

2. Dada la matriz generadora G de $\mathcal{C}$. Podemos definir $\mathcal{C}$ como $\mathcal{C} = \{xG : x \in \mathbb{F}_q^k\}$. Al proceso de multiplicar x con G es llamado **codificación**, normalmente x es el mensaje a enviar, y se codifica en el código multiplicando por la matriz generadora G.

Existe una forma alternativa para definir el código $\mathcal{C}$, y es mediante una matriz H tal que $H\mathbf{c}^t = \mathbf{0}, \forall \mathbf{c} \in \mathcal{C}$. La matriz H existe, ya que si resolvemos el siguiente sistema de ecuaciones lineales:

$$\begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{pmatrix}_{k\times n} \times \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{k\times 1}$$

Tendremos $n-k$ variables libres, ya que los g_i son linealmente independientes. Así, el rango de la matriz G es k, y tendremos $n-k$ variables libres de los x_i. Entonces, por cada variable libre tenemos una solución $x \in \mathbb{F}_q^n$ (donde todas las variables libres son cero, salvo una), tal que $Gx^t = \mathbf{0}_{k\times n}$.

Por construcción de las soluciones anteriores, tenemos $n-k$ soluciones linealmente independientes. Sean $h_1, h_2, \ldots, h_{n-k} \in \mathbb{F}_q^n$ las soluciones creadas a partir de las $n-k$ variables libres. Entonces, la matriz H es:

$$H = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-k} \end{pmatrix}.$$

Donde $Gh_i^t = \mathbf{0}_{k\times 1}, i \in \{1, 2, \ldots, n-k\}$. Como cada fila de H da cero con cada fila de G, entonces, también dará cero para cualquier combinación lineal formada por las filas de G. Por lo tanto, tenemos que H es una matriz de dimensión $n-k \times n$, $\forall x \in \mathcal{C}$ verifica que $Hx^t = \mathbf{0}_{n-k\times 1}$, y $GH^t = \mathbf{0}_{k\times n-k}$.

De lo anterior, tenemos la siguiente definición.

Definición 2.1.18. *Sea $\mathcal{C}$ un código lineal de $\mathbb{F}_q^n$. Si H es una matriz de tamaño $(n-k)\times n$ de rango máximo, tal que $\mathcal{C} = \{x \in \mathbb{F}_q^n : Hx^t = \mathbf{0}\}$, entonces H es llamada **matriz control de paridad**.*

Definición 2.1.19. *Sea $\mathcal{C}$ un código lineal de $\mathbb{F}_q^n$. Definimos el **peso mínimo** de $\mathcal{C}$ como:*

$$\mathrm{wt}(\mathcal{C}) = \text{mín}\{wt(x) : \forall x \in \mathcal{C}, x \neq 0\}.$$

Con estas definiciones, tenemos los siguientes resultados para un código lineal $\mathcal{C}$.

Proposición 2.1.20. *Sea $\mathcal{C}$ un código lineal $[n,k,d]$ y H la matriz de control de paridad de $\mathcal{C}$. Entonces*

a) $\text{wt}(\mathcal{C}) = d$.

b) *d es igual al mínimo número de columnas linealmente dependientes en H.*

c) $k + d \leq n + 1$.

Observación: Por c) de la Proposición 2.1.20, da una nueva cota para los códigos lineales y es llamada **cota de Singleton**. Además, los códigos lineales que cumplen la igualdad en c) son llamados código de **máxima distancia de separación** o **MDS** por sus siglas en inglés.

Demostración.

a) Por definición de d tenemos:

$$\begin{aligned} d &= min\{d(v,w) : \forall v, w \in \mathcal{C}, v \neq w\} \\ &= min\{wt(v-w) : \forall v, w \in \mathcal{C}, v - w \neq 0\} \\ &= min\{wt(x) : \forall x \in \mathcal{C}, x \neq 0\} \text{ ya que } v - w \in \mathcal{C} \text{ y } x = v - w \\ &= \text{wt}(\mathcal{C}). \end{aligned}$$

b) Por a) sabemos que existe un palabra x de $\mathcal{C}$ tal que $\text{wt}(x) = d$, y por definición de H tenemos:

$$Hx^t = \mathbf{0}$$

$$\begin{pmatrix} c_1 & c_2 & \dots & c_n \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = 0.$$

Donde:

$$H = \begin{pmatrix} c_1 & c_2 & \dots & c_n \end{pmatrix}_{(n-k)\times n}, \text{ para } c_i \in \mathbb{F}_q^n, \forall i = 1, 2, \dots, n.$$
$$x = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix}_{1\times n}, \text{ para } x_i \in \mathbb{F}_q, \forall i = 1, 2, \dots, n.$$

Así:

$$\begin{pmatrix} c_1 & c_2 & \dots & c_n \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = 0$$

$$x_q c_1 + x_2 c_2 + \ldots + x_n c_n = 0.$$

De esta última ecuación tenemos que d es igual al mínimo número de columnas linealmente dependientes de H, ya que, si hubieran menos tendríamos que existe un $x = (x_1, x_2, \dots, x_n) \in \mathcal{C}$, tal que $\text{wt}(x) < d$ y eso sería una contradicción.

c) Por b) sabemos que d es el mínimo número de columnas linealmente dependientes, en otras palabras $d-1$ columnas son linealmente independientes. Además, por definición de H tenemos que el rango de H es $n-k$; y tendremos que $d-1 \leq n-k$. Así:

$$\begin{aligned} k+d-1 &\leq k+n-k=n \\ k+d-1 &\leq n \\ k+d &\leq n+1. \end{aligned}$$ □

Una consecuencia inmediata de la construcción de H, es que las filas de H son linealmente independientes, ya que el rango de H es $n-k$. Así, podemos considerar el código generado por dichas filas.

Definición 2.1.21. *Sea H la matriz de control de paridad del código lineal $\mathcal{C}$. Si tomamos la matriz H como matriz generadora, el código que genera es llamado código **dual** de $\mathcal{C}$, y se denota por $\mathcal{C}^{\perp}$.*

Proposición 2.1.22. *Sea H la matriz de control de paridad de $\mathcal{C}$, donde $\mathcal{C}$ es un código lineal $[n,k,d]$. Entonces:*

a) $\mathcal{C}^{\perp}$ es un código lineal $[n, n-k]$.

b) $\mathcal{C}^{\perp} = \{x \in \mathbb{F}_q^n : x \cdot c = 0, \forall c \in \mathcal{C}\}$, donde $\cdot$ es el producto escalar usual en $\mathbb{F}_q^n$.[1]

Demostración.

a) Como el rango de H es $n-k$, quiere decir que las filas de H ya son linealmente independientes y de longitud n. Por lo tanto, forman la base para $\mathcal{C}^{\perp}$ y como subespacio lineal tendrá dimensión $n-k$. Así, $\mathcal{C}^{\perp}$ es un código $[n, n-k]$.

b) Como H es la matriz generadora de $\mathcal{C}^{\perp}$, tenemos:

$$\begin{aligned} \mathcal{C}^{\perp} &= \{x \in \mathbb{F}_q^n : x = vH, v \in \mathbb{F}_q^{n-k}\} \\ &= \{x \in \mathbb{F}_q^n : x \cdot c = vH \cdot c, \forall c \in \mathcal{C}, v \in \mathbb{F}_q^{n-k}\} \\ &= \{x \in \mathbb{F}_q^n : x \cdot c = 0, \forall c \in \mathcal{C}\}. \end{aligned}$$ □

Observación: b) de la Proposición 2.1.22 nos dice que el dual del dual de un código es el mismo código.

2.1.2. Decodificación para códigos lineales

Así, como tenemos un método de codificación fácil para códigos lineales, también podemos definir un método para la decodificación. Este método utiliza la matriz de control de paridad de los códigos lineales.

[1] Si $x, y \in \mathbb{F}_q^n \Rightarrow x \cdot y = \sum_{i=1}^{n} x_i y_i$, donde $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n)$.

Definición 2.1.23. *Sea $\mathcal{C}$ un código lineal $[n,k,d]$, y sea H la matriz de control de paridad de $\mathcal{C}$. Definimos aplicación $r : \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$, tal que $r(x) = Hx^t, \forall x \in \mathbb{F}_q^n$. La aplicación r se llama* ***función síndrome*** *y $r(x)$ es llamado simplemente* ***síndrome de*** *x.*

Por definición, r es una aplicación lineal. Además, por la definición 2.1.18 el código lineal $\mathcal{C}$ es el núcleo de r. Luego hay una relación inmediata de los elementos de $\mathbb{F}_q^n/\mathcal{C}$ y r, y está expresada en la siguiente proposición.

Proposición 2.1.24. *Sea $\mathcal{C}$ un código lineal $[n,k,d]$, y sea H la matriz de control de paridad de $\mathcal{C}$. Para dos elementos que estén en la misma clase de $\mathbb{F}_q^n/\mathcal{C}$, tenemos que $x+\mathcal{C} = y+\mathcal{C}$ si, y sólo si $r(x) = r(y)$.*

Observación: Dicho de otra manera, las palabras que están en la misma clase tienen el mismo síndrome.

Demostración. Primero sea $x, y \in \mathbb{F}_q^n$, tales que $x+\mathcal{C} = y+\mathcal{C}$. Entonces, aplicando r a la igualdad tenemos:

$$\begin{aligned} x+\mathcal{C} &= y+\mathcal{C} \\ r(x+\mathcal{C}) &= r(y+\mathcal{C}) \\ r(x)+r(\mathcal{C}) &= r(y)+r(\mathcal{C}) \text{ ya que } r(\mathcal{C}) = 0 \\ r(x) &= r(y). \end{aligned}$$

Ahora si $r(x) = r(y)$, para $x, y \in \mathbb{F}_q^n$. Tenemos:

$$\begin{aligned} r(x) &= r(y) \\ r(x)-r(y) &= 0 \\ r(x-y) &= 0. \end{aligned}$$

Entonces, $x-y \in \mathcal{C} \Rightarrow x+\mathcal{C} = y+\mathcal{C}$. □

Debido a que los elementos de $\mathbb{F}_q^n$ que están en la misma clase, tienen el mismo síndrome. Podemos simplemente fijarnos en un elemento de la clase.

Definición 2.1.25. *Sea $\mathcal{C}$ un código lineal $[n,k,d]$, y sea H la matriz de control de paridad de $\mathcal{C}$. Llamaremos* ***líder*** *al elemento de menor peso de la clase $x+\mathcal{C} \in \mathbb{F}_q^n/\mathcal{C}$. Al líder lo denotamos por $\mathbf{1}$, si no hay ambigüedad a la clase que se hace referencia, o $\mathrm{cl}(r(x))$ para indicar a qué clase pertenece el líder (por definición $\mathbf{1} = \mathrm{cl}(r(x))$). Además, $r(\mathrm{cl}(r(x))) = r(\mathbf{1}) = r(x)$, por estar en la misma clase.*

Observación: Si hay más de un elemento con peso mínimo en las clases, se puede elegir uno al azar como líder (si no hay mayor información para la decodificación).

Proposición 2.1.26. *Sea $\mathcal{C}$ un código lineal $[n,k,d]$, y sea H la matriz de control de paridad de $\mathcal{C}$. Entonces:*

a) $d(x,C) = \text{wt}(\text{cl}(r(x))), \forall x \in \mathbb{F}_q^n$.

b) El radio de recubrimiento $\rho = \text{máx}\{\text{wt}(\boldsymbol{1}) : \boldsymbol{1}$ *es el líder de las clases*$\}$.

Demostración.

- Para a). Sea $x \in \mathbb{F}_q^n$ un elemento arbitrario, así:

$$\begin{aligned} d(x,\mathcal{C}) &= \text{mín}\{d(x,y) : y \in \mathcal{C}\} \\ &= \text{mín}\{\text{wt}(x-y) : y \in \mathcal{C}\} \\ &= \text{mín}\{\text{wt}(x+z) : z = -y, z \in \mathcal{C}\} \text{ ya que } \mathcal{C} \text{ es un subespacio lineal} \\ &= \text{mín}\{\text{wt}(x+z) : x+z \in x+\mathcal{C}\} \text{ por definición de clase} \\ &= \text{wt}(\text{cl}(r(x))) \text{ por definición de } \text{cl}(r(x)). \end{aligned}$$

- Para b).

$$\begin{aligned} \rho &= \text{máx}\{d(x,\mathcal{C}) : x \in \mathbb{F}_q^n\} \\ &= \text{máx}\{\text{cl}(r(x)) : x \in \mathbb{F}_q^n\} \text{ por a)} \\ &= \text{máx}\{\text{wt}(\mathbf{1}) : \mathbf{1} \text{ es el líder de las clases}\}. \end{aligned}$$ □

Consideremos e como en la Definición 2.1.8, entonces $e = x - c$, para $x \in \mathbb{F}_q^n$, y $c \in \mathcal{C}$. Así, al aplicar r a la ecuación anterior, tenemos $r(e) = r(x-c) = r(x) - r(c) = r(x) - 0 = r(x)$. En otras palabras, tanto el error e como x (el mensaje que se recibe) tienen el mismo síndrome.

Lo anterior, permite establecer un método para la decodificación para los códigos lineales, ya que tanto el error como el mensaje recibido tiene el mismo síndrome. Entonces, es fácil calcular el mensaje enviado, considerando que el error de la transmisión es el líder de las clases[2].

Teorema 2.1.27 (Decodificación por síndrome). *Sea $\mathcal{C}$ un código lineal $[n,k,d]$, y sea H la matriz de control de paridad de $\mathcal{C}$. Entonces la aplicación* $\text{dec}_{\mathcal{C}} : \mathbb{F}_q^n \to \mathcal{C}$ *tal que* $\text{dec}_{\mathcal{C}}(x) = x - \text{cl}(r(x))$ *es llamado* ***decodificación por síndrome****.*

Demostración. Vamos a verificar que $\text{dec}_{\mathcal{C}}$ definida como $\text{dec}_{\mathcal{C}}(x) = x - \text{cl}(r(x)), \forall x \in \mathbb{F}_q^n$, cumplen la Definición 2.1.6. Así, veamos primero que $\text{dec}_{\mathcal{C}}(x) \in \mathcal{C}$.

$$\begin{aligned} r(x - \text{cl}(r(x))) &= r(x) - r(\text{cl}(r(x))) \\ &= r(x) - r(x) \\ &= 0. \end{aligned}$$

[2]Se considera el líder, ya que se espera que el error ocasionado en la transmisión sea mínimo en relación a la capacidad de corrección del código.

Así, tenemos $x - \text{cl}(x) \in \mathcal{C} \Rightarrow \text{dec}_{\mathcal{C}}(x) \in \mathcal{C}$.

Por otro lado, como $\text{dec}_{\mathcal{C}}(x) \in \mathcal{C}$, supongamos por contradicción que $d(x, \text{dec}_{\mathcal{C}}(x)) \neq d(x, \mathcal{C})$. Entonces, existe $c' \in \mathcal{C}$, tal que $d(x, c') < d(x, \text{dec}_{\mathcal{C}}(x))$, así:

$$\begin{aligned} d(x, c') &< d(x, \text{dec}_{\mathcal{C}}(x)) \\ \text{wt}(x - c') &< \text{wt}(x - \text{dec}_{\mathcal{C}}(x)) \\ \text{wt}(x - c') &< \text{wt}(x - (x - \text{cl}(r(x)))) \\ \text{wt}(x - c') &< \text{wt}(\text{cl}(r(x))) \\ \text{wt}(x - c') &< \text{wt}(\mathbf{1}). \end{aligned}$$

Además, $r(x - c') = r(x) - r(c') = r(x) - 0 = r(x)$, entonces por 2.1.24, $x - c'$ y x están en la misma clase, y además, $x - c'$ tiene peso menor al líder, lo cual es una contradicción a la definición del líder. Por lo tanto, debe cumplirse que $d(x, \text{dec}_{\mathcal{C}}(x)) = d(x, \mathcal{C})$. Así, hemos probado que se verifica 2.1.6, y por lo cual tenemos un método de decodificación para $\mathcal{C}$. □

A continuación, veamos un ejemplo de cómo funciona este tipo de decodificación.

Ejemplo 2.1.28. *Sea $\mathcal{C} = \{0000, 1011, 0101, 1110\}$ de $\mathbb{F}_2^4$, encontrar cual fue la palabra v de $\mathcal{C}$ más probable de haber sido enviado si se recibe $w = 1101$.*

Solución

Primero calculamos las clases de $\mathcal{C}$. Las clases las colocamos en columnas de la siguiente manera:

0000	1000	0100	0010
1011	0011	1111	1001
0101	1101	0001	0111
1110	0110	1010	1100

La clase donde está contenida la palabra $w = 1101$ es la segunda columna. Entonces de esa clase tomamos la palabra de menor peso la cual es 1000. Así, tendremos que v es:

$$v = u + w = 1000 + 1101 = 0101$$

Por lo tanto, la palabra del código **más cercana** al vector recibido w es 0101.

2.2. Códigos de Hamming

Luego de haber hecho un repaso de la teoría que corresponde a los códigos correctores de errores, presentamos un tipo de código muy utilizado por sus propiedades. A lo largo de esta sección trabajaremos sobre el cuerpo $\mathbb{F}_q$, donde q es una potencia de la primo.

Los códigos de Hamming[3] son códigos detectores y correctores de errores. Estos códigos son capaces de detectar un máximo de dos errores y corregir un error. A continuación, mostramos cómo se define un código de Hamming sobre $\mathbb{F}_q$.

Definición 2.2.1. *Sea r un entero positivo diferente de 1. El código $\mathcal{C}$ con matriz de control de paridad H, tal que sus columnas son todas las palabras de $\mathbb{F}_q^r$ diferentes de cero, que no son múltiplos escalares dos a dos entre ellas, es llamado* ***código de Hamming****.*

Observación: Un código de Hamming se denota por: $\mathrm{Ham}(r, q)$.

Notemos que cada palabra en $\mathbb{F}_q^r$ tiene $q-1$ múltiplos escalares diferentes de cero. Así, en $\mathbb{F}_q^r$ debe haber $(q^r-1)/(q-1)$ palabras diferentes que no son múltiplos escalares dos a dos. Por lo tanto, $\mathrm{Ham}(r, q)$ es un código que pertenece a $\mathbb{F}_q^n$, donde $n = \frac{q^r-1}{q-1}$.

A continuación, veamos los parámetros principales de $\mathrm{Ham}(r, q)$.

Proposición 2.2.2. *Sea $r \geq 2$ un número entero. El código $\mathrm{Ham}(r, q)$ tiene distancia $d = 3$.*

Demostración. Sea H la matriz de control de paridad de $\mathrm{Ham}(r, q)$. Por la Definición 2.2.1, tenemos que las columnas de H al tomar dos a dos no son múltiplos escalares, luego son linealmente independientes. Pero al tomar tres, por ejemplo tendríamos $100\ldots00$, $010\ldots00$ y $110\ldots00$ palabras de $\mathbb{F}_q^r$ que son linealmente dependientes. Por lo tanto, por la Proposición 2.1.20, tenemos $d = 3$. □

Proposición 2.2.3. *Sea $r \geq 2$ un número entero. El código $\mathrm{Ham}(r, q)$ tiene dimensión $k = n - r$, donde $n = (q^r-1)/(q-1)$.*

Demostración. Sea H la matriz de control de paridad de $\mathrm{Ham}(r, q)$. Por la Definición 2.1.21, H es la matriz generadora del dual de $\mathrm{Ham}(r, q)$. Además, por definición de $\mathrm{Ham}(r, q)$ la matriz H tiene rango r, ya que las columnas son palabras de $\mathbb{F}_q^r$, y están las palabras canónicas (solo tienen una entrada diferente de cero). Esto hace que las r filas sean linealmente independientes, entonces el rango de H es r. Así, $\dim(\mathrm{Ham}^{\perp}(r, q)) = r$. Además, por la Definición 2.2.1, $n = (q^r-1)/(q-1)$.

[3]El nombre de **código de Hamming** es por su descubridor Richard Wesley Hamming (Chicago, Illinois, 11 de febrero de 1915 – Monterey, California, 7 de enero de 1998).

Por otro lado, por la Proposición 2.1.22 tenemos que $\dim(\text{Ham}^{\perp}(r,q)) = n-k$, donde k es la dimensión de $\text{Ham}(r,q)$. Así, por todo lo anterior tenemos:

$$\begin{aligned} n-k &= \dim(\text{Ham}^{\perp}(r,q)) \\ n-k &= r \\ k &= n-r. \end{aligned}$$

Por lo tanto, la dimensión de $\text{Ham}(r,q) = n-r$, donde $n = (q^r-1)/(q-1)$. □

Así, hemos demostrado que $\text{Ham}(r,q)$ es un código con parámetros $[n, n-r, 3]$, donde $n = (q^r-1)/(q-1)$. Con la información anterior podemos demostrar la propiedad de los códigos de Hamming, que es la capacidad de detección (errores y borrones), y su capacidad de corrección.

Teorema 2.2.4. *Sea $r \geq 2$ un número entero. El código $\text{Ham}(r,q)$ detecta un máximo de dos errores (o borrones), y corrige un error.*

Demostración. Por la Proposición 2.2.2, la distancia del código es 3, entonces por la Proposición 2.1.11, detecta 2 errores o borrones. Además, al calcular t (la capacidad de corrección), tenemos:

$$\begin{aligned} t &= \lfloor (d-1)/2 \rfloor \\ &= \lfloor (3-1)/2 \rfloor \\ &= \lfloor 2/2 \rfloor \\ &= 1. \end{aligned}$$

Por lo tanto, la capacidad de corrección del código es de 1. □

Teorema 2.2.5. *Sea $r \geq 2$ un número entero. El código $\text{Ham}(r,q)$ es un código perfecto.*

Demostración. Como la capacidad de corrección del código es 1, calculemos cuántas palabras están en la bola de radio 1 y centro una palabra del código. Así, sea $c \in \text{Ham}(r,q)$ arbitrario, entonces:

$$\begin{aligned} |B(c,1)| &= \sum_{i=0}^{1} \binom{n}{i} (q-1)^i \\ &= \binom{n}{0}(q-1)^0 + \binom{n}{1}(q-1)^1 \\ &= 1 + n(q-1). \end{aligned}$$

Además, por la Proposición 2.2.3, tenemos:

$$\begin{aligned} n &= \frac{q^r-1}{q-1} \\ n(q-1) &= q^r - 1 \\ q^r &= n(q-1)+1. \end{aligned}$$

También, sabemos que la dimensión del código es $k = n - r$. Entonces, la cantidad de palabras que tiene el código es $M = q^{n-r}$ (esto es así, porque tenemos $n - r$ palabras en la base, y q escalares para las combinaciones lineales). Finalmente, la cantidad de palabras de $\mathbb{F}_q^n$ que están en las bolas de radio 1 y centro una palabra del código, es:

$$\begin{aligned} M|B(c,1)| &= q^{n-r}(n(q-1)+1) \\ &= q^{n-r}q^r \\ &= q^n. \end{aligned}$$

Que es precisamente todas las palabras de $\mathbb{F}_q^n$. Así, tenemos que cumple la igualdad de la cota de Hamming. Por lo tanto, es un código perfecto. □

Ejemplo 2.2.6. *Sea $r = q = 3$, entonces el código* $\mathrm{Ham}(3,3)$ *tiene matriz control de paridad:*

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 2 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 \\ 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{pmatrix}.$$

Los parámetros del código anterior son $[13, 10, 3]$.

Ejemplo 2.2.7. *Sea $r = 3$ y $q = 2$, entonces el código* $\mathrm{Ham}(3,2)$ *tiene matriz control de paridad:*

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Los parámetros del código anterior son $[7, 4, 3]$. Además, usando la observación de la Proposición 2.1.22 y el proceso descrito para calcular H de un código. Tenemos que la matriz generadora de $\mathrm{Ham}(3,2)$ es:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

En ambos ejemplos, los procesos de codificación, detección y corrección de errores son como se describen en la sección anterior[4].

[4]Para codificar es multiplicar por la matriz generadora, y la detección con la corrección, es multiplicar la palabra recibida por la matriz control de paridad y seguir lo descrito en la subsección 2.1.2.

Capítulo 3

Esteganografía y códigos

En este último capitulo aprovecharemos la teoría desarrollada en los capítulos anteriores para desarrollar esquemas a partir de códigos.

3.1. Esteganografía a partir de códigos

El objetivo de esta sección es establecer la relación que existe entre la esteganografía y los códigos correctores de errores. Primero recordemos qué es una partición de un conjunto.

Definición 3.1.1. *Sea $\mathcal{A}$ un alfabeto de q elementos, y n un entero positivo. Entonces una familia de conjuntos $\{\mathcal{Y}_i\}_{i=1}^{t}$, donde $\mathcal{Y}_i \subset \mathcal{A}^n, \forall i = 1, 2, \ldots, t$, es una **partición** de $\mathcal{A}^n$, si $\bigcup_{i=1}^{t} \mathcal{Y}_i = \mathcal{A}^n$, y $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset, \forall i, j \in \{1, 2, \ldots, t\}$.*

Ahora, pasemos a mostrar los dos resultados que relacionan la esteganografía y los códigos correctores de errores.

Teorema 3.1.2 (De esteganografía a códigos). *Sea $\mathcal{S} = (emb, rec)$ un esquema del tipo $[n, k]$ adecuado sobre el alfabeto $\mathcal{A}$ de q elementos. Para cada $m \in \mathcal{A}^k$ consideramos el código $\mathcal{C}_m = \{x \in \mathcal{A}^n : rec(x) = m\}$. Entonces, la familia $\{\mathcal{C}_m : m \in \mathcal{A}^k\}$ da una partición de $\mathcal{A}^n$. Además, para todo $m \in \mathcal{A}^k$ la aplicación $\mathrm{dec}_m : \mathcal{A}^n \to \mathcal{C}_m$ definido por $dec_m(x) = emb(x, m)$ es una aplicación de decodificación por distancia mínima para el código $\mathcal{C}_m$.*

Demostración. Por Proposición 1.2.3, *rec* es sobreyectiva. Entonces, para cada $m \in \mathcal{A}^k$ el código $\mathcal{C}_m = \{x \in \mathcal{A}^n : rec(x) = m\} \neq \emptyset$. Además, por construcción, si $m \neq m'$, entonces $\mathcal{C}_m \cap \mathcal{C}_{m'} = \emptyset$, y además $\bigcup_{i=1}^{q^k} \mathcal{C}_i = \mathcal{A}^n$. Por lo tanto, la familia $\{\mathcal{C}_m : m \in \mathcal{A}^k\}$ es una partición para $\mathcal{A}^n$.

Además, por definición de $\mathcal{S}$ tenemos $rec(emb(x,m)) = m, \forall x \in \mathcal{A}^n, \forall m \in \mathcal{A}^k$. Entonces por definición de $\mathcal{C}_m$, se tienen que $emb(x,m) \in \mathcal{C}_m \Rightarrow \text{dec}_m(x) \in \mathcal{C}_m, \forall x \in \mathcal{A}^n, \forall m \in \mathcal{A}^k$. También, como $\mathcal{S}$ es adecuado, cumple lo siguiente:

$$\begin{aligned} d(x, \text{dec}_m(x)) &= d(x, emb(x,m)) \text{ por definición de } \text{dec}_m \\ &= d(x, rec^{-1}(m)) \text{ por 1.2.12} \\ &= d(x, \mathcal{C}_m) \text{ por definición de } \mathcal{C}_m. \end{aligned}$$

Así, tenemos que cumple 2.1.6. □

El resultado anterior nos permite establecer cómo crear códigos y sus métodos de decodificación a partir de un esquema esteganográfico. Ahora, veamos cómo construir un esquema esteganográfico a partir de códigos.

Teorema 3.1.3 (De códigos a esteganografía). *Sea $\{\mathcal{C}_m : m \in \mathcal{A}^k\}$ una familia de códigos indexados por $\mathcal{A}^k$ y dando una partición de $\mathcal{A}^n$. Para cada $m \in \mathcal{A}^k$, sea dec_m es una aplicación de decodificación por distancia mínima de $\mathcal{C}_m$. Considere las aplicaciones $emb : \mathcal{A}^n \times \mathcal{A}^k \to \mathcal{A}^n$ y $rec : \mathcal{A}^n \to \mathcal{A}^k$ definidos por $emb(x,m) = \text{dec}_m(x)$ y $rec(x) = m$, si $x \in \mathcal{C}_m$. Entonces $\mathcal{S} = (emb, rec)$ es un esquema del tipo $[n,k]$ adecuado sobre $\mathcal{A}$ un alfabeto de q elementos.*

Demostración. Primero, veamos que $\mathcal{S}$ es un esquema esteganográfico. Por definición dec_m, tenemos que $\text{dec}_m(x) \in \mathcal{C}_m, \forall x \in \mathcal{A}^n, \forall m \in \mathcal{A}^k \Rightarrow rec(\text{dec}_m(x)) = m, \forall x \in \mathcal{A}^n, \forall m \in \mathcal{A}^k$. Además, por definición de emb, tendremos $rec(emb(x,m)) = m, \forall x \in \mathcal{A}^n, \forall m \in \mathcal{A}^k$. Por lo cual tenemos que $\mathcal{S} = (emb, rec)$ es un esquema del tipo $[n,k]$.

Finalmente, probemos que $\mathcal{S}$ es un esquema adecuado. Como $rec(x) = m$, si $x \in \mathcal{C}_m$, y como $\{\mathcal{C}_m : m \in \mathcal{A}^k\}$ forman una partición para $\mathcal{A}^n$. Así, para $x \in \mathcal{A}^n$ solo pertenece a uno y solamente a uno de los códigos de la familia, entonces tenemos que $rec(m)^{-1} = \mathcal{C}_m$. Con todo lo anterior tendremos:

$$\begin{aligned} d(x, emb(x,m)) &= d(x, \text{dec}_m(x)) \text{ por definición de } \text{dec}_m \\ &= d(x, \mathcal{C}_m) \text{ por ser la decodificación de } \mathcal{C}_m \\ &= d(x, rec^{-1}(m)). \end{aligned}$$

Por lo tanto, $\mathcal{S}$ es adecuado. □

Hemos probado y mostrado una forma de cómo construir un esquema esteganográfico a partir de una familia de códigos. Además, como conclusión general tenemos que los dos objetos siguientes son equivalentes:

- $\mathcal{S} = (emb, rec)$ es un esquema del tipo $[n,k]$ adecuado sobre $\mathcal{A}$ un alfabeto de q elementos.

- $\{\mathcal{C}_m : m \in \mathcal{A}^k\}$ una familia de códigos indexados por $\mathcal{A}^k$. Además, la familia de códigos da una partición de $\mathcal{A}^n$. Para cada $m \in \mathcal{A}^k$, sea dec_m es una aplicación de decodificación de distancia mínima de $\mathcal{C}_m$.

Finalmente, hemos logrado dar una relación de construcción entre la esteganografía y los códigos. Algo que aprovecharemos a lo largo de este trabajo.

3.1.1. Esteganografía basado en el código de Hamming binario

En el capítulo 1, vimos un ejemplo de cómo incrustar información en el mapa de bits. Sin embargo, no era tan eficiente, además de ser muy predecible el lugar donde se estaba colocando el mensaje. Sin embargo, considerando los Teoremas 3.1.2 y 3.1.3, podemos aprovechar las herramientas de los códigos lineales para crear métodos de incrustación y recuperación más eficientes a partir de las propiedades de los códigos.

Para comenzar, consideremos $z \in \mathbb{F}_q^n$ y su clase $\mathcal{C} + z$, donde $\mathcal{C}$ es un código lineal en $\mathbb{F}_q^n$, con decodificación completa. Si $x \in \mathbb{F}_q^n$ al considerar los trasladados de ambos objetos (de él y su clase), tendremos: $x - z$ y $\mathcal{C} + z - z = \mathcal{C}$. Entonces, para $x - z$, podemos encontrar $y = \text{dec}(x - z)$, por definición tenemos que:

$$d(x - z, y) < d(x - z, c), \forall c \in \mathcal{C}.$$

Lo anterior se puede trasladar a la clase de z, entonces:

$$\begin{aligned} d(x - z, y) &< d(x - z, c), \forall c \in \mathcal{C} \\ \text{wt}(x - z - y) &< \text{wt}(x - z - c), \forall c \in \mathcal{C} \\ d(x, y + z) &< d(x, c + z), \forall c \in \mathcal{C}. \end{aligned}$$

Los procesos son reversibles, y por lo tanto, significa que si podemos decodificar en $\mathcal{C}$ también lo podemos hacer en la clase $\mathcal{C} + z$.[1] Ahora notemos que el síndrome será $r(x - z) = H(x - z)^t$, donde H es la matriz de control de paridad para le código $\mathcal{C}$. Además, notemos que:

$$\begin{aligned} r(x - z) &= H(x - z)^t \\ &= Hx^t - Hz^t. \end{aligned}$$

Esta última igualdad tenemos que el único que varía será Hx^t, y además tendremos:

$$\begin{aligned} y = \text{dec}(x - z) &= x - z - \text{cl}(r(x - z)) \quad \text{por el Teorema 2.1.27,} \\ &= x - z - \text{cl}(Hx^t - Hz^t). \end{aligned}$$

[1] En otras palabras, las propiedades y métodos de codificación y decodificación de los códigos lineales, se mantienen por traslaciones.

Que para el código trasladado sería: $y+z = \text{dec}(x-z)+z = x-\text{cl}(Hx^t - Hz^t)$. Esto quiere decir que el método de decodificación queda perfectamente definido, a partir del síndrome de la palabra usada para crear la clase. Además, Hz^t será nuestro mensaje a incrustar m ($m = Hz^t$). Así, esta es la idea principal para crear las familias de códigos del Teorema 3.1.3. A continuación, veamos un ejemplo de las ideas expuesta en $\mathbb{F}_2^n$.

Consideremos $\text{Ham}(r,2)$, sabemos por el Teorema 2.2.5 que es un código perfecto, esto quiere decir que las clases forman una partición de $\mathbb{F}_2^n$. Además, el código tiene dimensión $n-r = (2^r-1)-r$, así concluimos que hay r clases diferentes. Por lo tanto, tendremos el esquema $\mathcal{S}$ del tipo $[2^r-1, r]$, donde las funciones de incrustación y recuperación están definidas por:

$$\begin{aligned} emb : \mathbb{F}_2^n \times \mathbb{F}_2^r &\longrightarrow \mathbb{F}_2^n & \qquad rec : \mathbb{F}_2^n &\longrightarrow \mathbb{F}_2^r \\ (x,m) &\longmapsto x+\text{cl}(Hx^t+m) & \qquad v &\longmapsto Hv^t \end{aligned}$$

Observación: Se escribe $x+\text{cl}(Hx^t+m)$ en vez de $x-\text{cl}(Hx^t+m)$, porque estamos trabajando con el cuerpo $\mathbb{F}_2$.

Por último, calculemos la carga útil (tasa de incrustación) de este esquema. Entonces, por los parámetros del capítulo 1, tendremos:

$$\alpha = \frac{r}{2^r-1} < \frac{2^r-1-r}{2^r-1}, \quad \text{Para } r \geq 3.$$

La fracción de la derecha es la carga útil del método presentado al final del capítulo 1. Esto quiere decir que a comparación del esquema visto en el capítulo 1, es mejor el que genera el código de Hamming; por requerir menos bits de la cubierta para incrustar el mismo mensaje.

3.2. Selección no compartida

Otro hecho que hemos pasado por alto, es que se ha asumido (como se vio en la sección anterior), que la decodificación es completa. Lo cual es necesario, porque sino no podríamos incrustar; ya que la función incrustar depende de la decodificación del código. En ese sentido, debemos siempre garantizar que la decodificación sea completa[2].

Para comenzar a tratar el problema de la selección no compartida, daremos una definición que serán necesarias a lo largo de esta sección.

[2]Incluso a nivel computacional, aunque tengamos una decodificación completa, será siempre más útil un código que sea computacionalmente eficiente su decodificación, y este hecho es lo que vuelve tan buenos los códigos de Hamming.

Definición 3.2.1. *Un esquema esteganográfico se llama **adaptativo**, cuando el esquema depende de la imagen de la portada.*

Precisamente, cuando el esquema está sujeto al tipo de imagen a usar, es cuando tenemos que el receptor no sabrá dónde está oculto el mensaje; ya que se buscarán los píxeles de la imagen donde no son homogéneos. Así, aunque se haya acordado los lugares con anterioridad, la imagen impedirá usarlos (porque lo que se busca es no llamar la atención, con los cambios realizados). A este hecho es lo que se conoce como **selección no compartida**.

3.2.1. Esquemas de papel mojado

El método a tratar para esquemas adaptativos son conocidos como **esquemas de papel mojado**. Recordemos que siempre estamos considerando a q como una potencia de un número primo.

Definición 3.2.2. *Sean n y r enteros positivos, con $n > r$. Además, considere los subconjuntos D y L de $\{1, 2, \ldots, n\}$, tal que $|D| = \delta \geq r$ y $|L| = l = n - \delta$, con $L = \{1, 2, \ldots, n\} - D$. Diremos que el conjunto D es un conjunto de **coordenadas secas**, y L un conjunto de **coordenadas mojadas**.*

Observación: La idea es que dado $x \in \mathbb{F}_q^n$, los x_i, con $i \in D$, se podrán modificar (están libres). Por otro lado, los x_i, con $i \in L$, no se podrán modificar (están bloqueadas).

Definición 3.2.3. *Sea el esquema $\mathcal{S}$ del tipo $[n, r]$. Dados L y D como en la Definición 3.2.2. Llamaremos **esquema de papel mojado**, cuando $\mathcal{S}$ deja sin cambios los bits con coordenadas en L de la cubierta.*

Por la definición anterior, es que recibe el nombre de **papel mojado**. Teniendo en cuenta la Definición 3.2.2, y los métodos descritos en la sección anterior, el problema a resolver es:

$$[\mathcal{S}] = \begin{cases} Hv^t = m \\ \\ v_i = x_i \qquad si \quad i \in L \end{cases}$$

Para x, m, tal que $emb(x, m) = v \in \mathbb{F}_q^n$, con $d(v, x)$ más pequeño posible verificando el sistema anterior, y alguna matriz de control de paridad H de un $[n, n-r]$ código lineal $\mathcal{C}$. Ahora nos concentraremos en el caso de $q = 2$.

Proposición 3.2.4. *Sea $\mathcal{C}$ un código lineal de $\mathbb{F}_2^n$. $[\mathcal{S}]$ tiene una solución para $x \in \mathbb{F}_2^n$, $m \in \mathbb{F}_2^r$, y $D \subseteq \{1, 2, \ldots, n\}$, si y solo si $\pi_L(x) \in \pi_L(\mathrm{cl}(m) + \mathcal{C})$, donde π_L es la función proyección sobre las coordenadas mojadas L.*

Demostración. Sea $L = \{1, 2, \ldots, n\} - D$. Por las propiedades de las clases de un código lineal $\mathcal{C}$ (2.1.24), podemos considerar la clase $\{v \in \mathbb{F}_2^n : Hv^t = m\} = \text{cl}(m) + \mathcal{C}$, y la función proyección π sobre las posiciones del conjunto L. Tendremos que $[\mathcal{S}]$ es equivalente a:

$$[\mathcal{S}] = \begin{cases} v \in \text{cl}(m) + \mathcal{C} \\ \\ \pi_L(v) = \pi_L(x) \end{cases}$$

Ahora, si v es solución de $[\mathcal{S}]$, entonces $v \in \text{cl}(m) + \mathcal{C}$ y $\pi_L(v) = \pi_L(x)$ que equivalente a decir que $\pi_L(x) \in \pi_L(\text{cl}(m) + \mathcal{C})$.

Por otro lado, si $\pi_L(x) \in \pi_L(\text{cl}(m) + \mathcal{C})$, entonces existe $v \in \mathbb{F}_2^n$, tal que $v \in \text{cl}(m) + \mathcal{C}$ y $\pi_L(v) = \pi_L(x)$, que significa que v es solución de $[\mathcal{S}]$. □

Las condiciones de bloqueo (los x_i que no cambian) harán que el sistema $[\mathcal{S}]$ sea difícil de resolver, y además que no siempre exista una solución. Sin embargo, en el siguiente resultado, podemos tener algunas condiciones para garantizar solución al problema propuesto, sobre ciertos parámetros de los códigos.

Teorema 3.2.5. *Sea $\mathcal{C}$ un código lineal $[n, n-r]$ de $\mathbb{F}_2^n$. $[\mathcal{S}]$ tiene una solución para cada $x \in \mathbb{F}_2^n$, y cada $m \in \mathbb{F}_2^r$, y $D \subseteq \{1, 2, \ldots, n\}$ con $|D| = \delta$, si y solo si $\delta \geq n - d^\perp + 1$. Donde $d^\perp$ es la distancia del código dual $\mathcal{C}^\perp$.*

Demostración. Sea $L = \{1, 2, \ldots, n\} - D$ y tomemos $x \in \mathbb{F}_2^n$, $m \in \mathbb{F}_2^r$ arbitrarios. Si $\delta \geq n - d^\perp + 1 \iff d^\perp \geq n - \delta + 1 > n - \delta = |L|$. Lo cual significa que no existe $c \in \mathcal{C}^\perp$ diferente de cero, tal que $\text{supp}(c) \subset L$, ya que por definición de distancia mínima de un código $d^\perp \leq \text{wt}(c) = |\text{supp}(c)|$.

Lo anterior, quiere decir que al tomar la matriz generadora G de $\mathcal{C}$, y al ser la matriz de control de paridad para $\mathcal{C}^\perp$, por la Proposición 2.1.20 aplicado a $\mathcal{C}^\perp$. La mínima cantidad de columnas que podrán ser linealmente independientes en G es $n - \delta + 1$. Esto quiere decir que la matriz G_L de dimensión $n - r \times n - \delta$, que se obtiene de G (eliminando las columnas con indices en D), tiene rango $l = n - \delta$; ya que por la Definición 3.2.2, tenemos $\delta \geq r \Rightarrow n - \delta \leq n - r$.

Por otro lado, notemos que:

$$|\pi_L(\text{cl}(m) + \mathcal{C})| = |\pi_L(\mathcal{C})|.$$

Y $\pi_L(\mathcal{C})$ está generado por G_L; al tener rango l, quiere decir que tiene 2^l elementos que es exactamente la misma cantidad de palabras que $\mathbb{F}_2^l$, Así:

$$\pi_L(\text{cl}(m) + \mathcal{C}) = \mathbb{F}_2^l.$$

Luego $\pi_L(x) \in \mathbb{F}_2^l = \pi_L(\text{cl}(m) + \mathcal{C}) \Rightarrow \pi_L(x) \in \pi_L(\text{cl}(m) + \mathcal{C})$ y por la Proposición 3.2.4, tenemos que $[\mathcal{S}]$ tiene solución.

Ahora, si $[\mathcal{S}]$ tiene solución para cada $x \in \mathbb{F}_2^n$, y cada $m \in \mathbb{F}_2^r$, entonces por la Proposición 3.2.4, tenemos que $\pi_L(x) \in \pi_L(\text{cl}(m) + \mathcal{C})$, y por los últimos procesos de la demostración anterior tendremos que:

$$\pi_L(\text{cl}(m) + \mathcal{C}) = \mathbb{F}_2^l.$$

Lo que significa que el rango de G_L es $l = n - \delta$. Esto quiere decir que en G podemos encontrar $n - \delta + 1$ columnas linealmente dependientes y por la Proposición 2.1.20, para $\mathcal{C}^\perp$, tendremos que $d^\perp \geq n - \delta + 1 \Rightarrow \delta \geq n - d^\perp + 1$. □

Observación: Cuando tengamos que el rango de G_L es $n - \delta$, entonces tenemos $2^{\delta - r}$ soluciones para $[\mathcal{S}]$[3].

Otro hecho importante es que $[\mathcal{S}]$ puede tener solución, aun cuando $\delta < n - d^\perp + 1$; esto dependerá de los datos que definan el problema $[\mathcal{S}]$.

Para resolver $[\mathcal{S}]$, lo que tenemos que hacer primero es eliminar las posiciones bloqueadas (estas no cambian en el proceso de incrustación). Entonces, la solución v se puede escribir como $v = x + u$, donde $\text{supp}(u) \subseteq D$. Luego despejando u y aplicando H, tendremos:

$$\begin{aligned} Hu^r &= H(v + x)^t \\ &= Hv^t + Hx^t \\ &= m + Hx^t = z \in \mathbb{F}_2^r. \end{aligned}$$

Entonces, para determinar u, basta resolver $Hu^t = z$, ya que buscamos v, dados x y m. Así, eliminamos los ceros que corresponden a las posiciones en L, y de igual forma las columnas de H con posiciones en L. Lo cual transforma nuestro problema al sistema $My^t = z$, donde $y = \pi_L(u)$ y M es una matriz de tamaño $r \times \delta$, resultado de eliminar las columnas con posiciones en L de H. Cuando M tiene rango r (recordemos que $\delta \geq r$)[4] tenemos que M es una matriz de control de paridad de algún código $[\delta, \delta - r]$. Por lo tanto, calcular u es usar la función de decodificación del código con matriz de control de paridad M, y obtener u lo más cercano posible a x.

Así, encontrar una solución a $[\mathcal{S}]$ es usar un método de decodificación de un código $[\delta, \delta - r]$, con matriz de control de paridad M como se describe en el párrafo anterior. Además, notemos que el código $[\delta, \delta - r]$ depende de u y esté depende del conjunto D, que significa que al cambiar D cambiará el código a usar, a pesar de usar siempre $\mathcal{C}$, con matriz de control de paridad H. También el código $[\delta, \delta - r]$ al ser cualquiera puede suceder que no tenga una decodificación computacionalmente eficiente o completa. Lo anterior, hace que el problema sea más complicado de tratar que en el caso de no tener posiciones bloqueadas.

[3]Ya que para cada x, m y D, tendremos $\frac{|\mathcal{C}|}{|\pi_L(\mathcal{C})|} = \frac{2^{n-r}}{2^{n-\delta}} = 2^{\delta - r}$ soluciones.

[4]En este caso, la matriz recibe el nombre de matriz de **rango máximo**, porque $rank(M) = \text{mín}\{\delta, r\}$.

3.3. Descodificación: sin distancia mínima

En está última sección, buscaremos un método para decodificar sin distancia mínima, para un código de Hamming, que permita una decodificación eficiente para el problema $[\mathcal{S}]$. Primero estudiaremos las propiedades de un código que resulta de acortar un código de Hamming de forma arbitraria, para luego desarrollar un método de decodificación.

Definición 3.3.1. *Sea* $\mathrm{Ham}(r,2)$ *de* $\mathbb{F}_2^n$*, con matriz de control de paridad* H*, y* D *y* L *como en la Definición 3.2.2. Llamaremos* ***código acortado*** *de* $\mathrm{Ham}(r,2)$ *al conjunto de todas las palabras* $\pi_D(c)$*, tal que* $c \in \mathrm{Ham}(r,2)$ *y* $\pi_L(c) = 0$*. Al código acortado lo denotaremos por* $\mathcal{D}$*.*

Observación: Esta misma definición se puede aplicar a un código $\mathcal{C}$ arbitrario.

Por la condición $\pi_L(c) = 0$, tenemos que una matriz de control de paridad para $\mathcal{D}$ se obtiene de H eliminando las columnas con posiciones en L. La matriz de control de paridad de $\mathcal{D}$ la denotaremos por $\mathbf{D}$.

Proposición 3.3.2. *Sea* $\mathcal{D}$ *un código acortado de* $\mathrm{Ham}(r,2)$*, y* D *y* L *como en la Definición 3.2.2. El código* $\mathcal{D}$ *es un código lineal* $[n-l,k,d]$*, donde* $l = |L|$*,* $k \geq n-r-|L|$ *y* $d \geq 3$*.*

Demostración. Por definición de $\mathcal{D}$, las palabras son de tamaño $n-l$, donde $l = |L|$. La distancia de $\mathcal{D}$ debe ser mayor o igual que 3, ya que para $\mathrm{Ham}(r,2)$ la cantidad mínima de columnas linealmente dependientes es 3. Entonces, $\mathbf{D}$ tendrá como mínimo 3 columnas linealmente dependientes (porque si el mínimo fuera mayor que 3, por construcción también lo tendría H y eso sería una contradicción). Por lo tanto, $d \geq 3$.

Por construcción de la matriz de control de paridad para $\mathcal{D}$, la matriz $\mathbf{D}$ tiene dimensiones $r \times (n-l)$, que por la Definición 3.2.2, $n-l = \delta \geq r$. Así, el código dual de $\mathcal{D}$ podrá tener como máximo dimensión r, que para el código significará, que su dimensión es por lo menos $(n-l)-r$. Por lo tanto, tendremos: $k = \dim(\mathcal{D}) \geq n-l-r = n-r-|L|$. □

Si $\delta \geq n - d^{\perp} + 1$, donde $d^{\perp}$ la distancia de $\mathrm{Ham}(r,2)^{\perp}$, por el Teorema 3.2.5, tendremos solución para $[\mathcal{S}]$. Además, por lo descrito en la sección anterior, la matriz de control de paridad $\mathbf{D}$ de $\mathcal{D}$, es de rango máximo. Por lo cual tendremos que $\mathcal{D}$ es un código $[\delta, \delta - r]$, que es lo que nos interesa estudiar.

El parámetro ρ que se definió en el capítulo 1, puede ser aplicados para un código en especifico. Recordando que la función *emb* es la función de decodificación del código.

Proposición 3.3.3. *Sea* $\mathcal{C}$ *un código binario* $[\delta, \delta - r]$ *de distancia* $d > 2$*. Si* $\delta \geq 2^{r-1}$*, entonces* $\delta = 2^r - 1$ *y* $\mathcal{C}$ *es un código de Hamming con* $\rho = 1$*, o* $\delta < 2^r - 1$*, con* $\rho = 2$*.*

Demostración. Sea $F = \{f_1, f_2, \ldots, f_\delta\} \subset \mathbb{F}_2^r$ el conjunto de columnas de la matriz de control de paridad para $\mathcal{C}$. Como $d > 2$, entonces no existen i y j, tal que $\alpha_i f_i + \alpha_j f_j = 0$, con

$\alpha_i, \alpha_j \in \mathbb{F}_2$, lo que significa que el conjunto F no tiene elementos repetidos (ni el cero). Entonces, $\delta < 2^r$, y por hipótesis $\delta \geq 2^{r-1}$. Así, analicemos por casos.

Si $\delta = 2^r - 1$, entonces $F = \mathbb{F}_2^r - \{0\}$, entonces por la Definición 2.2.1 $\mathcal{C}$ es de Hamming. Esto quiere decir que tiene capacidad de corregir 1 error, y ser un código perfecto. Luego podemos incrustar un bit, y así $\rho = 1$.

Si $\delta < 2^r - 1$, entonces para $x \in \mathbb{F}_2^\delta$, tendremos que $[f_1, f_2, \ldots, f_\delta]x^t = s \in \mathbb{F}_2^r$, donde s es el síndrome de x. En este caso podemos caracterizar el síndrome. Si $s = 0$, entonces $x \in \mathcal{C}$, lo cual significa que $d(x, \text{dec}_\mathcal{C}(x)) = d(x, x) = 0$. Si $s \in F$, entonces existe i, tal que $s = f_i$. Así, tendremos que $x + e_i \in \mathcal{C} \Rightarrow d(x, \text{dec}_\mathcal{C}(x)) = d(x, x + e_i) = 1$. Si $s \notin F$, entonces al considerar el conjunto $s + F$, tendremos:

$$\begin{aligned} |\{0\}| + |F| + |s + F| &= 1 + \delta + \delta \\ &= 2\delta + 1 \\ &\geq 2(2^{r-1}) + 1 \\ &= 2^r + 1 \\ &> 2^r. \end{aligned}$$

Esto quiere decir que entre F y $s + F$ hay al menos un elemento en común, ya que no puede suceder que $\{0\} \subset s + F$. Entonces, tendríamos que $s = f_i$, para algún i, entonces $s \in F$, lo cuál es una contradicción. Por lo tanto, existen i y j, tal que $s + f_i = f_j$. Así, $s = f_i + f_j$, entonces $x + e_i + e_j \in \mathcal{C}$. Por lo cual, $d(x, \text{dec}_\mathcal{C}(x)) = d(x, x + e_i + e_j) = 2$, para los $2^r - \delta - 1$ síndromes restantes s. Finalmente, por definición de ρ, tendremos que $\rho = 2$. □

En la demostración, cuando $\delta < 2^r - 1$, obtuvimos varios valores para $d(x, \text{dec}_\mathcal{C}(x))$. Entonces, es normal preguntarse por el promedio. Este valor lo llamamos **promedio de incrustación** y lo denotamos por $\widetilde{\rho}$. Así, tenemos:

Si $\delta = 2^r - 1$,

$$\begin{aligned} \widetilde{\rho} &= \frac{2^r - 1}{2^r} \\ &= \frac{2(2^r - 1) - (2^r - 1)}{2^r} \\ &= \frac{2^{r+1} - 2 - \delta}{2^r} \\ &= \frac{2^{r+1} - (\delta + 2)}{2^r} \\ &= 2 - \frac{\delta + 2}{2^r}. \end{aligned}$$

Si $\delta < 2^r - 1$,

$$\begin{aligned}\widetilde{\rho} &= \frac{\delta + 2(2^r - \delta - 1)}{2^r} \\ &= \frac{\delta + 2^{r+1} - 2\delta - 2}{2^r} \\ &= \frac{2^{r+1} - (\delta + 2)}{2^r} \\ &= 2 - \frac{\delta + 2}{2^r}.\end{aligned}$$

Por lo tanto, un código como el de la Proposición 3.3.3 tiene un promedio de incrustación $\widetilde{\rho} = 2 - (\delta + 2)/2^r$.

Además, de la misma demostración se tiene que si $\delta < 2^{r-1}$, tendremos que pueden haber más síndromes s que sean de una palabra x, que esté a una distancia de $\mathcal{C}$ mayor a dos unidades. Por lo que $\widetilde{\rho} \geq 2 - (\delta + 2)/2^r$.

Con los resultados obtenidos podemos caracterizar aún más los esquemas $\mathcal{S}$ que son resultados de un código de Hamming, para resolver el problema $[\mathcal{S}]$.

Corolario 3.3.4. *Sea* $\mathrm{Ham}(r, 2)$ *con un dual de distancia* $d^\perp = 2^{r-1}$. *Dada* $x \in \mathbb{F}_2^n$ *una cubierta y* $L \subset \{1, 2, \ldots, n\}$ *con* $l \leq n/2$, *con* $n = 2^r - 1$. *El estegoesquema de papel mojado* $\mathcal{S}$ *que produce* $\mathrm{Ham}(r, 2)$, *incrusta con éxito cada mensaje* $m \in \mathbb{F}_2^r$ *en* x *con un máximo de dos modificaciones y con promedio de incrustación* $(n + l)/(n + 1) = \widetilde{\rho}(\mathrm{Ham}(r, 2)) + l/(n + 1)$.

Nota: Se usa la notación $\widetilde{\rho}(\cdot)$ para denotar a qué código corresponde el promedio de incrustación.

Demostración. Por definición del conjunto D, tenemos que:

$$\begin{aligned}\delta = n - l &\geq n - n/2 \\ &= n - (2^r - 1)/2 \\ &= n - 2^{r-1} + 1/2 \\ &= n - d^\perp + 1/2.\end{aligned}$$

Así, tenemos que $\delta \geq n - d^\perp + 1/2 \Rightarrow \delta \geq n - d^\perp + 1$, y por el Teorema 3.2.5 $[\mathcal{S}]$ tiene solución. Por lo tanto, el código $\mathcal{D}$ que resulta de eliminar las posiciones en L de $\mathrm{Ham}(r, 2)$, es un código $[\delta, \delta - r]$, y resuelve $[\mathcal{S}]$ para m.

Además,

$$\begin{aligned}\delta = n - l &\geq n - n/2 \\ &= n/2 \\ &= (2^r - 1)/2 \\ &= 2^{r-1} - 1/2.\end{aligned}$$

De lo que concluimos que $\delta \geq 2^{r-1}$, y por las Proposiciones 3.3.2 y 3.3.3, tenemos que $\rho(\mathcal{D}) \leq 2$, y un promedio de incrustación:

$$\begin{aligned}\widetilde{\rho}(\mathcal{D}) &= 2 - \frac{\delta + 2}{2^r} \\ &= \frac{2^{r+1} - \delta - 2}{2^r} \\ &= \frac{2(2^r - 1) - (n - l)}{2^r - 1 + 1} \\ &= \frac{2n - n + l}{n + 1} \\ &= \frac{n + l}{n + 1}.\end{aligned}$$

Haciendo un procedimiento parecido al descrito en la demostración de la Proposición 3.3.3, llegamos a que:

$$\begin{aligned}\widetilde{\rho}(\mathrm{Ham}(r, 2)) &= \frac{2^r - 1}{2^r} \\ &= \frac{n}{n + 1}.\end{aligned}$$

Finalmente tenemos:

$$\begin{aligned}\widetilde{\rho}(\mathcal{D}) &= \frac{n + l}{n + 1} \\ &= \frac{n}{n + 1} + \frac{l}{n + 1} \\ &= \widetilde{\rho}(\mathrm{Ham}(r, 2)) + \frac{l}{n + 1}.\end{aligned}$$

□

Lo que nos dice el corolario anterior, es que si las posiciones de bloqueo son mayores a $n/2$, no tenemos garantía de una solución para $[\mathcal{S}]$.

Hasta este punto, el método de incrustación de los esquemas $\mathcal{S}$, está en función del método de decodificación del código $\mathcal{C}$ a usar. Aunque es una buena idea, computacionalmente no todos los códigos cumplen tener un método de decodificación completo y eficiente (en el sentido operativo). Entonces, la idea es crear un método de decodificación que no use la distancia mínima (o sea que no devuelva la palabra más cercana). Pero que la tasa de cambio (incrustación) del método por distancia mínima ($\widetilde{\rho}(\mathcal{C})$) sea lo bastante parecido al nuevo método ($\widetilde{\rho}(\mathrm{dec})$). Lo anterior significa obtener una palabra que no esté tan lejos a la palabra que el método tradicional devolvería. Usando estos parámetros sabremos qué tan preciso será el método a desarrollar.

Definición 3.3.5. *Sea $\mathcal{C}$ un código lineal binario $[n, n - r]$, con H matriz de control de paridad. Consideremos $\{h_1, h_2 \ldots, h_n\}$, donde $h_i \in \mathbb{F}_2^r, \forall i = 1, 2, \ldots, n$ son las columnas de H. Llamaremos*

*a H matriz **sistemática**, cuando exista $\{i_1, \ldots, i_r\}$ tal que $[h_{i_1}, h_{i_2}, \ldots, h_{i_n}]$ es la matriz identidad I_r.*

Definición 3.3.6. *Sea $\mathcal{C}$ un código lineal binario $[n, n-r]$, con H matriz de control de paridad sistemática. Llamaremos a los síndromes de H como **síndromes estándar**. En este caso lo denotaremos por s cuando sea estándar.*

Definición 3.3.7. *Sea $S \subseteq \{1, 2, \ldots, n\}$. Dado $x \in \mathbb{F}_2^n$, tal que $\text{supp}(x) = S$. Entonces, escribiremos simplemente e_S en vez de x ($e_S = x$).*

Observación: Las palabras canónicas de $\mathbb{F}_2^n$, cumplen: $e_{\{i\}} = e_i$. Así, si S solo tiene un número, se colocará ese número como subíndice.

Recordemos que, dado $x \in \mathbb{F}_2^n$, la forma de decodificarlo mediante $\mathcal{C}$, es calcular su síndrome $r(x) = Hx^t$. Lo anterior significa encontrar un conjunto de columnas $\{h_i : i \in S \subseteq \{1, 2, \ldots, n\}\}$, tal que $Hx^t = \sum_{i \in S} h_i$. Así, determinando S, podemos decodificar x por $\text{dec}(x) = x + e_S \in \mathcal{C}$. Cuando el conjunto S es lo más pequeño posible, entonces dec verifica la propiedad de la distancia mínima[5].

Lema 3.3.8. *Sea $\mathcal{C}$ un código lineal binario $[n, n-r]$, con H matriz de control de paridad sistemática. Dado $x \in \mathbb{F}_2^n$, si $s(x) \neq 0$, entonces existe i, tal que $\text{wt}(s(x + e_i)) < \text{wt}(s(x))$.*

Demostración. Como $s(x) \neq 0$, entonces $\text{supp}(s(x)) \neq \emptyset$. Así, al ser H estándar, entonces existe i, tal que h_i es una columna con $\text{supp}(h_i) = \{i\} \subset \text{supp}(s(x))$. Entonces:

$$\begin{aligned} \text{supp}(s(x + e_i)) &= \text{supp}(H(x + e_i)^t) \\ &= \text{supp}(Hx^t + He_i^t) \\ &= \text{supp}(s(x) + h_i) \\ &= \text{supp}(s(x)) - \{i\} \qquad \text{por } \mathbb{F}_2, \\ &\subset \text{supp}(s(x)). \end{aligned}$$

Lo anterior es equivalente a decir $\text{wt}(s(x + e_i)) < \text{wt}(s(x))$. □

Proposición 3.3.9. *Sea $\mathcal{C}$ un código lineal binario $[n, n-r]$, con H matriz de control de paridad sistemática. Dado $x \in \mathbb{F}_2^n$, se verifica que $d(x, \mathcal{C}) \leq \text{wt}(s(x))$.*

Demostración. Si $s(x) = 0$, se verifica de manera inmediata $d(x, \mathcal{C}) = 0 = \text{wt}(s(x))$, ya que $s(x) = 0$, significa que $x \in \mathcal{C}$. Si $s(x) \neq 0$, entonces podemos aplicar el Lema 3.3.8 un máximo de $\text{wt}(s(x))$ veces. Sea $A \subseteq \text{supp}(s(x))$, entonces por el Lemma 3.3.8, tendremos:

$$\text{supp}(s(x + e_A)) = 0$$

[5]Con propiedad de distancia mínima, nos referimos a encontrar la palabra del código, más cercana a x.

Lo anterior es equivalente a decir $s(x+e_A)=0 \Rightarrow H(x+e_A)^t=0 \Rightarrow x+e_A \in \mathcal{C}$. Así, sea $c=x+e_A$, entonces tendremos:

$$\begin{aligned} d(x,\mathcal{C}) &\leq d(x,c) \\ &= \mathrm{wt}(x+x+e_A) \\ &= \mathrm{wt}(e_A) \\ &\leq \mathrm{wt}(s(x)). \end{aligned}$$

□

El resultado anterior nos dice que los síndromes estándar dan una estimación de la distancia de x al código. Además, que por esta forma de calcular e_S, no necesariamente la palabra del código encontrada es la más cercana a x. Veamos con el resultado anterior qué tan buena es la estimación, para nuestros propósitos.

Proposición 3.3.10. *Si $\mathcal{C}$ un código lineal binario $[n, n-r]$, con H matriz de control de paridad sistemática. Entonces el promedio de incrustación verifica:*

$$\frac{2^{r+1}-n-2}{2^r} \leq \widetilde{\rho} \leq \frac{r}{2}.$$

Demostración. Razonando como en la Proposición 3.3.3. Dada una clase C con síndrome igual a una columna de H. Entonces, todas las palabras de C están a una distancia de una unidad de $\mathcal{C}$, lo que significa n palabras a esa distancia. Por otro lado, si el síndrome de la clase C no es una columna de H, entonces las palabras estarán a una distancia por lo menos de dos unidades de $\mathcal{C}$, esto para las 2^r-n-1 palabras restantes y diferentes de cero. Así, tenemos:

$$\begin{aligned} \widetilde{\rho} &\geq \frac{2(2^r-n-1)+n}{2^r} \\ &= \frac{2^{r+1}-2n-2+n}{2^r} \\ &= \frac{2^{r+1}-n-2}{2^r}. \end{aligned}$$

Por otro lado, la Proposición 3.3.9 nos dice que lo máximo que podría ser $d(x,\mathcal{C})$ es el peso del síndrome de x. Así, solo debemos calcular los pesos de todas las palabras de $\mathbb{F}_2^r$, lo cual es:

$$\binom{r}{1}+2\binom{r}{2}+\ldots+r\binom{r}{r}=r2^{r-1}.$$

La igualdad anterior se verifica a partir de $\binom{n}{0}+x\binom{n}{1}+x^2\binom{n}{2}+\ldots+x^n\binom{n}{n}=(1+x)^n$. Con la información anterior tenemos:

$$\begin{aligned} \widetilde{\rho} &\leq \frac{r2^{r-1}}{2^r} \\ &= \frac{r}{2}. \end{aligned}$$

□

Los resultados anteriores, nos muestran una nueva forma de decodificar, usando síndromes sistemáticos.

Algoritmo 3.3.11 (Col). *Si $\mathcal{C}$ un código lineal binario $[n, n-r]$, con H matriz de control de paridad sistemática. Entonces el siguiente pseudocódigo es un método de decodificación por síndrome.*

```
Entrada: s %Palabra de longitud r.
Salida: S %Conjunto indices de columnas de H.
    S={}
    while s~= 0:
        Encontrar h_i, tal que wt(s+h_i) sea el más
        pequeño entre las demás columnas de H;
        S=S+{i};
    end
```

Donde tendremos $\mathrm{dec}(x) = x + e_S \in \mathcal{C}$, *y* $S = \mathrm{Col}(s(x))$.

El método descrito anteriormente, por la demostración hecha en la Proposición 3.3.9 es completa, ya que siempre podremos calcular una palabra del código para un x arbitrario.

Además, notemos que S no es siempre el conjunto más pequeño, tal que s sea suma de columnas de H (por la condición que $\mathrm{wt}(s + h_i)$ sea lo más pequeño posible, lo que probablemente agregue más columnas de las necesarias). También, el algoritmo hace que en una sola iteración se tenga $\mathrm{wt}(s + h_i) < r/2$, si la matriz H tiene una columna de todos 1. Así, solo necesitaremos un máximo de $(r+1)/2$ iteraciones, lo que significa que $\widetilde{\rho} \leq (r+1)/2$ (porque cada iteración cuenta como un cambio).

Proposición 3.3.12. *Si $\mathcal{C}$ un código lineal binario $[n, n-r]$, con H matriz de control de paridad sistemática. El método de decodificación que proporciona 3.3.11, verifica para cada $x \in \mathbb{F}_2^n$:*

1) $d(x, \mathrm{dec}(x)) \leq \mathrm{wt}(s(x))$.

2) Si $\mathrm{wt}(s(x)) \leq d/2$, *entonces* $\mathrm{dec}(x)$ *es la palabra del código más cercana a* x, *donde* d *es la distancia del código.*

3) $\rho(\mathrm{dec}) \leq n - k$.

4) $\widetilde{\rho}(\mathrm{dec}) \leq (n-k)/2$.

Demostración. Sea $x \in \mathbb{F}_2^n$ arbitrario.

1. Sea $S = \mathrm{Col}(s(x))$, entonces $\mathrm{dec}(x) = x + e_S$. Además, por definición $S \subseteq \mathrm{supp}(s(x))$. Así:

$$\begin{aligned} d(x, \mathrm{dec}(x)) &= \mathrm{wt}(x + x + e_S) \\ &= \mathrm{wt}(e_S) \\ &\leq \mathrm{wt}(s(x)). \end{aligned}$$

2. Asumamos por contradicción que $\text{dec}(x)$ no es la palabra más cercana a x. Entonces existe $c \in \mathcal{C}$, tal que es la palabra más cercana a x, así $d(x,c) < d(x,\text{dec}(x))$. Además, por 1) tenemos:

$$d(x,\text{dec}(x)) \leq \text{wt}(s(x)) \leq d/2.$$

Entonces:

$$\begin{aligned} d(c,\text{dec}(x)) &\leq d(c,x) + d(x,\text{dec}(x)) \\ &< d(x,\text{dec}(x)) + d(x,\text{dec}(x)) \\ &= 2d(x,\text{dec}(x)) \\ &\leq 2(d/2) = d. \end{aligned}$$

Lo que significa que $d(c,\text{dec}(x)) < d$, lo cual es una contradicción a la distancia mínima del código. Por lo tanto, $\text{dec}(x)$ es la palabra más cercana a x.

3. Por 1), tenemos que los cambios están acotados por el peso del síndrome. Así, por definición de ρ.

$$\begin{aligned} \rho(\text{dec}) &\leq r \\ &= n - k. \end{aligned}$$

Ya que la dimensión del código es $k = n - r$.

4. Por 1) tenemos simplemente calcular el promedio de los pesos de todas las palabras de $\mathbb{F}_2^r$ (el cual calculamos en la demostración de la Proposición 3.3.10). Entonces:

$$\begin{aligned} \widetilde{\rho}(\text{dec}) &\leq \frac{r2^{r-1}}{2^r} \\ &= \frac{r}{2} = \frac{n-k}{2}. \end{aligned}$$

□

Notemos que la Proposición 3.3.10 y el numeral 4 del resultado anterior, nos garantiza lo que buscábamos; que el método por distancia mínima y el nuevo método de decodificación se parezcan bastante en sus promedios de incrustación.

Ahora, apliquemos todo lo desarrollado, a los códigos de Hamming y veamos cómo resolver $[\mathcal{S}]$. Recordemos que en la sección anterior vimos que resolver $[\mathcal{S}]$ es equivalente a resolver $Hu^t = m + Hx^t = z \in \mathbb{F}_2^r$, para $\text{supp}(u) \subseteq D$. Que también es equivalente a resolver $My^t = z$, donde M es la matriz obtenida de H eliminando las columnas con índices en L, y $y = \pi_L(u)$.

Entonces, para resolver $My^t = z$, consideramos la matriz $\mathbf{D}'$ que es sistemática, aplicando operaciones por renglón, podemos transformar M a $\mathbf{D}'$. Así, siempre que $[\mathcal{S}]$ tenga solución, $\mathbf{D}'$ será una matriz de rango máximo. De hecho, será la matriz de control de paridad de un código acortado $[\delta, \delta - r]$, por la Proposición 3.3.2. Con la matriz $\mathbf{D}'$ resolvemos

$\mathbf{D}'y^t = z$, y aplicando el Algoritmo 3.3.11 obtendremos $S \subseteq D$. Lo que finalmente nos llevará a los métodos de incrustación y recuperación para $\mathcal{S}$, como se describe a continuación.

$$emb(x, m) = x + e_S$$
$$rec(v) = Hv^t.$$

Este esquema tiene capacidad de incrustación r y una tasa de incrustación $\rho(\text{dec})/n$ y promedio de incrustación $\widetilde{\rho}(\text{dec})/n$.

Hemos logrado resolver el problema $[\mathcal{S}]$ para el caso binario y definido un método de decodificación completo, que no dependa de la propiedad de la mínima distancia. Además, de especificar cómo crear un esquema $\mathcal{S}$ a partir del código $\text{Ham}(r, 2)$, ya sea con selección compartida o no.

Hasta este punto hemos trabajado entorno a los bits menos significativos, en otras palabras, hemos considerado el mensaje como ruido dentro de la cubierta. Entonces, lo que trataremos en este capítulo es considerar el mensaje como un cambio en la intensidad del color de un píxel, que guardará el mensaje, en este sentido, las reglas para elegir el píxel siguen siendo las mismas, pues siempre se elegirá un píxel que no levante sospechas al momento que se le alteré su intensidad (por tal motivo, recibe el nombre de esteganografía más menos).

3.4. Caso ternario (± 1)

En esta sección estudiaremos el caso de cambio de un grado de intensidad, en otras palabras veremos cómo aumentar o disminuir la intensidad de un píxel, en función del mensaje a ocultar.

Ahora, estaremos considerando $\mathbb{F}_q$, con $q = 3$, como alfabeto. Además, recordemos que por la definición 1.2.7, los píxeles son codificados en el conjunto $\{0, 1, 2, \ldots, 2^D - 1\}$. Así, agrupamos el clases los píxeles módulo 3, y tendremos la siguiente definición.

Definición 3.4.1. *Sea $x \in \{0, 1, 2, \ldots, 2^D - 1\}$, para un $D \in \mathbb{Z}^+$. Diremos que x **está en la clase de** w, si $x \equiv w \mod (3)$, para $w \in \{-1, 0, 1\}$.*

Además, consideremos una distancia para el conjunto $\{0, 1, 2, \ldots, 2^D - 1\}$.

Definición 3.4.2. *Sean $x, y \in \{0, 1, 2, \ldots, 2^D - 1\}$, para un $D \in \mathbb{Z}^+$. La distancia D entre x e y, se define como $D(x, y) = (x - y)^2$.*

Observación: D es el cuadrado de la distancia usual en $\mathbb{R}$.

Nota: D también es llamado **distorsión del error al cuadrado**.

Con las dos definiciones anteriores, podemos definir un método para incrustar un mensaje, dado los píxeles.

Definición 3.4.3 (Esquema ternario no codificado). *Sea el mensaje* $m \in \{-1, 0, 1\}$ *y* $x \in \{1, 2, \ldots, 2^D - 2\}$*, para un* $D \in \mathbb{Z}^+$*. Entonces,*

$$emb(m, x) = \begin{cases} x \quad si \qquad\qquad x \equiv m \mod (3) \\ \\ y \quad si \quad D(x, y) = \min\{D(x, z) : z \equiv m \mod (3)\} \end{cases}$$

$$rec(y) = \begin{cases} -1 \quad si \quad y \equiv -1 \mod (3) \\ \\ 0 \quad si \quad y \equiv 0 \mod (3) \\ \\ 1 \quad si \quad y \equiv 1 \mod (3). \end{cases}$$

Observación: Se dice que es *no codificado*, porque los píxeles son tomados como tal. Además, se dejan a fuera los píxeles 0 y $2^D - 1$, porque al cambiarlos puede resultar muy evidente[6] el cambio en la imagen.

Nota: En la definición de *emb* se elige el más cercano a x, de la clase de w, para que el cambio no sea brusco.

Al método de la definición 3.4.3, podemos examinar su carga útil, y saber qué tan buen método es. Usando la definición de E, dada en el capítulo 1, tenemos:

$$\begin{aligned} E &= \frac{1}{1}\log_2(3) \\ &= \log_2(3) \approx 1.58 \end{aligned}$$

Lo anterior nos dice que por cada símbolo de la cubierta, somos capaces de ingresar 3 símbolos del mensaje, lo que es bastante bueno, ya que podemos ocultar una buena cantidad de información.

Ejemplo 3.4.4. *Veamos a continuación cómo actúa 3.4.3, sobre algunos caso particulares.*

x \ m	0	1	2
9	9	10	8
10	9	10	11
11	12	10	11

Observación: *Los píxeles cambiados están al interior de la tabla. Además, todos los valores de* x*, siempre quedan dentro de la clase del mensaje* m*.*

[6]Si tuviéramos la escala de grises y cambiamos $2^D - 1$, por una profundidad más, resultaría 0, lo que significa que podríamos pasar del color negro al blanco, y es claro que este cambio sera notorio.

En 3.4.3 se ha definido un método de incrustación y recuperación bastante sencillo, con una carga útil aceptable. Esta idea la podemos aplicar a las clases que podemos formar con un código lineal, y utilizar su matriz de control de paridad, para poder crear un método que nos permita incrustar una mayor cantidad de información. Además, al pedir el elemento más cercano de la clase a x, tácitamente estamos creando un método de decodificación por distancia mínima.

De nuevo, los códigos de Hamming son una buena opción, ya que al ser perfectos tenemos que cubren perfectamente el espacio de trabajo, además de poder crear herramientas bastante fáciles de aplicar. Así, por 3.1.3 tenemos los métodos de incrustación y recuperación sobre $\mathbb{F}_3$.

En este caso, dado el código $\mathrm{Ham}(r,3)$, tendremos una carga útil de:

$$\begin{aligned} E &= \frac{r}{(3^r-1)/2}\log_2(3) \\ &= \frac{2r}{3^r-1}\log_2(3). \end{aligned}$$

Y utilizando inducción se puede probar que $(2r)/(3^r-1) < 1$, esto quiere decir que el método definido por el código de Hamming, es mucho mejor que el 3.4.3, ya que requerirá menos bits de la cubierta para ocultar la información. Además, recordando la carga útil del código binario de Hamming que es $r/(2^r-1)$, vemos que el código ternario, tiene mejor carga útil, lo cual lo vuelve una mejor opción a usar.

Hasta este punto, hemos logrado establecer una relación entre los códigos líneales y la esteganografía. Además, verificamos que podemos mejorar los esquemas, incrementando el cuerpo sobre el que trabajamos, o creando nuevas formas de decodificación (que no necesariamente sean eficientes para la corrección) para poder tener esquemas eficientes para la práctica. Por último, sentamos las ideas de cómo sería un esquema que permita modificar intensidades de los píxeles, solo usando el mensaje a ocultar.

Bibliografía

[1] Marten Van Dijk, Frans Willems e Y Nx. «Embedding Information in Grayscale Images». En: *Proc. 22nd Symp. Inform. Theory in the Benelux*, págs. 147-154.

[2] Florence Jessie MacWilliams. *The theory of error correcting codes*. North - Holland, 2006.

[3] Carlos Munuera. «Steganography from a Coding Theory Point of View». En: (feb. de 2013). DOI: 10.1142/9789814335768_0003.

[4] Carlos Munuera. «Hamming Codes for Wet Paper Steganography». En: *Des. Codes Cryptography* 76.1 (jul. de 2015), 101–111. ISSN: 0925-1022. DOI: 10.1007/s10623-014-9998-5. URL: https://doi.org/10.1007/s10623-014-9998-5.

[5] Vera Pless. *Introduction to the Theory of Error-Correcting Codes*. Vol. 48. John Wiley & Sons, 2011.

[6] Steven Roman. *Coding and Information Theory*. Vol. 134. Springer Science & Business Media, 1992.