UNIVERSIDAD D SEVILLA

TRABAJO FIN DE MÁSTER

Funciones Bent en Criptografía

Realizado por

Álvaro García Zambrano

Dirigido por

Dr. José Andrés Armario Sampalo

Realizado en el departamento de

Matemática Aplicada I

Firma del alumno

Firma del tutor

Facultad de Matemáticas

Abstract

In this project a certain class of Boolean function is introduced, the so called *bent function*. Since our approach is fundamentally cryptographic, at the beginning of the memory some preliminaries about basic modern Cryptography can be found after the Introduction chapter. Next, the project is divided in two parts. The first one (Chapter 3) describes the main properties and characterizations of Boolean bent function and even some of their uses and several generalizations. The second one (Chapter 4) is about bent function in $\mathbb{F}_p$. Specifically, we focus our study on two properties, nonlinearity and autocorrelation, since both of them measure the resistance against two attacks in specific. Following this, we present an algorithm which generates balanced function from an arbitrary function and keeps its nonlinearity and autocorrelation in a good shape. Finally, a conclusions chapter can be found at the end of this memory which summarize the whole project and comments some open problems.

"*Random numbers should not be generated with a method chosen at random*".

Donald Knuth

Resumen

En este proyecto se introduce una clase de funciones Booleanas con propiedades criptográficas muy interesantes, las conocidas como *funciones bent*. Puesto que el enfoque que damos es fundamentalmente criptográfico, al principio de la memoria, tras la Introducción, pueden encontrarse unos preliminares que presentan ciertas generalidades de la Criptografía moderna. A continuación el trabajo se divide en dos partes. En la primera (Capítulo 3) se describen las funciones bent Booleanas, mostrando varias de sus propiedades y caracterizaciones, así como algunos de sus usos y generalizaciones. En la segunda (Capítulo 4) se traslada la noción bent al caso p-ario, y se analizan dos cualidades en específico, la no linealidad y la autocorrelación, ambas muy importantes de cara a resistir cierto tipo de ataques. Tras ello se presenta un algoritmo capaz de obtener funciones equilibradas que parece preservar en gran medida las propiedades criptográficas de la función a la que se aplica. Por último se incluye un capítulo de conclusiones en el que se resume lo visto hasta el momento y se comentan ciertas líneas de investigación abiertas.

Índice general

1. Introducción

NRR6RFBv5dN3PQIBqyRBPI67+dlKLJwwBWx7GNxqIzgH
oflME/JVR/eWLumQ1tZnYdOL3O22EVTMdw4ZJOuj/TicD
hoL+kSh1VWYt4TLcq2BTlJ+wSZxhjBILWp3oK5f+HUCXC
4E7XE5Rysgg7jq1w3vpicOjRj3+S4aDNMXGUTgHXcCyBwL
gEE+tTKua0mwyvR2+DWS4eM

Desde el mismo momento en que surge la comunicación surgen los secretos, ya sea la cosecha recolectada ese año, en qué aldea se realizará el próximo ataque o una declaración de amor. Estos tres casos tienen en común una cosa: ni al emisor ni al receptor les interesa que el mensaje sea conocido por otros. Si dos personas quieren tener una conversación privada, siempre pueden reunirse en un lugar donde nadie les escuche. Esto, no obstante, exige presencialidad por ambas partes, algo que no siempre se puede garantizar. Comandante y general pueden estar a millas de distancia, y no parece razonable que uno de ellos deba desplazarse para informar al otro del momento de atacar. A causa de una necesidad imperiosa es que surge la escritura. El general puede enviar un heraldo con su mensaje por escrito a sus oficiales, y los amantes pueden enviarse cartas a pesar de estar separados. Sin embargo, ¿qué ocurre si alguien intercepta dicho mensaje? Si un soldado enemigo atrapa al mensajero y lee el mensaje, conocerá los detalles del próximo ataque y su bando podrá actuar en consecuencia. Igualmente, una tercera persona, locamente enamorada de uno de los amantes, podría abrir el buzón antes de que la carta sea enviada y leer su contenido. Más aun, podría incluso modificar alguna partes y hacerse pasar por quien no es, todo con el fin de los amantes se peleen. ¿Qué pueden hacer los tortolitos y el general frente a esto? Afortunadamente, los tres son muy aficionados a los acertijos y a los enigmas, y se les ocurre una forma de escribir el mensaje de forma que sólo lo puedan leer quienes deban leerlo, y nadie más. A nuestro general se le ocurre reordenar el alfabeto, de forma que sólo quien conozca dicha permutación puede ser capaz de conocer el mensaje que porta su heraldo. Por otro lado, los amantes, que en un alarde de originalidad llamaremos Alicia y Bob, son bastante aficionados a las matemáticas. Gracias a sus conocimientos en Geometría Algebraica deciden aplicar el algoritmo de ElGamal para curvas elípticas sobre cuerpos finitos, lo primero que a uno se le ocurre cuando quiere encriptar una carta de amor. Ahí está la palabra clave, *encriptar*.

La criptografía es el conjunto de herramientas que ha desarrollado el ser humano para este propósito del que hablamos: transmitir un mensaje sin que nadie que no esté autorizado pueda leerlo. Desde el cifrado Cesar hasta la máquina enigma, todos tienen eso como objetivo común y fundamental. Y hablando de factores comunes, es bien sabido que estos dos criptosistemas en particular han podido *romperse*. Los ordenadores supusieron un cambio para la criptografía clásica, pues son capaces de barrer en un instante tantas opciones que una sola persona tardaría años. Siempre ha existido esta dualidad en una maratón sin línea de meta. Si un equipo desarrollaba un

nuevo criptosistema, otro se encargaba de buscar posibles brechas y romperlo, lo que obligaba a diseñar uno nuevo que no presentara dichas fisuras, y así sucesivamente. Un ejemplo clásico es el caso del DES. El DES, o *Data Encryption Standard*, es un cifrador a bloques de tipo Feistel que fue diseñado en los años setenta a petición de las NSA para satisfacer sus necesidades de seguridad informática. No es ningún secreto que hoy en día este criptosistema se considera inseguro y, por tanto, ha dejado de utilizarse como estándar. Una de las razones que motivó a Alicia a comenzar a usar criptografía sobre curvas elípticas es que en una ocasión cifró un mensaje utilizando DES, y Eva, que está perdidamente enamorada de Alicia y siente celos de Bob, interceptó el mensaje. Eva también es muy ingeniosa, y fue capaz de averiguar lo que decía, no le costó demasiado deducir que la clave era "*bent*". Fue así que pudo descifrar el mensaje, que se encuentra al inicio de este capítulo, y que reza

```
Hola, Bob. ¿Te parece si nos vemos este jueves a las ocho bajo el puente
 de Triana?

Tengo una sorpresa para ti. Un beso. <3
```

Como veremos en la Sección 3.4, una de las debilidades del DES (además de el reducido tamaño de la clave) es que una de las funciones de sus S-boxes es bastante simple (cuando hablamos de funciones, estamos hablando, por supuesto, de funciones Boolenas $f : \mathbb{F}_2^n \to \mathbb{F}_2$, con $n \in \mathbb{N}$). En concreto presenta una *no linealidad* bastante baja. En Matemáticas la linealidad siempre es una propiedad deseable, pues a partir de ella se suelen desprender otras muchas, lo que da lugar a estructuras ricas. En definitiva, la linealidad es siempre sinónimo de simpleza y de buen comportamiento. Pero en criptografía no queremos nada de esto, todo lo contrario, buscamos complejidad, mal comportamiento. No queremos al chico que se sienta en primera fila, sino el que siempre está al fondo de la clase y no estamos seguro de cuánta atención está prestando. Queremos funciones lo más alejadas posibles de la linealidad, si es que existen. Porque esa es una buena pregunta: ¿*existen funciones lo menos lineales posibles*? Pues, de hecho, no sólo existen (bajo ciertas circunstancias) sino que además tienen un nombre, son las conocidas popularmente como *funciones bent*.

La primera vez que se habló de funciones bent en un artículo fue en 1966, en [Rot76] de Oscar Rothaus, como comenta Dillon en su tesis, cuya versión final no fue publicada hasta diez años más tarde. En esos años, el propio John Dillon publicó un par de artículos, [Dil72] y [Dil74], aunque no tuvieron mucha difusión. No obstante, Rothaus cuenta que algunos otros autores se interesaron por el tema poco después, tales como el reconocido especialista en Teoría de Códigos Lloyd Welch, y Gerry Mitchel, quien hizo grandes aportaciones al área de la Computación. Históricamente se atribuye a Dillon y Rothaus el inicio del estudio de las funciones bent, quienes le dieron un enfoque principalmente criptográfico y muy en relación con el diseño del DES. Sin embargo, Natalia Tokareva cuenta en una nota histórica de [Tok15] que las funciones bent ya habían sido estudiadas por los soviéticos en 1962 bajo el nombre de *funciones mínimas*. De hecho, V.A. Eliseev y O.P. Stepchenkov ya habían probado en 1962 que una cierta cota del grado algebraico de las funciones bent y habían propuesto un análogo de la construcción de McFarland.

Desde un punto de vista criptográfico, las funciones bent tienen dos intereses

principales: su derivada $D_a f(x) := f(x) \oplus f(x \oplus a)$, $x \in \mathbb{F}_2^n$ siempre está equilibrada para cada $a \in \mathbb{F}_2^n$ (lo que de lugar a una autocorrelación nula), y que su no linealidad es máxima. Lo primero tiene una estrecha relación con los ataques diferenciales a los cifradores a bloques [BS91], y lo segundo las hace buenas candidatas para resistir ataques de correlación rápida [MS88] en cifradores a flujo y ataques lineales en cifradores a bloques [Mat94]. No obstante, una gran pega que tienen las funciones bent es que nunca están balanceadas, por lo que se desaconseja su uso directo. Por ejemplo, en el caso de los cifradores a flujo, si se emplea una función que no esté equilibrada, la clave generada resulta ser también no equilibrada, lo que provoca la existencia de un bias estadístico entre el texto claro y el texto cifrado. Sin embargo, hemos dicho que se desaconseja su uso directo, no su uso a secas. Las funciones bent son una herramienta muy útil para construir criptosistemas robustos, veremos algunos ejemplos en la ya mencionada Sección 3.4.

Una vez introducido el objeto en que se centra este texto, pasamos a comentar la estructura del mismo. Comenzamos con unos preliminares que introducen unas nociones básicas acerca de la Criptografía. Entre otros aspectos, describimos los cifradores a flujo y los cifradores a bloques, en los que es posible emplear funciones bent como primitivas. También comentamos distintos tipos de ataques, en concreto nos centramos en el criptoanálisis lineal y en el criptoanálisis diferencial, ya que más adelante haremos referencia a ellos. A continuación pasaríamos al capítulo acerca de las funciones Booleanas, en el que definimos propiamente las funciones bent. En él comentamos varios resultados y caracterizaciones de ellas, así como algunos de sus usos y generalizaciones. A continuación, en el Capítulo 4, nos centramos en una generalización particular de las funciones bent, la extensión a $\mathbb{F}_p$ con p un primo impar. Comentaremos los paralelismos y diferencias existentes entre el caso Booleano y el caso p-ario, centrándonos en particular en la no linealidad y en la autocorrelación, debido a la relación de estos parámetros con el criptoanálisis lineal y diferencial, respectivamente. Para acabar este capítulo presentamos un algoritmo que genera funciones equilibradas con alta no linealidad y baja autocorrelación a partir de funciones bent. Dicho algoritmo fue inspirado por [Sch20]. Nosotros decidimos dar un paso más y extenderlo al caso general sobre $\mathbb{F}_p$. Tras una implementación en el software *Mathematica* haremos varias simulaciones sobre funciones bent conocidas y analizaremos los resultados obtenidos. Por último, en el Capítulo 5 haremos un balance global de la memoria y daremos las conclusiones que se pueden seguir, además de algunas posibles líneas futuras de investigación. Con esto, daremos por terminado nuestro trabajo.

2. Preliminares

Dedicamos este capítulo de preliminares a introducir unas nociones básicas de criptografía. Comenzamos describiendo los criptosistemas de clave simétrica y diferenciándolos de los de clave pública/privada, y pasamos a definir la noción de *seguridad*. A continuación procederemos a detallar el funcionamiento de los cifradores a flujo y los cifradores a bloque, comentando los puntos fuertes y los puntos débiles de cada uno. Por último, hablaremos de ataques, centrándonos en concreto en el criptoanálisis lineal y el criptoanálisis diferencial. Principalmente seguiremos la línea marcada por el primer capítulo de [MOS96], aunque en la sección sobre criptoanálisis recurriremos a bibliografía variada.

2.1. Conceptos elementales en Criptografía

Naturalmente, no podemos empezar de otro modo que no sea dando la definición formal de *criptografía*.

Definición 2.1.1. *La criptografía es el estudio de las herramientas y técnicas matemáticas que tratan la confidencialidad de la información, la integridad de los datos, la autentificación de la identidad, y la verificación del origen de los datos, entre otros aspectos.*

Describimos estas nociones más detalladamente.

1. *Confidencialidad*: esto es evitar que no acceda al contenido de la información nadie que no esté autorizado.
2. *Integridad de los datos*: evitar que la información se vea alterada de cualquier forma por parte de terceros.
3. *Autenticación*: esta función se aplica tanto a las partes involucradas en la comunicación como a la información en sí. Principalmente se divide en estas dos clases: la *autenticación de la identidad* y *la autenticación del origen de los datos*. De forma implícita, la autentificación de los datos también garantiza su integridad.
4. *No repudio*: este servicio se encarga de evitar que una de las partes niegue haber realizado una acción o un compromiso tras haberlo hecho. Por ejemplo durante una compra el comprador acuerda enviar el dinero correspondiente, debe quedar constancia de esa transacción para que el comprador no pueda negar haberla realizado.

Para lograr estos objetivos, quienes se dedican a este área de la Matemática Aplicada diseñan lo que se conocen como *criptosistemas*:

Definición 2.1.2. *Un criptosistema consiste en un conjunto de claves $\mathcal{K}$, un conjunto de funciones $\{E_k : \mathcal{M} \to \mathcal{C} : k \in \mathcal{K}\}$ (llamadas funciones de encriptación) y un*

conjunto $\{D_k : \mathcal{C} \to \mathcal{M} : k \in \mathcal{K}\}$ *(llamadas funciones de desencriptación). Además, debe verificarse que:*

- *Cada* $k \in \mathcal{K}$ *determina de forma unívoca una biyección de* $\mathcal{M}$ *(el conjunto de textos claros) en* $\mathcal{C}$ *(el conjunto de textos cifrados).*
- *Para cada* $k \in \mathcal{K}$ *existe una única* $h \in \mathcal{K}$ *tal que* $D_h = E_k^{-1}$*, es decir,* $D_h(E_k(m)) = m$ *para todo* $m \in \mathcal{M}$*.*

2.2. Clasificación de los criptosistemas

La definición de criptosistema, pese a su concreción, admite una gran variedad de matices, lo que permite la existencia de una gran cantidad de criptosistemas diferentes. No obstante, principalmente se dividen en dos categorías, los de clave simétrica y los de clave pública/privada. Describimos ambos a continuación.

Clave simétrica

Los primeros que surgieron, y que llevan siendo utilizados desde hace siglos. Son aquellos en los que se usa la misma clave para la encriptación y para la desencriptación. Pese al gran problema que resulta la transmisión de la clave, plantean una serie de puntos fuertes.

Ventajas de los criptosistemas de clave simétrica:

- Pueden ser diseñados para tener una alta tasa de transferencia de información.
- Las claves suelen ser relativamente cortas.
- A menudo se usan para construir otros criptosistemas más complejos, como es el caso de los generadores de números pesudoaleatorios o las funciones hash.

Desventajas de los criptosistemas de clave simétrica.

- En una comunicación a dos, la clave debe permanecer en secreto pero al mismo tiempo ambas partes deben disponer de ella. Esto implica que para compartirla necesitan otro criptosistema.
- Por seguridad, las máximas de la criptografía obligan a que la clave sea cambiada con frecuencia, puede que incluso tras cada sesión. Esto pone aún más en evidencia el inconveniente anterior.
- Los mecanismos de firma digital basados en clave simétrica a menudo requieren claves largas.

Clave pública y clave privada

Surgen a finales del siglo XX, y su principal característica es que en ellos la clave con la que se encripta y con la que se desencripta no son la misma. Normalmente una de las partes, ya sea el emisor o el receptor, hace *pública* una clave. Con ella, la otra parte encripta el mensaje y también lo hace público. Haciendo uso de la clave privada es posible desencriptar el mensaje, de modo que esta permanece oculta.

Ventajas de los criptosistemas de pública:

- Sólo debe garantizarse la confidencialidad de la clave privada. Si embargo esto obliga a garantizar también la autenticación de la clave pública.
- Dependiendo del método de uso, es posible mantener las mismas claves pública y privada durante un tiempo considerables, a veces incluso años.
- Muchos esquemas de clave pública-privada proporcionan mecanismos de firma digital bastante eficientes. La longitud de la clave de estos suele ser mucho menor que la de su contraparte de clave simétrica.

Desventajas de los criptosistemas de pública:

- La tasa de información de los criptosistemas más populares de esta clase suele ser varios órdenes de magnitud menor que los criptosistemas de clave simétrica más conocidos.
- Los tamaños de las claves requeridos suelen ser mucho mayores que en el caso de clave simétrica.
- No ha sido probada la seguridad de ningún criptosistema de clave pública-privada. Los más efectivos basan su seguridad en la dificultad de un pequeño conjunto de problemas.

2.2.1. Seguridad

Ya mencionamos los objetivos fundamentales de la criptografía, y a pesar de sus diferencias, existe una palabra que los engloba a todos ellos: seguridad. Por tanto, debemos poder medir la seguridad de los criptosistemas de alguna forma. En la literatura se encuentran varios criterios evalúan dicha seguridad. Comentamos los principales a continuación.

Seguridad incondicional

Esta es una medida de la Teoría del Información. Se asume que un adversario posee recursos computacionales ilimitados y la cuestión es si existe información suficiente para romper el sistema. Si la respuesta es negativa, se dice que el sistema tiene

secreto perfecto. Para ello es necesario que la incertidumbre sobre el texto claro conociendo el texto cifrado sea la misma que sin conocerlo. Es decir, el texto cifrado no debe aportar ninguna información. Una condición necesaria para que un cifrador de flujo tenga secreto perfecto es que la longitud de la clave sea, al menos, la del texto a encriptar. El *One-time-Pad* es un ejemplo de ello, no obstante, la mayoría de esquemas de encriptación no ofrecen esta propiedad.

Seguridad en complejidad teórica

Dado un modelo de computación se asume que los atacantes poseen potencia computacional de orden polinómico. Se construye entonces una prueba de seguridad relativa al modelo. Usualmente se llevan a cabo análisis asintóticos y análisis en los que se asume el peor caso posible. Esto es fundamental para que las pruebas tengan credibilidad, aunque en ciertas instancias no son aplicables. No obstante, es posible que los ataques de orden polinómico no sean factibles en la práctica y se queden sólo a nivel teórico. Cabe mencionar además que los análisis de la complejidad teórica no son válidos para formular principios fundamentales ni para confirmar intuiciones. Este es uno de esos ámbitos en los que se descubren aplicaciones prácticas antes de que se haya consolidado una base teórica sólida.

Seguridad probable

Un método de encriptación se dice que tiene *seguridad probable* si la dificultad para romperlo es esencialmente la misma que la de un problema que se sepa que es supuestamente difícil. Por ejemplo, el RSA se basa en la asunción de que la factorización de un número no tiene un algoritmo eficiente.

Seguridad computacional

Mide el esfuerzo computacional requerido por los mejores métodos conocidos y asumiendo que se llevan a cabo sólo los ataques relevantes. Por tanto decimos que un criptosistema es *computacionalmente seguro* si la capacidad de computación necesaria para romperlo excede sobradamente los recursos computacionales con los que pueda contar un adversario.

A menudo, los sistemas de este tipo se basan en problemas difíciles, aunque al contrario que en la seguridad probable, no existe ninguna prueba formal de la equivalencia en la complejidad. Gran parte de los sistemas de clave simétrica y los de clave pública y clave privada presentan seguridad computacional. A veces se nombra a esta clase por *seguridad práctica*.

2.3. Algunos criptosistemas

2.3.1. Cifrado en flujo

El cifrado en flujo encripta cada carácter (generalmente un bit) de forma individual usando una transformación que varía con el tiempo. Contrariamente, el cifrado en bloque, encripta agrupaciones de caracteres con una transformación fija.

Los cifradores en flujo son, en general, más fáciles de implementar respecto a los de bloque, al menos en lo que a hardware se refiere. Además, en muchos casos son más apropiados y de hecho, en ciertos otros son obligatorios. Otro de sus puntos fuertes es su muy limitada o incluso inexistente propagación de errores, lo que los convierte en un candidato idóneo en casos en los que es probable que se produzcan errores.

Comentamos el ejemplo más elemental de cifrado en flujo: el *cifrado de Vernam* sobre un alfabeto binario, definido por

$$c_i = m_i \oplus k_i, \text{ para cada } i,$$

siendo $m_1, m_2, \ldots$ los dígitos del texto claro, $k_1, k_2, \ldots$ los dígitos de la secuencia encriptadora, y $c_1, c_2, \ldots$ los dígitos del texto cifrado. A la hora de desencriptar, simplemente hacemos $m_i = c_i \oplus k_i$. Si los dígitos de la clave son generados de forma aleatoria e independiente, entonces este cifrado de Vernam recibe el nombre de *one-time-pad*, y es incondicionalmente seguro frente ataques al texto cifrado. Más concretamente, si M, C y K son, respectivamente, el texto claro, el texto cifrado y la clave secreta, entonces $H(M|C) = H(M)$. De este modo, el texto cifrado no aporta ninguna información acerca del texto claro.

En Teoría de la Información se define lo que es la entropía de una variable aleatoria como el promedio de la información o incertidumbre inherentes a las salidas de la variable. Dada una variable aleatoria discreta X que toma valores sobre un alfabeto finito $\mathcal{A}$ y tiene por función de distribución $p : \mathcal{A} \to [0, 1]$ se define

$$H(X) = -\sum_{a \in \mathcal{A}} p(a) \log_2 p(a)$$

Se observa, no obstante, un problema evidente: la clave debe ser tan larga como el texto a encriptar, lo que aumenta la complejidad del manejo y distribución de la clave. Esto motiva el diseño de cifrados en flujo cuya secuencia cifradora sea generada de forma pseudoaleatoria a partir de una clave semilla más pequeña. Cabe destacar, sin embargo, que estos cifradores no ofrecen seguridad incondicional, pues $H(K) \ll H(M)$, pero se espera que sí ofrezcan seguridad computacional.

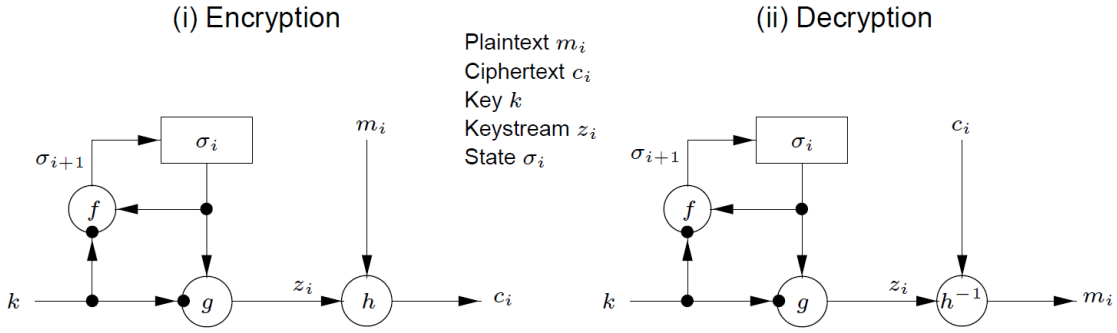
A menudo, los cifradores en flujo se clasifican en *síncronos* y *asíncronos*.

Definición 2.3.1. *Un cifrador en flujo se dice que es síncrono si la secuencia de claves es generada de forma independiente con respecto al texto claro y al texto encriptado.*

El proceso de encriptación de un cifrador de este tipo puede describirse mediante las siguientes ecuaciones

$$\sigma_{i+1} = f(\sigma_i, k),$$
$$z_i = g(\sigma_i, k),$$
$$c_i = h(z_i, m_i),$$

siendo σ_0 el estado inicial (que puede ser determinado por la clave base k), f es *la función del siguiente paso*, g es la función que produce la clave z_i, y h es la *función de salida*, que combina la clave y el dígito del texto claro m_i para producir c_i.



Presentamos alguna propiedades de los cifrados en flujo síncronos.

1. Requisitos de sincronización: En un cifrador síncrono, para lograr una correcta desencriptación, tanto el emisor como el receptor deben estar *sincronizados*, es decir, deben usar la misma clave y operar en la misma posición cada vez. Si durante la transmisión se pierden algunos dígitos del texto cifrado o aparecen ciertos otros, entonces la desencriptación falla y es necesario un proceso de *resincronización*.

2. No propagación de errores: Si un dígito del texto cifrado sufre una modificación (sin ser eliminado), el resto de dígitos no se ven afectados.

3. Ataques activos: Como consecuencia de la propiedad 1, la inserción o eliminación de dígitos en el texto cifrado por parte de un atacante activo causa una inmediata pérdida de sincronización, que puede ser detectada por el desencriptdor. Además, de la propiedad 2 se desprende que un atacante activo puede modificar ciertos dígitos concretos del texto cifrado y ver cómo afecta esto al texto claro. Estas observaciones muestran la necesidad de implementar mecanismos adicionales para subsanar estas fisuras.

El siguiente es uno de los cifradores de flujo más conocidos popularmente.

Definición 2.3.2. *Un cifrador en flujo binario aditivo es un cifrador en flujo síncrono en el que los dígitos de la clave, los del texto claro y los del texto encriptado son binarios y la función de salida h es la función XOR.*

Por otro lado tenemos los cifradores en flujo asíncronos.

Definición 2.3.3. *Un cifrador en flujo se dice que es auto-síncrono o asíncrono si la clave es generada a partir de una función de la clave base y un número fijo de dígitos del texto cifrado.*

El proceso de encriptación de un cifrador de este tipo puede describirse mediante las siguientes ecuaciones

$$\sigma_i = (c_{i-t}, c_{i-(t-1)}, \ldots, c_{i-1}),$$
$$z_i = g(\sigma_i, k),$$
$$c_i = h(z_i, m_i),$$

donde $\sigma_0 = (c_{-t}, \ldots, c_{-1})$ es el estado inicial (que no es secreto), k es la clave, g es la función que genera la secuencia encriptadora z_i, y h es la función de salida, que combina la secuencia encriptadora z_i y el texto claro m_i para producir el texto cifrado c_i.

Mostramos algunas de las propiedades de estos encriptadores.

1. Autosincronización: es posible llevar a cabo una autosincronización si alguno o varios dígitos del texto cifrado son eliminados o insertados, ya que la desencriptación depende sólo de un número fijo de caracteres del texto cifrado.

2. Error de programación limitado: Supongamos que el estado de un encriptador de este tipo depende de los t dígitos previos del texto cifrado. Si un único dígito del texto cifrado es modificado, inlcuso si es eliminado o insertado, entonces la desencriptación de hasta t dígitos anteriores puede ser incorrecta, pero posteriormente sí es correcta.

3. Ataques activos: La propiedad anterior implica que cualquier modificación de los dígitos del texto cifrado por parte de un atacante activo afecta a la desencriptación de otros tantos dígitos, lo que facilita que dicho ataque sea detectado respecto al caso síncrono. En cambio, la propiedad 1 hace ver que la inserción, eliminación o modificación de dígitos del texto cifrado es más difícil de percibir. Esto ilustra la necesidad de implementar otros mecanismos adicionales.

4. Difusión de estadísticas del texto claro: Como cada dígito del texto claro afecta a todos los dígitos del texto cifrado que hay a continuación, las propiedades estadísticas del texto claro están diseminadas por todo el texto cifrado. Esto hace a estos encriptadores más resistentes frente a ataques basados en la redundancia del texto claro que los de tipo síncrono.

2.3.2. Cifrado en bloque

Los encriptadores en bloque son uno de los elementos más destacados e importantes en muchos criptosistemas. De forma individual proporcionan confidencialidad, pero como componentes dentro de otros criptosistemas pueden servir para construir generadores de números pseudoaletorios, cifradores de flujo y funciones hash entre otros.

No obstante, los cifradores en bloque no son adecuados para todos los casos, como es de suponer. En ocasiones esto puede deberse a las propiedades inherentes de estos encriptadores, por ejemplo, en caso de haber ciertos requisitos de velocidad o limitación de memoria. De hecho, a menudo se sacrifica eficiencia a cambio de seguridad.

En términos generales, un cifrador en bloque es una función que transforma n bits del texto claro en m bits de texto cifrado. El parámetro n recibe el nombre de *longitud del bloque*. Esta función se parametriza mediante una clave de k bits que denotaremos por K, tomando valores de un subconjunto $\mathcal{K}$ (conocido como espacio de claves) del conjunto de todos los vectores de k bits V_k. Normalmente se asume que esta clave se elige de forma aleatoria. Además, para que se produzca una adecuada desencriptación, la función de encriptación debe ser biyectiva. Más formalmente:

Definición 2.3.4. *Un cifrador en bloque (de n bits) es una función* $E : V_n \times \mathcal{K} \to V_n$ *tal que para cada clave* $K \in \mathcal{K}$*, la aplicación* $E(P, K)$ *de* V_n *en* V_n *es invertible. Esta función recibe el nombre de función encriptadora y es denotada por* $E_K(P)$*. Su inversa recibe el nombre de función desencriptadora, y la denotamos por* $D_C(K)$*, donde* $C = E_K(P)$*.*

Estas funciones suelen consistir en una composición de otras funciones de dos tipos, de sustitución (difusión) y de transposición (cofusión).

Cada una de las claves K define en general una biyección diferente, si el número de claves es $\#\mathcal{K}$, entonces el tamaño efectivo de clave es $\log(\#\mathcal{K})$. Si todas las claves son equiprobables, y cada una define una biyección distinta, entonces la entropía del espacio de claves es también $\log(\#\mathcal{K})$.

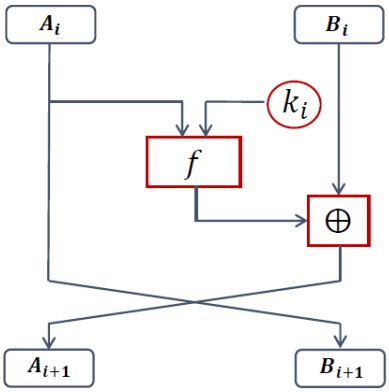
Puesto que este tipo de cifradores siempre recibe como entrada una cantidad fija de bits, a la hora de encriptar un mensaje este se divide en bloques de la longitud correspondiente, añadiéndose al final un cierto número de bits en caso de ser necesario.

Principalmente existen dos tipos en los que se dividen los cifradores a bloques:

- Modo ECB: Se caracteriza por que cada bloque se cifra y descifra independientemente. Esto permite que ambos procesos puedan ejecutarse en paralelo. No obstante, uno de sus puntos débiles es que bloques iguales siempre se cifran de la misma forma, por lo que un mensaje concreto siempre da lugar al mismo cifrado.
- Modo CBC: El cifrado y descifrado son procesos encadenados, por lo que la clave de cada bloque depende del anterior. A diferencia del modo CBC, al utilizar vectores de inicialización distintos es posible obtener cifrados distintos para un mismo mensaje. Debido a que presenta una mayor seguridad, es mucho más usado, ya que es capaz de resistir ataques de inserción y de modificación. Por otro lado, aunque el proceso de descifrado sí puede paralelizarse, el de cifrado no debido a que es secuencial.

Uno de los ejemplos clásicos de cifradores a bloques son las redes de Feistel. En ellas se divide el mensaje en bloques de longitud N y cada uno de ellos es procesado a lo largo de n *rondas*. Una ronda no es más que el paso de una entrada por un determinado algoritmo. Cada bloque sufre una primera división en dos subbloques A_0

y B_0 de longitud $\frac{N}{2}$, y a continuación se van obteniendo $A_{i+1} = B_i \oplus f(k_i, A_i)$, $B_{i+1} = A_i$ de forma iterativa.



En general los esquemas de encriptación suelen consistir en varias rondas. Además, existe una clave k que se divide en n subclaves $k_1, \ldots, k_n$, una para cada ronda. En cada una de ellas se emplea una misma función f (no necesariamente invertible) que toma como entrada $\frac{N}{2}$ bits del mensaje y la clave k_i. La salida es un bloque de $\frac{N}{2}$ bits.

A la hora del descifrado, si se conoce la clave k es posible obtener A_i y B_i a partir de A_{i+1} y B_{i+1}. En primer lugar $A_i = B_{i+1}$, y como $A_{i+1} = B_i \oplus f(k_i, A_i)$, tenemos que $B_i = A_{i+1} \oplus f(k_i, A_i)$. De este modo, a partir de A_n y B_n es posible retroceder hasta A_0 y B_0 y recuperar el bloque correspondiente del mensaje original. Observamos por tanto que no es necesario que la función f en cuestión tenga inversa. El único requisito que debemos exigirle a f es que sea lo bastante robusta.

Uno de los algoritmo de encriptación de tipo Feistel más conocidos es el ya mencionado DES. Cuenta con 16 rondas, y como entrada toma bloques de 64 bits. Cada ronda cuenta con 3 fases:

1. Expansión: Cada subbloque de 32 bits se expande a 48 bits la *permutación de expansión* (lo que duplica algunos bits).

2. Mezcla: Estos bits se combinan con una subclave mediante la operación XOR.

3. Sustitución: El nuevo bloque se divide en 8 partes de 6 bits cada uno y pasan a a ser procesados por las *S-boxes* (*substitution-box*), que por sus características son altamente no lineales. Sobre ellas se cimienta la seguridad de este algoritmo. La siguiente imagen muestra un ejemplo de S-box de DES.

S_5		4 bit de entrada internos															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Bits externos	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1100	0011	1001	1000	0110
	10	0100	0010	0001	1011	1100	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1100	0100	0101	0011

4. Permutación: Por último, las 32 salidas de las S-boxes se reordenan según la *P-box*, que determina una permutación fija (y por tanto es de caracter lineal).

Gracias a las S-boxes se garantiza la confusión, y a la P-box, la difusión, lo que da lugar a un criptosistema aparentemente seguro. Como hemos comentado, DES plantea diversas fisuras, siendo una de ellas la baja no linealidad de la función empleada en una de sus S-boxes, lo que detallaremos más adelante.

2.4. Ataques más comunes

Si la criptografía vela por la seguridad de la información, debe ser porque hay alguien que pretende atentar contra dicha seguridad. Este es el caso de los ataques, que admiten una primera división en dos categorías:

- Los *ataques pasivos* son aquellos en los que el adversario sólo monitorea el canal de comunicación, es decir, sólo puede "ver" lo que se transmite, pero no modificarlo. Ante este tipo de ataque sólo está en riesgo la confidencialidad de la información.
- En un *ataque activo*, el adversario puede añadir, eliminar e incluso modificar bits del mensaje que se está transmitiendo. Ante este tipo de ataque sólo está en riesgo la autenticidad y la integridad de la información, además de la confidencialidad.

Además, en función de la información de la que dispone el atacante podemos distinguir varios tipos de ataques:

- *Ataque al texto cifrado*: en él, el criptoanalista trata de deducir la clave de desencriptación o el texto claro disponiendo sólo del texto cifrado. Cualquier criptosistema vulnerable a ataques de este tipo se considera que es *absolutamente inseguro*.
- *Ataque al texto claro conocido*: es en el que el adversario cuenta con una parte del texto claro y con el correspondiente texto cifrado. Típicamente, este tipo de ataque sólo es moderadamente difícil de llevar a cabo.
- *Ataque al texto claro elegido*: aquel en que el atacante elige el texto claro y recibe el texto cifrado. Entonces trata de obtener toda la información posible para poder desencriptar otros textos cifrados de los que desconoce los correspondientes textos claros.
- *Ataque adaptativo al texto claro elegido*: se trata de un ataque al texto claro elegido en el que la elección de los textos claros puede depender de de los textos cifrados recibidos previamente.
- *Ataque al texto cifrado elegido*: es en el que el criptoanalista elige el texto cifrado y entonces recibe el correspondiente texto claro. Puede parecer difícil de llevar a cabo, pero es posible que el atacante sea capaz de acceder al equipamiento utilizado para la desencriptación pero no a la clave en sí. El objetivo es entonces tratar de deducir el correspondiente texto claro de cualquier texto cifrado.

- *Ataque adaptativo al texto cifrado elegido*: análogo al ataque adaptativo al texto claro elegido sólo que en el caso de del ataque el texto cifrado elegido.

2.4.1. Ataque algebraico

Los ataques algebraicos se basan en la resolución de sistemas de ecuaciones no lineales en las que figuran bits del texto claro, del texto encriptado y de la clave. El atacante debe tratar de obtener las suficientes ecuaciones no lineales de grado bajo, a partir de las cuales obtiene los bits de la clave.

La obtención de estas ecuaciones se realiza una única vez para cada encriptador. Por otro lado, el método más habitual para resolver estas ecuaciones es mediante linealización y algoritmos basados en bases de Gröbner. La linealización consiste en sustituir los términos no lineales por nuevas variables. Una vez hecho esto el sistema resultante puede resolverse con facilidad mediante eliminación Gaussiana, por ejemplo.

En general no resulta nada sencillo obtener estas ecuaciones tan simples a partir de un encriptador bien diseñado, lo que hace que la mayoría de los cifradores sean inmunes a los ataques algebraicos triviales.

2.4.2. Criptoanálisis lineal

A pesar de los años transcurridos, el método del criptoanálisis lineal sigue siendo uno de los más eficientes estadísticamente, a la par con el criptoanálisis diferencial. Se trata de un ataque al texto claro conocido, y fue propuesto para el FEAL en 1992 por Matsui y Yamagishi [MY92], y para el DES en 1993 por Matsui [Mat93].

La idea de este método consiste en encontrar una cierta relación lineal (una serie de ecuaciones lineales) entre los bits del texto cifrado y los del texto claro. Si el criptosistema está bien diseñado, entonces la probabilidad de que se cumpla dicha relación para ciertos $m \in \mathcal{M}$ y $c \in \mathcal{C}$ concretos será de $p = 1/2$. No obstante, en general la probabilidad diferirá algo de este valor, digamos $p = 1/2 + \varepsilon$ con $\varepsilon \in \mathbb{R}$. A este ε se le conoce como *bias*, y cuanto mayor sea su valor, mejor será el rendimiento del criptoanálisis lineal sobre este criptosistema. El atacante cuenta con una cierta cantidad de parejas (m, c), donde m es un texto claro y c es su correspondiente texto cifrado. Para una clave desconocida K se realiza es un contraste de hipótesis, en el que la hipótesis nula es que dicha relación se cumple para K, y la hipótesis alternativa es que no se verifica. Para que este método proporcione buenos resultados es aconsejable que el tamaño de la muestra sea aproximadamente $|\varepsilon|^{-2}$.

Desde su formulación hace casi treinta años se han publicado muchos artículos dedicados a generalizaciones y aplicaciones de este método, así como algunas mejoras.

2.4.3. Criptoanálisis diferencial

Su desarrollo suele atribuirse a Eli Biham y Adi Shamir [BA93], quienes a finales de los ochenta publicaron varios ataques contra una serie de cifradores en bloque y funciones hash, incluyendo una debilidad teórica del DES.

El criptoanálisis diferencial se centra en la probabilidad de que se den ciertas diferencias en el texto claro y en la última ronda del encriptador. Si el correspondiente cifrado de $m' = (m'_1, \ldots, m'_n)$ es $c' = (c'_1, \ldots, c'_n)$ y el de $m'' = (m''_1, \ldots, m''_n)$ es $c'' = (c''_1, \ldots, c''_n)$, la diferencia entre las entradas viene dada por $\Delta_m = m' \oplus m''$, y entonces $\Delta_m = (\Delta_{m_1}, \ldots, \Delta_{m_n})$, donde $\Delta_{m_i} = m'_i \oplus m''_i$. Del mismo modo, se tiene $\Delta_c = (\Delta_{c_1}, \ldots, \Delta_{c_n})$.

En un cifrador completamente aleatorio, la probabilidad de que una cierta diferencia en la salida Δ_c se dé para una diferencia concreta en la entrada Δ_m es de $\frac{1}{2^n}$. El objetivo del criptoanálisis diferencial es tratar de buscar una situación en la que la probabilidad de observar un Δ_c en particular para un Δ_m concreto sea mucho mayor que $\frac{1}{2^n}$. Al par (Δ_m, Δ_c) se le conoce como *diferencial*.

El criptoanálisis diferencial es un ataque al texto claro elegido, es decir, el atacante puede decidir la entrada y ver cómo se refleja en la salida para tratar de obtener la clave. En concreto, el atacante elegirá un par de entradas m', m'' que verifiquen un cierto Δ_m, sabiendo que para dicho Δ_m se obtiene un Δ_c en específico con una alta probabilidad.

A continuación presentamos una tabla que contrasta la principales características del criptoanálisis lineal y del criptoanálisis diferencial.

Criptoanálisis Lineal	**Criptoanálisis Diferencial**
Desarrollado por Matsui y Yamagashi en 1992.	Definido por primera vez en 1990 por Eli Biham and Adi Shamir.
Se trata de un ataque al texto claro conocido.	Se trata de un ataque al texto claro elegido.
Trabaja en un único bit cada vez.	Puede trabajar en múltiples bits a la vez.
Principalmente se basa en ataques al texto cifrado.	Principalmente se basa en ataques al texto claro elegido.
En esencia consiste en averiguar las relaciones lineales existentes entre los bits del texto claro, los del texto cifrado y algunos de la clave.	La idea principal es averiguar ciertas "pistas" sobre algunos bits críticos, minimizando así la búsqueda.
Los atributos de varias entradas reflejan la estructura interna de una sola entrada.	La estructura subyacente de cada entrada es importante, pues los atributos de la entrada son diferenciales.
El atacante desencripta cada texto cifrado usando todas las subclaves disponibles y analizando el texto cifrado intermedio que resulta para determinar la salida de una ronda entera.	Tras varias ronda, el atacante analiza los cambios obtenidos en el texto cifrado intermedio. La combinació de varios asaltos se conoce como *criptoanálisis diferencial lineal*.
Se selecciona un texto claro aleatorio.	El texto claro utilizado es elegido meticulosamente.
Los textos claros se usan uno por uno.	Los textos claros se usan por parejas.
La complejidad de los ataques es razonablemente baja.	La complejidad de los ataques es elevada.
Las relaciones matemáticas que se establecen entre los textos claros utilizados tienen aproximaciones lineales.	Las relaciones matemáticas entre los textos claros usados tienen diferencias específicas.
El objetivo del ataque es identificar relaciones lineales entre algunos bits del texto claro, del texto cifrado, y de la clave.	El objetivo del ataque es identificar algunos bits de la clave.

3. Introducción a las funciones Booleanas y a las funciones bent

Una vez establecidas unas nociones básicas en criptografía podemos comenzar a introducir el tema en torno al cual gira este proyecto. Las funciones bent son un caso particular de funciones Booleanas, por lo que nuestra primera parada será hablar de estas últimas, las funciones Booleanas. Tras ello introduciremos una herramienta fundamental de la que haremos uso muy a menudo: la *Transformada de Walsh Hadamard*. Esto dará pie a definir por fin lo que se conoce como una *función bent*. Una vez las hayamos estudiado en razonable profundidad, dedicaremos una sección a hablar de algunas de sus aplicaciones en la Criptografía, y para terminar con este tercer capítulo, comentaremos varias de las generalizaciones del concepto de bent que pueden encontrarse en la literatura. Una de ellas en concreto será la protagonista del capítulo siguiente, pero no adelantemos acontecimientos aún.

3.1. Funciones Booleanas

Comenzamos pues presentando las funciones Booleanas, que no son más que funciones binarias. Sus aplicaciones en criptografía son notorias, ya que a fin de cuentas los ordenadores trabajan en binario. En concreto, existen varias propiedades (criptográficas) deseables para dichas funciones, las cuales describiremos brevemente, comentando el motivo por el que son importantes.

3.1.1. Definiciones básicas

Comenzamos dando la definición matemática de función Booleana.

Definición 3.1.1. *Dado $n \in \mathbb{N}$, una función Booleana es cualquier aplicación $f : \mathbb{F}_2^n \to \mathbb{F}_2$.*

Definición 3.1.2. *Dados $n, m \in \mathbb{N}$, una función Booleana vectorial es cualquier aplicación $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$.*

Toda función Booleana puede expresarse a través de una tabla de verdad.

$x_1 \cdots x_n$	$f(x_1, \ldots, x_n)$
$0 \cdots 0$	$f(0, \ldots, 0)$
$1 \cdots 0$	$f(1, \ldots, 0)$
$\vdots$	$\vdots$
$1 \cdots 1$	$f(1, \ldots, 1)$

Veamos un ejemplo de esto.

Ejemplo 3.1.3. *Sea* $f : \mathbb{F}_2^2 \to \mathbb{F}_2$ *dada por* $f(0,0) = 1$, $f(0,1) = 0$, $f(1,0) = 0$ *y* $f(1,1) = 1$. *Su tabla de verdad es:*

x_1x_2	$f(x_1, x_2)$
0 0	1
1 0	0
0 1	0
1 1	1

Esta representación no parece demasiado práctica, sobre todo cuando el número de variables es grande. Existe otra forma más compacta de representar las funciones Boolenas, la *forma normal algebraica*:

$$f(x_1, \ldots, x_n) = \bigoplus_{I \subset \{1,\ldots,n\}} \left(\prod_{i \in I} x_i \right)$$

Por ejemplo, la función f del Ejemplo 3.1.3 tiene por forma algebraica normal $f(x_1, x_2) = x_1 \oplus x_2 \oplus 1$. Observamos que esta forma no es más que expresar la función en forma de "polinomio".

Se define el grado algebraico de una función Boolena como el número de variables del mayor de los términos. En el caso de nuestra f, claramente es 1. Diremos que una función Boolena es afín si su grado es 1, y cuadrática cuando este sea 2.

Comentar por último que existe un algoritmo para obtener la forma normal algebraica de una función f a partir de su tabla de verdad. Puede enconctrarse, por ejemplo, en [Mes16] como Algoritmo 1.1.

3.1.2. Función Traza y representación polinómica de una función Booleana

Es bien conocido el isomorfismo entre $\mathbb{F}_2^n$ y $\mathbb{F}_{2^n}$ (como espacios vectoriales, naturalmente). Esto permite traducir las funciones Booleanas $f : \mathbb{F}_2^n \to \mathbb{F}_2$ a funciones $g : \mathbb{F}_{2^n} \to \mathbb{F}_2$ mediante lo que se conoce como representación polinómica. Además, esto no se adscribe únicamente al caso Booleano, sino que se aplica a cualquier cuerpo $\mathbb{F}_p$ con p un primo arbitrario. Por tanto, podremos extender la noción de representación polinómica al caso p-ario, es decir, a funciones $f : \mathbb{F}_p^n \to \mathbb{F}_p$. Estas consideraciones nos serán de utilidad en la Sección 4.2.

Además, recordamos la definición de la función traza, así como algunas de sus propiedades.

Definición 3.1.4. *Definimos la traza como una función sobre un cuerpo finito* $\mathbb{F}_{2^n}$ *como*

$$tr(c) = c + c^2 + \ldots + c^{2^n-1}$$

Propiedades de la traza:

- $tr^2(c) = tr(c^2) = tr(c)$.
- $tr(c + c') = tr(c) + tr(c')$.
- La traza es una función equilibrada.

Dado que existe un isomorfismo entre $\mathbb{F}_2^n$ y $\mathbb{F}_{2^n}$, es posible identificar la función traza con una función Booleana en n variables. Consideremos el siguiente ejemplo:

Ejemplo 3.1.5. *Sea $\mathbb{F}_{2^3}$ el cuerpo construido a partir del polinomio irreducible $g(x) = x^3 + x + 1$, que tiene como elemento primitivo $\alpha := x + 1$. En la siguiente tabla mostramos cómo calcular la traza de cada elemento de $\mathbb{F}_{2^3}$, apareciendo en la última columna los valores de la función Booleana correspondiente a $tr(\cdot)$. Vemos que se trata de una función lineal, $f(x_1, x_2, x_3) = x_3$.*

Vector	Polynomial	α^k	Trace $\mathrm{tr}(c) = c + c^2 + c^4$	f
(000)	0	–	0	0
(001)	1	α^0	$\alpha^0 + \alpha^{0\cdot 2} + \alpha^{0\cdot 4} = 1$	1
(010)	x	α^5	$\alpha^5 + \alpha^{5\cdot 2} + \alpha^{5\cdot 4} = \alpha^5 + \alpha^3 + \alpha^6 = 0$	0
(011)	$x + 1$	α^1	$\alpha^1 + \alpha^{1\cdot 2} + \alpha^{1\cdot 4} = 1$	1
(100)	x^2	α^3	$\alpha^3 + \alpha^{3\cdot 2} + \alpha^{3\cdot 4} = \alpha^3 + \alpha^6 + \alpha^5 = 0$	0
(101)	$x^2 + 1$	α^2	$\alpha^2 + \alpha^{2\cdot 2} + \alpha^{2\cdot 4} = \alpha^2 + \alpha^4 + \alpha^1 = 1$	1
(110)	$x^2 + x$	α^6	$\alpha^6 + \alpha^{6\cdot 2} + \alpha^{6\cdot 4} = \alpha^6 + \alpha^5 + \alpha^3 = 0$	0
(111)	$x^2 + x + 1$	α^4	$\alpha^4 + \alpha^{4\cdot 2} + \alpha^{4\cdot 4} = \alpha^4 + \alpha^1 + \alpha^2 = 1$	1

En general, la función traza es lineal para cualquier isomorfismo entre $\mathbb{F}_2^n$ y $\mathbb{F}_{2^n}$. De hecho, se tiene el siguiente teorema.

Teorema 3.1.6. *El conjunto de funciones Booleanas lineales en n variables coincide con el conjunto de funciones $\ell_a(c) = tr(a \cdot c)$, donde a recorre todo $\mathbb{F}_{2^n}$.*

Esto nos permite dar el salto a la *representación polinómica* de una función Booleana (identificamos $\mathbb{F}_2^n$ con $\mathbb{F}_{2^n}$ mediante el isomorfismo correspondiente).

Proposición 3.1.7. *Toda función Booleana vectorial $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ puede expresarse de forma única en su representación polinomial sobre $\mathbb{F}_{2^n}$ como*

$$F(c) = \sum_{j=0}^{2^n-1} a_j c^j$$

para unos $a_j \in \mathbb{F}_{2^n}$ concretos.

Demostración.

En efecto, vemos que el número de funciones Booleanas vectoriales en n variables es $(2^n)^{2^n}$, y el número de polinomios distintos $\sum_{j=0}^{2^n-1} a_j x^j$ es también $(2^n)^{2^n}$, y polinomios distintos definen funciones diferentes.

□

Como $\mathbb{F}_2$ es siempre un subcuerpo de $\mathbb{F}_{2^n}$ para cualquier n, es claro que una función Booleana $f : \mathbb{F}_{2^n} \to \mathbb{F}_2$ es "una parte" de una función vectorial. Por tanto, dicha f también puede expresarse como un polinomio sobre $\mathbb{F}_{2^n}$ mediante

$$f(c) = \sum_{j=0}^{2^n-1} a_j c^j$$

para unos $a_j \in \mathbb{F}_{2^n}$ concretos. Observamos que un polinomio arbitrario en el lado derecho de la igualdad toma valores en $\mathbb{F}_2$ si y sólo si a_{2^n-1} pertenece a $\mathbb{F}_2$ y $a_{2i} = a_i^2$ para cada $1 \leq i \leq 2^n - 2$, tomando los índices $2i$ módulo $2^n - 1$.

Como adelantábamos, también es posible definir el concepto de representación polinomial para funciones $f : \mathbb{F}_p^n \to \mathbb{F}_p$ siguiendo un desarrollo completamente análogo al dado para el caso Booleano.

3.1.3. Uso de las funciones Booleanas en la Criptografía

Uno de los usos más inmediatos de las funciones Booleanas en criptografía es en los cifradores a flujo. A menudo, las secuencias pseudoaleatorias que actúan como secuencia encriptadora son generadas mediante generadores por registro de desplazamiento con retroalimentación lineal, o LFSR (*Linear Feedback Shift Register*). Como su nombre indica, la dependencia entre la entrada y la salida es lineal, lo que hace que los cifradores que sólo se basan en LFSR sean altamente inseguros.

Para producir esquemas de encriptación más seguros se utilizan funciones Booleanas con las que generar la secuencia cifradora a partir de las entradas del LFSR, favoreciendo que la relación entre el texto claro y el cifrado sea más compleja. Más concretamente, los bits del texto cifrado se obtienen sumando un bit de la clave que da como salida la función Booleana en cuestión, cuya dependencia respecto de la entrada del LFSR (la información secreta) es no lineal. Por tanto, la seguridad de estos criptosistemas depende íntimamente de la elección de la función Booleana.

Los modelos clásicos basan su diseño en el *one-time-pad* y en el cifrado de Vernam. Para ello se diseña un generador pseudoaleatorio. Habitualmente el modelo filtrador se compone de varios LFSR y de una función combinadora o de filtro f que genera la salida. En el modelo del generador combinador, se combinan las salidas de varios LFSR mediante una función Booleana. Estos modelos pueden verse en las siguientes figuras.

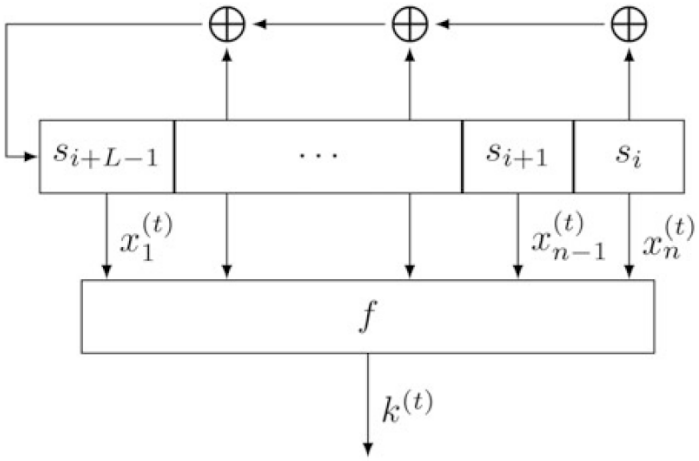


Figura 3.1: Modelo de filtro

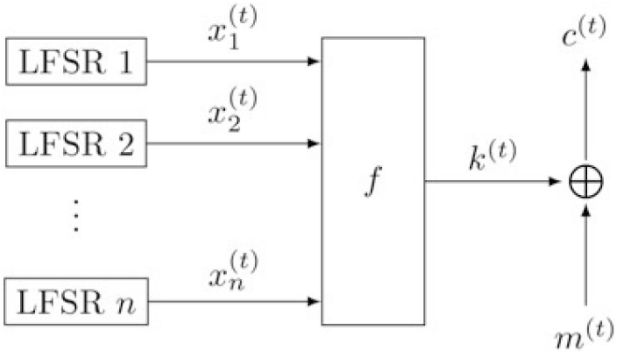


Figura 3.2: Modelo combinador

3.1.4. Criterios criptográficos para funciones Booleanas

El diseño de los criptosistemas convencionales se basa en dos principios fundamentales introducidos por Claude Shanon en su paper "Communication Theory of Secrecy Systems" [Sha49], publicado en 1949. Estos dos principios son *confusión* y *difusión*. En criptografía, la confusión se centra en ocultar cualquier tipo de estructura algebraica, y en el caso de las funciones Booleanas, está estrechamente relacionada con la complejidad de estas. De forma más clara, para Shanon la confusión consistía en hacer tan compleja como fuese posible la relación entre el texto claro y el cifrado.

Por otro lado, la difusión se refiere a que las redundancias en las estadísticas del texto claro se vean "disipadas" en las estadísticas del texto cifrado. Es decir, la no uniformidad en la distribución de ciertas letras en el texto claro debe redistribuirse

durante la encriptación. Un ejemplo de criptosistema que presenta una buena difusión (incluso óptima) es el cifrado de Vigenère.

Así, la dependencia entre los bits de salida y los de entrada debe ser muy compleja, y por tanto, el cambio de un sólo bit del texto claro debe provocar cambios en el texto cifrado del todo impredecibles.

Estos dos principios llevan vigentes más de medio siglo, y muchos de los ataques que se han encontrado frente múltiples criptosistemas no han hecho sino reafirmar la relevancia de ambos. Cada uno de estos ataques muestra una debilidad que el diseño de un criptostema debe tener en cuenta. Esto permite medir la resistencia frente a ciertos ataques bien conocidos estableciendo ciertos criterios. Enumeramos algunos de ellos.

Grado algebraico

La complejidad lineal de un generador pseudoaleatorio depende del grado algebraico de su función mezcla para resistir los ataques de Berlekamp–Massey [Mas69], y en modelo del filtro para los ataques de Rønjom–Helleseth [RH07], es por ello que desde un punto de vista criptográfico es importante que la función en cuestión presente un grado algebraico alto. A partir de la defineción de la forma algebraica normal de una función Boolena puede comprobarse que cualquier función Booleana en n variables tiene grado algebraico, como mucho, n.

Sea $n \geq 4$ par y consideremos la función

$$f : \mathbb{F}_2^n \to \mathbb{F}_2, \quad f(x) = x_1x_2 \oplus x_3x_4 \oplus \ldots \oplus x_{n-1}x_n \tag{3.1}$$

Es claro que f tiene grado algebraico 2. Más adelante veremos que se trata de una función bent.

Balanceamiento

Para evitar la dependencia estadística entre el texto claro y el texto cifrado y para prevenir ataques distinguidores [Car10] es importante que la función criptográfica esté *equilibrada* o *balanceada*. Esto significa que al evaluarla en todas sus posibles entradas, debe dar como salida tantos 0 como 1. Equivalentemente, una función Booleana en n variables f está equilibrada si su peso de Hamming es $w_H(f) = 2^{n-1}$. Obsérvese que cualquier función Booleana en n variables, si está equilibrada, entonces su grado algebraico es, como mucho, $n - 1$.

Es posible probar que la función 3.1 no esta balanceada.

No linealidad

La cualidad a la que damos más importancia en este texto y una de las más deseables, pues en el caso de los cifrados de flujo previene los *ataques de correlación rápida* y los mejores *ataques de aproximación afín*. Cuanto mayor es la no linealidad, menor es la eficiencia de estos ataques. Ofrece muchas garantías frente al criptoanálisis

lineal [Mat94]. La definiremos más adelante, y probaremos además que la función vista en 3.1 tiene una no linealidad de $2^{\frac{n}{2}}$

Inmunidad a la correlación y resiliencia

Para evitar los *ataques de correlación* en el modelo combinador, la función combinadora debe evitar la correlación de bajo orden. Existen dos formas de caracterizar la *inmunidad a la correlación*, presentamos ambas. Para la primera debemos conocer lo que es una *subfunción de grado k*.

Si f es una función Booleana en n variables, una *subfunción* de orden k de f es una función $f_{i_1,\ldots,i_k}^{a_1,\ldots,a_k}$ en la que cada varibles x_{i_j} tiene valor fijo a_{i_j}, con $1 \leq j \leq k$ y $0 < k \leq n$.

Definición 3.1.8. *Una función Booleana f se dice que es inmune a la correlación de grado k si el peso (de Hamming) de cualquiera de sus subfunciones de orden k es $w_H(f)/2^k$.*

Definición 3.1.9. *Una función Booleana f se dice que es t-resiliente si cualquiera de sus subfunciones de grado, como mucho t, está equilibrada.*

Como comentábamos, existe otra aproximación al concepto de inmunidad a la correlación. No es difícil ver la equivalencia entre ambos.

Teorema 3.1.10. *Una función Booleana f es inmune a la correlación de grado k si y sólo si su transformada de Walsh Hadamard (que se definirá más adelante) se anula en todos los vectores que como mucho tiene peso de Hamming k.*

Comentamos que una de las debilidades de las funciones bent es que no son inmunes a la correlación. Esto se debe a que su espectro de Walsh Hadamard es no nulo. Tenemos el siguiente resultado.

Teorema 3.1.11. *Si f es una función bent en n variables, entonces f no es inmune a la correlación, y por tanto tampoco es resiliente.*

Probaremos más adelante que la función 3.1 es bent, de modo que no es inmune a la correlación.

Inmunidad algebraica

Como ya comentamos, los ataques algebraicos se basan en la resolución de ciertas ecuaciones en varias variables. En el caso del los cifradores de flujo es posible tratar con sistemas sobredefinidos, es decir, que el número de ecuaciones linealmente independientes es mucho mayor que el de variables. Es por ello que el estudio de los *anulador* de una función Booleana ha cobrado gran relevancia, y en la práctica toda función Booleana debería tener una alta *inmunidad algebraica*.

Definición 3.1.12. *Sea f una función Boolena en n variables. Una función Boolena no nula g se dice que es un anulador de f si $fg = 0$.*

Definición 3.1.13. *La inmunidad algebraica de f, denotada por $AI(f)$, es el menor d tal que f o $1 \oplus f$ admite un anulador de grado algebraico d.*

Observamos que la inmunidad algebraica de una función Booleana f es, como mucho, su grado algebraico, ya que $1 \oplus f$ es un anulador de f. De hecho, es conocido que la inmunidad algebraica de una función de de n variables está acotada por $\lceil \frac{n}{2} \rceil$.

Actualmente una alta inmunidad algebraica es absolutamente necesaria para resistir ante ataques algebraicos, aunque no es suficiente debido a la existencia de los *ataques algebraicos rápidos*.

3.2. Transformada de Walsh Hadamard

Dedicamos esta sección a introducir una noción fundamental a la hora de definir y trabajar con funciones bent: la *Transformada de Walsh Hadamard*. Presentaremos también algunos resultados.

Definición 3.2.1. *Dada una función Booleana en n variables f, definimos la transformada de Walsh Hadamard de f como la función que va de $\mathbb{F}_2^n$ en $\mathbb{Z}$ y que está definida para cada $y \in \mathbb{F}_2^n$ por:*

$$W_f(y) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle x,y \rangle \oplus f(x)}.$$

El espectro de Walsh Hadamard de f es el conjunto

$$W_f = \{W_f(x) : x \in \mathbb{F}_2^n\}$$

Teorema 3.2.2. *Para cada función Boolena en n variables f y para cada $x \in \mathbb{F}_2^n$ se verifica la siguiente relación*

$$(-1)^{f(x)} = \frac{1}{2^n} \sum_{y \in \mathbb{F}_2^n} W_f(y)(-1)^{\langle x,y \rangle}.$$

El siguiente es un resultado bien conocido que utilizaremos en varias ocasiones.

Lema 3.2.3. *Para cada $x \in \mathbb{F}_2^n$ se tiene que*

$$\sum_{y \in \mathbb{F}_2^n} (-1)^{\langle x,y \rangle} = \begin{cases} 2^n, & \text{si } x = 0. \\ 0, & \text{si } x \neq 0. \end{cases}$$

Demostración.

Si $x = 0$, es claro. Supongamos que $x \neq 0$ y consideremos los hiperplanos $H = \{y \in \mathbb{F}_2^n : \langle x, y \rangle \equiv 0 \mod 2\}$, $H' = \{y \in \mathbb{F}_2^n : \langle x, y \rangle \equiv 1 \mod 2\}$, que forman una partición de $\mathbb{F}_2^n$. Para cada $y \in H$ el sumando correspondiente es 1 y para cada $y \in H'$ el sumando correspondiente es -1, como H y H' tienen el mismo cardinal ($2^n/2 = 2^{n-1}$), el resultado queda probado. □

Este lema nos permite probar el siguiente, que a su vez emplearemos para probar la conocida como *Identidad de Parseval*.

Lema 3.2.4. *Se verifica que*

$$\sum_{y\in\mathbb{F}_2^n} W_f(y)W_f(y\oplus x) = \begin{cases} 2^{2n}, & si\ x=0. \\ 0, & si\ x\neq 0. \end{cases}$$

Demostración.

Tenemos que

$$\begin{aligned}\sum_{y\in\mathbb{F}_2^n} W_f(y)W_f(y\oplus x) &= \sum_{y\in\mathbb{F}_2^n}\Big(\sum_{w\in\mathbb{F}_2^n}(-1)^{\langle y,w\rangle+f(w)}\sum_{z\in\mathbb{F}_2^n}(-1)^{\langle y\oplus x,z\rangle+f(z)}\Big)\\ &= \sum_{w\in\mathbb{F}_2^n}\sum_{z\in\mathbb{F}_2^n}\Big((-1)^{\langle x,z\rangle+f(w)+f(z)}\sum_{y\in\mathbb{F}_2^n}(-1)^{\langle y,w\oplus z\rangle}\Big)\\ &= 2^n\sum_{w\in\mathbb{F}_2^n}(-1)^{\langle y,w\rangle}\cdot(-1)^{2f(w)} = 2^n\sum_{w\in\mathbb{F}_2^n}(-1)^{\langle y,w\rangle}\\ &= \begin{cases} 2^{2n}, & \text{si } x=0. \\ 0, & \text{si } x\neq 0. \end{cases}\end{aligned}$$

La última igualdad se obtiene del Lema 3.2.3.

□

Como comentábamos, a partir de este lema (tomando $x = y$) se obtiene de forma inmediata la siguiente ecuación, conocida como *Identidad de Parseval*.

Teorema 3.2.5 (Identidad de Parseval)**.** *Para toda función Booleana en n variables f se verifica la siguiente igualdad*

$$\sum_{y\in\mathbb{F}_2^n} W_f(y)^2 = 2^{2n}.$$

Por último, resulta sencillo obtener una cota para el máximo en valor absoluto de W_f para una función f.

Teorema 3.2.6. *Para toda función Booleana en n variables se cumple que*

$$\max_{y\in\mathbb{F}_2^n}|W_f(y)| \geq 2^{\frac{n}{2}}.$$

Demostración.

Basta con despejar de

$$2^{2n} = \sum_{y\in\mathbb{F}_2^n} W_f(y)^2 \leq \sum_{y\in\mathbb{F}_2^n}|W_f(y)|^2 \leq 2^n \max_{y\in\mathbb{F}_2^n}|W_f(y)|^2.$$

□

3.3. Funciones bent

Las funciones bent fueron introducidas por Oscar Rothaus a finales de los años sesenta, y desde entonces muchos otros autores como John Dillon han dedicado parte de sus trabajos a ellas. Sus aplicaciones a la Criptografía y a la Teoría de Códigos son múltiples, lo que hace que no sea de extrañar que aún hoy muchos otros investigadores como Natalia Tokareva o Sihem Mesnager sigan centrándose en su estudio.

A lo largo de esta sección daremos a conocer lo que se conoce como una función bent, y enunciaremos varios de los resultados clásicos que pueden encontrarse en la literatura, así como algunas de sus caracterizaciones y representaciones equivalentes. Muchos de estos resultados son teoremas clásicos, motivo por el cual no resulta fácil encontrar sus demostraciones en la literatura, la mayoría de los autores simplemente los enuncian. Es por ello que las pruebas de teoremas y proposiciones como 3.3.3, 3.3.7, 3.3.8 y 3.3.12 son de aportación propia.

3.3.1. Definición y propiedades de las funciones bent

Comenzamos definiendo la noción de *no linealidad* de una función Booleana. Ya mencionamos su importancia anteriormente.

Definición 3.3.1. *La no linealidad de una función Booleana en n variables f se define como la distancia de Hamming de f al conjunto de funciones afines, es decir,*

$$N_f := \min_{a\in\mathbb{F}_2^n, b\in\mathbb{F}_2} \delta(f, \ell_{a,b}),$$

donde

$$\delta(f,g) = \#\{x \in \mathbb{F}_2^n : f(x) \neq g(x)\}, \quad \ell_{a,b} := \langle a, x\rangle \oplus b.$$

En términos de Teoría de Códigos, la no linealidad de una función f es igual a la distancia mínima de un código Reed-Muller $\mathcal{R}(1,n)\cup\mathcal{R}(f+\mathcal{R}(1,n))$. En la Sección 4.1.2 hablaremos en más detalle de los Códigos Reed Muller.

Ejemplo 3.3.2. *Determinemos la no linealidad de una función concreta, la de la función $f(x_1,x_2,x_3) = x_1 \oplus x_2x_3$. Al evaluar f en todo $\mathbb{F}_2^3$ en orden lexicográfico inverso obtenemos el vector* (00011110)*, de peso* 4*. Dado que estamos trabajando en una dimensión baja, no resulta costoso calcular los vectores correspondientes a todas las funciones afines en $\mathbb{F}_2^3$:*

0	00000000	1	11111111
x_1	00001111	$x_1 \oplus 1$	11110000
x_2	00110011	$x_2 \oplus 1$	11001100
x_3	01010101	$x_3 \oplus 1$	10101010
$x_1 \oplus x_2$	00111100	$x_1 \oplus x_2 \oplus 1$	11000011
$x_1 \oplus x_3$	01011010	$x_1 \oplus x_3 \oplus 1$	10100101
$x_2 \oplus x_3$	01100110	$x_2 \oplus x_3 \oplus 1$	10011001
$x_1 \oplus x_2 \oplus x_3$	01101001	$x_1 \oplus x_2 \oplus x_3 \oplus 1$	10010110

Como f no es afín y su peso es par, tenemos que $N_f \geq 2$, y dado que $\delta(f, x_1) = 2$, concluimos que $N_f = 2$.

Pasamos a definir lo que es una función bent. Previo a ello observamos que

$$\begin{aligned} W_f(a) &= \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle x,a \rangle \oplus f(a)} \\ &= \#\{x \in \mathbb{F}_2^n : \langle x, a\rangle \oplus f(a) = 0\} - \#\{x \in \mathbb{F}_2^n : \langle x, a\rangle \oplus f(a) \neq 0\} \\ &= \#\{x \in \mathbb{F}_2^n : \langle x, a\rangle \oplus f(a) = 0\} - \big(2^n - \#\{x \in \mathbb{F}_2^n : \langle x, a\rangle \oplus f(a) = 0\}\big) \\ &= 2^n - 2\#\{x \in \mathbb{F}_2^n : \langle x, a\rangle \oplus f(a) = 0\} \\ &= 2^n - 2\delta(f, \ell_{a,0}). \end{aligned}$$

Despejando se obtiene que

$$\delta(f, \ell_{a,0}) = 2^{n-1} - \frac{1}{2} W_f(a).$$

De forma análoga se sigue también que

$$\delta(f, \ell_{a,1}) = 2^{n-1} - \frac{1}{2} W_f(a).$$

A partir de estas dos igualdades se deduce el siguiente teorema.

Teorema 3.3.3. *Si f es una función Booleana en n variables, entonces*

$$N_f = 2^{n-1} - \frac{1}{2} \max_{y \in \mathbb{F}_2^n} |W_f(y)|.$$

Demostración.

Basta ver que

$$\begin{aligned} N_f &= \min_{a \in \mathbb{F}_2^n, b \in \mathbb{F}_2} \delta(f, \ell_{a,b}) \\ &= \min_{a \in \mathbb{F}_2^n} \{\delta(f, \ell_{a,0}), \delta(f, \ell_{a,1})\} \\ &= \min_{a \in \mathbb{F}_2^n} \left(2^{n-1} - \frac{1}{2} W_f(a)\right) \\ &= 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{F}_2^n} |W_f(a)|. \end{aligned}$$

□

A partir del teorema anterior y de la desigualdad de Parseval se deduce que $N_f \leq 2^{n-1} - 2^{\frac{n}{2}-1}$.

Dada una función Booleana se dice que su no linealidad es máxima si N_f alcanza el máximo valor posible. Es evidente entonces que:

Teorema 3.3.4. *Si n es par, entonces el máximo valor posible de la no linealidad es $2^{n-1} - 2^{\frac{n}{2}-1}$.*

Si n es impar, entonces el máximo valor es aún desconocido en general.

Definición 3.3.5. *Una función Booleana f en n variables (con n par) se dice que es una función bent si su no linealidad es máxima, por tanto es $N_f = 2^{n-1} - 2^{\frac{n}{2}-1}$.*

Las funciones bent han sido ampliamente estudiadas, y es por ello que cuentan con no pocas caracterizaciones. De hecho, es habitual que se dé la siguiente como definición de función bent.

Definición 3.3.6. *Una función Boolena en n variables f (con n par) se dice que es bent si para todo $y \in \mathbb{F}_2^n$ se tiene que $W_f(y) = \pm 2^{\frac{n}{2}}$.*

Naturalmente:

Proposición 3.3.7. *Las definiciones 3.3.5 y 3.3.6 son equivalentes.*

Demostración.

Si f es bent, por el Teorema 3.3.3, $\max^2_{y \in \mathbb{F}_2^n} |W_f(y)| = 2^n$, y por la desigualdad de Parseval, $2^n = \frac{1}{2^n} \sum_{y \in \mathbb{F}_2^n} |W_f(y)|^2$. Necesariamente $|W_f(y)| = 2^{\frac{n}{2}}$, pues de esta forma y sólo de esta forma se tendría que $\frac{1}{2^n} \sum_{y \in \mathbb{F}_2^n} |W_f(y)|^2 = \frac{1}{2^n} 2^n \cdot (2^{\frac{n}{2}})^2 = 2^n$. El recíproco es inmediato a partir del Teorema 3.3.3.

□

Del mismo modo que definimos las funciones Booleanas vectoriales, las funciones bent vectoriales son aquellas $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ dadas por $f(x) = (f_i(x), \ldots, f_m(x))$ en las que todas las f_i son funciones bent.

Una propiedad particularmente interesante de las funciones bent es la siguiente.

Proposición 3.3.8. *Una función Booleana en n variables f (con n par) es bent si y sólo si para todo $y \in \mathbb{F}_2^n$ con $y \neq 0$, su derivada respecto de y $D_y f(x) = f(x) \oplus f(x \oplus y)$ está equilibrada.*

Demostración.

Según la observación posterior al Teorema 3.3.12, la matriz $H_{\hat{f}} = (\hat{h}_{x,y})_{x,y}$ donde $\hat{h}_{x,y} = (-1)^{f(x \oplus y)}$ es una matriz de Hadamard, luego

$$\sum_{z \in \mathbb{F}_2^n} (-1)^{f(x \oplus z) \oplus f(z \oplus y)} = 0 \text{ para cada } x \neq y \in \mathbb{F}_2^n.$$

Cuando z recorre $\mathbb{F}_2^n$, $z \oplus y$ también lo hace. Por tanto, podemos reescribir lo anterior como

$$\sum_{z \in \mathbb{F}_2^n} (-1)^{f(x \oplus y \oplus z) \oplus f(z)} = 0 \text{ para cada } x \neq y \in \mathbb{F}_2^n,$$

lo que equivale a que $Df_{x \oplus y}$ está equilibrada. El recíproco se obtiene invirtiendo este razonamiento.

□

El interés detrás de esta propiedad radica en la relación existente entre la derivada y la autocorrelación de una función Boolena.

Definición 3.3.9. *Sea $f : \mathbb{F}_2^n \to \mathbb{F}_2$ una función Booleana, y sea $x \in \mathbb{F}_2^n \setminus \{0\}$. Se define el coeficiente de autocorrelación de f en x mediante*

$$C_f(x) = \sum_{y \in \mathbb{F}_2^n} (-1)^{f(x \oplus y) \oplus f(x)}$$

A partir de la Proposición 3.3.8 se deduce por tanto que una función Booleana es bent si y sólo si todos sus coeficientes de autocorrelación (excepto el asociado a 0) son nulos. Una baja autocorrelación da buenas garantías frente al criptoanálisis diferencial, lo que suma otro punto a favor de las funciones bent.

No obstante, uno de los aspectos que sacrifican las funciones bent a cambio de tener máxima no linealidad es que nunca están equilibradas. Esto se debe a que el peso de Hamming de cualquier función bent es siempre $2^{n-1} \pm 2^{\frac{n}{2}-1}$, lo que es inmediato a partir de $\omega_H(f) = \delta(f, 0) = \delta(f, \ell_{0,0}) = 2^{n-1} - \frac{1}{2} W_f(0) = 2^{n-1} - \frac{1}{2}(\pm 2^{\frac{n}{2}}) = 2^{n-1} \pm 2^{\frac{n}{2}-1}$. En consecuencia, existe una relación estadística (un *bias*) entre el texto claro y texto cifrado mediante un generador pseudoaleatorio que utilice una función bent en su modelo combinador. En estas circunstancias se presenta debilidad frente a ciertos ataques como el ataque algebraico rápido.

Otro de los sacrificados es la inmunidad a la correlación, ya que todos los coeficientes de la transformada de Walsh Hadamard son no nulos. Ver Teorema 3.1.10.

Mostramos a continuación un ejemplo clásico de función bent.

Ejemplo 3.3.10. *Sea $n \geq 4$ par, la función dada por $f(x) = x_1 x_2 \oplus x_3 x_4 \oplus \ldots \oplus x_{n-1} x_n$ es un ejemplo clásico de función bent. Para ver que efectivamente f es bent tomamos $x' = (x_1, x_3, \ldots, x_{n-1})$, $x'' = (x_2, x_4, \ldots, x_n)$ dos vectores de $\frac{n}{2}$ coordenadas cada uno y con las componentes pares e impares, respectivamente. Si x es el vector que resulta de unir x' y x'', entonces $f(x) = f(x', x'') = \langle x', x'' \rangle$. Dado $y \in \mathbb{F}_2^n$ y aplicando el Lema 3.2.3 observamos que*

$$\begin{aligned} W_f(y) &= \sum_{x\in\mathbb{F}_2^n}(-1)^{\langle x,y\rangle\oplus f(x)} = \sum_{x',x''\in\mathbb{F}_2^{\frac{n}{2}}}(-1)^{\langle x',y'\rangle\oplus\langle x'',y''\rangle\oplus\langle x',x''\rangle} \\ &= \sum_{x'\in\mathbb{F}_2^{\frac{n}{2}}}(-1)^{\langle x'',y''\rangle}\cdot\sum_{x',x''\in\mathbb{F}_2^{\frac{n}{2}}}(-1)^{\langle x',y'\oplus x''\rangle} = \sum_{x''=y'}(-1)^{\langle x'',y''\rangle}\cdot 2^{\frac{n}{2}} \\ &= (-1)^{\langle y',y''\rangle}\cdot 2^{\frac{n}{2}} = \pm 2^{\frac{n}{2}}, \end{aligned}$$

luego $|W_f(y)| = 2^{\frac{n}{2}}$*, lo que prueba que* f *es bent.*

Comentamos a continuación varias propiedades elementales de las funciones bent. La mayoría de ellas puede encontrarse en [Mes16] o en [Tok15].

- El grado algebraico de cualquier función bent sobre $\mathbb{F}_{2^n}$ es, como mucho, $\frac{n}{2}$.
- El conjunto de las funciones bent en n variables es invariante por la acción del grupo afín general de $\mathbb{F}_{2^n}$ y la adicción de funciones Booleanas afines en n variables. Más concretamente, si f y f' son dos funciones Boolenas en n variables linealmente equivalentes (es decir, podemos pasar de una a otra mediante un $\mathbb{F}_2$-automorfismo lineal de $\mathbb{F}_{2^n}$), entonces f es bent si y sólo si f' lo es.
- El grupo de automorfismos del conjunto de funciones bent (es decir, el grupo de permutaciones de las variables π tal que $f\circ\pi$ es bent para toda función bent f) es el grupo afín general, esto es, el grupo de los automorfismos lineales compuestos con traslaciones. Además, si f es bent y ℓ es afín, entonces $f+\ell$ es bent.
- Dada una función Booleana f podemos definir su función Booleana dual $\tilde{f}$ mediante mediante la relación $W_f(x) = 2^{\frac{n}{2}}(-1)^{\tilde{f}}$. En tal caso, si f es bent, entonces $\tilde{f}$ también lo es. Además, pudiera ocurrir que $f=\tilde{f}$, en cuyo caso se dice que f es *autodual*.

3.3.2. Representaciones equivalentes de las funciones bent

Una de las principales riquezas de las funciones bent es la abundancia de representaciones equivalentes que presentan, resulta casi *mágico*. Presentamos a continuación varias de ellas.

Matrices de Hadamard

Rothaus estudió la muy estrecha relación entre las matrices de Hadamard y las funciones bent. Recordamos la definición de matriz de Hadamard.

Definición 3.3.11. *Sea* H *una matriz cuadrada de orden* n*. Decimos que* H *es una matriz de Hadamard si todas sus entradas son* ± 1 *y* $HH^t = nI_n$*, donde* H^t *es la matriz traspuesta de* H *e* I_n *denota a la matriz identidad de orden* n*.*

Si H es de Hadamard, puede observarse que al multiplicar todas las entradas de una fila o de una columna por -1, la matriz resultante también es de Hadamard. Mediante una serie de multiplicaciones de este tipo es posible transformar cualquier matriz de Hadamard en una en la que las entradas de la pimera fila y la primera columna son todas 1. Una matriz de este tipo se dice que está *normalizada*.

Tenemos el siguiente resultado, que fue probado por primera vez por Rothaus en [Rot76].

Teorema 3.3.12. *Sea f una función Booleana en n variables. Definimos la matriz $H_f = (h_{x,y})_{x,y}$ mediante $h_{x,y} = 2^{-\frac{n}{2}} W_f(x \oplus y)$ para cada $x, y \in \mathbb{F}_2^n$. Entonces f es bent si y sólo si H_f es una matriz de Hadamard.*

Demostración.

Supongamos que f es bent, en ese caso $W_f(x) = \pm 2^{\frac{n}{2}}$, de modo que todas las entradas de H_f son ± 1. Debemos ver ahora que $H_f H_f^t = 2^n I_{2^n}$, es decir, que el producto de dos filas de H_f es 2^{2n} cuando son la misma y es 0 cuando son distintas, pero esto se obtiene de forma inmediata a partir del Lema 3.2.4.

El recíproco resulta bastante directo, si H_f es una matriz de Hadamard entonces por las relaciones de ortogonalidad entre filas y columnas se dan las condiciones de la Proposición 3.3.7, y por tanto se deduce que f es bent.

□

De hecho, que f sea bent también equivale a que la matriz $H_{\hat{f}} = (\hat{h}_{x,y})_{x,y}$ sea de Hadamard. Aplicando el el Teorema 3.2.2 y el Lema 3.2.3 observamos que

$$\begin{aligned}
\sum_{y \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(x \oplus y)} &= \sum_{y \in \mathbb{F}_2^n} \frac{1}{2^n} \Big(\sum_{z \in \mathbb{F}_2^n} W_f(z) (-1)^{\langle x, z \rangle} \frac{1}{2^n} \sum_{w \in \mathbb{F}_2^n} W_f(w) (-1)^{\langle x+y, w \rangle} \Big) \\
&= \frac{1}{2^{2n}} \sum_{y \in \mathbb{F}_2^n} \Big(\sum_{z \in \mathbb{F}_2^n} W_f(z) (-1)^{\langle x, z \rangle} \sum_{w \in \mathbb{F}_2^n} W_f(w) (-1)^{\langle x, w \rangle} (-1)^{\langle y, w \rangle} \Big) \\
&= \frac{1}{2^{2n}} \sum_{y \in \mathbb{F}_2^n} \sum_{z \in \mathbb{F}_2^n} W_f(z)^2 (-1)^{\langle y, z \rangle} = \begin{cases} \frac{1}{2^{2n}} \sum_{y \in \mathbb{F}_2^n} 2^{2n} = 2^n, & \text{si } y = 0. \\ 0, & \text{si } y \neq 0. \end{cases}
\end{aligned}$$

Esto permite probar de forma muy sencilla que una función es bent si y sólo si su derivada $D_y f$ está equilibrada para cada $y \in \mathbb{F}_2^n$.

Ejemplo 3.3.13. *Sea $f(x_1, x_2) = x_1 x_2$. La imagen de f es $[0, 0, 0, 1]$ y la de W_f es $[1, 1, 1, -1]$. La matriz H_f correspondiente es*

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Es inmediato comprobar que efectivamente se trata de una matriz de Hadamard de orden 4.

Conjuntos diferencia

Sea G un grupo abeliano con $|G| = v < \infty$. Un subconjunto $D \subset G$ de cardinal k es llamado *conjunto diferencia* de parámetros (v, k, λ, n) si todo elemento no nulo $g \in G$ puede expresarse como $g = b - d$ de exactamente λ formas distintas, donde $b, d \in D$. Se define $n = v - k$. Una de las caracterizaciones clásicas de las funciones bent es través de los recién definidos conjuntos diferencia. Para este punto seguiremos el desarrollo que ofrece [CS09] en el quinto capítulo.

Comenzamos con el siguiente resultado.

Como cada uno de los $v - 1$ elementos no triviales de G ocurren λ veces entre los $k(k-1)$ elementos no triviales de D, debe ser $\lambda(v-1) = k(k-1)$.

La matriz de incidencia asociada a D es la $v \times v$ matriz Υ_D, dada por

$$\Upsilon_D(x, y) = \begin{cases} -1, & \text{si } x - y \in D. \\ 0, & \text{caso contrario.} \end{cases}$$

A partir de la definición de la matriz de incidencia se sigue lo siguiente:

Lema 3.3.14. *D es un conjunto diferencia de parámetros (v, k, λ, n) si y sólo si*

$$\Upsilon_D^2 = nI_v + \lambda \mathcal{J},$$

donde $\mathcal{J}$ es la $v \times v$ matriz cuyas entradas son todas 1.

De este hecho se deduce lo siguiente:

Lema 3.3.15. *Si D es un (v, k, λ, n)-conjunto diferencia, entonces su complementario $\bar{D} = G \setminus D$ es un conjunto diferencia de parámetros $(v, v - k, v - 2k + \lambda, n)$ en G.*

Existen unos conjuntos diferencia particularmente interesantes que sólo existen en grupos abelianos cuyo orden es un cuadrado.

Teorema 3.3.16. *D es un (v, k, λ, n)-conjunto diferencia con $v = 4n$ si y sólo si $\mathcal{J} - 2\Upsilon_D$ es una matriz de Hadamard.*

Estos conjuntos diferencia reciben el nombre de *conjuntos diferencia de Hadamard*.

El siguiente resultado fue establecido por Kesava Menon en [Men60].

Teorema 3.3.17. *Un conjunto diferencia de Hadamard tiene parámetros $(4N^2, 2N^2 - N, N^2 - N, N^2)$ o $(4N^2, 2N^2 + N, N^2 + N, N^2)$.*

Demostración.

De las relaciones $n = k - \lambda$, $k(k-1) = \lambda(v-1)$ y $v = 4n$ se sigue que

$$\begin{aligned} 0 &= k(k-1) - \lambda(v-1) = k^2 - k - (k-n)(4n-1) \\ &= k^2 - 4nk + n(4n-1) = (k-2n)^2 - n, \end{aligned}$$

lo que prueba el teorema.

□

Dos conjuntos diferencia D_1, D_2 en un grupo abeliano G se dice que son *equivalentes* si existe un automorfismo α de G tal que

$$D_1^\alpha = D_2 + g$$

para algún $g \in G$. Si esta relación se verifica para $D_1 = D_2 = D$, entonces α es llamado un *multiplicador* de D. Si además es de la forma $g \mapsto g^t$ con $t \in \mathbb{Z}$, entonces se dice que es un *multiplicador numérico*.

H.B. Mann y R.L. McFarland [Dil72] probaron que todo multiplicador de un conjunto diferencia debe dejar fija al menos una traslación de dicho conjunto. Denotamos por $M(D)$ al subgrupo de los multiplicadores de D en el grupo de automorfismos de G. Puede comprobarse que si $D_1^\alpha = D_2 + g$, entonces $M(D_1) = \alpha M(D_2)\alpha^{-1}$, es decir, D_1 y D_2 son isomorfos.

Teorema 3.3.18. *Una función Booleana en n variables f es una función Bent si y sólo si el conjunto $f^{-1}(1)$ (o $f^{-1}(0)$) es un conjunto diferencia en $\mathbb{Z}_{2^n}$. Los parámetros de dicho conjunto son $(2^n, 2^{n-1} \pm 2^{\frac{n}{2}-1}, 2^{n-2} \pm 2^{\frac{n}{2}-1}, 2^n - 2^{n-1} \pm 2^{\frac{n}{2}-1})$.*

Demostración.

Si $D = f^{-1}(1)$, vemos que la entrada (x, x) de la matriz $\mathcal{J} - 2\Upsilon_D$ son

$$\begin{cases} -1, & \text{si } x - y \in D. \\ 1, & \text{caso contrario.} \end{cases} = \begin{cases} -1, & \text{si } x \oplus y \in f^{-1}(1). \\ 1, & \text{caso contrario.} \end{cases} = \begin{cases} -1, & \text{si } f(x \oplus y) = 1. \\ 1, & \text{caso contrario.} \end{cases} = (-1)^{f(x \oplus y)}.$$

Como f es bent si y sólo si la matriz $H = ((-1)^{f(x \oplus y)})_{x,y}$ es una matriz de Hadamard, por el Teorema 3.3.16, la equivalencia queda probada.

□

Terminamos este apartado ilustrando lo visto en él a través de un ejemplo.

Ejemplo 3.3.19. *Sea $f : \mathbb{F}_2^4 \to \mathbb{F}_2$ dada por $f(x_1, x_2, x_3, x_4) = 1 \oplus x_1 x_2 \oplus x_1 x_3 \oplus x_1 x_4 \oplus x_2 x_3 \oplus x_2 x_4 \oplus x_3 x_4$. Puede comprobarse que $W_f(x) = \pm 4 = \pm 2^{\frac{4}{2}}$, de modo que f es bent. Si recorremos $\mathbb{F}_2^4$ ($\cong \mathbb{Z}_{16}$ como espacio vectorial) en orden lexicográfico tenemos que*

$$f \equiv 1110100010000001,$$

y por tanto, si $D = f^{-1}(1)$, entonces

$$D = \{f(0,0,0,0), (0,0,0,1), (0,0,1,0), (0,1,0,0), (1,0,0,0), (1,1,1,1)\}.$$

El conjunto D es un conjunto diferencia de parámetros $(v, k, \lambda, n) = (16, 6, 2, 4)$.

Diseños

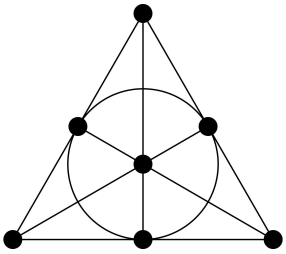
Los conjuntos diferencia están estrechamente relacionados con los *diseños* (esquemas de bloques). Damos su definición.

Definición 3.3.20. *Un diseño (o esquema de bloques) con parámetros (v, k, λ) es un sistema de subconjuntos de k-elementos (bloques) de un conjunto de v-elementos tal que cualquier par de elementos distintos está contenido en exactamente λ bloques.*

Además, decimos que un diseño es simétrico si el número de bloques es el mismo que el número de elementos (es decir, igual a v).

Varios ejemplos clásicos de diseños se encuentran en la geometría proyectiva, como es el caso del *plano de Fano*.

Ejemplo 3.3.21. *El plano de Fano se define como el plano proyectivo de dimensión* 2 *sobre $\mathbb{F}_2$, es decir, $\mathbb{P}^2(\mathbb{F}_2)$. Se trata de un diseño de parámetros $\nu = 7, k = 3$ y $\lambda = 1$ en el que los elementos del diseño son los puntos del plano y los bloques son las rectas.*



Es conocido que toda recta de $\mathbb{P}^2(\mathbb{F}_2)$ pasa por exactamente 3 *puntos y todo punto de $\mathbb{P}^2(\mathbb{F}_2)$ se encuentra en exactamente* 3 *rectas.*

La matriz de incidencia $A = (a_{ij})$ de un diseño simétrico es una $v \times v$ matriz binaria cuyas filas y columnas vienen dadas por bloques y elementos, respectivamente, y tal que $a_{ij} = 1$ si el elemento j pertenece al i-ésimo bloque, y $a_{ij} = 0$ en caso contrario.

Teorema 3.3.22. *Una función Booleana en n variables f es bent si y sólo si el sistema de conjuntos $D_z = \{D \oplus z : z \in \mathbb{F}_2^{n+1}\}$ es un diseño simétrico de parámetros $(2^{n+1}, 2^n, 2^{n-1})$, con $D = \{(x, f(x)) : x \in \mathbb{F}_2^n)\}$*

El rango de una función bent f es el rango de la matriz de incidencia del diseño correspondiente.

Existen más caracterizaciones de las funciones bent haciendo uso de grafos fuertemente regulares, la transformada de Fourier normalizada y los rectángulos bent. Para más información puede consultarse el capítulo 6 de [Tok15].

3.3.3. Secuencias bent

También es posible tratar el concepto de bent desde otro punto de vista, a partir de las conocidas como *secuencias bent*. Esto corta a nuestros propósitos de forma muy tangencial, por lo que sólo damos una pequeña pincelada.

Dado $n \in \mathbb{N}$, definimos la matriz $S_n = ((-1)^{\langle x,y \rangle})_{x,y}$, que recibe el nombre de matriz de Sylvester. Es inmediato comprobar que se trata de una matriz de Hadamard. Vemos que es posible expresar la transformada de Walsh Hadamard de una función Booleana f a partir de esta matriz y de su correspondiente función signo $\hat{f}$ mediante

$$W_f = S_n \hat{f}$$

Sea H una matriz de Hadamard normalizada de orden n, que habitualmente se tratará de la matriz de Sylvester de orden n, S_n. Nos planteamos el problema de resolver el sistema

$$HX = Y,$$

para ciertos $X, Y \in \mathbb{F}_2^n$. De existir tal X, recibe el nombre de *secuencia bent*. Además, si $X = Y$, entonces f es autodual. Vemos que en tal caso el vector X sería un autovector asociado al autovalor 1. Si H no es una matriz de Sylvester S_n entonces debemos suponer de antemano que el valor 1 se encuentra en el espectro de H y que la dimensión del espacio de autovectores asociado no es demasiado grande. En general esto se considera un problema abierto. Si el lector tiene más interés, puede consultar [SL23].

3.4. Aplicaciones de las funciones bent en criptografía

Ya hablamos de los usos habituales de las funciones Booleanas en Criptografía, en concreto nos centramos en los cifradores a flujo y en la importancia que tiene la no linealidad de la función utilizada en el modelo de filtro y en el modelo combinador.

Comentamos también un caso especialmente interesante que muestra el peso que tiene la no linealidad en ciertos criptosistemas. En concreto nos referimos al caso del DES. Matsui probó en 1993 que este criptosistema no es resistente al criptoanálisis lineal [Mat94]. Esto se debe a la baja no linealidad de sus S-boxes (que son ciertas funciones Boolenas vectoriales). El DES cuenta con 8 S-boxes, cada una recibe 6 bits y devuelve 4 bits, y las ocho de ellas están perfectamente determinadas y son de acceso público. La gran fisura de este criptosistema se encuentra en la quinta S-box, S_5, ya que presenta una no linealidad muy baja. En concreto, si $S_5(x_1, x_2, x_3, x_4, x_5, x_6) = (y_1, y_2, y_3, y_4)$, para 52 de entre las 64 entradas posibles se verifica la siguiente relación: $x_2 = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus 1$. Como 52/64 se diferencia de 1/2 en 20/64, esto se traduce en que el bias de S_5 es $20/64 = 0{,}3125$. Visto de otra forma, la función Booleana dada por $f(x_1, x_2, x_3, x_4, x_5, x_6) = y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus 1$ puede aproximarse por la función lineal $g(x_1, x_2, x_3, x_4, x_5, x_6) = x_2$. No debería sorprendernos que la no linealidad de f

(la distancia de f a g) sea $N_f = 10$, muy por debajo de la máxima no linealidad posible en 6 variables, de $2^{6-1} - 2^{\frac{6}{2}-1} = 28$. Mitsui probó que con 2^{43} parejas de textos claros y textos cifrados conocidos es posible obtener la clave utilizada (de 56 bits) con una probabilidad de 0,85.

Las funciones bent, por su propia definición, pueden proporcionar resistencia a los ataques de este tipo, un ejemplo de esto es el cifrador CAST, que detallamos a continuación. No obstante, debemos recordar que las funciones bent sacrifican otras propiedades criptográficas interesantes: nunca están equilibradas y no son inmunes a la correlación.

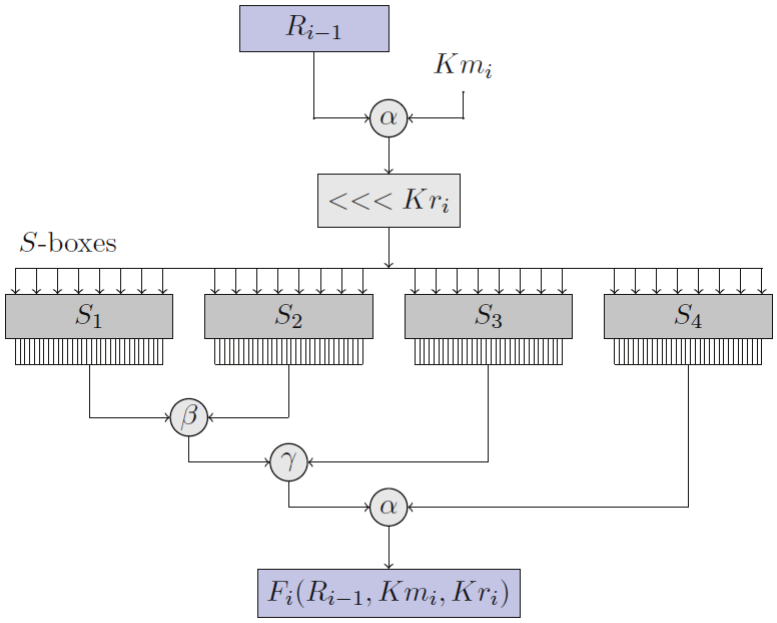
3.4.1. Algunos usos de las funciones bent en Criptografía

Presentamos a continuación tres criptosistemas en los que se han empleado funciones bent, un cifrador a bloque, un cifrador a flujo, y por último, una función hash.

El cifrador en bloque CAST

El cifrador en Bloque CAST fue introducido en 1997 por Carlisle Adams y Stafford Tavares [AT], y fue aprobado por el gobierno canadiense como el sustituto oficial del DES.

Este esquema de cifrado consiste en una red de Feistel clásica de 16 rondas que encripta bloques de longitud 64. La longitud de la clave es de 128 bits, y la función F_i de cada ronda depende del número de la ronda. A continuación mostramos el esquema de una ronda



$$\text{donde } \alpha, \beta, \gamma \text{ denotan: } \begin{array}{l} \boxplus, \oplus, \boxminus \text{ para } i = 1, 4, 7, 10, 13, 16. \\ \oplus, \boxminus, \boxplus \text{ para } i = 2, 5, 8, 11, 14. \\ \boxminus, \boxplus, \oplus \text{ para } i = 3, 6, 9, 12, 15. \end{array}$$

Como vemos, existen cuatro S-boxes en la función de cada ronda, cada una de las cuales convierte 8 bits en 32 bits. Es habitual representar la k-esima S-box S_k como la colección de las 32 funciones Booleanas $f_j^{(k)}$:

$$S_k(x_1, \ldots, x_8) = (y_1, \ldots, y_{32}) \text{ siendo } y_j = F_j^{(k)}(x_1, \ldots, x_8) \text{ para } j \in \{1, \ldots 32\}.$$

Lo que nos interesa de CAST es que ha sido diseñado de forma que todas sus funciones $f_j^{(k)}$ sean bent. Más aún, cualquier combinación lineal de las funciones bent $f_j^{(k)}$ para un k fijo tiene una alta no linealidad. Para más detalles puede consultarse [Ada97].

El cifrador en flujo Grain

Grain es un cifrador en flujo diseñado por Hell et al. [HJM06] y que forma parte del proyecto eSTREAM. Fue concebido principalmente para entornos con restricción de hardware. Toma como entrada una clave de 80 bits y un vector inicial de 64 bits, y consiste en un NFSR (*nonlinear feedback shift register*) y un LFSR (*linear feedback shift register*), ambos con una longitud de 80 bits.

El polinomio de retroalimentación no lineal del NSFR, $g(x)$, se construye como la suma de un función lineal y una función bent. Esto da lugar a una alta resiliencia además de una notable no linealidad. Se toma la función bent cuadrática más simple que existe en 14 variables:

$$b(x) = x_0x_1 \oplus x_2x_3 \oplus x_4x_5 \oplus x_6x_7 \oplus x_8x_9 \oplus x_{10}x_{11} \oplus x_{12}x_{13},$$

siendo su linealidad 8128. Para aumentar su resiliencia se añaden cinco términos lineales, lo que da como resultado una función equilibrada con una resiliencia de 4 y una no linealidad de $2^5 \cdot 8128 = 260096$.

La función Booleana resultante para el NFSR de Grain tiene 19 variables y su implementación en el hardware es poco costosa. Las mejores aproximaciones lineales proporcionan una función lineal en 19 variables que contiene al menos todos los términos lineales de la función en cuestión. Existen 2^{14} funciones de tales características, y tienen bias $\varepsilon_g = 2^{-8}$.

Funciones Bent en HAVAL

La función hash HAVAL fue diseñada por Zheng et al. en 1993. Se trata de una función Booleana vectorial que transforma bloques de 1024 bits en resúmenes de 256 bits. El algoritmo de actualización H que la caracteriza procesa un bloque en tres, cuatro, o cinco *pases*, que son especificados en el campo *PASS* del último bloque. Cada uno de estos pases $H_1, \ldots, H_5$ tiene 32 rondas de operaciones, y en la construcción de cada uno de ellos se utilizan distintas funciones Booleanas en 7 variables f_i. Estas

funciones tienen una importancia crucial en el algoritmo del *hashing*, y se obtuvieron de entre las cuatro funciones bent afínmente no equivalentes en 6 variables:

$$g_1(x_6,\ldots,x_1) = x_1x_4 \oplus x_2x_5 \oplus x_3x_6.$$

$$g_2(x_6,\ldots,x_1) = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_2 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_5 \oplus x_4x_5.$$

$$g_3(x_6,\ldots,x_1) = x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6.$$

$$\begin{aligned} g_4(x_6,\ldots,x_1) = {} & x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_4 \oplus x_3x_5 \oplus x_3x_6 \\ & \oplus x_4x_5 \oplus x_4x_6. \end{aligned}$$

Las funciones f_i para $1 \leq i \leq 4$ se obtienen a partir de las g_i como sigue:

$$f_i(x_6,\ldots,x_0) = g_i(x_6,\ldots,x_1) \oplus x_0x_i \oplus x_0.$$

La quinta función f_5 viene dada por

$$f_5(x_6,\ldots,x_0) = g_1(x_6,\ldots,x_1) \oplus x_0x_1x_2x_3 \oplus x_0x_5 \oplus x_0.$$

Las funciones g_i tienen una no linealidad de $2^6 - 2^3 = 56$, que es el máximo posible para funciones Booleanas en 7 variables. Todas estas funciones son linealmente no equivalentes. Destacar que todas las propiedades mencionadas de las f_i son importantes para que HAVAL sea resistente a los ataques lineales.

3.5. Generalizaciones de las funciones bent

Dado que las funciones bent nunca están equilibradas resulta natural estudiar ciertas súperclases de funciones bent que mantengan una no linealidad lo suficientemente alta, pero que también exhiban otras propiedades criptográficas interesantes. Dedicamos esta última sección de nuestro tercer capítulo a estudiar algunas de las generalizaciones de las funciones bent que pueden encontrarse en la literatura, en concreto presentamos algunas de las que aparecen en los capítulos 15 y 16 de [Mes16]. No pretendemos profundizar demasiado en ninguna de ellas, nuestro objetivo es simplemente presentarlas y dar una idea general de ellas. Si el lector tiene más interés, puede consultar la bibliografía indicada. Comenzamos con las funciones parcialmente bent.

3.5.1. Funciones parcialmente bent

Una de estas súperclases, cuyas funciones presentan alta no linealidad y además están equilibradas, es la de las funciones que logran $\mathcal{N}_{\Delta_f} \cdot \mathcal{N}_{W_f} \geq 2^n$, donde $f : \mathbb{F}_2^n \to \mathbb{F}_2$ es una función Booleana en n variables y

$$\mathcal{N}_{\Delta_f} = \#\{b \in \mathbb{F}_2^n : C_f(b) \neq 0\},$$

$$\mathcal{N}_{W_f} = \#\{b \in \mathbb{F}_2^n : W_f(b) \neq 0\}.$$

Más aún, $\mathcal{N}_{\Delta_f} \cdot \mathcal{N}_{W_f} = 2^n$ si y sólo si para cada $b \in \mathbb{F}_2^n$ la derivada $D_b f$ está equilibrada o es constante. Esta propiedad es equivalente a que existan dos subespacios lineales E (de dimensión par) y E' de $\mathbb{F}_2^n$ cuya suma directa sea $\mathbb{F}_2^n$, y dos funciones Booleanas g bent en E y h afín en E' tales que para todo $x \in E$ y para todo $y \in E'$ $f(x \oplus y) = g(x) \oplus h(y)$.

Observemos que la relación $\mathcal{N}_{\Delta_f} \cdot \mathcal{N}_{W_f} \geq 2^n$ de alguna forma expresa una cierta compensación entre el número de derivadas no equilibradas y la cantidad de valores no nulos de su transformada de Walsh Hadamard.

Definición 3.5.1. *Las funciones que cumplen esta condición se las conoce como funciones parcialmente bent.*

Preneel conjeturó la existencia de tales funciones en [PGV91], y más tarde fue probada por Carlet en [Car93]. Esta familia presenta ciertas propiedades bastante interesantes: las funciones de este tipo pueden estar equilibradas y ser altamente no lineales, además, su espectro de Walsh Hadamard es más fácil de calcular que el de una función bent, y tienen un buen orden de resiliencia, lo que las hace buenas candidatas para ser usadas en cifradores a flujo y en cifradores en bloque.

La clase de las funciones *plateaud*, que veremos a continuación, es una extensión natural de la clase de las funciones parcialmente bent. Además, se han estudiado otras tantas generalizaciones, por ejemplo sobre el anillo $\mathbb{Z}_N$ para un cierto natural N. De hecho, algunos de los resultados establecidos por Carlet para las funciones parcialmente bent han podido generalizarse al caso sobre $\mathbb{Z}_N$, algunos de los cuales pueden encontrarse en [WZ07]. En concreto, se prueba que si N es primo, ciertas funciones funciones parcialmente bent generalizadas pueden descomponerse como suma de una función afín y una función bent. Esta descomposición facilita la construcción de tales funciones. También ha sido estudiada la generalización al caso de característica $p \neq 2$. Se definen las funciones *p-arias parcialmente bent* como aquellas $f : \mathbb{F}_p^n \to \mathbb{F}_p$ tales que para todo $a \in \mathbb{F}_p$ la derivada de f en a; $D_f(a) = f(x + a) - f(x)$ está equilibrada o es constante, es decir, cada valor de $\mathbb{F}_p$ se toma p^{n-1} veces. Para más información también puede consultarse [CAT14].

3.5.2. Funciones plateaud

Las llamadas funciones plateaud fueron introducidas por Zheng y Zhang en 1999. Los primeros en estudiarlas fueron Carlet y Prouff, quienes las identificaron como buenas candidatas para el diseño de funciones criptográficas.

Definición 3.5.2. *Una función Booleana $f : \mathbb{F}_2^n \to \mathbb{F}_2$ se dice que es r-plateaud si los valores de la transformada de Walsh Hadamard se encuentran en el conjunto $\{0, 2^{\frac{n+r}{2}}$ para algún $0 \leq r \leq n$ fijo.*

Las funciones plateaud proporcionan protección frente los ataques de correlación rápida y el criptoanálisis lineal. Además, esta familia de funciones exhibe otras características interesantes, como una alta no linealidad, resiliencia, baja autocorrelación aditiva y un alto grado algebraico, entre otras [ZZ99]. Incluyen también tres clases de

funciones Booleanas muy significativas: las funciones bent, las funciones near-bent, y las funciones semi-bent. En concreto, las funciones bent son las 0-plateaud (como es evidente en base a la Definición 3.5.2), las near-bent son las 1-plateaud, y las semi-bent son las 2-plateaud. Es conocido que en $\mathbb{F}_{2^n}$ las funciones 0-plateaud y las 2-plateaud existen cuando n es par, mientras que las 1-plateaud existen cuando n es impar.

Centrándonos en las 2-plateaud, o semi-bent, fueron introducidas en 1994 por Chee, Lee y Kim [CLK94], aunque ya habían sido estudiadas bajo el nombre de *funciones Booleanas tres-evaluadas casi óptimas* en [CCF01]. Uno de sus puntos fuertes es que, al contrario que las funciones bent, estas sí están equilibradas y son resilientes. Además, poseen una baja autocorrelación y la máxima no linealidad alcanzable entre las funciones plateaud equilibradas. Por otro lado, su uso no se adscribe sólo a la criptografía, también son ampliamente utilizadas en el acceso múltiple por división de código en sistemas de comunicación.

3.5.3. Funciones bent de rotación simétrica y funciones bent idempotentes

Las funciones de rotación simétrica fueron introducidas por Filiol y Fountain en [FF98] y en [FF99] bajo el nombre de funciones idempotentes. Más tarde, Pieprzyk y Qu les darían su nombre definitivo en [PQ99].

Definición 3.5.3. *Una función bent se dice que es simétrica si la salida no se ve alterada por ninguna permutación de los datos de entrada.*

Es inmediato entonces que una función de estas características toma el mismo valor en vectores que tienen el mismo peso de Hamming.

Definición 3.5.4. *Una función bent se dice que es simétrica por rotación (RS) si la salida no se ve alterada por ninguna permutación cíclica de los datos de entrada.*

Una de las motivaciones principales para el estudio de este tipo de funciones es que parece elevada la probabilidad de encontrar funciones interesantes de este tipo mediante una búsqueda aleatoria. De hecho, puede verse a simple vista que el espacio de búsqueda de estas funciones tiene tamaño del orden de $2^{\frac{2^n}{n}}$, mucho menor que el espacio total, de tamaño 2^{2^n}. Esto permite buscar funciones RS para un número mayor de variables. Además, investigaciones recientes muestran que la clase de las funciones Booleanas de rotación simétrica es potencialmente rica en funciones con aplicaciones criptográficas. Por ejemplo, Kavut, Maitra y Yucel [KMY07] han encontrado una función de 9 variables con una no linealidad de 241, lo que ha resulto un problema que estuvo abierto durante treinta años, (recordemos que las funciones bent se definían siempre para un número par de variables).

Por otro lado, el hecho de que las funciones Booleanas de rotación simétrica sean invariantes por la acción del grupo cíclico las hace propensas a tener una inmunidad algebraica óptima.

Encontrar funciones bent de rotación simétrica no resulta una tarea sencilla,

aunque se conocen varias familias infinitas de funciones de este tipo, muchas de las cuales se incluyen en clase de las de Maiorana–McFarland [Car13]. Una de ellas es, por ejemplo, la dada por

$$\sum_{i=1}^{n-1}(x_i x_{t+1} x_{m+i} + x_i x_{t+i}) + \sum i = 0^{m-1} x_i x_{m+i}, \text{ con } \frac{m}{\gcd(m,t)} \text{ impar [GZL12].}$$

Definimos a continuación las funciones idempotentes.

Definición 3.5.5. *Una función Booleana en n variables f se dice que es idempotente si* $f(x) = f(x^2)$ *para cada* $x \in \mathbb{F}_2^n$.

Las funciones idempotentes y las funciones RS están íntimamente ligadas. Más explícitamente, dada cualquier función Booleana $f(x)$ sobre $\mathbb{F}_{2^n}$, y dada cualquier base normal $\{\alpha, \alpha^2, \ldots, \alpha^{2^{n-1}}\}$ de $\mathbb{F}_{2^n}$, la función dada por

$$(x_0, \ldots, x_{n-1}) \longmapsto f\left(\sum_{i=0}^{n-1} x_i \alpha^{2^i}\right)$$

es RS si y sólo si f es idempotente. No obstante, conocer una clase infinita de funciones idempotentes no garantiza poder obtener una familia infinita de funciones RS, ya que no existe ninguna expresión válida para la descomposición de $f\left(\sum_{i=0}^{n-1} x_i \alpha^{2^i}\right)^j$ sobre la base normal $\{\alpha, \alpha^2, \ldots, \alpha^{2^{n-1}}\}$ excepto cuando j es una potencia no negativa de 2.

Curiosamente, las funciones bent univariantes y las funciones de rotación simétrica tienen ciertas similitudes:

- Pueden tener una estructura y representación simples.
- Operar con ellas puede ser bastante rápido, lo que las hace más adecuadas para ciertas aplicaciones.

Dos ejemplos de funciones bent idempotentes cuadráticas son la dada por $f(z) = Tr^m(z^{2m+1})$ y la dada por $g(z) = Tr^m(z^{2m+1}) + \sum_{i=1}^{m-1} Tr^n(z^{2^i+1})$.

Por último, cabe mencionar que las funciones de rotación simétrica y las funciones idempotentes son realmente infrecuentes en comparación con las funciones bent generales.

3.5.4. Funciones negabent

Sea $f : \mathbb{F}_2^n \to \mathbb{F}_2$ una función Boolena, y consideremos su imagen sobre un $2 \times \ldots \times 2$ array n-dimensional de la forma $\hat{f} = (-1)^{f(x)}$, $x \in \mathbb{F}_2^n$. Por ejemplo, para $n = 2$ y $f(x) = x_0 x_1$ es

$$\hat{f} = (-1)^{f(x)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

La transformada de Fourier periódica n-dimensional de $\hat{f}$, $\hat{F}$, se obtiene mediante la acción de $H^{\otimes n}$ sobre $\hat{f}$, donde $H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

Una función Booleana se dice que es bent si todos los elementos de la transformada de Fourier periódica de $\hat{f}$ tienen la misma magnitud. En nuestro ejemplo, f es bent, ya que $\hat{F} = H^{\otimes 2}\hat{f} = \begin{bmatrix} 2 & 2 \\ 2 & -2 \end{bmatrix}$.

Podríamos tratar de generalizar esto de alguna forma. Parece razonable considerar la transformada de Fourier negaperiódica, $\tilde{F}$, de $\hat{f}$, dada por la acción de $N^{\otimes n}$ sobre $\hat{f}$, siendo $N = \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix}$ la matriz Negahadamard. Una función Boolena se dice que es negabent si los elementos de su transformada de Fourier negaperiódica tienen la misma magnitud.

Por ejemplo, la función f dada por $f(x) = x_0x_1$ no es negabent, ya que $\tilde{F} = N^{\otimes 2}\hat{f} = \begin{bmatrix} 2+2i & 0 \\ 0 & 2-2i \end{bmatrix}$. En cambio, la función dada por $f(x) = x_0 + x_1$ sí es negabent, pues $= N^{\otimes 2}\hat{f} = \begin{bmatrix} -2i & 2 \\ 2 & 2i \end{bmatrix}$.

Vemos que tanto H como N son matrices unitarias salvo por un factor de $\dfrac{1}{\sqrt{2}}$.

Del mismo modo que resulta interesante construir funciones bent (o que estén cerca de serlo), también lo es encontrar funciones negabent. Es fácil ver que todas las funciones afines son negabent. Una cuestión más desafiante es tratar de obtener funciones Booleanas que sean bent y negabent. Un ejemplo es $f : \mathbb{F}_2^4 \to \mathbb{F}_2$ dada por $f(x) = x_0x_1 + x_1x_2 + x_2x_3$. Existe una muy conveniente correspondencia entre las funciones bent y las funciones negabent [Mes16]:

Lema 3.5.6. *Sea $f : \mathbb{F}_2^n \to \mathbb{F}_2$ una función bent y sea $\sigma(x) = \sum_{i<j} x_ix_j$. Entonces $f(x) + \sigma(x)$ es una función negabent. Recíprocamente, si $f(x)$ es negabent, entonces $f(x) + \sigma(x)$ es bent cuando n es par (y casi bent para n impar).*

Supongamos que $\hat{f}$ describe los coeficientes de un polinomio $p(z) = p(z_0, \ldots, z_{n-1})$, en ese caso calcular la transformada de Hadamard de $\hat{f}$ es equivalente a sumergir $p(z)$ en un módulo $(z_0^2 - 1)\ldots(z_{n-1}^2 - 1)$ y a continuación calcular los 2^n restos de $p(z)$ módulo $(z_0 \pm 1)\ldots(z_{n-1} \pm 1)$.

Por ejemplo, en $\mathbb{F}_2^3$ consideramos la función f dada por $f(x_0, x_1, x_2) = x_0x_1x_2 + x_0x_2 + x_1$. Tenemos que $p(z) = 1 + z_0 - z_1 - z_0z_1 + z_2 - z_0z_2 - z_1z_2 - z_0z_1z_2$, y los $2^3 = 8$ residuos módulo $(z_0^2 - 1)$ son $-2, 2, 6, 2, 2, -1, 2, -2$. De forma similar, el cálculo de la transformación Negahadamard de $\hat{f}$ equivale a sumergir $s(z)$ módulo $(z_0^2 + 1)\ldots(z_{n-1}^2 + 1)$ y calcular los 2^n residuos módulo $(z_0 \pm i)\ldots(z_{n-1} \pm i)$. Por ejemplo, para $s(z) = 1 + z_0 - z_1 - z_0z_1 + z_2 - z_0z_2 - z_1z_2 - z_0z_1z_2$, los 2^3 residuos módulo $(z_0^2 + 1)(z_1^2 + 1)(z_2^2 + 1)$ son $4i + 2, -2, 2, 2, -2, -2, 2, 4i - 2$.

La autocorrelación aperiódica de $\hat{f}$ puede calcularse mediante los coeficientes de $a(z)$, donde $a(z) = s(z)s^*(z^{-1})$ y $z^{-1} = (z_0^{-1}, \ldots, z_{n-1}^{-1})$.

Si el lector tiene interés en profundizar más en este tema puede consultar [ST22].

3.5.5. Funciones bent generalizadas: funciones bent sobre $\mathbb{Z}_p$

Hasta ahora hemos trabajado con el cuerpo $\mathbb{F}_2 \equiv \mathbb{Z}_2$. No obstante, podemos plantearnos cómo se trasladan ciertas definiciones y resultados al caso de funciones sobre $\mathbb{F}_p \equiv \mathbb{Z}_p$ para un primo p impar. Ahondaremos mucho más en dicha cuestión en el capítulo siguiente, aunque aprovechamos ahora para dar una ligera idea.

Sea $f : \mathbb{Z}_p^n \to \mathbb{Z}_p$ una función p-aria en m variables. La transformada de Fourier de f se viene dada por

$$F(\lambda) := \frac{1}{\sqrt{p^n}} \sum_{y \in \mathbb{Z}_p^n} f(y) \xi_p^{-\langle \lambda, y \rangle},$$

para $\lambda \in \mathbb{Z}_p^n$, y siendo ξ_p una raíz p-ésima primitiva de la unidad.

Decimos entonces que f es bent si los coeficientes de la transformada de Fourier de ξ_p^f tienen todos magnitud unitaria. Esta generalización del concepto de función bent fue introducida por Rothaus. De forma equivalente, una función es bent si y sólo si su transformada de Walsh Hadamard, dada por

$$W_f(\lambda) = \sum_{y \in \mathbb{Z}_p^n} \xi_p^{f(y) - \langle \lambda, y \rangle},$$

cumple $\|W_f(\lambda)\| = p^{\frac{n}{2}}$ para todo $\lambda \in \mathbb{Z}_p^n$ (algo similar a lo que sucedía para funciones Boolenas).

De hecho, esta noción puede definirse también para $\mathbb{Z}_n$ con $n \in \mathbb{N}$ cualquiera, no tiene por qué ser un primo necesariamente. Es habitual tomar una potencia de un primo $q = p^m$, aunque este no siempre es el caso. Redirigimos al lector a [Sch19a] y a [AEF22] si quiere conocer más acerca de este caso.

Como adelantábamos, estudiaremos mucho más a fondo esta familia de funciones en el capítulo siguiente.

3.5.6. Funciones $\mathbb{Z}$-Bent

Uno de los obstáculos en el estudio de las funciones bent es la escasez de leyes de recurrencia. Por ello, Dobbertin [DL08], [Lea05] tuvo la idea de considerar funciones de $\mathbb{F}_2^n$ en $\mathbb{Z}$ cuya transformada de Fourier normalizada también esté evaluada en $\mathbb{Z}$. Definimos la transformada de Fouirer normalizada en $a \in \mathbb{F}_2^n$ de una función f como

$$F(a) = \frac{1}{2^{\frac{n}{2}}} \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{\langle a, x \rangle}.$$

Dobbertin presentó una cadena natural de conjuntos W_r, $r \geq 0$. Denotamos $W_0 = \{\pm 1\}$, y para $r \geq 1$, $W_r = \{w \in \mathbb{Z} : -2^{r-1} \leq w \leq 2^{r-1}\}$. Tenemos que $W_r \pm W_r = W_{r+1}$, y la unión de todas las funciones W_r-bent da lugar al conjunto de las funciones $\mathbb{Z}$-bent.

Definición 3.5.7. *Una función $f : \mathbb{F}_2^r \to W_r$ se dice que es una función $\mathbb{Z}$-bent de tamaño $\frac{n}{2}$ y nivel r si su transformada de Fourier F también está evaluada en W_r. En tal caso, como la transformada de Fourier es autoinversa, F también es una función $\mathbb{Z}$-bent llamada la dual de f.*

Las funciones $\mathbb{Z}$-bent pueden clasificarse en dos niveles en función del máximo valor absoluto alcanzado por f y por F. Es posible probar que pueden obtenerse funciones $\mathbb{Z}$-bent de bajo nivel pegando algunas de nivel alto. De hecho, las funciones de nivel r en n variables pueden ser utilizadas para construir las de nivel $(r-1)$ en $(n+2)$ variables. De este modo, cualquier función $\mathbb{Z}$-bent de nivel 0 en $(n+2r)$ variables puede obtenerse procediendo recursivamente.

3.5.7. Funciones bent sobre un grupo finito

En 1997, Logachev y Yashenko [LSY97] introdujeron la noción de función bent sobre un grupo abeliano finito, que en el caso de grupos de orden 2 coincide con la noción usual de función Booleana bent.

En 2002, Solodonvnikov [Sol02] propuso la aproximación más general a la noción de función bent desde un punto de vista algebraico al considerar funciones de un grupo abeliano finito en otro.

Por otro lado, Carlet y Ding [CD04] consideraron la no linealidad perfecta de un grupo abeliano en otro: sean A, B dos grupos abelianos (con notación aditiva). La distancia de Hamming entre dos funciones $f, g : A \to B$ se define como

$$\delta(f, g) := \#\{x \in A : f(x) - g(x) \neq 0\}.$$

Una forma de medir la no linealidad de una función f de A en B es mediante la distancia de f al conjunto de funciones afines de A en B. De esta forma, podemos definir $N_f = \min_{\ell \in \mathcal{A}} \delta(f, \ell)$, donde $\mathcal{A}$ es el conjunto de todas las funciones afines de A en B. Esta medida tiene relación con el criptoanálisis lineal pero no es útil en general. Por ello se define una medida más robusta relacionada con el criptoanálisis diferencial:

$$P_f = \max_{0 \neq a \in A} \max_{b \in B} Pr[D_a f(x) = b],$$

donde $Pr[S]$ denota la probabilidad de que ocurra un cierto suceso S, y $D_a f(x) := f(x + a) - f(x)$. Cuanto menor se P_f, mayor será la no linealidad de f (si f es lineal, entonces $P_f = 1$). Observamos que las dos medidas presentadas guardan relación con la operación de cada grupo. P_f está acotada inferiormente por $\frac{1}{|B|}$, y puede considerarse una cota superior de la no linealidad de f. Aquellas funciones que tienen el menor valor posible de P_f son de gran importancia en la teoría de códigos y en la criptografía. Una función $f : A \to B$ se dice que es no lineal perfecta si $P_f = \frac{1}{|B|}$. En el caso de las

funciones Boolenas, la noción de no linealidad perfecta se resume en la palabra *bent*. Del mismo modo que teníamos una caracterización de las funciones bent en base a si su derivada está equilibrada, tenemos un resultado análogo para la no linealidad perfecta. Decimos que la derivada $D_a f$ de una función $f : A \to B$ está equilibrada si el cardinal de $f^{-1}(b)$ es el mismo para cada $b \in B$ (siendo entonces $|A|/|B|$). Esto sólo es posible si $|B|$ es divisible por $|A|$.

Teorema 3.5.8. *Una función $f : A \to B$ entre grupos finitos tiene una no lineal perfecta si y sólo si su derivada $D_a f$ está equilibrada para cada $a \in A \setminus \{0\}$.*

Para más detalles acerca de estas funciones puede consultarse [L.12] y [Pot04].

4. Funciones p-Arias

Una vez estudiadas las funciones Boolenas y en particular las funciones bent, tratamos de trasladar este estudio al caso de las funciones p-arias, con p un primo impar. Ya comentamos en la sección acerca de generalizaciones de las funciones bent que es posible extender esta noción a $\mathbb{F}_p$ definiendo lo que se conoce como transformada de Walsh Hadamard generalizada en $\mathbb{F}_p$. Sin embargo, que podamos extender la definición de función bent no significa que sus propiedades también se conserven. En el caso Booleano vimos que para n par era equivalente que una función f tuviese máxima no linealidad y que para cada $x \in \mathbb{F}_2^n$ fuese $W_f(x) = \pm 2^{\frac{n}{2}}$, y dimos esto como definición de función bent. No obstante, veremos en breve que dicha equivalencia no se mantiene en el caso p-ario. Definiremos las funciones bent mediante la transformada de Walsh Hadamard generalizada, y trataremos de calcular su no linealidad. Adelantamos que la propiedad de ser bent *no* garantiza la máxima no linealidad posible, una clara diferencia con el caso Booleano. Además, sobre $\mathbb{F}_p^n$ *sí* es posible definir la noción de ser bent para n impar, al contrario que sobre $\mathbb{F}_2^n$.

Tras ello, pasaremos a estudiar la noción de *autocorrelación*, definiendo para ello lo que se conoce como *correlación cruzada* entre dos funciones. Al aplicar esta noción a las funciones bent sobre $\mathbb{F}_p$ veremos que sí nos aporta una caracterización para esta familia de funciones.

Por último, presentaremos un algoritmo que a partir de una función p-aria arbitraria $f : \mathbb{F}_p^n \to \mathbb{F}_p$ es capaz de generar otra función $f' : \mathbb{F}_p^n \to \mathbb{F}_p$ equilibrada. Trataremos de ver en qué medida se trasladan las buenas propiedades de f a f'. Como no podía ser de otro modo, en concreto aplicaremos dicho algoritmo a funciones bent, pues (a estas alturas) conocemos bastante acerca de ellas, y en concreto sabemos que tienen buena no linealidad y una autocorrelación óptima.

4.1. Conceptos previos

Antes de entrar en materia propiamente, conviene introducir una serie de resultados variados de los que haremos uso a lo largo del capítulo. En concreto trataremos con cuerpos ciclotómicos y con ciertas propiedades de los códigos Reed Muller, ya que estos últimos guardan una muy estrecha relación con las funciones bent p-arias.

4.1.1. Cuerpos ciclotómicos

La Teoría de Cuerpos ciclotómicos aporta valiosas herramientas de las que precisaremos en este capítulo. Es por ello que optamos por dedicar esta sección a introducir algunas nociones elementales y ciertas propiedades interesantes. En concreto seguimos la línea marcada por P. V. Kumar et al en [KSW85], pioneros en el estudio de las funciones bent sobre $\mathbb{F}_p$. Aprovechamos también para definir las sumas cuadráticas de

Gauss.

Por supuesto, comenzamos definiendo lo que es un *cuerpo ciclotómico.*

Definición 4.1.1. *Sea k un natural, y denotemos por ξ_k a la raíz k-ésima de la unidad, $e^{\frac{2\pi i}{k}}$. Llamamos k-ésimo cuerpo ciclotómico al cuerpo $\mathbb{Q}(\xi_k)$ de la extensión $\mathbb{Q}(\xi_k)|\mathbb{Q}$. También puede definirse como el cuerpo de descomposición del polinomio $x^k-1 \in \mathbb{Q}[x]$.*

Nos interesa especialmente el caso en que k es un número primo p, o más generalmente, una potencia de él: $q = p^m$, $m \in \mathbb{N}$. Además, en adelante denotaremos al q-ésimo cuerpo ciclotómico por $C_q := \mathbb{Q}(\xi_q)$.

Consideramos el grupo de Galois de la extensión mencionada, $Gal(C_q|\mathbb{Q})$. Dicho grupo es abeliano, y sus elementos vienen unívocamente determinados por la potencia de ξ_q a la que envían dicha raíz q-ésima de la unidad.

Recordamos la definición de *entero algebraico.* Dada una extensión algebraica $K|\mathbb{Q}$, se dice que $a \in K$ es un entero algebraico sobre $\mathbb{Q}$ si existe un polinomio mónico $f(x) \in \mathbb{Z}[x]$ tal que $f(a) = 0$. En particular vamos a centrarnos en los enteros algebraicos de la extensión $C_q|\mathbb{Q}$.

Es conocido que un entero algebraico x sobre un C_q es una raíz de la unidad si y sólo si tiene magnitud 1. Si en lugar de C_q tenemos un cuerpo general, entonces $\sigma(x)$ debe tener magnitud 1 para cada σ del grupo de Galois de la extensión correspondiente. Como $Gal(C_q|\mathbb{Q})$ es abeliano se tiene que

$$x \cdot \sigma^*(x) = 1 \Longrightarrow \sigma(x) \cdot \sigma^*(\sigma(x)) = 1$$

para cada $\sigma \in Gal(C_q|\mathbb{Q})$, siendo σ^* el automorfismo dado por la conjugación compleja. Hemos probado lo siguiente:

Proposición 4.1.2. *Si $x \in C_q$ es un entero algebraico de magnitud 1 entonces es una raíz de la unidad.*

Denotemos por B al anillo de enteros algebraicos sobre C_q, que podemos obtener como $B = \mathbb{Z}[\xi_q]$. Consideremos el ideal $\langle p\rangle \subset B$ y su descomposición en ideales primos

$$\langle p\rangle = \langle 1 - \xi_q\rangle^{\phi(q)},$$

donde ϕ denota la función ϕ de Euler. Denotemos por G_p al subgrupo de $Gal(C_q|\mathbb{Q})$ (llamado subgrupo de descomposición del ideal $\langle 1-\xi_p\rangle$) que envía $\langle 1-\xi_q\rangle$ en sí mismo:

$$G_p = \{\sigma \in Gal(C_q|\mathbb{Q}) : \sigma(\langle 1 - \xi_q\rangle) = \langle 1 - \xi_q\rangle\}.$$

Para cada $\sigma \in Gal(C_q|\mathbb{Q})$, los enteros algebraicos $1 - \xi_q$ y $1 - \sigma(\xi_q)$ generan el mismo ideal, pues se diferencian en una unidad. Por tanto, todo el grupo $Gal(C_q|\mathbb{Q})$ deja fijo $\langle 1 - \xi_q\rangle$, luego en particular σ^* también lo hace. Hemos probado lo siguiente:

Proposición 4.1.3. *Sea $q = p^m$ con p un primo y $m \in \mathbb{N}$, entonces $\sigma^* \in G_p$.*

Pasamos ahora a estudiar de forma muy superficial las *sumas de Gauss cuadráticas*, ya que ayudan a determinar de forma sencilla si el radical $\sqrt{q}$ pertenece al cuerpo C_q o no.

La suma de Gauss cuadrática $G(q)$ viene dada por

$$G(q) = \sum_{k=0}^{q-1} \xi_q^{k^2}$$

y toma los valores

$$G(q) = \begin{cases} (1+i)\sqrt{q}, & q = 0 \mod 4. \\ \sqrt{q}, & q = 1 \mod 4. \\ 0, & q = 2 \mod 4. \\ i\sqrt{q}, & q = 3 \mod 4. \end{cases}$$

Una de las peculiaridades de la suma de Gauss cuadrática para un primo p es que se trata de un entero algebraico sobre el cuerpo ciclotómico C_p. Por tanto, a partir de esto puede establecerse el siguiente resultado.

Proposición 4.1.4. *Sea p un primo, entonces:*

1. *Si $q \equiv 0, 1 \mod 4$, entonces $\sqrt{q} \in C_q$.*
2. *Si $q \equiv 2, 3 \mod 4$, entonces $\sqrt{q} \in C_{4q} \setminus C_{2q}$.*

Para acabar esta sección, presentamos un último resultado que también involucra sumas de Gauss, y que puede encontrarse en [Nyb90].

Proposición 4.1.5. *Dado p un número primo, existe una única solución entera a la ecuación*

$$a_1\xi_p + \ldots + a_{p-1}\xi_p^{p-1} = \begin{cases} \sqrt{p}, \text{ si } p \equiv 1 \mod 4. \\ i\sqrt{p}, \text{ si } p \equiv 3 \mod 4. \end{cases}$$

Dicha solución es $a_k = \left(\frac{k}{p}\right)$, para $1 \leq k \leq p-1$, donde $\left(\frac{*}{*}\right)$ denota el símbolo de Legendre.

Recordamos que dado un primo p y un entero $k \in \mathbb{Z}$, el símbolo de Legendre de k módulo p se define como

$$\left(\frac{k}{p}\right) = \begin{cases} 0, \text{ si } p \text{ divide a } k. \\ 1, \text{ si } p \text{ no divide a } k \text{ y } k \text{ es residuo cuadrático mód } p. \\ -1, \text{ si } p \text{ no divide a } k \text{ y } k \text{ no es residuo cuadrático mód } p. \end{cases}$$

4.1.2. Teoría de códigos. Códigos Reed-Muller

En la sección siguiente trataremos la no linealidad de las funciones p-arias, y veremos que es posible expresarla en términos de ciertas magnitudes. Una de ellas guarda una estrecha relación con el radio de recubrimiento de los códigos Reed-Muller generalizados. En concreto daremos unas cotas para la no linealidad de una función p-aria cualquiera, entre otros resultados. Es por ello que dedicamos esta sección a introducirlos y presentar algunas nociones básicas.

Seguiremos la línea marcada por [Led13], pues nos serán de utilidad la Proposición 11 (que nosotros incluimos como Proposición 4.1.14) y el Lema 19 (que nosotros incluimos como Proposición 4.1.15) a la hora de probar la ya citada cota. Para su prueba haremos uso de algunos otros resultados presentes en dicho artículo, de los que omitiremos su demostración. Remitimos al lector a [Led13] si tiene interés en ella.

Comenzamos con varias definiciones elementales de la Teoría de Códigos.

Definición 4.1.6 (Código)**.** *Sea $\mathcal{A}$ un alfabeto finito. Un código es un subconjunto $\mathcal{C} \subset \mathcal{A}^n$, $n \in \mathbb{N}$. A los elementos de $\mathcal{C}$ se les denomina palabras del código.*

Definición 4.1.7 (Código Lineal)**.** *Sea p un primo, entonces $\mathbb{F}_p^n$ tiene estructura de $\mathbb{F}_p$ espacio vectorial. Un código lineal es un subconjunto $\mathcal{C} \subset \mathbb{F}_p^n$ que tiene estructura de subespacio vectorial.*

En en el caso más general también es posible considerar un cuerpo $\mathbb{F}_q$ con $q = p^m$, $m \in \mathbb{N}$.

Definición 4.1.8 (Distancia mínima)**.** *Dado un código $\mathcal{C}$ sobre un alfabeto finito $\mathcal{A}$, se define la distancia mínima de $\mathcal{C}$ como*

$$d(\mathcal{C}) = \min\{\delta(c, c') : c, c' \in \mathcal{C}\}.$$

Definición 4.1.9 (Radio de recubrimiento)**.** *El radio de recubrimiento un código $\mathcal{C}$ se define como*

$$\rho(C) := \max_{x \in \mathbb{F}_q^n} \min_{c \in \mathcal{C}} \delta(x, c).$$

En general, a la hora de calcular la distancia mínima de un código $\mathcal{C}$ hay que evaluar $\mathcal{O}((\#\mathcal{C})^2)$ valores. No obstante, es bien conocido que para el caso de los códigos lineales esto resulta mucho más sencillo.

Proposición 4.1.10. *Sea $\mathcal{C} \subset \mathbb{F}_p^n$ un código lineal, entonces*

$$d(\mathcal{C}) = \min\{\omega_H(c) : c \in \mathcal{C} \setminus \{(0, \ldots, 0)\}\}$$

Una vez introducidas estas nociones, pasamos a dar la construcción de los códigos Reed-Muller generalizados.

Sean p un número primo (en el caso clásico se toma $p = 2$), un natural $n \in \mathbb{N}$ y llamemos $q = p^n$, entonces $\mathbb{F}_q$ es un cuerpo finito de característica p con q elementos.

Dado $m \in \mathbb{N}$, denotemos por B_m^q a la $\mathbb{F}_q$-álgebra de las funciones de $\mathbb{F}_q^m$ en $\mathbb{F}_q$ y por $\mathbb{F}_q[x_1, \ldots, x_m]$ a la $\mathbb{F}_q$-álgebra de polinomios en m variables con coeficientes en $\mathbb{F}_q$.

Consideramos el homomorfismo de $\mathbb{F}_q$-álgebras $\varphi : \mathbb{F}_q[x_1, \ldots, x_m] \to B_m^q$ que a cada polinomio $P \in \mathbb{F}_q[x_1, \ldots, x_m]$ le asocia la función $f \in B_m^q$ tal que

$$f(x) = P(x_1, \ldots, x_m), \ \forall (x_1, \ldots, x_m) \in \mathbb{F}_q^m.$$

Este homomorfismo es sobreyectivo, y su núcleo es el ideal generado por los polinomios $x_1^q - x_1, \ldots, x_m^q - x_m$. Así, para cada $f \in B_m^q$ existe un único $P \in \mathbb{F}_q[x_1, \ldots, x_m]$ cuyo grado en cada variable es como mucho $q-1$ y $\varphi(P) = f$. Decimos entonces que P es la forma reducida de f, y definimos el grado de f, $\deg(f)$, como el grado de su forma reducida. El soporte de f es el conjunto $\{x \in \mathbb{F}_q^m : f(x) \neq 0\}$, y denotamos por $|f|$ al cardinal de dicho conjunto.

Observamos que podemos identificar B_m^q con $\mathbb{F}_q^{q^m}$ evaluando un elemento de B_m^q en cada elemento de $\mathbb{F}_q^m$. De este modo, $|f|$ es en realidad el peso de Hamming de f.

Dado $0 \leq r \leq m(q-1)$, se define el código de Reed-Muller generalizado de r-ésimo orden y de longitud q^m al conjunto

$$\mathcal{R}_q(r, m) := \{f \in B_m^q : \deg(f) \leq r\}.$$

Para $1 \leq r \leq m(q-1) - 2$, el grupo de automorfismos de $\mathcal{R}_q(r, m)$ es el grupo afín de $\mathbb{F}_q^m$, denotado por $GA_m(\mathbb{F}_q)$.

Necesitamos un par de definiciones más de la Teoría de Códigos.

Definición 4.1.11 (Código Autocomplementario)**.** *Un código $\mathcal{C}$ se dice que es autocomplementario si para cada $c \in \mathcal{C}$ y cada $w \in \mathbb{F}_q$ se tiene que*

$$\overline{c}^w := c - (w, \ldots, w) \in \mathcal{C}.$$

Definición 4.1.12 (Fuerza)**.** *Un código $\mathcal{C}$ se dice que tiene fuerza s si para cada s-subconjunto de coordenadas $\{i_1, \ldots, i_s\}$ el cardinal de la preimagen de cada $x \in \mathbb{F}_q^s$ por*

$$\begin{aligned} \mathcal{C} &\longrightarrow \mathbb{F}_q^s \\ c &\longmapsto (c_{i_1}, \ldots, c_{i_s}) \end{aligned}$$

no depende de x ni de $\{i_1, \ldots, i_s\}$.

El siguiente resultado puede encontrarse en [Led13] como Teorema 6.

Teorema 4.1.13. *Si $\mathcal{C}$ es un código autocomplementario sobre $\mathbb{F}_q$ de longitud n y fuerza 2, entonces*

$$\rho(\mathcal{C}) \leq \frac{(q-1)}{q} n - \frac{\sqrt{n}}{q}.$$

Por último, mostramos dos cotas del radio de recubrimiento $\rho(m)$ de un Código de Reed Muller $\mathcal{R}(1,m)$.

Proposición 4.1.14. *Para cada primo p y cada entero $m \geq 0$, si $q = p^n$, se satisface*

$$\rho(m) \leq (q-1)q^{m-1} - q^{\frac{m}{2}-1}.$$

Demostración.

Como $\mathcal{R}_q(0,m) \subset \mathcal{R}_q(1,m)$, tenemos que $\mathcal{R}_q(1,m)$ es autocomplementario, luego por el Teorema 4.1.13, basta probar que $\mathcal{R}_q(1,m)$ tiene fuerza 2.

Sean $y, z \in \mathbb{F}_q^m$ distintos, y sea $(a,b) \in \mathbb{F}_q^2$. Consideremos la aplicación

$$\begin{aligned} u : \mathcal{R}_q(1,m) &\longrightarrow \mathbb{F}_q^2 \\ f &\longmapsto (f(y), f(z)) \end{aligned}$$

Si identificamos f con los coeficientes de su forma reducida, entonces la matriz de esta aplicación es la $2 \times (m+1)$ matriz $\begin{bmatrix} y & 1 \\ z & 1 \end{bmatrix}$.

Como $y \neq z$, su rango es 2, y el núcleo tiene dimensión $m-1$. Esto implica que u es sobreyectiva, y para cada $(a,b) \in \mathbb{F}_q^2$ se tiene que $\#u^{-1}(a,b) = q^{m-1}$. Esto prueba que el código en cuestión tiene fuerza 2, y por tanto por la Proposición 4.1.13 tenemos que $\rho(\mathcal{C}) \leq \frac{(q-1)}{q}q^m - \frac{\sqrt{q^m}}{q} = (q-1)q^{m-1} - q^{\frac{m}{2}-1}$, ya que $\mathcal{R}(1,m)$ tiene longitud q^m.

□

Proposición 4.1.15. *Para cada primo p y cada entero $m \geq 0$, si $q = p^n$, se satisface*

$$(q-1)^2 q^m + q\rho(m) \leq \rho(m+2).$$

Demostración.

Sea $f \in \mathcal{R}_q(1,m+2)$, en tal caso podemos expresar f como $f(x_1,\ldots,x_{m+2}) = g(x_1,\ldots,x_m) + \alpha x_{m+1} + \beta x_{m+2}$, con $g \in \mathcal{R}_q(1,m)$ y $\alpha, \beta \in \mathbb{F}_q$. Denotemos los elementos de $\mathbb{F}_q$ por $\omega_0,\ldots,\omega_{q-1}$, fijando $\omega_0 = 0$. De este modo,

$$\mathcal{R}_q(1,m+2) = \{M(c,\alpha,\beta) : c \in \mathcal{R}_q(1,m), \alpha, \beta \in \mathbb{F}_q\},$$

donde c es la identificación de $g(x_1,\ldots,x_m) \in B_m^q$ visto como elemento de $\mathbb{F}_q^{q^m}$, es decir, $c = (c_1,\ldots,c_{q^m}) = (g(e_1),\ldots,g(e_{q^m}))_{e_i \in \mathbb{F}_q^m}$, y

$$M(c,\alpha,\beta) = \big(c_i + \alpha\omega_j + \beta\omega_k\big)_{1 \leq i \leq q^m,\ 0 \leq j,k \leq q-1}.$$

De forma más clara, $M(c,\alpha,\beta)$ es de la forma

$$\big(g(e_1)+\alpha\omega_0+\beta\omega_0, g(e_1)+\alpha\omega_0+\beta\omega_1, \ldots, g(e_{q^m})+\alpha\omega_{q^m-1}+\beta\omega_{q^m}, g(e_{q^m})+\alpha\omega_{q^m}+\beta\omega_{q^m}\big).$$

Si v_0 es tal que $d(v_0, \mathcal{R}_q(m)) = \rho(m)$, entonces para cada $c \in \mathcal{R}_q(1,m)$ se tiene $|v_0 + c| \geq \rho(m)$.

Sea ahora

$$u = \left(\bar{v_0}^{\omega_0\omega_0}, \ldots, \bar{v_0}^{\omega_0\omega_{q-1}}, \bar{v_0}^{\omega_1\omega_0}, \ldots, \bar{v_0}^{\omega_1\omega_{q-1}}, \ldots, \bar{v_0}^{\omega_{q-1}\omega_{q-1}}\right) \in B^q_{m+2}.$$

Dados $\alpha, \beta \in \mathbb{F}_q$ y $c \in \mathcal{R}_q(1,m)$ tenemos que

$$\begin{aligned}
|u + M(c,\alpha,\beta)| &= \sum_{i=0}^{q-1}\sum_{j=0}^{q-1} |\bar{v_0}^{\omega_i\omega_j} + \bar{c}^{\alpha\omega_i+\beta\omega_j}| \\
&= \sum_{i=0}^{q-1}\sum_{j=0}^{q-1} |\bar{v_0}^{(\beta+\omega_i)\omega_j} + \bar{c}^{\alpha\omega_i}| \\
&= q|(\bar{c}^{\alpha(-\beta)}) + v_0| + \sum_{\omega_i \neq -\beta}\sum_{j=0}^{q-1} |\bar{v_0}^{(\beta+\omega_i)\omega_j} + \bar{c}^{\alpha\omega_i}| \\
&\geq q\rho(1,m) + (q-1)^2 q^m,
\end{aligned}$$

lo que concluye la prueba.

□

4.2. No linealidad

Del mismo modo que estudiamos la no linealidad de las funciones Booleanas, tratamos de trasladar este estudio al caso de característica $p \neq 2$. Veremos que la situación es algo más compleja y habremos de usar ciertas herramientas basadas en la Teoría de Cuerpos Ciclotómicos y en la Teoría de Códigos (en particular los de Reed-Muller), ya introducidas en las secciones anteriores. Seguiremos principalmente [Sch19c].

Sea p un primo impar, denotamos por $\mathcal{V}_n$ al conjunto de funciones p-arias $f : \mathbb{F}_p^n \to \mathbb{F}_p$, y por $\mathcal{A}_n$ al conjunto de funciones p-arias afines, es decir, las funciones de la forma $g(x) = a_0x_0 + \ldots + a_{n-1}x_{n-1} + a_n$, con $a_i \in \mathbb{F}_p$ para $0 \leq i \leq n$.

De forma similar a como se define la distancia de Hamming entre dos funciones Boolenas, podemos definir la distancia de Hamming entre dos funciones p-arias $f, g : \mathbb{F}_p^n \to \mathbb{F}_p$ como

$$\delta(f,g) = \#\{x \in \mathbb{F}_p^n : f(x) \neq g(x)\}.$$

Esto nos permite extender el concepto de no linealidad de una función Boolena al caso p-ario para una cierta f como

$$\mathcal{N}_f = \min_{g \in \mathcal{A}_n} \delta(f,g).$$

Antes de continuar, destacamos que en adelante usaremos la notación $q = p^m$ donde p es un primo y m un natural. Trataremos con funciones más generales $f : \mathbb{F}_q^n \to \mathbb{F}_q$, ya que los resultados presentados en la sección anterior acerca de los códigos Reed Muller nos lo permiten.

Buscamos aquellas funciones que presenten la máxima no linealidad. Por ser sintéticos damos una notación específica a este valor máximo:

$$\rho_q(n) = \max_{f \in \mathcal{V}_n} \min_{g \in \mathcal{A}_n} \delta(f, g).$$

De este modo, sería interesante tratar de encontrar funciones $f \in \mathcal{V}_n$ tales que $N_f = \rho_q(n)$. El número $\rho_2(n)$ es igual al radio de recubrimiento del código Reed-Muller binario de orden uno $\mathcal{R}_2(1, n)$ [HKM13], y de forma parecida, $\rho_q(n)$ se corresponde con el radio de recubrimiento de un código Reed-Muller generalizado sobre $\mathbb{F}_p$ de primer orden $\mathcal{R}_q(1, n)$ [Led13]. Gran parte del desarrollo que haremos a continuación se cimienta en este hecho.

Definimos a continuación

$$\mu_q(n) = \frac{q^{n-1}(q-1) - \rho_q(n)}{q^{\frac{n}{2}-1}},$$

de donde

$$\rho_q(n) = q^{n-1}(q-1) - \mu_q(n) q^{\frac{n}{2}}.$$

Si queremos la función que de la máxima no linealidad, buscamos maximizar $\rho_q(n)$. Por la expresión anterior, esto equivale a minimizar $\mu_q(n)$. Del mismo modo que hemos definido μ_q y ρ_q para todo el espacio $\mathbb{F}_q^n$, podemos ligar estos valores a funciones concretas. Así, dada $f : \mathbb{F}_q^n \to \mathbb{F}_q$ tenemos

$$\rho_q(f) = \min_{g \in \mathcal{A}_n} \delta(f, g) = q^{n-1}(q-1) - \mu_q(f) q^{\frac{n}{2}}, \tag{4.1}$$

siendo

$$\mu_q(f) = \frac{q^{n-1}(q-1) - \rho_q(f)}{q^{\frac{n}{2}-1}}. \tag{4.2}$$

Es inmediato que $\rho_q(f) = \mathcal{N}_f$, de modo que es posible obtener la no linealidad de f a partir de $\mu_q(f)$. Convenientemente, existe una forma de calcular este valor sin recurrir a $\rho_q(f)$ (que es lo que queremos conocer en última instancia), haciendo uso de la *transformada de Fourier*. Antes de mostrar dicho calculo conviene establecer la siguiente notación:

Dada una extensión de Galois $K|F$, se define la función traza $Tr_{K|F} : K \to F$ mediante

$$Tr_{K|F}(x) = \sum_{\sigma \in Gal(K|F)} \sigma(x).$$

En lo que a nosotros nos atañe, la extensión en cuestión será $\mathbb{F}_{q^n}|\mathbb{F}_q$. Denotamos por η, ψ a los caracteres aditivos canónicos de $\mathbb{F}_{q^n}$ y $\mathbb{F}_q$, respectivamente:

$$\eta(x) = \exp\left(\frac{2\pi i \cdot Tr_{\mathbb{F}_q^n|\mathbb{F}_p}(x)}{p}\right), \quad x \in \mathbb{F}_q.$$
$$\psi(x) = \eta(Tr_{\mathbb{F}_{q^n}|F_q}(x)), \quad x \in \mathbb{F}_{q^n}.$$

Lema 4.2.1. *Para todo $x \in \mathbb{F}_q^n$, $n \in \mathbb{N}$, se verifica*

$$\sum_{y\in\mathbb{F}_q^n} \xi_q^{\langle y,x\rangle} = \begin{cases} q^n, & si\ x = 0. \\ 0, & si\ x \neq 0. \end{cases}$$

Demostración.

Tenemos que

$$\sum_{y\in\mathbb{F}_q^n} \xi_q^{\langle y,x\rangle} = \sum_{y\in\mathbb{F}_q^n} \xi_q^{x_0y_0+\ldots+x_{n-1}y_{n-1}} = \prod_{i=1}^{n} \sum_{x_i\in\mathbb{F}_q^n} \xi_q^{x_iy_i}$$
$$= \prod_{i=1}^{n} \left(\frac{1-(\xi_q^{x_i})^q}{1-\xi_q^{x_i}}\right) = \begin{cases} q^n, & \text{si } x_i = 0\ \forall i. \\ 0, & \text{en caso contrario.} \end{cases}$$

□

Definición 4.2.2. *Dada una función $g : \mathbb{F}_{q^n} \to \mathbb{F}_q$, se define la transformada de Fourier de g como la función $\hat{g} : \mathbb{F}_{q^n} \times \mathbb{F}_q \to \mathbb{C}$ dada para cada $\lambda \in \mathbb{F}_q$, $a \in \mathbb{F}_{q^n}$ por*

$$\hat{g}(a,\lambda) = \frac{1}{p^{\frac{n}{2}}} \sum_{y\in\mathbb{F}_{q^n}} \eta(\lambda g(y))\overline{\psi(ay)}.$$

Ahora sí, tenemos la siguiente relación entre $\mu_q(f)$ y su transformada de Fourier. Esto permite obtener computacionalmente $\mu_q(f)$ con el fin de compararlo con $\mu_q(n)$.

Proposición 4.2.3. *Para cada $f : \mathbb{F}_q^n \to \mathbb{F}_q$ es posible calcular $\mu_q(n)$ como*

$$\mu_q(f) = \max_{a\in\mathbb{F}_{q^n}, b\in\mathbb{F}_q} \sum_{\lambda\in\mathbb{F}_q^*} \overline{\eta(\lambda b)}\hat{g}(\lambda a, \lambda). \tag{4.3}$$

Demostración.

Para cada $z \in \mathbb{F}_q$ se tiene que

$$\frac{1}{q}\sum_{\lambda\in\mathbb{F}_q} \eta(\lambda z) = \begin{cases} 1, & \text{si } z = 0. \\ 0, & \text{si } z \neq 0. \end{cases}$$

Por tanto, para toda función $h : \mathbb{F}_{q^n} \to \mathbb{F}_q$ tenemos

$$\begin{aligned}\delta(g,h) &= q^n - \frac{1}{q}\sum_{y\in\mathbb{F}_{q^n}}\sum_{\lambda\in\mathbb{F}_q}\eta(\lambda(g(y)-h(y))) \\ &= q^{n-1}(q-1) - \frac{1}{q}\sum_{y\in\mathbb{F}_{q^n}}\sum_{\lambda\in\mathbb{F}_q^*}\eta(\lambda(g(y)-h(y))).\end{aligned}$$

Observamos a continuación que las funciones afínes de $\mathbb{F}_{q^n}$ en $\mathbb{F}_q$ son precisamente las q^{n+1} funciones $h_{a,b}$ dadas por

$$h_{a,b}(x) := Tr_{\mathbb{F}_{q^n}|\mathbb{F}_q}(ax) + b$$

para cada $a \in \mathbb{F}_q^n$, $b \in \mathbb{F}_q$. Por tanto,

$$\begin{aligned}\delta(g,h_{a,b}) &= q^{n-1}(q-1) - \frac{1}{q}\sum_{\lambda\in\mathbb{F}_q^*}\overline{\eta(\lambda b)}\sum_{y\in\mathbb{F}_q^n}\eta(\lambda(g(y))\overline{\psi()\lambda a y} \\ &= q^{n-1}(q-1) - q^{\frac{n}{2}-1}\sum_{\lambda\in\mathbb{F}_q^*}\sum_{\lambda\in\mathbb{F}_q^*}\overline{\eta(\lambda(g(y))}\hat{g}(\lambda a,\lambda).\end{aligned}$$

Tomando mínimo en $\delta(g,h_{a,b})$ para $h_{a,b}$ y por la forma de $\mathcal{N}_g = \rho_q(g)$ se deduce la igualdad del enunciado.

□

Si $q = p$, entonces la fórmula de la proposición anterior se simplifica considerablemente, quedando como

$$\mu_q(f) = \max_{a\in\mathbb{F}_{q^n}, b\in\mathbb{F}_q}\sum_{\lambda\in\mathbb{F}_q^*}\xi_p^{-\lambda b}\tilde{f}(\lambda a,\lambda), \tag{4.4}$$

donde

$$\tilde{f}(\lambda a,\lambda) = \frac{1}{p^{\frac{n}{2}}}\sum_{y\in\mathbb{F}_{p^n}}\xi_p^{\lambda f(y)-Tr(\lambda a y)}.$$

Recordamos que $Tr(\lambda a y) = \sum_{i=0}^{n-1}(\lambda a y)^{p^i}$.

Esta fórmula nos será de gran utilidad en la última sección de este capítulo a la hora de calcular la no linealidad de funciones concretas.

Calcular el valor exacto de $\mu_q(n)$ es un problema abierto, aunque sí se conocen algunos valores para ciertos q concretos. Es sencillo ver que $\mu_q(n) = 1$ cuando n es par. Del Corolario 13 de [Led13] (que presentamos a continuación) puede deducirse que $\mu_2(n) = 1$ para n par.

Proposición 4.2.4. *Dado un primo p y un natural n par, si $q = p^m$, se verifica que*

$$\rho_q(m) = (q-1)q^{n-1} - q^{\frac{n}{2}-1}.$$

Demostración.

La Proposición 12 de [Led13] nos dice que

$$\rho_q(n) \geq (q-1)q^{n-1} - q^{\lceil \frac{n}{2} \rceil} - 1,$$

y la Proposición 4.1.14 que

$$\rho_q(n) \leq (q-1)q^{n-1} - q^{\frac{n}{2}-1},$$

luego para n par se tiene que

$$(q-1)q^{n-1} - q^{\frac{n}{2}} - 1 \leq \rho_q(n) \leq (q-1)q^{n-1} - q^{\frac{n}{2}-1},$$

y por tanto

$$\rho_q(n) = (q-1)q^{n-1} - q^{\frac{n}{2}} - 1.$$

□

De este modo, $\mu_2(n) = \dfrac{2^{n-1}(2-1) - (2-1)2^{n-1} - 2^{\frac{n}{2}-1}}{2^{\frac{n}{2}-1}} = 1$. También se conoce una relación entre los valores de $\mu_q(n)$ y $\mu_q(m)$ cuando n y m tienen la misma paridad.

Proposición 4.2.5. *Dado un primo p y un natural $n \in \mathbb{N}$, si $q = p^m$, se verifica que*

$$1 \leq \mu_q(n+2k) \leq \mu_q(n) \text{ para cada } k \in \mathbb{N}.$$

Demostración.

De la Proposición 4.1.14 se sigue que

$$\begin{aligned}
\rho_q(n) &\leq (q-1)q^{n-1} - q^{\frac{n}{2}-1} \\
\rho_q(n) &\leq q^{n-1}(p-1) - q^{\frac{n}{2}-1} \\
0 &\leq q^{n-1}(q-1) - q^{\frac{n}{2}-1} - \rho_q(n) \\
1 &\leq \frac{q^{n-1}(q-1) - \rho_q(n)}{q^{\frac{n}{2}-1}} = \mu_q(n).
\end{aligned}$$

Y de la Proposición 4.1.15,

$$\begin{aligned}
(p-1)^2 q^n + q\rho_q(n) &\leq \rho_q(n+2) \\
(q-1)^2 q^n - \rho_q(n+2) &\leq -q\rho_q(n) \\
q^n(q-1) + (q-1)^2 q^n - \rho_q(n+2) &\leq q^n(q-1) - \rho_q(n) \\
q^{n+1}(q-1) - \rho_q(n+2) &\leq q^n(q-1) - q\rho_q(n) \\
\mu_q(n+2) = \frac{q^{n+1}(q-1) - \rho_q(n+2)}{q^{\frac{n+2}{2}-1}} &\leq \frac{q^{n+1}(q-1) - \rho_q(n)}{q^{\frac{n}{2}-1}} = \mu_q(n).
\end{aligned}$$

Por tanto, de forma inductiva se tiene que $\mu_q(n+2k) \leq \mu_q(n)$ para cada $k \in \mathbb{N}$.

□

De aquí se deduce de forma inmediata que si n es par, entonces $\mu_q(n) = 1$, ya que $\mu_q(2) = 1$. En estas circunstancias, $\rho_q(n) = q^{n-1}(q-1) - q^{\frac{n}{2}-1}$. Observamos además que si $p = 2$, entonces $\rho_2(n) = 2^{n-1} - 2^{\frac{n}{2}-1}$, justo la cota que mostramos en la Definición 3.3.5 para las funciones bent Booleanas. Más aún, de 4.3 se deduce que $\mu_2(n) = \max_{x \in \mathbb{F}_2^n} |W_f(x)|$.

En el caso de n impar la situación resulta algo más compleja. Para $q = 2$ se sabe que $\mu_2(3) = \mu_2(5) = \mu_2(7) = \sqrt{2}$ y que $\mu_2(9) < \sqrt{2}$ [Sch19b]. A partir de $n > 9$ el valor de $\mu_q(n)$ es desconocido, aunque sí se sabe que $\mu_q(1) = \sqrt{q}$ [Sch19b], y por tanto

$$1 \leq \mu_q(n) \leq \sqrt{q}.$$

Patterson y Wiedman conjeturaron en 1983 que $\lim_{n\to\infty} \mu_q(n) = 1$. Fue probado en 2019 por Kai-Uwe Schmidt en [Sch19b].

Hemos establecido varias generalidades acerca de las funciones p-arias y ciertas expresiones para la no linealidad en función del primo p en cuestión y el número de variables n. A continuación procedemos a definir las funciones bent para característica impar. En primer lugar, dada una función $f : \mathbb{F}_q^n \to \mathbb{F}_q$ se define la transformada de Walsh Hadamard normalizada de f en un cierto $x \in \mathbb{F}_q$ mediante

$$W_f(x) = \frac{1}{q^{\frac{n}{2}}} \sum_{y \in \mathbb{F}_q^n} \xi_q^{f(y) - \langle x, y \rangle}.$$

Ahora sí:

Definición 4.2.6. *Sea p un primo impar y $q = p^m$, y sea $f : \mathbb{F}_q^n \to \mathbb{F}_q$ una función q-aria. Decimos que f es bent si $|W_f(x)| = 1$ para todo $x \in \mathbb{F}_q$.*

Ejemplo 4.2.7. *Dado un primo p y un $n \in \mathbb{N}$ par, la función $f_{p,n} : \mathbb{F}_p^n \to \mathbb{F}_p$, $\quad f_{p,n}(x) = x_1x_2 + \ldots + x_{n-1}x_n$ es bent. Notemos que esta función no es más que la generalización de la dada en el Ejemplo 3.3.10.*

Vemos que la definición dada no hace sino extender al caso general la ya dada en 3.3.5, y si tomamos $p = 2$ se obtiene esta última. Existen muchas propiedades que las funciones bent preservan al pasar del caso Booleano al caso de característica impar, aunque veremos que no todas se conservan. Una de las que sí, es la relación que mantienen las funciones bent con las matrices de Hadamard, en el caso q-ario esta relación se da con lo que se conoce como *matriz de Buston Hadamard*:

Definición 4.2.8. *Una matriz cuadrada H de orden k se dice que es una matriz de Buston Hadamard si todas sus entradas son raíces k-ésimas de la unidad y además $HH^* = nI$, donde H^* es la conjugada traspuesta de H.*

Se tiene la siguiente equivalencia, la cual puede encontrarse en [KSW85] como Propiedad 4.

Teorema 4.2.9. *Una función $f : \mathbb{F}_q^n \to \mathbb{F}_q$ es bent si y sólo si la matriz $H = (h_{x,y})_{x,y}$, donde $h_{x,y} = \xi_q^{f(x-y)}$, es una matriz de Buston Hadamard.*

Para el caso Booleano sabemos que el espectro de Walsh Hadamard normalizado de una función bent es $\{\pm 1\}$, que no son más que las raíces 2-ésimas de la unidad. Cabría esperar que para q general obtuviésemos las raíces q-ésimas de la unidad. ¿Será cierta esta intuición? Uno de los propósitos fundamentales de esta sección ser ver que, en efecto, *esto es así*, salvo por un factor ± 1 o $\pm i$ que dependerá del primo p en cuestión y de la paridad de n.

Los primeros en estudiar esta serie de cuestiones fueron P.V. Kumar, R.A. Scholtz y L.R. Welch en [KSW85]. Varios de los resultados que mostramos a continuación proceden o se basan en dicho artículo.

En primer lugar probamos que los valores que toma la transformada de Walsh Hadamard normalizada de una función bent p-aria son raíces de la unidad [KSW85].

Proposición 4.2.10. *Sea p un primo y f una función bent q-aria con $q = p^m$ para un primo p, $m \in \mathbb{N}$. Entonces para cada $x \in \mathbb{F}_q^n$, el valor de $W_f(x)$ es una raíz de la unidad.*

Demostración.

Sea $x \in \mathbb{F}_q^n$, tenemos que $W_f(x) \cdot \sigma^*\big(W_f(x)\big) = 1$.

De la Proposición 4.1.3 se sigue que σ^* pertenece al grupo de descomposición G_p. Al descomponer $\langle q \rangle$ en el producto de sus ideales primos puede verse que $W_f(x)$ y $\sigma^*(W_f(x))$ generan el mismo ideal en $B = \mathbb{Z}[\xi_q]$, y por tanto

$$\langle W_f(x)^2 \rangle = \langle 1 \rangle.$$

Esto significa que $W_f(x)^2$ es una unidad, y por tanto el coeficiente $W_f(x)$ es un entero algebraico. Como f es bent, dicho entero tiene magnitud 1, y por la Proposición 4.1.2, es una raíz de la unidad.

□

Observación 4.2.11. *Observamos sin embargo que esta proposición* ***no*** *indica si el valor de $W_f(x)$ se corresponde en concreto con una raíz q-ésima de la unidad, cosa que, de hecho, no sucede en general.*

A partir de la Proposición 4.1.4 y de la propia definición de W_f se sigue que

$$\begin{aligned} n \text{ par o } n \text{ impar y } q \equiv 0, 1 \mod 4 &\Longrightarrow W_f(x) \in C_q. \\ n \text{ impar y } q \equiv 2, 3 \mod 4 &\Longrightarrow W_f(x) \in C_{4q} \setminus C_{2q}. \end{aligned}$$

A partir de esto y dado que las raíces de la unidad en C_q son de la forma ξ_{2q}^k para algún $k \in \mathbb{N}$, en [KSW85] deduce la siguiente proposición. En dicho artículo podemos encontrarla como Propiedad 8.

Proposición 4.2.12. *Sean p un primo y $q = p^m$, y sea $f : \mathbb{F}_q^n \to \mathbb{F}_q$ una función bent q-aria. Dado $x \in \mathbb{F}_q^n$, si $W_f(x)$ es una raíz de la unidad, entonces es de la forma*

$$W_f(x) = \begin{cases} \xi_q^{\frac{k}{2}}, & si\ q\ es\ impar\ y\ n \equiv 0 \mod 2. \\ \xi_q^{k}, & si\ n\ es\ impar\ y\ q \equiv 0 \mod 4. \\ \xi_q^{\frac{k}{2}}, & si\ n\ es\ impar\ y\ q \equiv 1 \mod 4. \\ \xi_q^{\frac{2k+1}{4}}, & si\ n\ es\ impar\ y\ q \equiv 2 \mod 4. \\ \xi_q^{\frac{2k+1}{4}}, & si\ n\ es\ impar\ y\ q \equiv 3 \mod 4. \end{cases}$$

Para algún entero k que dependerá de x: $k(x)$.

A partir de esta proposición puede probarse el siguiente resultado, el cual ya adelantábamos.

Teorema 4.2.13. *Sea $f : \mathbb{F}_q^n \to \mathbb{F}_q$ una función bent. Para cada $x \in \mathbb{F}_q^n$ se verifica que*

$$W_f(x) = \begin{cases} \pm\xi_q^{f^*(x)}, \ si\ q \equiv 1 \mod 4. \\ \pm i\xi_q^{f^*(x)}, \ si\ q \equiv 3 \mod 4. \end{cases}$$

Siendo f^* una función de $\mathbb{F}_q^n$ en $\mathbb{F}_q$ llamada, al igual que en el caso binario, dual de f.

Demostración.

Si $q \equiv 1 \mod 4$, tomando $f^*(x) = \frac{k}{2}$ tenemos que $W_f(x) = (1 \cdot \xi_q)^{\frac{k}{2}} = 1^{\frac{k}{2}} \cdot \xi_q^{\frac{k}{2}} = (\pm 1) \cdot \xi_q^{f^*(x)}$.

Si $q \equiv 3 \mod 4$, tomando $f^*(x) = \frac{2k+1}{4}$ tenemos que $W_f(x) = (1 \cdot \xi_q)^{\frac{2k+1}{4}} = 1^{\frac{2k+1}{4}} \cdot \xi_q^{\frac{2k+1}{4}} = (\pm i) \cdot \xi_q^{f^*(x)}$.

□

No deja de ser curioso que esa suma, aparentemente arbitraria, dé como resultado precisamente un raíz q-ésima de la unidad multiplicada por un cierto $\varepsilon \in \{\pm 1, \pm i\}$ (que no son más que las raíces cuartas de la unidad).

Una función bent f se dice que es *débilmente regular* si para cada $x \in \mathbb{F}_q^n$ cumple $W_f(x) = \varepsilon \cdot \xi_q^{f^*(x)}$ para algún $\varepsilon \in \{\pm 1, \pm i\}$ fijo que no depende de x. Si en concreto $\varepsilon = 1$, entonces se dice que f es *regular*. A partir del teorema anterior es inmediato que no exsiten funciones bent regulares cuando $q \equiv 3 \mod 4$. Las funciones bent débilmente regulares pertenecen a la clase de las funciones bent cuya función dual también es bent. Más aún, como $f^{**}(x) = f(-x)$, se tiene que $f^{****}(x) = f(x)$.

En el caso Booleano, las funciones bent se definen como aquellas que presentan máxima no linealidad. También se comprueba que que $|W_f(x)| = 2^{\frac{n}{2}}$ para todo x, luego las preimágenes $|f^{-1}(0)|, |f^{-1}(1)|$ tienen tamaño $2^{n-1} - 2^{\frac{n}{2}-1}$. Esto, en Teoría de Códigos, se traduce en que la distancia de la función f al conjunto de las funciones afines es el radio de recubrimiento de un código de Reed-Muller de primer orden $\mathcal{R}_2(1, n)$, como ya comentamos anteriormente.

Esta caracterización no se traslada de forma inmediata al caso de p impar, con $\mathcal{R}_p(1,n)$. La distribución de los valores, sin llegar a ser homogénea, sí sigue una cierta regularidad en caso de que f sea bent.

Kaysa Niberg describe esta situación en [Nyb90], en los Teoremas 3.2 a 3.5. Nosotros mostramos el Teorema 1 de [Mei22], que sintetiza estos resultados de forma más compacta.

Teorema 4.2.14. *Sea p un primo impar y sea $f : \mathbb{F}_p^n \to \mathbb{F}_p$ una función p-aria, y dado $\lambda \in \mathbb{F}_p$ sea $b_\lambda = |f^{-1}(\lambda)|$. Entonces:*

1. *Si n es par, entonces existe un único $c \in \mathbb{F}_p$ tal que $b_c = p^{n-1} \pm (p-1)p^{\frac{n}{2}-1}$ y $b_\lambda = p^{n-1} \mp p^{\frac{n}{2}-1}$ para todo $\lambda \in \mathbb{F}_p$. Más aún, si f es regular, se toman los signos superiores. La distancia de Hamming a la función afín más cercana es $(p-1)p^{n-1} - p^{\frac{n}{2}-1}$, o $(p-1)(p^{n-1} - p^{\frac{n}{2}-1})$ cuando f es regular.*
2. *Si n es impar, entonces la distribución de los valores viene dada por $(b_0, \ldots, b_{p-1})$ o alguna permutación cíclica suya, siendo $b_0 = p^{n-1}$ y $b_\lambda = p^{n-1} + (\frac{\lambda}{p})p^{\frac{n-1}{2}}$ para cada $\lambda \in \mathbb{F}_p$, o bien $b_\lambda = p^{n-1} - (\frac{\lambda}{p})p^{\frac{n-1}{2}}$ para cada $\lambda \in \mathbb{F}_p$, donde $(\frac{*}{*})$ denota el símbolo de Legendre. La distancia de Hamming a la función afín más cercana es $(p-1)p^{n-1} - p^{\frac{n-1}{2}}$.*

Demostración.

1. De acuerdo con el Teorema 4.2.12, existe un entero s tal que $W_f(0) = \xi_p^{\frac{s}{2}}$, luego
$$\sum_{k=0}^{p-1} b_k \xi_p^k = p^{\frac{n}{2}} \xi_p^{\frac{s}{2}}.$$
Si s es par, entonces la ecuación anterior resulta
$$\sum_{k=0}^{p-1} b_k \xi_p^{k-r} = p^{\frac{n}{2}}$$
para algún $r \in \mathbb{Z}$. Esto siempre sucede cuando f es regular. Si s es impar, entonces tomamos $r = \frac{s-p}{2}$, lo que da lugar a
$$\xi_p^{\frac{s}{2}-r} = \xi_p^{\frac{p}{2}} = -1.$$
Así, la ecuación se convierte en
$$\sum_{k=0}^{p-1} b_k \xi_p^{k-r} = -p^{\frac{n}{2}}$$
para algún $r \in \mathbb{Z}$. Como p es primo, se sigue que
$$b_0 \mp p^{\frac{n}{2}} = b_1 = \ldots = b_{p-1},$$
o alguna de sus permutaciones cíclicas. Por otro lado, como $\sum_{k=0}^{p-1} b_k = p^n$, se tiene que
$$b_0 \mp p^{\frac{n}{2}} = b_1 = \ldots = b_{p-1} = p^{n-1} \mp p^{\frac{n}{2}-1}.$$

Veamos ahora el valor de la distancia de Hamming de f a la función afín más cercana. Para ello sea $A : \mathbb{F}_p^n \to \mathbb{F}_p$ una función afín dada por $f(x) = \langle w, x\rangle + r$, y denotemos

$$a_k = \{x \in \mathbb{F}_p^n : f(x) - \langle w, x\rangle - r = k\}.$$

Ahora, por la Proposición 4.1.5, para $w \neq 0$ procedemos de forma análoga y obtenemos

$$a_0 \mp p^{\frac{n}{2}} = a_1 = \ldots = a_{p-1} = p^{n-1} \mp p^{\frac{n}{2}-1}.$$

La distancia de Hamming de f a la función afín A es entonces $\sum_{k=1}^{p-1} a_k$. El valor mínimo se alcanza cuando r es tal que el valor de a_0 es máximo, pudiendo ser $p^{n-1} + (p-1)p^{\frac{n}{2}-1}$ o $p^{n-1} + p^{\frac{n}{2}-1}$.

2. Consideramos primero el caso $p \equiv 1 \mod 4$. De nuevo, por el Teorema 4.2.12, existe un entero s tal que

$$W_f(0) = \xi_p^{\frac{s}{2}} = \frac{1}{\sqrt{p^n}} \sum_{k=0}^{p-1} b_k u^k.$$

Procediendo de forma similar al caso anterior la ecuación anterior se transforma en

$$b_0 + b_1\xi_p + \ldots + b_{p-1}\xi_p^{p-1} = \pm\sqrt{p} \cdot p^{\frac{n-1}{2}}.$$

Por la Proposición 4.1.5, la solución de dicha ecuación es de la forma

$$b_k = b_0 \pm \left(\frac{k}{p}\right) p^{\frac{n-1}{2}},\ 1 \leq k \leq p-1.$$

De $\sum_{k=0}^{p-1} = p^n$ se sigue que $b_0 = p^{n-1}$.

Supongamos ahora que $p \equiv 3 \mod 4$. De nuevo por el Teorema 4.2.12, existe $s \in \mathbb{Z}$ tal que $W_f(0) = \xi_p^{\frac{2s+1}{4}}$. Si tomamos $r = \frac{p^2-1}{4}(2s+1) \in \mathbb{Z}$, tenemos que

$$W_f(0) = \xi_p^{p(2s+1)\cdot\frac{p}{4}-r} = \pm i \cdot \xi_p^{-r},$$

ya que $p(2s+1)$ es impar. A continuación procedemos de forma análoga al primer caso y obtenemos el resultado.

Para calcular la distancia de Hamming al conjunto de funciones afines repetimos la prueba tomando $w \neq 0$.

□

El teorema anterior nos informa de otra (mala) propiedad que mantienen las funciones bent al pasar del caso Booleano al caso general: nunca están equilibradas. Esto sigue siendo unos de los principales puntos débiles de las funciones bent y uno

de los motivos por los que se desaconseja su uso directo. Por otro lado, este resultado también nos indica el valor de la no linealidad de una función bent, de modo que a partir de la expresión 4.1 podemos obtener el valor de $\mu_q(f)$ correspondiente a cada caso:

$$N_f = \begin{cases} (q-1)q^{n-1} - q^{\frac{n-1}{2}}, & \text{si } n \text{ es impar, luego } \mu_q(f) = \sqrt{q}. \\ (q-1)q^{n-1} - q^{\frac{n}{2}-1}, & \text{si } n \text{ es par y } f \text{ no es regular, luego } \mu_q(f) = 1. \\ (q-1)q^{n-1} - (q-1)\cdot q^{\frac{n}{2}-1}, & \text{si } n \text{ es par y } f \text{ es regular, luego } \mu_q(f) = q-1. \end{cases}$$

Vemos que para n impar y f bent es $1 \leq \mu_q(n) \leq \sqrt{q} = \mu_q(f)$, por lo que nada impide encontrar (y de hecho es posible construirlas) una función $g : \mathbb{F}_q^n \to \mathbb{F}_q$ tal que $1 \leq \mu_q(g) < \sqrt{q}$. Dicha g, que obviamente no puede ser bent, daría un valor de la no linealidad mayor que cualquier función bent.

Para n par vemos que si f es regular, entonces $1 = \mu_q(n) < q - 1 = \mu_q(f)$, por lo que debe existir alguna función $g : \mathbb{F}_q^n \to \mathbb{F}_q$ que no sea bent tal que $\mu_q(g) = \mu_q(n) = 1$. Si f es bent pero no es regular entonces sí se alcanza el valor óptimo de μ_q.

En otras palabras, la propiedad de ser bent *no* caracteriza la máxima no linealidad en $\mathbb{F}_q$, con q una potencia de un primo impar, al contrario de lo que sucedía en $\mathbb{F}_2$. Esta es una diferencia importante entre el caso Booleano y el caso q-ario.

Esta es una de las principales características de las funciones bent sobre $\mathbb{F}_p$. En [KSW85], de Kumar et al., pueden encontrarse algunas otras propiedades. Mostramos varias de ellas para terminar esta sección.

1. Toda traslación lineal o afín de una función bent es también bent.
2. Toda función bent sigue siendo bent bajo la acción de cualquier transformación lineal o afín en sus coordenadas.
3. Si f y g son funciones bent sobre $\mathbb{F}_p^m$ y $\mathbb{F}_p^n$, respectivamente, entonces la función $f + g$ definida en $\mathbb{Z}_p^{m+n}$ dada por $(f+g)(x_1, \ldots, x_{n+m}) = f(x_1, \ldots, x_m) + g(x_{m+1}, \ldots, x_{m+n})$ también es bent.

4.3. Autocorrelación

Estudiamos en esta sección lo que se conoce como *autocorrelación*. Esta es una medida que de alguna forma refleja las coincidencias que tiene la secuencia imagen de una función consigo misma. Es decir, si evaluamos una función $f : \mathbb{F}_p^n \to \mathbb{F}_p$ en todo $\mathbb{F}_p$ se obtiene una secuencia $\ell \in \mathbb{F}_p^{p^n}$; la autocorrelación mide si existen fragmentos de ℓ que, aun estando separados, sean muy parecidos o incluso idénticos. En criptografía siempre se busca la mayor aleatoriedad posible, por lo que es fundamental que apenas existan dichas coincidencias. Del mismo modo que una alta no linealidad ofrece

garantías frente al criptoanálisis lineal, una baja autocorrelación proporciona fortaleza frente al criptoanálisis diferencial. De hecho, es a partir de los principios de confusión y difusión de la Teoría de la información de Shannon [Sha49] de donde se sigue que la autocorrelación de las funciones empleadas en un criptosistema debe ser baja.

Comenzamos presentando el concepto de *correlación cruzada*.

Definición 4.3.1. *Dadas dos funciones $f, g : \mathbb{F}_q^n \to \mathbb{F}_q$ se define la correlación cruzada entre f y g en un cierto $x \in \mathbb{F}_q^n$ como la suma*

$$C_{f,g}(x) = \sum_{y \in \mathbb{F}_q^n} \xi_q^{f(y+x)-g(x)}.$$

Si $g = f$, entonces se habla de autocorrelación, y se denota por $C_f(x)$.

Además de esta noción, Zhou et al. han presentado un par de indicadores más: el indicador *suma de cuadrados*, y el *indicador absoluto* [SL11], que mostramos a continuación.

Definición 4.3.2. *Dadas dos funciones $f, g : \mathbb{F}_q^n \to \mathbb{F}_q$ se define el indicador suma de cuadrados (SSMI) de f y g como*

$$\sigma_{f,g} = \sum_{y \in \mathbb{F}_q^n} |C_{f,g}(u)|^2$$

Definición 4.3.3. *Dadas dos funciones $f, g : \mathbb{F}_q^n \to \mathbb{F}_q$ se define el indicador absoluto de f y como*

$$\Delta(f) = \frac{1}{q^{\frac{n}{2}}} \max_{0 \neq x \in \mathbb{F}_q^n} |C_f(x)|.$$

Al igual que en la sección anterior dimos un nombre al máximo valor de la no linealidad alcanzable por una función fijados q y n, actuamos de forma similar. Denotamos por $\Delta_q(n)$ al mínimo valor posible de $\Delta(f)$, es decir,

$$\Delta_q(n) = \min_{f \in \mathcal{V}_n} \Delta_q(f).$$

Si $\Delta_q(n) = \Delta_q(f)$, entonces f se dice que f es óptima frente al criptoanálisis diferencial.

Un resultado interesante y muy práctico acerca de la autocorrelación es que es posible calcularla a partir de la transformada de Walsh Hadamard, lo que mostramos a continuación. Este y los sucesivos pueden encontrarse en [BK13].

Teorema 4.3.4. *Sean $f, g : \mathbb{F}_q^n \to \mathbb{F}_q$, para cada $u, y \in \mathbb{F}_q^n$ se tiene que*

1. $\displaystyle\sum_{x \in \mathbb{F}_q^n} C_{f,g}(u) \xi_q^{\langle u,y \rangle} = q^n W_f(y) \overline{W_g(y)}.$
2. $\displaystyle C_{f,g}(u) = \sum_{y \in \mathbb{F}_q^n} W_f(y) \overline{W_g(y)} \xi_q^{\langle -u,y \rangle}.$

Demostración.

La correlación cruzada entre f y g se calcula como

$$C_{f,g}(u) = \sum_{x \in \mathbb{F}_q^n} \xi_q^{f(x)-f(x+u)},$$

luego por el Lema 4.2.1,

$$\begin{aligned}
\sum_{u \in \mathbb{F}_q^n} C_{f,g}(u)\xi_q^{\langle u,y\rangle} &= \sum_{u \in \mathbb{F}_q^n} \sum_{x \in \mathbb{F}_q^n} \xi_q^{f(x)-g(x+u)+\langle u,y\rangle} \\
&= \sum_{x \in \mathbb{F}_q^n} \xi_q^{f(x)} \sum_{u \in \mathbb{F}_q^n} \xi_q^{-g(x+u)+\langle u,y\rangle} \\
&= \sum_{x \in \mathbb{F}_q^n} \xi_q^{f(x)} \sum_{u \in \mathbb{F}_q^n} \xi_q^{-g(u)+\langle x-u,y\rangle} \\
&= \sum_{x \in \mathbb{F}_q^n} \xi_q^{f(x)+\langle x,y\rangle} \sum_{u \in \mathbb{F}_q^n} \xi^{-(g(u)+\langle u,y\rangle)} \\
&= q^n W_f(y)\overline{W_g(y)}.
\end{aligned}$$

De aquí se sigue que

$$\begin{aligned}
\sum_{y \in \mathbb{F}_q^n} W_f(y)\overline{W_g(y)}\xi_q^{\langle -u,x\rangle} &= \frac{1}{q^n} \sum_{y \in \mathbb{F}_q^n} \sum_{v \in \mathbb{F}_q^n} C_{f,g}(v)\xi_q^{\langle v,y\rangle+\langle -u,y\rangle} \\
&= \frac{1}{q^n} \sum_{v \in \mathbb{F}_q^n} C_{f,g}(v) \sum_{y \in \mathbb{F}_q^n} \xi_q^{\langle v-u,y\rangle} \\
&= C_{f,g}(u).
\end{aligned}$$

□

A partir de este teorema, tomando $f = g$, se sigue de forma inmediata el siguiente corolario.

Corolario 4.3.5. *Sea $f : \mathbb{F}_q^n \to \mathbb{F}_q$ una función q-aria. Para cada $x \in \mathbb{F}_q^n$, la autocorrelación de f en x puede calcularse mediante*

$$C_f(x) = \sum_{y \in \mathbb{F}_q^n} |W_f(y)|^2 \xi_q^{-\langle x,y\rangle}.$$

Una prueba alternativa de este resultado puede encontrarse en [AEF22] (Lema 2.3).

Esto nos permite calcular la autocorrleación de una función de forma computacional. De este modo, podemos obtener de forma sencilla no sólo la no linealidad de una función, sino también su autocorrelación.

Por otro lado, vemos que si evaluamos en $x = 0$ obtenemos

$$\sum_{y \in \mathbb{F}_q^n} |W_f(y)|^2 = q^n,$$

es decir, se prueba la Identidad de Parseval (ver Teorema 3.2.5 para el caso Booleano) para $\mathbb{F}_q$.

Más aún, también resulta inmediato el siguiente resultado. Puede probarse a partir del Teorema 4.2.9, aunque se deduce de forma directa del corolario anterior y del Lema 4.2.1.

Corolario 4.3.6. *Una función $f : \mathbb{F}_q^n \to \mathbb{F}_q$ es bent si y sólo si $C_f(u) = 0$ para todo $u \in \mathbb{F}_q^n \setminus \{0\}$.*

Es decir, al contrario que la no linealidad, la autocorrelación *sí* caracteriza a las funciones bent en el caso q-ario. Como además gracias al Corolario 4.3.5 es posible calcular la autocorrelación a partir de la transformada de Walsh Hadamard, es posible determinar computacionalmente si una función f es bent de forma sencilla.

Esto nos dice además que si existe alguna función bent f sobre $\mathbb{F}_q^n$, entonces $\Delta_q(n) = 0$, ya que de hecho $\Delta_q(n) = 0 = \Delta_q(f)$. Por tanto, siempre podemos alcanzar el valor de autocorrelación óptimo. Se sabe además que para todo primo impar p siempre existe una función bent $f : \mathbb{F}_p^n \to \mathbb{F}_p$ tal que $\delta_p(f) = 0$ [KSW85].

Esto supone una diferencia con el caso Booleano, ya que sólo existen funciones bent sobre $\mathbb{F}_2^n$ cuando n es par. Por tanto, sólo podríamos asegurar $\Delta_2(n) = 0$ para $n = 2k$, $k \in \mathbb{N}$. Si n es impar, entonces se sabe que $\Delta_2(n) \leq \sqrt{2}$.

4.4. Balanceamiento

Como ya hemos comentado anteriormente, las funciones bent nunca están balanceadas, motivo por el cual su uso directo está totalmente desaconsejado. No obstante, siguen presentando unas muy buenas cualidades, su máxima no linealidad las hace resistentes al criptoanálisis lineal, y su autocorreleación nula da buenas garantías frente al criptoanálisis diferencial. Es por eso surge la siguiente pregunta: ¿es posible transformar de alguna forma una función bent para equilibrarla pero manteniendo en cierta medida sus otras cualidades? En [Sch20], Schmidt aborda esta cuestión, la de construir funciones Booleanas balanceadas con baja autocorrelación. Inspirándonos en dicho artículo tratamos de definir funciones Booleanas equilibradas con baja autocorrelación también y con alta no linealidad. Para ellos planteamos el siguiente algoritmo:

Dada una función bent Booleana $f : \mathbb{F}_2^n \to \mathbb{F}_2$ se siguen los siguientes pasos.

1. Se calcula el valor $e \in \{0, 1\}$ que hay en exceso.
2. Se recorre el espacio $\mathbb{F}_2^n$, evaluando cada $x \in \mathbb{F}_2^n$. Si $f(x) \neq e$, entonces se define $f'(x) = f(x)$, y si $f(x) = e$, entonces se define $f'(x) = 1 - e$ con probabilidad $\frac{2^{\frac{n}{2}-1}}{2^{n-1}+2^{\frac{n}{2}-1}}$.

3. Volvemos al paso 1 hasta que f' esté equilibrada.

Hemos implementado dicho algoritmo en el software *Mathematica*, si se desea consultar el código puede abrirse el archivo adjunto (y de hecho se anima a hacerlo), aunque lo comentamos brevemente más adelante.

Optamos por aplicarlo a la función del Ejemplo 3.3.10 y las usadas en el cifrador Haval (3.4.1), es decir, las siguientes:

$$f_n : \mathbb{F}_2^n \to \mathbb{F}_2, \quad f_n(x) = x_1x_2 \oplus x_3x_4 \oplus \ldots \oplus x_{n-1}x_n.$$

$$g_2 : \mathbb{F}_2^6 \to \mathbb{F}_2, \quad g_2(x) = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_2 \oplus x_1, x_4 \oplus x_2x_6 \oplus x_3x_5 \oplus x_4x_5.$$

$$g_3 : \mathbb{F}_2^6 \to \mathbb{F}_2, \quad g_3(x) = x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6.$$

$$g_4 : \mathbb{F}_2^6 \to \mathbb{F}_2, \quad g_4(x) = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_3x_4x_6 \oplus x_1x_4 \oplus x_2x_6 \oplus x_3x_4 \oplus x_3x_5 \oplus x_3x_6 \oplus x_4x_5 \oplus x_4x_6.$$

Mostramos en la siguiente tabla los resultados obtenidos, es decir, para cada función f comparamos los parámetros $\mu(f)$ (recordamos que mide la no linealidad) y $\Delta(f)$ (que mide la autocorrelación) con los de la función f': $\mu(f')$ y $\Delta(f')$, generada por el algoritmo.

p	n	f	$\mu(f)$	$\mu(f')$	$\Delta(f)$	$\Delta(f')$
2	6	f_6	1	2	0	2
2	8	f_8	1	1,75	0	1,5
2	10	f_{10}	1	1,75	0	1,5
2	6	g_2	1	1,875	0	2
2	6	g_3	1	2	0	2
2	6	g_4	1	2	0	2

Al tratarse de un algoritmo probabilístico, los valores obtenidos pueden variar levemente al aplicarlo varias veces a la misma función. No obstante, para los parámetros p, n y las funciones dadas parecen mantenerse bastante estables. Los valores de $\mu(f')$ y $\Delta(f')$ permanecen prácticamente idénticos aun generando varias f' distintas.

Nada nos impide a priori extender este algoritmo al caso general sobre $\mathbb{F}_p$ con p un primo impar. En este caso el algoritmo es como sigue:

1. Se calcula el valor $e \in \{0, \ldots, p-1\}$ que hay en exceso.
2. Se recorre el espacio $\mathbb{F}_p^n$, evaluando cada $x \in \mathbb{F}_p^n$. Si $f(x) \neq e$, entonces se define $f'(x) = f(x)$, y si $f(x) = e$, entonces se define $f'(x) = e'$ con $e' \in \{0, \ldots, p-1\}$ elegido aleatoria y uniformemente con probabilidad $\frac{(p-1)p^{\frac{n}{2}-1}}{p^{n-1}+(p-1)p^{\frac{n}{2}-1}}$.
3. Volvemos al paso 1 hasta que f' esté equilibrada.

Una vez presentado el algoritmo para el caso p-ario pasamos a comentar brevemente el código implementado. En primer lugar hay ciertas definiciones básicas tales como los cuerpos finitos $\mathbb{F}_p$ y $\mathbb{F}_{p^n}$ y el espacio $\mathbb{F}_p^n$, así como los isomorfismos correspondientes entre $\mathbb{F}_p^n$ y $\mathbb{F}_{p^n}$. Esto será fundamental a la hora de calcular el valor μ_f de un función $f : \mathbb{F}_p^n \to \mathbb{F}$, ya que para ello emplearemos la fórmula 4.4, que por supuesto también está implementada y aparece como "μ". Tres de las funciones que más se utilizan son "$H\omega f$", que determina el peso de Hamming de f, "*balanced*", que determina si una función está equilibrada y "WH", que calcula la transformada de Walsh Hadamard de f en un cierto $x \in \mathbb{F}_p^n$. Entre las funciones básicas se encuentra por último "$Traza$", que es exactamente lo que su nombre indica. Acabamos de hablar de la función "μ" para el cálculo de $\mu(f)$, para la autocorrelación definimos "Cc", que calcula el coeficiente de autocorrelación de f en un cierto x, y "Δ", que se corresponde con el indicador absoluto de f. Cabe destacar que hemos creado versiones de "μ", "Cc" y "Δ" exclusivas para el caso binario, ya que en él se simplifican ciertos cálculos. En concreto estos nombre hacen referencia a las versiones Booleanas, las versiones p-arias aparecen como "μp", "Ccp" y "Δp". Mencionar también que a la hora de definir las funciones a las que se les aplica el algoritmo lo hacemos mediante su forma algebraica normal, pero sus correspondientes "hermanas" generadas por el algoritmo vienen dadas por la lista de sus imágenes sobre el correspondiente $\mathbb{F}_p^n$.

Al implementar este algoritmo en Mathematica surge un pequeño inconveniente, si los valores de p y n son mínimamente elevados, entonces los cálculos resultan ser realmente lentos. Para el cálculo de $\mu(f)$ y $\Delta(f)$ es necesario recorrer los espacios $\mathbb{F}_5^4$ y $\mathbb{F}_{5^4}$, ambos de $5^4 = 625$ elementos. De hecho, para obtener $\mu(f)$ aplicamos la Fórmula 4.4, en la que se toman (en nuestro caso) $a \in \mathbb{F}_{5^4}$ y $b \in \mathbb{F}_5$, lo que da lugar a $5^4 \cdot 5 = 625 \cdot 5 = 3{,}125$ parejas (a, b). Además, a continuación se realiza una suma en la que un cierto λ va variando en $\mathbb{F}_5^*$, y a su vez se realiza otra suma en la que un y se mueve en todo $\mathbb{F}_{5^4}$. Esto son otras $(5-1) \cdot 5^4 = 4 \cdot 625 = 2{,}500$ parejas de elementos (λ, y). En total tenemos, *como mínimo*, $3{,}125 \cdot 2{,}500 = 7{,}812{,}500$ operaciones. Además, alcanzada esta magnitud, comienza a perderse precisión en las operaciones.

Por este motivo no parece razonable realizar pruebas en funciones con parámetros p y n demasiado grandes. Muchas de las funciones bent p-arias que pueden encontrarse en la literatura superan el umbral que estamos dispuestos a admitir, lo que reduce considerablemente la gama de funciones que podemos tomar. Para nuestras pruebas optamos por tomar la función bent del Ejemplo 4.2.7

$$f_{p,n} : \mathbb{F}_p^n \to \mathbb{F}_p, \quad f_{p,n}(x) = x_1x_2 + \ldots + x_{n-1}x_n,$$

para ciertos valores p y n (que debe ser par) concretos.

En particular tomamos $p = 5$ y $n = 2$. Mostramos en la siguiente tabla los valores obtenidos al aplicar el algoritmo varias veces.

p	n	f	$\mu(f)$	$\mu(f')$	$\Delta(f)$	$\Delta(f')$
5	2	$f_{5,2}$	4	3	0	1,7591
5	2	$f_{5,2}$	4	3	0	1,53975
5	2	$f_{5,2}$	4	3	0	1,31286
5	2	$f_{5,2}$	4	5	0	1,84879
5	2	$f_{5,2}$	4	3	0	1,403

En este caso sí se observa una variabilidad mayor, por lo que decidimos hacer un análisis estadístico. En una muestra de 100 funciones obtenemos un valor medio de $\overline{\mu(f')} = 3{,}18$, y un valor medio de $\overline{\Delta(f')} = 1{,}44552$.

Observamos dos hechos: el primero es que la autocorrelación empeora. Esto era de esperar, pues la propiedad de ser bent caracteriza la auotocorrelación óptima, y la función f' obtenida *no* puede ser bent al ser equilibrada. Sin embargo, el otro dato a destacar es que $\mu(f')$ ha disminuido, lo que da lugar a una no linealidad mayor. Es decir, a pesar de empeorar ligeramente (y de forma controlada) la autocorrelación, aplicar el algoritmo a esta función bent da lugar a una función *equilibrada y con una mayor no linealidad*. Al aplicar el algoritmo a una función Booleana se obtenía otra que sacrificaba no linealidad a cambio de estar equilibrada, sin embargo, aquí parece se mejoran ambos parámetros, el único que empeora es la autocorrelación, lo cual era inevitable, pero podría decirse que la función que obtenemos es esencialmente *mejor*.

Y esto no parece restringirse sólo a $f_{5,2}$. Aplicando el algoritmo a $f_{3,4}$ varias veces se tienen los siguientes valores.

p	n	f	$\mu(f)$	$\mu(f')$	$\Delta(f)$	$\Delta(f')$
3	4	$f_{3,4}$	2	3	0	1,20185
3	4	$f_{3,4}$	2	1	0	1,52753
3	4	$f_{3,4}$	2	2	0	1,45297
3	4	$f_{3,4}$	2	1,33333	0	1,20185
3	4	$f_{3,4}$	2	2,33333	0	1,45297

De nuevo realizamos un análisis estadístico. En una muestra de 100 funciones obtenemos un valor medio de $\overline{\mu(f')} = 1{,}93333$, y un valor medio de $\overline{\Delta(f')} = 1{,}27286$.

El valor de $\mu(f')$ *vuelve a ser menor* que el de $\mu(f)$. Nada de esto debería extrañarnos, pues ya vimos que existen funciones p-arias con mayor no linealidad que cualquier función bent. De hecho, vimos que para funciones bent p-arias no regulares se cumplía $\mu(f) = p-1$, lo que encaja perfectamente con los datos obtenidos. Para $f_{5,2}$ los cálculos dan $\mu(f_{5,2}) = 4 = 5-1$, y para $f_{3,4}$ los cálculos resultan $\mu(f_{3,4}) = 2 = 3-1$.

En conclusión, vemos que, pese a las limitaciones computacionales (que quizá podrían solventarse implementándolo en otro lenguaje como Python o C++), el algoritmo mostrado ofrece muy buenos resultados (al menos para la muestra empleada). En el caso binario no empeora demasiado la no linealidad y la autocorrelación, y en el caso p-ario llega incluso a aumentar la no linealidad (disminuyendo el valor de μ) manteniendo controlada la autocorrelación. Esto evidencia una clara utilidad práctica y sería interesante analizar más a fondo las funciones obtenidas para ver si exhiben otras buenas propiedades como las mencionadas en la Sección 3.1.4, como podría ser el estudio del grado algebraico al construir la forma normal algebraica de las funciones obtenidas, lo que podría suponer una futura ampliación de esta sección.

5. Conclusiones

Comenzamos este capítulo de conclusiones realizando una breve recapitulación de todo los visto hasta ahora. En primer lugar y dadas las características de este proyecto, dimos unas nociones básicas de criptografía, ya que el enfoque que damos a las funciones bent es puramente criptográfico. Definimos los cifradores a flujo y a bloque, en los que es común su uso (aunque no de forma directa), así como varios tipos de ataques. En concreto nos centramos en el criptoanálisis lineal y en el criptoanálisis diferencial, debido a las garantías que ofrecen las funciones bent frente a estos.

En el tercer capítulo comenzamos a hablar de funciones Booleanas y dimos propiamente la definición de función bent: "aquellas con máxima no linealidad". La caracterizamos mediante la transformada de Walsh Hadamard, y vimos que existen una gran variedad de representaciones equivalentes de las funciones bent. En particular describimos algunas de ellas, como aquella que emplea matrices de Hadamard y la basada en diseños. Seguidamente vimos varias de las aplicaciones que tienen estas funciones en criptosistemas concretos, tales como Grain y HAVAL. No obstante, comprobamos que en dichos esquemas no se empleaban directamente funciones bent, ya que algo que sacrifican a cambio de tener máxima no linealidad es que nunca están equilibradas, resultado que también probamos. Por último presentamos varias de las generalizaciones clásicas de las funciones bent, una de las cuales daría pie a desarrollar el siguiente capítulo.

En él, se ahonda en la noción de función bent p-aria, haciendo uso de la transformada de Walsh Hadamard generalizada para dar su definición. En concreto nos centramos en estudiar dos propiedades, la no linealidad, y la autocorrelación, para lo que precisamos de algunas nociones de la Teoría de Códigos y de la Teoría de Cuerpos Ciclotómicos. Un resultado llamativo al que llegamos fue que las funciones bent p-arias no siempre presentan máxima no linealidad, al contrario de lo que sucedía en el caso Booleano. No obstante, comprobamos que una autocorrelación óptima sí caracteriza a las funciones bent sobre $\mathbb{F}_p$.

Con estas dos últimas propiedades sobre la mesa, nos planteamos la posibilidad de generar función p-arias (pudiendo ser $p = 2$) que exhibiesen buenas cualidades. Para ello empleamos en algoritmo mostrado en la Sección 4.4, el cual es capaz de generar funciones balanceadas a partir de cualquier función. Naturalmente optamos por tomar como entrada para el algoritmo funciones bent, pues conocemos sus buenas propiedades y sabemos que una de las que le falta es que estén equilibradas. Fue una sorpresa comprobar que en el caso Booleano la no linealidad y la autocorrelación sacrificadas a cambio de obtener el balanceamiento son realmente bajas. Más sorprendente aun fue ver que en las funciones p-arias la no linealidad incluso aumentaba. Por pasmoso que esto resultara, no debíamos olvidar que sobre $\mathbb{F}_p$ existen funciones con una no linealidad mayor que cualquier función bent, por lo que lo realmente interesante es que hayamos podido construir dichas funciones, no su existencia en sí. Esto daría punto y final a este proyecto.

Cabe destacar, no obstante, que esto no supone sino una *introducción*. Las funciones bent son tema muy rico y muy vivo, con muchas líneas de investigación abiertas, como puede ser el estudio de las secuencias bent autoduales generadas a partir de matrices de Buston Hadamard [AEC22]. Si bien nuestro enfoque ha sido fundamentalmente teórico reiteramos de nuevo que las funciones bent tienen una sólida aplicación en la Criptografía que no se queda sólo en el papel y su estudio puede conllevar grandes avances.

6. Bibliografía

[Ada97] C. Adams. Constructing symmetric ciphers using the CAST. *Designs, Codes and Cryptography; 12(3): pp. 283-316*, 1997.

[AEC22] J.A. Armario, R. Egan, and P.O. Catháin. On the matrix equation MX = $\bar{X}$ and self-dual butson bent sequences. *Aceptado en The 8th International Workshop on Boolean Functions and their Applications (BFA23). September 3-8, 2023. Voss, Norway.*, 2022.

[AEF22] J.A. Armario, R. Egan, and D.L. Flannery. Generalized partially bent functions, generalized perfect arrays and cocyclic butson matrices. *Aceptado en Cryptography and Communications*, 2022.

[AT] C. Adams and S. Tavares. Rfc 2144-the CAST-128 encryption algorithm. http://www.faqs.org/rfcs/rfc2144.html.

[BA93] E. Biham and Shamir A. Differential cryptanalysis of the data encryption standard. *New York: Springer Verlag*, 1993.

[BK13] M. Bhaintwal and B. Kumar. Some results on q-ary bent functions. *International Journal of Computer Mathematics*, 2013.

[BS91] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol. 4(1), pp 3–72*, 1991.

[Car93] C. Carlet. Partially-bent functions. *Designs Codes and Cryptography, 3, pp 135–145*, 1993.

[Car10] C. Carlet. Boolean functions for cryptography and error correcting codes. *Chapter of the monography "Boolean Models and Methods in Mathematics, Computer Science, and Engineering" published by Cambridge University Press, Yves Crama and Peter L. Hammer (eds.), pp 257–397*, 2010.

[Car13] C. Carlet. Open problems on binary bent functions. *Proceedings of the conference "Open problems in mathematical and computational sciences", September pp 18–20*, 2013.

[CAT14] W. Cesmelioglu A., Meidl and A. Topuzoglu. Partially bent functions and their properties. *Applied Algebra and Number Theory, pp 22*, 2014.

[CCF01] A. Canteaut, C. Carlet, and C. Fontaine. On cryptographic properties of the cosets of R(1,m). *IEEE Transactions on Information Theory, vol. 47, pp 1494–1513*, 2001.

[CD04] C. Carlet and C. Ding. Highly nonlinear mappings. *In Special Issue: "Complexity Issues in Coding and Cryptography" of the Journal of Complexity, 20(2–3), pp 205–244*, 2004.

[CLK94] S. Chee, S. Lee, and K. Kim. Semi-bent functions. *Advances in Cryptology-ASIACRYPT94. Proc. 4th Int. Conf. on the Theory and Applications of Cryptology, Wollongong, Australia. Pieprzyk, J. and Safavi-Naini, R., Eds., Lect. Notes Comp. Sci, vol 917, pp 107–118*, 1994.

[CS09] T. W. Cusick and P. Stănică. *Cryptographic Boolean Functions and Applications*. Elsevier, 2009.

[Dil72] JF. Dillon. A survey of bent functions. *NSA Technical Journal, pp 191–215.*, 1972.

[Dil74] JF. Dillon. Elementary Hadamard difference sets. PhD dissertation. *University of Maryland, College Park*, 1974.

[DL08] H. Dobbertin and G. Leander. Bent functions embedded into the recursive framework of Z-bent functions. *Designs, Codes and Cryptography volume 49, pp 3–22*, 2008.

[FF98] E. Filiol and C. Fontaine. Highly nonlinear balanced boolean functions with a good correlation-immunity. *Proceedings of EUROCRYPT'98, Lecture Notes in Comput. Sci., vol. 1403, pp. 475–488*, 1998.

[FF99] E. Filiol and C. Fontaine. On some cosets of the first Reed-Muller code with high minimum weight. *IEEE Trans. Inform. Theory 45 (4) pp 1237–1243*, 1999.

[GZL12] G. Gao, X. Zhang, and W. Liu. Constructions of quadratic and cubic rotation symmetric bent functions. *IEEE Trans. Inform. Theory, No 58, pp 4908–4913*, 2012.

[HJM06] M. Hell, T. Johansson, and W. Meier. A stream cipher proposal: Grain-128. *eSTREAM ECRYPT Stream Cipher Project*, 2006.

[HKM13] T. Helleseth, T. Kløve, and J. Mykkeltveit. On the covering radius of binary codes. *IEEE Transactions on Information Theory*, 2013.

[KMY07] S. Kavut, S. Maitra, and M.D. Yucel. Search for boolean functions with excellent profiles in the rotation symmetric class. *IEEE Trans. Inform. Theory 53 (5), pp 1743–1751*, 2007.

[KSW85] P. V. Kumar, R. A. Scholtz, and L. R. Welch. Generalized bent functions and their properties. *Journal of Combinatorial Theory, Series A 40, pp 90-107*, 1985.

[L.12] Poinsot L. Non abelian bent functions. *Cryptography and Communications volume 4, pp 1–23*, 2012.

[Lea05] H. Leander, G.and Dobbertin. Cryptographer's toolkit for construction of 8-bit bent functions. *Cryptology ePrint Archive, Report 2005/089*, 2005.

[Led13] E. Leducq. On the covering radius of first-order generalized Reed–Muller codes. *IEEE Transactions on Information Theory*, 2013.

[LSY97] O. A. Logachev, A. A. Sal'nikov, and V. V. Yashenko. Bent functions on a finite abelian group. *Diskr. Mat., Volume 9, Issue 4, pp 3–20*, 1997.

[Mas69] J. L. Massey. Shift-register synthesis and bch decoding. *Information Theory, IEEE Transactions on, 15(1), pp 122–127*, 1969.

[Mat93] M. Matsui. Linear cryptanalysis method for DES cipher. *Advances in cryptology EUROCRYPT'93*, 1993.

[Mat94] M. Matsui. Linear cryptanalysis method for DES cipher. *Proceedings of EUROCRYPT'93, Lecture Notes in Computer Science 765, pp 386–397*, 1994.

[Mei22] W. Meidl. A survey on p-ary and generalized bent functions. *Cryptography and Communications Volume 14 Issue 4 pp 737–782*, 2022.

[Men60] K. Menon. Difference sets in abelian groups. *Proc AMS, 11, pp368–77*, 1960.

[Mes16] S. Mesnager. *Bent Functions, Fundamentals and Results*. Springer, 2016.

[MOS96] A. Menezes, P. van Oorschot, and Vanstone S. *Handbook of Applied Cryptography*. CRC Press, 1996.

[MS88] W. Meier and O. Staffelbach. Fast correlation attacks on stream ciphers. *Proceedings of EUROCRYPT' 88. Lecture Notes in Computer Science, vol. 330, pp 301–314*, 1988.

[MY92] M. Matsui and A. Yamagishi. A new method for known plaintext attack of feal cipher. *Advances in cryptology EUROCRYPT'92*, 1992.

[Nyb90] K. Nyberg. Constructions of bent functions and difference sets. *Advances in cryptology–EUROCRYPT '90*, 1990.

[PGV91] B. Preneel, R. Govaerts, and J. Vandevalle. Boolean functions satisfying higher order propagation criteria. *Proceedings of EUROCRYPT'91, Lecture Notes in Computer Sciences 547, pp 141–152*, 1991.

[Pot04] A. Pott. Nonlinear functions in abelian groups and relative difference sets. *Discrete Applied Mathematics, vol. 138, issue 1–2, pp 177–193*, 2004.

[PQ99] J. Pieprzyk and C. Qu. Fast hashing and rotation symmetric functions. *J. Univers. Comput. Sci. No. 5, pp 20–31*, 1999.

[RH07] S. Rønjom and T. Helleseth. A new attack on the filter generator. *IEEE Transactions on Information Theory, vol. 53, no. 5, pp 1752–1758*, 2007.

[Rot76] O.S. Rothaus. On "bent" functions. *Journal of Combinatorial Theory, Series A, Volume 20, Issue 3, May 1976, pp 300-305*, 1976.

[Sch19a] B. Schmidt. A survey of group invariant Butson matrices and their relation to generalized bent functions and various other objects. *Radon Ser. Comput. Appl. Math. 23, pp 241-251*, 2019.

[Sch19b] KU. Schmidt. Asymptotically optimal boolean functions. *Journal of Combinatorial Theory, Series A, volume 164, pp 50-59*, 2019.

[Sch19c] K.U. Schmidt. Highly nonlinear functions over finite fields. *Finite Fields and Their Applications, volume 63, pp 101640*, 2019.

[Sch20] K.U. Schmidt. Recent results on the nonlinearity of boolean functions. The 5th International Workshop on Boolean Functions and their Applications (BFA), September 15 – September 17, 2020.

[Sha49] C.E. Shannon. Communication theory of secrecy systems. *Bell system technical journal 28, pp 656–715*, 1949.

[SL11] M. Shi and Y. Li. On cross-correlation properties of boolean functions,. *International Journal of Computer Mathematics 88(10), pp 2035-2041*, 2011.

[SL23] M. Shi and Y. Li. Self-dual Hadamard bent sequences. *J. Syst. Sci. Complex. 36(2), pp 894-908*, 2023.

[Sol02] V. I. Solodovnikov. Bent functions from a finite abelian group into a finite abelian group. *Diskr. Mat., Volume 14, Issue 1, pp 99–113*, 2002.

[ST22] D. Sharma and S. Tiwari. On generalized bent and negabent functions. *Mathematics and Statistics 10(3), pp 542-548*, 2022.

[Tok15] N. Tokareva. *Bent Functions: Results and Applications to Cryptography.* Elsevier, 2015.

[WZ07] X. Wang and J. Zhou. Generalized partially bent functions. *Proceedings Future Generation Communication and Networking (FGCN 2007), Vol. 1, pp 16–21*, 2007.

[ZZ99] Y. Zheng and X.M. Zhang. Plateaued functions. *Advances in Cryptology-ICICS 1999 (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, vol.1726, pp 284–300*, 1999.