Algunos Resultados Sobre Funciones Booleanas, Cajas de Sustitución y Z_{2^k}-Códigos Lineales

Tesis Doctoral en Ciencias Matemáticas

que presenta

Gerardo Vega Hernández

Departamento de Matemáticas,
Universidad Autónoma Metropolitana-Iztapalapa
México, D.F., Marzo, 2001

Director de Tesis: Horacio Tapia-Recillas

Se agradece el apoyo de las siguientes Instituciones:

- Dirección General de Servicios de Cómputo Académico de la Universidad Nacional Autónoma de México.

- Departamento de Matemáticas de la Universidad Autónoma Metropolitana, Unidad Iztapalapa.

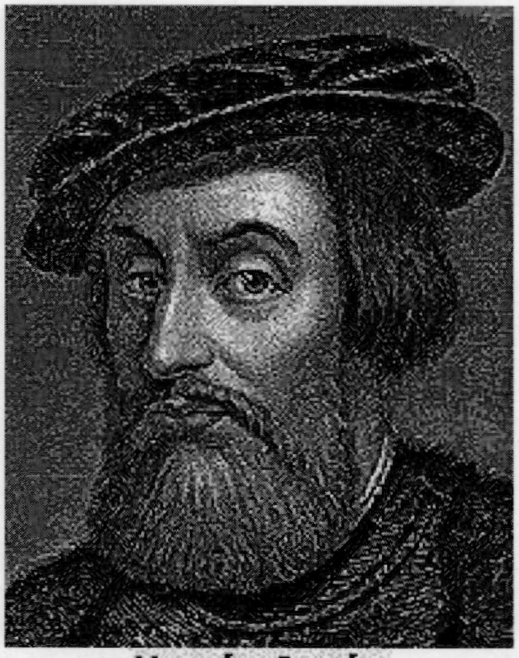
- CONACyT, a través del proyecto L0076E9607, "Teoría Algebraica de Códigos, Álgebra Conmutativa y Geometría Algebraica", bajo la responsabilidad de Dr. Horacio Tapia Recillas (Depto. de Matemáticas, UAM-I).

Quiero agradecer de manera muy especial, los consejos, la orientación y sobre todo la amistad que he recibido de Horacio Tapia Recillas.

Con todo mi cariño dedico este trabajo a
Arte, Sofi y Sari

... la tendencia de los cronistas e historiadores mexicanos a acrecentar el pasado glorioso de una de las dos ramas de su estirpe.

Martín Luis Guzmán,
La Querella de México.



Hernán Cortés

Algunos Resultados Sobre Funciones Booleanas, Cajas de Sustitución y Z_{2^k}-Códigos Lineales

Gerardo Vega Hernández

Universidad Autónoma
Metropolitana-Iztapalapa
México, D.F., Marzo, 2001

Índice General

Introducción

La organización social, a nivel mundial, se encamina hacia lo que se podría denominar una Sociedad Digital, en la cual los "bits" pueden tener mucho más valor que los bienes materiales. Por tal razón, la correcta y segura transmisión y almacenamiento de la información es fundamental no sólo para ciertas funciones de una sociedad moderna sino prácticamente para todas sus actividades.

La Teoría de Códigos y la Criptografía son las técnicas más usadas en la actualidad para brindar esos requerimientos. En este trabajo se presentan algunas investigaciones originales [39, 40, 41] en estas áreas. En el caso de Teoría de Códigos, son dos los resultados principales que se presentan. El primer resultado retoma una transformación entre funciones Booleanas que fuera propuesta en [8]. Esta transformación tiene la peculiaridad de que al ser aplicada a cualquier función Booleana permite asociarle a ésta otra función con la cual mantendrá una sencilla relación entre los pesos de Hamming de ambas funciones. En [8] se prueba que aplicando ahora, de manera iterativa esta transformación, es posible asociarle a cualquier función Booleana otra función con la cual, por un lado, se siga manteniendo una relación entre sus pesos de Hamming, y por el otro, tal función resultante tenga grado menor o igual a 3. De lo anterior, se desprende entonces que es tan difícil caracterizar a cualquier función Booleana de grado 3 como lo es para cualquier otra función de grado mayor o igual a 4. De esta manera, el resultado que se presenta en este trabajo se refiere a un método de factorización sobre funciones Booleanas que permite establecer una cota superior sobre el número de iteraciones necesarias para la transformación de cualquier función Booleana de grado mayor o igual a 4 a una función de grado 3.

Para el caso de criptografía se presentan varios resultados relacionados con ciertos agrupamientos de funciones Booleanas, los cuales dan lugar a un componente criptográfico conocido como *caja de sustitución* o *S-caja*. Tal componente se reconoce [1, 11, 27, 43] como uno de los más importantes en

el diseño de cierto tipo de sistemas de cifrado simétricos. Por tal razón, no es extraño reconocer que los esfuerzos encaminados a vulnerar este tipo de sistemas se han enfocado sobre tales cajas de sustitución. Una característica básica deseable de las cajas de sustitución es que éstas se comporten como mapeos regulares entre dos conjuntos de vectores binarios. En esta investigación se ha realizado el estudio de estos mapeos regulares desde el marco de ciertas tablas, conocidas como *tablas de distribución de diferencias*, a las que estos mapeos dan lugar. Particularmente aquí se presenta una caracterización de los mapeos regulares en términos de sus tablas de distribución de diferencias. Por otro lado, se retoma el concepto de ϵ-robustez introducido originalmente en [34], el cual se propone como una medida de vulnerabilidad de las cajas de sustitución frente a los ataques de criptoanálisis diferencial o lineal. Usando la caracterización antes mencionada y con ayuda otra vez de las tablas de distribución de diferencias se prueba aquí una nueva cota para la ϵ-robustez de cualquier caja de sustitución.

Para el tercer resultado de este trabajo, se exhibe la existencia de un nuevo tipo de códigos sobre anillos de la forma $\mathbb{Z}_{2^{k+1}}$, para $k > 1$. Para tales códigos, los cuales se han dado en llamar *códigos hpo-cíclicos*, se prueba que son la generalización de un tipo de códigos sobre $\mathbb{Z}_4$, conocidos como *códigos negacíclicos*, los que a su vez, fueran introducidos originalmente en [44]. Para tal generalización, se propone aquí una isometría entre códigos sobre $\mathbb{Z}_{2^{k+1}}$ y códigos sobre $\mathbb{Z}_4$, con la cual, se prueba que la imagen de un código *hpo*-cíclico bajo esta isometría es básicamente la concatenación de 2^{k-1} códigos negacíclicos. Por otro lado, se demuestra también que a través de esta isometría es posible definir otra isometría entre códigos sobre $\mathbb{Z}_{2^{k+1}}$ y códigos sobre $\mathbb{Z}_2$, donde esta última isometría resulta ser una generalización del mapeo de Gray equivalente a la propuesta en [9]. Como parte final de esta investigación, se usan estas dos isometrías para estudiar los efectos que tienen éstas sobre códigos lineales *hpo*-cíclicos.

Con el objeto de hacer más clara la exposición de estos resultados, este trabajo se ha dividido en seis Capítulos. En los primeros dos se expondrán los antecedentes de Teoría de Códigos y de Criptografía que son relevantes al material que aquí se presenta. Sin embargo, debido a que es imposible cubrir todos los detalles, se recomienda ampliamente la consulta del material en [2, 5, 7, 15, 19, 22, 23, 33], entre otros. Se hace la aclaración de que estos primeros dos Capítulos son de apoyo y por tanto, bien pueden ser omitidos por un lector conocedor. Por otro lado, los tres Capítulos siguientes serán dedicados precisamente a la exposición de los resultados de las investigaciones antes descritas, mientras que el último Capítulo será dedicado a las Conclusiones.

Capítulo 1

Antecedentes Sobre Teoría de Códigos

1.1 Introducción

La Teoría de Códigos recibe un gran impulso a raíz de un artículo de Claude Shannon, titulado *A mathematical theory of communication* publicado en 1948 [37]. En este trabajo, entre otras cosas, Shannon introduce una cantidad medible denominada "capacidad de un canal", la cual es atribuible a los canales de comunicación. Shannon prueba que sobre un determinado canal, la comunicación libre de error es posible si se emplean esquemas adecuados de codificación y siempre y cuando esta información codificada viaje por debajo de la capacidad de dicho canal. Desde la presentación de este resultado, se han venido realizado diversas investigaciones tendientes a desarrollar estrategias bajo las cuales la información digital puede ser codificada para su transmisión o almacenamiento confiable en medios ruidosos. Tales investigaciones se enmarcan dentro de área de los *Códigos Detectores-Correctores de Errores.*

1.2 Códigos Detectores-Correctores de Errores

Los errores en la transmisión o almacenamiento son inherentes a la naturaleza misma de los sistemas de transmisión o almacenamiento que se emplean. Estos sistemas, según sus características, pueden erróneamente alterar en mayor o menor medida la información que procesan o almacenan. Por ejemplo, supóngase que dos equipos electrónicos están intercambiando información en forma de cadenas de bits a través de un enlace dado por

un cable. El equipo emisor transfiere su información binaria a través del enlace en forma de una señal que varía su voltaje en dos posiciones; una posición indicaría la transmisión de un cero mientras que la otra indicaría un uno. Estas variaciones son percibidas por el receptor, y a partir de éstas, deduce la información binaria que está recibiendo. Sin embargo, debido a la atenuación natural de este tipo de señales, es posible que la información que deduce el receptor no se la misma que el emisor envió y en tales circustancias se dice que se han producido errores en la transmisión (Fig. 1.1).

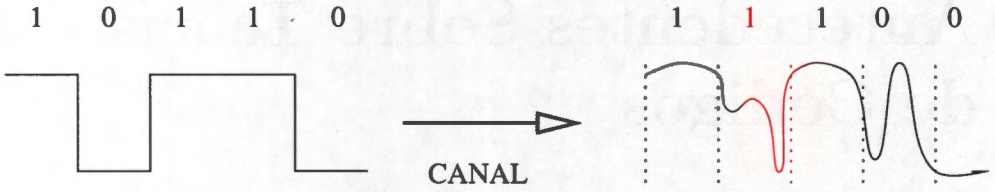


Figura 1.1: Distorsión de la información a través de un canal.

En el ejemplo de la Figura anterior se tiene que el emisor envía la secuencia de bits $x = 10110$ y el receptor recibe la secuencia $y = 11100$, la cual difiere con x en el segundo y cuarto bit. Observe que las secuencias x y y pueden ser vistas como vectores binarios de longitud 5 sobre el campo finito de dos elementos $\mathbb{F}_2$, esto es $x, y \in \mathbb{F}_2^5$. Al campo $\mathbb{F}_2$ se le conoce también, de manera natural, como el campo de los números binarios. En tal campo, la operación de suma se acostumbra denotar por "$\oplus$", mientras que la operación de multiplicación se denota simplemente por la yuxtaposición de sus operandos, tal como se hace en el caso de $\mathbb{R}$. Ver las secuencias x y y como vectores binarios resulta de utilidad pues de esta manera es posible expresar el error e como la diferencia[1] de x y y, es decir, $e = x \oplus y$. Para el ejemplo de la Fig. 1.1 se tiene que $e = 01010$ y observe que los bits en 1 de e indican los bits que se han recibido con error. Así pues, utilizando la notación anterior, es conveniente ahora describir la problemática de la Fig. 1.1 a través del modelo de la Fig. 1.2.

Ahora bien, la pregunta es: ¿Qué hacer para proteger la información de los errores inherentes a un canal de comunicación? La solución típica, consiste en agregar redundancia a la información con la esperanza de que a través de esta redundancia, la entidad receptora sea capaz de deducir con una alta probabilidad cuál es la información real que se ha transmitido. Esto último se comprende aún mejor si se reconoce que la información que normalmente intercambiamos los humanos, lleva consigo cierto nivel de re-

[1] En $\mathbb{F}_2^n$ las operaciones de suma y diferencia coinciden, para cualquier entero $n \geq 1$.

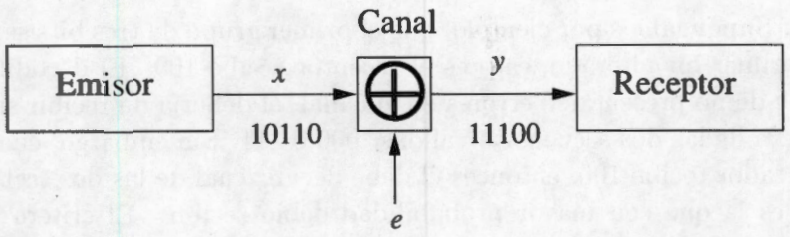


Figura 1.2: Modelo para la distorsión de la información a través de un canal ruidoso.

dundancia, y por tal razón es posible, hasta cierto grado, la reconstrucción de un mensaje que ha sido recibido con errores. De esta manera, es necesario agregar al modelo de la Fig. 1.2 un componente en el emisor que se encargue de agregar la redundancia. Del lado del receptor también será necesario un nuevo componente, el cual tendrá como función la de detectar y/o corregir, a través de la redundancia, las posibles alteraciones de la información que se recibe. El procedimiento más común para agregar redundancia es a través de tablas de codificación, por tal razón, estos dos nuevos componentes, en el emisor y receptor, se denominan *codificador* y *decodificador* respectivamente (Fig. 1.3).

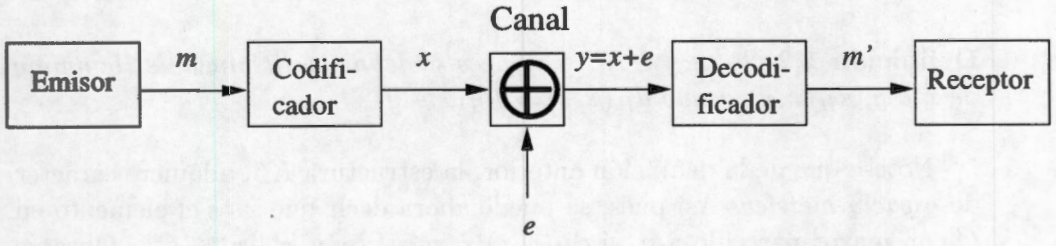


Figura 1.3: Nuevo modelo.

Claramente uno de los temas fundamentales de los códigos detectores-correctores de errores, es el diseño de los codificadores y decodificadores, de tal manera que éstos sean capaces de detectar y/o corregir el mayor número de errores. Por ejemplo, supóngase que se desea diseñar un codificador-decodificador que sea capaz de detectar y corregir un error. Para ello, se propone que por cada dígito binario que reciba el codificador, éste envíe por el canal tres copias de dicho dígito. De esta manera, si desea transmitir el mensaje $m = 01101$, lo que el codificador en realidad enviaría sería $x = 000111111000111$. Ahora bien, el decodificador en el lado receptor, sabe que por cada grupo de tres bits que reciba, debe entregar al receptor final un

solo bit. Supongamos por ejemplo, que el primer grupo de tres bits se recibe con el primer bit alterado, esto es el receptor recibe 100. El decodificador sabe que de no presentarse errores en el canal, él debería de recibir siempre cualquiera de las dos secuencias válidas: 000 ó 111. Sin embargo, cuando el decodificador recibe 100, entonces él debe decidir cuál de las dos secuencias válidas es la que con mayor probabilidad debió recibir. El critero que el decodificador utiliza para tomar esta decisión es la de "mayor parecido", esto es, si y es la secuencia que recibe el decodificador y si se denota por C al conjuto de todas las secuencias válidas que un codificador puede generar, entonces el decodificador decodificará a y como $\hat{x}$ si $\hat{x}$ es el elemento en C con mayor parecido a y. Para nuestro ejemplo, 000 es la secuencia válida que más se parece a 100 y por tanto el bit 0 será el que producirá el decodificador a su salida para este caso. Ahora bien, la pregunta que está en el aire es: ¿De qué manera se cuantifica el mayor parecido entre secuencias de bits de igual longitud? La respuesta a esta pregunta está en términos de las siguientes definiciones:

Definición 1.1 *Sea K un campo o anillo y para un entero positivo n sea K^n el conjunto de vectores de longitud n con entradas en K. Entonces para todo $x \in K^n$, se define el peso de Hamming de x, $P_H(x)$, como el número de entradas diferentes de cero en el vector x.*

Definición 1.2 *Sea $x, y \in K^n$, entonces se define la distancia de Hamming de x a y, $d_H(x,y)$, como $d_H(x,y) = P_H(x-y)$.*

Nótese que de la definición anterior, la estructura K^n, adquiere carácter de *espacio métrico*. Así pues, se puede ahora decir que $\hat{x}$ es el elemento en C con mayor parecido a y, si $d_H(y,\hat{x}) = min\{d_H(y,x^*)|x^* \in C\}$. Observe que $\hat{x}$ ¡no necesariamente es único!

Debe ser claro que un codificador-decodificador con las características del ejemplo anterior será capaz, entonces, de detectar y aún corregir cualquier error simple que se presente en todo grupo de tres bits que el decodificador reciba. Sin embargo, observe que si son dos o más errores los que se presentan, entonces probablemente el decodificador fallará.

Hasta ahora, se ha dicho que los codificadores y decodificadores reciben y entregan secuencias de bits. Sin embargo, una mejor aproximación del tipo de información que estos dispositivos manipulan, se describe en la Fig. 1.4. En esta Figura, se esquematiza que los mensajes m que un codificador recibe, están formados por k símbolos, a saber $m = (m_1, m_2, ..., m_k)$, donde los símbolos m_i se toman sobre alguna estructura algebraica K, la cual

típicamente es un campo o anillo finito. El codificador procesa el vector m y genera como salida el vector $x = (x_1, x_2, ..., x_n)$ en donde nuevamente los x_i se toman sobre alguna estructura algebraica, la cual comúnmente coincide con K. Normalmente el número de símbolos en x es mayor al número de símbolos en m, esto es: $k \leq n$. Al vector x, se le denomina comúnmente como *palabra codificada o de código* y al conjunto, C, de palabras de código se denomina *código*. Observe que C es en principio simplemente un subconjunto en K^n.

La palabra codificada x es transmitida por el canal, donde puede sufrir alteraciones en términos del vector $e = (e_1, e_2, ..., e_n) \in K^n$ según la relación $y = x + e$, en donde la operación $+$ denota la suma vectorial en K^n. El decodificador recibe el vector $y = x + e$, el cual deberá procesar para encontrar la palabra de código $\hat{x}$ que minimize la distancia de Hamming con y. Una vez elegido tal vector $\hat{x}$ el decodificador entrega como estimación del mensaje a aquel vector $m' \in K^k$ cuya codificación corresponde con $\hat{x}$ (Fig. 1.4).

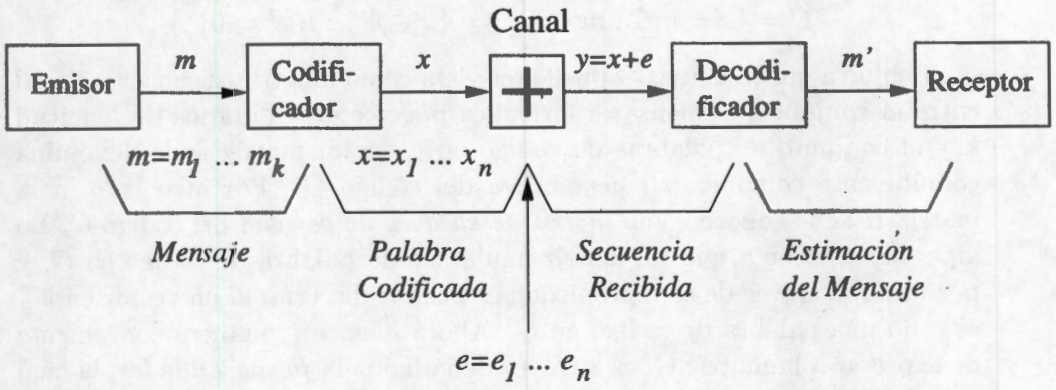


Figura 1.4: Una mejor aproximación del modelo anterior.

Aunque en principo, como se sugiere en la Figura anterior, es posible el diseño de sistemas codificación y decodificación que manipulen símbolos no binarios, en este Capítulo, por simplicidad, nos restringiremos sólo a sistemas que manipulan la información en binario. No obstante, se hace la aclaración que toda la teoría expuesta en este Capítulo es generalizable sobre cualquier campo finito.

Para el diseño de los sistemas de detección y corrección de errores, es indispensable definir al conjunto C, de palabras de código. Existen diversos tipos de códigos, sin embargo, los códigos lineales ofrecen varias ventajas sobre otro tipo de códigos, como son: la facilidad de descripción y la facilidad de codificación y decodificación.

1.3 Códigos Lineales

La noción de código detector-corrector de errores fue introducida por R.W. Hamming en 1950, y aparece descrita por primera vez en [14]. Por otro lado, el concepto de código lineal fue propuesta por primera vez en 1956, por D. Slepian [38]. Para mayores detalles sobre nociones y resultados básicos sobre códigos lineales se puede consultar, por ejemplo [7, 19, 23].

Definición 1.3 *Sean n y k enteros positivos con $k \leq n$. Un código lineal binario de longitud n y dimensión k, denotado por $C[n,k]$ o simplemente C, es un subespacio vectorial de $\mathbb{F}_2^n$.*

Como C es un subespacio de $\mathbb{F}_2^n$, entonces es posible asociarle al código C dos matrices binarias, G y H, de tamaños $k \times n$ y $n-k \times n$, respectivamente, tales que:

$$C = \{x = mG : m \in \mathbb{F}_2^k\} = \{x \in \mathbb{F}_2^n : Hx^t = 0\} .$$

Debido a que la matriz G puede ser vista como una transformación lineal entre el conjunto de mensajes formados por vectores binarios de longitud k y el conjunto de palabras de código en C, a tal matriz se le denomina comúnmente como *matriz generadora* del código C. Por otro lado, a la matriz H se le conoce como *matriz de chequeo de paridad* del código C. Lo anterior, se debe a que tal matriz anula a toda palabra de código en C, y por tanto, a través de esta propiedad es posible discernir si un vector en $\mathbb{F}_2^n$ es o no una palabra de código en C. Ahora bien, una manera conveniente de expresar a la matriz G, es a través de la llamada *forma estándar*, la cual está dada por:

$$G = [I_k|A] ,$$

donde I_k es la matriz identidad de orden k y A es una matriz binaria de tamaño $k \times (n-k)$. Observe que si la palabra de código $x = (x_1, x_2, ..., x_n) \in C$ es tal que $x = mG$ para alguna $m = (m_1, m_2, ..., m_k) \in \mathbb{F}_2^k$, entonces $x_i = m_i \; \forall \, i = 1, 2, ..., k$. Por esta razón, cuando G está en su forma estándar, a los bits x_i, $i = 1, 2, ..., k$ de cualquier palabra de código x, se les llama *bits de información*, mientras que a los bits restantes x_i, $i = k+1, ..., n$, se les llama *bits de redundancia* o *de paridad*. Si G está expresada en su forma estándar, entonces la matriz H puede ser llevada a la forma:

$$H = [A^t|I_{n-k}] ,$$

donde A^t denota la matriz transpuesta de A.

Como se expuso en la sección anterior, la estrategia de decodificación será que cada vez que se reciba por el canal un vector y, el decodificador buscará aquella palabra de código $\hat{x}$ que tenga un mayor parecido -o más precisamente que esté a una menor distancia de Hamming- con y. Lo anterior implica que una característica deseable de un código es que sus palabras de código estén lo más separadas entre sí. De esta manera, un parámetro importante en todo código C, sea éste lineal o no, es:

Definición 1.4 *La distancia mínima de un código C, donotado por d, es:*

$$\begin{aligned} d &= min\{d_H(u,v)\} \\ &\quad min\{P_H(u-v)\} \quad \forall\, u,v \in C, u \neq v\,. \end{aligned} \tag{1.1}$$

Ahora bien, si C es un código lineal entonces C es un subespacio vectorial de $\mathbb{F}_2^n$, y por tanto si $u, v \in C$ entonces $u - v \in C$ también. Por lo anterior se tiene que si C es un código lineal, entonces su distancia mínima d puede ser calculada simplemente como $d = min\{P_H(u) : 0 \neq u \in C\}$.

Con la introducción de la distancia mínima de un código podemos ahora enfatizar que los tres parámetros más importantes de todo código lineal son: su longitud n, su dimensión k y su distancia mínima d. Debido a la importancia de estos parámetros, es común denotar a un código lineal como $C[n,k,d]$.

Para terminar la exposición en esta sección, debe notarse que existe una relación entre la distancia mínima de un código lineal o no lineal y el número de errores que tal código es capaz de detectar y corregir. Lo anterior quedó asentado en el siguiente resultado (cf. [23]).

Teorema 1.5 *Un código C con distancia mínima d puede corregir hasta $\lfloor (d-1)/2 \rfloor$[1]. Si d es par, entonces el código C puede simultáneamente corregir $\lfloor (d-1)/2 \rfloor$ errores y detectar $d/2$ errores.*

1.4 Códigos Cíclicos

Los códigos cíclicos forman una importante familia de códigos por diversas razones. Desde un punto de vista teórico, tales códigos poseen una estructura algebraica muy rica, mientras que desde un punto de vista práctico éstos pueden ser implementados de una manera muy eficiente por medio de

[1] $\lfloor x \rfloor$ denota el entero más grande $\leq x$.

dispositivos electrónicos muy simples conocidos como registros de corrimiento (*shift registers*). Más aún, muchos de los códigos importantes como son los códigos binarios de Hamming , los códigos de Golay y los códigos BCH, pueden expresarse en términos de códigos que son cíclicos y lineales.

Definición 1.6 *Un subconjunto C de $\mathbb{F}_2^n$, es un código binario cíclico de longitud n, si tal subconjunto es invariante ante corrimientos circulares, esto es si:*

$$\forall\ (c_0, c_1, ..., c_{n-1}) \in C \text{ entonces } (c_{n-1}, c_0, c_1, ..., c_{n-2}) \in C\ .$$

Una descripción algebraica de los códigos cíclicos se puede lograr si se reconoce que cada vector binario de longitud n en $\mathbb{F}_2^n$ puede ser asociado con un sólo polinomio en el anillo cociente $R_n = \mathbb{F}_2[x]/(x^n - 1)$, esto es:

$$\forall\ (c_0, c_1, ..., c_{n-1}) \in \mathbb{F}_2^n \longleftrightarrow c_0 + c_1 x + ... + c_{n-1}x^{n-1} \in R_n\ .$$

A tal asociación se le conoce como representación polinomial y es a través de ésta que podemos ver a las palabras de código en C como un subconjunto de polinomios en R_n. Bajo esta óptica podemos entonces redefinir un código cíclico de longitud n, como un subconjunto en R_n tal que éste sea cerrado bajo la multiplicación por x, esto es si:

$$\forall\ c(x) \in C \text{ entonces } xc(x) \in C\ .$$

Observe que los códigos lineales en R_n serán subconjuntos cerrados bajo la suma de R_n. Por otro lado, se dice que un subconjunto en el anillo cociente R_n es un *ideal* (ver por ejemplo [22]) si tal subconjunto es cerrado bajo la suma y producto por x. Por tal razón, es común encontrar en la bibliografía sobre códigos cíclicos la siguiente:

Definición 1.7 *Un código binario de longitud n lineal y cíclico, es un ideal en R_n.*

Considerando la definición anterior, es entonces posible usar buena parte de la teoría de anillos para estudiar a los códigos binarios que son lineales y cíclicos. Más aún, existe una definición equivalente para el caso de códigos sobre cualquier campo finito (ver por ejemplo [23]), he incluso se han hecho investigaciones recientes [28, 20] que han extendido el concepto a códigos lineales y cíclicos sobre anillos de la forma $\mathbb{Z}_{2^k}$. En el Capítulo cinco de este trabajo se estudiarán un tipo de ideales en $\mathbb{Z}_{2^k}$, los cuales son la generalización de los códigos lineales negacíclicos que fueran introducidos originalmente en [44].

1.5 Funciones Booleanas

El conjunto de funciones Booleanas son de importancia debido a que, entre otras cosas, es a través de este tipo de funciones que es posible definir una familia de códigos lineales muy importantes conocidos como *códigos de Reed-Muller*. Este tipo de códigos fueron de los primeros en ser estudiados ampliamente y también fueron de los primeros códigos que se emplearon en diversas aplicaciones prácticas, por ejemplo, la sonda espacial Mariner 9 utilizó códigos de Reed-Muller para codificar las imágenes que se enviaban a la Tierra.

Definición 1.8 *Una función Booleana en m variables es una función de $\mathbb{F}_2^m$ a $\mathbb{F}_2$, para cualquier entero $m > 0$.*

Cada uno de los 2^m vectores en $\mathbb{F}_2^m$ puede ser identificado de manera natural con algún entero i en el conjunto $\{0, 1, ..., 2^m - 1\}$. Usando tal identificación se define la *tabla de verdad* de una función Booleana f, como el vector en $\mathbb{F}_2^{2^m}$ dado por $(f(\bar{\alpha}_0), f(\bar{\alpha}_1), ..., f(\bar{\alpha}_{2^m-1}))$, donde $\bar{\alpha}_i \in \mathbb{F}_2^m$ es la representación binaria a m bits del entero $i \in \{0, 1, ..., 2^m - 1\}$. De lo anterior se puede ahora considerar a todo vector binario de longitud 2^m como una función Booleana en m variables. A través de esta visión de funciones Booleanas por tablas de verdad, se define la suma de dos funciones Booleanas como la función Booleana cuya tabla de verdad resulta de la suma, en $\mathbb{F}_2^{2^m}$, de las dos tablas de verdad de las funciones sumadas. De manera equivalente se establece el producto de funciones Booleanas. Observe como a través de estas dos operaciones, $\mathbb{F}_2^{2^m}$ adquiere ahora estructura de anillo conmutativo con unidad. Por otro lado, definimos *el peso de Hamming de una función Booleana f*, $P_H(f)$, como el número de unos presentes en su tabla de verdad.

Una representación algebraica para las funciones Booleanas se establece si éstas son vistas como polinomios en m variables $(x_1, x_2, ..., x_m)$ sobre $\mathbb{F}_2$, esto es, como polinomios en el anillo $\mathcal{B}_m = GF(2)[X_1, ..., X_m]/(X_1^2 - X_1, ..., X_m^2 - X_m)$. Ahora bien, observe que en este anillo es posible que dos polinomios con expresiones algebraicas diferentes puedan tener, no obtante, la misma tabla de verdad. Por tanto, diremos que dos polinomios en $\mathcal{B}_m$ son iguales si sus tablas de verdad lo son. Se puede demostrar [2] que empleando la definición de suma y producto de funciones Booleanas, antes establecida, y a través de la suma y producto usual entre elementos del anillo $\mathcal{B}_m$, es posible establecer un isomorfismo de anillos entre el conjunto de polinomios en $\mathcal{B}_m$ y el conjunto de palabras binarias de longitud 2^m.

Una base natural para el anillo de polinomios $\mathcal{B}_m$ está dada a través del conjunto de monomios $\mathcal{M} = \{\mu^\alpha \doteq \prod_{i \in \alpha} x_i | \alpha \subseteq S\}$, donde $S = \{1, 2, ..., m\}$.

Esto es, $\mathcal{B}_m = \{f | f = \sum_{\alpha \in I} \mu^\alpha; I \subseteq 2^S\}$. De lo anterior, se observa que cada subconjunto I de 2^S, define unívocamente una función Booleana f en $\mathcal{B}_m$. Usando esta última relación se define de manera directa el concepto de *grado* de una función Booleana $f = \sum_{\alpha \in I} \mu^\alpha$, como $gr(f) = max\{|\alpha| \ : \ \alpha \in I\}$. Una función Booleana que se expresa en términos del conjunto de monomios $\mathcal{M}$, se dice que esta en su *forma normal.*

1.5.1 Códigos de Reed-Muller

Los códigos de Reed-Muller fueron presentados inicialmente por I.S. Reed [29] y por D.E. Muller [25]. Este tipo de códigos pertenecen a la familia de los códigos lineales los cuales se caracterizan por su sencillez tanto en su descripción como en el proceso de codificación y decodificación [23].

Definición 1.9 *Por código binario de Reed-Muller de longitud 2^m y orden r $(r = 0, 1, ..., m)$ se entenderá al código lineal binario, $RM(r, m)$, de todas las palabras binarias de longitud 2^m, las cuales se pueden obtener como las tablas de verdad de polinomios en $\mathcal{B}_m$ de grado a lo más r.*

De la definición anterior se puede observar que los códigos de Reed-Muller son tales que $RM(0, m) \subset RM(1, m) \subset ... \subset RM(m, m) = \mathcal{B}_m \cong \mathbb{F}_2^{2^m}$. Más aún, dado que $f = \sum_{\alpha \in I} \mu^\alpha \in RM(r, m)$ si y sólo si para toda $\alpha \in I$ se tiene que $|\alpha| \leq r$, entonces es claro que:

$$dimensión(RM(r, m)) = \binom{m}{0} + \binom{m}{1} + ... + \binom{m}{r}$$

Una pregunta muy importante en teoría de códigos es la que se refiere a la distribución de pesos de un código , es decir, saber cuántas palabras de código existen con un determinado peso. En el caso de los códigos binarios de Reed-Muller esto es equivalente al problema de determinar cuántas funciones Booleanas de un determinado grado dan lugar a cuántas palabras de código con un peso de Hamming dado. Para los códigos binarios de Reed-Muller de órdenes 0, 1, 2, $m-1$ y $m-3$ se conoce una solución general, sin embargo para los órdenes restantes, éste ha sido un problema abierto por mucho tiempo [23]. No obstante, en el Capítulo tres de este trabajo se presentará un método que permite relacionar eficientemente el peso de Hamming de cualquier función Booleana de grado ≥ 4 con una de grado a lo más 3.

Capítulo 2

Antecedentes Sobre Criptografía

2.1 Introducción

La *criptografía* trata sobre los diferentes mecanismos que permiten salvaguardar la privacidad e integridad de la información que se almacena y/o transmite entre entidades, típicamente dispositivos electrónicos (por ejemplo computadores). En condiciones normales la integridad de la información no sufre ningún daño cuando es intercambiada entre dos entidades tales como un emisor y un receptor. Sin embargo, cuando la información se transfiere a través de mecanismos no seguros, es posible que una tercera entidad pueda tener acceso a esta información, comprometiéndose de esta forma la integridad de la misma. Esta tercera entidad u oponente puede entorpecer la comunicación entre emisor y receptor a través de alguno de los siguientes ataques:

- **Interrupción**
- **Intercepción**
- **Modificación**
- **Falsificación**

La interrupción se presenta cuando el oponente captura y retiene toda comunicación entre emisor y receptor. La intercepción se presenta cuando el oponente sólo captura la información sin impedir la comunicación. Por otro

lado, cuando el oponente altera la información que captura y después la reexpide al receptor, se presenta una modificación. Por último la falsificación se presenta cuando el oponente envía información al receptor haciéndole creer a éste que proviene del emisor.

El mecanismo que generalmente se emplea en criptografía para proteger la información, se denomina *Cifrado de Datos.* El cifrado de datos, es la técnica o arte de transformar mensajes en claro a mensajes cifrados, de tal manera que esta transformación y la transformación inversa sólo pueden ser factibles con el conocimiento de ciertas llaves (Fig. 2.1).

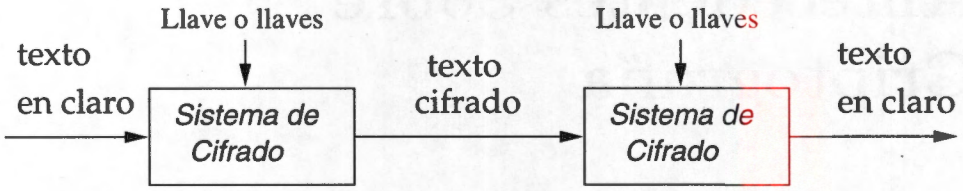


Figura 2.1: Cifrado y descifrado de la información.

A un sistema como el descrito en la Figura anterior se le denomina *Sistema de Cifrado.*

2.2 Sistemas de Cifrado

El componente fundamental de un sistema de cifrado es el procedimiento bajo el cual se realiza el cifrado y descifrado de la información. Tal procedimiento se basa en los llamados *Algoritmos de Cifrado.* Un buen sistema de cifrado requiere que su algoritmo cumpla con las siguientes características:

- Seguros
- Confiables
- Eficientes
- Sencillos
- Independencia de la plataforma de cómputo.

La seguridad o robustez de un algoritmo de cifrado **no** debe depender de la no publicación del algoritmo. Esto es, el algoritmo debe ser lo suficientemente robusto, de tal manera que no baste su conocimiento para poder

descifrar la información generada a través de este algoritmo. La confiabilidad de un algoritmo se refiere a la correctez del mismo, es decir se debe garantizar que siempre será posible el cifrado y descrifrado de la información, con el conocimiento de la o las llaves correspondientes. La eficiencia se refiere a que el algoritmo no debe consumir demasiados recursos para funcionar, principalmente en cuestión de tiempo. Es decir, cifrar o descifrar un mensaje no debe provocar cuellos de botella. Un algoritmo de cifrado deberá ser sencillo, esto es, un algoritmo demasiado rebuscado no implica de ninguna manera un mejor sistema de cifrado. Por último, el algoritmo de cifrado debe poder mudarse fácilmente de una a otra arquitectura de cómputo, esto es, debe requerir de cambios mínimos para ejecutarse en cualquier plataforma de cómputo.

2.3 Tipos de cifrado

Existen en la actualidad una gran variedad de sistemas de cifrado, sin embargo la mayoría de éstos pueden ser clasificados dentro de cualquiera de los dos siguientes tipos:

- *Sistemas simétricos o de llave secreta .*
- *Sistemas asimétricos o de llave pública .*

Los sistemas de llave secreta implican el uso de una sola llave para cifrar y descifrar la información. Es decir, tanto el emisor como el receptor comparten una única llave y de ahí que la seguridad de estos sistemas dependa de tal llave (Fig. 2.2). Dos de los algoritmos de llave secreta, más conocidos y utilizados son el *Data Encryption Standard* (DES) [26] y el *International Data Encryption Algorithm* (IDEA) [21].

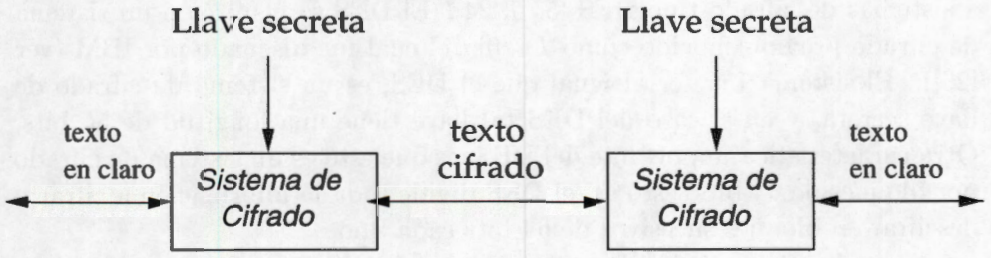


Figura 2.2: Cifrado y descifrado con llave secreta. Note la simetría en la Figura.

Los sistemas de llave pública (Fig. 2.3), a diferencia de los sistemas de llave secreta, involucran el uso de dos llaves; una para cifrar y otra descifrar. De esta manera, lo que se cifra con una llave sólo puede ser descifrado con la otra. A este tipo de sistemas se les denomina de llave pública, debido a que una de las dos llaves puede hacerse pública, mientras que la otra se mantiene en secreto (*llave privada*). Donde, esta última llave es en la práctica computacionalmente imposible de obtener de la primera.

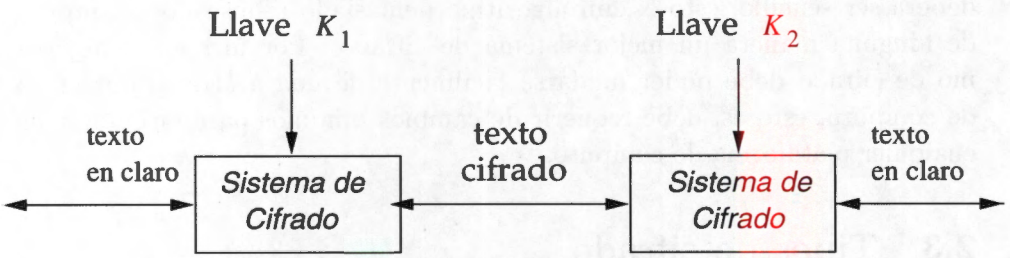


Figura 2.3: Cifrado y descifrado de llave pública, $K_1 \neq K_2$. Note la asimetría en la Figura.

A continuación se dará una breve descripción del sistema de cifrado DES.

2.4 El *Data Encryption Standard (DES)*

El *Data Encryption Standard* (DES), es uno de los algoritmos de cifrado moderno más antiguo y también uno de los que se ha empleado con mayor frecuencia. La razón de esta popularidad se debió principalmente a que en 1977 el *National Bureau of Standards* de los E.U., lo designó como el sistema de cifrado estándar para aplicaciones no clasificadas del gobierno de este país. Sin embargo, actualmente se está buscando reemplazarlo por otro, debido a la aparición de algunos ataques de criptoanálisis especialmente aplicables a sistemas de cifrado tipo DES [5, 6, 24]. El DES se inspiró en un sistema de cifrado previo conocido como *Lucifer* el cual fue diseñado por IBM (ver [26]). El sistema Lucifer al igual que el DES, es un sistema de cifrado de llave secreta, y en el caso del DES tal llave tiene una longitud de 56 bits. Otra característica importante del DES, es que éste es un sistema de cifrado por bloques de 64 bits, esto es, el DES divide toda la información a cifrar o descifrar en bloques sucesivos de 64 bits cada uno.

Antes de iniciar el cifrado o descrifrado, el DES primero procesa la llave secreta, K, de 56 bits. Tal proceso consiste en generar 16 subllaves secretas K_i $(i = 1, 2, ..., 16)$ las cuales se emplearán en el cifrado o descrifrado de cada

uno de los bloques de 64 bits que reciba posteriormente. Cada una de las subllaves K_i está formada por 48 bits extraídos de la llave K. El algoritmo que se emplea para la generación de estas subllaves consiste básicamente en permutar, para cada una de las subllaves, los 56 bits de la llave secreta y extraer de esta permutación 48 bits que darán lugar a la subllave. Los procedimientos de permutación y extracción de bits están bien definidos para cada subllave y son descritos a través de una serie de tablas las cuales pueden ser consultadas en [5, 33, 26].

Una vez que se tienen las 16 subllaves K_i, el cifrado y descifrado de cada bloque de 64 bits se realiza mediante:

- Permutaciones de bits.
- Sumas binarias tipo XOR entre los datos y la llave secreta.
- Funciones de sustitución que mediante tablas fijas mapean un grupo de 6 bits a un grupo de 4 bits.

Ahora bien, para comprender de una manera más fácil la forma en que el DES cifra o descifra, es conveniente recordar que si se tienen dos vectores binarios A y B de longitud $n > 0$, es decir $A, B \in \mathbb{F}_2^n$, entonces:

$$(A \oplus B) \oplus A = A \,. \tag{2.1}$$

Usando esta última propiedad, podemos ahora construir un sistema simétrico de cifrado. Supóngase que tenemos un dato D al cual podemos ver también como un vector binario de longitud n es decir $D \in \mathbb{F}_2^n$. De igual forma, supóngase que se tienen m vectores binarios B_i $(i = 1, 2, ..., m)$ de la misma longitud n. Estos vectores B_i pudieron haber sido generados en términos de una llave secreta. Ahora bien, si se suman todos los vectores B_i al dato D, se tiene:

$$D \oplus B_1 \oplus B_2 \oplus \cdots \oplus B_m = C \,. \tag{2.2}$$

El vector resultante C de la suma anterior, puede verse como el dato cifrado de D. Para descifrar a C bastará, según la idea en (2.1), con sumar a éste los mismos vectores B_i:

$$C \oplus B_m \oplus B_{m-1} \oplus \cdots \oplus B_1 = D \,. \tag{2.3}$$

Observe que lo expresado en la última ecuación siempre será posible para aquella entidad que sea capaz de generar los vectores B_i, lo cual a su

vez sólo será posible, en principio, con el conocimiento de la llave secreta. Así pues, para desarrollar un sistema de cifrado que opere bajo esta idea es necesario contar con una función que genere la secuencia de vectores B_i. Tal función en el DES, se conoce como función F. Esta función, descrita en la Figura 2.4, manipula 32 bits de entrada y 48 bits de una subllave a través de permutaciones, sustituciones y sumas binarias, de tal manera que a su salida se obtiene un grupo de 32 bits. La función F se emplea iterativamente 16 veces para cifrar o descifrar cada bloque de 64 bits.

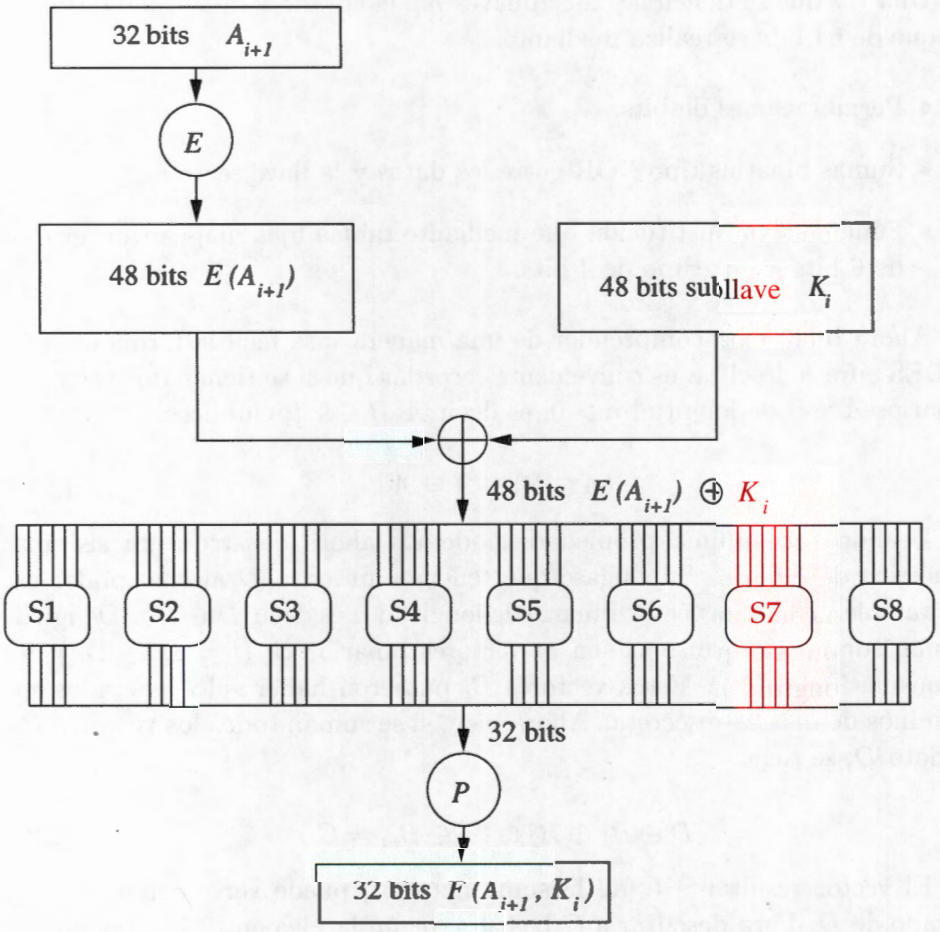


Figura 2.4: La función F del DES ($i = 1, 2, ..16$).

En la primera iteración el DES recibe un bloque D de 64 bits de longitud el cual se divide en 2 sub-bloques A_1 y A_2 de 32 bits cada uno. El sub-bloque A_2 junto con una subllave K_1 es usado como entrada a la función F, la cual

como ya se dijo, produce como resultado un grupo de 32 bits de salida $F(A_2, K_1)$ (ver Fig. 2.4). Este resultado es ahora sumado con el sub-bloque A_1, lo cual da lugar al sub-bloque A_3 de 32 bits también. Ahora A_3 es usado como entrada de F con la subllave K_2 y el resultado de F es sumado a A_2 lo que a su vez da lugar al sub-bloque A_4. El proceso anterior se itera hasta completar 16 aplicaciones de la función F y al final de la última aplicación se obtendrán dos sub-bloques C_1 y C_2 de 64 bits cada uno. Estos dos sub-bloques constituyen los 64 bits de salida del DES. El procedimiento anterior se ilustra a través de las siguientes relaciones:

$$\underbrace{\underbrace{\underbrace{A_1 \oplus F(A_2, K_1)}_{A_3} \oplus F(A_4, K_3)}_{A_5} \oplus \cdots \oplus F(A_n, K_{n-1})}_{A_{n+1}=C_2} = C_2 \,, \tag{2.4}$$

$$\underbrace{\underbrace{\underbrace{A_2 \oplus F(A_3, K_2)}_{A_4} \oplus F(A_5, K_6)}_{A_6} \oplus \cdots \oplus F(A_{n+1}, K_n)}_{A_{n+2}=C_1} = C_1 \,. \tag{2.5}$$

Observe que el proceso expresado en las ecuaciones anteriores se puede continuar para cualquier entero positivo par n, sin embargo para el DES, $n = 16$. La Figura 2.5 es una esquematización gráfica de las ecuaciones (2.4) y (2.5).

A continuación veremos como trabaja la función F. Como se puede ver en la Figura 2.4, F recibe dos datos de entrada: un vector A_{i+1} ($i = 1, 2, ..., 16$) de 32 bits y una subllave K_i de 48 bits, produciendo la salida $F(A_{i+1}, K_i)$ de 32 bits. Dicha función, toma el vector A_{i+1} y lo expande a 48 bits según la función de expansión E. La forma en que la función E realiza esta expansión es simplemente permutando las entradas de A_{i+1} y repitiendo algunas de éstas de tal manera que al final se tenga el vector de salida $E(A_{i+1})$ de 48 bits. Esta salida se suma (XOR) con la subllave K_i de 48 bits, obteniéndose como resultado otro vector de 48 bits $(E(A_{i+1}) \oplus K_i)$. Estos 48 bits son ahora agrupados en 8 grupos de 6 bits cada uno, donde el primer grupo está formado por los primeros 6 bits de estos 48 bits, el segundo grupo de los 6 siguientes bits y así sucesivamente hasta el octavo grupo. Cada uno de estos 8 grupos será la entrada a una estructura criptográfica conocida como *Caja de Sustitución* o *S-cajas* . Así pues, cada una de estas S-cajas, $S1, S2, ..., S8$, recibe 6 bits de entrada y entrega 4 bits de salida,

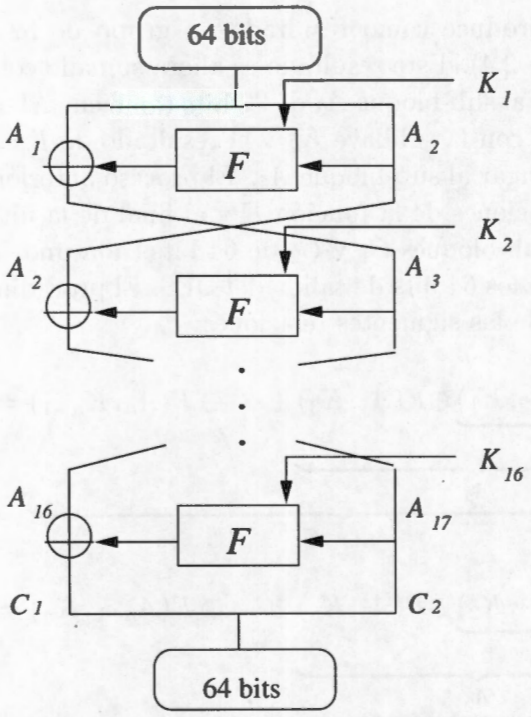


Figura 2.5: Diagrama Clásico para la descripción del DES.

por lo que a la salida de todas las S-cajas se tendrá un vector de 32 bits. Esta salida es pasada finalmente a una función de permutación P la cual simplemente re-posiciona estos 32 bits a su salida. Las tablas que definen a las funciones de expansión E, de permutación P y las cajas de sustitución $S1, S2, ..., S8$, son fijas y pueden igualmente ser consultadas en [5, 33, 26].

Ahora bien, las cajas de sustitución pueden ser vistas como funciones vectoriales multivaluadas que van de $\mathbb{F}_2^6$ a $\mathbb{F}_2^4$, esto es:

$$Si : \mathbb{F}_2^6 \longrightarrow \mathbb{F}_2^4 \quad i = 1, 2, ..., 8 \ .$$

Claramente la cardinalidad del dominio de cada Si es cuatro veces más grande que la cardinalidad de su imagen y por tanto deben existir $u_i, u_2 \in \mathbb{F}_2^6$ con $u_1 \neq u_2$, tal que $Si(u_1) = Si(u_2)$. Esto último indica que si se conoce la salida de una S-caja, entonces no se puede conocer con seguridad su entrada para todos los casos. Observe que esta característica no se presenta ni en la función de expansión E ni en la permutación P. De esta forma, la función principal de las S-cajas es incrementar la confusión en el proceso de cifrado.

De hecho, los principales ataques de criptoanálisis que se han desarrollado contra sistemas tipo DES se han dirigido contra deficiencias en el diseño de las S-cajas [5, 6, 24]. Por lo anterior, se reconoce que los componentes criptográficos más importantes de la función F y por ende del mismo DES, son las S-cajas.

2.4.1 Descifrado

Para descifrar los sub-bloques C_1 y C_2 en las ecuaciones (2.4) y (2.5), será necesario, según la idea expresada en las ecuaciones (2.2) y (2.3), sumar a éstos los mismos sub-bloques $F(A_{i+1}, K_i)$. Lo anterior puede ser logrado a través de las siguientes ecuaciones:

$$\underbrace{\underbrace{\underbrace{(A_{n+2} = C_1) \oplus F(A_{n+1}, K_n)}_{A_n} \oplus F(A_{n-1}, K_{n-2})}_{A_{n-2}} \oplus \cdots \oplus F(A_3, K_2)}_{A_2} = A_2 \ ,$$

$$\underbrace{\underbrace{\underbrace{(A_{n+1} = C_2) \oplus F(A_n, K_{n-1})}_{A_{n-1}} \oplus F(A_{n-2}, K_{n-3})}_{A_{n-3}} \oplus \cdots \oplus F(A_2, K_1)}_{A_1} = A_1 \ .$$

De lo anterior se desprende que para descifrar C_1 y C_2 se usará el mismo algoritmo que se usó para cifrar (ecuaciones (2.4) y (2.5)), con la única diferencia que las subllaves K_i's se habrán de emplear en orden inverso, esto es, para descifrar se usará primero la subllave K_{16} y después la K_{15} y así sucesivamente hasta que la última en usar sea K_1.

2.5 Las S-cajas y los Mapeos Regulares

Como ya se mencionó en la sección anterior, las S-cajas son los componentes criptográficos más importantes en los sistemas de cifrado tipo DES. Las S-cajas pueden generalizarse como mapeos que van de $\mathbb{F}_2^n$ a $\mathbb{F}_2^s$:

$$S : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^s \quad \text{con } 0 < s \leq n \text{ enteros.}$$

Una primera característica deseable de una S-caja es que ésta distribuya de manera uniforme las evaluaciones de su dominio en su imagen. Esto es, se desea que la S-caja sea un mapeo regular:

Definición 2.1 *Sean $0 < s \leq n$ enteros. Un mapeo $S : \mathbb{F}_2^n \to \mathbb{F}_2^s$ se dice que es regular si para toda $y \in \mathbb{F}_2^s$*

$$|S^{-1}(y)| = 2^{n-s} .$$

Observe que si S no fuera regular entonces, algunos elementos en $\mathbb{F}_2^s$ aparecerían con más frecuencia según el mapeo S y esto sería de gran utilidad en un ataque de criptoanálisis. Durante varios años el criterio de regularidad y otros criterios como el llamado *criterio estricto de avalancha* (Strict Avalanche Criterium; SAC) fueron fundamentales para el diseño de las S-cajas [43]. Sin embargo, con la publicación de [4] en 1991, se aceptó que un criterio deseable más, en el diseño de las cajas de sustitución, era que éstas fueran resistentes contra ataques de criptoanálisis diferencial .

El criptoanálisis diferencial [4, 5, 6] cae dentro de un tipo de ataque que se conoce como *texto en claro conocido* ("known plaint text attack"), el cual consiste en que, de alguna forma, el atacante ha sido capaz de colectar o conseguir varios mensajes cifrados con el mismo sistema y la misma llave, cuyo texto en claro él conoce. Dicho de otra manera, el atacante ha sido capaz de conseguir un banco de datos constituído por dos conjuntos: un conjunto C de mensajes cifrados y un conjunto T de los mensajes en claro correspondientes. Además, el atacante conoce también qué mensaje cifrado en C corresponde a qué mensaje en claro en T. De esta forma, la misión del atacante es usar esta información para tratar de deducir cuál fue la llave secreta que da lugar a los mensajes cifrados en C.

Como se verá más adelante, la táctica del criptoanálisis diferencial consiste en emplear el conjunto T para buscar parejas de texto en claro para las cuales sea factible estimar el valor de algunos bits que correspondan a la diferencia del cifrado parcial de cada pareja hasta antes de la penúltima iteración del DES. Esta estimación de algunos bits en las diferencias, es entonces usada junto con las parejas de texto cifrado en C para deducir algunos bits de la última subllave. Si los conjuntos C y T son lo suficientemente heterogéneos y grandes, es posible repetir el proceso anterior para buscar otro conjunto de parejas que permitan deducir otros bits de la última subllave. De esta manera, después de la elección de varios subconjuntos de parejas en T, se tendrá ya deducido la mayoría de los bits de la última subllave y por tanto la mayoría de los bits de la llave secreta. Los bits restantes de la llave secreta pueden ahora fácilmente ser encontrados de manera exhaustiva.

Debido a que esta técnica de criptoanálisis emplea diferencias de parejas de vectores binarios, se utilizará (x,x^*) para denotar a la pareja de vectores binarios x y x^* de la misma longitud. La diferencia de la pareja se denotará como x', esto es: $x' = x \oplus x^*$. Observe que el conocimiento de la diferencia no implica el conocimiento de la pareja.

Para efectuar las estimaciones antes mencionadas, el criptoanálisis diferencial construye para cada caja de sustitución S una estructura conocida como *tabla de distribución de diferencias*. La idea de la tabla de distribución de diferencias es describir la forma en cómo las diferencias a la entrada de una S-caja repercuten en diferencias a su salida (ver. Fig. 2.6).

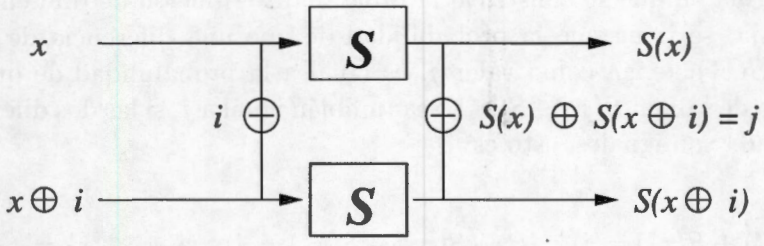


Figura 2.6: Diferencias i al entrar y diferencias j al salir de una S-caja.

Definición 2.2 *La tabla de distribución de diferencias de un mapeo* $S : \mathbb{F}_2^n \to \mathbb{F}_2^s$, *denotada por* $T(S)$, *es la matriz* $2^n \times 2^s$ *cuyas entradas* (i,j) *están dadas por:*

$$d_{ij} = |\{x | S(x) \oplus S(x \oplus i) = j\}| .$$

En los renglones de la matriz anterior, indexados por vectores en $\mathbb{F}_2^n$, se representan los cambios en la entrada de la S-caja, mientras que las columnas, indexadas por vectores en $\mathbb{F}_2^s$, representan los cambios en la salida de la S-caja. Una entrada en la tabla indexada por (i, j) indica el número de parejas de vectores en $\mathbb{F}_2^n$, los cuales difieren en i (en el sentido de la operación XOR entre bits) y cuyas salidas según S difieren en j (también en el sentido de la operación XOR entre bits). Observe que una entrada en la tabla de distribución de diferencias sólo puede tomar un valor par y que la suma de los valores en un renglón siempre es 2^n. En el caso especial del primer renglón, éste siempre tendrá la forma $(2^n, 0, ..., 0)$. Observe por otro lado que la primera columna indica la suavidad (*smoothness*) de la S-caja, es decir, la característica de que un cambio a la entrada de la caja no resulta en un cambio a su salida.

En [34] se usan las tablas de distribución de diferencias para introducir el concepto de ϵ-robustez, el cual se propone como una medida de vulnerabilidad de las cajas de sustitución frente a los ataques de criptoanálisis diferencial o lineal. En el Capítulo cuatro de este trabajo se presentará una caracterización de los mapeos regulares en términos de sus tablas de distribución de diferencias. Más aún, usando dicha caracterización se establecerá una nueva cota para la ϵ-robustez de las cajas de sustitución.

2.6 Criptoanálisis Diferencial en el DES

Por la forma en que se construye la tabla de distribución de diferencias de una S-caja, se tiene que la probabilidad de que una diferencia de salida, $S(x_1) \oplus S(x_1^*)$, tenga como valor j, es igual a la probabilidad de que otra diferencia de salida, $S(x_2) \oplus S(x_2^*)$, sea también igual a j, si las dos diferencias a la entrada son iguales, esto es:

$$P_{rob}(S(x_1) \oplus S(x_1^*) = j) = P_{rob}(S(x_2) \oplus S(x_2^*) = j) \quad \text{si} \quad x_1 \oplus x_1^* = x_2 \oplus x_2^* .$$

Por otro lado, debe observarse que las funciones de expansión E y permutación P en la función F del DES son tales que:

$$E(x \oplus x^*) = E(x) \oplus E(x^*) ,$$
$$P(x \oplus x^*) = P(x) \oplus P(x^*) .$$

para toda pareja (x, x^*) con $x, x^* \in \mathbb{F}_2^{32}$. Además se debe observar también que para toda subllave $k \in \mathbb{F}_2^{48}$ se tiene que:

$$(E(x) \oplus k) \oplus (E(x^*) \oplus k) = E(x) \oplus E(x^*) .$$

Las últimas igualdades son importantes pues de éstas se deduce que la función F del DES es tal que:

$$P_{rob}(F(x, k_1) \oplus F(x^*, k_1) = y') = P_{rob}(F(x, k_2) \oplus F(x^*, k_2) = y') .$$

para toda pareja (x, x^*) y todo par de subllaves k_1 y k_2. Lo anterior dice que la probabilidad del valor de la diferencia $F(x, k) \oplus F(x^*, k)$ no depende de la subllave k, y por tanto sólo en este caso se escribirá $F(x) \oplus F(x^*)$ en lugar de $F(x, k) \oplus F(x^*, k)$.

Si se considera dentro de las tablas de distribución de diferencias de las 8 S-cajas del DES, aquellas entradas con valores grandes, es posible

buscar parejas (x, x^*) en el conjunto T para las cuales la diferencia $y' = F(x) \oplus F(x^*)$ ocurra con buena probabilidad (ver. Fig. 2.7). Más aún, si los conjuntos T y C son lo suficientemente grandes y debido a que la $P_{rob}(y')$ es independiente de las subllaves k's, entonces es posible en principio [5, 6], elegir parejas (x, x^*) dentro del banco del texto en claro conocido T, para las cuales se puede estimar con una probabilidad mayor que cero algunos bits de la diferencia w' que sirve como entrada en la función F en la penúltima iteración en el DES. Ahora bien, para cada pareja (x, x^*) de texto en claro se conoce su pareja cifrada (c, c^*). Por tanto se conoce, por un lado, a la pareja y su diferencia a la entrada de F en la última iteración, y por otro lado, a la pareja y su diferencia del resultado de la suma de la salida de F en la última iteración con la entrada de F en la penúltima iteración. (ver. Fig. 2.8).

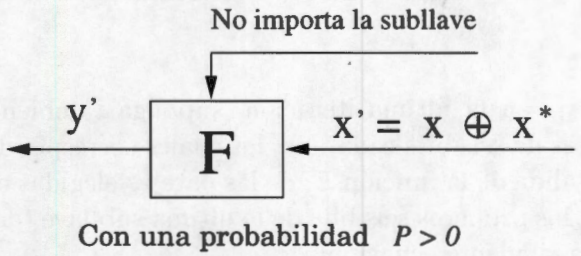


Figura 2.7: La diferencia y' a la salida de F se sucede con una probabilidad $P > 0$ para el conjunto de parejas de entradas (x, x^*) cuya diferencia es x'.

Con el conocimiento de las parejas cifradas (c, c^*) y sus diferencias es posible, con la ayuda del grupo de bits estimados en w', estimar ahora los correspondientes bits de diferencia del vector y' que salen de la función F en la última iteración. Si las parejas en T y C son elegidas apropiadamente, entonces los bits estimados en y' permiten a su vez estimar todos los bits de diferencia a la salida de algunas S-cajas. Por otro lado, se conocen completamente las parejas y sus diferencias que entran a la función F en la última iteración. De esta manera, se tiene un conocimiento total de las parejas y sus diferencias a la salida de la función de expansión E y se tiene también, la estimación de las diferencias a la salida de algunas de las cajas de sustitución, todo esto en la última iteración del DES. Con el conocimiento anterior, es posible ahora deducir algunos bits de la última subllave, los cuales sirven como entrada a aquellas S-cajas cuya diferencia a la salida está estimada.

Por ejemplo, suponga que $S1'_o$ es la diferencia estimada para la salida

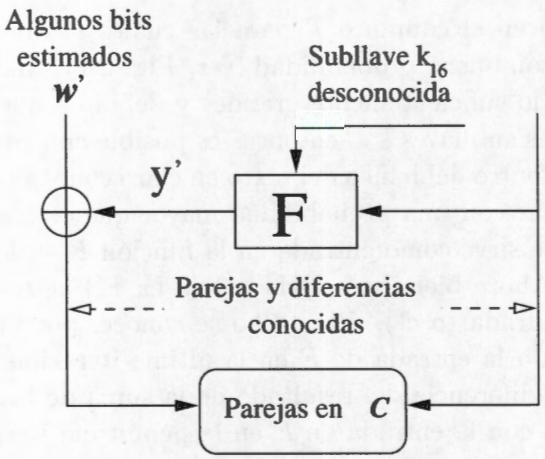


Figura 2.8: Parejas y diferencias estimadas o conocidas en la última iteración del DES.

de la primera S-caja en la última iteración, suponga también que $E_{(1)}$ y $E^*_{(1)}$ son dos vectores de seis bits cada uno, los cuales corresponden a los seis primeros bits a la salida de la función E, de las parejas elegidas en C. Ahora bien, si $k_{(1)}$ denota los primeros seis bits de la última subllave (desconocida), entonces se tiene la siguiente ecuación:

$$S1(E_{(1)} \oplus k_{(1)}) \oplus S1(E^*_{(1)} \oplus k_{(1)}) = S1'_o \tag{2.6}$$

Para deducir finalmente $k_{(1)}$, se usa la ecuación anterior en donde se cuentan para cada posible $k_{(1)}$ (existen 64 posibilidades) el número de parejas $(E_{(1)}, E^*_{(1)})$ para las cuales la ecuación (2.6) se cumple. El valor correcto para $k_{(1)}$ será (con la esperanza de que sea único) aquel que ha sido sugerido por todas las parejas.

Capítulo 3

Una Cota Sobre Funciones Booleanas

Es bien conocida la dificultad de caracterizar, en general, la distribución de pesos de los códigos de Reed-Muller de orden mayor o igual a 3. Es decir, la descripción de la distribución de pesos de todas las funciones Booleanas de grado ≥ 3 en m variables, es desde hace mucho tiempo, un problema abierto [23]. En [8] el autor introduce una transformación sobre funciones Booleanas, que al ser aplicada a éstas, cambia sus pesos de una manera sencilla de seguir, y en la cual, cuando se itera es posible reducir el grado de una función a 2 ó 3. Concluyendo que es tan difícil encontrar una manera de caracterizar el peso de cualquier función de grado 3, como lo es para cualquier otro grado mayor. Sin embargo, la aplicación de esta transformación sobre una función Booleana definida sobre $\mathbb{F}_2^m$, $m \in N$, lleva consigo la necesaria expansión de su dominio. Por otro lado, para reducir el grado de una función a 2 ó 3 es necesario aplicar la transformación un número de veces que crece de una manera exponencial respecto a m. En este Capítulo, se presenta un método de factorización para funciones Booleanas que permite establecer una cota superior para el número de aplicaciones de esta transformación, mostrando con esto, que en general, es posible ahorrar un buen número de iteraciones en este proceso de reducción de grado. Los resultados de este Capítulo pueden ser consultados en [40].

3.1 Introducción

En la Proposición 1 de [8] el autor introduce una transformación que permite asociar a cualquier función Booleana $f : \mathbb{F}_2^m \to \mathbb{F}_2$ otra función

$f^{[1]} : \mathbb{F}_2^{m+2} \to \mathbb{F}_2$, manteniendo una sencilla relación sobre sus pesos de Hamming. Aplicada de manera apropiada, esta transformación cambia cualquier sumando en f de grado $r \geq 4$ en tres sumandos en $f^{[1]}$ de grados 2, s y t, donde $r = s + t - 2$, y $s, t > 2$. Si el sumando es un monomio, entonces los tres nuevos sumandos serán también monomios y, en este caso, la transformación puede ser aplicada de tal manera que $t = 3$, y por tanto $s = r - 1$. Ahora, si $s \geq 4$, entonces la transformación puede ser aplicada otra vez al nuevo monomio de grado s para obtener una nueva reducción de grado, y claramente, si se continúa con este proceso, entonces $r - 3$ aplicaciones de esta transformación serán necesarias para transformar el monomio original en suma de monomios de grado ≤ 3. Si f se expresa en su forma normal, esto es, en términos de suma de monomios, entonces la transformación puede ser aplicada iterativamente, digamos p veces, sobre todos los monomios de f de grado $r \geq 4$ de tal manera que al final se obtiene una función $f^{[p]}$ de grado ≤ 3 y las dos funciones f y $f^{[p]}$ siguen manteniendo una relación sencilla entre sus pesos de Hamming. Sin embargo, dado que en general una función Booleana sobre $\mathbb{F}_2^m$ puede tener $\binom{m}{r}$ monomios de grado r, entonces el número de veces que la transformación pudiera ser aplicada para obtener $f^{[p]}$ es:

$$p = \sum_{i=4}^{m} (i-3) \binom{m}{i} . \tag{3.1}$$

Claramente el valor p crece exponencialmente con respecto a m. En este Capítulo, se presenta un método de factorización sobre funciones Booleanas, el cual permite establecer que una cota superior sobre el número de iteraciones para obtener la función $f^{[p]}$ es:

$$2^{n_1} + 2^{n_2} + 2^{n_3} - (m+3) , \tag{3.2}$$

donde $n_1 = \lceil m/3 \rceil$, $n_2 = \lceil (m - n_1)/2 \rceil$ y $n_3 = m - n_1 - n_2$ ($\lceil x \rceil$ denota el menor entero $\geq x$). Es evidente que la cota en la ecuación (3.2) también depende exponencialmente de m. No obstante, en general, a través de este método es posible reducir significativamente el número de aplicaciones iteradas de la transformación. Por ejemplo si $m = 9$, entonces de acuerdo con (3.1), $p \leq 825$, mientras que de acuerdo a (3.2), p no puede ser mayor que 12.

3.2 Notación y Definiciones Básicas

En esta sección se establecen algunas notaciones y hechos básicos sobre funciones Booleanas, así como también se presenta la trasnformación introducida en [8].

Sea $\mathbb{F}_2$ el campo de los números binarios y para un entero positivo m sea $S = \{1, 2, ..., m\}$ y $\mathbb{F}_2^m$, como un espacio vectorial sobre $\mathbb{F}_2$. El conjunto $\mathbb{F}_2^m$ puede ser identificado de una manera natural con el conjunto $\{0, 1, ..., 2^m - 1\}$.

El *peso de Hamming* de un vector $x \in \mathbb{F}_2^m$ denotado por $P_H(x)$, es el número de unos en x. Una *función Booleana* f es un mapeo $f : \mathbb{F}_2^m \to \mathbb{F}_2$ y su *tabla de verdad* es un vector en $\mathbb{F}_2^{2^m}$ dado por $(f(\alpha_0), ..., f(\alpha_{2^m-1}))$ donde α_i es tomado como la representación en binario a m bits del entero $i = 0, 1, ..., 2^m - 1$. El *peso de Hamming de una función Booleana* f, denotado por $P_H(f)$, es el número de unos en su tabla de verdad. El conjunto de todas las funciones Booleanas de $\mathbb{F}_2^m$ a $\mathbb{F}_2$ será denotado por $\mathcal{B}_m$. Nótese que $f(x)$ puede ser visto como un polinomio sobre $\mathbb{F}_2$ en las coordenadas $x_1, x_2, ..., x_m$ de x esto es, $\mathcal{B}_m$ puede ser identificado con el anillo de polinomios $\mathbb{F}_2[X_1, ..., X_m]/(X_1^2 - X_1, ..., X_m^2 - X_m)$, de tal manera que cualquier función Booleana puede ser escrita en términos de la base (natural) de monomios $\mathcal{M} = \{\mu^\alpha \doteq \prod_{i \in \alpha} x_i | \alpha \subseteq S\}$, esto es $\mathcal{B}_m = \{f | f = \sum_{\alpha \in I} \mu^\alpha; I \subseteq 2^S\}$. Si $f = \sum_{\alpha \in I} \mu^\alpha$, el *grado de* f se define como $gr(f) = max\{|\alpha| : \alpha \in I\}$.

Con las notaciones antes descritas se recordará la transformación dada en [8].

Proposición 3.1 *Sean f, f_1, f_2 y g funciones Booleanas sobre $\mathbb{F}_2^m$, donde $f(u) = f_1(u)f_2(u) + g(u)$ $\forall u \in \mathbb{F}_2^m$. Sea $f^{[1]}$ la función Booleana sobre $\mathbb{F}_2^{m+2}$ dada por $f^{[1]}(u, a, b) = ab + af_1(u) + bf_2(u) + g(u)$ $\forall u \in \mathbb{F}_2^m$ y $a, b \in \mathbb{F}_2$. Entonces los pesos de Hamming de f y $f^{[1]}$ satisfacen la siguiente relación:*

$$P_H(f) = \frac{1}{2} P_H(f^{[1]}) - 2^{m-1} .$$

Observe que cuando f_1 y f_2 son elegidas en f de tal manera que $gr(f_1)$, $gr(f_2) \geq 2$, entonces $gr(f - g) > gr(f^{[1]} - g)$ y esto quiere decir que un término de grado $gr(f_1 f_2)$ en f ha sido reemplazado por 3 términos de grado $max(gr(f_1), gr(f_2)) + 1$ en $f^{[1]}$. Claramente es posible continuar con este proceso, reemplazando en $f^{[1]}$ un término que es producto de dos funciones f_1 y f_2 que satisfagan $gr(f_1), gr(f_2) \geq 2$, por términos de menor grado que $gr(f_1 f_2)$. Este procedimiento puede ser aplicado, digamos p veces, antes de

obtener una función $f^{[p]}$ de grado ≤ 3. El hecho antes mencionado ha sido asentado en el resultado (Corolario 1 de [8]):

Corolario 3.2 *Para cualquier función Booleana f sobre $\mathbb{F}_2^m$, existe un entero p y una función Booleana $f^{[p]}$ sobre $\mathbb{F}_2^{m+2p}$ tal que $gr(f^{[p]}) \leq 3$ y:*

$$P_H(f) = 2^{m-1} - 2^{m+p-1} + \frac{1}{2^p} P_H(f^{[p]}) .$$

Si el Corolario anterior es aplicado a una función Booleana que es expresada en su forma normal, entonces se puede concluir que el valor de p está acotado por (3.1). De esta forma, para cada $I \subseteq 2^S$, definimos *el valor no óptimo de p para la $f = \sum_{\theta \in I} \mu^\theta$ a través del Corolario 3.2*, como:

$$N_f = \sum_{\theta \in I,\, |\theta| \geq 4} (|\theta| - 3) . \tag{3.3}$$

Note que N_f está acotada por (3.1). Por otro lado, para cada función $f \in \mathcal{B}_m$ definimos *el valor óptimo de p para la función f a través del Corolario 3.2*, O_f, como el valor más pequeño posible para tal función en el Corolario 3.2. Claramente $O_f \leq N_f$, y por tanto O_f es acotado también por (3.1). Observe, que en el caso especial cuando f es un monomio de grado $r \geq 4$, entonces $O_f = N_f = r - 3$, pero en el caso general el valor de O_f no se conoce. La intención de este Capítulo es establecer una nueva cota superior de O_f, para lo cual será necesario reducir el número de veces que la Proposición 3.1 debe ser aplicada sobre cualquier función Booleana f. En la siguiente sección, se presenta un método de factorización el cual permitirá el establecimiento de esta nueva cota superior.

3.3 El método

La idea básica del método es no aplicar la Proposición 3.1 directamente a cada uno de los monomios de la función f, como se sugiere por N_f en (3.3), sino realizar una serie de factorizaciones ordenadas entre estos monomios antes de aplicar la Proposición 3.1. En este sentido, se presenta el siguiente:

Lema 3.3 *Sea S un conjunto de índices con $n = |S| \geq 2$, y para toda $\alpha \subseteq S$ con $|\alpha| \geq 2$, sea B_α cualquier función Booleana en cualquier número de variables con la sóla restricción de que $gr(B_\alpha) \leq 2$. Si $f = \sum_{\alpha \subseteq S, |\alpha| \geq 2} \mu^\alpha B_\alpha$, entonces*

$$O_f \leq 2^n - n - 1 .$$

Demostración: Sea $\rho = |\{\alpha \subseteq S : |\alpha| \geq 2\}| = 2^n - n - 1$ y re-escribimos $f = \sum_{i=1}^{\rho} \mu^{\alpha_i} B_{\alpha_i}$ con $\alpha_i \subseteq S$, $|\alpha_i| \geq 2$ y $|\alpha_i| \geq |\alpha_j|$ para $i < j$ y $i, j = 1, 2, ..., \rho$. Observe que $\alpha_1 = S$. Para cada α_i, siempre que $B_{\alpha_i} \neq 0$, la Proposición 3.1 será aplicada en orden ascendente con respecto a $i = 1, 2, ..., \rho$, tomando como $f_1 = \mu^{\alpha_i}$, $f_2 = B_{\alpha_i}$, y g como los términos restantes de f. Para α_1 si $B_{\alpha_1} \neq 0$, entonces en la primera aplicación de la Proposición 3.1 se obtiene $f^{[1]} = a_1\mu^{\alpha_1} + b_1 B_{\alpha_1} + \sum_{i=2}^{\rho} \mu^{\alpha_i} B_{\alpha_i} + a_1 b_1$, donde a_1 y b_1 son dos nuevas variables. Claramente el término $b_1 f_2 = b_1 B_{\alpha_1}$ tiene grado ≤ 3 y por tanto éste no requiere de más aplicaciones de la Proposición 3.1. Si el término $a_1 f_1 = a_1 \mu^{\alpha_1}$ tiene grado ≤ 3, entonces éste también no requerirá de más aplicaciones de la Proposición 3.1. Por otro lado, si éste último término tiene grado ≥ 4, entonces deberá existir $j > 1$ tal que $\alpha_j \subset \alpha_1$ y $|\alpha_1| = |\alpha_j| + 1$, y por tanto, $f^{[1]}$ puede expresarse como $f^{[1]} = a_1 b_1 + b_1 B_{\alpha_1} + \sum_{i=2,\, i\neq j}^{\rho} \mu^{\alpha_i} B_{\alpha_i} + \mu^{\alpha_j}(B_{\alpha_j} + a_1 \mu^{\alpha_1 \setminus \alpha_j})$. Observe que $gr(B_{\alpha_j} + a_1 \mu^{\alpha_1 \setminus \alpha_j}) \leq 2$ y, consecuentemente, la función $\sum_{i=2,\, i\neq j}^{\rho} \mu^{\alpha_i} B_{\alpha_i} + \mu^{\alpha_j}(B_{\alpha_j} + a_1 \mu^{\alpha_1 \setminus \alpha_j})$ satisface la hipótesis del Lema, por lo que el análisis previo puede ser aplicado a esta nueva función, esta vez tomando α_2. Claramente, este proceso puede ser aplicado recursivamente hasta llegar a α_ρ, y por tanto, el número de aplicaciones de la Proposición 3.1 no será mayor al número de subconjuntos α en S con $|\alpha| \geq 2$, esto es, $p \leq \rho$. □

Para disminuir el valor de p en el Corolario 3.2, es posible tomar ventaja del Lema previo, sin embargo, será necesario antes re-expresar a f en términos de factorizaciones de sus monomios. Por tal razón, la siguiente notación y definición serán de utilidad en tal re-expresión de f:

Para cualquier entero positivo $m \geq 4$, sea $n_1 = \lceil m/3 \rceil$, $n_2 = \lceil (m - n_1)/2 \rceil$ y $n_3 = m - n_2 - n_1$. También sean $S = \{1, 2, ..., m\}$, $S_1 = \{1, 2, ..., n_1\}$, $S_2 = \{n_1 + 1, ..., n_1 + n_2\}$, $S_3 = \{n_1 + n_2 + 1, ..., m\}$, y 2^S el conjunto potencia de S.

Definición 3.4 *Sea $f \in \mathcal{B}_m$ dada en su forma normal $f = \sum_{\theta \in I} \mu^\theta$, para algún $I \subseteq 2^S$. Para tal función f, definimos $F_f : 2^S \to \mathbb{F}_2$ como:*

$$F_f(\theta) = \begin{cases} 1 & \textit{si el monomio } \mu^\theta \textit{ aparece en } f \\ 0 & \textit{de otro modo} \end{cases} \qquad \forall\, \theta \subseteq S\,.$$

Ahora sea $f \in \mathcal{B}_m$ y sin pérdida de generalidad, se asume que f en su forma normal no tiene monomios de grado ≤ 3. Entonces, se definen las siguientes funciones $\tilde{f}_1$, $\tilde{f}_2$ y $\tilde{f}_3$:

$$\tilde{f}_1 = \overbrace{\sum_{\substack{\alpha\subseteq S_1\\ |\alpha|\geq 2}}}^{A} \mu^{\alpha} \overbrace{\left(\sum_{\substack{\beta\subseteq S_2,\gamma\subseteq S_3\\ |\theta|\geq 4}} F_f(\theta)\mu^{\beta\cup\gamma} \right)}^{B}, \tag{3.4}$$

$$\tilde{f}_2 = \overbrace{\sum_{\substack{\beta\subseteq S_2\\ |\beta|\geq 2}}}^{C} \mu^{\beta} \underbrace{\left(\overbrace{\sum_{\gamma\subseteq S_3}}^{D} \mu^{\gamma} \overbrace{\left(\sum_{\substack{\alpha\subseteq S_1,|\alpha|\leq 1\\ |\theta|\geq 4}} F_f(\theta)\mu^{\alpha} \right)}^{E} \right)}_{F}, \tag{3.5}$$

$$\tilde{f}_3 = \overbrace{\sum_{\substack{\gamma\subseteq S_3\\ |\gamma|\geq 2}}}^{G} \mu^{\gamma} \underbrace{\left(\overbrace{\sum_{\substack{\beta\subseteq S_2\\ |\beta|\leq 1}}}^{H} \mu^{\beta} \overbrace{\left(\sum_{\substack{\alpha\subseteq S_1,|\alpha|\leq 1\\ |\theta|\geq 4}} F_f(\theta)\mu^{\alpha} \right)}^{I} \right)}_{J}, \tag{3.6}$$

donde $\theta = \alpha \cup \beta \cup \gamma$.

Observación 3.5 *Note que los términos indicados por las letras E, H e I son funciones Booleanas afines, esto es, de grado ≤ 1 y los términos indicados por J son funciones de grado ≤ 2.*

Proposición 3.6 *Con la notación introducida anteriormente $f = \tilde{f}_1 + \tilde{f}_2 + \tilde{f}_3$.*

Demostración: Sea X_1, X_2 y $X_3 \subseteq 2^S$ tal que $\tilde{f}_1 = \sum_{\alpha\in X_1} \mu^{\alpha}$, $\tilde{f}_2 = \sum_{\beta\in X_2} \mu^{\beta}$ y $\tilde{f}_3 = \sum_{\gamma\in X_3} \mu^{\gamma}$. Dado que $\alpha \in X_1 \iff |\alpha \cap S_1| \geq 2$, $\beta \in X_2 \iff (|\beta \cap S_2| \geq 2$ y $|\beta \cap S_1| \leq 1)$ y, $\gamma \in X_3 \iff (|\gamma \cap S_3| \geq 2$, $|\gamma \cap S_2| \leq 1$ y $|\gamma \cap S_1| \leq 1)$, se tiene necesariamente que $X_i \cap X_j = \phi$, para $i \neq j$, $i, j = 1, 2, 3$. Por la forma en que fueron construídas las funciones $\tilde{f}_1$, $\tilde{f}_2$ y $\tilde{f}_3$, se sigue que cualquier monomio μ^{θ}, $\theta \subseteq S$ que aparezca en $\tilde{f}_1 + \tilde{f}_2 + \tilde{f}_3$ también aparecerá en f. Inversamente, si μ^{θ}, $\theta \subseteq S$ es un monomio en f y dado que f no tiene monomios de grado ≤ 3, se sigue que, si $|\theta \cap S_1| \geq 2$, μ^{θ} aparece en $\tilde{f}_1$, y si $(|\theta \cap S_2| \geq 2$ y $|\theta \cap S_1| \leq 1)$, μ^{θ} aparece en $\tilde{f}_2$. Por último, si $(|\theta \cap S_3| \geq 2$, $|\theta \cap S_2| \leq 1$ y $|\theta \cap S_1| \leq 1)$, μ^{θ} aparece en $\tilde{f}_3$. □

Como consecuencia del resultado anterior, una cota superior sobre el número de veces que la Proposición 3.1 se aplica para obtener $f^{[p]}$ con $gr(f^{[p]}) \leq 3$, se establece a través del siguiente Corolario:

Corolario 3.7 *Sean f, $\tilde{f}_1$, $\tilde{f}_2$ y $\tilde{f}_3$ como antes. Entonces $O_f \leq 2^{n_1} + 2^{n_2} + 2^{n_3} - (m+3)$, y*

$$P_H(f) = 2^{m-1} - 2^{m+p-1} + \frac{1}{2^p} P_H(f^{[p]}) \,. \tag{3.7}$$

Demostración: De la Proposición 3.6, se tiene que $f = \tilde{f}_1 + \tilde{f}_2 + \tilde{f}_3$. Ahora, note que $\tilde{f}_1$ tiene la forma $\sum_{\alpha \subseteq S_1, |\alpha| \geq 2} \mu^\alpha B_\alpha$, donde para cada α, B_α es el término correspondiente dentro de los que se indican por la letra B en la ecuación (3.4). De esta manera, los productos de $\tilde{f}_1$ en f, pueden ser procesados en la misma forma como en la prueba del Lema 3.3, considerando en esta etapa del proceso a $\tilde{f}_2$ y $\tilde{f}_3$ como parte de la función g en la Proposición 3.1. Sin embargo, en este caso no se tiene la condición $gr(B_\alpha) \leq 2$. Por tal razón, después de la i-ésima aplicación de la Proposición 3.1, $1 \leq i \leq 2^{n_1} - n_1 - 1$, todos los monomios de grado ≥ 4 que aparezcan en $b_i B_{\alpha_i}$ deberán ser incluídos dentro de los productos de $\tilde{f}_2$ o $\tilde{f}_3$, dependiendo de los valores de β y γ. Esto es, si el monomio $b_i \mu^{\beta \cup \gamma}$ tiene grado ≥ 4 y aparece en $b_i B_{\alpha_i}$, entonces este monomio deberá ser factorizado con un producto en $\tilde{f}_2$ si $|\beta| \geq 2$, o bien con un producto en $\tilde{f}_3$ si $|\beta| \leq 1$. Tomando en consideración a γ, la factorización anterior puede hacerse, en cualquier caso, de tal manera que las funciones indicadas por las letras E, H e I en las ecuaciones (3.5) y (3.6) continuen siendo afines. Por tanto, el número de aplicaciones de la Proposición 3.1, para procesar los productos en $\tilde{f}_1$, será $p_1 \leq 2^{n_1} - n_1 - 1$.

Ahora, si M_2 y M_3 son, respectivamente, los conjuntos de monomios que fueron incluídos dentro de productos de $\tilde{f}_2$ y $\tilde{f}_3$, en la etapa previa, entonces las funciones $\tilde{f}_2{}' = \tilde{f}_2 + M_2$ y $\tilde{f}_3{}' = \tilde{f}_3 + M_3$ tienen la misma forma que $\tilde{f}_2$ y $\tilde{f}_3$ en (3.5) y (3.6), excepto que algunas nuevas funciones lineales de la forma b_i, $i = 1, ..., p_1$, pudieran aparecer dentro de los términos indicados por las letras E y I. Entonces, de manera análoga a $\tilde{f}_1$, es posible ahora procesar los productos, en $\tilde{f}_2{}'$, indicados por las letras C y F, tomando para este caso, el término corespondiente F_β indicado en (3.5) por F para cada $\beta \subseteq S_2$, $|\beta| \geq 2$. Una vez más, después de la j-ésima aplicación de la Proposición 3.1, $1 \leq j \leq 2^{n_2} - n_2 - 1$, todos los monomios de grado ≥ 4 que aparezcan en $b_{p_1+j} F_{\beta_j}$, deberán ser incluídos dentro de los productos de $\tilde{f}_3{}'$. Esto último simpre será posible, ya que cualquier monomio en $b_{p_1+j} F_{\beta_j}$ con grado ≥ 4

tiene la forma $b_{p_1+j}\mu^{\alpha\cup\gamma}$ con $|\alpha| \leq 1$ o $b_{p_1+j}b_i\mu^{\gamma}$, donde b_i y b_{p_1+j} son dos nuevas variables que fueron introducidas cuando se procesaron los productos en $\tilde{f}_1$ y $\tilde{f}_2{}'$, respectivamente. Por tanto, en cualquier caso, estos monomios pueden ser factorizados con productos en $\tilde{f}_3{}'$, de una manera obvia, y de tal forma que después de esta factorización, las funciones indicadas por la letra J en (3.6) continuen teniendo grado ≤ 2. De esta manera, el número de aplicaciones de la Proposición 3.1, para procesar los productos de $\tilde{f}_2{}'$, será $p_2 \leq 2^{n_2} - n_2 - 1$.

Por último, si M_4 es el conjunto de monomios que fueron incluídos en $\tilde{f}_3{}'$ en la etapa previa, entonces la función $\tilde{f}_3 + M_3 + M_4$ satisface las condiciones del Lema 3.3, y por tanto es posible procesar sus productos en $p_3 \leq 2^{n_3} - n_3 - 1$. De esta manera, el número total de aplicaciones de la Proposición 3.1 será a lo más $2^{n_1} + 2^{n_2} + 2^{n_3} - (m+3)$. La relación (3.7) se demuestra de manera análoga al Corolario 3.2 (cf. [8]). □

3.4 Un Ejemplo

Sea $m = 7$, entonces $S_1 = \{1,2,3\}$, $S_2 = \{4,5\}$ y $S_3 = \{6,7\}$, tomando la función Booleana f sobre $\mathbb{F}_2^7$ dada por:

$$\begin{aligned} f(X) \;=\; & x_1x_2x_3x_4x_5x_6x_7 + x_1x_2x_3x_4x_5x_7 + x_1x_2x_3x_5x_6x_7 + \\ & x_2x_3x_4x_5x_6x_7 + x_1x_2x_3x_4x_6 + x_1x_3x_4x_6x_7 + x_2x_3x_4x_5x_6 + \\ & x_2x_4x_5x_6x_7 + x_1x_2x_4x_7 + x_1x_3x_5x_7 + x_1x_4x_6x_7 + \\ & x_2x_5x_6x_7 + x_3x_5x_6x_7 + x_4x_5x_6x_7 \,, \end{aligned}$$

donde $X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in \mathbb{F}_2^7$. Es fácil ver que para esta función $N_f = 27$. Para reducir este valor, f se re-expresa como se sugiere en las ecuaciones (3.4), (3.5) y (3.6):

$$\begin{aligned} f(X) \;=\; & [x_1x_2x_3(x_4x_5x_6x_7 + x_4x_5x_7 + x_5x_6x_7 + x_4x_6) + x_1x_2(x_4x_7) + \\ & x_1x_3(x_4x_6x_7 + x_5x_7) + x_2x_3(x_4x_5x_6x_7 + x_4x_5x_6)] + \\ & [x_4x_5(x_6x_7(x_2+1))] + [x_6x_7(x_4(x_1) + x_5(x_2+x_3))] \,, \end{aligned}$$

Donde las funciones correspondientes $\tilde{f}_1$, $\tilde{f}_2$ y $\tilde{f}_3$ son aquellas que aparecen, respectivamente, en el primero, segundo y tercer grupo de corchetes cuadrados. Ahora bien, sea $f_1 = x_1x_2x_3$, $f_2 = x_4x_5x_6x_7 + x_4x_5x_7 + x_5x_6x_7 + x_4x_6$,

y g como los otros términos en f. Al aplicar la Proposición 3.1 se obtiene $f^{[1]}$, la cual a su vez puede se factorizada como sigue:

$$\begin{aligned} f^{[1]}(X, C_1) &= [x_1x_2(x_4x_7 + x_3a_1) + x_1x_3(x_4x_6x_7 + x_5x_7) + \\ &\quad x_2x_3(x_4x_5x_6x_7 + x_4x_5x_6)] + x_4x_6b_1 + a_1b_1 + \\ &\quad [x_4x_5(x_6x_7(x_2 + 1 + b_1) + x_7b_1)] + \\ &\quad [x_6x_7(x_4(x_1) + x_5(x_2 + x_3 + b_1))] , \end{aligned}$$

donde $C_t = (a_1, ..., a_t, b_1, ..., b_t) \in \mathbb{F}_2^{2t}$, para toda $t \in \mathbb{N}$. Aplicando ahora la Proposición 3.1 a los monomios con factores x_1x_2, x_1x_3 y x_2x_3, y realizando la refactorización, se tiene que:

$$\begin{aligned} f^{[4]}(X, C_4) &= [x_4x_5(x_6x_7(x_2 + 1 + b_1 + b_4) + x_7b_1 + x_6b_4)] + \\ &\quad [x_6x_7(x_4(x_1 + b_3) + x_5(x_2 + x_3 + b_1))] + x_4x_6b_1 + \\ &\quad x_4x_7b_2 + x_3a_1b_2 + x_5x_7b_3 + x_1x_2a_2 + x_1x_3a_3 + \\ &\quad x_2x_3a_4 + a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 . \end{aligned}$$

Aplicando la Proposición 3.1 a los términos con factor x_4x_5, y refactorizando tenemos:

$$\begin{aligned} f^{[5]}(X, C_5) &= [x_6x_7(x_4(x_1 + b_3) + x_5(x_2 + x_3 + b_1) + \\ &\qquad b_5(x_2 + 1 + b_1 + b_4))] + x_4x_6b_1 + x_4x_7b_2 + x_3a_1b_2 + \\ &\quad x_5x_7b_3 + x_1x_2a_2 + x_1x_3a_3 + x_2x_3a_4 + x_4x_5a_5 + \\ &\quad x_7b_1b_5 + x_6b_4b_5 + a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 + a_5b_5 . \end{aligned}$$

Si la Proposición 3.1 es aplicada una vez más a los términos con factor x_6x_7, se tiene finalmente:

$$\begin{aligned} f^{[6]}(X, C_6) &= b_6(x_4(x_1 + b_3) + x_5(x_2 + x_3 + b_1) + \\ &\qquad b_5(x_2 + 1 + b_1 + b_4)) + x_4x_6b_1 + x_4x_7b_2 + x_3a_1b_2 + \\ &\quad x_5x_7b_3 + x_1x_2a_2 + x_1x_3a_3 + x_2x_3a_4 + x_4x_5a_5 + x_7b_1b_5 + \\ &\quad x_6b_4b_5 + x_6x_7a_6 + a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 + a_5b_5 + a_6b_6 . \end{aligned}$$

Observe que el método en el ejemplo previo es igualmente aplicable a cualquier otra función $f \in \mathcal{B}_7$ y por tanto el costo para obtener $f^{[p]}$ con $gr(f^{[p]}) \leq 3$ para cualquier otra $f \in \mathcal{B}_7$ será a lo más de $p = 6$.

3.5 Comentario

Como se demostró en el Corolario 3.7, la ecuación (3.2) representa una cota superior para el número de veces que la Proposición 3.1 pudiera ser aplicada para obtener $f^{[p]}$ con $gr(f^{[p]}) \leq 3$. Sin embargo, para una función Booleana en general no es claro si existe o no una cota mejor. Para funciones Booleanas particulares, esta cota puede ser mejorada. Por ejemplo, la función $f \in \mathbb{F}_2^m$ dada por $f = \sum_{\alpha \in 2^S} \mu^\alpha$ con $S = \{1, 2, ..., m\}$ y $m \geq 3$, puede ser expresada como $f = (x_1+1)(x_2+1)\cdots(x_m+1)$, y si ésta es re-escrita como $f = y_1 y_2 \cdots y_m$ con $y_i = x_i + 1$, $i \in S$, entonces es posible procesar el monomio $y_1 y_2 \cdots y_m$ en tan sólo $m - 3$ aplicaciones de la Proposición 3.1.

Capítulo 4

Los Mapeos Regulares en Criptografía

En este Capítulo se presentan diversos resultados acerca de los mapeos regulares en conexión con sus cajas de sustitución o S-cajas . Se presenta concretamente una caracterización de los mapeos regulares, así como también se establece una nueva cota superior para la ϵ-robustez de las cajas de sustitución contra un ataque de criptoanálisis diferencial . Los resultados de este Capítulo pueden ser consultados en [39].

4.1 Introducción

Las cajas de sustitución o S-cajas son el componente más importante en los sistemas de cifrado por iteración tipo DES. Por esta razón, desde que hicieron aparición las técnicas de criptoanálisis diferencial [5, 6] y lineal [24], se han realizado importantes esfuerzos en el diseño de cajas de sustitución que sean inmunes a estas técnicas. Existen diferentes procedimientos para la construcción de las S-cajas [34, 27, 1, 43, 11], sin embargo, en este Capítulo se estudiará la posibilidad de construir tales cajas en términos de ciertas características que muestran sus tablas de distribución de diferencias. Es decir, se tiene el interés de estudiar la estructura misma de dichas tablas de tal manera que permita proponer una tabla que, por un lado muestre inmunidad contra el criptoanálisis diferencial, y por otro, tenga una caja de sustitución asociada. El problema en general no ha sido resuelto, sin embargo se han obtenido los siguientes resultados:

1. Una demostración alternativa para la caracterización de mapeos regu-

lares, originalmente dada en [22].

2. Una nueva caracterización de mapeos regulares en términos de su tabla de distribución de diferencias correspondiente.

3. En base a esta caracterización se establece una nueva cota superior para la ϵ-robustez de mapeos regulares dada en [34].

4. Se introduce una partición del conjunto de mapeos regulares en clases de equivalencia. Estas clases aportan información interesante acerca de tales mapeos cuyas tablas de distribución de diferencias son iguales y por tanto comparten una inmunidad equivalente ante un ataque de criptoanálisis diferencial.

4.2 Notación y Definiciones Básicas

En esta sección se establece la notación a usarse a lo largo de este Capítulo, así como también se recordarán algunas definiciones y resultados básicos acerca de las cajas de sustitución, los mapeos regulares, las tablas de distribución de diferencias y la medida ϵ-robustez.

Sea $\mathbb{F}_2$ el campo de los números binarios y para un entero positivo m sea $\mathbb{F}_2^m$, como espacio vectorial sobre $\mathbb{F}_2$. El conjunto $\mathbb{F}_2^m$ puede ser identificado de manera natural con el conjunto $\{0, 1, ..., 2^m - 1\}$.

El *peso de Hamming* de un vector $x \in \mathbb{F}_2^m$ denotado por $P_H(x)$, es el número de unos en x. Una *función Booleana* f es un mapeo $f : \mathbb{F}_2^m \to \mathbb{F}_2$ y su *tabla de verdad* es el vector en $\mathbb{F}_2^{2^m}$ dado por $(f(\alpha_0), ..., f(\alpha_{2^m-1}))$ donde α_i es tomado como la representación binaria a m bits del entero $i = 0, 1, ..., 2^m - 1$. Se dice que una función Booleana es *balanceada* si su tabla de verdad tiene el mismo número de unos que de ceros.

Sea $L_n = \{f | f : \mathbb{F}_2^n \to \mathbb{F}_2; f(x) = a_1x_1 \oplus ... \oplus a_nx_n\}$, el $\mathbb{F}_2$-espacio vectorial de las funciones *lineales* de $\mathbb{F}_2^n$ en $\mathbb{F}_2$, donde $x = (x_1, ..., x_n) \in \mathbb{F}_2^n$ y $a_i \in \mathbb{F}_2$. Sea $A_n = \{f | l \in L_n; f = l \oplus a_0\}$, con $a_0 \in \mathbb{F}_2$, el conjunto de funciones *afines* de $\mathbb{F}_2^n$ a $\mathbb{F}_2$. Nótese que toda función no constante que sea lineal o afín es balanceada. Una $n \times s$, *S-caja* es un mapeo de $\mathbb{F}_2^n$ a $\mathbb{F}_2^s$.

Definición 4.1 *Una función f de $\mathbb{F}_2^n$ a $\mathbb{F}_2$ se dice que satisface el criterio estricto de avalancha (Strict Avalanche Criterium; SAC) si $f(x) \oplus f(x \oplus \alpha)$ es balanceada para toda $\alpha \in \mathbb{F}_2^n$ con $P_H(\alpha) = 1$.*

En general estaremos interesados sólo en los mapeos regulares de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$, pues un primer requisito para la construcción de una buena caja de sustitución es que ésta sea un mapeo regular (ver Definición 2.1).

En un ataque de criptoanálisis diferencial [5] resultan de gran utilidad tanto las entradas de la tabla de distribución de diferencias (ver Definición 2.2) que muestren valores grandes (exceptuando la primera entrada) como las entradas diferentes de cero en la primera columna (también exceptuando la primera entrada). Por esta razón, una medida para la inmunidad contra ataques de criptoanálisis diferencial, deberá tomar en cuenta tanto el valor más alto en la tabla de distribución de diferencias como el número de entradas diferentes de cero en la primera columna. Una medida con tales características es la *ϵ-robustez*:

Definición 4.2 *Sea S un mapeo de $\mathbb{F}_2^n$ a $\mathbb{F}_2^s$. Denotemos por L el valor más grande dentro de la tabla de distribución de diferencias de S, y por R el número de entradas diferentes de cero en la primera columna de dicha tabla. En ambos casos el valor 2^n en el primer renglón no es tomado en cuenta. Entonces, se dirá que S es ϵ-robusto contra un ataque de criptoanálisis diferencial, donde:*

$$\epsilon = (1 - \frac{R}{2^n})(1 - \frac{L}{2^n})$$

Observe que la ϵ-robustez de una S-caja es pequeña si R y L toman valores grandes y por eso la ϵ-robustez de una S-caja con una buena resistencia contra un ataque de criptoanálisis diferencial deberá ser cercana a uno.

4.3 Una Caracterización de los Mapeos Regulares

Comenzaremos nuestra discusión sobre mapeos regulares dando una demostración alternativa para la caracterización de mapeos regulares presentada originalmente en el Corolario 7.39 de [22]. Para tal caracterización la siguiente Definición y Lema serán de utilidad.

Definición 4.3 *Sea f una función Booleana sobre $\mathbb{F}_2^n$ y sea $R \subset \mathbb{F}_2^n$. Decimos que f es marginalmente balanceada sobre R si la imagen de R bajo f tiene igual número de ceros y unos.*

Lema 4.4 *Sea $R \subset \mathbb{F}_2^n$ y para $s > 1$ entero, sean $f_1, f_2, ..., f_s$ funciones Booleanas sobre $\mathbb{F}_2^n$ de tal forma que cualquier combinación lineal diferente de cero de ellas es marginalmente balanceable sobre R. Entonces cualquier combinación lineal diferente de cero de $f_2, ..., f_s$ es marginalmente balanceable sobre $R \cap f_1^{-1}(0)$ o $R \cap f_1^{-1}(1)$.*

Demostración: Sea $g = a_2 f_2 \oplus ... \oplus a_s f_s$ una combinación lineal diferente de cero y sean $A = |R \cap f_1^{-1}(0) \cap g^{-1}(0)|$, $B = |R \cap f_1^{-1}(0) \cap g^{-1}(1)|$, $C = |R \cap f_1^{-1}(1) \cap g^{-1}(0)|$ y $D = |R \cap f_1^{-1}(1) \cap g^{-1}(1)|$. Ahora, dado que f_1, g, y $f_1 \oplus g$ son marginalmente balanceados sobre R, entonces $A+B = C+D = A+C = B+D = |R|/2$, y $|R \cap (f_1 \oplus g)^{-1}(0)| = A+D = |R \cap (f_1 \oplus g)^{-1}(1)| = B+C = |R|/2$. Esto implica que $A = B = C = D = |R|/4$ y por tanto el Lema se cumple. □

Corolario 4.5 *Sean R y $f_1, f_2, ..., f_s$ como en el Lema previo, y sea $(y_1, y_2, ..., y_s) \in \mathbb{F}_2^s$. Entonces f_s es marginalmente balanceado sobre $W = R \cap f_1^{-1}(y_1) \cap f_2^{-1}(y_2) \cap ... \cap f_{s-1}^{-1}(y_{s-1})$ y por tanto $|W \cap f_s^{-1}(y_s)| = |R|/2^s$.*

Teorema 4.6 *Sea $F(x) = (f_1(x), ..., f_s(x))$, donde cada f_i es una función Booleana sobre $\mathbb{F}_2^n$ y $n \geq s$. Entonces F es regular si y sólo si cualquier combinación lineal diferente de cero de $f_1, ..., f_s$ es balanceada.*

Demostración: Supóngase que F es regular. Sea $a_1 f_1 \oplus ... \oplus a_s f_s$ una combinación lineal diferente de cero, y sea h la función sobre $\mathbb{F}_2^s$ dada por $h(y_1, ..., y_s) = a_1 y_1 \oplus ... \oplus a_s y_s$. La función h es balanceada debido a que es una función lineal diferente de cero, y por tanto $|h^{-1}(0)| = 2^{s-1}$. Ahora bien, dado que F es regular, entonces para cada $y = (y_1, ..., y_s) \in h^{-1}(0)$ existen 2^{n-s} vectores en $\mathbb{F}_2^n$ que son mapeados por F a y, y esto quiere decir que $2^{s-1} \cdot 2^{n-s}$ vectores en $\mathbb{F}_2^n$ son mapeados por F a cero. Esto demuestra la implicación directa. La implicación inversa es obvia si $s = 1$. Si $s > 1$, entonces también es inmediata del Corolario 4.5 cuando $R = \mathbb{F}_2^n$, debido a que en este caso, para cada $y \in \mathbb{F}_2^s$, existen $|\mathbb{F}_2^n|/2^s = 2^{n-s}$ vectores en $\mathbb{F}_2^n$ que son mapeados por F a y. □

4.4 Otra Caracterización de los Mapeos Regulares

Una distinción importante sobre las tablas de distribución de diferencias es reconocer cuándo una de estas tablas proviene de un mapeo regular. Se presenta a continuación un resultado que establece cuáles son las condiciones necesarias y suficientes para que una tabla de distribución de diferencias provenga de un mapeo regular, pero antes el siguiente:

Lema 4.7 *Sea m un entero positivo y sean k_i, $i = 1, ..., m$ también enteros tales que $\sum_{i=1}^{m} k_i = mn$, para algún entero n. Entonces $\sum_{i=1}^{m} k_i^2 = mn^2$ si y sólo si $k_i = n$ para todo i.*

Demostración: Supongamos que $\sum_{i=1}^{m} k_i^2 = mn^2$ y sea r_i enteros tales que $k_i + r_i = n$ para toda $i = 1, ..., m$. Entonces $\sum_{i=1}^{m}(k_i + r_i)^2 = \sum_{i=1}^{m} k_i^2 = mn^2$ lo cual implica que $\sum_{i=1}^{m}(2k_i r_i + r_i^2) = 0$. Por otra parte, dado que $k_i r_i = r_i n - r_i^2$ entonces $\sum_{i=1}^{m}(2r_i n - r_i^2) = 0$, pero como $\sum_{i=1}^{m} r_i = 0$ se tiene que $\sum_{i=1}^{m} r_i^2 = 0$, de lo cual se concluye que $r_i = 0$ y por tanto $k_i = n$ para toda $i = 1, ..., m$. La otra dirección es obvia. □

Del Lema anterior se puede concluir que para algún entero n y para m enteros k_i tales que $\sum_{i=1}^{m} k_i = mn$, entonces $\sum_{i=1}^{m} k_i^2 \geq mn^2$ y la igualdad se logra cuando $k_i = n$ para toda $i = 1, ..., m$.

Estamos ahora en posición de dar otra caracterización de los mapeos regulares en términos de su tabla de distribución de diferencias.

Teorema 4.8 *Sea S un mapeo de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$. Entonces S es regular si y sólo si la suma de las entradas de cualquier columna de la matriz $T(S)$ (ver Definición 2.2) es igual a 2^{2n-s}.*

Demostración: Supóngase que el mapeo S es regular. Sea $j \in \mathbb{F}_2^s$ fijo y sea $y \in \mathbb{F}_2^s$ cualquier elemento. Note que por cada pareja $a, b \in \mathbb{F}_2^n$ con $a \in S^{-1}(y)$ y $b \in S^{-1}(y \oplus j)$ se tiene que $a, b \in \{x | S(x) \oplus S(x \oplus i) = j\}$ con $i = a \oplus b$. Ahora bien, como S es regular se tiene que $|S^{-1}(y)| = |S^{-1}(y \oplus j)| = 2^{n-s}$ y por tanto $|S^{-1}(y) \times S^{-1}(y \oplus j)| = 2^{n-s} \cdot 2^{n-s}$ de lo cual se sigue que $\sum_{i \in \mathbb{F}_2^n} d_{ij} = 2^s(2^{n-s} \cdot 2^{n-s}) = 2^{2n-s}$. Para probar el inverso, sea $y_0, ..., y_{2^s-1}$ una lista de todos los elementos en $\mathbb{F}_2^s$ y sea $k_j = |S^{-1}(y_j)|$ para cada $j \in \mathbb{F}_2^s$. Es obvio que $\sum_{j \in \mathbb{F}_2^s} k_j = 2^n = 2^s \cdot 2^{n-s}$. Note que $\sum_{i \in \mathbb{F}_2^n} |\{x | S(x) \oplus S(x \oplus i)\}| = \sum_{j \in \mathbb{F}_2^s} |\{(a, b) | a, b \in S^{-1}(y_j)\}|$ y $|\{(a, b) | a, b \in S^{-1}(y_j)\}| = k_j^2$. Entonces $\sum_{i \in \mathbb{F}_2^n} d_{i0} = \sum_{j \in \mathbb{F}_2^s} k_j^2 = 2^{2n-s} = 2^s(2^{n-s})^2$. Por tanto se tiene que $\sum_{j \in \mathbb{F}_2^s} k_j = 2^s \cdot 2^{n-s}$ y $\sum_{j \in \mathbb{F}_2^s} k_j^2 = 2^s(2^{n-s})^2$, y por el Lema 4.7, $k_j = 2^{n-s}$ para toda $j \in \mathbb{F}_2^s$. □

Como una consecuencia del Teorema anterior, se tiene el siguiente Corolario:

Corolario 4.9 *Sea S un mapeo de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$. entonces $\sum_{i \in \mathbb{F}_2^n} |\{x | S(x) = S(x \oplus i)\}| \geq 2^{2n-s}$ y la igualdad se alcanza si y sólo si S es regular.*

Como una aplicación del Teorema 4.8, se presenta una prueba alternativa del siguiente resultado, el cual se presentó originalmente en [35].

Lema 4.10 *Sea S un mapeo regular de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$, y sea δ el valor más grande en $T(S)$, sin tomar en cuenta el valor 2^n en el primer renglón. Si S es regular entonces $\delta > 2^{n-s}$.*

Demostración: Obsérvese primero que $2^{n-s} \leq \delta \leq 2^n$ para cualquier mapeo S. Supóngase que $\delta = 2^{n-s}$ y dado que S es regular, del Teorema 4.8 se sigue que la suma de las entradas en cualquier columna de $T(S)$ es 2^{2n-s}. Esto último implica a su vez que todas las entradas en $T(S)$ deben ser igual a 2^{n-s}, lo cual es una contradicción dado que la primera entrada en $T(S)$ es 2^n. □

4.5 Una cota superior para la ϵ-robustez

El Teorema 4.8 establece que para un mapeo regular S de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$, la suma de las entradas en la primera columna de su tabla de distribución de diferencias es 2^{2n-s}. Esto quiere decir entonces que la ϵ-robustez para tal mapeo S pudiera ser mejorada si existiera un mapeo regular S' también de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$, tal que su tabla de distribución de diferencias fuera igual a la del mapeo S exceptuando que en las entradas de la primera columna (sin contar la primera entrada) se encontrarán mayoritariamente los valores de cero o L (L según se introdujo en la Definición 4.2). De esta manera, lo anterior permite introducir la siguiente cota para la ϵ-robustez de los mapeos regulares.

Lema 4.11 *Sea S un mapeo regular de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$. Entonces una cota superior para la ϵ-robustez del mapeo S es:*

$$\epsilon^0 = (1 - \sqrt{2^{-s} - 2^{-n}})^2$$

Demostración: Como el mapeo S es regular entonces la suma de las entradas de la primera columna de $T(S)$ es 2^{2n-s} y por tanto la el valor $\epsilon = (1 - \frac{R}{2^n})(1 - \frac{L}{2^n})$ para el mapeo será máximo si $R = (2^{2n-s} - 2^n)/L$. Consecuentemente, la ϵ-robustez será $\epsilon = (1 - \frac{2^{n-s}-1}{L})(1 - \frac{L}{2^n})$. Es fácil comprobar que esta función tiene el valor máximo de $\epsilon^0 = (1 - \sqrt{2^{-s} - 2^{-n}})^2$ en $L^0 = \sqrt{2^n(2^{n-s} - 1)}$. □

En [34] se introdujo $\epsilon' = (1 - \frac{1}{2^n})(1 - 2^{-s+1})$ como una cota superior para la ϵ-robustez contra un ataque de criptoanálisis diferencial para una $n \times s$, S-caja. En dicho trabajo se dice que no se tiene la certeza de si tal cota superior es alcanzable o no. Se mostrará a continuación que para mapeos regulares de $\mathbb{F}_2^n$ a $\mathbb{F}_2^s$ con $n > s > 1$, entonces la cota introducida en el Lema 4.11 es mejor a la propuesta en [34].

Lema 4.12 *Sean n y s enteros tales que $n > s > 1$. Entonces:*

$$(1-\frac{1}{2^n})(1-2^{-s+1}) > (1-\sqrt{2^{-s}-2^{-n}})^2$$

Demostración: Es claro que $14 \leq 7 \cdot 2^{n-2}$ para $n > 2$. Sumemos a ambos lados de la desigualdad anterior el valor de $9 \cdot 2^{n-2} - 2^4$, teniéndose entonces $9 \cdot 2^{n-2} - 2 \leq 2^{n+2} - 2^4$. Ahora bien, observe que la función $f(s) = 2^{n+s} - 2^{2s}$ es creciente en $s \in \{2, ..., n-1\}$, por tanto $2^{n+s} - 2^4 \leq 2^{n+s} - 2^{2s}$ para $n > s > 1$. Por otro lado dado que $9 \cdot 2^{n-2} - 3 + 2^{-n} < 9 \cdot 2^{n-2} - 2$ y por consiguiente $9 \cdot 2^{n-2} - 3 + 2^{-n} < 2^{n+s} - 2^{2s}$. Al multiplicar ambos lados de la desigualdad anterior por 2^n se tiene que $(3 \cdot 2^{n-1} - 1)^2 < 2^{2(n+s)}(2^{-s} - 2^{-n})$. Al extraer raíz cuadrada y multiplicar ambos lados por -2^{1-n-s} se sigue que $-3 \cdot 2^{-s} + 2^{1-n-s} > -2\sqrt{2^{-s} - 2^{-n}}$. Finalmente si se suma $1 + 2^{-s} - 2^{-n}$, otra vez, a ambos lados de la desigualdad y si se factoriza, se tiene el resultado. □

En la Figura 4.1 se comparan gráficamente la cota $\epsilon' = (1-\frac{1}{2^n})(1-2^{-s+1})$ y la cota $\epsilon^0 = (1-\sqrt{2^{-s}-2^{-n}})^2$ para algunas $n \times s$, S-cajas. Esta gráfica muestra cualitativamente las dos cotas y de hecho, para el tipo de cajas que se muestra en la figura, la cota del Lema 4.11 está, en promedio, por debajo de la propuesta en [34] en un 27.23 %.

4.6 Una partición del conjunto de mapeos

Otra característica interesante de las tablas de distribución de diferencias es que existen en general varios mapeos S que comparten una misma tabla, como se mostrará a continuación.

Sean n, s enteros tales que $1 \leq s < n$ y sea G el grupo (aditivo) $\mathbb{F}_2^n \times \mathbb{F}_2^s$ con la siguiente operación: si $(r,t), (r',t') \in \mathbb{F}_2^n \times \mathbb{F}_2^s$ entonces $(r,t) + (r',t') = (r \oplus r', t \oplus t')$. Sea $\mathcal{S}_{n,s}$ el conjunto de todos los mapeos de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$.

La siguiente operación define una acción del grupo G sobre el conjunto $\mathcal{S}_{n,s}$:

$$G \times \mathcal{S}_{n,s} \to \mathcal{S}_{n,s},\ ((r,t), S) \to (r,t)S$$

donde $(r,t)S : \mathbb{F}_2^n \to \mathbb{F}_2^s$ es el mapeo definido como: $[(r,t)S](x) = S(x \oplus r) \oplus t$ para toda $x \in \mathbb{F}_2^n$. La órbita de un elemento $S \in \mathcal{S}_{n,s}$ será denotada por $\mathcal{O}_S$ y el correspondiente *grupo de isotropía* por G_S.

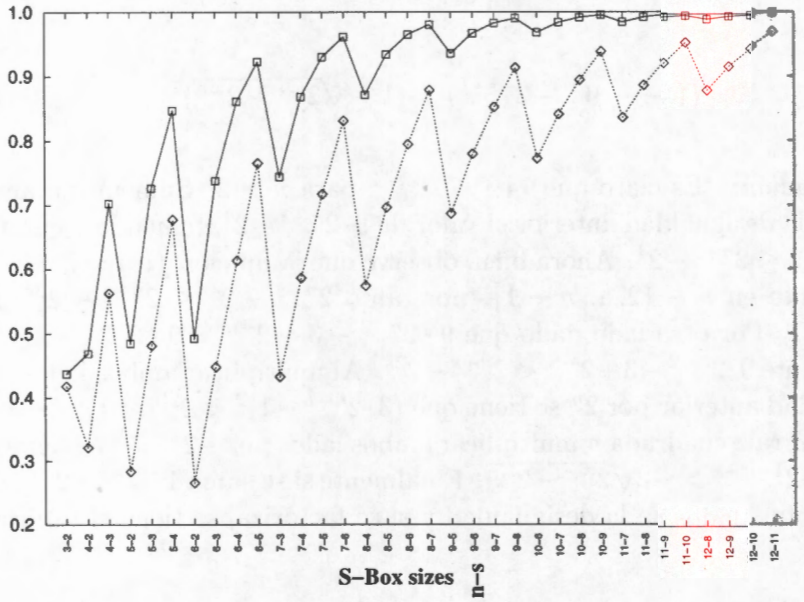


Figura 4.1: Comparación entre la cota ϵ' (gráfica continua) y la ϵ^0 (gráfica punteada). En la parte inferior se indican los tamaños $n \times s$ de las S-cajas.

Observe que la acción de G sobre $\mathcal{S}_{n,s}$, induce una *relación de equivalencia* sobre los elementos de $\mathcal{S}_{n,s}$: dos elementos $S, S' \in \mathcal{S}_{n,s}$ están relacionados, $S \sim S'$, si hay un elemento $(r,t) \in G$ tal que $S' = (r,t)S$. La clase de equivalencia de un elemento $S \in \mathcal{S}_{n,s}$ es justamente la órbita de S.

Lema 4.13 *Usando la notación anterior, si $S_1, S_2 \in \mathcal{S}_{n,s}$, están en la misma órbita, entonces $T(S_1) = T(S_2)$. Además la cardinalidad de cualquier órbita es tal que $2^s \leq |\mathcal{O}_S| \leq 2^{n+s-k}$ para algún entero $0 \leq k \leq n$.*

Demostración: Si S_1, S_2 están en la misma órbita entonces $S_2(x) = S_1(x \oplus r) \oplus t$ para algún $(r,t) \in \mathbb{F}_2^n \times \mathbb{F}_2^s$. Es claro que si $x' \in \{x \in \mathbb{F}_2^n : S_1(x) \oplus S_1(x \oplus i) = j\}$ entonces $x' \oplus r \in \{x \in \mathbb{F}_2^n : S_2(x) \oplus S_2(x \oplus i) = j\}$, e inversamente si $x' \in \{x \in \mathbb{F}_2^n : S_2(x) \oplus S_2(x \oplus i) = j\}$, entonces $x' \oplus r \in \{x \in \mathbb{F}_2^n : S_1(x) \oplus S_1(x \oplus i) = j\}$. Consecuentemente $|\{x \in \mathbb{F}_2^n : S_1(x) \oplus S_1(x \oplus i) = j\}| = |\{x \in \mathbb{F}_2^n : S_2(x) \oplus S_2(x \oplus i) = j\}|$, para toda $i \in \mathbb{F}_2^n$ y toda $j \in \mathbb{F}_2^s$. Por lo tanto $T(S_1) = T(S_2)$. La afirmación sobre la cardinalidad de la órbita se sigue del hecho de que $|\mathcal{O}_S| = |G|/|G_S|$. □

El Lema anterior es válido para cualquier mapeo de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$. Si el mapeo S es regular, entonces por el Lema 4.13, S y el mapeo $S'(x \oplus r) \oplus t$

tienen la misma tabla de distribución de diferencias, y por el Teorema 4.8 este último mapeo también es regular. Por tanto, la operación $(r,t)S$ manda mapeos regulares en mapeos regulares. De esta manera, si nos restringimos a mapeos regulares de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$, se tiene el siguiente resultado, el cual es consecuencia del Lema 4.13:

Corolario 4.14 *Si los mapeos regulares* $S_1, S_2 : \mathbb{F}_2^n \to \mathbb{F}_2^s$ *son equivalentes, entonces* $T(S_1) = T(S_2)$.

Note que bajo la transformación dada por la acción del grupo anterior, las componentes Booleanas, $f_1, ..., f_s$, de un mapeo $S(x) = (f_1(x), ..., f_s(x))$ de $\mathbb{F}_2^n$ en $\mathbb{F}_2^s$ mantienen la propiedad del *criterio estricto de avalancha (SAC)* (cf.[43] o sección 4.2). Como es bien sabido, esta propiedad es considerada como un requisito indispensable en la construcción de S-cajas. De esta manera, todos los elementos en una clase de equivalencia, de acuerdo con la relación "$\sim$", comparten las mismas fortalezas o debilidades ante un ataque del criptoanálisis diferencial. En otras palabras, de acuerdo al Lema 4.13, si S_0 es un mapeo con ciertas propiedades contra un criptoanálisis diferencial, entonces la anterior acción del grupo $\mathbb{F}_2^n \times \mathbb{F}_2^s$ da un método para encontrar otros mapeos con las mismas propiedades que S_0.

Capítulo 5

$\mathbb{Z}_{2^k}$-Códigos Lineales

Recientemente el concepto de código negacíclico fue presentado en [44], en el cual se prueban algunas relaciones entre los códigos negacíclicos y sus imágenes bajo el mapeo de Gray. En este Capítulo, se presenta, para $k \geq 1$, una isometría φ^k entre códigos sobre $\mathbb{Z}_{2^{k+1}}$ y códigos sobre $\mathbb{Z}_4$, la cual es empleada para establecer una generalización del mapeo de Gray semejante a la que se presenta en [9]. Con ayuda de esta isometría, el concepto de códigos negacíclicos se extiende ahora a códigos sobre el anillo $\mathbb{Z}_{2^{k+1}}$, a los cuales llamaremos códigos *hpo-cíclicos* (half plus one-cyclic codes). Una caracterización de estos códigos en términos de sus imágenes bajo φ^k, se presenta en este Capítulo. Además, también se prueba que la imagen generalizada de Gray de un código *hpo*-cíclico es un código binario cuasi-cíclico de distancia invariante (no necesariamente lineal). Por último, se discutirán también algunos códigos lineales *hpo*-cíclicos. Los resultados aquí expuestos han sido arbitrados y se presentaron en el "International Workshop on Coding and Cryptography", que se realizó en enero del 2001 en la Ciudad de París, Francia [41].

5.1 Introducción

En [44] se introduce el mapeo *negashift*, ν, sobre el anillo $\mathbb{Z}_4$ de los enteros módulo 4, como la permutación del módulo $\mathbb{Z}_4^n$, dado por:

$$\nu(a_0, a_1, ..., a_{n-1}) = (-a_{n-1}, a_0, a_1, ..., a_{n-2}) ,$$

con lo cual se define un *código negacíclico* de longitud n sobre $\mathbb{Z}_4$ como un subconjunto C de $\mathbb{Z}_4^n$ tal que $\nu(C) = C$.

Sea n un entero positivo y sea λ una unidad en el anillo $\mathbb{Z}_{2^{k+1}}$ de enteros módulo 2^{k+1}, con $k \geq 1$. Entonces, se define el mapeo λ-*shift*, ν_λ, como la permutación del módulo $\mathbb{Z}_{2^{k+1}}^n$ dada por:

$$\nu_\lambda(a_0, a_1, ..., a_{n-1}) = (\lambda a_{n-1}, a_0, a_1, ..., a_{n-2}) .$$

Análogamente al caso de negaciclicidad, se define un *código* λ-*cíclico* de longitud n sobre $\mathbb{Z}_{2^{k+1}}$ como un subconjunto C de $\mathbb{Z}_{2^{k+1}}^n$ tal que $\nu_\lambda(C) = C$. A estos códigos los llamaremos *hpo-cíclicos* cuando se cumpla que la unidad λ de $\mathbb{Z}_{2^{k+1}}$ es igual a $2^k + 1$ (half of 2^{k+1} plus one).

En la siguiente sección se definirá, para $k \geq 1$, una isometría entre códigos sobre $\mathbb{Z}_{2^{k+1}}$ y códigos sobre $\mathbb{Z}_4$ y se demostrará que con ayuda de esta isometría es posible dar una gereralización del mapeo de Gray equivalente a la introducida en [9]. En la sección tres de este Capítulo se dará una caracterización de los códigos *hpo*-cíclico y también se mostrará que la imagen generalizada de un código *hpo*-cíclico es un código binario cuasi-cíclico de distancia invariante (no necesariamente lineal). En la sección cuatro se estudiarán los códigos *hpo*-cíclico de longitud impar y finalmente en la sección cinco se discutirán algunos códigos lineales *hpo*-cíclicos.

5.2 Notación, Definiciones y Preliminares

Como en Capítulos anteriores $\mathbb{F}_2$ y $\mathbb{F}_2^n$ denotarán, respectivamente, al campo de los números binarios y al espacio vectorial sobre $\mathbb{F}_2$ de dimensión n. Por otro lado, para un entero positivo k denotaremos como $\mathbb{Z}_{2^{k+1}}$ al anillo de enteros módulo 2^{k+1} y como $\mathbb{Z}_{2^{k+1}}^n$ al módulo de todos los vectores de longitud n con entradas en $\mathbb{Z}_{2^{k+1}}$. Para evitar ambigüedades, se usará a lo largo de este Capítulo el símbolo "$\oplus$" para denotar la suma en $\mathbb{F}_2$, $\mathbb{F}_2[x]$ y $\mathbb{F}_2^n$, mientras que la suma en $\mathbb{Z}_{2^{k+1}}$, $\mathbb{Z}_{2^{k+1}}[x]$ y $\mathbb{Z}_{2^{k+1}}^n$ se denotará con "$+$".

Sea σ el *shift* (corrimiento circular) usual sobre $\mathbb{F}_2^{2n}$ y $\mathbb{Z}_{2^{k+1}}^n$. Para cualquier entero positivo s, sea σ_s el *cuasi-shift* sobre $(\mathbb{F}_2^{2n})^s$ y sea ν_s el *cuasi-negashift* sobre $(\mathbb{Z}_4^n)^s$ dados por:

$$\begin{aligned} \sigma_s(\tilde{a}^{(1)}|\tilde{a}^{(2)}|...|\tilde{a}^{(s)}) &= \sigma(\tilde{a}^{(1)})|\sigma(\tilde{a}^{(2)})|...|\sigma(\tilde{a}^{(s)}) , \\ \nu_s(a^{(1)}|a^{(2)}|...|a^{(s)}) &= \nu(a^{(1)})|\nu(a^{(2)})|...|\nu(a^{(s)}) , \end{aligned}$$

donde "|" denota la concatenación usual entre vectores y $\tilde{a}^{(i)} \in \mathbb{F}_2^{2n}$, $a^{(i)} \in \mathbb{Z}_4^n$, para $i = 1, ..., s$.

Un *código cíclico* de longitud $2n$ (respect. n) sobre $\mathbb{F}_2$ (respect. $\mathbb{Z}_{2^{k+1}}$), es un subconjunto C de $\mathbb{F}_2^{2n}$ (respect. $\mathbb{Z}_{2^{k+1}}^n$) tal que $\sigma(C) = C$. Un *código*

cuasi-cíclico de orden s y longitud $2ns$ sobre $\mathbb{F}_2$ es un subconjunto C de $(\mathbb{F}_2^{2n})^s$ tal que $\sigma_s(C) = C$. Equivalentemente, un *código cuasi-negacíclico* de orden s y longitud $2n$ sobre $\mathbb{Z}_4$ es un subconjunto C de $(\mathbb{Z}_4^n)^s$, tal que $\nu_s(C) = C$. Claramente, cuando $s = 1$, los conceptos de código cíclico (respect. negacíclico) y cuasi-cíclico (respect. cuasi-negacíclico) coinciden.

Ahora se recordará la definición de *mapeo de Gray* [16, 44], ϕ, que va de $\mathbb{Z}_4^n$ a $\mathbb{F}_2^{2n}$: para cualquier $Z = (z_1, z_2, ..., z_n) \in \mathbb{Z}_4^n$,

$$\phi(Z) = (r_1(z_1), ..., r_1(z_n), r_1(z_1) \oplus r_0(z_1), ..., r_1(z_n) \oplus r_0(z_n)) ,$$

donde r_1 y r_0 son dos mapeos de $\mathbb{Z}_4$ a $\mathbb{F}_2$ tales que, si $z \in \mathbb{Z}_4$ entonces la *expansión 2-ádica* de z es $z = r_0(z) + 2r_1(z)$. De manera similar, definimos $k+1$ mapeos r_i, $i = 0, 1, ..., k$, de $\mathbb{Z}_{2^{k+1}}$ a $\mathbb{F}_2$, los cuales estarán dados en términos de la expansión 2-ádica de cualquier elemento $a \in \mathbb{Z}_{2^{k+1}}$, esto es: $a = r_0(a) + 2r_1(a) + ... + 2^k r_k(a)$. Usando esta expansión 2-ádica se introduce ahora la operación "$\odot$" sobre $\mathbb{Z}_{2^{k+1}}$ como sigue: si $a = r_0(a) + ... + 2^k r_k(a), b = r_0(b) + ... + 2^k r_k(b) \in \mathbb{Z}_{2^{k+1}}$, entonces

$$a \odot b = (r_0(a)r_0(b)) + 2(r_1(a)r_1(b)) + ... + 2^k(r_k(a)r_k(b)) .$$

Tal operación se extiende de manera natural a $\mathbb{Z}_{2^{k+1}}^n$ como sigue: si $A = (a_0, ..., a_{n-1}), B = (b_0, ..., b_{n-1}) \in \mathbb{Z}_{2^{k+1}}^n$ entonces definimos $A \odot B = (a_0 \odot b_0, ..., a_{n-1} \odot b_{n-1})$.

Por otro lado, para $k \geq 2$, definimos ahora el siguiente mapeo ρ_k de $\mathbb{Z}_{2^{k+1}}$ a $\mathbb{F}_2^{k-1}$ dado por: $\rho_k(a) = (r_{k-1}(a), ..., r_2(a), r_1(a))$. Para toda $i \in \{0, 1, ..., 2^{k-1} - 1\}$, sea $\alpha_i^k \in \mathbb{F}_2^{k-1}$ la expresión binaria de i usando $k-1$ bits. Ahora bien, por medio de ρ_k y α_i^k, se definen las siguientes funciones $\varphi_i^k : \mathbb{Z}_{2^{k+1}} \to \mathbb{Z}_4$:

$$\varphi_i^k(a) = 2(r_k(a) \oplus (\rho_k(a) \cdot \alpha_i^k)) + r_0(a) , \quad \forall\, i = 0, 1, ..., 2^{k-1} - 1 , \quad (5.1)$$

donde "$\cdot$" denota el producto punto usual en $\mathbb{F}_2^{k-1}$. La acción de las funciones φ_i^k se extiende a $\mathbb{Z}_{2^{k+1}}^n$ como sigue: si $A = (a_0, a_1, ..., a_{n-1}) \in \mathbb{Z}_{2^{k+1}}^n$ entonces $\varphi_i^k(A) = (\varphi_i^k(a_0), \varphi_i^k(a_1), ..., \varphi_i^k(a_{n-1}))$. De esta forma, se introduce el siguiente mapeo $\varphi^k : \mathbb{Z}_{2^{k+1}}^n \to \mathbb{Z}_4^{2^{k-1}n}$:

$$\varphi^k(A) = (\varphi_0^k(A), \varphi_1^k(A), ..., \varphi_{2^{k-1}-1}^k(A)) , \quad \forall\, A \in \mathbb{Z}_{2^{k+1}}^n .$$

Por completes se define $\varphi^1 : \mathbb{Z}_4^n \to \mathbb{Z}_4^n$ como el mapeo identidad, esto es $\varphi^1(A) = A$. A través del mapeo φ^k, es ahora posible presentar una definición

para el *mapeo generalizado de Gray* $\Phi : \mathbb{Z}_{2^{k+1}}^n \to \mathbb{F}_2^{2^k n}$, equivalente al dado en [9]:

$$\Phi(A) = (\phi\varphi_0^k(A), \phi\varphi_1^k(A), ..., \phi\varphi_{2^{k-1}-1}^k(A)) , \quad \forall\, A \in \mathbb{Z}_{2^{k+1}}^n . \tag{5.2}$$

Observación 5.1 *Una manera alternativa de definir el mapeo generalizado de Gray es:* $\Phi(A) = \phi\varphi^k(A)$. *La diferencia fundamental entre esta definición y la dada por la ecuación (5.2) es la forma en la cual se ordenan los bits de la imagen del mapeo generalizado de Gray. De cualquier manera, observe que tanto* Φ *como* φ^k *son mapeos inyectivos.*

El *peso de Lee*, p_L, de $0, 1, 2, 3 \in \mathbb{Z}_4$ es respectivamente $0, 1, 2, 1$, y el peso de Lee $p_L(A)$ de un vector A en $\mathbb{Z}_4^n$ es la suma racional de los pesos de Lee de sus componentes. La *distancia de Lee*, d_L, se define como $d_L(A, B) = p_L(A - B)$ para toda pareja $A, B \in \mathbb{Z}_4^n$. Para $k \geq 1$, definimos ahora el *peso homogéneo*, p_{hom} (ver tambien [12, 18]) sobre $\mathbb{Z}_{2^{k+1}}$ como:

$$p_{\text{hom}}(a) = \begin{cases} 0 & \text{si } a = 0 \\ 2^k & \text{si } a = 2^k \\ 2^{k-1} & \text{de otro modo} \end{cases} , \quad \forall\, a \in \mathbb{Z}_{2^{k+1}} .$$

Nuevamente, para $A \in \mathbb{Z}_{2^{k+1}}^n$, el valor $p_{\text{hom}}(A)$ estará dado como la suma racional de los pesos homogéneos de sus componentes, y la *métrica homogénea*, δ_{hom}, estará dada por $\delta_{\text{hom}}(A, B) = p_{\text{hom}}(A - B)$ para toda pareja $A, B \in \mathbb{Z}_{2^{k+1}}^n$.

Para $n = 1$, $k \geq 2$, y $a \in \mathbb{Z}_{2^{k+1}}$, la función $f(i) = r_k(a) \oplus (\rho_k(a) \cdot \alpha_i^k)$ en la ecuación (5.1) es una función Booleana afín en $k - 1$ variables. Por tanto, el vector $\varphi^k(a) = (\varphi_0^k(a), \varphi_1^k(a), ..., \varphi_{2^{k-1}-1}^k(a))$ tiene alguna de las siguientes formas:

- el vector nulo si $a = 0$,
- todas sus entradas son iguales a 2 si $a = 2^k$,
- la mitad de sus entradas son iguales a 2 y la otra mitad iguales a 0 si a es un número par diferente de 0 y 2^k,
- todas sus entradas tienen valor de 1 ó 3 si a es impar.

La conclusión es, en cualquier caso, que $p_{\text{hom}}(a) = p_L(\varphi^k(a))$. De esta manera se tiene probado la siguiente:

Proposición 5.2 *La función φ^k es una isometría de $\mathbb{Z}_{2^{k+1}}^n$ a $\mathbb{Z}_4^{2^{k-1}n}$, esto es,*

$$\delta_{\text{hom}}(A,B) = d_L(\varphi^k(A),\varphi^k(B)) , \quad \forall\, A,B \in \mathbb{Z}_{2^{k+1}}^n .$$

La distancia binaria de Hamming, d_H, sobre $\mathbb{F}_2^{2n}$, es empleada en [16] para probar que el mapeo de Gray, ϕ, de $\mathbb{Z}_4^n$ a $\mathbb{F}_2^{2n}$ es una isometría. Por tanto, como una consecuencia de este hecho y la Proposición previa se tiene, como en [9], el siguiente:

Corolario 5.3 *El mapeo generalizado de Gray, Φ, es una isometría de $(\mathbb{Z}_{2^{k+1}}^n, \delta_{\text{hom}})$ a $(\mathbb{F}_2^{2^k n}, d_H)$.*

En [42] se estudia con más detalle la isometría φ^k y dentro de algunas de sus propiedades, las siguientes dos serán de utilidad en este Capítulo.

Proposición 5.4 *Sean $A = (a_0,...,a_{n-1}), B \in \mathbb{Z}_{2^{k+1}}^n$. Entonces*

$$\varphi^k(2^k A + B) = \bar{1} \otimes 2\bar{r}_0 + \varphi^k(B) ,$$

donde $\bar{1}$ es el vector de sólo unos de longitud 2^{k-1}, "$\otimes$" es el producto de Kronecker [23] y $\bar{r}_0$ es un vector binario de longitud n, el cual tiene un uno en su i-ésima entrada si y sólo si a_{i-1} en impar, para $i = 1,...,n$.

Teorema 5.5 *Las imágenes binaria y cuaternaria $\Phi(C)$ y $\varphi^k(C)$ de un código lineal C sobre $\mathbb{Z}_{2^{k+1}}$ son ambas lineales si y sólo si*

$$\forall\, A,B \in C, \text{ entonces } (2^k-2)(A \odot B) \text{ y } 2^k(A \odot B) \text{ están en } C .$$

5.3 Códigos *Hpo*-cíclicos y Negacíclicos

En esta sección se presenta una caracterización de los códigos *hpo*-cíclicos en términos de sus imágenes bajo la isometría φ^k. Además se presenta también un resultado similar al Teorema 3.5 de [44].

Proposición 5.6 *Sean r_i, $i = 0,1,...,k$, los mapeos definidos anteriormente y sea $\lambda = 2^k+1$ una unidad de $\mathbb{Z}_{2^{k+1}}$. Entonces, para toda $a \in \mathbb{Z}_{2^{k+1}}$:*

$$r_i(\lambda a) = \begin{cases} r_i(a) & si\ \ i = 0,1,...,k-1 \\ r_k(a) \oplus r_0(a) & si\ \ i = k \end{cases} .$$

Demostración: $\lambda a = (2^k+1)(\sum_{i=0}^{k} 2^i r_i(a)) = \sum_{i=0}^{k-1} 2^i r_i(a) + 2^k(r_0(a) \oplus r_k(a))$. □

Proposición 5.7 *Sean φ_i^k, $i = 0, 1, ..., 2^{k-1}-1$, los mapeos definidos anteriormente y sea $\lambda = 2^k + 1$. Entonces, para toda $a \in \mathbb{Z}_{2^{k+1}}$:*

$$\varphi_i^k(\lambda a) = -\varphi_i^k(a) \,. \tag{5.3}$$

Demostración: La demostración es una consecuencia inmediata de la definición de los mapeos φ_i^k en la ecuación (5.1) y Proposición 5.6. □

El siguiente resultado constituye una caracterización de los códigos *hpo*-cíclicos en términos de sus imágenes bajo la isometría φ^k.

Teorema 5.8 *Sea C un código sobre $\mathbb{Z}_{2^{k+1}}$ de longitud n. Entonces, C es un código hpo-cíclico si y sólo si $\varphi^k(C)$ es un código cuasi-negacíclico de orden 2^{k-1} y longitud $2^{k-1}n$.*

Demostración: Si $\lambda = 2^k + 1 \in \mathbb{Z}_{2^{k+1}}$, se sigue de la Proposición 5.7 que para todo mapeo φ_i^k se cumple que:

$$\varphi_i^k(\nu_\lambda(A)) = \nu\varphi_i^k(A) \,, \quad \forall\, A \in \mathbb{Z}_{2^{k+1}}^n \,. \tag{5.4}$$

De esta manera, si C un código *hpo*-cíclico en $\mathbb{Z}_{2^{k+1}}^n$ entonces $\nu_\lambda(C) = C$, y por tanto $\varphi_i^k(\nu_\lambda(C)) = \varphi_i^k(C) = \nu\varphi_i^k(C)$. Inversamente, si $\varphi_i^k(C) = \nu\varphi_i^k(C)$, para toda $i = 0, ..., 2^{k-1}-1$, entonces de la ecuación (5.4) se sigue que $\varphi_i^k(\nu_\lambda(C)) = \varphi_i^k(C)$, y dado que φ^k es un mapeo inyectivo, se tiene que $\nu_\lambda(C) = C$. □

Observe que cuando $k = 1$ en el Teorema anterior entonces los conceptos de *hpo*-ciclicidad y negaciclicidad coinciden. De esta manera, este resultado muestra que en verdad el concepto de código *hpo*-cíclico es una generalización del concepto de código negacíclico, según este último concepto fuera introducido en [44]. El siguiente resultado se encuetra también en [44]:

Proposición 5.9 *Sea ν el negashift sobre $\mathbb{Z}_4^n$, σ el shift sobre $\mathbb{F}_2^{2n}$, y si ϕ es el mapeo de Gray de $\mathbb{Z}_4^n$ a $\mathbb{F}_2^{2n}$, entonces*

$$\phi\nu = \sigma\phi \,.$$

Como una consecuencia de la ecuación (5.4) y la Proposición previa, se tiene el siguiente resultado, el cual constituye una generalización de la Proposición 5.9.

Proposición 5.10 *Para* $\lambda = 2^k + 1 \in \mathbb{Z}_{2^{k+1}}$ *entonces*

$$\Phi\nu_\lambda = \sigma_{2^{k-1}}\Phi\,. \tag{5.5}$$

Demostración: Sea $A \in \mathbb{Z}_{2^{k+1}}^n$, entonces

$$\begin{aligned}\Phi(\nu_\lambda(A)) &= (\phi\varphi_0^k(\nu_\lambda(A)), \phi\varphi_1^k(\nu_\lambda(A)), ..., \phi\varphi_{2^{k-1}-1}^k(\nu_\lambda(A)))\\ &= (\phi\nu\varphi_0^k(A), \phi\nu\varphi_1^k(A), ..., \phi\nu\varphi_{2^{k-1}-1}^k(A))\\ &= (\sigma\phi\varphi_0^k(A), \sigma\phi\varphi_1^k(A), ..., \sigma\phi\varphi_{2^{k-1}-1}^k(A)) = \sigma_{2^{k-1}}(\Phi(A))\,. \quad \square\end{aligned}$$

La generalización correspondiente al Teorema 3.5 de [44] es como sigue:

Teorema 5.11 *La imagen bajo el mapeo generalizado de Gray de un código hpo-cíclico sobre* $\mathbb{Z}_{2^{k+1}}$ *de longitud* n*, es un código binario cuasi-cíclico de distancia invariante (no necesariamente lineal) de orden* 2^{k-1} *y longitud* $2^k n$.

Demostración: Es directa del Corolario 5.3, Teorema 5.8 y ecuación (5.5). $\square$

5.4 Códigos *Hpo*-cíclicos de longitud impar

El resultado principal de esta sección es la generalización del Corolario 3.8 de [44].

Proposición 5.12 *Sea* n *un entero positivo impar,* $\lambda = 2^k + 1 \in \mathbb{Z}_{2^{k+1}}$ *y sea* $\tilde{\mu}_\lambda$ *la permutación sobre* $\mathbb{Z}_{2^{k+1}}^n$ *dada por:*

$$\tilde{\mu}_\lambda(a_0, a_1, ..., a_i, ..., a_{n-1}) = (a_0, \lambda a_1, ..., \lambda^i a_i, ..., \lambda^{n-1} a_{n-1})\,.$$

Entonces $D \subseteq \mathbb{Z}_{2^{k+1}}^n$ *es un código lineal cíclico si y sólo si* $\tilde{\mu}_\lambda(D)$ *es un código lineal hpo-cíclico.*

Demostración: Debido a que λ es una unidad de $\mathbb{Z}_{2^{k+1}}$ y $\tilde{\mu}_\lambda$ es lineal, entonces D es lineal si y sólo si $\tilde{\mu}_\lambda(D)$ es también lineal. Dado que n es impar y $\lambda^2 = 1$, entonces $\tilde{\mu}_\lambda(\lambda\sigma(A)) = \nu_\lambda(\tilde{\mu}_\lambda(A))$ para toda $A \in D$. De esta manera, $\sigma(A) \in D \Leftrightarrow \lambda\sigma(A) \in D \Leftrightarrow \tilde{\mu}_\lambda(\lambda\sigma(A)) = \nu_\lambda(\tilde{\mu}_\lambda(A)) \in \tilde{\mu}_\lambda(D)$. $\square$

Ahora bien, como $-1 = 3$ en $\mathbb{Z}_4$ y $\lambda^2 = 1$ en $\mathbb{Z}_{2^k+1}$, entonces una relación entre la función $\tilde{\mu}_\lambda$ y las funciones φ_i^k, equivalente a la que se muestra en la ecuación (5.4), es:

$$\varphi_i^k(\tilde{\mu}_\lambda(A)) = \tilde{\mu}_3\varphi_i^k(A)\ , \quad \forall\, A \in \mathbb{Z}_{2^{k+1}}^n\ . \tag{5.6}$$

La siguiente definición presenta la *permutación de Nechaev* como en [16] (ver también [44]):

Definición 5.13 *Sea n un entero positivo impar y sea τ la permutación sobre $\{0, 1, ..., 2n-1\}$ dada por:*

$$\tau = (1, n+1)(3, n+3)\cdots(2i+1, n+2i+1)\cdots(n-2, 2n-2)\ .$$

Entonces, la permutación de Nechaev, π, sobre $\mathbb{F}_2^{2n}$ se define como:

$$\pi(a_0, a_1, ..., a_{2n-1}) = (a_{\tau(0)}, a_{\tau(1)}, ..., a_{\tau(2n-1)})\ .$$

La siguente extensión de la permutación de Nechaev, será de utilidad para la obtención del resultado principal de esta sección.

Definición 5.14 *Sea n un entero positivo impar y para todo entero positivo s, definimos la extensión de la permutación de Nechaev, π_s, sobre $(\mathbb{F}_2^{2n})^s$ como:*

$$\pi_s(a^{(1)}|a^{(2)}|...|a^{(s)}) = \pi(a^{(1)})|\pi(a^{(2)})|...|\pi(a^{(s)})\ ,$$

donde $a^{(i)} \in \mathbb{F}_2^{2n}$.

Para $k = 1$, $\lambda = 3 \in \mathbb{Z}_4$ y n un entero positivo impar, en [44] (Proposición 3.7) se prueba la siguiente relación:

$$\phi\tilde{\mu}_3 = \pi\phi\ .$$

Como una consecuencia de la ecuación (5.6), Definición 5.14 y la relación previa, se tiene:

Proposición 5.15 *Sea n un entero positivo impar y $\lambda = 2^k + 1 \in \mathbb{Z}_{2^{k+1}}$. Entonces*

$$\Phi\tilde{\mu}_\lambda = \pi_{2^{k-1}}\Phi\ . \tag{5.7}$$

Demostración: Sea $A \in \mathbb{Z}_{2^k+1}^n$, entonces

$$\begin{aligned}\Phi(\tilde{\mu}_\lambda(A)) &= (\phi\varphi_0^k(\tilde{\mu}_\lambda(A)), \phi\varphi_1^k(\tilde{\mu}_\lambda(A)), ..., \phi\varphi_{2^{k-1}-1}^k(\tilde{\mu}_\lambda(A))) \\ &= (\phi\tilde{\mu}_3\varphi_0^k(A), \phi\tilde{\mu}_3\varphi_1^k(A), ..., \phi\tilde{\mu}_3\varphi_{2^{k-1}-1}^k(A)) \\ &= (\pi\phi\varphi_0^k(A), \pi\phi\varphi_1^k(A), ..., \pi\phi\varphi_{2^{k-1}-1}^k(A)) = \pi_{2^{k-1}}(\Phi(A)) \ . \quad \square\end{aligned}$$

La generalización natural del Corolario 3.8 de [44] es como sigue:

Corolario 5.16 *Sea n un entero positivo impar y para $k \geq 1$ sea $\pi_{2^{k-1}}$ la extensión de la permutación de Nechaev. Si Γ es la imagen bajo el mapeo generalizado de Gray de un código lineal cíclico sobre $\mathbb{Z}_{2^k+1}$, entonces $\pi_{2^{k-1}}(\Gamma)$ es un código cuasi-cíclico.*

Demostración: Se deriva inmediatamente del Teorema 5.11, Proposición 5.12 y ecuación (5.7). $\square$

5.5 Códigos Lineales *Hpo*-cíclicos

En esta sección será útil representar a los elementos en $\mathbb{Z}_{2^k+1}^n$ como polinomios en $\mathbb{Z}_{2^k+1}[x]$. De esta manera, la *representación polinomial de* $\mathbb{Z}_{2^k+1}^n$ es a través del mapeo $\mathcal{P}$ de $\mathbb{Z}_{2^k+1}^n$ a $\mathbb{Z}_{2^k+1}[x]$ tal que:

$$\mathcal{P}(a_0, a_1, ..., a_{n-1}) = \sum_{j=0}^{n-1} a_j x^j \ .$$

Proposición 5.17 *Sea $\lambda = 2^k + 1 \in \mathbb{Z}_{2^k+1}$. Entonces un subconjunto C de $\mathbb{Z}_{2^k+1}^n$ es un código lineal hpo-cíclico de longitud n si y sólo si su representación polinomial es un ideal en el anillo cociente $\mathbb{Z}_{2^k+1}[x]/(x^n - \lambda)$.*

Demostración: La demostración es similar a la que se da para el caso de códigos lineales cíclicos sobre un campo finito (ver por ejemplo [23]). $\square$

Proposición 5.18 *Sea n un entero positivo impar, $\lambda = 2^k + 1 \in \mathbb{Z}_{2^k+1}$ y sea μ_λ el mapeo de $\mathbb{Z}_{2^k+1}[x]/(x^n - 1)$ a $\mathbb{Z}_{2^k+1}[x]/(x^n - \lambda)$ dado por:*

$$\mu_\lambda(A(x)) = A(\lambda x) \ ,$$

entonces μ_λ es un isomorfismo de anillos.

Demostración: La demostración es similar a la que se da en [44] para el mapeo μ en la Proposición 2.3. □

Una consecuencia inmediata del resultado anterior es el siguiente:

Corolario 5.19 *Sea I un subconjunto del anillo cociente $\mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$, entonces I es un ideal de $\mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$ si y sólo si $\mu_\lambda(I)$ es un ideal de $\mathbb{Z}_{2^{k+1}}[x]/(x^n-\lambda)$.*

La acción de los mapeos φ_i^k puede ahora ser extendida a $\mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$ como sigue: si $A(x)=\sum_{j=0}^{n-1} a_j x^j \in \mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$ entonces

$$\varphi_i^k(A(x)) = \sum_{j=0}^{n-1} \varphi_i^k(a_j)x^j \qquad i=0,1,...,2^{k-1}-1\ , \tag{5.8}$$

y por tanto la acción del mapeo φ^k sobre $\mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$ estará dada por:

$$\varphi^k(A(x)) = (\varphi_0^k(A(x)),...,\varphi_{2^{k-1}-1}^k(A(x)))\ .$$

Ahora, denotaremos como $(\mathbb{F}_2[x]/(x^n-1))^m$ (respect. $(\mathbb{Z}_{2^{k+1}}[x]/(x^n-1))^m$) al conjunto de vectores de longitud m cuyas entradas son polinomios en el anillo $\mathbb{F}_2[x]/(x^n-1)$ (respect. $\mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$). Observe que con esta notación, la isometría φ^k puede pensarse como un mapeo de $\mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$ a $(\mathbb{Z}_4[x]/(x^n-1))^{2^{k-1}}$. La acción del producto de Kronecker, es también extendida, para cualquier entero positivo m, de la siguiente manera: si $\bar{a}(x)=(a_1(x),...,a_m(x)) \in (\mathbb{F}_2[x]/(x^n-1))^m$ y $b(x)\in \mathbb{F}_2[x]/(x^n-1)$, entonces

$$\bar{a}(x)\otimes b(x) = (a_1(x)b(x),...,a_m(x)b(x))\ ,$$

y equivalentemente para $\bar{A}(x)\otimes B(x)$ con $\bar{A}(x)\in(\mathbb{Z}_{2^{k+1}}[x]/(x^n-1))^m$ y $B(x)\in\mathbb{Z}_{2^{k+1}}[x]/(x^n-1)$.

Como una consecuencia de la representación polinomial de $\mathbb{Z}_{2^{k+1}}^n$ y las Proposiciones 5.4 y 5.18 se tiene la siguiente:

Proposición 5.20 *Sea $\lambda = 2^k+1 \in \mathbb{Z}_{2^{k+1}}$ y $A(x)=\sum_{j=0}^{n-1} a_j x^j$, $B(x)\in \mathbb{Z}_{2^{k+1}}[x]/(x^n-\lambda)$, entonces*

$$\varphi^k(2^k A(x)+B(x)) = \bar{1}\otimes 2\tilde{a}(x)+\varphi^k(B(x))\ ,$$

donde $\tilde{a}(x)$ es el polinomio en $\mathbb{F}_2[x]/(x^n-1)$, cuyo coeficiente del monomio de grado j es uno si y sólo si a_j es impar, para $j=0,...,n-1$. El producto $2\tilde{a}(x)$ se realiza en $\mathbb{Z}_4[x]/(x^n-1)$.

El siguiente resultado en [44], describe como son los códigos lineales negacíclicos de longitud n, cuyas imágenes bajo el mapeo de Gray son códigos binarios lineales y cíclicos de longitud $2n$.

Teorema 5.21 *Sea n un entero positivo impar y sea $\tilde{a}(x), \tilde{b}(x)$ en $\mathbb{F}_2[x]$ tales que $x^n - 1 = (x-1)\tilde{a}(x)\tilde{b}(x)$, donde $(x-1), \tilde{a}(x)$ y $\tilde{b}(x)$ son primos relativos a parejas. Sea $a_1(x), b_1(x)$ los levantamientos de Hensel (Hensel lifts) de $\tilde{a}(x)$ y $\tilde{b}(x)$ a $\mathbb{Z}_4[x]$, respectivamente, y definimos $a(x) = a_1(-x)$ y $b(x) = b_1(-x)$. If $\tilde{C}$ es el código binario lineal y cíclico de longitud $2n$ generado por $\tilde{g}(x) = \tilde{a}(x)^2\tilde{b}(x)$, entonces $\tilde{C}$ es la imagen bajo el mapeo de Gray de un código lineal negacíclico de longitud n generado por $g(x) = a(x)(b(x)+2)$. Más aún, si $u(x) \in \mathbb{Z}_4[x]/(x^n+1)$ y $\tilde{u}(x) \in \mathbb{F}_2[x]$ son tales que el coeficiente del monomio de grado j en $\tilde{u}(x)$ es uno si y sólo si el coeficiente del monomio de grado j en $\tilde{u}(x)$ es impar, entonces*

$$\phi(u(x)g(x)) = [f(u(-1))\tilde{b}(x) \oplus \tilde{u}(x)(x+1)]\tilde{g}(x)\,,$$

donde f es la función de $\mathbb{Z}_4$ a $\mathbb{F}_2[x]$ dada por:

$$f(a) = \begin{cases} 0 & si\ \ a = 0 \\ x & si\ \ a = 1 \\ 1+x & si\ \ a = 2 \\ 1 & si\ \ a = 3 \end{cases}.$$

La función f del Teorema previo puede extenderse a $\mathbb{Z}_4^m$ de la siguiente manera: si $A = (a_1, ..., a_m) \in \mathbb{Z}_4^m$, entonces $f(A) = (f(a_1), ..., f(a_m))$.

Los códigos lineales cíclicos sobre $\mathbb{Z}_{2^k+1}$ han sido ampliamente estudiados en [28, 20]. Ahora bien, considerando las Proposiciones 5.12, 5.17, 5.18 y el Corolario 5.19, se tiene que si $\tilde{a}(x)$ y $\tilde{b}(x)$ son como en el Teorema previo y si $A_1(x), B_1(x)$ son los levantamientos de Hensel de $\tilde{a}(x)$ y $\tilde{b}(x)$ a $\mathbb{Z}_{2^k+1}[x]$, respectivamente, entonces para el caso de código lineal *hpo-cíclico* generado por polinomios de la forma $A_1(\lambda x)(B_1(\lambda x) + 2^k)$, se tiene el siguiente resultado:

Teorema 5.22 *Con la notación previa sea $A(x) = A_1(\lambda x)$ y $B(x) = B_1(\lambda x)$. Si C es el código lineal hpo-cíclico generado por $G(x) = A(x)(B(x) + 2^k)$, entonces la imagen $\varphi^k(C)$ es un código cuaternario lineal y cuasi-negacíclico de orden 2^{k-1} y longitud $2^{k-1}n$. Por otro lado, $\Phi(C)$ es un código binario lineal y cuasi-cíclico de orden 2^{k-1} y longitud $2^k n$. Más aún, si $U(x) \in \mathbb{Z}_{2^k+1}[x]/(x^n - \lambda)$ y $\tilde{u}(x) \in \mathbb{F}_2[x]$ son tales que el coeficiente del*

monomio de grado j en $\tilde{u}(x)$ es uno si y sólo si el coeficiente del monomio de grado j en $U(x)$ es impar, entonces

$$\begin{aligned} \varphi^k(U(x)G(x)) &= \varphi^k(U(\lambda)) \otimes [a(x)b(x)] + \bar{1} \otimes [2\tilde{u}(x)\tilde{a}(x)] \,, \\ \Phi(U(x)G(x)) &= [f(\varphi^k(U(\lambda))) \otimes \tilde{b}(x) \oplus \bar{1} \otimes (\tilde{u}(x)(x+1))] \otimes \tilde{g}(x) \,. \end{aligned} \tag{5.9}$$

Demostración: La cuasi-negaciclicidad del código $\varphi^k(C)$ se sigue del Teorema 5.8, y la cuasi-ciclicidad del código $\Phi(C)$ se obtiene del Teorema 5.11. Dado que $U(x)A(x)B(x) = U(\lambda)\sum_{j=0}^{n-1}\lambda^j x^j$ en $\mathbb{Z}_{2^{k+1}}[x]/(x^n - \lambda)$ y $2(\lambda v \odot \lambda w) = 2\lambda(v \odot w)$, para toda pareja $v, w \in \mathbb{Z}_{2^{k+1}}$, entonces la linealidad de estos códigos se sigue del Teorema 5.5.

Ahora bien, a través de la Proposición 5.20 se tiene que $\varphi^k(U(x)G(x)) = \varphi^k(U(x)A(x)B(x)) + \bar{1} \otimes [2\tilde{u}(x)\tilde{a}(x)]$, y de las ecuaciones (5.3) y (5.8) se sigue que, $\varphi_i^k(U(x)A(x)B(x)) = \varphi_i^k(U(\lambda))\sum_{j=0}^{n-1}(-1)^j x^j = \varphi_i^k(U(\lambda))a(x)b(x)$, para toda $i = 0, ..., 2^{k-1} - 1$, con lo cual se demuestra la primera ecuación en (5.9).

Observe que para cada i, $\varphi_i^k(U(\lambda))$, es un número impar si y sólo si $\tilde{u}(1)$ es también un número impar, y por lo tanto debe existir $u_i(x) \in \mathbb{Z}_4[x]/(x^n + 1)$ tal que $u_i(-1) = \varphi_i^k(U(\lambda))$ y el coeficiente del monomio de grado j en $\tilde{u}(x)$ es impar si y sólo si el coeficiente del monomio de grado j en $\tilde{u}(x)$ es uno. De esta forma, cada polinomio $\varphi_i^k(U(\lambda))a(x)b(x) + 2\tilde{u}(x)\tilde{a}(x)$ en las entradas de $\varphi^k(U(x)G(x))$ puede re-expresarse como $u_i(x)g(x)$ en el anillo cociente $\mathbb{Z}_4[x]/(x^n + 1)$. Así, la parte final de la demostración se sigue de aplicar el Teorema 5.21 a estos polinomios. □

5.5.1 Un Ejemplo

Si $k = 2$, $n = 7$ y $x^7 - 1 = (x-1)\tilde{a}(x)\tilde{b}(x)$, con $\tilde{a}(x) = x^3 + x + 1$ y $\tilde{b}(x) = x^3 + x^2 + 1$, entonces

$$\begin{aligned} a(x) &= 3x^3 + 2x^2 + 3x + 3 \,, \\ b(x) &= 3x^3 + 3x^2 + 2x + 3 \,, \\ A(x) &= 5x^3 + 6x^2 + x + 7 \,, \\ B(x) &= 5x^3 + 3x^2 + 2x + 7 \,, \\ \tilde{g}(x) &= x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \,, \\ g(x) &= x^6 + 3x^5 + x^4 + x^3 + x^2 + x + 3 \,, \\ G(x) &= x^6 + 5x^5 + x^4 + x^3 + x^2 + x + 5 \,. \end{aligned}$$

Capítulo 6

Conclusiones

Los dos objetivos fundamentales de este trabajo de Tesis han sido, por un lado, el estudio del estado del arte de buena parte de las áreas de Teoría de Códigos y Criptografía, y por el otro lado, el hacer investigaciones tendientes a obtener contribuciones originales en tales áreas. Diferentes fueron los temas de investigación que se abordaron, sin embargo, en tres de estos temas ha sido donde se han obtenido los resultados más relevantes, los cuales se han descrito en los tres Capítulos anteriores. De esta manera, se considera que el presente trabajo no sólo es importante por los resultados que aquí se han expuesto sino también porque este tipo de investigaciones son un claro ejemplo de la clase de problemas que son interesantes a resolver en esta enorme rama de las Matemáticas Aplicadas que es la Teoría de Códigos y Criptografía.

En el Capítulo tres de este trabajo se retomó una transformación que fuera originalmente propuesta en [8]. Tal transformación, como se vió, tiene la propiedad de que al ser aplicada iterativamente sobre una función Booleana de grado mayor a 3, es posible asociarle a ésta otra función Booleana de grado a lo más 3, con la cual mantendrá una sencilla relación entre los pesos de Hamming de ambas funciones. En dicho Capítulo, se estableció que a través de un método de factorización sobre funciones Booleanas, es posible fijar una nueva cota superior sobre el número de iteraciones necesarias para la transformación de cualquier función Booleana de grado mayor o igual a 4 a una función de grado 3. Una posible continuación a esta línea de investigación es estudiar hasta qué punto este método de factorización puede ser útil en el problema de encontrar la distribución de pesos de los códigos de binarios de Reed-Muller de cualquier orden y de cualquier número de variables (ver Capítulo uno). La hipótesis que al respecto se tiene es que

quizá a través de este método sea posible concluir que el problema para la descripción de la distribución de pesos de los códigos binarios de Reed-Muller de orden mayor a 3, es equivalente a la descripción de los de orden 3.

En cuanto a futuros trabajos relacionados con los resultados que se han expuesto en el Capítulo cuatro, es posible quizá continuar con el estudio de la S-cajas en términos de sus tablas de distribución de diferencias. Aún más, quizá a través de la partición introducida en la Sección seis de este Capítulo y con el establecimiento de la nueva cota para la ϵ-robustez sea ahora posible encontrar familias de S-cajas que compartan el mismo valor de ϵ-robustez, el cual podría encontrarse muy cerca de esta nueva cota.

Para el tercer conjunto de resultados que han sido presentados en el quinto Capítulo, se considera de interés el continuar con el estudio de las propiedades de la isometría, φ^k. En ese sentido, se tiene en preparación el trabajo [42], en el cual se reportarán varias generalizaciones de los códigos sobre $\mathbb{Z}_4$ a códigos sobre $\mathbb{Z}_{2^k+1}$. Otra posible investigación de interés sería continuar con el estudio que se iniciara en la Sección 5.5, sobre los códigos lineales *hpo*-cíclicos. En concreto, se cosidera que sería de interés el estudiar las condiciones generales bajo las cuales la imagen bajo φ^k o Φ de un código lineal *hpo*-cíclicos, resulta ser la concatenación de 2^{k-1} códigos lineales y negacíclicos o cíclicos, ya sean estos códigos, cuaternarios o binarios. Más aún, resultaría también de interés el estudiar la manera bajo la cual se relacionan los ideales sobre $\mathbb{Z}_4$ o sobre $\mathbb{F}_2$ y los ideales sobre $\mathbb{Z}_{2^k+1}$, según la acción de las isometrías φ^k y Φ sobre los códigos lineales *hpo*-cíclicos.

Bibliografía

[1] C.M. Adams and S.E. Tavares. "The structured design of cryptographically good S-boxes." *Journal of Cryptology*, Vol. 3(1), pp. 27-41, 1990.

[2] E.F. Assmus Jr. and J.D. Key. *Designs and Their Codes*, Cambridge University Press, 1992.

[3] T. Beth and C. Ding C. "On permutations against differential cryptanalysis." *Advances in Cryptology - EUROCRYPT'93, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 765, pp. 65-76, 1994.

[4] E. Biham and A. Shamir. "Differential Cryptanalysis of FEAL and N-Hash." *In Advances in Cryptology - EUROCRYPT'91, Lecture Notes in Computer Science*, Springer-Verlag, pp. 1-16, 1991.

[5] E. Biham A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* New York: Springer-Verlag, 1993.

[6] E. Biham and A. Shamir. "Differential Cryptanalysis of DES-like cryptosystems." *Journal of Cryptology*, Vol. 4(1), pp. 3-72, 1993.

[7] R.E. Blahut. *Theory and practice of error control codes.* New York: Addison-Wesley, 1983.

[8] C. Carlet. "A Transformation on Boolean Functions, its Consequences on Some Problems Related to Reed-Muller Codes." *Eurocode 90, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 514, pp. 42-50, 1990.

[9] C. Carlet. "$\mathbb{Z}_{2^k}$-Linear Codes." *IEEE Trans. Inform. Theory*, Vol. 44, pp. 1543-1547, 1998.

[10] I. Duursma, C. Rentería and H. Tapia-Recillas. "Reed-Muller Type Codes on Complete Intersections.", *Appl. Algebra in Engng., Comm. and Comp. (AAECC)*, Springer-Verlag, to appear.

[11] R. Forré. "Methods and instruments for designing S-boxes." *Journal of Cryptology*, Vol. 2(3), pp. 115-130, 1990.

[12] M. Greferath, and S. E. Schmidt, "Gray Isometries for Finite Chain Rings and a Nonlinear Ternary (36,3^{12},15) code," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2522-2524, 1999.

[13] N. Gutiérrez-Herrera, H. Tapia-Recillas and G. Vega. "A Parseval type of relation on $\mathbb{Z}_n$.", *Proc. of the World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, USA, Vol. 7, pp. 672-674, 2000.

[14] R.W. Hamming. "Error detecting and error correcting codes." *Bell System Tech. J.*, Vol. 29, pp. 147-160, 1950.

[15] R.W. Hamming. *Coding and Information Theory*. Englewood Cliffs, N.J.: Prentice-Hall, 1980.

[16] A.R. Hammons Jr., P.V. Kumar, A.R. Calderbank, N.J.A. Sloane and P. Solé. "The $\mathbb{Z}_4$-Linearity of Kerdock, Preparata, Goethals, and related codes." *IEEE Trans. Inform. Theory*, Vol. 40, pp. 301-319, 1994.

[17] P. Heijnen, H. van Tilborg, T. Verhoeff and S. Weijs. "Some New Binary, Quasi-Cyclic Codes." *IEEE Trans. Inform. Theory*, Vol. 44, pp. 1994-1996, 1998.

[18] W. Heise, T. Honold, and A. A. Nechaev, "Weighted modules and representations of codes," in *Proc. ACCT 6* (Pskov, Russia, 1998), pp. 123-129.

[19] R. Hill. *A First Course in Coding Theory*. Oxford Applied Mathematics and Computing Science Series, New York: Clarendon Press-Oxford, 1986.

[20] P. Kanwar and S.R. López-Permounth. "Cyclic codes over the integers modulo p^m." *Finite Fields and Their Applications*, Vol. 3, no. 4, pp. 334-352, 1997.

[21] X. Lai and J.L. Massey. "A Proposal for a New Block Encryption Standard.", *Advances in Cryptology - EuroCrypt'90, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 473, pp. 389-404, 1990.

[22] R. Lidl and H. Niederreiter. *Finite Fields*. Encyclopedia of Mathematics and its applications, Vol. 20, Reading, Massachusetts: Addison-Wesley, 1983.

[23] F.J. MacWilliams and N.J. Sloane. *The theory of error-correcting codes*, Amsterdam: North Hollan, 1977.

[24] M. Matsui. "Linear cryptanalysis method for DES cipher." *Advances in Cryptology - EuroCrypt'93, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 765, pp. 386-397, 1994.

[25] D.E. Muller. "Application of boolean algebra to swiching circuit design and error detection." *IEEE Trans. Computers*, Vol. 3, pp. 6-12, 1954.

[26] National Institute of Standards and Technology (NIST). FIPS Publication 46-1: "Data Encryption Standard. January 22", 1988.

[27] K. Nyberg. "Perfect nonlinear S-boxes." *Advances in Cryptology - EuroCrypt'91, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 547, pp. 378-386, 1991.

[28] V. Pless and Z. Qian. "Cyclic Codes and Quadratic Residue Codes over $\mathbb{Z}_4$." *IEEE Trans. Inform. Theory*, Vol. 42, pp. 1594-1600, 1996.

[29] I.S. Reed. "A class of multiple-error-correcting codes and the decoding scheme." *IEEE Trans. Info. Theory*, Vol. 4, pp. 38-49, 1954.

[30] C. Rentería, H. Tapia-Recillas. "Reed-Muller Codes: An Ideal Theory Approach." *Communications in Algebra*, Vol. 25(2), pp. 401-413, 1997.

[31] C. Rentería, H. Tapia-Recillas. "The a-invariant of some Reed-Muller Codes." *Appl. Algebra in Engng., Comm. and Comp. (AAECC)*, Springer-Verlag, Vol. 10(1), pp. 33-40, 1999.

[32] C. Rentería, H. Tapia-Recillas. "Reed-Muller type codes over the Veronese variety." *Proc. of the Int. Conf. on Coding Theory, Cryptography and Related Areas, J. Buchmann, T. Hoholdt, H. Stichtenoth, H. Tapia-Recillas, eds.*, ISBN 3-540-66248-0, Springer-Verlag, pp. 237-243, 2000.

[33] D.E. Robling. *Cryptography and Data Security* New York: Addison-Wesley, 1986.

[34] J. Seberry, X. Zhang and Y. Zheng. "Systematic generation of cryptographically robust S-boxes." In *Proceedings of the firts ACM Conference on Computer and Communications Security*, pp. 172-182. The Association for Computing Machinery, New York, 1993.

[35] J. Seberry, X. Zhang and Y. Zheng. "Relationships among nonlinearity criteria." *Advances in Cryptology - EuroCrypt'95, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 950, pp. 376-388, 1995.

[36] A.G. Shanbag, P. V. Kumar, and T. Helleseth. "Improved binary codes and sequence families from $\mathbb{Z}_4$-linear codes." *IEEE Trans. Inform. Theory*, Vol. 42, pp. 1582-1587, 1996.

[37] C.E. Shannon. "A mathematical theory of communication." *Bell Syst. Tech. J.*, Vol. 7, pp. 379-423 and 623-656, 1948.

[38] D. Slepian. "A note on two binary signaling alphabets." *IEEE Trans. Information Theory*, Vol. 2, pp. 84-86, 1956.

[39] H. Tapia-Recillas and G. Vega. "Some Results on Regular Mappings." *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes - AAECC-12, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 1255, pp. 321-328, 1997.

[40] H. Tapia-Recillas and G. Vega. "An upper bound on the number of iterations for transforming a Boolean function of degree greater or equal than 4 to a function of degree 3." *Designs, Codes and Cryptography*, to appear.

[41] H. Tapia-Recillas and G. Vega. "A Generalization of Negacyclic Codes." *Proceedings: International Workshop on Coding and Cryptography, WCC 2001*, Paris, France, pp. 519-529, 2001.

[42] G. Vega and H. Tapia-Recillas. "On $\mathbb{Z}_{2^k}$-Linear and Quaternary Codes." *Submitted to IEEE Trans. Inform. Theory.*

[43] A.F. Webster and S.E. Tavares. "On the designs of S-boxes." *In Advances in Cryptology - Crypto'85, Lecture Notes in Computer Science*, Springer-Verlag, Vol. 219, pp. 523-534, 1986.

[44] J. Wolfmann. "Negacyclic and Cyclic Codes Over $\mathbb{Z}_4$." *IEEE Trans. Inform. Theory*, Vol. 45, pp. 2527-2532, 1999.

Índice de Materias