Curso 1994/95
CIENCIAS Y TECNOLOGÍAS

PINO CABALLERO GIL

Avances en el estudio de la complejidad lineal del filtrado no lineal

Directora
AMPARO FÚSTER SABATER

SERVICIO DE PUBLICACIONES
UNIVERSIDAD DE LA LAGUNA

SOPORTES AUDIOVISUALES E INFORMÁTICOS
Serie Tesis Doctorales

¿Qué es más grande-el mar o la palabra con que lo nombramos?

Emilio Adolfo Westphalen

El presente trabajo ha sido realizado bajo la dirección de la Doctora Dña Amparo Fúster Sabater, a quien deseo expresar mi más profundo agradecimiento por su inestimable ayuda y permanente dedicación.

También quiero hacer constar mi gratitud a los miembros del Laboratorio de Criptografía del Consejo Superior de Investigaciones Científicas de Madrid, a los compañeros del Departamento de Estadística, Investigación Operativa y Computación de la Universidad de La Laguna y a todas aquellas personas que de una u otra forma me han ayudado, especialmente a Carlos Bruno Castañeda, a Antonio Sedeño Noda, y a mi tutor el doctor D. Joaquín Sicilia Rodríguez.

La Laguna, a uno de Junio de 1995.

Pino Caballero Gil

Índice general

Prólogo

La importancia de la Información y por tanto de su Protección es un hecho ampliamente reconocido y manifestado. Actualmente, en lo que se ha dado en llamar la Era Informática toma aún mayor sentido la conocida frase de Bacon (1561-1626) *Nam et ipsa scientia potestas est (La información es poder).*

Las formas de llevar a cabo esa protección han variado mucho a lo largo de los años. De hecho, dada la actual exigencia de preservar la información mediante procesos inmediatos, los viejos procedimientos de la Criptografía Clásica se han visto relegados a meras curiosidades precientíficas. A pesar de que otros conceptos criptográficos gozan de mayor popularidad, no cabe duda de que es el Cifrado en Flujo la única solución factible para optimizar el recurso tiempo. Su origen es práctica y tecnológicamente la plasmación del único cifrado matemáticamente perfecto: el Cifrado de Vernam. Las secuencias aleatorias propuestas por Vernam [169] son sustituidas en este caso por secuencias seudoaleatorias, nacidas de la tecnología informática. Sus métodos de generación han sido analizados en profundidad durante casi dos décadas debido a su manifiesta utilidad en una amplia variedad de situaciones tecnológicas que incluyen la Seguridad y Eficiencia de las Comunicaciones, la Simulación de Procesos Aleatorios o la Generación de Códigos entre otras.

Entre las características que se le exigen a una secuencia seudoaleatoria para ser de utilidad criptográfica destaca la de una alta complejidad lineal (cantidad de secuencia necesaria para describir el resto de la secuencia). Por eso se hace necesario que a las secuencias generadas con un simple Registro de Desplazamiento con Realimentación Lineal (RDRL) se les aplique una transformación no lineal conocida con el nombre de Filtrado no Lineal. Dicho generador resulta óptimo en casi todos los aspectos, sin embargo paradójicamente se han llevado a cabo pocos esfuerzos en la investigación sobre la complejidad lineal resultante debido fundamentalmente a la dificultad que su análisis plantea.

Por tanto el propósito fundamental de la presente memoria es el estudio de la complejidad lineal de varios casos de filtrados no lineales. En particular se hace hincapié en aquéllos con un único término de orden máximo. Esto se debe a que gran parte de este trabajo puede considerarse como una continuación del trabajo realizado por Rueppel con su 'test de presencia de raíces' [152].

Por estar las técnicas que se exponen en sus comienzos, no se pretende

dar ninguna solución cerrada, sino que más bien se abre la posibilidad de continuar la investigación en el futuro según las directrices que aquí se inician.

Para su desarrollo se ha dividido la presente memoria en cuatro capítulos organizados de la siguiente forma:

En el capítulo 1, con el propósito de que la memoria sea autocontenida, se efectúa en primer lugar una revisión de los conceptos fundamentales que se manejan. En segundo lugar se lleva a cabo un recorrido bibliográfico por los trabajos que conforman el punto de partida para las nuevas teorías que se plantean y desarrollan en los capítulos posteriores. Además se añaden algunas rectificaciones y ampliaciones a las teorías de cada autor.

El capítulo 2 se dedica íntegramente a la deducción teórica de cotas de la complejidad lineal del filtrado. Para ello se explotan dos maneras duales de enfocar el problema. Por un lado se obtienen para un amplio grupo de funciones no lineales cotas del valor de complejidad lineal muy generales. Por otro lado se determina un grupo de funciones no lineales para las que se tiene garantizado un alto valor de complejidad lineal. Los resultados obtenidos de ambas maneras se presentan al final del capítulo en forma de prácticas sugerencias para la elección del filtrado.

En el capítulo 3 se desarrolla una técnica para el cálculo de cotas inferiores a la complejidad lineal de los filtrados no lineales. A diferencia de los esquemas existentes en los que se requiere el cálculo de determinantes en cuerpos finitos, este método se basa únicamente en operaciones lógicas sobre cadenas binarias. Esta técnica queda plasmada en dos algoritmos. Ambos resultan ser de gran aplicabilidad ya que sólo requieren que el filtrado tenga un único término de orden máximo. Además, los resultados obtenidos permiten acotar inferiormente la complejidad lineal mediante una curva polinómica, lo que queda reflejado en estimaciones de complejidades lineales muy altas, válidas para muchos casos prácticos.

En el capítulo 4 se plantean dos procedimientos alternativos cuyas bases se sitúan en los trabajos de Key [85] y Kumar y Scholtz [90]. Con ambos análisis se pone de manifiesto al mismo tiempo la posibilidad de enfocar el problema utilizando herramientas muy diversas, y la gran dificultad que entraña el estudio exhaustivo del valor concreto de la complejidad lineal.

Por último y a modo de apéndice figuran los diagramas de flujo de los algoritmos y los listados de los programas realizados para el cálculo numérico.

Capítulo 1

Conceptos Básicos y Antecedentes

Este capítulo aporta las herramientas necesarias para la investigación realizada. En las dos primeras secciones se establecen los conceptos y resultados básicos correspondientes respectivamente a las estructuras lineales y no lineales que se manejan. Obsérvese que muchas definiciones sencillas se exponen de forma abreviada entre paréntesis.

En la última sección se hace un recorrido bibliográfico donde se destacan aquellos resultados que constituyen el punto de partida de las nuevas teorías que se desarrollan en capítulos posteriores. En esta parte se incluyen además para cada autor referenciado, una serie de observaciones tales como rectificaciones y ampliaciones.

1.1. Teoría Lineal

El único cifrado incondicionalmente seguro conocido hasta el momento es el cifrado de Vernam o de cinta aleatoria. Ahora bien, dada la longitud de la clave necesaria, este tipo de cifrado resulta poco práctico. De ahí la tendencia a usar su esquema de cifrado y descifrado para diseñar nuevos sistemas que eviten el problema de la longitud de la clave. Esa es la idea de los **cifrados en flujo** cuyo esquema de funcionamiento se muestra en la figura 1.

La situación ideal para la seguridad de estos esquemas sería que la **secuencia de clave** o **secuencia cifrante** fuera aleatoria, tal y como ocurre

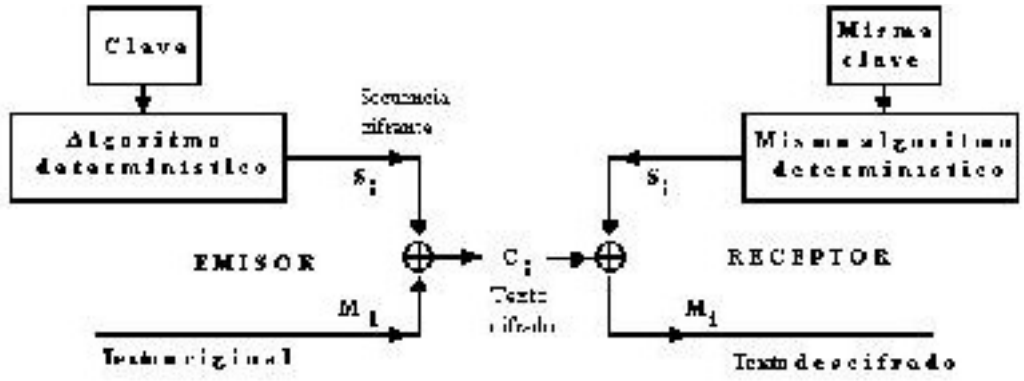


Figura 1.1: Cifrado en Flujo

en el cifrado de Vernam. Pero dada la necesidad práctica de usar un algoritmo determinístico para generar la secuencia, se tiene que ésta es siempre una secuencia finita y por tanto nunca puede ser considerada verdaderamente aleatoria. Para garantizar su seguridad criptográfica, a estas secuencias se les exigen una serie de características que las hacen parecer aleatorias. Las secuencias que las verifican se conocen con el nombre de **seudoaleatorias**. Las propiedades que deben cumplir están contenidas en los llamados **postulados de aleatoriedad** que Golomb formula en [68] y que se pueden describir en el caso binario de la forma siguiente:

G1. En cualquier secuencia de periodo p, el número de unos debe ser casi igual al número de ceros, es decir, $\frac{p}{2}$ si p es par y $\frac{p\pm1}{2}$ si p es impar.

G2. En cualquier secuencia periódica, aproximadamente la mitad de las rachas debe tener longitud uno, la cuarta parte de las rachas longitud dos, la octava parte longitud tres y así sucesivamente. Además, para cada una de estas longitudes deben existir tantas rachas de ceros como de unos.

G3. La función de autocorrelación debe tomar los valores siguientes:

$$\text{Si } p|t, AC(t) = 1.$$

$$Si\ p\ \not|t,\ AC(t) = \frac{A-D}{p} = \begin{cases} -1/(p-1) & si\ p\ es\ par \\ -1/p & si\ p\ es\ impar \end{cases},$$

siendo A y D respectivamente el número de coincidencias y diferencias entre $\{s_n\}$ y $\{s_{n+t}\}$

Dada una secuencia $\{s_n\}$, la secuencia $\{s_{n+t}\}$=$s_t, s_{t+1}, s_{t+2}, ...$ se llama **desplazamiento de fase** o **fase** t-ésima de $\{s_n\}$, y la secuencia $\{s_{n\cdot d}\}$=$s_0, s_d, s_{2d}, ...$ se llama **decimación** d de la secuencia $\{s_n\}$.

El último postulado G3 implica que el recuento de coincidencias entre una secuencia $\{s_n\}$ y su desplazamiento de fase $\{s_{n+t}\}$ no debe proporcionar ninguna información sobre el periodo de la secuencia salvo cuando t es múltiplo del periodo.

Para comprobar la seudoaleatoriedad de las secuencias se utilizan contrastes de hipótesis tales como el test de frecuencias (para comprobar G1), el test de las rachas y el test serial (para G2) y el test de correlación (para G3). Dado que el objetivo de este trabajo no es el estudio de estos tests, para mayor información sobre los mismos pueden consultarse diversas fuentes como [102] y [88].

Por otra parte, además de la aleatoriedad, existen otras propiedades que una secuencia debe cumplir para ser de utilidad criptográfica [112], tales como:

C1. El periodo p de la secuencia debe ser muy grande (aproximadamente del orden de 10^{50}).

C2. La secuencia ha de ser fácil de generar.

C3. El conocimiento de una parte de la secuencia cifrante no debe permitir a un criptoanalista generar la secuencia completa.

Después de haber dejado claras las propiedades que debe cumplir una secuencia cifrante, se estudiará en el próximo apartado el más extendido de todos los métodos de generación de dichas secuencias, los llamados registros de desplazamiento realimentados.

1.1.1. Registros de Desplazamiento Realimentados

Existen muchas y variadas aplicaciones de las secuencias en cuerpos finitos cuyos términos dependen de una forma sencilla de sus predecesores. Una ventaja computacional de estas secuencias reside en su facilidad para ser generadas mediante procedimientos recursivos. Además, tal y como veremos, estas secuencias presentan algunas propiedades estructurales muy útiles. En particular dichas secuencias resultan de gran interés tanto en Criptografía como en Teoría de la Codificación y algunas ramas de la Ingeniería Eléctrica.

Los cuerpos finitos a los que se hace referencia en este trabajo son los cuerpos de Galois GF(q), formados por los enteros {0,1,...,q-1} siendo q un número primo (cuerpo con la suma y el producto módulo q), y los cuerpos extensión finita de éste, $GF(q^n)$. En concreto este trabajo se dedica exclusivamente al caso binario (q=2).

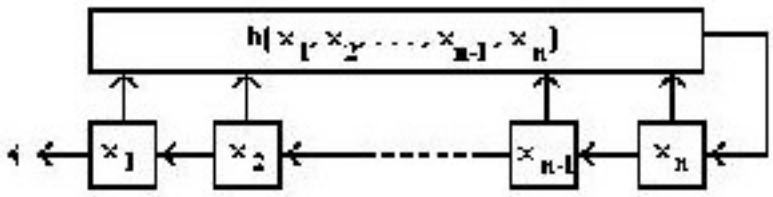


Figura 1.2: RDR

Los generadores de las secuencias mencionadas responden todos a la forma general $g(x_1, x_2, ..., x_n) = (x_2, x_3, ..., x_{n+1})$, donde g es una aplicación de $(\mathrm{GF}(2))^n$ en $(\mathrm{GF}(2))^n$. En particular resultan de mayor utilidad práctica aquéllos tales que $\mathrm{g}(x_1, x_2, ..., x_n) = (x_2, x_3, ..., h(x_1, x_2, ..., x_n))$, siendo h una aplicación de $(\mathrm{GF}(2))^n$ en GF(2). Dichos generadores se denominan **registros de desplazamiento realimentados** (**RDR** ó **FSR** del inglés Feedback Shift Register) y pueden implementarse fácilmente mediante circuitos electrónicos.

La figura 2 muestra el esquema general de un generador de este tipo.

A intervalos periódicos determinados por un reloj, los contenidos de x_i son transferidos a x_{i-1}. Para obtener cada nuevo valor de x_{n+1} se calcula una función $\mathrm{h}(x_1, x_2, ..., x_n)$ de todos los términos presentes en el registro. De esta forma la salida del generador resulta ser una secuencia de elementos recibidos en cada intervalo de una unidad de tiempo.

Entre estos generadores, los más estudiados y utilizados son los lineales. Se habla de **registro de desplazamiento con realimentación lineal** (**RDRL** ó **LFSR** del inglés Linear Feedback Shift Register) cuando la función de realimentación h se puede expresar de una forma lineal mediante $h(x_1, x_2, ..., x_n) = c_1 x_n + c_2 x_{n-1} + \cdots + c_n x_1$.

La notación de la figura 3 será la utilizada a lo largo de todo el trabajo.

Tanto los coeficientes c_i como los dígitos de salida s_j son elementos del cuerpo de Galois GF(2). El número L de celdas se llama **longitud** del RDRL,

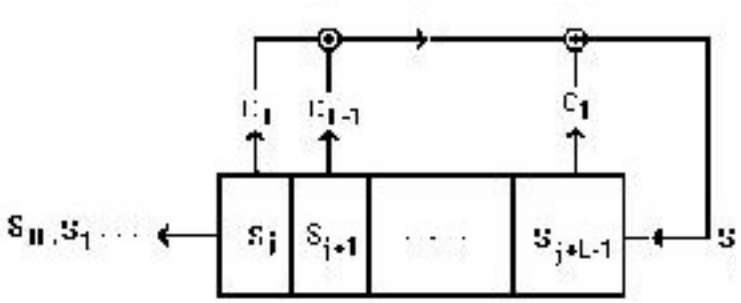


Figura 1.3: RDRL

y cada celda se llama **etapa** del RDRL. A cada vector formado por los contenidos de las L etapas $s_j, ..., s_{j+L-1}$ se le denomina **estado** del RDRL y se le denota mediante $\hat{s}_j$. El vector $\hat{s}_0$ formado por los L dígitos $s_0, ..., s_{L-1}$ inicialmente cargados en las L etapas se llama **estado inicial** del RDRL. Si se representan como nodos cada uno de los posibles estados y se unen mediante arcos aquellos estados que van uno a continuación de otro, al digrafo resultante se le llama **diagrama de transición de estados**.

Los coeficientes c_i se caracterizan mediante el llamado **polinomio de realimentación** o de conexión, $C(x)= 1 + c_1x + c_2x^2 + \cdots + c_Lx^L$ que tiene como grado máximo L. Se denota mediante **s**=$\{s_n\}_{n\in N}$ a la secuencia producida por el RDRL. Los dígitos s_j $(j > L-1)$ se determinan a partir del RDRL y de su estado inicial según la expresión $s_{j+L} = \sum_{i=1}^{L} c_i s_{j+L-i},\ \forall j \geq 0$, ó también $s_{j+L} + c_1 s_{j+L-1} + \cdots + c_L s_j = 0$, conocida como **relación de recurrencia lineal** de orden L.

Asociada a dicha relación se tiene la matriz cuadrada de orden L y elementos binarios,

$$\mathrm{A}=\begin{pmatrix} 0 & 0 & ... & 0 & c_1 \\ 1 & 0 & ... & 0 & c_2 \\ 0 & 1 & ... & 0 & c_3 \\ .. & . & ... & . & . \\ 0 & 0 & ... & 0 & c_{L-1} \\ 0 & 0 & ... & 1 & c_L \end{pmatrix}.$$

Mediante esta matriz se puede describir cada estado del RDRL en función

del estado inicial según la expresión $\widehat{s}_j = \widehat{s}_0 \mathrm{A}^j$.

Si $c_L \neq 0$ se dice que el RDRL es **no singular**. En ese caso, cualquiera de los posibles estados del RDRL tiene un predecesor único, por lo que su diagrama de transición de estados está formado por varias componentes conexas que son circuitos. Obsérvese que en ese caso cualquier secuencia $\{s_n\}$ producida por un RDRL con un determinado estado inicial también puede obtenerse mediante un desplazamiento de fase $\{s_{n+t}\}$ generado por el mismo RDRL pero con distinto estado inicial. El número de tales desplazamientos de fase coincide con el número de nodos de la componente conexa a la que pertenece el estado inicial de partida.

Se define la **D-transformada** o **función generatriz** S(x) de la secuencia $\{s_n\}$ como la serie $S(x) = \sum_{j=0}^{\infty} s_j x^j = s_0 + s_1 x + s_2 x^2 + \cdots$. La idea subyacente en esta expresión consiste en que en ella quedan almacenados de forma ordenada todos los términos de la secuencia, luego de alguna forma refleja sus propiedades.

Mediante esta función y el polinomio de realimentación se puede construir el polinomio, P(x)=C(x)S(x), llamado **polinomio de estado inicial**. Gracias a la relación de recurrencia lineal se puede demostrar que este polinomio es siempre de grado menor que L. La expresión que lo describe se conoce como **identidad fundamental**.

Según [71], si mcd(P(x),C(x))=1, entonces el RDRL de polinomio de realimentación C(x) es el de menor longitud que puede usarse para generar la secuencia $\{s_n\}$ que determina la función S(x). Esta propiedad jugará un papel fundamental en el análisis de Fourier que se presentará en el apartado 1.2.2.

Una de las características más destacables de las secuencias recurrentes lineales en un cuerpo finito es que, después de un posible comportamiento irregular, siempre terminan manifestando su naturaleza periódica.

Se denomina **periodo** de la secuencia $\{s_n\}$ al menor entero p tal que $\{s_{n+p}\}= \{s_n\}$ $\forall n > n_0$ y se llama **comienzo no periódico** de $\{s_n\}$ a la subsecuencia $s_0, s_1, \ldots, s_{n_0}$.

Teorema 1.1.1

Toda secuencia producida por un RDRL de longitud L es periódica y su periodo p cumple que $p \leq 2^L - 1$.

Demostración Dado un estado inicial, si uno de los estados del RDRL es el estado nulo, entonces el periodo es exactamente p=1$\leq 2^L$-1 ya que la

secuencia se convierte en la secuencia nula después de pasar por el estado nulo. Supóngase ahora que el RDRL no pasa por el estado nulo. Excluyendo el vector nulo existen exactamente $2^L - 1$ cadenas binarias distintas de longitud L. Por tanto, si se consideran los 2^L primeros estados $\widehat{s}_j$ del RDRL, se tiene forzosamente que $\widehat{s}_j = \widehat{s}_i$ para algún par de valores i y j tales que $0 \leq i < j \leq 2^L - 1$. A partir de esa igualdad, usando la relación de recurrencia lineal y un procedimiento de inducción se llega a que $\widehat{s}_{n+j-i} = \widehat{s}_n \ \forall n \geq i$, lo que demuestra la periodicidad de la secuencia y que el menor periodo es $p \leq j - i \leq 2^L - 1$.

Concretamente, si el RDRL es no singular se tiene que la secuencia producida es periódica desde el principio, es decir, no contiene un comienzo no periódico. Por eso en este trabajo sólo se consideran RDRLs no singulares.

Ejemplo 1.1.1

Dados L=4, $c_1 = c_2 = 0, c_3 = c_4 = 1$ y el estado inicial (1,0,0,0) se tiene el siguiente RDRL de polinomio de realimentación C(x)=1+x^3+x^4.

A continuación se muestran los sucesivos estados del RDRL a cada golpe de reloj.

reloj	s_0	s_1	s_2	s_3	*reloj*	s_0	s_1	s_2	s_3	*reloj*	s_0	s_1	s_2	s_3
0	*1*	*0*	*0*	*0*	*5*	*0*	*0*	*1*	*1*	*10*	*1*	*0*	*1*	*1*
1	*0*	*0*	*0*	*1*	*6*	*0*	*1*	*1*	*0*	*11*	*0*	*1*	*1*	*1*
2	*0*	*0*	*1*	*0*	*7*	*1*	*1*	*0*	*1*	*12*	*1*	*1*	*1*	*1*
3	*0*	*1*	*0*	*0*	*8*	*1*	*0*	*1*	*0*	*13*	*1*	*1*	*1*	*0*
4	*1*	*0*	*0*	*1*	*9*	*0*	*1*	*0*	*1*	*14*	*1*	*1*	*0*	*0*

Su diagrama de transición de estados está formado por el nodo que representa el estado nulo y el circuito con los 15 estados distintos.

La D-transformada queda de la forma S(x)=1+x^4+x^7+x^8+$\cdots$, luego el polinomio de estado inicial es P(x)=(1+ x^4+ x^7+ x^8+ $\cdots$)+ (x^3+ x^7+ x^{10}+ $\cdots$)+ (x^4+ x^8+ x^{12}+ $\cdots$)= 1+ x^3.

1.1.2. Polinomio Característico y Polinomio Minimal

Con el fin de analizar la periodicidad de las secuencias generadas, a continuación se define un polinomio asociado a cada RDRL mediante el cual se pueden describir los términos de las secuencias. En esta sección también se

describen las secuencias de máximo periodo que, según el requerimiento C1, son las que se utilizan en criptografía.

Al polinomio mónico asociado al RDRL definido en GF(2)[x] mediante $c(x) = x^L + c_1x^{L-1} + \cdots + c_{L-1}x + c_L$ se le conoce como **polinomio característico del RDRL**.

Este polinomio sólo depende de la relación de recurrencia lineal, por lo que puede ser expresado en función de la matriz A según c(x)=det(x·I_L+A). Por otra parte, la relación que le une con el polinomio de realimentación viene dada por C(x)=x^Lc(x^{-1}), es decir, son polinomios recíprocos.

Como primera aplicación de este polinomio se presenta la siguiente representación de los términos de la secuencia.

Teorema 1.1.2

Dada una secuencia binaria $\{s_n\}$ *producida por un RDRL de longitud L y polinomio característico c(x), si las L raíces de c(x)* $\alpha_1, ..., \alpha_L$ *son distintas, entonces* $s_n = \sum_{j=1}^{L} \beta_j \alpha_j^n$, $\forall n = 0, 1, ...$ *siendo* $\beta_1, ..., \beta_L$ *elementos de GF(*2^L*) unívocamente determinados por el estado inicial.*

Demostración El estado inicial se puede expresar según las ecuaciones $s_n = \sum_{j=1}^{L} \beta_j \alpha_j^n$, n=0,1,...,L-1 ya que el determinante asociado a este sistema es un determinante de Vandermonde no nulo por ser las raíces α_j todas distintas. Los restantes elementos definidos según esa relación verifican la relación de recurrencia lineal $s_{n+L} + c_1 s_{n+L-1} + \cdots + c_L s_n = \sum_{j=1}^{L} \beta_j \alpha_j^{n+L} + c_1 \sum_{j=1}^{L} \beta_j \alpha_j^{n+L-1} + \cdots + c_L \sum_{j=1}^{L} \beta_j \alpha_j^n = \sum_{j=1}^{L} \beta_j \alpha_j^n c(\alpha_j) = 0$, $\forall n$. Por tanto, ya que cada relación de recurrencia lineal y estado inicial determinan unívocamente una secuencia, se tiene que la secuencia definida mediante esa expresión coincide con la secuencia $\{s_n\}$ generada por el RDRL.

En adelante se usará la notación Tr_1^L para representar a la función traza $\mathrm{Tr}_{GF(2^L)/GF(2)}$ que aplica cada elemento $\beta \in GF(2^L)$ en el cuerpo GF(2) de la forma $\mathrm{Tr}_{GF(2^L)/GF(2)}(\beta) = \beta + \beta^2 + \beta^{2^2} + \cdots + \beta^{2^{L-1}}$. Esta función traza no es sólo una transformación lineal de GF(2^L) sobre GF(2), sino que además sirve como descripción de cualquier otra transformación lineal entre ambos cuerpos [101].

A continuación se muestra un resultado análogo al teorema 1.2 para el caso en el que el polinomio característico sea irreducible. En este caso todas las raíces de c(x) corresponden a distintas potencias de una de ellas y los elementos de la secuencia se expresan en términos de una función traza.

Teorema 1.1.3

Dadas una secuencia binaria $\{s_n\}$ producida por un RDRL de longitud L y polinomio característico c(x) irreducible sobre GF(2) y $\alpha \in GF(2^L)$ una raíz de c(x), entonces existe un único elemento $A \in GF(2^L)$ tal que

$$s_n = Tr_1^L(A\alpha^n) = \sum_{i=0}^{L-1} (A\alpha^n)^{2^i}, \ \forall n = 0, 1, ...$$

Demostración Dado que $\{1,\alpha,\alpha^2,...,\alpha^{L-1}\}$ es una base de GF(2^L) sobre GF(2) por ser c(x) irreducible, se puede definir una aplicación lineal f de GF(2^L) sobre GF(2) de la forma $f(\alpha^n) = s_n$ ($\forall n = 0, 1, ..., L-1$). Según lo ya comentado [101], toda transformación lineal de GF(2^L) sobre GF(2) se puede describir mediante una función traza. En particular, la transformación definida se puede expresar de la forma $f(\alpha^n) = s_n = Tr_1^L(A\alpha^n)$ para un único A$\in GF(2^L)$, [101]. La secuencia así definida coincide con la secuencia generada por el RDRL de polinomio característico c(x) ya que verifica la relación de recurrencia lineal $Tr_1^L(A\alpha^{n+L}) + c_1 Tr_1^L(A\alpha^{n+L-1}) + \cdots + c_L Tr_1^L(A\alpha^n) = Tr_1^L(A\alpha^{n+L} + c_1 A\alpha^{n+L-1} + \cdots + c_L A\alpha^n) = Tr_1^L(c(\alpha)A\alpha^n) = 0 \ \forall n \geq 0$, y posee el mismo estado inicial.

Según la expresión de la secuencia dada en el teorema anterior, cada uno de los 2^L posibles valores del coeficiente A determina una secuencia diferente que puede ser generada mediante el RDRL. Dado que existen exactamente 2^L posibles estados iniciales del RDRL, se puede establecer una biyección entre los coeficientes $A \in$GF(2^L) y los posibles estados iniciales del RDRL. Concretamente, para cada estado inicial, mediante la expresión del teorema 1.3 se define un sistema de ecuaciones lineales que permite calcular el coeficiente A correspondiente. Por tanto siempre se puede escoger como estado inicial aquél que produzca el coeficiente A=1, quedando en ese caso la expresión anterior de la forma $s_n = Tr_1^L(\alpha^n)$. Ésta será ampliamente utilizada en el último capítulo de este trabajo.

En este punto hacemos un inciso en el desarrollo normal de la memoria para introducir varios conceptos de Teoría de Galois [125].

Sea $\beta \in \mathrm{GF}(2^L)$ entonces los elementos $\beta, \beta^2, \beta^{2^2}, ..., \beta^{2^{L-1}}$ se llaman **raíces conjugadas** de β respecto a GF(2). Si $\alpha \in \mathrm{GF}(2^L)$ es un elemento primitivo de $\mathrm{GF}(2^L)$ (su orden, 2^L-1, coincide con el orden de $\mathrm{GF}(2^L)$), entonces los elementos no nulos de $\mathrm{GF}(2^L)$ se pueden expresar como potencias de α. Siempre se puede definir una partición del conjunto de elementos β no nulos de $\mathrm{GF}(2^L)$ en conjuntos disjuntos de raíces conjugadas. Los conjuntos disjuntos de exponentes de esas raíces conjugadas expresadas como potencias de un elemento primitivo jugarán un papel fundamental en el desarrollo de esta memoria.

Se define ahora la siguiente relación de equivalencia sobre $\mathrm{Z}^*_{2^L-1}$: $\mathrm{E}_1 \sim \mathrm{E}_2$, $\mathrm{E}_1, \mathrm{E}_2 \in \mathrm{Z}^*_{2^L-1}$ si y sólo si $\exists$i, $0 \leq i \leq L-1$ tal que $\mathrm{E}_1 2^i \equiv \mathrm{E}_2 \bmod(2^L\text{-}1)$. Las clases de equivalencia resultantes de esta partición coinciden con los conjuntos disjuntos de exponentes de las raíces conjugadas anteriormente mencionadas.

Definición 1.1.1

Se denomina **coset** *al conjunto de todos los enteros de la forma* $E \cdot 2^i \bmod(2^L\text{-}1)$ *siendo* $0 \leq i \leq L-1$ *y* $E \in Z^*_{2^L-1}$.

Observación 1.1.1

En este trabajo usamos la terminología coset por razones de conveniencia y simplicidad en lugar de una traducción como podría ser clase de equivalencia.

Por extensión a veces también se asigna este nombre a los conjuntos de raíces conjugadas. Dado que los cosets son disjuntos, siempre podemos referirnos al coset que contiene un elemento E como **coset E**. Al menor de sus elementos se le conoce como **líder** del coset. Según la definición de coset se deduce que, si se expresan dichos enteros en base 2, todos los elementos de un mismo coset tienen exactamente el mismo peso de Hamming (número de unos en su representación binaria). A este valor se le llama **peso del coset**. Obsérvese que en la expresión de s_n dada en el teorema 1.3 están incluidos todos los elementos del coset $\{1, 2, 2^2, ..., 2^{L-1}\}$ de peso 1. Se llama **cardinal del coset** al número de elementos del coset.

Golomb en [68] distingue los cosets E tales que $\mathrm{mcd}(\mathrm{E}, 2^L - 1) = 1$ del resto. A los primeros los llama **cosets propios** y a los demás, **cosets impropios**.

Ejemplo 1.1.2

Para $L = 4, 2^L - 1 = 15 = 3 \cdot 5$, *GF(*$2^4$*)* $= \{0, 1, \alpha, \alpha^2, ..., \alpha^{14}\}$, *GF(*$2^4$*)*-$\{0\} = \{\{1\}, \{\alpha, \alpha^2, \alpha^4, \alpha^8\}, \{\alpha^3, \alpha^6, \alpha^{12}, \alpha^9\}, \{\alpha^5, \alpha^{10}\}, \{\alpha^7, \alpha^{14}, \alpha^{13}, \alpha^{11}\}\}$

$$\textit{Coset } E_1 : 1, 2, 4, 8 \ : 0001, 0010, 0100, 1000 \textit{ coset propio de peso 1}$$
$$\textit{Coset } E_2 : 3, 6, 12, 9 \ : 0011, 0110, 1100, 1001 \textit{ coset impropio de peso 2}$$
$$\textit{Coset } E_3 : 5, 10 \ : 0101, 1010 \textit{ coset impropio de peso 2}$$
$$\textit{Coset } E_4 : 7, 14, 13, 11 : 0111, 1110, 1101, 1011 \textit{ coset propio de peso 3}$$

Tanto la caracterización dada en el teorema 1.3 como los cosets recien definidos juegan un papel fundamental en el tema que se analiza ya que constituyen la base de los análisis de algunos autores y de gran parte del presente trabajo.

Tal y como ya se ha mencionado, el periodo es una de las características más relevantes de las secuencias cifrantes. Dado que el periodo de las secuencias producidas por cualquier RDRL de longitud L está acotado por 2^L-1, a continuación se describen las secuencias cuyo periodo coincide exactamente con ese valor.

Si el periodo de una secuencia producida por un RDRL de longitud L es 2^L-1, se dice que es una **secuencia de longitud máxima**, **m-secuencia** o **PN-secuencia**, y también que el RDRL que la genera es un **RDRL de longitud máxima**. Un RDRL no singular de longitud L genera una m-secuencia si y sólo si su polinomio característico es primitivo (polinomio de menor grado tal que una de sus raíces es un elemento primitivo de GF(2^L)) y el estado inicial es no nulo.

Toda secuencia producida por un RDRL satisface varias relaciones de recurrencia lineal aparte de la usada para generarla. Por ejemplo, dado el periodo p de una secuencia, siempre se tiene la relación válida $s_{n+p} = s_n$, $\forall n = 0, 1, ...$ En el caso extremo de la secuencia nula se tiene que satisface cualquier relación de recurrencia.

Dada una secuencia binaria producida por un RDRL cualquiera, al polinomio característico de menor orden posible que permita describir una relación de recurrencia lineal válida para dicha secuencia se le conoce como **polinomio minimal de la secuencia**. Este polinomio puede ser de menor o igual grado que el polinomio característico del RDRL usado para generar la secuencia, y siempre divide a los polinomios característicos de todos los RDRLs que sirvan para generarla.

El polinomio minimal es de importancia crucial para las secuencias producidas por un RDRL dado que su orden (menor entero positivo t tal que el polinomio divide a $(x^t - 1)$) coincide con el periodo de la secuencia.

Teorema 1.1.4

Dada una secuencia $\{s_n\}$ de polinomio minimal m(x)∈GF(2)[x], entonces el periodo de $\{s_n\}$ es igual a ord(m(x)).

Demostración Si p es el periodo de la secuencia $\{s_n\}$ y n_0 es la longitud de su comienzo no periódico, entonces $s_{n+p} = s_n,\ \forall n \geq n_0$ y en particular $s_{n+n_0+p} = s_{n+n_0},\ \forall n \geq 0$. Entonces, por su condición de polinomio minimal de la secuencia, m(x) divide a x^{n_0+p}-x^{n_0}, luego m(x)=x^hg(x) con h≤ n_0 y g(x)∈GF(2)[x] que divide a $x^p - 1$. Por tanto ord(m(x))= ord(g(x))≤ p. Por otro lado, dado que el periodo p siempre divide a ord(m(x)) [101], se tiene la igualdad.

A continuación se da un criterio para descubrir si un polinomio característico de un RDRL corresponde o no al polinomio minimal de la secuencia generada.

Teorema 1.1.5

Dada una secuencia binaria $\{s_n\}$ no nula generada por un RDRL de longitud L y polinomio característico c(x)∈GF(2)[x], entonces c(x) es el polinomio minimal de la secuencia si y sólo si los estados $\widehat{s}_0$,$\widehat{s}_1$,...,$\widehat{s}_{L-1}$ del RDRL son linealmente independientes sobre GF(2).

Demostración '$\Longrightarrow$' Se procede por reducción al absurdo. Si $\widehat{s}_0$,$\widehat{s}_1$,...,$\widehat{s}_{L-1}$ fueran linealmente dependientes sobre GF(2), se tendría que $b_0\widehat{s}_0 + b_1\widehat{s}_1 + \cdots + b_{L-1}\widehat{s}_{L-1} = 0,\ b_0, b_1, ...b_{L-1} \in$GF(2). Multiplicando dicha ecuación por la matriz A^n se tiene que $b_0\widehat{s}_n + b_1\widehat{s}_{n+1} + \cdots + b_{L-1}\widehat{s}_{n+L-1} = 0,\ \forall n \geq 0$. Si b_j=0 $\forall j \neq 0$, se obtendría la secuencia nula $s_n = 0\ \forall n \geq 0$, cosa que es contradictoria con las hipótesis. Si j≥ 1 es el mayor valor tal que $b_j \neq 0$, entonces la secuencia satisface una relación de recurrencia lineal de orden $j < L$, lo que entra en contradicción con que c(x) sea el polinomio minimal de la secuencia. Luego $\widehat{s}_0$,$\widehat{s}_1$,...,$\widehat{s}_{L-1}$ son linealmente independientes sobre GF(2).

'$\Longleftarrow$' Se supone por reducción al absurdo que c(x) no es el polinomio minimal de la secuencia. En ese caso la secuencia satisface una relación lineal de orden m con $1 \leq m < L$, lo que contradice la independencia lineal de los L estados.

A lo largo de esta memoria se trabajará siempre con PN-secuencias, es decir, con secuencias tales que su polinomio minimal es primitivo. Además se considerará siempre el RDRL cuyo polinomio característico es el polinomio minimal de la secuencia.

1.1.3. Propiedades de las PN-secuencias

En este apartado se analiza si las PN-secuencias binarias (generadas mediante un RDRL no singular de polinomio primitivo y estado inicial no nulo) cumplen o no los postulados de Golomb y los requerimientos criptográficos comentados en el apartado 1.1. Estas secuencias pasan con cierto margen los contrastes de hipótesis mencionados allí, tal y como se deduce de las siguientes observaciones hechas para cada uno de los postulados.

G1. Dado que el RDRL genera cíclicamente los 2^L-1 estados no nulos, cada estado no nulo ocurre exactamente una vez por periodo. Luego el número de unos por periodo es 2^{L-1} mientras que el de ceros es 2^{L-1}-1.

G2. Hay 2^{L-k-2} estados cuyos k+2 bits más significativos son 011...10 (k unos) y otros 2^{L-k-2} estados para el caso 100...01 (k ceros). Luego de longitud k$\leq$L-2 existen 2^{L-k-2} rachas, tanto de ceros como de unos. El estado 011...11 aparece exactamente una vez y su sucesor 111...11 va seguido a su vez por el estado 111...10, mientras que al estado 100...00 le sigue forzosamente 000...01. De todo esto se deduce que existe una racha de ceros y ninguna racha de unos de longitud L-1 y sólo hay una racha de unos y ninguna de ceros de longitud L.

G3. El número de coincidencias por periodo entre $\{s_n\}$ y $\{s_{n+t}\}$ viene dado por el número de ceros por periodo de $\{s_n+s_{n+t}\}$, considerando la suma módulo 2. Este número es 2^{L-1}-1 y en consecuencia el número de diferencias es 2^{L-1}. Por tanto la autocorrelación es AC(t)=$\frac{-1}{2^L-1}$, cuando $1\leq t < 2^L - 1$.

De lo anterior se concluye que las PN-secuencias cumplen satisfactoriamente los tres postulados de Golomb. Sin embargo, con respecto a las propiedades de naturaleza criptográfica a continuación se ve que, debido a la propiedad C3 del apartado 1.1, dichas secuencias no resultan suficientemente seguras como claves de un cifrado en flujo.

C1. Dado que el periodo de una PN-secuencia generada por un RDRL de longitud L es 2^L-1, se pueden conseguir fácilmente grandes periodos. Por ejemplo para L=166 el periodo 2^{166}-1 ronda el valor 10^{50}. Además ni siquiera resulta difícil encontrar, para cualquier valor de L, suficientes polinomios primitivos entre los que elegir el polinomio característico del RDRL. En par-

ticular, existen exactamente $\frac{\phi(2^L-1)}{L}$ polinomios primitivos de grado L. Por tanto, si se consideran iguales aquellas secuencias que difieran sólo en su estado inicial, se tienen $\frac{\phi(2^L-1)}{L}$ PN-secuencias distintas. Este número crece exponencialemente a medida que L crece, de manera que, mientras para L=11 hay 176, para L=24 hay 276480.

C2. Los RDRLs son muy fáciles de implementar en hardware ya que su estructura está basada en puertas simples.

C3. Las PN-secuencias resultan muy inseguras dado que el conocimiento de 2L dígitos consecutivos $s_k, s_{k+1}, ..., s_{k+2L-1}$ permite al criptoanalista determinar exactamente los coeficientes de realimentación c_i y de ahí la secuencia completa. Esto es posible mediante la ecuación matricial

$$\begin{pmatrix} s_{j+L-1} & s_{j+L-2} & \cdot & s_j \\ s_{j+L} & s_{j+L-1} & \cdot & s_{j+1} \\ \cdot\cdot & \cdot & \cdot & \cdot \\ s_{j+2L-2} & s_{j+2L-3} & \cdot & s_{j+L-1} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \cdot\cdot \\ c_L \end{pmatrix} = \begin{pmatrix} s_{j+L} \\ s_{j+L+1} \\ \cdot\cdot \\ s_{j+2L-1} \end{pmatrix}$$

Por el teorema 1.5 se tiene que cualesquiera L estados consecutivos son siempre linealmente independientes. Por tanto, la ecuación matricial anterior representa un sistema compatible de L ecuaciones cuyas soluciones son los coeficientes c_i.

De lo anterior se concluye que el único de estos requerimientos que no cumplen las PN-secuencias es el requerimiento C3, por tanto en lo que resta de capítulo se procederá de la siguiente manera:

1. Se presentarán herramientas que permitan analizar el requerimiento C3.

2. Se estudiarán con ellas otros tipos de generador de secuencia cifrante.

Con respecto a este último paso, dado que las PN-secuencias cumplen el resto de propiedades, resulta bastante natural aprovechar esas buenas cualidades de los RDRLs que las generan. Para ello lo lógico es utilizar estos RDRLs de una forma no lineal de manera que no se modifiquen dichas propiedades, pero sí se verifique el requerimiento C3. Esto se hará en la sección 1.2.

Por otro lado, el primer paso mencionado se desarrollará a lo largo de los dos siguientes apartados.

1.1.4. Complejidad Lineal

La complejidad lineal de una secuencia periódica es el concepto central de este trabajo.

Toda secuencia binaria de periodo p puede generarse siempre mediante un RDRL. Al menos siempre se puede considerar como estado inicial el primer periodo y como polinomio característico $1+x^p$.

En general se define la **complejidad lineal** de una secuencia periódica $\{s_n\}$ como la longitud del menor RDRL que puede utilizarse para generarla. A este RDRL se le conoce como **equivalente lineal** de la secuencia. El polinomio característico del equivalente lineal es el polinomio minimal de la secuencia $\{s_n\}$. En el caso de secuencias infinitas se habla de **complejidad lineal global** y se denota como $\Lambda(\{s_n\})$ mientras que en el caso de secuencias finitas $s^n = s_0, s_1, ..., s_{n-1}$ se habla de **complejidad lineal local** y se denota mediante $\Lambda(s^n)$ (ó también L_n). Evidentemente una secuencia finita s^n, realización concreta de una secuencia infinita $\{s_n\}$, tiene su complejidad lineal local acotada inferiormente por la complejidad lineal global de ésta, $\Lambda(s^n) \leq \Lambda(\{s_n\})$.

Dado que se trata de determinar la secuencia entera a partir del conocimiento de una parte, resulta evidente la importancia del estudio de la complejidad lineal como medida de impredecibilidad de la secuencia. Según lo comentado en el apartado anterior, siempre se puede obtener la secuencia completa $\{s_n\}$ a partir de $2 \cdot \Lambda(\{s_n\})$ dígitos conocidos.

Existe un algoritmo bien conocido para calcular la complejidad lineal local de una secuencia conocida a partir de sus dígitos. El propósito inicial del algoritmo diseñado por Berlekamp [4] era la decodificación de códigos BCH, pero posteriormente Massey [111] demostró su utilidad para el cálculo de la complejidad lineal local. A partir de entonces el algoritmo se conoce como **algoritmo de Berlekamp-Massey**.

El algoritmo de Berlekamp-Massey relaciona las complejidades lineales locales asociadas respectivamente a los primeros n y n-1 dígitos de una misma secuencia binaria $\Lambda(s^n)$ y $\Lambda(s^{n-1})$. Se llama **discrepancia** y se denota como δ_{n-1} a la diferencia entre el n-ésimo dígito de la secuencia analizada s_{n-1} y el n-ésimo dígito generado por el equivalente lineal de la secuencia s^{n-1}. Obviamente si el equivalente lineal de s^{n-1} también genera la secuencia s^n, entonces $\delta_{n-1}=0$ y ambas complejidades lineales locales coinciden $\Lambda(s^n)=\Lambda(s^{n-1})$. Si, por el contrario dicho RDRL no sirve para generar s^n, entonces $\delta_{n-1}=1$ y la complejidad lineal local de s^n es mayor que la de s^{n-1} cuando la de ésta es

menor que $\frac{n}{2}$. Por tanto, el proceso de actualización de la complejidad lineal local que se lleva a cabo en el algoritmo de Berlekamp-Massey, cuya completa descripción se puede encontrar en [112], se resume como sigue.

* *Si* $\delta_{n-1} = 0$ *entonces* $\Lambda(s^n) = \Lambda(s^{n-1})$
* *Si* $\delta_{n-1} = 1$ *entonces* $\begin{cases} \Lambda(s^n) = \Lambda(s^{n-1}) & si\ \Lambda(s^{n-1}) \geq \frac{n}{2} \\ \Lambda(s^n) = n + 1 - \Lambda(s^{n-1}) & si\ \Lambda(s^{n-1}) < \frac{n}{2} \end{cases}$

Por último hay que destacar que el algoritmo de Berlekamp-Massey es aplicable a todo tipo de secuencias binarias, tanto las generadas linealmente como las no lineales.

1.1.5. Perfil de Complejidad Lineal

La complejidad lineal de una secuencia es una medida de su impredecibilidad, mientras que el llamado perfil de complejidad lineal refleja su aleatoriedad.

Ya se ha mencionado que si una secuencia tiene periodo p entonces puede ser generada con un RDRL de longitud p, polinomio característico $1+x^p$ y estado inicial correspondiente al primer periodo de la secuencia. A este generador se le conoce como **RDRL cíclico puro**. De lo anterior se tiene que la complejidad lineal de una secuencia nunca puede superar el valor de su periodo.

Como se ha dicho en el apartado anterior, el algoritmo de Berlekamp-Massey sirve para calcular la complejidad lineal local L_n. Además, para calcular L_n este algoritmo necesita previamente calcular todas las complejidades lineales L_1, L_2,..., L_{n-1} de las subsecuencias s_0, s_0s_1,..., $s_0s_1...s_{n-2}$.

Se llama **perfil de complejidad lineal** (**PCL**) de la secuencia s^n al vector de complejidades lineales locales $(L_1, L_2, .., L_n)$. El PCL representa el comportamiento dinámico de la complejidad lineal local frente al número n de dígitos considerados. Se dice que el PCL tiene un **salto** en s_{k-1} si $L_k - L_{k-1} > 0$ y esta diferencia se llama **peso del salto**. A partir del algoritmo de Berlekamp-Massey se sabe que sólo puede haber salto en s_{k-1} si $2L_{k-1} < k$ y que si lo hay entonces $L_k = k + 1 - L_{k-1}$.

La secuencia 10001111010000110111110100010100 obtenida lanzando una moneda 31 veces puede ser utilizada para ilustrar el concepto de perfil de complejidad lineal como medida de aleatoriedad. Se considera la secuencia s obtenida repitiendo sucesivamente la secuencia anterior. Si se representa su PCL frente al número de dígitos n se observa que se aproxima a la recta $\frac{n}{2}$

desde n=1 hasta n=62. A partir de n=62 las sucesivas complejidades lineales locales permanecen constantes e iguales a 31 (periodo de la secuencia).

De ahí se puede deducir que el polinomio característico de menor orden que se puede usar para generar la secuencia es $1+x^{31}$ y por tanto que la complejidad lineal global de esta secuencia toma su valor máximo, el periodo.

Por otro lado, para mostrar que la exigencia de un alto valor de complejidad lineal no es condición suficiente para garantizar aleatoriedad, considérese ahora la secuencia formada por treinta ceros y un uno. Esta secuencia puede ser generada, igual que en el caso anterior, por el RDRL cíclico puro de longitud 31. Sin embargo, al contrario que en el caso anterior, esta secuencia no cumple ninguna de las propiedades de aleatoriedad. Esto puede asociarse a su PCL, pues en él las complejidades lineales locales se mantienen iguales a 0 hasta el caso n=31 en que pasa a valer 31, donde se mantiene para valores mayores de n.

Según estos sencillos ejemplos es fácil asociar la idea de aleatoriedad de una secuencia a la cercanía de su PCL a la recta $\frac{n}{2}$, cuestión demostrada teóricamente en [91]. Además hay que mencionar que la no aleatoriedad reflejada en el PCL ha demostrado ser de gran utilidad práctica ya que en muchos casos pasa desapercibida para los tests estadísticos.

No obstante como última indicación sobre el PCL hay que aclarar que aunque la cercanía a la recta $\frac{n}{2}$ es una cualidad necesaria para la aleatoriedad de la secuencia, esta condición no es suficiente ya que una regularidad excesiva en el PCL es incompatible con los postulados de Golomb. Sirva como muestra de ello la secuencia $1101000010^710^{15}10^{31}...$, donde 0^x representa una secuencia de x ceros consecutivos. Tal y como primero conjeturó Rueppel [152] y luego demostró Dai [27], esa secuencia tiene complejidad lineal local de valor $\left\lfloor \frac{n+1}{2} \right\rfloor$. Esta función ciertamente está muy cercana a $\frac{n}{2}$ pero su regularidad manifiesta claramente la no aleatoriedad de la secuencia, que queda por otra parte demostrada mediante contrastes de hipótesis.

1.2. Teoría no Lineal

Para intentar conseguir una alta impredecibilidad de las secuencias producidas, los generadores de secuencia cifrante normalmente incorporan uno o varios RDRLs y alguna transformación no lineal. Tal y como se vió en la sección anterior, la complejidad lineal proporciona una medida de dicha impredecibilidad. Por tanto resulta bastante conveniente poder analizar las

secuencias cifrantes en términos de su complejidad lineal. De ahí que un factor a tener en cuenta a la hora de diseñar un generador es la posibilidad de análisis de esta propiedad. En esta sección se presentan algunas herramientas teóricas que permiten calcular o acotar inferiormente la complejidad lineal global de algunas secuencias cifrantes.

La transformación no lineal más sencilla de todas es sin duda el producto de dos dígitos binarios (operación AND). Ésta encuentra su generalización natural en el caso de las funciones booleanas. De hecho existe una expresión canónica para las funciones booleanas llamada **forma algebraica normal** (**FAN**) que representa dichas funciones como sumas de productos. Dado x= $(x_1, x_2, ..., x_L) \in \{0,1\}^L$, la FAN de una función booleana no lineal f se puede escribir como

$$f(x) =$$
$$a_0 + a_1x_1 + \cdots + a_Lx_L + a_{1,2}x_1x_2 + \cdots + a_{L-1,L}x_{L-1}x_L + \cdots + a_{1,2,...,L}x_1x_2...x_L.$$

Se llama **producto de orden n** a un producto de n variables y se llama **orden de una función** booleana al mayor de los órdenes de todos los términos de su FAN.

La FAN representa un sistema de referencia válido para toda la teoría no lineal de las secuencias que, además resultará de gran utilidad ya que existe cierta relación entre el orden de la función no lineal y la complejidad lineal de la secuencia resultante.

Según el tipo de transformación no lineal utilizada se distinguen dos métodos de generación de secuencia cifrante. El llamado **filtrado no lineal** de un RDRL consiste en una función booleana no lineal que se aplica a las etapas de un RDRL tal y como se muestra en la figura 4.

En adelante se denota mediante $\mathbf{z} = \{\mathbf{z}_n\}_{n\in N}$ a la secuencia resultante a la salida de un filtrado no lineal f, llamada en muchos casos **secuencia filtrada**.

Se tiene que $z_0 = f(\widehat{s}_0)$, $z_1 = f(\widehat{s}_1)$,..., $z_{2^L-2} = f(\widehat{s}_{2^L-2})$.

En este caso la tendencia de la mayoría de autores, según se verá en la sección 1.3, ha sido centrarse en la búsqueda de conjuntos de funciones no lineales que produzcan secuencias de complejidad lineal global predecible. Se suele seguir esta vía porque en general, tal y como se menciona en [152], 'es extremadamente difícil acotar inferiormente (o garantizar) la complejidad lineal de las secuencias producidas por un filtrado no lineal de un RDRL'. Nosotros en este trabajo abordaremos este problema.

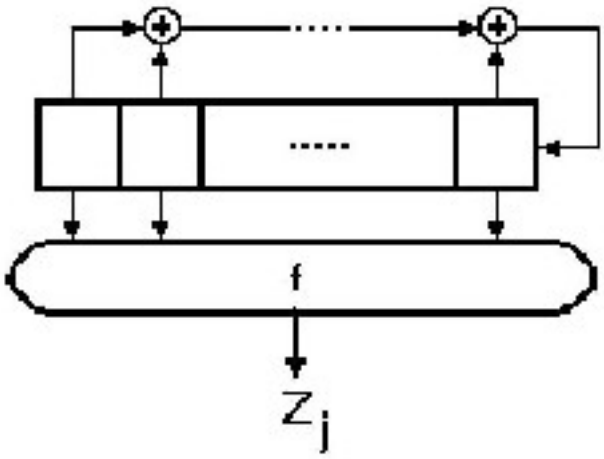


Figura 1.4: Filtrado no Lineal

El segundo método de generación de secuencias cifrantes utiliza la combinación no lineal de RDRLs. Las secuencias producidas por varios RDRLs se combinan mediante una función no lineal F obteniéndose un generador conocido como **combinador no lineal**, cuyo funcionamiento se muestra en la figura 5.

En el estudio de la complejidad lineal global del combinador no lineal el problema consiste generalmente en, dada una función no lineal F, encontrar tipos de RDRLs para los que el generador resultante tenga una complejidad lineal global predecible. Este problema es bastante más fácil que el anterior y prueba de ello es que ya se han encontrado ([85], [180], [153], [56]) reglas sencillas para la elección de los RDRLs que permiten un completo análisis de la complejidad lineal global del combinador no lineal resultante.

En ambos casos, filtrado y combinador no lineales:

a) Los RDRLs proporcionan a la secuencia de salida una buena distribución estadística y un gran periodo.

b) La función no lineal proporciona 'confusión' (no existe una relación sencilla entre el texto original y el texto cifrado) [158] y una gran complejidad lineal.

c) La clave secreta determina el estado inicial del RDRL(s) y puede también determinar la elección de la función no lineal concreta.

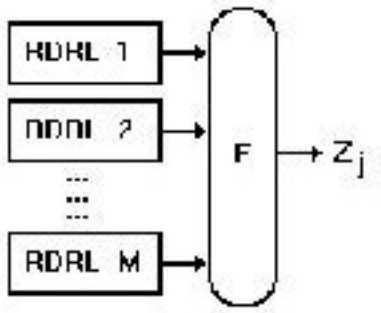


Figura 1.5: Combinador no Lineal

Dada la estructura del filtrado no lineal, la mejor complejidad lineal global que se puede obtener es $2^L - 1$. El interés de este trabajo consiste en intentar demostrar que muchas secuencias filtradas no linealmente tienen una complejidad lineal global muy grande.

Tal y como se menciona en la sección anterior, si un RDRL no singular tiene polinomio de conexión primitivo y estado inicial no nulo, entonces los $2^L - 1$ estados siguientes son todos distintos y no nulos. Dado que para definir unívocamente una función hay que asignar una salida a cada una de las posibles entradas, tal y como se afirma en [152], para cada RDRL de máxima longitud y estado inicial no nulo existe una única correspondencia entre cada una de las posibles secuencias no nulas de longitud 2^L-1 y alguna función f de GF(2^L) en GF(2).

Esto implica que, para cualquiera de las posibles secuencias binarias de longitud 2^L-1 siempre se puede encontrar una función no lineal que aplicada a un RDRL de máxima longitud genere esa secuencia. En particular existe dicha función para todas las secuencias periódicas cuya complejidad lineal global está cercana al valor del periodo.

El problema consiste en encontrar la relación existente entre cada función y la complejidad lineal global de la secuencia generada. El principal objetivo de este trabajo es intentar descubrir dicha relación para una clase amplia de funciones.

En el próximo apartado se ven una serie de condiciones mínimas que debe cumplir un filtrado no lineal para poder ser utilizado como generador de secuencia cifrante.

1.2.1. Principios de Diseño para el Filtrado no Lineal

Los principios de diseño [112] están señalados con P1, P2,....

En primer lugar, tal y como se mostró en la sección anterior, para partir de una buena base es necesario utilizar un RDRL de longitud máxima y estado inicial no nulo. De esta forma se consiguen un periodo grande (C1) y buenas propiedades estadísticas (postulados de Golomb). En particular, si se parte de un RDRL de estas características y longitud L, se obtiene a la salida una secuencia filtrada cuyo periodo es 2^L-1 ó un divisor suyo. Por tanto el primer principio se puede escribir como sigue.

P1. Utilizar un RDRL de longitud máxima y estado inicial no nulo para obtener un gran periodo y buenas características estadísticas.

Tal y como ya se ha mencionado, la secuencia generada por un RDRL cíclico puro con estado inicial 00...001 tiene complejidad lineal global máxima pero a cambio no cumple los postulados de Golomb. En relación con esto se tiene el siguiente resultado.

Lema 1.2.1

El producto de todas las etapas de un RDRL no singular de longitud L, polinomio característico primitivo y estado inicial no nulo siempre produce una secuencia de máxima complejidad lineal global 2^L-1.

Demostración La secuencia generada por un RDRL no singular con polinomio característico primitivo es una PN-secuencia, por lo que en el diagrama de transición de estados el circuito contiene exactamente una vez el estado formado por L unos. Éste es el único estado que produce para el filtrado no lineal de la hipótesis una salida de valor 1, por lo que la secuencia de salida de dicho filtrado es la secuencia formada por $2^L - 2$ ceros y un uno, que ya sabemos tiene complejidad lineal $2^L - 1$.

Según el comentario anterior al lema, se puede concluir la inutilidad de dicho producto como filtrado no lineal por producir una secuencia que, a pesar de tener máxima complejidad lineal global, no tiene las propiedades de aleatoriedad requeridas.

Esto mismo se puede hacer extensivo a todas las transformaciones no lineales que incluyan sólo productos de un orden alto, ya que las secuencias resultantes pueden aproximarse mediante la secuencia nula cometiendo sólo errores en unos pocos bits. Esta predecibilidad queda de manifiesto en su PCL ya que no se aproxima a la recta n/2. Todo este razonamiento conduce a la

recomendación de utilizar transformaciones no lineales que incluyan varios productos de todos los órdenes, y que tengan un orden máximo no muy alto. La primera parte está contenida en el siguiente principio.

P2. Incluir en la función no lineal varios términos de cada orden para conseguir una buena confusión.

Por otro lado, según se verá más adelante, el orden k de la función no lineal juega un papel fundamental en la determinación de la complejidad lineal global de la secuencia producida, hasta el punto de que ésta siempre está acotada superiormente por $\sum_{i=1}^{k}\binom{L}{i}$. Esta propiedad sugiere el uso de un orden k alto. Luego para llegar a un compromiso entre este razonamiento y el anterior al P2 se recomienda utilizar una función de orden de valor aproximado a L/2, lo que queda recogido como sigue.

P3. Escoger una función no lineal de orden k que permita obtener una complejidad lineal global cercana al valor máximo $k \sim L/2$.

Las buenas características estadísticas de la PN-secuencia pueden verse perjudicadas por una mala elección de la función booleana. Para intentar evitarlo, al menos se puede garantizar el cumplimiento del primer postulado de Golomb mediante el uso de un término lineal ya que la función de la forma $x_1 + f_s(x_2, ..., x_n)$ produce una secuencia binaria que contiene aproximadamente la mitad de unos, cuestión que se resume en el siguiente principio.

P4. Incluir en la función no lineal un término lineal para garantizar algunas propiedades estadísticas.

Por último, dado que la clave es secreta resulta bastante apropiado extender la influencia de este secreto no sólo al estado inicial del RDRL sino también a la función no lineal. De esa manera se podría hacer que cada clave estuviera formada por un estado inicial y una función no lineal, tal y como se indica en el siguiente principio.

P5. Hacer que la clave determine el estado inicial del RDRL y algunos términos de la función no lineal.

Estos cinco principios pueden ser complementados en casos particulares con otros principios adicionales. De hecho, al final del próximo capítulo se aportarán nuevas sugerencias que se sumarán a éstos.

No obstante, aunque se sigan todos estos criterios en la elección del filtrado no lineal, la dificultad principal en el manejo de estas transformaciones no lineales reside en su dificultad de análisis. Este hecho se manifiesta claramente en el próximo apartado donde se plantean varias formas diferentes de abordar este problema.

1.2.2. Aproximaciones a las Secuencias Filtradas no Linealmente

Aproximación Matricial

Forma Algebraica Normal La primera aproximación viene dada por la FAN de la función no lineal ya que, como veremos, cualquier secuencia filtrada puede expresarse como combinación lineal de una serie de vectores linealmente independientes que quedan determinados a partir de la FAN del filtrado no lineal.

En particular, dada una PN-secuencia s=$\{s_n\}$, si se denota como $\widehat{s}_{n+t} \in \{0,1\}^{2^L-1}$ al vector formado por el primer periodo del desplazamiento de fase $\{s_{n+t}\}$ y como $\widehat{z}_n \in \{0,1\}^{2^L-1}$ al vector formado por el primer periodo de la secuencia filtrada, según la FAN se tiene que

$$\widehat{z}_n = a_1\widehat{s}_{n+1} + \cdots + a_L\widehat{s}_{n+L} + a_{1,2}\widehat{s}_{n+1}\widehat{s}_{n+2} + \cdots + a_{1,2,\ldots,L}\widehat{s}_{n+1}\cdots\widehat{s}_{n+L},$$

donde los productos entre vectores son productos vectoriales.

Por tanto, dado que cualquier vector $\widehat{z}_n \in \{0,1\}^{2^L-1}$ puede expresarse como la combinación lineal anterior, los 2^L-1 vectores

$$\widehat{s}_{n+1}, ..., \ \widehat{s}_{n+L}, \ \widehat{s}_{n+1}\widehat{s}_{n+2}, ..., \ \widehat{s}_{n+1} \ \widehat{s}_{n+2} \ \cdots \widehat{s}_{n+L}$$

son linealmente independientes tal y como indica el siguiente resultado.

Lema 1.2.2

Si una función no lineal f se aplica sobre las etapas de un RDRL de máxima longitud, entonces las 2^L-1 secuencias correspondientes a los términos de la FAN son linealmente independientes y constituyen una base del espacio vectorial de las secuencias de periodo 2^L-1.

Dado que cada coeficiente de la FAN de f indica la participación o no de cada secuencia de la base anterior, la FAN de f proporciona una descripción natural de la secuencia filtrada.

El problema de esta representación surge cuando hay que obtener el primer periodo de la secuencia $\{z_n\}$ a partir de la expresión matricial $\widehat{z}_n = P^t \cdot A$, donde P es una matriz cuadrada de orden 2^L-1 cuyas filas están formadas respectivamente por $\widehat{s}_{n+1}, ..., \widehat{s}_{n+1}...\widehat{s}_{n+L}$ y A aquí denota el vector de coeficientes de la FAN. Dada la dimensión de la matriz P, incluso para valores de L pequeños, esta aproximación implica un cálculo matricial irrealizable.

Equivalente Lineal Descompuesto Cualquier secuencia de periodo $2^L - 1$ se puede generar mediante un RDRL cíclico puro de longitud $2^L - 1$ tomando simplemente como estado inicial el primer periodo de la secuencia. De este hecho se obtiene una segunda representación [152] de la secuencia filtrada como suma de secuencias tal y como se ve a continuación.

El polinomio de conexión del RDRL cíclico puro de longitud $2^L - 1$ es $x^{2^L-1}+1$, por lo que puede expresarse como producto de todos los polinomios irreducibles $C_i(x)$ de grados divisores de L (salvo el polinomio x): $x^{2^L-1}+1 = \prod_i C_i(x)$.

Si esta propiedad se aplica a la identidad fundamental se tiene que $S(x) = \frac{P(x)}{x^{2^L-1}+1} = \sum_i \frac{P_i(x)}{C_i(x)}$, siendo el grado de $P_i(x)$ menor que el de $C_i(x)$. Por tanto se ha logrado expresar la D-transformada de la secuencia filtrada como suma de D-transformadas de secuencias generadas por varios RDRLs de polinomios de conexión $C_i(x)$ cuya suma de longitudes coincide con 2^L-1.

De ahí que las 2^L-1 secuencias correspondientes a las etapas de cada uno de estos RDRLs, tomando como estados iniciales vectores con un único uno, puedan considerarse como una segunda base para el espacio vectorial de las secuencias de periodo 2^L-1.

Esta segunda representación implica la descomposición del RDRL de longitud L filtrado no linealmente en un generador formado por varios RDRLs cuyos polinomios de conexión son polinomios irreducibles de grados divisores de L. A este generador se le conoce como **equivalente lineal descompuesto** y un ejemplo de su estructura se muestra en la figura 6.

Cada polinomio de estado inicial $P_i(x)$ nulo implica un estado inicial nulo para el RDRL correspondiente al polinomio de conexión $C_i(x)$, es decir, implica la ausencia de dicho componente en el equivalente.

La desventaja de este método viene dada, al igual que en el primer caso, por la necesidad de manejo de matrices de orden 2^L-1. Los estados iniciales de los RDRLs componentes del equivalente han de obtenerse mediante la expresión matricial $\widehat{z}_n = D^t \cdot R$, donde D es una matriz cuadrada de orden 2^L-1 cuyas filas están formadas por las 2^L-1 subsecuencias de longitud 2^L-1 (correspondientes a las etapas de cada uno de los componentes y generadas a partir de estados iniciales con un único 1), y R es un vector de dimensión 2^L-1 formado por los estados iniciales de todas las componentes del equivalente.

El equivalente lineal descompuesto planteado es el antecedente de un nuevo equivalente que se presentará en el último capítulo.

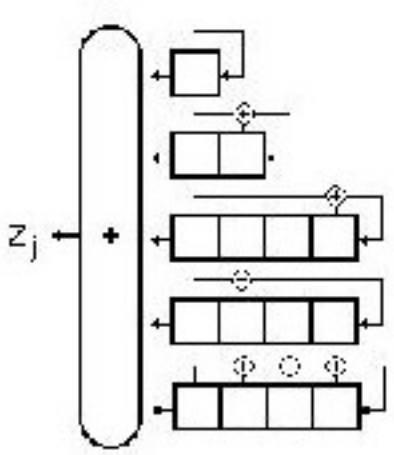


Figura 1.6: Equivalente Lineal Descompuesto

Estas dos aproximaciones proporcionan las herramientas necesarias para generar una función no lineal que produzca una secuencia de complejidad lineal global garantizada, tal y como se ve en el siguiente apartado.

Cómo Garantizar la Complejidad Lineal Dado un valor de complejidad lineal, para un diseñador de generadores de secuencia cifrante sería muy conveniente poder encontrar una función no lineal que produzca una secuencia filtrada con ese valor de complejidad lineal.

Combinando las dos expresiones matriciales dadas en los apartados anteriores se obtiene $A = (P^t)^{-1}D^tR = (D \cdot P^{-1})^tR$. Según esta expresión, siempre pueden escogerse los estados iniciales para cada una de las componentes del equivalente lineal descompuesto (R) y deducir mediante A qué función no lineal concreta produce la secuencia filtrada. Esto implica la elección previa de la complejidad lineal global del generador mediante la adjudicación de estados iniciales no nulos a aquellas componentes cuya suma de longitudes proporciona la complejidad lineal deseada.

De nuevo, lo que inutiliza este método es la enorme dimensión de las matrices que hay que manejar (2^L-1).

En el siguiente apartado se plantea un acercamiento muy distinto al problema del análisis del filtrado no lineal, mediante transformadas discretas de

Fourier.

Aproximación Mediante el Análisis de Fourier Las transformadas de Fourier en un cuerpo de Galois juegan un papel muy importante en el estudio y procesamiento de las señales. En este apartado se extiende su aplicación al análisis del filtrado no lineal.

Se dan aquí sin demostración algunos resultados fundamentales obtenidos mediante esta aproximación por varios autores, Blahut [7], Massey y Serconek [116] y Paterson [143].

Dada una secuencia $\{s_n\}$, se define la **transformada discreta de Fourier** (**TDF**) de $s^p = s_0, ..., s_{p-1}$ como el vector $\mathbf{S}^p = (S_0, ..., S_{p-1})$ donde $S_i = \sum_{k=0}^{p-1} s_k w^{ik}$ con i=0,...,p-1 y w un elemento primitivo de orden p.

La secuencia puede obtenerse mediante la inversa de la TDF definida como $s_j = \frac{1}{p} \sum_{i=0}^{p-1} S_i w^{-ij}$ con j=0,...,p-1.

Las componentes de las p-uplas s^p y S^p pueden utilizarse respectivamente como coeficientes de dos polinomios s(x) y S(x) de grados menores o iguales que p-1, $\mathbf{s}(\mathbf{x}) = s_0 + s_1 x + \cdots + s_{p-1} x^{p-1}$ y $\mathbf{S}(\mathbf{x}) = S_0 + S_1 x + \cdots + S_{p-1} x^{p-1}$. Según estas representaciones las definiciones de TDF e inversa de TDF se pueden escribir como $S_i = s(w^i)$ y $s_j = \frac{1}{p} S(w^{-j})$ con i,j$\in \{0,1,...,p-1\}$.

El siguiente resultado debido a Blahut [7] confirma la utilidad de la TDF para el cálculo de la complejidad lineal global de una secuencia periódica.

Teorema 1.2.1

Si $\{s_n\}$ tiene periodo p y S^p es la TDF de s^p, entonces $\Lambda(\{s_n\}) = W_H(S^p)$.

La TDF constituye una herramienta útil para el análisis de las transformaciones no lineales de RDRLs, según se observa en las siguientes aplicaciones a los casos de un filtrado no lineal de orden 2 [116] y un producto de fases equidistantes [143].

En 1994 [116] Massey y Serconek demuestran el siguiente resultado utilizando la herramienta propuesta.

Proposición 1.2.1

Dadas una secuencia $\{s_n\}$ de polinomio minimal c(x)$\in GF(2)[x]$ primitivo y de grado L primo, y la secuencia $\{z_n\}$ definida mediante $z_n = s_n s_{n+\delta}$, entonces la complejidad lineal de la secuencia resultante viene dada por $\Lambda(\{z_n\}) = L + \binom{L}{2}$.

Este mismo resultado será demostrado en el último capítulo sin necesidad de utilizar transformadas discretas de Fourier.

También en 1994 [143] Paterson amplía este resultado a los casos de un producto de fases equidistantes y de una suma de productos de fases equidistantes. En el primer caso demuestra el siguiente resultado.

Teorema 1.2.2

Dadas una secuencia $\{s_n\}$ *de polinomio minimal c(x)*$\in GF(2)[x]$ *primitivo y de grado L primo, y la secuencia* $\{z_n\}$ *definida mediante* $z_n = s_n\ s_{n+\delta}\ s_{n+2\delta} \cdots s_{n+(k-1)\delta}$, *si t es el menor entero positivo tal que* $2^L - 1 | \delta(2^t - 1)$, *entonces* $\Lambda(\{z_n\}) \left(\begin{array}{c} t \\ k \end{array} \right) (\frac{L}{t})^k$.

Al generalizar el resultado anterior se obtiene el siguiente.

Teorema 1.2.3

Dadas una secuencia $\{s_n\}$ *de polinomio minimal c(x)*$\in GF(2)[x]$ *primitivo y de grado L primo, y la secuencia* $\{z_n\}$ *definida mediante* $z_n = \sum_{i=0}^{N-1} b_i s_{n+i}\ s_{n+i+\delta} \cdots s_{n+i+(k-1)\delta}$, *si t es el menor entero positivo tal que* $2^L - 1 | \delta(2^t - 1)$, *entonces* $\Lambda(\{z_n\}) \geq \left(\begin{array}{c} t \\ k \end{array} \right) (\frac{L}{t})^k - (N-1)$.

Obsérvese que la mejor cota que se puede obtener según este último resultado viene dada para el caso t=L. La cota alcanzada para este caso también se demostrará en un apartado posterior sin necesidad de utilizar transformadas discretas de Fourier.

En la próxima sección se verán por orden cronológico las diferentes aproximaciones al estudio de la complejidad lineal del filtrado no lineal llevadas a cabo por algunos de los autores más representativos en este campo.

1.3. Recorrido Bibliográfico

1.3.1. Estudio de Groth sobre las Funciones de Segundo Orden

En 1971 Groth [71] estudia el caso de sumas de productos de segundo orden de etapas de un RDRL con polinomio característico primitivo. Presenta la complejidad lineal como un parámetro controlable y directamente

proporcional al orden de la función y no admite la existencia de posibles irregularidades.

Para realizar su análisis impone las siguientes restricciones. El RDRL debe ser de longitud máxima y la función no lineal una suma de productos de orden dos tales que las fases consideradas no sobrepasen la longitud del RDRL. Además como última restricción considera en todo caso el estado inicial $s_0 = s_1 = \cdots = s_{L-2} = 0,\ s_{L-1} = 1$.

Bajo estas condiciones la función generatriz de la secuencia filtrada $\{b_n\}$ se puede escribir como $B(x) = \sum_{n=0}^{\infty} b_n x^n = \sum_{i=1}^{L-1} B^i(x)$ donde $B^i(x) = \sum_{n=0}^{\infty} b_n^i x^n$ y $b_n^i = \sum_{l=1}^{L-1} e_{l,l+i}^i s_{L+n-l} s_{L+n-(l+i)}$.

La expresión de $B^1(x)$ se simplifica de la siguiente forma

$$B^1(x) = \sum_{l=1}^{L-1} e_{l,l+1}^1 x^l([s_{L-l}s_{L-(l+1)}x^{-l} + \cdots + s_{L-1}s_{L-2}x^{-1}] + \sum_{n=0}^{\infty} s_{L+n}s_{L+n-1}x^n)$$

Debido a la última restricción considerada la expresión entre corchetes se anula. Además, utilizando la relación de recurrencia lineal y esa misma restricción se tiene que $B^i(x) = \sum_{l=1}^{L-i} e_{l,l+i}^i x^l S_i$, donde $S_i = \sum_{k=1}^{L} c_k x^i S_{k-i}$.

De todo esto se obtiene la siguiente expresión de la función generatriz de la secuencia $B(x) = \sum_{i=1}^{L-1} \sum_{l=1}^{L-i} e_{l,l+i}^i x^l S_i$. De la definición recursiva de S_i resultan también expresiones con subíndices negativos, de manera que utilizando de nuevo la última restricción se obtiene que $S_{-i} = x^{-i} S_i$. Esta igualdad se aplica al siguiente sistema de L-1 ecuaciones con L-1 incógnitas,

$$\begin{pmatrix} 1 + C_2 x & C_3 x & \cdot & C_{L-1} x & x \\ C_1 x + C_3 x^2 & 1 + C_4 x^2 & \cdot & x^2 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ C_{L-3} x^{L-3} + C_{L-1} x^{L-2} & C_{L-4} x^{L-4} + x^{L-2} & \cdot & 1 & 0 \\ C_{L-2} x^{L-2} + x^{L-1} & C_{L-3} x^{L-3} & \cdot & C_1 x & 1 \end{pmatrix} \begin{pmatrix} S_1 \\ S_2 \\ \cdot \\ S_{L-2} \\ S_{L-1} \end{pmatrix} =$$

$$S_0 \begin{pmatrix} C_1 x \\ C_2 x^2 \\ \cdot \\ C_{L-2} x^{L-2} \\ C_{L-1} x^{L-1} \end{pmatrix}$$

Debido a la presencia de L-1 potencias de x en la diagonal menor se tiene que, según la regla de Cramer, en el denominador de las soluciones aparece un polinomio de grado L(L-1)/2. El determinante del numerador corresponde a un polinomio del mismo grado. Al desarrollar este determinante por los elementos de la matriz término independiente siempre se puede sacar como factor común una potencia de x. Denotando $i_0 = \min_{c_i \neq 0} i,\ p_{i_0} = i_0$ y $\forall i \neq i_0 : p_i \geq i_0$, se tiene que

$$S_i = \frac{x^{p_i}\left(a_0 + a_1 x + \cdots + a_{\frac{L(L-1)}{2}-p_i} x^{\frac{L(L-1)}{2}-p_i}\right)}{a_0 + a_1 x + \cdots + a_{\frac{L(L-1)}{2}} x^{\frac{L(L-1)}{2}}} S_0.$$

De la expresión de S_0 que resulta después de aplicar la última restricción se llega a que

$$S_i = \frac{x^{p_i}\left(a_0 + a_1 x + \cdots + a_{\frac{L(L-1)}{2}-p_i} x^{\frac{L(L-1)}{2}-p_i}\right)}{a_0 + a_1 x + \cdots + a_{\frac{L(L+1)}{2}} x^{\frac{L(L+1)}{2}}}$$

En S_i numerador y denominador son primos entre sí, y B(x) no es sino una combinación de los S_i multiplicados por x^l, que a su vez es primo también con el denominador de S_i. De todo esto Groth deduce que el numerador y el denominador de B(x) tienen que ser primos entre sí. De esta forma garantiza una complejidad lineal de valor el grado del polinomio del denominador que es L(L+1)/2. Ahora bien, a pesar de que el resto del análisis realizado es correcto, esta última conclusión es errónea y por tanto ese valor de complejidad lineal no está en modo alguno garantizado.

Rectificaciones y Ampliaciones Como demostración del último comentario damos un contraejemplo.

Se considera el RDRL de polinomio de realimentación $x^6 + x^5 + 1$ y la función no lineal $s_n s_{n+7} + s_{n+2} s_{n+5}$. Siguiendo los pasos dados por Groth se llega a que $S_1 = x^6(1 + x^3 + x^6 + x^5 + x^9)S_0,\ S_4 = x^{10}(1 + x^3 + x^5)S_0,\ S_0 = \frac{1}{1+x^6+x^5}$, y para esta función particular se tiene que

$$B(x) = \frac{x^6(1 + x^3 + x^8 + x^{10} + x^{12} + x^{14} + x^{15})}{(1 + x + x^3)(1 + x^5 + x^6)(1 + x + x^4 + x^5 + x^6)(1 + x^2 + x^4 + x^5 + x^6)}$$

En este ejemplo, numerador y denominador no son primos entre sí, al contrario de lo que aseguró Groth. Luego al simplificar queda de la forma

$$B(x) = \frac{x^6(1+x^3+x^5+x^6+x^9)}{(1+x+x^3)(1+x+x^4+x^5+x^6)(1+x^2+x^4+x^5+x^6)}$$

Por tanto, la complejidad lineal no es máxima ya que como en este cociente numerador y denominador sí son primos entre sí se puede deducir que la complejidad lineal global del generador es 15. De hecho se observa con este ejemplo que el error proviene de asegurar que el numerador y el denominador de B(x) son primos entre sí siempre.

Por otro lado, lo que sí podemos asegurar, dada la forma de B(x), es que siempre se puede sacar $x^{i_0+\min l}$ como factor común en el numerador, luego siempre queda asegurada una complejidad lineal de valor $L + i_0 - (\max l - \min l)$ (denotando $maxl$ y $minl$ respectivamente al mayor y menor valor l tal que $e_{l,l+i} \neq 0$ para algún i$\in \{1, 2, ..., L-1\}$).

El mayor valor que se puede obtener para esta expresión es 2L, que resulta de tomar aquellos productos cuyos términos menores sean muy cercanos ($\max l \approx \min l$) y considerar aquellos RDRLs cuyos polinomios característicos tengan sólo potencias altas de x ($i_0 \approx L$).

Por otro lado, si se consideran las sumas de productos de orden 2 de fases equidistantes $s_n s_{n+\delta} + s_{n+\delta_1} s_{n+\delta_1+\delta} + \cdots + s_{n+\delta_k} s_{n+\delta_k+\delta}$, se tiene que el numerador de B(x) queda de la forma $S_\delta x^{L-(\delta_k+\delta)}[x^{\delta_k} + x^{\delta_k-\delta_1} + x^{\delta_k-\delta_2} + \cdots + 1]$. Si el polinomio entre corchetes es primo con el denominador de B(x), entonces se obtiene una complejidad lineal máxima.

Por último, el análisis planteado se puede generalizar al caso de funciones no lineales que incluyan también términos lineales. Eso se consigue mediante la simple adición de un término $B^0(x) = \sum_{n=0}^{\infty} b_n^0 x^n$, $b_n^0 = \sum_{l=1}^{L} e_{l,l}^0 a_{n-l} a_{n-l}$, de manera que B(x)$=\sum_{i=0}^{L-1} B^i(x)$. De esta forma amplía el análisis a una clase mayor de funciones.

1.3.2. Caracterización de las Secuencias Según Key

En 1976 Key [85] investiga tanto el filtrado no lineal de un RDRL como el combinador no lineal de varios RDRLs. Su aportación fundamental consiste en que establece una relación entre la complejidad lineal de la secuencia generada y las raíces de su polinomio minimal.

Al igual que Groth, Key comete un error en sus conclusiones. Por un lado asegura que un producto de segundo orden siempre tiene complejidad lineal máxima y por otro, al generalizar esta propiedad a un producto de varios términos, deduce que 'sin dificultad teórica ni limitación práctica, la complejidad lineal puede ser escogida de forma determinística'. No obstante hay que mencionar que él, al contrario que Groth, sí admite que pueden presentarse excepciones.

Key parte del teorema 1.3 según el cual la secuencia producida por un RDRL se puede representar en función de las raíces de su polinomio característico. En su trabajo demuestra que las operaciones no lineales inyectan en dicha representación raíces del polinomio minimal de la secuencia resultante. Además concluye que el número de tales raíces que no desaparecen de la expresión de la secuencia resultante sirve como medida de su complejidad lineal global ya que coincide con la longitud del equivalente lineal.

Ejemplo 1.3.1

Dado un RDRL de longitud L=4 y una función de la forma $s_n s_{n+1}$*, se tiene que:*

$s_n = A\alpha^n + A^2\alpha^{2n} + A^4\alpha^{4n} + A^8\alpha^{8n}$ *y* $s_{n+1} = A\alpha^{n+1} + A^2\alpha^{2n+2} + A^4\alpha^{4n+4} + A^8\alpha^{8n+8}$*. Luego la función producto se puede expresar como* $s_n s_{n+1} \equiv A\alpha^8\alpha^n + A^2\alpha\alpha^{2n} + A^4\alpha^2\alpha^{4n} + A^8\alpha^4\alpha^{8n} + A^3(\alpha^2 + \alpha)\alpha^{3n} + A^6(\alpha^4 + \alpha^2)\alpha^{6n} + A^{12}(\alpha^8 + \alpha^4)\alpha^{12n} + A^9(\alpha^8 + \alpha)\alpha^{9n} + A^5(\alpha^4 + \alpha)\alpha^{5n} + A^{10}(\alpha^8 + \alpha^2)\alpha^{10n}$*.*

En la expresión anterior, además del conjunto de raíces conjugadas $\{\alpha, \alpha^2, \alpha^4, \alpha^8\}$ *presentes en las expresiones de* s_n *y* s_{n+1}*, aparecen dos nuevos conjuntos de raíces conjugadas* $\{\alpha^3, \alpha^6, \alpha^{12}, \alpha^9\}$ *y* $\{\alpha^5, \alpha^{10}\}$*. De las diez potencias de* α *anteriores, aquéllas cuyos coeficientes no se anulan son las raíces del polinomio minimal de la secuencia filtrada. Por tanto, el número de estas raíces determina el grado del polinomio característico del equivalente lineal, que es la complejidad lineal.*

A la vista del ejemplo anterior se observa que:

Los únicos cosets que pueden aparecer en la expresión de una secuencia filtrada son aquéllos que tienen peso menor o igual que el orden de la función.

Dada la forma de los coeficientes que acompañan a las raíces se deduce que dichas raíces se presentan siempre en conjuntos de conjugadas.

Dados un RDRL de longitud L y polinomio característico c(x) primitivo sobre GF(2) y $\alpha \in GF(2^L)$ una raíz de c(x), entonces la secuencia producida viene dada por $s_n = \sum_{i=0}^{L-1} A_i(\alpha^{2^i})^n,\ A_i = A^{2^i} \in GF(2^L)$ según lo visto en el apartado 1.1.2.

Key considera la secuencia producto $s_n s_n^* = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} A_i A_j^*(\alpha^{2^i+2^j})^n$. Utilizando la notación referente a cosets, se tiene que el coset $2^i + 2^j$ tiene diferente peso según sea i=j ó i$\neq$ j. El caso i=j define el coset de peso uno y cardinal L, mientras que cuando i$\neq$j se tienen los cosets de peso dos cuya suma de cardinales es L(L-1)/2. Por tanto hay L(L+1)/2 potencias distintas de α presentes en la secuencia producto y, si ninguno de los coeficientes se anula, el generador tiene complejidad lineal global de valor L(L+1)/2. Key en este punto trata de demostrar que esos coeficientes nunca se anulan. Para ello parte de que la secuencia $\{s_n^*\}$ es simplemente un desplazamiento de fase de la secuencia $\{s_n\}$, $s_n^* = s_{n+\delta}\ \ 0 < \delta < L$ luego $A_j^* = A_j(\alpha^{2^j})^\delta$. De esa manera tiene que la expresión anterior queda de la forma $s_n s_{n+\delta} = \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} A_i A_j(\alpha^{2^j})^\delta(\alpha^{2^i+2^j})^n$.

Si i=j el coeficiente $A_i A_i^* = A_i^2\alpha^{2^i\delta}$ nunca se anula. Si i$\neq$j, debido a la simetría de los exponentes 2^i+2^j, se concluye que los coeficientes $A_i A_j^*$ y $A_j A_i^*$ acompañan siempre a la misma potencia. Por consiguiente, esta potencia desaparece siempre que la suma de ambos coeficientes se anula. Es decir, considerando el líder del coset (i=0), esta potencia desaparece siempre que $A_0 A_j(\alpha^{2^j})^\delta + A_j A_0(\alpha^{2^0})^\delta = 0$ en GF(2^L). Ahora bien, según Key esto no puede ocurrir porque $2^j\delta \not\equiv \delta(mod(2^L - 1))$.

El análisis llevado a cabo por Key es correcto pero la última conclusión sólo se verifica para δ tal que $0 < \delta <$L. En el caso en el que δ pertenece al intervalo [L,2^L-2] puede no cumplirse la relación anterior. Como demostración de esto vemos un contraejemplo.

En el caso de un RDRL de longitud L=6 y una secuencia producto $s_n s_{n+9}$, el coeficiente correspondiente al coset $2^0 + 2^3$ se anula $A_0 A_3(\alpha^{2^3 9} + \alpha^9) \equiv 0$ en GF(2^6) puesto que 72$\equiv$9 (mod (2^6-1)).

El trabajo de Key planteado en este apartado constituye la base del equivalente lineal propuesto en el último capítulo.

1.3.3. Cotas de Kumar y Scholtz para Algunas Secuencias

En 1983 Kumar y Scholtz [90] fijan una serie de condiciones restrictivas sobre la función no lineal y la longitud del RDRL (L= $\dot{4}$, m=L/2, k $\leq$m) para obtener unas cotas superiores e inferiores a la complejidad lineal. Dichas cotas resultan válidas para una familia de secuencias concreta [140] que verifican las condiciones mencionadas. A continuación se ven sin demostración los resultados más destacables.

En primer lugar definen los siguientes conjuntos de líderes de cosets

$$Q_r = \{1 \leq Q \leq 2^L - 1 / W_H(Q) \leq r,\ Q \leq 2^j Q (mod\ 2^L - 1),\ j = 1, ..., L-1\},$$

$$H_r = \{Q \in Q_r / \alpha^Q \in GF(2^m),\ Q \leq 2^i Q (mod\ 2^L - 1),\ i = 1, ..., L-1\},$$

$$E_r = \{Q \in Q_r / W_H(Q) = r,\ Q = \textstyle\sum_{i=1}^{r} 2^{v_i},\ v_{i_0} - v_{j_0} = m,\ 1 \leq i_0 < j_0 \leq r,\ Q \leq 2^i Q (mod\ 2^L - 1),\ i = 1, ..., L-1\}.$$

El primer conjunto está formado por los líderes de los cosets de peso menor o igual que r. Los otros dos son subconjuntos suyos $H_r \subset Q_r,\ E_r \subset Q_r$.

Estos autores utilizan la expresión de la secuencia generada por el RDRL de polinomio característico irreducible como una función traza demostrada en el teorema 1.3 y la noción del equivalente lineal descompuesto definido en el apartado 1.2.2. A partir de ambos conceptos obtienen la expresión de las secuencias filtradas como suma de funciones traza. Denotan como $A_Q^{2^i}$ los coeficientes que acompañan al conjunto de raíces conjugadas de α^Q.

Obsérvese que en la notación que ellos utilizan las fases $s_{n+\delta_i}$ sobre las que se aplica la función no lineal f están indicadas mediante variables $g_i = \alpha^{\delta_i}$.

El primer resultado del trabajo garantiza que en el caso de un filtrado no lineal en que las fases consideradas corresponden a $g_i \in GF(2^m)$ (subcuerpo de GF(2^L)), los cosets E tales que $\alpha^E \in GF(2^m)$ no contribuyen a la complejidad lineal del generador.

Lema 1.3.1

Si f es una función booleana de orden k en m variables, $g_1, g_2, ..., g_m$ (base de $GF(2^m)$) y $s_{n+\delta_i} = Tr_1^L(g_i \alpha^n) = \sum_{j=0}^{L-1} (g_i \alpha^n)^{2^j}$, i=1,...,m, donde α es un elemento primitivo de GF(2^L), entonces se tiene que en

$f(s_{n+\delta_i}) = \sum_{Q \in Q_k} Tr_1^{p_Q}(A_Q \alpha^{Q_n})$ $A_Q = 0$ $\forall Q \in H_k \subset Q_k$.

El siguiente resultado afirma que, si la secuencia filtrada resulta de una función booleana aplicada sobre las m fases correspondientes a $g_1, g_2, ..., g_m \in GF(2^m)$, entonces los cosets Q $\in E_k$ (cosets de peso k tales que en su expresión binaria tengan algún par de unos separados m lugares) no contribuyen a la complejidad lineal del generador.

Lema 1.3.2

Si f es una función booleana de orden k en m variables $g_1, g_2, ..., g_m$ *(base de* $GF(2^m)$*) y* $s_{n+\delta_i} = Tr_1^L(g_i\alpha^n) = \sum_{j=0}^{L-1}(g_i\alpha^n)^{2^j}$*, i=1,...,m siendo* α *un elemento primitivo de GF(*2^L*), entonces en la expresión* $f(s_{n+\delta_i}) = \sum_{Q\in Q_k} Tr_1^{p_Q}(A_Q\alpha^{Qn})$ *se tiene que* $A_Q = 0,\ \forall Q \in E_k$.

Para aclarar estos resultados presentamos a continuación un ejemplo.

Ejemplo 1.3.2

Con L=8, k=3, m=4, GF(2^8*)* $= \{0, 1, \alpha, \alpha^2, ..., \alpha^{254}\}$, $GF(2^4) = \{0, 1, \alpha^{85}$, $\alpha^{170}, \alpha^{17}, \alpha^{34}, \alpha^{68}, \alpha^{136}, \alpha^{51}, \alpha^{102}, \alpha^{204}, \alpha^{153}, \alpha^{119}, \alpha^{238}, \alpha^{221}, \alpha^{187}$*. Si se toman las tres fases de la función según los exponentes de elementos no nulos de GF(*2^4*), por ejemplo* $g_1 = \alpha^0$, $g_2 = \alpha^{17}$ *y* $g_3 = \alpha^{85}$*, es decir, f:*$s_n s_{n+17} s_{n+85}$*, entonces se tiene que por el lema 1.1, el coeficiente* A_{17}*=0 y por el lema 1.2 los coeficientes* $A_{19} = A_{21} = A_{25}$*=0, luego los cosets 17, 19, 21 y 25 no contribuyen a la complejidad lineal de la secuencia generada.*

El resultado más relevante del trabajo de Kumar y Scholtz garantiza que los cosets de peso k contribuyen a la complejidad lineal cuando la función es un producto de fases escogidas de una forma muy específica.

Teorema 1.3.1

Si f es una función no lineal de orden k de la forma $f(s_{n+\delta_1}, ..., s_{n+\delta_L}) = \prod_{i=1}^{k} s_{n+i}$*,* α *es un elemento primitivo de GF(*2^L*), el conjunto* $\{\beta^{2^i}\}_{i=0,...,L-1}$ *es una base normal de GF(*2^L*) sobre GF(2) y la secuencia* $s_{n+\delta_i} = Tr_1^L(\beta^{2^{i-1}}\alpha^n)$*,* $i = 1, 2, ..., L$*, entonces en la expresión de la función* $f(s_{n+\delta_i}) = \sum_{Q\in Q_k} Tr_1^{p_Q}(A_Q\alpha^{Qn})$*, se tiene que* $A_Q \neq 0,\ \forall Q : W_H(Q) = k$.

En las demostraciones realizadas por Kumar y Scholtz de los lemas 1.4 y 1.5 en ningún momento utilizan la condición de base de GF(2^m) sino la

simple pertenencia a dicho cuerpo. Por tanto se puede relajar dicha hipótesis en ambos enunciados.

Después de dar estos resultados generales, en su trabajo Kumar y Scholtz los aplican a unas secuencias concretas, conocidas como secuencias bent, que parecen hechas a medida ya que verifican todas las condiciones exigidas en las hipótesis. Para estas secuencias obtienen una cota inferior de la complejidad lineal ligeramente superior a $\binom{L/2}{L/4} 2^{L/4}$.

En relación con el trabajo realizado por Kumar y Scholtz remitimos al lector al último capítulo donde se presenta una generalización de su teoría.

1.3.4. Test de Presencia de Raíces de Rueppel

El trabajo realizado por Rueppel en 1986 [152] tiene como punto de partida el análisis llevado a cabo por Key. Rueppel parte de una función no lineal f de orden k aplicada sobre las k etapas $s_{n+t_0}, s_{n+t_1}, ..., s_{n+t_{k-1}}$ de un RDRL de máxima longitud. En su trabajo Rueppel da explícitamente la cota superior de la complejidad lineal de las secuencias filtradas $\sum_{i=1}^{k} \binom{L}{i}$, que se encontraba implícita en el trabajo de Key. A partir de este punto en su trabajo deja bien claro que, dado que la línea lógica a seguir en criptografía es intentar conseguir cifrados lo más seguros posibles, esto se traduce en el caso del cifrado en flujo en intentar generar secuencias con una complejidad lineal mínima garantizada. Con este objetivo presenta el siguiente resultado conocido como **'test de presencia de raíces'**, que permite, mediante el cálculo de una serie de determinantes en un cuerpo finito, obtener una cota inferior a la complejidad lineal de una secuencia producto. Para conseguirla hay que aplicar el siguiente test a cada uno de los conjuntos de raíces conjugadas correspondientes a cosets de peso k determinando así cuáles contribuyen a la complejidad lineal.

Teorema 1.3.2

Dadas $\{s_{n+t_0}\}, \{s_{n+t_1}\}, ..., \{s_{n+t_{k-1}}\}$ *k fases distintas de la secuencia* $s=\{s_n\}$ *cuyo polinomio minimal c(x) es primitivo y de grado L,* $\alpha \in GF(2^L)$ *una raíz de c(x), y* $z=\{z_n\}$ *la secuencia producto de las k fases distintas* $z_n = s_{n+t_0} s_{n+t_1} \cdots s_{n+t_{k-1}} = \prod_{i=0}^{k-1} s_{n+t_i}$, *entonces* α^E *tal que* $W_H(E)=k$ *es una raíz del polinomio minimal de z si y sólo si el siguiente determinante no se anula*

en GF(2^L)

$$A_E = \begin{vmatrix} \alpha^{t_0 2^{e_0}} & \alpha^{t_1 2^{e_0}} & \cdot & \cdot & \alpha^{t_{k-1} 2^{e_0}} \\ \alpha^{t_0 2^{e_1}} & \alpha^{t_1 2^{e_1}} & \cdot & \cdot & \alpha^{t_{k-1} 2^{e_1}} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \alpha^{t_0 2^{e_{k-1}}} & \alpha^{t_1 2^{e_{k-1}}} & \cdot & \cdot & \alpha^{t_{k-1} 2^{e_{k-1}}} \end{vmatrix} \neq 0,$$

donde $E=2^{e_0} + 2^{e_1} + \cdots + 2^{e_{k-1}}$ *y* $0 \leq e_0 < e_1 < \cdots < e_{k-1} < L$.

Demostración Se supone sin pérdida de generalidad que el estado inicial de la secuencia s es tal que $s_j = Tr_1^L(\alpha^j)$. Entonces por la propiedad $s_{t+j} = Tr_1^L(\alpha^t \alpha^j)$, se tiene que $z_j = \prod_{i=1}^{k} Tr_1^L(\alpha^{t_i}\alpha^j) = \prod_{i=1}^{k}(\alpha^{t_i}\alpha^j + \alpha^{2t_i}\alpha^{2j} + \cdots + \alpha^{2^{L-1}t_i}\alpha^{2^{L-1}j})$. En el anillo de enteros módulo 2^L-1 cualquier elemento j con $W_H(j) \leq k$ se puede describir como suma de k potencias de dos, pero éstas sólo son distintas cuando W_H(j)=k. La expresión de z_j puede descomponerse en suma de dos términos $z_j = y_j + \sum_{\{E:W_H(E)=k\}} A_E \alpha^{Ej}$ donde y_j denota una suma de raíces α^E tales que W_H(E)¡k. Para obtener un exponente E de peso k a partir de la suma de k potencias de 2, éstas deben ser todas dife/-rentes, es decir, $E= 2^{e_0} + 2^{e_1} + \cdots + 2^{e_{k-1}}$ *y* $0 \leq e_0 < e_1 < \cdots < e_{k-1} < L$. Luego el último sumatorio contiene $\binom{L}{k}$ sumandos. Un mismo exponente E se puede obtener de k! formas diferentes, luego el coeficiente de α^{Ej} resulta ser $A_E = \sum_{m \in P_E} (\alpha^{t_1})^{2^{m_1}} (\alpha^{t_2})^{2^{m_2}} \cdots (\alpha^{t_k})^{2^{m_k}}$, donde P_E es el conjunto de todas las permutaciones de $\{e_0, e_1, ..., e_{k-1}\}$. Teniendo en cuenta que suma y diferencia coinciden en este cuerpo, se deduce que A_E coincide con el determinante de orden k definido en la tesis del teorema. Se concluye por tanto que el determinante A_E es distinto de cero si y sólo si la raíz α^E contribuye a la complejidad lineal de z.

Observación 1.3.1

Este resultado permite determinar cuáles de los cosets de peso k contribuye a la complejidad lineal global. Sin embargo no dice nada acerca de los cosets de peso menor que k.

Se dice que hay **degeneración** cuando alguna de las $\sum_{i=1}^{k}\binom{l}{i}$ potencias de α no está presente en la expresión de z_n, es decir, cuando la complejidad

lineal de la secuencia filtrada no es máxima. Decimos que el coset E de peso k es un **coset degenerado** cuando el determinante A_E correspondiente se anula, y en consecuencia, que es no degenerado cuando no se anula.

El teorema anterior no sólo sirve para comprobar la no degeneración de los cosets de peso k, sino que además permite encontrar clases de funciones no lineales para las que el determinante A_E nunca se anula. De esta forma se logra una cota inferior para la complejidad lineal de determinadas secuencias filtradas.

Corolario 1.3.1

Si $\{s_{n+t}\}, \{s_{n+t+\delta}\}, ..., \{s_{n+t+(k-1)\delta}\}$, $1 \leq k \leq L$ *son fases equidistantes de la misma PN-secuencia* $s=\{s_n\}$ *y* $mcd(\delta, 2^L-1)=1$, *entonces para la función producto de las k fases se tiene que los* $\binom{L}{k}$ *cosets de peso k son todos no degenerados. Por tanto la complejidad lineal de la secuencia producto z está acotada inferiormente mediante* $\Lambda(z) \geq \binom{L}{k}$.

Demostración Se supone sin pérdida de generalidad que t=0. El determinante A_E es siempre un determinante de Vandermonde, por lo que se anula si y sólo si al menos uno de los factores $(\alpha^{\delta 2^{e_n}} - \alpha^{\delta 2^{e_j}})$ (n≠j) se anula. Es decir, si y sólo si los exponentes de α de al menos uno de estos factores coinciden en módulo 2^L-1. Esto último no es posibler ya que $2^{e_n} \neq 2^{e_j} \bmod (2^L-1)$, $\forall n \neq j$ y como $\text{mcd}(\delta, 2^L-1) = 1$, la multiplicación por δ produce una permutación de todos los elementos del anillo de enteros módulo 2^L-1. De ahí se deduce que todos los cosets de peso k contribuyen a la complejidad lineal de z.

El próximo resultado es una generalización de éste a una función combinación lineal de productos de fases equidistantes.

Corolario 1.3.2

Sean $s_{n+t}s_{n+t+\delta}\cdots s_{n+t+(k-1)\delta}$ *un producto de k (<L) fases equidistantes de una PN-secuencia s, con* $mcd(\delta, 2^L-1)=1$ *y z una secuencia producida por una combinación lineal no nula de N productos consecutivos del tipo anterior* $z_n = \sum_{i=0}^{N-1} b_i \; s_{n+i}s_{n+i+\delta}\cdots s_{n+i+(k-1)\delta}$, *entonces la complejidad lineal de z está acotada inferiormente según* $\Lambda(z) \geq \binom{L}{k} - (N-1)$.

Observación 1.3.2

En el caso N≤L y L primo, ninguna de las raíces del polinomio minimal de la secuencia filtrada puede desaparecer de la expresión de z porque ninguna puede ser raíz de un polinomio de grado menor que L. De ahí que la cota $\binom{L}{k}$ *se mantenga en este caso particular.*

Al comparar la cota obtenida por Rueppel con la de Kumar y Scholtz se observa que, por ejemplo para L=24 la complejidad lineal de las secuencias bent está acotada inferiormente por 76864 mientras que la de la clase de secuencias estudiada por Rueppel para k=12 está acotada inferiormente por $2'7 \cdot 10^6$. No obstante en ambos casos las cotas obtenidas sólo son aplicables a ejemplos concretos de secuencias filtradas y en ningún caso valdrían para una función no lineal general. En este trabajo sin embargo se darán cotas válidas para funciones no lineales generales.

Rueppel en su análisis separa los cosets cuyos pesos coinciden con el orden de la función no lineal de aquéllos con peso menor. La razón es la existencia de una representación única de los exponentes E de peso k.

Los cosets de peso menor que k también pueden estar presentes en la expresión de z. Sin embargo, debido a que no poseen una única representación en base 2, no es posible hacer un análisis análogo al anterior. Por eso en adelante, siempre que no se indique lo contrario, se entenderá que los cosets analizados son de peso k.

Rueppel en su análisis concluye que 'el caso de un RDRL filtrado no linealmente es extremadamente difícil de manejar desde un punto de vista matemático. Pueden ocurrir degeneraciones muy grandes en la complejidad lineal de las secuencias producidas sin posibilidad de predecirlas. Esta impredecibilidad de la complejidad lineal resultante puede haber causado que muchos diseñadores de generadores de secuencias cifrantes se resistan a usar el tipo de generador no lineal discutido'.

En este trabajo se abordará el problema de la impredecibilidad de la complejidad lineal del filtrado no lineal.

1.3.5. Recapitulación de Resultados

Hasta aquí se han introducido los diversos puntos de vista de los autores más renombrados en la materia. Se observan claramente dos vías de actuación en el análisis de la complejidad lineal:

(a) Mediante el análisis de los dígitos de la secuencia.

(b) Mediante el estudio de las características de la función no lineal utilizada como filtrado.

El único autor de los mencionados que sigue la primera vía de acercamiento es Massey. El método que utiliza es el mencionado en el apartado 1.1.4, el algoritmo de Berlekamp-Massey. El resto de autores, cuyo trabajo se ha comentado en los últimos apartados, utilizan la segunda vía de acercamiento. En particular definen clases de funciones para las que son capaces de garantizar unas determinadas cotas de la complejidad lineal de las secuencias filtradas resultantes. En la siguiente tabla quedan sintetizados por orden cronológico los resultados obtenidos según esta forma de actuar.

Autores	*Condiciones*	*Cota*
Key	*orden* k	$\Lambda \leq \sum_{i=1}^{k} \binom{L}{i}$
Kumar y Scholtz	*secuenciasbent* $L = \dot{4}, k \leq L/4$	$\Lambda \geq \binom{L/2}{L/4} 2^{L/4}$
Rueppel	$mcd(\delta, 2^L - 1) = 1$ *término de mayor orden* : $\sum_{i=0}^{N-1} b_i s_{n+i} s_{n+i+\delta} \cdots s_{n+i+(k-1)\delta}$	$\Lambda \geq \binom{L}{k} - (N-1)$
Massey y Serconek	L *primo*, $s_n s_{n+\delta}$	$\Lambda = L + \binom{L}{2}$
Paterson	L *primo*, *término de mayor orden* : $\sum_{i=0}^{N-1} b_i s_{n+i} s_{n+i+\delta} \cdots s_{n+i+(k-1)\delta}$	$(t : 2^L - 1 \mid \delta(2^t - 1))$ $\Lambda \geq \binom{t}{k} (\frac{L}{t})^k - (N-1)$

Obsérvese que en todos los casos en los que se obtiene una cota inferior las condiciones exigidas a la función son bastante restrictivas. En el próximo capítulo se dará una cota inferior de la complejidad lineal válida para una clase de funciones mucho más amplia.

Capítulo 2

Cotas de la Complejidad Lineal

En este capítulo se toma como base el 'test de presencia de raíces' presentado en el capítulo anterior, para obtener mediante algunos tipos especiales de cosets, varias cotas superiores e inferiores de la complejidad lineal global de la secuencia filtrada. La característica especial de dichos cosets consiste en que, a diferencia del resto de cosets, su degeneración o no degeneración se puede analizar completamente. Además, en muchos casos dicho análisis será llevado a cabo de manera independiente del filtrado no lineal considerado. Por tanto, nuestra estrategia en cierta forma va contra corriente ya que el resto de autores, según lo visto, se centran básicamente en la búsqueda de funciones no lineales concretas para las que se puede garantizar la no degeneración de todos los cosets. Nosotros por el contrario en este capítulo trabajamos en tres sentidos distintos al anterior:

1. encontramos un grupo de cosets que resultan ser no degenerados para todos los filtrados no lineales (apartado 2.4),

2. encontramos un grupo de cosets que resultan ser degenerados para un grupo amplio de filtrados no lineales (apartado 2.8),

3. encontramos un grupo amplio de cosets que resultan ser degenerados para un grupo de filtrados no lineales (apartado 2.8).

   También hacemos una incursión en el mismo sentido que los autores anteriormente mencionados ya que:

4. encontramos un grupo de filtrados no lineales para los que se garantiza la no degeneración de todos los cosets (apartado 2.5).

El punto 1 constituye la primera de las dos ideas fundamentales correspondientes al análisis que se comienza en este capítulo y se completa en el próximo:

(I1) Independientemente de la función no lineal y del RDRL de máxima longitud escogidos, algunos cosets nunca son degenerados.

(I2) Si algunos cosets son degenerados, otros no pueden serlo.

Los resultados del apartado 2.4 son sin lugar a dudas los más destacables de todo el capítulo ya que proporcionan cotas inferiores a la complejidad lineal global completamente válidas para una amplia clase de filtrados no lineales. Esto resulta doblemente provechoso ya que por un lado, siguiendo las recomendaciones de Rueppel [152], se garantizan complejidades lineales mínimas, es decir, se garantiza la seguridad de los filtrados no lineales implicados. Por otro lado, dada la ausencia de restricciones, la cantidad de filtrados no lineales implicados es bastante considerable.

Dado que todo este análisis se basa en la manipulación de cosets, daremos en el primer apartado algunas representaciones de los mismos.

2.1. Representaciones de los Cosets

A partir del 'test de presencia de raíces' se pueden extraer varias caracterizaciones válidas para cualquier coset E [18]. Nos referiremos a ellas como C1, C2,...

En primer lugar, la caracterización más obvia se obtiene al considerar el elemento E como un entero expresado como suma de potencias de 2. Ésta es la forma utilizada por Rueppel en su trabajo [152].

C1. E=$2^{e_0} + 2^{e_1} + \cdots + 2^{e_{k-1}}$ siendo $0 \leq e_0 < e_1 < \cdots < e_{k-1} < L$.

La representación anterior conduce a una representación de E como cadena binaria cuando se expresa en base 2.

C2. E se puede representar como una cadena binaria de longitud L y k unos en las posiciones $\{e_i\}_{i=0,1,\ldots,k-1}$ contadas a partir de la derecha.

El 'test de presencia de raíces' proporciona una tercera representación ya que el determinante A_E define el coset E.

C3. El determinante asociado a E, A_E.

Esta representación depende además del polinomio característico y del término de mayor orden de la función no lineal.

A partir de ese determinante se puede definir un sistema de ecuaciones

dado por

$$\begin{cases} 0 = & d_0\alpha^{t_0 2^{e_0}} + d_1\alpha^{t_1 2^{e_0}} + \cdots + d_{k-1}\alpha^{t_{k-1} 2^{e_0}} \\ 0 = & d_0\alpha^{t_0 2^{e_1}} + d_1\alpha^{t_1 2^{e_1}} + \cdots + d_{k-1}\alpha^{t_{k-1} 2^{e_1}} \\ & \vdots \\ 0 = & d_0\alpha^{t_0 2^{e_{k-1}}} + d_1\alpha^{t_1 2^{e_{k-1}}} + \cdots + d_{k-1}\alpha^{t_{k-1} 2^{e_{k-1}}} \end{cases} \tag{2.1}$$

donde $d_j \in GF(2^L)\ \forall j$.

Por tanto, una última caracterización de E es el sistema asociado a A_E.

C4. El sistema lineal homogéneo (1) asociado a A_E.

En este trabajo se utiliza principalmente la caracterización C2, pero en muchos casos el resto de caracterizaciones son utilizadas indistintamente.

Para el uso de las cadenas binarias se hace necesaria cierta notación adicional.

Dadas dos cadenas binarias E y F de longitud L, cuyos unos están situados en las posiciones indicadas respectivamente por $\{e_i\}_{i=0,1,...,k-1}$ y $\{f_i\}_{i=0,1,...,l-1}$ con k¡l, la notación **E⊂F** significa que todos los unos de E están también en F, es decir, que $\{e_i\}_{i=0,1,...,k-1} \subset \{f_i\}_{i=0,1,...,l-1}$.

Dado un grupo de cadenas binarias $\{E_n\} = \{E_1, E_2, ..., E_N\}$, **OR**[**{E**$_n$**}**] denota la cadena binaria resultante de una operación OR entre las N cadenas binarias del grupo. Obviamente se tiene que $\forall n \in \{1, 2, ..., N\},\ E_n \subset OR[\{E_n\}]$.

Esta notación se utilizará primordialmente en el sexto apartado de este capítulo.

En el próximo apartado se introduce una de las herramientas más destacadas de este trabajo. Se define un grupo de cosets que son no degenerados independientemente del filtrado no lineal.

2.2. Cosets de Distancia Fija

El grupo de cosets al que hace referencia el título, se puede definir de la siguiente forma [51].

Se llama **coset de distancia fija d** de peso k al coset E_d tal que $E_d = 2^{e_0} + 2^{e_1} + \cdots + 2^{e_{k-1}}$ con $e_i \equiv d \cdot i\ (mod\ L)$ (i=0,1,...,k-1) siendo d un entero positivo menor que L y tal que mcd(d,L)=1.

El apelativo 'de distancia fija' hace alusión a la distancia fija d existente entre los unos de la cadena binaria asociada a E_d.

Ejemplo 2.2.1

Para L=8 se tienen 1, 3, 5 y 7 como posibles valores de d primos con L. Consideramos pues d=1 y d=3. Los cosets de distancia fija para cada posible valor de k≤ L, correspondientes a d=1 son 00000001, 00000011, 00000111, 00001111, 00011111, 00111111, 011111111 y 11111111. Los cosets de distancia fija para cada posible valor de k≤ L, correspondientes a d=3 son 00000001, 00001001, 01001001, 01001011, 01011011, 11011011, 11011111 y 11111111. Para k= 4 los cosets de distancia fija son 00001111 (d=1), 01001011 (d=3), 10100101 (d=5) y 11100001 (d=7). A la vista de este ejemplo se deduce que los cosets de distancias fijas d=1 y d=7 coinciden porque sus correspondientes elementos representativos son rotaciones cíclicas. Lo mismo sucede para d=3 y d=5. Por tanto para este ejemplo sólo se tienen dos cosets de distancia fija distintos.

En adelante se llamará **j-ésimo uno** de un coset de distancia fija E_d al uno situado en la posición e_j. Para simplificar la notación se denotará mediante $\mathbf{A}_d$ el determinante A_{E_d}.

Mediante estos cosets, en el cuarto apartado de este capítulo se concretará la primera idea fundamental con la que trabajamos (I1).

Para llevar a cabo su demostración es necesario el resultado presentado en el próximo apartado.

2.3. Orden de la Función

A continuación se da un resultado [15] que proporciona una condición sencilla para garantizar el orden de la función no lineal.

Lema 2.3.1

El producto de k fases distintas $\{s_{n+t_j}\}$ de una PN-secuencia es una función de orden k si y sólo si las potencias $\alpha^{t_j} \in GF(2^L)$ (j=0,1,...,k-1) son linealmente independientes sobre GF(2).

Demostración El resultado se deduce a partir del trabajo de Key [85] según el cual cualquier α^{t_l} verifica que $\alpha^{t_l} = \sum_{j=0, j\neq l}^{k-1} d_j \alpha^{t_j}$, $d_j \in GF(2)$ si y sólo si s_{n+t_l} puede escribirse como $s_{n+t_l} = \sum_{j=0, j\neq l}^{k-1} d_j s_{n+t_j}$. Además esta igualdad se da si y sólo si la función producto se puede escribir como $\mathrm{s}_{n+t_0} s_{n+t_1} \cdots s_{n+t_l}$.

$\cdot\cdot\, s_{n+t_{k-2}} s_{n+t_{k-1}} = s_{n+t_0} s_{n+t_1} \cdots \sum_{j=0, j\neq l}^{k-1} d_j s_{n+t_j} \cdots s_{n+t_{k-2}} s_{n+t_{k-1}}$. Así pues, si las potencias α^{t_j} (j=0,1,...,k-1) fueran linealmente dependientes sobre GF(2), entonces la función producto sería o bien una función de orden k-1 (cuando el número de $d_j \neq 0$ es impar), o bien la función idénticamente nula (cuando ese número es par).

Recíprocamente, si la función producto no fuera de orden k entonces las potencias α^{t_j} (j=0,1,...,k-1) serían linealmente dependientes sobre GF(2).

Una forma fácil de garantizar la condición del lema anterior consiste en tomar las k fases $\{s_{n+t_j}\}$ dentro de un estado del RDRL, ya que $\{\alpha^0, \alpha^1, ..., \alpha^{L-1}\}$ constituyen siempre una base de GF(2^L) sobre GF(2).

A continuación se demuestra mediante los cosets de distancia fija definidos en el apartado anterior la idea (I1) anteriormente mencionada.

2.4. Cota Inferior General

La demostración de la idea (I1) descrita anteriormente proporciona una cota inferior general a la complejidad lineal global por lo que representa una de las principales aportaciones de este trabajo.

Concretamente en este apartado se demuestra [51] que todos los cosets de distancia fija son no degenerados independientemente de la función no lineal y del RDRL considerados, siempre que la función contenga un único término de orden máximo.

Esta condición se debe a que la base sobre la que se asienta el análisis realizado es el test de presencia de raíces que sirve para analizar la contribución a la complejidad lineal del término de orden máximo. También como consecuencia de dicho test, los únicos cosets analizados en éste y posteriores apartados de este capítulo son los de peso coincidente con el orden de la función.

En primer lugar se presenta un resultado bastante destacable en dos aspectos. Por un lado no se exige ninguna condición sobre el filtrado no lineal y por otro lado, servirá para obtener en posteriores resultados, cotas inferiores a la complejidad lineal global del filtrado no lineal.

Teorema 2.4.1 (Teorema Principal)

Sea f una función no lineal con un único término de orden máximo, entonces f es una función no lineal de orden k si y sólo si todos los cosets de distancia fija y peso k son no degenerados.

Demostración '⇒' Por inducción sobre k. Para k=1, obviamente el coset de peso 1 es no degenerado para un filtrado de orden 1. Se supone cierto para k-1 de manera que, para cualquier función de orden k-1 con un único término de orden máximo, todos los cosets de distancia fija y peso k-1 son no degenerados. Se demuestra para k. Se supone que para la función f de orden k existe un coset de distancia fija d y peso k que es degenerado. Esto significa que el correspondiente determinante A_d se anula. En ese caso, el sistema lineal homogéneo (1) asociado a A_d es compatible y tiene soluciones no triviales $d_i \in GF(2^L)$ (i=0,1,...,k-1). Si se elevan a 2^d las ecuaciones de dicho sistema, se obtiene un nuevo sistema lineal homogéneo donde las primeras k-1 ecuaciones coinciden con las k-1 últimas del sistema original. Según la hipótesis de inducción aplicada sobre la función producto de orden k-1, se tiene que en particular ese subsistema de k-1 ecuaciones tiene su determinante asociado no nulo. Según esto, resolviendo por la regla de Cramer se deduce que las soluciones del sistema asociado a A_d deben ser de la forma $d_i \equiv (d_i)^{2^d}$ en GF(2^L) (i=0,1,...,k-1). Esto implica que el orden de d_i divide a 2^d-1 y, como 2^d-1 y 2^L-1 son primos entre sí, se concluye que los d_i han de ser coeficientes binarios. Por tanto, según el lema anterior, la función f no puede ser de orden k.

'⇐' Por reducción al absurdo. Si la función f no fuera de orden k, según el lema anterior, las potencias α^{t_j} (j=0,1,...,k-1) serían linealmente dependientes sobre GF(2). Es decir, $\exists l : 0 \leq l \leq k-1$ tal que $\alpha^{t_l} = \sum_{j=0, j\neq l}^{k-1} d_j \alpha^{t_j}$, siendo $d_j \in$GF(2) no todos nulos. De ahí que $d_j \equiv (d_j)^{2^d}$ en GF(2^L) por lo que para cualquier coset E, el sistema (1) asociado a A_E sería compatible con soluciones binarias no triviales. Consecuentemente, todos los cosets de peso k serían degenerados, y en particular cualquier coset de distancia fija también lo sería.

La repercusión de este resultado es clara. Proporciona una cota inferior general a la complejidad lineal global de cualquier filtrado no lineal con un único término de máximo orden. Esto se muestra en los siguientes resultados, donde Λ representa la complejidad lineal global del filtrado no lineal.

Corolario 2.4.1

Sea f una función no lineal con un único término de orden máximo, entonces la complejidad lineal global del filtrado no lineal resultante está acotada inferiormente según $\Lambda \geq N_L \cdot L$, *donde* $N_L = \frac{\Phi(L)}{2}$ *(siendo* $\Phi(L)$ *la función de Euler) representa el número de cosets de distancia fija y L el cardinal de esos cosets.*

Demostración Obviamente el número N_L se expresa en función de $\Phi(L)$ por propia definición de coset de distancia fija. Concretamente N_L es igual a $\frac{\Phi(L)}{2}$ por el siguiente razonamiento: Para los valores d y L-d primos con L se obtiene el mismo coset de distancia fija E_d=E_{L-d} ya que $\forall i = 0, 1, ..., k-1$ $di + (1-k)d \equiv di + d - dk \equiv (-d)(k-i-1) \equiv (L-d)(k-i-1)$ (mod L), luego los conjuntos $\{e_i^d\}_{i=0,...,k-1}$ y $\{e_i^{L-d}\}_{i=0,...,k-1}$, correspondientes respectivamente a los cosets de E_d y E_{L-d} coinciden.

De esta forma se tiene que la complejidad lineal global de cualquier filtrado no lineal con un único término de orden máximo pertenece al intervalo cerrado cuyos límites vienen dados por la cota inferior obtenida aquí y la cota superior dada por Key.

El mayor valor que puede tomar la cota anterior resulta cuando L es primo ya que entonces existen muchos más cosets de distancia fija. En ese caso concreto, de manera trivial se llega a la siguiente cota.

Corolario 2.4.2

Sea f una función no lineal con un único término de orden máximo, si L es primo, entonces la complejidad lineal global del filtrado no lineal resultante está acotada inferiormente según $\Lambda \geq \binom{L}{2}$.

Obsérvese que ambos resultados son independientes tanto del polinomio característico del RDRL como del orden y forma particulares de la función no lineal. Esto significa que para cualquier función no lineal con un único término de orden máximo aplicado sobre un RDRL cualquiera de longitud L siempre se sabe que $\Phi(L)/2$ cosets de cardinal L son no degenerados. Si se pretendiera llegar a esta misma conclusión mediante el test de presencia de raíces habría que calcular al menos $\Phi(L)/2$ determinantes de orden k en el cuerpo finito $GF(2^L)$ para cada uno de los posibles filtrados no lineales, es decir para cada uno de los posibles polinomios característicos primitivos del RDRL y para cada una de las posibles funciones producto de orden k.

Por ejemplo, si L=17 y k=8, para cada uno de esos posibles casos habría que calcular al menos 8 determinantes de orden 8 en el cuerpo GF(2^{17}) ya que la cota dada por el corolario 2.2 es 136.

En conclusión, los resultados presentados en este apartado proporcionan una complejidad lineal mínima para una amplia clase de filtrados no lineales, con lo que se consigue para éstos un mínimo de seguridad garantizada.

El teorema 2.1 puede replantearse desde un punto de vista dual, consiguiéndose así una cota inferior mejor a base de exigir mayores restricciones al filtrado no lineal. Esto es lo que se hace en el próximo apartado.

2.5. Producto de Fases $2^{((d))}$-distantes

El título del apartado hace referencia a un grupo de funciones no lineales que se definen de la siguiente forma.

Se llama **producto de k fases $2^{((d))}$-distantes** a cualquier función que sea el producto de las k fases distintas $\{s_{n+2^{r_j}}\}_{j=0,1,\ldots,k-1}$ de una PN-secuencia, donde $r_j \equiv d \cdot j \ (mod\ L)$ siendo d un entero positivo menor que L tal que mcd(d,L)=1. Es decir, se llama de esta manera a la función no lineal de la forma

$$s_{n+1} \cdot s_{n+2^{((d))}} \cdot s_{n+2^{((2d))}} \cdots s_{n+2^{(((k-1)d))}}$$

donde el doble paréntesis (()) indica que el entero que se encuentra dentro debe ser considerado en módulo L. A este tipo de funciones se las denotará como $\mathbf{f}_d$.

En primer lugar se presenta un resultado que determina los valores de k para los que se puede asegurar que el orden de los productos de k fases $2^{((d))}$-distantes es realmente k.

Lema 2.5.1

Dada una función no lineal f cuyo único término de orden máximo es un producto de k fases $2^{((d))}$-distantes f_d, entonces f es una función de orden k si y sólo si k$\leq$ L.

Demostración '$\Longrightarrow$' Si se supone por reducción al absurdo que $k > L$, entonces f_d quedaría de la forma

$$s_{n+1} \cdot s_{n+2^{((d))}} \cdot s_{n+2^{((2d))}} \cdots s_{n+2^{(((L-1)d))}} \cdot s_{n+2^{((Ld))}} \cdot s_{n+2^{(((L+1)d))}} \cdots s_{n+2^{(((k-1)d))}} =$$
$$s_{n+1} \cdot s_{n+2^{((d))}} \cdot s_{n+2^{((2d))}} \cdots s_{n+2^{(((L-1)d))}} \cdot s_{n+1} \cdot s_{n+2^{((d))}} \cdots =$$
$$s_{n+1} \cdot s_{n+2^{((d))}} \cdot s_{n+2^{((2d))}} \cdots s_{n+2^{(((L-1)d))}}.$$

Luego la función f no sería de orden k sino de orden L.

'$\Longleftarrow$' Dado que $\{\alpha, \alpha^2, \alpha^{2^2}, \alpha^{2^3}, ..., \alpha^{2^{L-1}}\}$ es una base de GF(2^L) sobre GF(2) y $\{\alpha, \alpha^{2^{((d))}}, \alpha^{2^{((2d))}}, \alpha^{2^{((3d))}}, ..., \alpha^{2^{(((k-1)d))}}\} \subseteq \{\alpha, \alpha^2, \alpha^{2^2}, \alpha^{2^3}, ..., \alpha^{2^{L-1}}\}$, se tiene que $\alpha, \alpha^{2^{((d))}}, \alpha^{2^{((2d))}}, \alpha^{2^{((3d))}}, ..., \alpha^{2^{(((k-1)d))}}$ son linealmente independientes sobre GF(2) y por el lema 2.1 se tiene que la función f es de orden k.

Según lo anterior y los principios de diseño vistos en el apartado 1.2.1, de ahora en adelante y salvo que se indique lo contrario, siempre que se mencione un producto de k fases $2^{((d))}$-distantes se debe entender que k es menor que L.

A continuación se demuestra, para las funciones producto de fases $2^{((d))}$-distantes, un resultado análogo al demostrado por Rueppel para los productos de fases equidistantes [152]. Según el próximo teorema, los productos de fases $2^{((d))}-$distantes constituyen una nueva clase de funciones para las que se tiene asegurada la no degeneración de todos los cosets de peso k.

Teorema 2.5.1

Dada una función no lineal f cuyo único término de máximo orden es un producto de k fases $2^{((d))}$-distantes f_d, entonces f es una función de orden k si y sólo si todos los cosets de peso k son no degenerados.

Demostración '$\Rightarrow$' Por inducción sobre k. Para k=1 obviamente el único coset de peso uno es no degenerado. Se supone cierto para k-1, de manera que para cualquier función de orden k-1 de esas características, todos los cosets de peso k-1 son no degenerados. Se demuestra para k. Para ello se supone que para alguna de esas funciones de orden k existe un coset E de peso k que sí es degenerado. Esto significa que el determinante correspondiente A_E se anula. En ese caso, el sistema lineal homogéneo asociado a A_E

$$\begin{cases} 0 = & d_0\alpha^{2^{r_0}2^{e_0}} + d_1\alpha^{2^{r_0}2^{e_1}} + \cdots + d_{k-1}\alpha^{2^{r_0}2^{e_{k-1}}} \\ 0 = & d_0\alpha^{2^{r_1}2^{e_0}} + d_1\alpha^{2^{r_1}2^{e_1}} + \cdots + d_{k-1}\alpha^{2^{r_1}2^{e_{k-1}}} \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad . \\ 0 = & d_0\alpha^{2^{r_{k-1}}2^{e_0}} + d_1\alpha^{2^{r_{k-1}}2^{e_1}} + \cdots + d_{k-1}\alpha^{2^{r_{k-1}}2^{e_{k-1}}} \end{cases}$$

con $r_j \equiv d \cdot j \ (mod\ L)$ y mcd(d,L)=1, es compatible con soluciones no triviales $d_i \in GF(2^L)$ (i=0,1,...,k-1). Si se elevan a 2^d las ecuaciones de dicho sistema, se obtiene un nuevo sistema lineal homogéneo donde las primeras k-1 ecuaciones coinciden con las k-1 últimas del sistema anterior. Según la

hipótesis de inducción aplicada a la función producto de k-1 fases $2^{((d))}$-distantes $s_{n+2^{r_1}} s_{n+2^{r_2}} \cdots s_{n+2^{r_{k-1}}}$ se tiene en particular que ese subsistema de k-1 ecuaciones tiene su determinante no nulo. Según esto, resolviendo por la regla de Cramer se deduce que las soluciones del sistema asociado a A_E deben ser de la forma $d_i \equiv (d_i)^{2^d}$ (i=0,1,...,k-1). Esto implica que el orden de d_i divide a 2^d-1, y como 2^d-1 y 2^L-1 son primos entre sí, se concluye que los d_i han de ser coeficientes binarios. Por tanto, según el resultado del apartado 2.3, f no puede ser una función de orden k.

'$\Leftarrow$' Por reducción al absurdo. Si f no fuera una función de orden k, según el mismo resultado del apartado 2.3, las potencias $\alpha^{2^{r_j}}$ (j=0,1,...,k-1) deberán ser linealmente dependientes sobre GF(2). Es decir, $\exists l : 0 \leq l \leq k-1$ tal que $\alpha^{2^{r_l}} = \sum_{j=0, j\neq l}^{k-1} d_j \alpha^{2^{r_j}}$, $d_j \in$GF(2) no todos nulos. De ahí que $d_j = d_j^2$, por lo que para cualquier coset E, el sistema asociado a A_E sería compatible con soluciones binarias no triviales. Consecuentemente, todos los cosets de peso k serían degenerados.

Del resultado anterior se deduce directamente una cota inferior a la complejidad lineal global, que resulta ser la más alta que se puede conseguir mediante los cosets de peso igual al orden de la función.

Corolario 2.5.1

La complejidad lineal global de una función no lineal cuyo único término de máximo orden es un producto de k fases $2^{((d))}$-distantes está acotada inferiormente según $\Lambda \geq \binom{L}{k}$

El próximo corolario proporciona un recuento de las funciones a las que se pueden aplicar los resultados anteriores. Obsérvese que este recuento resulta favorable siempre que L sea primo y k sea menor que L. Esta última condición coincide con la mencionada anteriormente.

Corolario 2.5.2

El número de funciones producto de k fases $2^{((d))}$-distantes diferentes viene dado por

a) 1 si $k = L$

b) $\Phi(L)$ si $k < L$.

Demostración a) Si k=L entonces el producto de k fases $2^{((d))}$-distantes viene dado por $s_{n+1} \cdot s_{n+2^{((d))}} \cdot s_{n+2^{((2d))}} \cdots s_{n+2^{(((L-1)d))}}$. Para cualquier valor d primo con L, mediante la conmutatividad del producto se demuestra que ese término coincide con $s_{n+1} \cdot s_{n+2} \cdot s_{n+2^2} \cdots s_{n+2^{(L-1)}}$. Luego ésta es la única función producto de k fases $2^{((d))}$-distantes cuando k=L.

b) Si $k < L$ entonces para cada valor d primo con L, se ve que cada función producto de k fases $2^{((d))}$-distantes es diferente. Para ello se supone por reducción al absurdo que hay dos funciones producto de este tipo con distancias d_1 y d_2 ($d_1 \neq d_2$), que son iguales. En ese caso se tendría que los conjuntos de fases de cada producto coincidirían salvo rotaciones cíclicas de las secuencias, lo cual sería imposible dadas sus definiciones respectivas como productos de fases crecientes $s_{n+1} \cdot s_{n+2^{((d_1))}} \cdot s_{n+2^{((2d_1))}} \cdots s_{n+2^{(((k-1)d_1))}}$ y $s_{n+1} \cdot s_{n+2^{((d_2))}} \cdot s_{n+2^{((2d_2))}} \cdots s_{n+2^{(((k-1)d_2))}}$ con $d_1 \neq d_2$. Dado que existen $\Phi(L)$ valores d distintos, éste es el número de funciones distintas buscado.

Ejemplo 2.5.1

Para L=7 y k=4, los $\Phi(7) = 6$ *productos de 4 fases* $2^{((d))}$*-distantes, con* $d \in \{1, 2, 3, 4, 5, 6\}$ *son respectivamente* $s_{n+1} \cdot s_{n+2} \cdot s_{n+4} \cdot s_{n+8}$, $s_{n+1} \cdot s_{n+4} \cdot s_{n+16} \cdot s_{n+64}$, $s_{n+1} \cdot s_{n+4} \cdot s_{n+8} \cdot s_{n+64}$, $s_{n+1} \cdot s_{n+2} \cdot s_{n+16} \cdot s_{n+32}$, $s_{n+1} \cdot s_{n+2} \cdot s_{n+8} \cdot s_{n+32}$ *y* $s_{n+1} \cdot s_{n+16} \cdot s_{n+32} \cdot s_{n+64}$*, que tal como se aprecia, efectivamente son todos distintos.*

De los últimos resultados se puede concluir que el diseñador de un generador del tipo discutido podría simplemente considerar cualquier función no lineal de orden menor que k y sumarle un único producto de k fases $2^{((d))}$-distantes. La complejidad lineal global del filtrado no lineal resultante sería mayor o igual que $\binom{L}{k}$. Esta cota resulta ser un número muy grande a partir de unos valores de L y k bastante prácticos, como por ejemplo 127 y 64 respectivamente para los que la cota anterior toma un valor del orden 10^{37}. No obstante, tal como se mencionó en los principios de diseño del capítulo anterior, hay que ser cauteloso con las funciones con un único término de orden máximo. Por esto, a continuación presentamos un resultado análogo para una clase de funciones criptográficamente más interesantes.

Corolario 2.5.3

La complejidad lineal global de un filtrado no lineal cuyo término de orden máximo es una combinación lineal de productos de k fases $2^{((d))}$*-distantes,*

$\sum_{i=0}^{N-1} b_i s_{n+i+1} \cdot s_{n+i+2^{((d))}} \cdot s_{n+i+2^{((2d))}} \cdots s_{n+i+2^{(((k-1)d))}}$, *donde N es un entero positivo y no todos los* b_i *son nulos, está acotada inferiormente según* $\Lambda \geq \binom{L}{k} - (N-1)$.

Demostración Como se demostró en el último teorema, todas las funciones producto de k fases $2^{((d))}$-distantes tienen todos los cosets E de peso k no degenerados. Esto significa que cada producto de la forma $s_{n+i+1} \cdot s_{n+i+2^{((d))}} \cdot s_{n+i+2^{((2d))}} \cdots s_{n+i+2^{(((k-1)d))}}$ posee, en su expresión en función de las potencias de una raíz primitiva del polinomio minimal de la secuencia, un determinante denotado como $(\mathrm{A}_E)_i$ no nulo como coeficiente de cada raíz α^E. Ya que $(\mathrm{A}_E)_0 = A_E$ y $(\mathrm{A}_E)_i = \alpha^{iE} \cdot \mathrm{A}_E$, para la función de las hipótesis de este corolario se puede escribir el coeficiente que acompaña a cada raíz α^E de la forma $\sum_{i=0}^{N-1} b_i \alpha^{iE} A_E$. Por tanto, dado que $\mathrm{A}_E \neq 0$, el coset E es no degenerado siempre que $b_0 + b_1\alpha^E + \cdots + b_{N-1}(\alpha^E)^{N-1} \neq 0$ en GF(2^L). Dado que el polinomio $b_0 + b_1 x + \cdots + b_{N-1}x^{N-1}$ tiene como máximo N-1 raíces, se concluye que como máximo hay una disminución en la complejidad lineal de valor N-1.

Observación 2.5.1

Si N≤ L y L es primo, ninguno de los cosets de peso k puede ser degenerado, por lo que en este caso se mantiene la cota $\binom{L}{k}$.

La adición de una función f' de fases linealmente independientes no afecta a la cota obtenida siempre y cuando el orden de f' sea menor que k. De ahí se tiene la siguiente propuesta. Escoger el filtrado no lineal a aplicar sobre las etapas de un RDRL de máxima longitud con L etapas, según la expresión $z_n = \sum_{i=0}^{N-1} b_i s_{n+i} s_{n+i+\delta} \cdots s_{n+i+(k-1)\delta} + f'(s_n, s_{n+1}, ..., s_{n+L-1})$ siendo ord(f')<k. De esta forma la complejidad lineal de la secuencia resultante es mayor o igual que $\binom{L}{k} - (N-1)$.

En el próximo apartado, tomando como punto de partida los cosets de distancia fija, se define y analiza un nuevo grupo de cosets.

2.6. Quasicosets de Distancia Fija

A continuación presentamos un nuevo grupo de cosets [18] cuya definición obtenida a partir de los cosets de distancia fija, utiliza parte de la notación introducida en el primer apartado de este capítulo.

Dado un coset de distancia fija $E_d = 2^{e_0}+2^{e_1}+...+2^{e_{k-1}}$ y $j \in \{0,...,k-1\}$, se llama **j-ésimo quasicoset de distancia fija** (o en forma abreviada, **quasicoset d-f**) a cualquier coset que tenga un elemento F_d^j de la forma $F_d^j = 2^{f_0} + 2^{f_1} + ... + 2^{f_{k-1}}$ tal que $\{e_i\}_{i\neq j, i=0,...,k-1} \subset \{f_i\}_{i=0,...,k-1}$.

Ejemplo 2.6.1

Para L=9, k=5, los elementos representativos de los quasicosets de distancias fijas d=1, d=2 y d=4 son respectivamente

$$\left\{\begin{array}{l}000101111\\001001111\\010001111\\100001111\end{array}\right. \left\{\begin{array}{l}001010111\\001011101\\001110101\\011010101\end{array}\right. y \left\{\begin{array}{l}100011011\\100011101\\100111001\\101011001\end{array}\right.$$

Es decir, un j-ésimo quasicoset d-f F_d^j es cualquier coset cuya cadena binaria asociada contiene todos los unos de la cadena binaria asociada a E_d salvo el j-ésimo uno. De nuevo para simplificar notación, se denota mediante $\mathbf{A}_d^j$ al determinante $A_{F_d^j}$. Por otro lado, $\{\mathbf{F}_{d,n}^j\} = \{F_{d,1}^j, ..., F_{d,N}^j\}$ denota un conjunto de j-ésimos quasicosets d-f.

A partir de estos cosets, a continuación se presenta un resultado relacionado con la segunda idea fundamental que se maneja en este trabajo (I2).

Esta idea se utilizará en el próximo capítulo para incrementar la cota general obtenida en el apartado 2.4. En concreto se demostrará la no degeneración simultánea de grupos de quasicosets d-f cuando se verifican determinadas condiciones. Para realizar dicha demostración es necesario el siguiente resultado previo.

Lema 2.6.1

Sea F_d^j un j-ésimo quasicoset d-f cualquiera, entonces su determinante asociado A_d^j tiene al menos un menor de orden k-1 (sin la j-ésima fila y una

columna arbitraria) que no se anula:

$$\begin{vmatrix} \alpha^{t_0 2^{e_0}} & \cdot & \alpha^{t_{i-1} 2^{e_0}} & \alpha^{t_{i+1} 2^{e_0}} & \cdot & \alpha^{t_{k-1} 2^{e_0}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \alpha^{t_0 2^{e_{j-1}}} & \cdot & \alpha^{t_{i-1} 2^{e_{j-1}}} & \alpha^{t_{i+1} 2^{e_{j-1}}} & \cdot & \alpha^{t_{k-1} 2^{e_{j-1}}} \\ \alpha^{t_0 2^{e_{j+1}}} & \cdot & \alpha^{t_{i-1} 2^{e_{j+1}}} & \alpha^{t_{i+1} 2^{e_{j+1}}} & \cdot & \alpha^{t_{k-1} 2^{e_{j+1}}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \alpha^{t_0 2^{e_{k-1}}} & \cdot & \alpha^{t_{i-1} 2^{e_{k-1}}} & \alpha^{t_{i+1} 2^{e_{k-1}}} & \cdot & \alpha^{t_{k-1} 2^{e_{k-1}}} \end{vmatrix} \neq 0$$

Demostración F_d^j se define a partir del coset de distancia fija E_d que por el teorema 2.1 sabemos es no degenerado. Esto significa que su determinante asociado A_d no se anula, por tanto, $\forall j \in \{0, 1, ..., k-1\}$ existe al menos un menor de orden k-1 (sin la fila j-ésima y una columna i-ésima) que no se anula.

El resultado anterior se utiliza en la demostración del siguiente teorema, que sirve para apoyar la idea (I2).

Teorema 2.6.1

Dados E_d un coset de distancia fija y j∈{0,1,....,k-1}, si para algún grupo de j-ésimos quasicosets d-f $\{F_{d,n}^j\}$ existe al menos un coset de distancia fija $E_{d'}$ tal que $E_{d'} \subset OR[\{F_{d,n}^j\}]$, entonces los cosets de $\{F_{d,n}^j\}$ no pueden ser todos simultáneamente degenerados.

Demostración Se supone por reducción al absurdo que los cosets de $\{\mathrm{F}_{d,n}^j\}$ son simultáneamente degenerados. Esta degeneración simultánea es equivalente a la existencia de un grupo de sistemas compatibles asociados a cada determinante $\mathrm{A}_{d,n}^j$ que tienen solución no trivial y k-1 ecuaciones en común. Además, según el lema anterior, estas k-1 ecuaciones tienen una única solución expresada en función de $\alpha^{t_i 2^{e_j}}$. Como ésta es una solución común a todos los sistemas, puede deducirse que el sistema compuesto por todas las ecuaciones tiene soluciones no triviales. Finalmente, según la hipótesis, las k ecuaciones asociadas al determinante $\mathrm{A}_{d'}$ están entre las ecuaciones del sistema completo, lo que lleva a una contradicción al tener un sistema compatible con soluciones no triviales un subsistema homogéneo cuya única solución es la trivial.

El grupo de cosets definido en este apartado será analizado con mayor profundidad en el próximo capítulo, donde se desarrollará un algoritmo especialmente diseñado para la comprobación de las circunstancias referidas en el teorema anterior.

Por último y como parte más amplia de este capítulo se presenta en el próximo apartado un grupo de cosets cuya degeneración se demuestra en múltiples casos, por lo que son los responsables de una disminución en el valor de la complejidad lineal global de muchos filtrados no lineales.

2.7. Cosets Simétricos

A diferencia de los grupos de cosets definidos en los apartados anteriores, la existencia del nuevo grupo de cosets que se definen a continuación viene dada por los valores específicos de L y k. En particular juegan un papel fundamental los factores comunes de L y k, a los que se les denota $\mathbf{fc_i(L,k)}$. Para determinados pares (L,k) no existe ningún coset de los que se llamarán simétricos, mientras que para otros pares existe una cantidad muy superior a la de cosets de distancia fija y a la de quasicosets d-f.

Se llama **coset simétrico** a todo coset de cardinal menor que L.

Como la definición anterior no proporciona una idea clara de la forma particular de estos cosets, a continuación se presentan unas cuantas caracterizaciones equivalentes.

Dado que todo coset está formado por todas las rotaciones cíclicas de una cualquiera de sus cadenas binarias componentes, se dice que un coset es simétrico si al rotar cíclicamente una cualquiera de sus cadenas binarias menos de L posiciones se obtiene la misma cadena. Esto queda expresado mediante el siguiente teorema de caracterización de los cosets simétricos, donde se utiliza la caracterización C1 de los cosets.

Teorema 2.7.1

Dada una longitud L, un coset E es simétrico si y sólo si $\exists i : 0 < i < L$ *tal que* $2^i \cdot E \sim E \ (mod\ 2^L - 1)$.

Para comprender mejor la estructura de estos cosets, a continuación se analiza uno cualquiera de sus elementos a los que se les adjudica también el calificativo de simétrico.

Se llama **cadena simétrica** a cualquiera de las cadenas binarias que forman parte de un coset simétrico.

Dada su definición, toda cadena simétrica está constituida por repeticiones de una misma subcadena no simétrica. En particular, se trata de un número fc_i(L,k) de repeticiones y la subcadena mencionada es de longitud

L/fc$_i$(L,k) y peso k/fc$_i$(L,k). El cardinal del coset al que pertenece dicha cadena es L/fc$_i$(L,k).

Ejemplo 2.7.1

Para L=8, k=4 existen dos factores comunes fc$_1$(8,4)=2 y fc$_2$(8,4)=4. Cada factor define un coset simétrico, tal y como se muestra a continuación según las representaciones de los cosets en binario y como números enteros.

$$\begin{cases} 0011 : 0011 : 51 \\ 01100110 : 102 \\ 11001100 : 204 \\ 10011001 : 153 \end{cases} \quad y \quad \begin{cases} 01 : 01 : 01 : 01 : 85 \\ 10101010 : 170 \end{cases}$$

Obsérvese que, tal como se había indicado,las cuatro cadenas simétricas del primer coset están formadas por dos subcadenas de longitud cuatro y peso dos, y las del segundo están formadas por cuatro subcadenas de longitud dos y peso uno.

Observación 2.7.1

Este tipo de cosets está contenido dentro de la categoría de los cosets impropios que describe Golomb en [68], pero en general ambos conjuntos no coinciden.
Los cosets simétricos E se corresponden con los elementos de la forma α^E pertenecientes a los subcuerpos propios de GF(2^L) distintos de GF(2). Esto sin embargo no se verifica para todos los cosets impropios.

Ejemplo 2.7.2

Para L=4, utilizando las representaciones binarias y como números enteros

$$Coset\ E_1 : \begin{cases} 0001 : \ 1 \\ 0010 : \ 2 \\ 0100 : \ 4 \\ 1000 : \ 8 \end{cases}, \ Coset\ E_2 : \begin{cases} 0011 : \ 3 \\ 0110 : \ 6 \\ 1100 : \ 12 \\ 1001 : \ 9 \end{cases},$$

$$Coset\ E_3 : \begin{cases} 0101 : \ 5 \\ 1010 : \ 10 \end{cases}, \ Coset\ E_4 : \begin{cases} 0111 : \ 7 \\ 1110 : \ 14 \\ 1101 : \ 13 \\ 1011 : \ 11 \end{cases}.$$

Los cosets E_1 y E_4 son propios, los cosets E_2 y E_3 son impropios y el coset E_3 es simétrico. Este último coset se corresponde con los elementos α^5 y α^{10} pertenecientes al único subcuerpo propio de GF(2^4) distinto de GF(2), es decir, GF(2^2)=$\{0, 1, \alpha^5, \alpha^{10}\}$.

En general, cada factor común de L y k, fc_i(L,k) determina un grupo de cosets simétricos. A continuación se escoge representante para cada uno de estos grupos.

Se llama **representante** del grupo de cosets determinado por fc_i(L,k) al coset simétrico cuyo líder se forma a partir de la subcadena no simétrica con los k/fc_i(L,k) unos seguidos.

Esta cadena binaria es de la siguiente forma, donde cada bloque de unos es de longitud k/fc_i(L,k)

$$0..,0\underbrace{1..,11}: ... : 0..,0\underbrace{1..,11}: 0..,0\underbrace{1..,11}$$

Su caracterización equivalente como entero en base dos es

$$2^0 + 2^1 + \cdots + 2^{\frac{k}{fc_i(L,k)}-1} + 2^{\frac{L}{fc_i(L,k)}} + 2^{\frac{L}{fc_i(L,k)}+1} + \cdots + 2^{\frac{L+k}{fc_i(L,k)}-1} + 2^{\frac{2L}{fc_i(L,k)}} + 2^{\frac{2L}{fc_i(L,k)}+1} + \cdots + 2^{\frac{2L+k}{fc_i(Lk,)}-1} + \cdots + 2^{\frac{(fc_i(L,k)-1)L}{fc_i(L,k)}} + 2^{\frac{(fc_i(L,k)-1)L}{fc_i(L,k)}+1} + \cdots + 2^{\frac{(fc_i(L,k)-1)L+k}{fc_i(L,k)}-1}$$

De la anterior resulta una tercera caracterización de estos representantes, que viene dada explícitamente por el siguiente resultado.

Proposición 2.7.1

Sean L y k dos números enteros con factores comunes, el representante del coset simétrico correspondiente a fc_i(L,k) es un entero de forma general $E=(2^{\frac{k}{fc_i(L,k)}}-1)\ \frac{2^L-1}{2^{\frac{L}{fc_i(L,k)}}-1}$.

Demostración La representación binaria de E está constituida por fc_i(L,k) subcadenas con k/fc_i(L,k) unos consecutivos. La distancia entre bloques consecutivos de unos es de L/fc_i(L,k) posiciones. Sea n_j (j=1,2,...,fc_i(L,k)) la contribución al entero E de la j-ésima subcadena. Dado que n_j=($2^{\frac{k}{fc_i(L,k)}}$-1)$2^{(j-1)\frac{L}{fc_i(L,k)}}$ $\forall j = 1, 2, ..., fc_i(L,k)$, el sumatorio de esta expresión corresponde a la suma de los fc_i(L,k) primeros términos de una progresión geométrica de razón $2^{\frac{L}{fc_i(L,k)}}$, luego E= ($2^{\frac{k}{fc_i(L,k)}}$-1) (1+ $\frac{2^{\frac{L}{fc_i(L,k)}}(2^{L-\frac{L}{fc_i(L,k)}}-1)}{2^{\frac{L}{fc_i(L,k)}}-1}$)= ($2^{\frac{k}{fc_i(L,k)}}$-1) $\frac{2^L-1}{2^{\frac{L}{fc_i(L,k)}}-1}$.

Ejemplo 2.7.3

Para L=12, k=6 existen cosets simétricos para los distintos factores comunes $fc_1(12,6) = 2$, $fc_2(12,6) = 3$ *y* $fc_3(12,6) = 6$, *se tienen los representantes* $7\frac{4095}{63} = 455$, $3\frac{4095}{15} = 819$ *y* $\frac{4095}{3} = 1365$, *que se corresponden respectivamente con las cadenas binarias 000111000111, 001100110011 y 010101010101, representantes de los tres grupos de cosets simétricos. Al primer grupo de cosets simétricos pertenecen además los cosets simétricos 001011001011 y 010011010011. Sin embargo los otros dos cosets simétricos sólo están formados por los representantes.*

Tras dar tres caracterizaciones distintas para los cosets representantes, se presenta a continuación un resultado que caracteriza cada uno de los cosets simétricos determinados por un factor común de L y k. Se deriva directamente del hecho de que un mismo uno en dos subcadenas consecutivas están separados por L/fc$_i$(L,k) posiciones.

Proposición 2.7.2

Dados unos valores de L y k, todos los cosets de peso k, $E=2^{e_0} + 2^{e_1} + \cdots + 2^{e_{k-1}}$ *determinados por un* $fc_i(L,k)$ *cumplen que* $\forall j = 1, 2, ..., fc_i(L,k) - 1$,

$$\left(e_0, e_1, ..., e_{\frac{k}{fc_i(L,k)}-1}\right) + \frac{L}{fc_i(L,k)} j(1,1,...,1) = \left(e_{\frac{jk}{fc_i(L,k)}}, e_{\frac{jk}{fc_i(L,k)}+1}, ..., e_{\frac{jk}{fc_i(L,k)}+\frac{k}{fc_i(L,k)}-1}\right).$$

Una de las características más destacables de estos cosets es que para valores de L y k no muy grandes, el número de cadenas simétricas puede ser muy elevado. Por ejemplo para L= 128 y k= 64, este número es del orden de 10^{37}. Por ello, el siguiente resultado tiene bastante trascendencia en lo que resta de capítulo. Consiste en un recuento de cosets simétricos y se obtiene a partir de la siguiente idea básica. Para cada fc$_i(L,k)$ existen tantos cosets simétricos como cosets no simétricos hay de longitud L/fc$_i(L,k)$ y k/fc$_i(L,k)$ unos.

Proposición 2.7.3

Dados unos valores de L y k, el número de cosets simétricos de cardinal $L/fc_i(L,k)$ *viene dado por la fórmula recursiva*

$$S_{\frac{L}{fc_i(L,k)}} = \frac{(L)/fc_i(L,k)k/fc_i(L,k) - \sum_j r_j S_{r_j}}{L/fc_i(L,k)},$$

donde r_j es cada uno de los divisores propios de $L/fc_i(L,k)$ que pertenecen al conjunto $\{L/fc_{i'}(L,k)\}_{i'\neq i}$.

Demostración Cada una de las cadenas simétricas asociadas a un fc_i(L,k) está constituida por fc_i(L,k) repeticiones de una misma subcadena no simétrica de longitud L/fc_i(L,k) y peso k/fc_i(L,k). El coset simétrico al que pertenece dicha cadena es siempre de cardinal L/fc_i(L,k). Por tanto, el número de cadenas simétricas pertenecientes al grupo determinado por fc_i(L,k) coincide con el número de cadenas binarias de longitud L/fc_i(L,k) y peso k/fc_i(L,k) menos el número de subcadenas simétricas de esa longitud y peso (que son las asociadas al factor fc(L/fc_i(L,k),k/fc_i(L,k))). Ahora bien, si existe este factor común, estas subcadenas simétricas corresponden a subcadenas de las cadenas simétricas asociadas a algún factor común de L y k.

La fórmula recursiva anterior se logra expresar de forma no recursiva en muchos casos. Uno de ellos es el siguiente.

Corolario 2.7.1

Si $L = p^{l_1}l_2$ y $k = p^{k_1}k_2$ siendo mcd(l_2, k_2)=1, el número de cadenas binarias simétricas viene dado por $\binom{L/p}{k/p}$.

Demostración Se supone sin pérdida de generalidad que $k_1 < l_1$. Dado que en este caso los k_1 factores comunes de L y k se pueden escribir como $fc_i(L,k) = p^{k_1+1-i}$, i=1,2,...,k_1. Por la propiedad anterior se tiene que $\forall i > 1$

$$p^{l_1-k_1+i-1}l_2 S_{p^{l_1-k_1+i-1}l_2} = \binom{p^{l_1-k_1+i-1}l_2}{p^{i-1}k_2} - \binom{p^{l_1-k_1+i-2}l_2}{p^{i-2}k_2}$$

Se ve por inducción sobre i. Para i=1, $fc_1(L,k) = p^{k_1}$, $p^{l_1-k_1}l_2 S_{p^{l_1-k_1}l_2} = \binom{p^{l_1-k_1}l_2}{k_2}$. Se supone cierto para i-1, o sea

$$p^{l_1-k_1+i-2}l_2 S_{p^{l_1-k_1+i-2}l_2} = \binom{p^{l_1-k_1+i-2}l_2}{p^{i-2}k_2} - \binom{p^{l_1-k_1+i-3}l_2}{p^{i-3}k_2}$$

Se demuestra para i,

$$p^{l_1-k_1+i-1}l_2 S_{p^{l_1-k_1+i-1}l_2} = \binom{p^{l_1-k_1+i-1}l_2}{p^{i-1}k_2} - \sum_{j=1}^{i-1} p^{l_1-k_1+j-1}l_2 S_{p^{l_1-k_1+j-1}l_2} = \binom{p^{l_1-k_1+i-1}l_2}{p^{i-1}k_2} - \binom{p^{l_1-k_1+i-2}l_2}{p^{i-2}k_2}$$

De donde, en general se tiene que el número de cadenas binarias simétricas viene dado por

$$\sum_{i=1}^{k_1} p^{l_1-k_1+i-1}l_2 S_{p^{l_1-k_1+i-1}l_2} =$$
$$\binom{p^{l_1-k_1}l_2}{k_2} + \sum_{i=2}^{k_1} \binom{p^{l_1-k_1+i-1}l_2}{p^{i-1}k_2} - \binom{p^{l_1-k_1+i-2}l_2}{p^{i-2}k_2} = \binom{p^{l_1-1}l_2}{p^{k_1-1}k_2}$$

A continuación se presenta un ejemplo de coset simétrico no degenerado para demostrar que este tipo de cosets, a pesar de lo que pudiera parecer en lo que resta de capítulo, no siempre es degenerado.

Ejemplo 2.7.4

Dados L=8 y k=4, los únicos cosets simétricos son E_1=85 ≈ 01010101 y E_2=51≈00110011. A continuación se ve que la degeneración de estos cosets simétricos está determinada por la elección de la función, al contrario de lo que ocurría con los cosets de distancia fija. En particular para la función producto $s_n s_{n+17} s_{n+51} s_{n+52}$, ambos cosets resultan degenerados independientemente del RDRL, tal como se comprueba con el test de presencia de raíces.

$$\begin{vmatrix} 1 & \alpha^{17} & \alpha^{51} & \alpha^{52} \\ 1 & \alpha^{68} & \alpha^{204} & \alpha^{208} \\ 1 & \alpha^{17} & \alpha^{51} & \alpha^{67} \\ 1 & \alpha^{68} & \alpha^{204} & \alpha^{13} \end{vmatrix} = \begin{vmatrix} 1 & \alpha^{17} & \alpha^{51} & \alpha^{52} \\ 1 & \alpha^{34} & \alpha^{102} & \alpha^{104} \\ 1 & \alpha^{17} & \alpha^{51} & \alpha^{67} \\ 1 & \alpha^{34} & \alpha^{102} & \alpha^{134} \end{vmatrix} = 0$$

Por el contrario, con la función producto $s_n s_{n+17} s_{n+18} s_{n+19}$, el coset E_2 resulta ser no degenerado para el RDRL de polinomio característico.

$$\begin{vmatrix} 1 & \alpha^{17} & \alpha^{18} & \alpha^{19} \\ 1 & \alpha^{34} & \alpha^{36} & \alpha^{38} \\ 1 & \alpha^{17} & \alpha^{33} & \alpha^{49} \\ 1 & \alpha^{34} & \alpha^{66} & \alpha^{98} \end{vmatrix} =$$

$$\begin{vmatrix} 1 & \alpha^{17} \\ 1 & \alpha^{34} \end{vmatrix} [\alpha^{19}(\alpha^{36}+\alpha^{66}) + \alpha^{38}(\alpha^{18}+\alpha^{33}) + \alpha^{49}(\alpha^{36}+\alpha^{66}) + \alpha^{98}(\alpha^{18}+\alpha^{33})] =$$
$$\alpha^{17}(1+\alpha^{17})\,[(\alpha^{19}+\alpha^{49})(\alpha^{36}+\alpha^{66}) + (\alpha^{38}+\alpha^{98})(\alpha^{18}+\alpha^{33})] =$$
$$\alpha^{17}(1+\alpha^{17})\,[\alpha^{55}+\alpha^{115}+\alpha^{56}+\alpha^{71}+\alpha^{116}+\alpha^{131}] =$$
$$\alpha^{17}(1+\alpha^{17})\,[1+\alpha^3+\alpha^4+\alpha^7] \neq 0$$

El primer producto escogido presenta algunas de las características que debe cumplir la función para poder garantizar la degeneración de los cosets simétricos. Este ejemplo será retomado en el próximo apartado para señalar cuáles son las características mencionadas.

2.8. Cotas Superiores

En este apartado se den algunos resultados sobre la degeneración de los cosets simétricos bajo ciertas circunstancias. Concretamente el próximo resultado impone unas condiciones muy amplias para lograr la degeneración de un coset simétrico.

Teorema 2.8.1

Dada una función no lineal de orden k con un único término de orden máximo $s_n s_{n+t_1} \cdots s_{n+t_{k-1}}$ aplicada sobre un RDRL de longitud L, si k — L y $\exists i \in \{1,2,\ldots,k\text{-}1\}$ tal que $\alpha^{t_i} \in GF(2^{L/k})$, entonces el coset simétrico asociado a fc(L,k)= k es degenerado.

Demostración En primer lugar se demuestra que para el factor común fc(L,k)= k existe un único coset simétrico porque sólo existe un único coset no simétrico de cardinal L/k y peso uno. Este coset E coincide con el llamado coset representante, por lo que se puede expresar como $E = 2^0 + 2^{L/k} + 2^{2L/k} + \cdots + 2^{(k-1)L/k}$, de ahí que el determinante A_E sea de la siguiente forma

$$A_E = \begin{vmatrix} 1 & \cdots & \alpha^{t_{i-1}} & \alpha^{t_i} & \alpha^{t_{i+1}} & \cdots & \alpha^{t_{k-1}} \\ 1 & \cdots & \alpha^{t_{i-1}2^{L/k}} & \alpha^{t_i 2^{L/k}} & \alpha^{t_{i+1}2^{L/k}} & \cdots & \alpha^{t_{k-1}2^{L/k}} \\ \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & \cdots & \alpha^{t_{i-1}2^{(k-1)L/k}} & \alpha^{t_i 2^{(k-1)L/k}} & \alpha^{t_{i+1}2^{(k-1)L/k}} & \cdots & \alpha^{t_{k-1}2^{(k-1)L/k}} \end{vmatrix}$$

Según las hipótesis, $\alpha^{t_i} \in GF(2^{L/k})$. Por tanto $\alpha^{t_i} \equiv \alpha^{t_i 2^{L/k}} \equiv \alpha^{t_i 2^{2L/k}} \equiv \cdots \equiv \alpha^{t_i 2^{(k-1)L/k}}$ en GF(2^L), por ser GF($2^{L/k}$) subcuerpo de GF(2^L). En consecuencia A_E contiene dos columnas linealmente dependientes sobre GF(2^L) y por tanto se anula.

Observación 2.8.1

Según las hipótesis planteadas se ha exigido que la primera fase del término de orden máximo sea s_n, es decir, t_0=0. Esta condición no supone ninguna restricción por la equivalencia cíclica de las secuencias producidas por desplazamientos de fase distintos. Esto significa que para adaptar una función cualquiera de término de orden máximo $s_{n+t_0} s_{n+t_1} \cdots s_{n+t_{k-1}}$ a la hipótesis del teorema, sólo hay que considerar un desplazamiento de fase -t_0 quedando dicho término de la forma $s_n s_{n+t_1-t_0} \cdots s_{n+t_{k-1}-t_0}$.

Ejemplo 2.8.1

Dados L=8, k=4 y la función producto $s_n s_{n+17} s_{n+18} s_{n+85}$. El coset simétrico E=85=$2^0 + 2^2 + 2^4 + 2^6 \approx 01010101$ es el coset representante correspondiente al factor común 4. En este caso, como $\alpha^{85} \in GF(2^2)$ se tiene que el coset E es degenerado.

Del teorema anterior se deduce directamente una cota superior a la complejidad lineal de las secuencias producidas bajo esas hipótesis.

Corolario 2.8.1

Dada una función no lineal de orden k con un único término de orden máximo $s_n s_{n+t_1} \cdots s_{n+t_{k-1}}$ aplicada sobre un RDRL de longitud L, si k — L y $\exists i \in \{1,2,...,k\text{-}1\}$ tal que $\alpha^{t_i} \in GF(2^{L/k})$, entonces la complejidad lineal de las secuencias generadas está acotada superiormente por

$$\left[\sum_{i=1}^{k} \binom{L}{i}\right] - \frac{L}{k}$$

La cota anterior resulta de restar a la cota superior dada por Key [85], el cardinal del coset simétrico asociado a fc(L,k)=k.

En el próximo resultado, a base de imponer mayores restricciones sobre la función, se concluye la degeneración de un número mayor de cosets.

Teorema 2.8.2

Dada una función no lineal de orden k con un único término de orden máximo $s_{n+t_0} s_{n+t_1} \cdots s_{n+t_{k-1}}$ aplicada sobre un RDRL de longitud L, si $\exists\ fc(L,k) : \alpha^{t_i} \in GF(2^{L/fc(L,k)})\ \forall i = 0, 1, ..., k/fc(L,k)$, entonces todos los cosets simétricos determinados por fc(L,k) son degenerados.

Demostración Para cualquiera de los cosets simétricos E determinados por fc(L,k), utilizando la hipótesis $\alpha^{t_i} \in GF(2^{L/fc(L,k)})$, el determinante A_E queda de la siguiente forma

$$A_E =$$

$$\begin{vmatrix}
\alpha^{t_0 2^{e_0}} & \cdot & \alpha^{t_{\frac{k}{fc(L,k)}} 2^{e_0}} & \alpha^{t_{\frac{k}{fc(L,k)}+1} 2^{e_0}} & \cdot & \alpha^{t_{k-1} 2^{e_0}} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\alpha^{t_0 2^{e_{\frac{k}{fc(L,k)}-1}}} & \cdot & \alpha^{t_{\frac{k}{fc(L,k)}} 2^{e_{\frac{k}{fc(L,k)}-1}}} & \alpha^{t_{\frac{k}{fc(L,k)}+1} 2^{e_{\frac{k}{fc(L,k)}-1}}} & \cdot & \alpha^{t_{k-1} 2^{e_{\frac{k}{fc(L,k)}-1}}} \\
\alpha^{t_0 2^{e_0}} & \cdot & \alpha^{t_{\frac{k}{fc(L,k)}} 2^{e_0}} & \alpha^{t_{\frac{k}{fc(L,k)}+1} 2^{e_{\frac{k}{fc(L,k)}}}} & \cdot & \alpha^{t_{k-1} 2^{e_{\frac{k}{fc(L,k)}}}} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\alpha^{t_0 2^{e_{\frac{k}{fc(L,k)}-1}}} & \cdot & \alpha^{t_{\frac{k}{fc(L,k)}} 2^{e_{\frac{k}{fc(L,k)}-1}}} & \alpha^{t_{\frac{k}{fc(L,k)}+1} 2^{e_{\frac{2k}{fc(L,k)}-1}}} & \cdot & \alpha^{t_{k-1} 2^{e_{\frac{2k}{fc(L,k)}-1}}} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\alpha^{t_0 2^{e_{\frac{k}{fc(L,k)}-1}}} & \cdot & \alpha^{t_{\frac{k}{fc(L,k)}} 2^{e_{\frac{k}{fc(L,k)}-1}}} & \alpha^{t_{\frac{k}{fc(L,k)}} 2^{e_{k-1}}} & \cdot & \alpha^{t_{k-1} 2^{e_{k-1}}}
\end{vmatrix}$$

Por tanto al desarrollar por las $k - \frac{k}{fc(L,k)} - 1$ últimas columnas, el determinante A_E puede expresarse como combinación lineal de adjuntos de orden $\frac{k}{fc(L,k)} + 1$, todos ellos con dos filas idénticas. Luego $A_E = 0$.

Observación 2.8.2

En las hipótesis del teorema anterior no se ha exigido que las fases t_i sean consideradas en orden creciente, por lo que la condición de la hipótesis corresponde realmente a la existencia de $\frac{k}{fc(L,k)} + 1$ fases t_i cualesquiera del producto que verifiquen dicha propiedad.

Ejemplo 2.8.2

En un ejemplo anterior se demostró la degeneración del coset simétrico E=85 para L=8, k=4 y función producto $s_n s_{n+17} s_{n+51} s_{n+52}$. Según este último teorema, esa degeneración está probada por ser $1, \alpha^{17}$ y α^{51} elementos de GF(2^4).

También del teorema anterior se deduce una cota superior a la complejidad lineal de las secuencias producidas, pero en este caso la cota obtenida es considerablemente inferior. De hecho en los dos próximos resultados se dan condiciones para la degeneración de todos los cosets simétricos de peso k.

Corolario 2.8.2

Dada una función no lineal de orden k con un único término de orden máximo $s_{n+t_0} s_{n+t_1} \cdots s_{n+t_{k-1}}$ aplicada sobre un RDRL de longitud L, si $\forall j, \forall$ i= 0, 1,..., k/$fc_j(L,k)$: $\alpha^{t_i} \in GF(2^{L/fc_j(L,k)})$, entonces la complejidad lineal

de las secuencias generadas está acotada superiormente por

$$\sum_{n=1}^{k}\binom{L}{n} - \sum_{j}\frac{L}{fc_j(L,k)}S_{\frac{L}{fc_j(L,k)}}$$

Demostración Según las hipótesis, todos los cosets simétricos son degenerados. Dado que por la proposición 2.3 el número de cadenas simétricos es $\sum_j \frac{L}{fc_j(L,k)}S_{\frac{L}{fc_j(L,k)}}$, se concluye la tesis del corolario.

Ejemplo 2.8.3

Para L=8, k=4, para cualquier filtrado cuyo único término de orden máximo sea de la forma $s_n s_{n+85} s_{n+102} s_{n+t_3}$, siendo $0 \leq t_3 < 2^8 - 1$ distinto de 0, 85 y 102, la complejidad lineal global está acotada superiormente por 156 ya que los dos cosets simétricos de cardinales 2 y 4 son degenerados.

Para algunos valores de L y k las condiciones de los últimos resultados se cumplen con más facilidad, pudiéndose de esta forma relajar la hipótesis.

Corolario 2.8.3

Dados $L=p^{l_1}l_2$ y $k=p^{k_1}k_2$ siendo p un número primo y $mcd(l_2,k_2)=1$ y una función no lineal de orden k con un único término de orden máximo $s_{n+t_0}s_{n+t_1}\cdots s_{n+t_{k-1}}$ aplicada sobre un RDRL de longitud L, si $\forall i = 0,1,...,k/p:\ \alpha^{t_i} \in GF(2^{L/p})$, entonces la complejidad lineal de las secuencias generadas está acotada superiormente por

$$\sum_{n=1}^{k}\binom{L}{n} - \binom{L/p}{k/p}$$

Demostración Para cada uno de los cosets simétricos E determinados por cualquier fc(L,k)=p^j, el determinante A_E queda de la siguiente forma

$$A_E = \begin{vmatrix}
\alpha^{t_0 2^{e_0}} & \alpha^{t_1 2^{e_0}} & \cdots & \alpha^{t_{\frac{k}{p}} 2^{e_0}} & \alpha^{t_{\frac{k}{p}+1} 2^{e_0}} & \cdots & \alpha^{t_{k-1} 2^{e_0}} \\
\alpha^{t_0 2^{e_1}} & \alpha^{t_1 2^{e_1}} & \cdots & \alpha^{t_{\frac{k}{p}} 2^{e_1}} & \alpha^{t_{\frac{k}{p}+1} 2^{e_1}} & \cdots & \alpha^{t_{k-1} 2^{e_1}} \\
\cdot & \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\
\alpha^{t_0 2^{e_{\frac{k}{p}-1}}} & \alpha^{t_1 2^{e_{\frac{k}{p}-1}}} & \cdots & \alpha^{t_{\frac{k}{p}} 2^{e_{\frac{k}{p}-1}}} & \alpha^{t_{\frac{k}{p}+1} 2^{e_{\frac{k}{p}-1}}} & \cdots & \alpha^{t_{k-1} 2^{e_{\frac{k}{p}-1}}} \\
\alpha^{t_0 2^{e_0}} & \alpha^{t_1 2^{e_0}} & \cdots & \alpha^{t_{\frac{k}{p}} 2^{e_0}} & \alpha^{t_{\frac{k}{p}+1} 2^{e_{\frac{k}{p}}}} & \cdots & \alpha^{t_{k-1} 2^{e_{\frac{k}{p}}}} \\
\cdot & \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\
\alpha^{t_0 2^{e_{\frac{k}{p}-1}}} & \alpha^{t_1 2^{e_{\frac{k}{p}-1}}} & \cdots & \alpha^{t_{\frac{k}{p}} 2^{e_{\frac{k}{p}-1}}} & \alpha^{t_{\frac{k}{p}+1} 2^{e_{k-1}}} & \cdots & \alpha^{t_{k-1} 2^{e_{k-1}}}
\end{vmatrix}$$

Utilizando la hipótesis, $\forall i = 0, 1, ..., k/p : \ \alpha^{t_i} \in GF(2^{L/p})$, y desarrollando A_E por las k-1+k/p últimas columnas se concluye que A_E se anula porque cada uno de los adjuntos de la expresión obtenida tiene dos filas iguales. Por tanto, el valor de la complejidad lineal se ve reducida en el número de cadenas simétricas que, en virtud del corolario 2.6, es $\binom{L/p}{k/p}$.

Las hipótesis del corolario 2.9 son mucho menos restrictivas que las del corolario 2.8. De hecho existen muchas funciones que verifican esas condiciones. Esto queda reflejado en el siguiente corolario.

Corolario 2.8.4

Cumpliéndose las hipótesis del corolario anterior existe un número de funciones mayor que

$$\binom{L/p}{1+k/p}\binom{L-(1+k/p)}{k-(1+k/p)}$$

Demostración Se deduce fácilmente utilizando herramientas de cálculo combinatorio.

Ejemplo 2.8.4

Según esta expresión, para L=8 y k=4 existen más de 20 funciones distintas para las que se tiene asegurada una cota superior a la complejidad lineal global de valor 156.

Mediante los cosets simétricos se han deducido para varios grupos de funciones, unas cotas superiores a la complejidad lineal que son inferiores a la cota dada por Key. Sin embargo, de todo el estudio realizado para estos cosets sin duda el aspecto más práctico se refleja en el siguiente apartado. En él se reúnen conclusiones de todo el capítulo en forma de sugerencias para escoger la función no lineal de manera que se garantice la mayor complejidad lineal posible.

2.9. Sugerencias para la Elección del Filtrado

Las sugerencias señaladas con S1, S2,... constituyen la conclusión de todo lo expuesto en este capítulo. Se recomienda cumplir alguna de estas sugerencias a la hora de elegir un determinado tipo de filtrado no lineal. Es imposible

verificarlas todas a la vez ya que la mayoría corresponden a condiciones disjuntas.

En primer lugar, en relación con los resultados demostrados para los cosets de distancia fija, se obtiene la siguiente,

S1. Tomar un valor de L primo y la función no lineal con un único término de orden máximo.

De esa forma, la no degeneración garantizada de los cosets de distancia fija proporciona una cota inferior de la complejidad lineal que alcanza su valor máximo cuando L es primo.

Las siguientes observaciones se concluyen a partir de lo visto para los productos de fases $2^{((d))}$-distantes.

S2. Escoger la función no lineal con un único término de orden máximo que sea un producto de k fases $2^{((d))}$-distantes, con k$\cong$L/2.

El filtrado no lineal obtenido tiene una complejidad lineal global superior a $\binom{L}{k}$, y el mayor valor que puede tomar esta expresión viene dado cuando k$\cong L/2$.

Si no se quiere utilizar una función con un único término de orden máximo, se tiene como opción la siguiente función.

S3. Escoger la función no lineal de manera que su término de orden máximo sea una combinación lineal de N productos de k fases $2^{((d))}$-distantes, con k$\cong L/2$.

La cota inferior a la complejidad lineal garantizada para ese caso es $\binom{L}{k} - (N-1)$, cuyo máximo valor viene dado cuando k$\cong L/2$. Si además se tiene que N$\leq L$ y L es primo, la cota mejora ligeramente quedando de la forma $\binom{L}{k}$.

Para evitar posibles degeneraciones de los cosets simétricos se propone la siguiente sugerencia.

S4. Escoger L y k números primos entre sí.

De esta forma no existen cosets simétricos, y por tanto no pueden hacer disminuir con su degeneración el valor de la complejidad lineal.

La siguiente indicación no supone grandes restricciones pero a cambio tampoco proporciona un gran control sobre la complejidad lineal.

S5. Si k — L, tomar el término de mayor orden de la forma $s_n s_{n+t_1} \cdots s_{n+t_{k-1}}$ tal que $\forall i \in \{1, 2, ..., k-1\} : \ \alpha^{t_i} \notin GF(2^{L/k})$.

En realidad con esta condición sólo se intenta evitar la degeneración de

un coset simétrico, lo que no supone una gran disminución del valor de la complejidad lineal. La siguiente sí intenta prevenir una disminución mayor.

S6. Escoger el término de mayor orden de la forma $s_{n+t_0}s_{n+t_1}\cdots s_{n+t_{k-1}}$ tal que $\forall fc_j(L,k)$, existen k-$\frac{k}{p}$ etapas t_i tales que $\alpha^{t_i} \notin GF(2^{L/fc_j(L,k)})$.

Pues en caso contrario degenerarían todos los cosets simétricos. Si L y k toman unos determinados valores, la condición anterior se puede relajar quedando de la siguiente forma.

S7. Si L=$p^{l_1}l_2$ y k=$p^{k_1}k_2$ con p un número primo y mcd(l_2, k_2)=1, tomar el término de orden máximo de la forma $s_{n+t_0}s_{n+t_1}\cdots s_{n+t_{k-1}}$ tal que exista al menos k-k/p valores $t_i : \alpha^{t_i} \notin GF(2^{L/p})$.

Ya que en caso contrario el valor de la complejidad lineal disminuiría en una cantidad $\binom{L/p}{k/p}$.

En el próximo capítulo se retoman los dos primeros tipos de cosets tratados en este capítulo. En particular se diseñan dos algoritmos que manejan ambos conceptos y que sirven para mejorar la cota inferior general obtenida en el principal resultado de este capítulo, el de la no degeneración de los cosets de distancia fija.

Capítulo 3

Algoritmos de Cálculo de Cotas Inferiores

En este capítulo se demuestra de forma práctica la segunda idea fundamental de este trabajo, la idea (I2) descrita en el capítulo anterior. Para ello se presentan dos algoritmos, algoritmos 1 y 2, que permiten calcular cotas inferiores a la complejidad lineal de cualquier filtrado no lineal siempre y cuando éste tenga un único término de orden máximo.

Ambos algoritmos tienen como únicas entradas L (longitud del RDRL) y k (orden de la función). En consecuencia los valores que se obtienen con ellos son válidos para una amplia clase de filtrados.

Aunque los dos algoritmos tienen en común que se basan en el análisis de la degeneración simultánea de un grupo de cosets, el algoritmo 2, según se verá, produce unos valores de cota muy superiores a los producidos por el algoritmo 1. Esto se debe a que el grupo de cosets analizados con el algoritmo 2 es mucho mayor que el de los analizados con el 1.

En concreto los cosets analizados en ambos casos son de peso k y pertenecen al tipo de cosets presentado en el apartado 2.6, los quasicosets d-f. Las operaciones que se realizan en ambos algoritmos tienen como objeto comprobar las hipótesis del Teorema 2.3, por lo que este resultado se puede considerar la base teórica de los algoritmos.

En el próximo apartado se introduce la técnica común que se utiliza en ambos algoritmos. Aunque la idea que hay detrás es muy sencilla, gracias a esta técnica se evita el cálculo de determinantes en cuerpos finitos sustituyéndolo por manipulaciones de cadenas binarias, lo que representa un considerable ahorro en computación.

3.1. Paso de Determinante a Cadena Binaria

En el primer apartado del capítulo anterior quedó de manifiesto la equivalencia entre las cuatro posibles representaciones de los cosets. Entre ellas retomamos ahora las caracterizaciones C2 y C4, referentes respectivamente a cadenas binarias y sistemas lineales homogéneos. En el resto del capítulo se manejarán representaciones binarias idénticas en su forma a la C2, pero distintas en su definición. El origen de esta nueva representación es un sistema lineal homogéneo de la forma

$$\begin{cases} 0 = & d_0\alpha^{t_0 2^{e_0}} + d_1\alpha^{t_1 2^{e_0}} + \cdots + d_{k-1}\alpha^{t_{k-1} 2^{e_0}} \\ 0 = & d_0\alpha^{t_0 2^{e_1}} + d_1\alpha^{t_1 2^{e_1}} + \cdots + d_{k-1}\alpha^{t_{k-1} 2^{e_1}} \\ & \vdots \\ 0 = & d_0\alpha^{t_0 2^{e_{m-1}}} + d_1\alpha^{t_1 2^{e_{m-1}}} + \cdots + d_{k-1}\alpha^{t_{k-1} 2^{e_{m-1}}} \end{cases} \tag{3.1}$$

donde α es un elemento primitivo de GF(2^L), $t_i \neq t_j \ \forall i \neq j$, $0 \leq e_0 < e_1 < \cdots < e_{m-1} < L$ y $d_j \in GF(2^L) \ \forall j$.

A partir de él se construye una cadena binaria de longitud L, manera que la presencia de la ecuación i-ésima

$$0 = d_0\alpha^{t_0 2^{e_i}} + d_1\alpha^{t_1 2^{e_i}} + \cdots + d_{k-1}\alpha^{t_{k-1} 2^{e_i}}$$

implica la presencia en la cadena de un uno en la posición e_i (contada desde la derecha).

Observación 3.1.1

Hay que destacar que aunque un bit unitario en la cadena binaria significa que se da la igualdad de la ecuación correspondiente en el sistema asociado, un bit nulo por el contrario no significa que se dé la desigualdad de la ecuación correspondiente. Sólo significa que no se sabe con certeza si se da o no dicha igualdad.

De esta descripción se concluye que la cadena binaria así definida para el sistema descrito en el apartado 2.1 coincide totalmente con la cadena correspondiente a la caracterización C2 del mismo apartado. Sin embargo si sobre el sistema lineal asociado a un coset de peso k se realiza alguna operación que incremente o disminuya su número de ecuaciones, la cadena binaria asociada al sistema resultante según la descripción anterior no coincide con ninguna

cadena asociada a ningún coset de peso k. Por tanto esta nueva representación es esencialmente distinta de la C2.

Dada la forma del sistema lineal homogéneo anterior, la operación de elevar al cuadrado todas las ecuaciones se traduce inmediatamente en una rotación cíclica a izquierda de la cadena binaria asociada. La equivalencia entre ambas operaciones será de gran utilidad en los próximos apartados.

Ejemplo 3.1.1

Dados L=6 y k=3 se observa la equivalencia de ambas operaciones

$$\begin{cases} 0 = & d_0\alpha^{t_0} + d_1\alpha^{t_1} + d_2\alpha^{t_2} \\ 0 = & d_0\alpha^{t_0 2^2} + d_1\alpha^{t_1 2^2} + d_2\alpha^{t_2 2^2} \\ 0 = & d_0\alpha^{t_0 2^3} + d_1\alpha^{t_1 2^3} + d_2\alpha^{t_2 2^3} \\ 0 = & d_0\alpha^{t_0 2^5} + d_1\alpha^{t_1 2^5} + d_2\alpha^{t_2 2^5} \end{cases} \Longrightarrow \begin{cases} 0 = & d_0^2\alpha^{t_0 2} + d_1^2\alpha^{t_1 2} + d_2^2\alpha^{t_2 2} \\ 0 = & d_0^2\alpha^{t_0 2^3} + d_1^2\alpha^{t_1 2^3} + d_2^2\alpha^{t_2 2^3} \\ 0 = & d_0^2\alpha^{t_0 2^4} + d_1^2\alpha^{t_1 2^4} + d_2^2\alpha^{t_2 2^4} \\ 0 = & d_0^2\alpha^{t_0} + d_1^2\alpha^{t_1} + d_2^2\alpha^{t_2} \end{cases}$$

$$101101 \Longrightarrow 011011$$

En el próximo apartado se analizan una por una las operaciones básicas que se llevan a cabo en los algoritmos. Se explica la interpretación de cada una de las tres operaciones lógicas AND, OR-exclusiva (denotada como XOR) y OR, según la representación en sistemas de ecuaciones presentada en este apartado.

3.2. Interpretación de las Operaciones Lógicas

3.2.1. Operación AND

La operación AND entre dos cadenas binarias cualesquiera de igual longitud implica la construcción de una nueva cadena binaria de la misma longitud, donde cada bit unitario proviene de la coincidencia de dos bits unitarios en las dos cadenas de partida.

Por tanto según lo dicho en el apartado anterior, esta operación se puede interpretar de la siguiente forma. A partir de dos sistemas lineales homogéneos del tipo (2) asociados a las dos cadenas binarias de partida se construye un nuevo sistema lineal homogéneo con las ecuaciones que estén presentes a la vez en ambos sistemas. De esta forma dicha operación permite comprobar la presencia de ecuaciones en un sistema de este tipo. Para ello sólo hay que

realizar la AND entre la cadena binaria asociada al sistema que se quiere comprobar y la cadena binaria asociada a las ecuaciones que se buscan. Si el resultado coincide con esta última cadena, la comprobación resulta satisfactoria.

Esta operación de comprobación se llevará a cabo en los dos algoritmos que se presentan en este capítulo.

Dicha comprobación resulta especialmente útil en los dos siguientes casos. En primer lugar, la presencia de unas determinadas ecuaciones dentro de un sistema puede implicar que el sistema sólo tenga la solución trivial. Esto es así por ejemplo cuando las ecuaciones anteriormente mencionadas forman un subsistema del sistema general con solución trivial únicamente. Concretamente los subsistemas que se buscan en los algoritmos son los sistemas asociados a los cosets de distancia fija definidos en el capítulo anterior. Allí se demostró que estos sistemas sólo tienen la solución trivial, por lo que se pueden utilizar para aplicar el razonamiento anterior.

En segundo lugar, dicha operación se puede utilizar de manera obvia para despejar dudas acerca de si un coset ha sido ya analizado o no. Para ello sólo hay que hacer las operaciones AND entre el coset en cuestión y cada uno de los cosets analizados. Dependiendo de si el resultado coincide o no con el segundo operando se tiene una respuesta afirmativa o negativa. Esta segunda aplicación de la operación AND se utilizará en el algoritmo 2.

3.2.2. Operación XOR

La operación XOR de dos cadenas binarias cualesquiera de igual longitud implica la construcción de una nueva cadena binaria de la misma longitud, donde cada bit nulo proviene de la coincidencia de dos bits en las dos cadenas de partida.

De ahí se puede deducir la siguiente interpretación mediante sistemas de ecuaciones. A partir de los dos sistemas lineales homogéneos asociados a las dos cadenas binarias de partida se construye un nuevo sistema lineal homogéneo con las ecuaciones que pertenezcan exclusivamente a uno de los dos sistemas.

Esta operación permite comprobar no sólo las ecuaciones presentes en el sistema sino también las ausentes. Por ejemplo, si se quiere saber si un sistema lineal homogéneo de la forma (2) es subsistema de otro con una ecuación adicional, se puede utilizar esta operación simplemente comprobando que el sistema asociado al resultado de la operación XOR está formado por dicha

ecuación.

Esta comprobación se utilizará en el algoritmo 2. Concretamente se usa para dilucidar si un coset ha sido previamente estudiado a lo largo del algoritmo. Para ello se utilizan unas 'máscaras' que representan conjuntos de cosets, de manera que al realizar la XOR entre el coset puesto en duda y cada una de estas máscaras, se descubre si dicho coset pertenece o no a alguno de esos conjuntos.

De hecho, dado que las máscaras son cadenas binarias con k-1 unos la operación de comprobación se remite exactamente al ejemplo mencionado.

3.2.3. Operación OR

La operación OR de varias cadenas binarias cualesquiera de igual longitud implica la construcción de una nueva cadena binaria de la misma longitud, donde cada bit nulo proviene de la coincidencia de bits nulos en todas las cadenas de partida.

Esta es de las tres, la operación cuya interpretación mediante sistemas de ecuaciones resulta más clara ya que implica la construcción de un macrosistema formado por la unión de todos los sistemas lineales asociados a las cadenas binarias de partida.

En ambos algoritmos se utiliza esta operación como herramienta básica necesaria para comprobar la idea (I2), es decir, para vigilar la degeneración simultánea de varios cosets. La justificación es como sigue:

Si varios cosets de peso k son simultáneamente degenerados, entonces los respectivos sistemas de k ecuaciones asociados tienen simultáneamente solución no trivial. Por otro lado, si ambos cosets tienen k-1 unos comunes, los respectivos sistemas tienen k-1 ecuaciones comunes. Si además se cumple que la cadena binaria formada por los k-1 unos comunes es un coset de distancia fija de peso k-1, dado que el determinante asociado a este coset es no nulo, y coincide exactamente con el determinante asociado al subsistema de las k-1 ecuaciones comunes, entonces se puede deducir que el macrosistema formado por las k-1 ecuaciones comunes y todas las demás ecuaciones distintas, tiene solución no trivial. De hecho el fundamento teórico de ambos algoritmos consiste en suponer que se da la hipótesis de partida del razonamiento anterior, es decir, que varios cosets son simultáneamente degenerados. De esa forma, al realizar la operación OR entre dichos cosets se construye el mencionado macrosistema con solución no trivial.

3.3. Fundamento Teórico del Algoritmo 1

En este apartado se presenta en primer lugar la explicación teórica del primer algoritmo y a continuación, de forma paralela, las operaciones binarias que se llevan a cabo en él.

3.3.1. Sistemas de Ecuaciones

Para cada uno de los posibles valores d<L/2 tales que mcd(d,L)=1, de forma paralela se hace lo siguiente. Se consideran los L-k-1 sistemas lineales distintos del tipo (2) con k ecuaciones y $e_i \equiv i \cdot d \pmod{L}$ $\forall i \in \{0, 1, ..., k-2\}$, y se supone que tienen solución no trivial. Se utiliza un contador m decreciente cuyo primer valor es L-k-2.

Con dichos sistemas se forman todos los posibles grupos tomados de m en m y se construyen los macrosistemas correspondientes a las uniones de cada uno de estos grupos. Dado que todos los sistemas de partida tienen en común un subsistema de k-1 ecuaciones cuya única solución es la trivial, se tiene que el macrosistema tiene obligatoriamente solución no trivial. Se descubre si entre las ecuaciones de cada macrosistema se encuentran las k ecuaciones correspondientes al sistema asociado a algún coset de distancia fija, ya que en este caso se deduce que el macrosistema sólo puede tener la solución trivial, llegándose a un absurdo con la hipótesis de partida. Por tanto, en cuanto se consigue algún valor de m para el que no se cumple lo anterior, se concluye que aunque puede ocurrir que m sistemas del tipo mencionado tengan simultáneamente solución no trivial, es imposible que m+1 sistemas la tengan. En este caso se deduce que entre todos los sistemas considerados, como máximo pueden existir m que tengan simultáneamente solución no trivial, lo que se utiliza para incrementar la cota inferior de la complejidad lineal.

3.3.2. Operaciones Binarias

Paralelamente para cada uno de los posibles valores d¡L/2 tales que mcd(d,L)=1 se hace lo siguiente. Se generan todos los (k-1)-ésimos quasi cosets d-f distintos obtenidos a partir del coset de distancia fija d. Se utiliza un contador m decreciente cuyo primer valor es L-k-2. Con dichos cosets se forman todos los posibles grupos de m cosets y se hace la OR de cada grupo. Se descubre mediante una operación AND si la cadena binaria resultante de

la operación OR contiene algún coset de distancia fija. En cuanto aparece algún valor m para el que no se cumple lo anterior se concluye que aunque todos los cosets de algún grupo de m cosets pueden ser simultáneamente degenerados, no existe ningún grupo de m+1 cosets que lo sean. Por tanto, se tiene que entre todos los (k-1)-ésimos quasicosets d-f considerados, como máximo pueden existir m simultáneamente degenerados y en consecuencia los demás cosets siempre contribuyen a la complejidad lineal.

3.4. Algoritmo 1

Este primer algoritmo tiene como entradas L (longitud del RDRL) y k (orden del filtrado), y como salida una cota inferior a la complejidad lineal Δ. En primer lugar se aclarará la notación específica usada en el algoritmo, a continuación se da una descripción detallada del algoritmo y por último se presenta un ejemplo ilustrativo de su funcionamiento.

3.4.1. Notación

Los cosets de distancia fija de peso k se denotan mediante CDF(i) (i=1,2,...,N_L) y para representarlos se utiliza la cadena binaria resultante de su definición.

Los k-ésimos quasicosets d-f generados a partir del CDF(i) se denotan mediante $CD_i(j)$ (j=1,2,...,L-k-1).

A la cadena binaria resultante de la operación OR entre los m cosets de un grupo cualquiera de $CD_i(j)$ se le denota como VOR. A lo largo del algoritmo se realiza dicha operación con cada uno de los $\binom{L-k-1}{m}$ posibles grupos y siempre se almacena el resultado en la misma variable VOR, por tanto en cada momento sólo se conserva el resultado de la última operación realizada.

Observación 3.4.1

En el algoritmo, cuando se pide hacer la AND entre alguna cadena binaria y algún CDF(l) se entiende que se deben hacer las L operaciones AND entre la cadena y cada uno de los elementos del coset CDF(l).

Δ_i denota el incremento de la cota Δ calculado gracias al grupo de quasicosets d-f generado a partir de CDF(i).

3.4.2. Algoritmo

ENTRADA: L, k (2<k<L-2)

Paso 1:
Calcular los N_L valores d¡L/2 tales que mcd(d,L)=1.
Mediante ellos, generar los CDF(i) (i=1,2,...,N_L).
Para cada i$\in \{1, 2, ..., N_L\}$ por separado:
Paso 2:
Generar los CD_i(j) (j=1,2,...,L-k-1).
Hacer m=L-k-2, n=1.
Paso 3:
Si n$\leq \binom{L-k-2}{m}$,
escoger el n-ésimo grupo de m cosets CD_i(j),
hacer la OR de dicho grupo obteniendo VOR y
hacer VL=0, l=1.
En otro caso ir al Paso 5.
Paso 4:
Si l$\leq N_L$,
hacer la AND entre VOR y CDF(l),
si coincide con CDF(l), entonces hacer VL=1, n=n+1 e ir al Paso 3,
en otro caso hacer l=l+1 e ir al Paso 4.
Paso 5:
Si VL=0, entonces $\Delta_i = (L-k-m-1)L$.
En otro caso,
si m>2 entonces hacer m=m-1 e ir al Paso 3,
en otro caso $\Delta_i = (L-k-m)L$.
SALIDA: $\Delta = L \cdot N_L + \sum_{i=1}^{N_L} \Delta_i$.
El primer diagrama de flujo del apéndice A aclara la situación.

3.4.3. Ejemplo Numérico

Para L=9, k=5, d=1, 2 ó 4, $N_9 = 3$ y número de cosets de peso 5=14, los cosets de distancia fija de peso 5 se representan mediante

CDF(1) : 000011111

CDF(2) : 101010101

CDF(3) : 110011001

Se denota como CDF(i)* a un elemento del CDF(i) distinto de su representante.

Para i=1,
$CD_1(1)$=000101111, $CD_1(2)$=001001111, $CD_1(3)$=010001111,
m=2,
n=1,
OR[$CD_1(1)$, $CD_1(2)$]=001101111=VOR, VL=0, l=1,
AND[VOR, CDF(1)]≠CDF(1), l=2,
AND[VOR, CDF(2)]≠CDF(2), l=3,
AND[VOR, CDF(3)*]=CDF(3)* $\Longrightarrow VL = 1$,
n=2,
OR[$CD_1(1)$, $CD_1(3)$]=010101111=VOR, VL=0, l=1,
AND[VOR, CDF(1)]≠CDF(1), l=2,
AND[VOR, CDF(2)*]=CDF(2)* $\Longrightarrow VL = 1$,
n=3,
OR[$CD_1(2)$, $CD_1(3)$]=011001111=VOR, VL=0, l=1,
AND[VOR, CDF(1)]≠CDF(1), l=2,
AND[VOR, CDF(2)]≠CDF(2), l=3,
AND[VOR, CDF(3)*]=CDF(3)* $\Longrightarrow VL = 1$,
n=4$\Longrightarrow \Delta_1$=2·9=18.

Para i=2,
$CD_2(1)$=001010111, $CD_2(2)$=001011101, $CD_2(3)$=001110101,
m=2,
n=1,
OR[$CD_2(1)$, $CD_2(2)$]=001011111=VOR, VL=0, l=1,
AND[VOR, CDF(1)]≠CDF(1)$\Longrightarrow VL = 1$,
n=2,
OR[$CD_2(1)$, $CD_2(3)$]=001110111=VOR, VL=0, l=1,
AND[VOR, CDF(1)]≠CDF(1), l=2,
AND[VOR, CDF(2)]≠CDF(2), l=3,
AND[VOR, CDF(3)*]=CDF(3)* $\Longrightarrow VL = 1$,
n=3,

OR[$CD_2(2)$, $CD_2(3)$]=001111101=VOR, VL=0, l=1,
AND[VOR, CDF(1)*]=CDF(1)* $\Longrightarrow VL = 1$,
n=4$\Longrightarrow$ Δ_2=2·9=18.

Para i=3,
$CD_3(1)$=100011011, $CD_3(2)$=100011101, $CD_3(3)$=100111001,
m=2,
n=1,
OR[$CD_3(1)$, $CD_3(2)$]=100011111=VOR, VL=0, l=1,
AND[VOR, CDF(1)]=CDF(1)$\Longrightarrow VL = 1$,
n=2,
OR[$CD_3(1)$, $CD_3(3)$]=100111011=VOR, VL=0, l=1,
AND[VOR, CDF(1)]$\neq$CDF(1), l=2
AND[VOR, CDF(2)*]=CDF(2)* $\Longrightarrow VL = 1$,
n=3,
OR[$CD_3(2)$, $CD_3(3)$]=100111101=VOR, VL=0, l=1,
AND[VOR, CDF(1)]$\neq$CDF(1), l=2
AND[VOR, CDF(2)]$\neq$CDF(2), l=3
AND[VOR, CDF(3)*]=CDF(3)* $\Longrightarrow VL = 1$,
n=4$\Longrightarrow$ Δ_3=2·9=18.

$\Delta = N_9 \cdot 9 + \Delta_1 + \Delta_2 + \Delta_3 = 27 + 3 \cdot 18 = 81$.

Para este ejemplo el algoritmo ha servido para demostrar que de los 14 cosets de peso 5, 9 son no degenerados. Para obtener este mismo valor de cota con el 'test de presencia de raíces', para cada función y polinomio escogidos habría sido necesario calcular en el cuerpo finito GF(2^9) al menos 9 determinantes de orden 5. Más aún, el resultado obtenido mediante este método es válido para cualquier función de orden 5 y polinomio de grado 9, lo que lo convierte en un método de mayor aplicación.

3.5. Fundamento Teórico del Algoritmo 2

A continuación se presentan en un primer apartado las diferencias en la explicación teórica mediante sistemas de ecuaciones para el algoritmo 2 con respecto al anterior y en un segundo apartado de forma paralela las nuevas operaciones binarias que se requieren en este algoritmo.

3.5.1. Sistemas de Ecuaciones

En este caso hay que actuar de forma secuencial de manera que los cálculos llevados a cabo para cada d menor que L/2 y primo con L se realicen uno a continuación de otro. Para cada $i_0 \in \{1, 2, ..., k-1\}$ se considera un número mpri de sistemas lineales del tipo (2) distintos con k ecuaciones y $e_i \equiv i \cdot d \ (mod\ L)\ \forall i \neq i_0$. El cálculo secuencial se debe a que en este segundo algoritmo podría aparecer el mismo sistema de ecuaciones para dos valores d distintos. Por tanto en este algoritmo es prioritaria la eliminación de los sistemas de ecuaciones que hayan sido previamente considerados. El número de sistemas resultante después de dicha eliminación es el denotado como mpri. Se utiliza un contador m decreciente cuyo primer valor es este valor mpri. El resto de operaciones coincide con las realizadas en el algoritmo anterior.

3.5.2. Operaciones Binarias

De forma secuencial, para cada uno de los posibles valores d< L/2 primos con L y para cada $i_0 \in \{1, 2, ..., k-1\}$ se hace lo siguiente. Se genera el grupo de los i_0-ésimos quasicosets d-f distintos obtenidos a partir del coset de distancia fija d. Se descubren mediante operaciones AND con todos los cosets de distancia fija aquellos cosets del grupo anterior que no es necesario estudiar por saberse ya que son no degenerados, y se eliminan del grupo. Se descubren mediante operaciones XOR con las máscaras de los cosets analizados hasta el momento, aquellos cosets del grupo que ya hayan sido estudiados y se eliminan del grupo. El número de cosets que queden en el grupo después de estas eliminaciones se denota como mpri. Sobre estos cosets se realizan las mismas operaciones que en el algoritmo anterior.

3.6. Algoritmo 2

Siguiendo con el mismo esquema usado para el anterior algoritmo, primero se da la notación, luego una descripción completa del algoritmo y finalmente un ejemplo numérico.

3.6.1. Notación

El grupo de j-ésimos quasicosets d-f generados a partir del CDF(i) se almacena en CD. En esta variable no se contemplan los índices i y j porque

de hecho, al igual que ocurría con la variable VOR del algoritmo anterior, no es necesario guardar los grupos ya estudiados. Para su control se utilizan unas máscaras, lo que representa un ahorro en memoria.

CDM(i,j) denota la cadena binaria de longitud L y k-1 unos que sirve de máscara del grupo CD correspondiente a los índices i y j. Corresponde exactamente a la AND de todos los elementos del grupo. Además, dada la definición de j-ésimo quasicoset d-f, se tiene que CDM(i,j) contiene todos los unos de CDF(i) salvo el j-ésimo. A lo largo del algoritmo no sólo se eliminan los cosets de CD que hayan sido previamente estudiados sino que también se aprovechan los grupos de cosets CD que no hayan producido ningún aumento en la cota Δ ya que se elimina su correspondiente máscara CDM(i,j).

3.6.2. Algoritmo

ENTRADA: L,k (2<k<L-2)
Paso 1:
Calcular los N_L valores d<L/2 tales que mcd(d,L)=1.
Mediante ellos generar los CDF(i) (i=1,2,...,N_L).
Para cada i=1,2,...,N_L :
Para cada j=1,2,...,k-1 :
Paso 2:
Generar CDM(i,j).
Generar CD.
Hacer m=L-k.
Para cada l=1,2,...,N_L :
Paso 3:
Para cada coset de CD hacer la AND entre dicho coset y CDF(l).
Si coincide con CDF(l), entonces eliminar dicho coset de CD y
hacer m=m-1.
Para cada o=1,2,...,i-1, p=1,2,...,k-1 y
para cada(o,p)=(i,1),(i,2),...,(i,j-1):
Paso 4:
Para cada coset de CD hacer la XOR entre dicho coset y CDM(o,p).
Si la cadena resultante tiene peso de Hamming igual a 1, entonces eliminar dicho coset de CD y hacer m=m-1.

Paso 5:
Hacer mpri=m
Paso 6:
Hacer n=1 y VL=0.
Mientras n$\leq$ ($\begin{matrix} mpri \\ m \end{matrix}$) y m>1:
Paso 7:
Escoger el n-ésimo grupo de m cosets de CD y hacer la OR
de dicho grupo obteniendo VOR.
Hacer VL=0 y l=1.
Paso 8:
Si l$\leq N_L$, entonces hacer la AND entre VOR y CDF(l),
si coincide con CDF(l), entonces hacer VL=1, n=n+1
e ir al Paso 7,
en otro caso, hacer l=l+1 e ir al Paso 8.
Paso 9:
Si VL=1, entonces m=m-1,
si m¡2, entonces hacer $\Delta = \Delta$+(mpri-m)L,
en otro caso, ir al Paso 6.
En otro caso,
si mpri=m, entonces eliminar CDM(i,j),
en otro caso, hacer $\Delta = \Delta + (mpri - m)L$.

SALIDA: Δ.

El segundo diagrama de flujo del apéndice A resulta bastante aclaratorio.

3.6.3. Ejemplo Numérico

Para L=11, k=6, d=1, 2, 3, 4, 5 y N_{11}=5, los cosets de distancia fija d son los siguientes.

CDF(1) : 00000111111

CDF(2) : 10101010101

CDF(3) : 01001011011

CDF(4) : 01100110011

CDF(5) : 11000111001

Con * se indica una rotación del representante.
Se inicializa la cota inferior según $\Delta = 11 \cdot 5 = 55$.

Para i=1, j=1

CDM(1,1) : 00000111101, CD : $\left\{ \begin{array}{l} 00001111101 \\ 00010111101 \\ 00100111101 \\ 01000111101 \\ 10000111101 \end{array} \right\}$,m=5.

Se comprueba el primer coset de CD, 00001111101,
$\forall$l=1,2,3,4 y 5, AND[00001111101,CDF(l)]$\neq$CDF(l),
Se comprueba el segundo coset de CD, 00010111101,
$\forall$l=1,2,3,4 y 5, AND[00010111101,CDF(l)]$\neq$CDF(l),
Se comprueba el tercer coset de CD, 00100111101,
$\forall$l=1,2,3,4 y 5, AND[00100111101,CDF(l)]$\neq$CDF(l),
Se comprueba el cuarto coset de CD, 01000111101,
$\forall$l=1,2,3,4 y 5, AND[01000111101,CDF(l)]$\neq$CDF(l),
Se comprueba el quinto coset de CD, 10000111101,
$\forall$l=1,2,3,4 y 5, AND[10000111101,CDF(l)]$\neq$CDF(l),
mpri=5, n=1,
OR[CD]=11111111101=VOR, VL=0, l=1,
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=2,
m=4, n=1,
OR[CD - 10000111101]=01111111101=VOR, VL=0, l=1
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=2,
OR[CD - 01000111101]=10111111101=VOR, VL=0, l=1
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=3,
OR[CD - 00100111101]=11011111101=VOR, VL=0, l=1
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=4,
OR[CD - 00010111101]=11101111101=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)] = $CDF(2)$, VL=1, n=5,
OR[CD - 00001111101]=11110111101=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=6,
m=3, n=1,
OR[00001111101,00010111101,00100111101]=00111111101=VOR, VL=0, l=1

AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=2,
OR[00001111101,00010111101,01000111101]=01011111101=VOR, VL=0, l=1
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=3,
OR[00001111101,00010111101,10000111101]=10011111101=VOR, VL=0, l=1
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=4,
OR[00001111101,00100111101,01000111101]=01101111101=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=5,
OR[00001111101,00100111101,10000111101]=10101111101=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)] = $CDF(2)$, VL=1, n=6,
OR[00001111101,01000111101,10000111101]=11010111101=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=7,
OR[00010111101,00100111101,01000111101]=01101111101=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=8,
OR[00010111101,00100111101,10000111101]=10110111101=VOR, VL=0,
$\forall l = 1, 2$, AND[VOR, CDF(l)] $\neq CDF(l)$, l=3,
AND[VOR, CDF(3)*] = $CDF(3)^*$, VL=1, n=9,
OR[00010111101,01000111101,10000111101]=11010111101=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=10,
OR[00100111101,01000111101,10000111101]=11100111101=VOR, VL=0, l=1
$\forall l = 1, 2$, AND[VOR, CDF(l)] $\neq CDF(l)$, l=3,
AND[VOR, CDF(3)*] = $CDF(3)^*$, VL=1, n=11,
m=2, n=1,
OR[00001111101,00010111101]=00011111101=VOR, VL=0, l=1
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=2,
OR[00001111101,00100111101]=00101111101=VOR, VL=0, l=1

AND[VOR, CDF(1)]$\neq CDF(1)$, l=2,
AND[VOR, CDF(2)]$\neq CDF(2)$, l=3,
AND[VOR, CDF(3)*]=CDF(3)*, VL=1, n=3,
OR[00001111101,01000111101]=01001111101=VOR, VL=0,
$\forall l$ AND[VOR, CDF(l)]$\neq CDF(l)$, VL=0
$\Delta = 55 + (5-2) \cdot 11 = 88$

Para i=1, j=2

CDM(1,2) : 00000111011, CD :$\left\{ \begin{array}{l} 00001111011 \\ 00010111011 \\ 00100111011 \\ 01000111011 \\ 10000111011 \end{array} \right\}$, m=5.

Se comprueba el primer coset de CD, 00001111011,
$\forall$l=1,2,3,4 y 5, AND[00001111011,CDF(l)]$\neq$CDF(l),
(o,p)=(1,1), W$_H$(XOR[00001111011,CDM(1,1)*])=1, se elimina de CD, m=4.
Se comprueba el segundo coset de CD, 00010111011,
$\forall$l=1,2,3,4 y 5, AND[00010111011,CDF(l)]$\neq$CDF(l),
(o,p)=(1,1), W$_H$(XOR[00010111011,CDM(1,1)])$\neq$1.
Se comprueba el tercer coset de CD, 00100111011,
$\forall$l=1,2,3,4 y 5, AND[00100111011,CDF(l)]$\neq$CDF(l),
(o,p)=(1,1), W$_H$(XOR[00100111011,CDM(1,1)])$\neq$1.
Se comprueba el cuarto coset de CD, 01000111011,
$\forall$l=1,2,3,4 y 5, AND[01000111011,CDF(l)]$\neq$CDF(l),
(o,p)=(1,1), W$_H$(XOR[01000111011,CDM(1,1)])$\neq$1.
Se comprueba el quinto coset de CD, 10000111011,
$\forall$l=1,2,3,4 y 5, AND[10000111011,CDF(l)]$\neq$CDF(l),
(o,p)=(1,1), W$_H$(XOR[10000111011,CDM(1,1)])$\neq$1.
mpri=4, n=1,
OR[CD]=11110111011=VOR, VL=0, l=1,
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] $= CDF(2)^*$, VL=1, n=2,
m=3, n=1,
OR[00010111011,00100111011,01000111011]=01110111011=VOR, VL=0, l=1
AND[VOR, CDF(1)*] $= CDF(1)^*$, VL=1, n=2,

OR[00010111011,00100111011,10000111011]=10110111011=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=3,
OR[00010111011,01000111011,10000111011]=11010111011=VOR, VL=0, l=1
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=4,
OR[00100111011,01000111011,10000111011]=11100111011=VOR, VL=0,
∀l=1,2,3 AND[VOR,CDF(l)]≠CDF(l), l=4,
AND[VOR, CDF(4)] = $CDF(4)$, VL=1, n=5,
m=2, n=1,
OR[00010111011,00100111011]=00110111011=VOR, VL=0,
∀l=1,2,3 AND[VOR,CDF(l)]≠CDF(l), l=4,
AND[VOR,CDF(4)*] = $CDF(4)^*$, VL=1, n=2,
OR[00010111011,01000111011]=01010111011=VOR, VL=0, l=1
AND[VOR,CDF(1)]≠CDF(1), l=2,
AND[VOR,CDF(2)*] = $CDF(2)^*$, VL=1, n=3,
OR[00010111011,10000111011]=10010111011=VOR, VL=0,
∀l=1,2,3 y 4, AND[VOR,CDF(l)]≠CDF(l), VL=0
$\Delta = 88 + (4-2) \cdot 11 = 110$

Para i=1, j=3

CDM(1,3) : 00000110111, CD: $\left\{ \begin{array}{l} 00001110111 \\ 00010110111 \\ 00100110111 \\ 01000110111 \\ 10000110111 \end{array} \right\}$

Se comprueba el primer coset de CD, 00001110111,
∀l=1,2,3,4 y 5, AND[00001110111,CDF(l)]≠CDF(l),
(o,p)=(1,1), W_H(XOR[00001110111,CDM(1,1)])≠1,
(o,p)=(1,2), W_H(XOR[00001110111, CDM(1,2)*]=1, se elimina de CD, m=4.
Se comprueba el segundo coset de CD, 00010110111,
∀l=1,2,3,4 y 5, AND[00010110111,CDF(l)]≠CDF(l),
∀(o,p)=(1,1) y (1,2), W_H(XOR[00010110111,CDM(o,p)])≠1.
Se comprueba el tercer coset de CD, 00100110111,
∀l=1,2,3,4 y 5, AND[00100110111,CDF(l)]≠CDF(l),

∀(o,p)=(1,1) y (1,2), W_H(XOR[00100110111,CDM(o,p)])≠1.
Se comprueba el cuarto coset de CD, 01000110111,
∀l=1,2,3,4 y 5, AND[01000110111,CDF(l)]≠CDF(l),
∀(o,p)=(1,1) y (1,2), W_H(XOR[01000110111,CDM(o,p)])≠1.
Se comprueba el quinto coset de CD, 10000110111,
∀l=1,2,3,4 y 5, AND[10000110111,CDF(l)]≠CDF(l),
∀(o,p)=(1,1) y (1,2), W_H(XOR[10000110111,CDM(o,p)])≠1.
mpri=4, n=1,
OR[CD]=11110110111=VOR, VL=0, l=1,
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=2,
m=3, n=1,
OR[00010110111,00100110111,01000110111]=01110110111=VOR, VL=0, l=1
AND[VOR, CDF(1)] ≠ $CDF(1)$, l=2
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=2,
OR[00010110111,00100110111,10000110111]=10110110111=VOR, VL=0,
∀$l = 1, 2$, AND[VOR, CDF(l)] ≠ $CDF(l)$, l=3
AND[VOR, CDF(3)*] = $CDF(3)^*$, VL=1, n=3,
OR[00010110111,01000110111,10000110111]=11010110111=VOR, VL=0, l=1
AND[VOR, CDF(1)] ≠ $CDF(1)$, l=2
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=4,
OR[00100110111,01000110111,10000110111]=11100110111=VOR, VL=0, l=1
AND[VOR, CDF(1)*] = $CDF(1)^*$, VL=1, n=5,
m=2, n=1,
OR[00010110111,00100110111]=00110110111=VOR, VL=0,
∀l=1,2,3 AND[VOR,CDF(l)]≠CDF(l), l=4,
AND[VOR,CDF(4)*] = $CDF(4)^*$, VL=1, n=2,
OR[00010110111,01000110111]=01010110111=VOR, VL=0, l=1
AND[VOR,CDF(1)]≠CDF(1), l=2,
AND[VOR,CDF(2)*] = $CDF(2)^*$, VL=1, n=3,
OR[00010110111,10000110111]=10010110111=VOR, VL=0,
∀l=1,2, AND[VOR,CDF(l)]≠CDF(l), l=3,
AND[VOR,CDF(3)*]=CDF(3)*, VL=1, n=4,
OR[00100110111,01000110111]=0110110111=VOR, VL=0,
∀l=1,2 AND[VOR,CDF(l)]≠CDF(l), l=3, VL=0
AND[VOR,CDF(3)*]=CDF(3)*, VL=1, n=5,

OR[00100110111,10000110111]=10100110111=VOR, VL=0,
$\forall$l=1,2,3,4 y 5 AND[VOR,CDF(l)]$\neq$CDF(l), VL=0
$\Delta = 110 + (4-2) \cdot 11 = 132$

Para i=1, j=4

CDM(1,4) : 00000101111, CD:$\left\{ \begin{array}{l} 00001101111 \\ 00010101111 \\ 00100101111 \\ 01000101111 \\ 10000101111 \end{array} \right\}$

Se comprueba el primer coset de CD, 00001101111,
$\forall$l=1,2,3,4 y 5, AND[00001101111,CDF(l)]$\neq$CDF(l),
(o,p)=(1,1),(1,2) W_H(XOR[00001101111,CDM(o,p)])$\neq$1, (o,p)=(1,3),
W_H(XOR[00001101111,CDM(1,3)*])=1,se elimina de CD, m=4.
Se comprueba el segundo coset de CD, 00010101111,
$\forall$l=1,2,3,4 y 5, AND[00010101111,CDF(l)]$\neq$CDF(l),
$\forall$(o,p)=(1,1),(1,2) y (1,3), W_H(XOR[00010101111,CDM(o,p)])$\neq$1,
Se comprueba el tercer coset de CD, 00100101111,
$\forall$l=1,2,3,4 y 5, AND[00100101111,CDF(l)]$\neq$CDF(l),
$\forall$(o,p)=(1,1),(1,2) y (1,3), W_H(XOR[00100101111,CDM(o,p)])$\neq$1.
Se comprueba el cuarto coset de CD, 01000101111,
$\forall$l=1,2,3,4 y 5, AND[01000101111,CDF(l)]$\neq$CDF(l), (o,p)=(1,1),
W_H(XOR[01000101111,CDM(1,1)*])=1, se elimina de CD, m=3.
Se comprueba el quinto coset de CD, 10000101111,
$\forall$l=1,2,3,4 y 5, AND[10000101111,CDF(l)]$\neq$CDF(l),
$\forall$(o,p)=(1,1),(1,2) y (1,3), W_H(XOR[10000101111,CDM(o,p)])$\neq$1.
mpri=3, n=1,
OR[CD]=10110101111=VOR, VL=0, l=1,
AND[VOR, CDF(1)] $\neq CDF(1)$, l=2,
AND[VOR, CDF(2)*] = $CDF(2)^*$, VL=1, n=2,
m=3, n=1,
OR[00010101111,00100101111]=00110101111=VOR, VL=0, l=1
$\forall$l=1,2,3,4 y 5 AND[VOR, CDF(l)] $\neq CDF(l)$, VL=0
$\Delta = 132 + (3-2) \cdot 11 = 143$

Para i=1, j=5

CDM(1,5) : 00000011111, CD: $\left\{ \begin{array}{l} 00001011111 \\ 00010011111 \\ 00100011111 \\ 01000011111 \\ 10000011111 \end{array} \right\}$

Se comprueba el primer coset de CD, 00001011111,
$\forall$l=1,2,3,4 y 5, AND[00001011111,CDF(l)]$\neq$CDF(l),
(o,p)=(1,1),(1,2),(1,3) W_H(XOR[00001011111,CDM(o,p)])$\neq$1, (o,p)=(1,4),
W_H(XOR[00001011111,CDM(1,4)*])=1,se elimina de CD, m=4.
Se comprueba el segundo coset de CD, 00010011111,
$\forall$l=1,2,3,4 y 5, AND[00010011111,CDF(l)]$\neq$CDF(l),
$\forall$(o,p)=(1,1),(1,2),(1,3) y (1,4) W_H(XOR[00010011111,CDM(o,p)])$\neq$1,
Se comprueba el tercer coset de CD, 00100011111,
$\forall$l=1,2,3,4 y 5, AND[00100011111,CDF(l)]$\neq$CDF(l),
$\forall$(o,p)=(1,1),(1,2),(1,3) y (1,4), W_H(XOR[00100011111,CDM(o,p)])$\neq$1.
Se comprueba el cuarto coset de CD, 01000011111,
$\forall$l=1,2,3,4 y 5, AND[01000011111,CDF(l)]$\neq$CDF(l), (o,p)=(1,1),
W_H(XOR[01000011111,CDM(1,1)*])=1, se elimina de CD, m=3.
Se comprueba el quinto coset de CD, 10000011111,l=1,
AND[10000011111,CDF(1)*]=CDF(1)*, se elimina de CD, m=2.
mpri=2, n=1,
OR[CD]=00110011111=VOR, VL=0,
$\forall$l=1,2,3, AND[VOR, CDF(l)] $\neq CDF(l)$, l=4,
AND[VOR, CDF(4)*]=CDF(4)*, VL=1, n=2,
$\Delta = 143 + (2-1) \cdot 11 = 154$

Después de cálculos similares con i=2, 3, 4 y 5 se obtiene finalmente la cota $\Delta = 242$.

3.7. Comparación Entre Ambos Algoritmos

En primer lugar se destacan las similitudes generales que presentan ambos algoritmos sin entrar en los detalles comunes que ya se han señalado en los apartados anteriores. En segundo lugar y siguiendo el mismo criterio se resaltan las diferencias más marcadas. En particular se termina esta enumeración con la diferencia más significativa, una tabla comparativa de

resultados numéricos. Por último se acaba este apartado con algunas observaciones y conclusiones que se pueden extraer de ambos algoritmos.

3.7.1. Similitudes

El primer punto en común entre ambos algoritmos lo constituyen precisamente la entrada y la salida. En ambos algoritmos la entrada está formada por dos valores L y k, que corresponden respectivamente a la longitud del RDRL y el orden del filtrado. Para estos valores resulta válida la cota inferior de la complejidad lineal Δ, única salida que se obtiene con estos algoritmos.

La única condición que se impone al filtrado para que se considere válida la cota obtenida es que tenga un único término de orden máximo. Por tanto esta restricción es un segundo punto en común.

Por otra parte y debido precisamente al primer punto mencionado se tiene en cualquiera de los dos casos que, al depender la cota Δ únicamente de los valores L y k se deduce que sólo es necesario aplicar el algoritmo una vez para cada par (L,k), cuestión a la que se hará referencia de nuevo más adelante.

El valor Δ obtenido para ese par (L,k) es válido para cualquier filtrado no lineal de orden k con un único término de orden máximo aplicado sobre cualquier RDRL de longitud L. En ambos casos la cota obtenida sirve para evitar el cálculo de la serie de determinantes en un cuerpo finito que exige el 'test de presencia de raíces' (o al menos de unos cuantos si la cota no resulta suficiente). En este último caso mencionado se presenta el mismo tipo de problema abierto con ambos algoritmos, el del análisis de los cosets no analizados con cada algoritmo.

3.7.2. Diferencias

Al comienzo de la descripción de cada algoritmo se establece la primera diferencia, el algoritmo 1 se puede llevar a cabo según un procedimiento en paralelo mientras que en el 2 hay que llevar cabo una vigilancia de los cosets que se repiten, por lo que se hace necesaria la secuencialidad.

Además hay que señalar que en el algoritmo 1 la experiencia ha demostrado que los incrementos aditivos obtenidos con cada uno de los procedimientos paralelos coinciden mientras que en el segundo obviamente los incrementos resultan diferentes para cada grupo de cosets analizados.

En cuanto a los conjuntos de cosets no analizados con cada uno de los dos algoritmos se tiene claramente que el conjunto correspondiente al algoritmo

1 contiene al conjunto correspondiente al algoritmo 2. A cambio de esto se tiene que la complejidad computacional del algoritmo 2 es mucho mayor que la del 1. En concreto la complejidad computacional máxima del algoritmo 1 viene dada por el siguiente número de operaciones realizadas en paralelo

$$N_L \cdot L \left[\binom{L-k-1}{L-k-2} + \binom{L-k-1}{L-k-3} + \cdots + \binom{L-k-1}{2} \right] = N_L \cdot L \cdot (2^{L-k-1} - L + k) \leq N_L \cdot L \cdot 2^{L-k-1}.$$

Por el contrario, la del algoritmo 2 viene dada por el siguiente número de operaciones secuenciales

$$N_L^2 \cdot (k-1) \cdot L \left[\binom{L-k}{L-k} + \binom{L-k}{L-k-1} + \cdots + \binom{L-k}{2} \right] = N_L^2 \cdot (k-1) \cdot L \cdot (2^{L-k} - L + k) \leq N_L^2 \cdot (k-1) \cdot L \cdot 2^{L-k}.$$

Por tanto la complejidad computacional del primer algoritmo es $\mathrm{O}(2^{L-k-1})$ mientras que la del segundo es $\mathrm{O}(2^{L-k})$.

Por último se presenta una tabla en la que se contrastan los resultados numéricos obtenidos para algunos pares (L,k) con cada uno de los dos algoritmos. Obsérvese que, debido a la mayor complejidad computacional del algoritmo 2 y a la escasez de medios computacionales, no ha sido factible hasta el momento aplicar el algoritmo 2 sobre valores de L superiores a L=53, por lo que para esos valores se han hecho estimaciones (señaladas con $\sim$) en base al crecimiento manifestado por los valores obtenidos.

L	k	Cota 1	Cota 2	L	k	Cota 1	L	k	Cota 1
2	1	2	2	67	34	13266	157	79	73476
3	2	3	3	71	36	14910	163	82	79218
5	3	10	10	73	37	15768	167	84	97027
7	4	24	28	79	40	18486	173	87	89268
11	6	165	242	83	42	20418	179	90	111517
13	7	234	728	89	45	23496	181	91	97740
17	9	544	3128	97	49	27936	191	96	127015
19	10	684	4617	101	51	30300	193	97	129696
23	12	1012	8349	103	52	31518	197	99	135142
29	15	2030	22330	107	54	34026	199	100	137907
31	16	1860	24645	109	55	35316	211	106	155085
37	19	3330	47952	113	57	37968	223	112	173271
41	21	3280	58220	127	64	48006	227	114	179557
43	22	4515	75852	131	66	51090	229	115	182742
47	24	5405	99405	137	68	55896	233	117	189196
53	27	6890	143206	139	70	57546	239	120	199087
59	30	8555		149	75	66156	241	121	202440
61	31	10980		151	76	67950	251	126	219625

3.8. Observaciones y Conclusiones

En la tabla numérica anterior sólo se contemplan valores de L primos. Hay varias razones para esta elección. Por una parte evita el cálculo de los valores d<L/2 primos con L necesarios ya que si L es primo, todos los valores d<L/2 son primos con L.

Por otro lado, dado que para valores de L primos existe siempre un número mayor de cosets de distancia fija, se obtienen mayores cotas para estos casos.

Además, la ausencia de factores comunes de L y k cuando L es primo evita la existencia de cosets simétricos. Esto resulta favorable para la obtención de cotas grandes ya que, según los resultados sobre la posible degeneración de los cosets simétricos vistos en el capítulo anterior, su existencia produciría una disminución considerable en el valor de la cota inferior de la complejidad lineal dada la imposiblidad de asegurar su no degeneración para todas las funciones no lineales.

Hay que mencionar que ambos algoritmos permiten algunas modificaciones sencillas tales como variar el recorrido de los bucles. Estos cambios

podrían producir mejoras en las cotas obtenidas, aunque se puede estimar que no serían mejoras significativas.

Las complejidades computacionales máximas obtenidas en el apartado anterior (considerando el 'peor caso posible') son exponenciales, sin embargo con respecto a este punto se deben hacer las siguientes observaciones válidas para ambos algoritmos:

1. De los resultados experimentales se concluye que en realidad no es necesario hacer todas las operaciones consideradas en el cálculo de la complejidad lineal máxima por lo que el 'peor caso posible' no resulta muy representativo.

2. Para cada par (L,k) sólo es necesario aplicar el algoritmo una vez.

3. Las longitudes de los RDRLs utilizados en la práctica no suelen ser mayores que un valor relativamente pequeño (L$\in [120, 250]$).

No obstante la mejor aportación de estos algoritmos consiste en que un valor alto de complejidad lineal obtenido para un par (L,k) permite utilizar una amplia clase de filtrados no lineales con bastante seguridad. Esta libertad de elección de filtrado resulta bastante cómoda para el criptógrafo ya que no tiene que cuidarse de utilizar funciones inseguras, lo que por otra parte hace imposible un ataque por búsqueda exhaustiva dado el tamaño del conjunto de claves.

Todo esto se traduce en último término en la confirmación del cifrado en flujo como un cifrado realmente seguro, lo que lo convierte en el cifrado seguro más rápido de todos.

La siguiente gráfica muestra el crecimiento según los cálculos y estimaciones de los valores de cota inferior de la complejidad lineal que se obtienen con el algoritmo 2. Tal como se aprecia, el crecimiento que se manifiesta se puede aproximar según una curva polinómica, lo que permite estimar que el primer valor de L para el que se obtiene una cota inferior mayor que $5 \cdot 10^5$ es 73, y para un de las longitudes más frecuentemente utilizadas actualmente L=127 se consigue una cota inferior mayor que $4 \cdot 10^6$. De darse esta última cota significaría que, si se pretendiera realizar un criptoanálisis utilizando el algoritmo de Berlekamp-Massey, habría que analizar más de ocho millones de dígitos. Luego se podría concluir la seguridad de estos filtrados en cuanto a complejidad lineal.

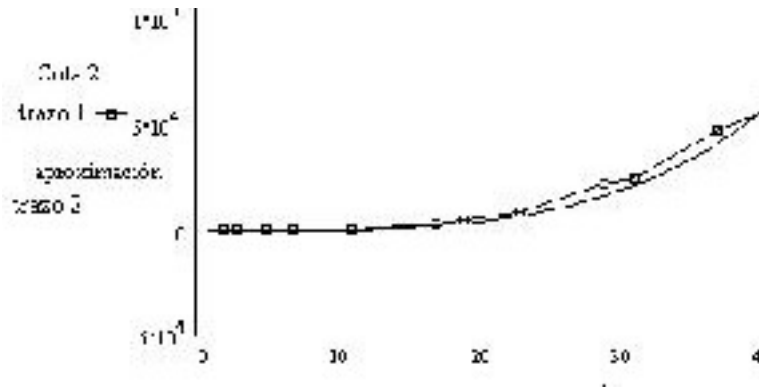


Figura 3.1: Crecimiento de la Cota

.

Capítulo 4

Procedimientos Alternativos

Este capítulo consta de dos partes bien diferenciadas. Cada una constituye una propuesta distinta de aproximación al problema de la complejidad lineal que difieren de la llevada a cabo en el resto del trabajo. La primera corresponde a un nuevo equivalente lineal descompuesto válido para el caso del filtrado no lineal. Su antecedente se remite al equivalente lineal descompuesto descrito en el primer capítulo, pero sin embargo su definición no sigue los mismos pasos. La segunda sección constituye la generalización del método de Kumar y Scholtz [90] presentado entre los antecedentes bibliográficos del primer capítulo.

4.1. Equivalente Lineal Descompuesto

En esta sección se presenta un nuevo equivalente lineal descompuesto, que en adelante se denotará como ELD. A diferencia del presentado en el primer capítulo, el propuesto aquí se deduce, utilizando algunas propiedades de los cosets, a partir de la expresión

$$s_n = \sum_{i=0}^{L-1} (\alpha^n)^{2^i} \ \ (\forall n = 0, 1, ...) \tag{4.1}$$

siendo α una raíz del polinomio minimal de $\{s_n\}$. Además este nuevo ELD sólo se describe para los filtrados no lineales.

En un primer apartado se plantea el ELD de la función no lineal más sencilla posible, el producto de orden dos. En los sucesivos apartados se obtiene su expresión para funciones más complejas.

4.1.1. Producto de Orden Dos

Se considera la función no lineal de orden dos $s_n s_{n+\delta}$ con $0 < \delta < 2^L - 1$.

A partir de la expresión (3) y de las propiedades de los cosets de peso dos en el anillo de los enteros mod(2^L-1), se obtiene, denotando r= $\lfloor L/2 \rfloor$, la siguiente expresión:

$$s_n s_{n+\delta} = \sum_{i=0}^{L-1} (\alpha^n)^{2^i} \sum_{j=0}^{L-1} (\alpha^{n+\delta})^{2^j} = \sum_{i=0}^{L-1} (\alpha^{2n+\delta})^{2^i} + \sum_{i=0}^{L-1} \sum_{j=0, j\neq i}^{L-1} (\alpha^n)^{2^i+2^j} (\alpha^\delta)^{2^j} \equiv$$
$$\sum_{i=0}^{L-1} (\alpha^n)^{2^i} (\alpha^{2^{L-1}\delta})^{2^i} + \sum_{i=0}^{L-1} \alpha^{(2^i+2^{i+1})n} (\alpha^{2^{i+1}\delta} + \alpha^{2^i\delta}) + \sum_{i=0}^{L-1} \alpha^{(2^i+2^{i+2})n} (\alpha^{2^{i+2}\delta} +$$
$$\alpha^{2^i\delta}) + \cdots + \sum_{i=0}^{L-1} \alpha^{(2^i+2^{i+r})n} (\alpha^{2^{i+r}\delta} + \alpha^{2^i\delta})$$

De donde resulta la siguiente

$$s_n s_{n+\delta} = \sum_{i=0}^{L-1} (\alpha^n)^{2^i} (\alpha^{2^{L-1}\delta})^{2^i} + \sum_{i=0}^{L-1} (\alpha^{(2+1)n})^{2^i} (\alpha^{2\delta} + \alpha^\delta)^{2^i} + \tag{4.2}$$

$$\sum_{i=0}^{L-1} (\alpha^{(2^2+1)n})^{2^i} (\alpha^{2^2\delta} + \alpha^\delta)^{2^i} + \cdots +$$
$$\begin{cases} \sum_{i=0}^{L-1} (\alpha^{(2^r+1)n})^{2^i} (\alpha^{2^r\delta} + \alpha^\delta)^{2^i} \ \textit{si } L \textit{ es impar} \\ \sum_{i=0}^{L-1} (\alpha^{(2^r+1)n})^{2^i} (\alpha^\delta)^{2^i} \textit{si } L \textit{ es par.} \end{cases}$$

Cuando δ es potencia de 2, la expresión (4) se puede simplificar quedando como suma de expresiones del tipo (2) de la forma siguiente

$$s_n s_{n+\delta} = s_{n+2^{L-1}\delta} + s_{(2+1)n+m_1\delta} + s_{(2^2+1)n+m_2\delta} + \cdots + s_{(2^r+1)n+m_r\delta} \tag{4.3}$$

siendo $\alpha^{m_i} \equiv \alpha^{2^i} + \alpha$ en GF(2^L) $\forall i$ y m_r tal que

$$\begin{cases} \alpha^{m_r} \equiv \alpha^{2^r} + \alpha & \textit{en } GF(2^L) \quad \textit{si } L \textit{ es impar} \\ m_r \equiv 1 \quad \textit{mod } (2^L - 1) & \textit{si } L \textit{ es par} \end{cases}$$

La expresión (5) se puede interpretar como **un equivalente lineal descompuesto** (**ELD**) formado por la suma de 1+r RDRLs. El primer RDRL corresponde al coset de peso uno mientras que los demás corresponden cada uno a los r cosets de peso dos. Todos los RDRLs componentes del ELD

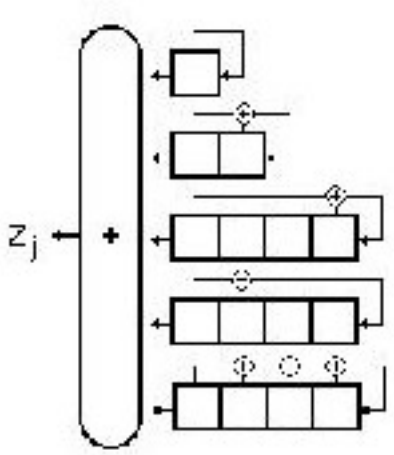


Figura 4.1: ELD

poseen el mismo polinomio característico del RDRL que genera la secuencia $\{s_n\}$. Sin embargo, la secuencia resultante de cada uno de los componentes se ve afectada por un desplazamiento de fase y una decimación distinta. Esto se muestra en la figura 8, donde para el caso que nos ocupa $df_0=2^{L-1}\delta$ y $df_i = m_i\delta \ \forall i = 1, 2, ..., r$.

Concretamente el ELD mostrado en esta figura es válido para cualquier filtrado de orden dos, tal y como se mostrará en los sucesivos apartados, donde se obtendrán los diferentes valores que toman los desplazamientos de fase para cada caso.

Dado que las decimaciones son valores fijos para cada L, y los m_i son independientes de δ, las modificaciones en el valor de δ no suponen cálculos complicados para la actualización del ELD.

Una de las ventajas más destacables del ELD propuesto consiste en una mejora en el tiempo de generación de la secuencia debida a la sustitución del producto de secuencias por la suma pues es bien sabido que en las implementaciones electrónicas la suma es mucho más fácil de realizar que el producto. Además hay que resaltar que las secuencias a sumar se obtienen todas con el mismo RDRL, por lo que sólo es necesario implementar un único circuito electrónico de este tipo. Ambas ventajas se valoran más a medida que se complica el filtrado no lineal considerado, tal y como se verá en posteriores

apartados.

Por otro lado, una segunda ventaja está relacionada con el cálculo de la complejidad lineal de la secuencia. Dicha complejidad lineal se puede obtener mediante el ELD debido a que coincide con la suma de las longitudes de los cosets a los que corresponden las componentes activas en el equivalente. En consecuencia la desactivación de una componente implica la degeneración del coset correspondiente y por tanto una disminución en el valor de la complejidad lineal. Dicha desactivación viene dada por la imposibilidad de cálculo del correspondiente desplazamiento de fase. Esta propiedad del ELD será utilizada en lo siguiente para demostrar algunos resultados sobre la complejidad lineal global del filtrado.

Dado que $\alpha^{2^i} + \alpha \not\equiv 0$ en GF(2^L) $\forall$i=1,2,...,r, se obtiene el siguiente resultado trivial.

Proposición 4.1.1

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta}$ *y* $0 < \delta < 2^L - 1$, *si* δ *es potencia de 2, entonces todos los cosets de peso 2 son no degenerados.*

Observación 4.1.1

El resultado anterior es completamente independiente del RDRL particular sobre el que se aplica el filtrado.

En general para un δ cualquiera, la única modificación que sufre el ELD anterior viene dada por la expresión siguiente:

$$s_n s_{n+\delta} = s_{n+df_0} + s_{(2+1)n+df_1} + s_{(2^2+1)n+df_2} + \cdots + s_{(2^r+1)n+df_r} \qquad (4.4)$$

donde los desplazamientos de fase quedan respectivamente de la forma $\mathrm{df}_0 = 2^{L-1}\delta$ y df_i, i=1,2,...,r, tales que $\alpha^{df_i} \equiv \alpha^{2^i\delta} + \alpha^{\delta}$ $\forall i \neq r$ y

$$\begin{cases} \alpha^{df_r} \equiv \alpha^{2^r\delta} + \alpha^{\delta} & en\ GF(2^L) \quad si\ L\ es\ impar \\ df_r \equiv \delta \quad mod\ (2^L - 1) & si\ L\ es\ par \end{cases}$$

De la expresión (6) se deducen directamente los dos resultados siguientes que determinan respectivamente la imposibilidad y la posibilidad de desactivación de algunas componentes del equivalente. Obsérvese que, de nuevo, ambos resultados son independientes del RDRL particular al que se aplica la función.

Proposición 4.1.2

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta}$ *y* $0 < \delta < 2^L - 1$, *el coset de peso uno es no degenerado.*

Proposición 4.1.3

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta}$ *y* $0 < \delta < 2^L - 1$, *el coset de peso 2,* 2^i+1 *(*$i\in\{1,2,\ldots,r\}$*) es degenerado si y sólo si* $2^i\delta \equiv \delta \ mod(2^L - 1)$.

De este último resultado se deduce que si $mcd(\delta, 2^L - 1) = 1$, entonces todos los cosets de peso dos son no degenerados. En consecuencia, para este caso, utilizando también el penúltimo resultado se obtiene que la complejidad lineal de la secuencia producida es $\binom{L}{2}+L$, lo que se presenta en el siguiente corolario.

Corolario 4.1.1

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta}$ *y* $0 < \delta < 2^L - 1$, *si* $mcd(\delta, 2^L - 1) = 1$, *entonces* $\Lambda(\{z_n\}) = L + \binom{L}{2}$.

Esta última expresión constituye un caso particular del resultado sobre productos de fases equidistantes demostrado por Rueppel [152]. Y el caso analizado previamente en que δ es potencia de dos es un caso particular de este último resultado.

La condición de degeneración de los cosets de peso dos dada en la proposición 4.3, $2^i\delta \equiv \delta \ mod(2^L - 1)$, coincide con la caracterización de los cosets simétricos presentada en el capítulo dos, $2^i E \equiv E \ mod(2^L - 1)$. Allí se demostró que cuando L es primo no existe ningún coset simétrico. Por tanto, cuando L es primo la hipótesis de la proposición 4.3 no se verifica para ningún i cuando L es primo, por lo que ningún coset de peso dos puede ser degenerado. En consecuencia se obtiene el siguiente resultado.

Corolario 4.1.2

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta}$ *y* $0 < \delta < 2^L - 1$, *si* L *es primo entonces* $\Lambda(\{z_n\}) = L + \binom{L}{2}$.

Observación 4.1.2

Este resultado coincide exactamente con el demostrado por Massey y Serconek en [116]. No obstante, para demostrarlo aquí se ha utilizado el resultado anterior deducido a partir del ELD conjuntamente con las condiciones de caracterización y existencia de cosets simétricos, mientras que en aquel caso se utilizaron transformadas discretas de Fourier.

4.1.2. Función de Orden Dos con un Único Término Producto

Dando un paso adelante en la generalización de la función, se considera ahora una función de la forma

$$s_n s_{n+\delta_0} + \sum_{i=1}^{m} s_{n+\delta_i} \text{ con}$$
$$0 \leq \delta_1 < \delta_2 < \cdots < \delta_m < 2^L - 1, 0 < \delta_0 < 2^L - 1 \text{ y } \ 0 < m < 2^L - 1.$$

En este caso la única modificación que sufre el último ELD considerado consiste en que el desplazamiento de fase de la primera componente deja de ser $2^{L-1}\delta$ para convertirse en df_0 tal que

$$\alpha^{df_0} \equiv \alpha^{2^{L-1}\delta_0} + \sum_{i=1}^{m} \alpha^{\delta_i} \text{ en GF}(2^L).$$

De ahí se deduce directamente el siguiente resultado.

Proposición 4.1.4

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta_0} + \sum_{i=1}^{m} s_{n+\delta_i}$, $0 \leq \delta_1 < \delta_2 < \cdots < \delta_m < 2^L - 1,\ 0 < \delta_0 < 2^L - 1$ *y* $0 < m < 2^L - 1$, *el coset de peso uno es degenerado si y sólo si* $\alpha^{2^{L-1}\delta_0} + \sum_{i=1}^{m} \alpha^{\delta_i} \equiv 0$ *en GF(*2^L*).*

La proposición 4.3 sigue siendo válida para la nueva función, considerando que δ_0 es la nueva notación para el δ utilizado allí.

Por otra parte los dos últimos corolarios de dicho apartado se traducen en el caso de esta función de la siguiente forma.

Corolario 4.1.3

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta_0} + \sum_{i=1}^{m} s_{n+\delta_i}$ *con* $0 \leq \delta_1 < \delta_2 < \cdots < \delta_m < 2^L - 1,\ 0 < \delta_0 < 2^L - 1$ *y* $0 < m < 2^L - 1$, *si mcd(*$\delta_0, 2^L - 1$*)=1 entonces* $\Lambda(\{z_n\}) \geq \binom{L}{2}$.

Corolario 4.1.4

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta_0} + \sum_{i=1}^{m} s_{n+\delta_i}$ *con* $0 \leq \delta_1 < \delta_2 < \cdots < \delta_m < 2^L - 1, 0 < \delta_0 < 2^L - 1$ *y* $0 < m < 2^L - 1$, *si* L *es primo entonces* $\Lambda(\{z_n\}) \geq \binom{L}{2}$.

4.1.3. Suma de Dos Productos de Orden Dos

Se considera ahora la función de orden dos de la forma $s_n s_{n+\delta_0} + s_{n+\delta_1} s_{n+\delta_2}$ con $0 < \delta_0 < 2^L - 1$ y $0 < \delta_1 < \delta_2 < 2^L - 1$.

El nuevo ELD sufre los siguientes cambios en los desplazamientos de fase, que ahora corresponden a los valores $df_0, df_1, .., df_r$ definidos según las siguientes expresiones

$$\begin{cases} \alpha^{df_0} \equiv (\alpha^{\delta_0} + \alpha^{\delta_1+\delta_2})^{2^{L-1}} & en\ GF(2^L) \\ \alpha^{df_i} \equiv \alpha^{\delta_0} + \alpha^{2^i\delta_0} + \alpha^{2^i\delta_1+\delta_2} + \alpha^{\delta_1+2^i\delta_2} & en\ GF(2^L)\quad \forall i = 1, 2, ..., r \end{cases}$$

A partir de la primera expresión se deduce en caso de degeneración del coset de peso uno, el siguiente resultado. Al igual que en casos anteriores, este resultado es independiente del RDRL.

Proposición 4.1.5

Para la secuencia $\{z_n\}_{n\in N}$ *con* $z_n = s_n s_{n+\delta_0} + s_{n+\delta_1} s_{n+\delta_2}$, $0 < \delta_0 < 2^L - 1$ *y* $0 < \delta_1 < \delta_2 < 2^L - 1$, *si* $\delta_0 \equiv \delta_1 + \delta_2$ *mod* $(2^L - 1)$, *entonces* $\Lambda(\{z_n\}) \leq \binom{L}{2}$.

La generalización de esta última función considerada al caso de una suma de n productos de orden dos es posible pero las expresiones que resultan no son tan fáciles de analizar. Análogamente el caso general de una función de orden dos con n productos de orden dos más m términos lineales corresponde a la conjunción de los últimos casos considerados. También en este caso se complican las expresiones impidiendo deducir fácilmente cotas a la complejidad lineal, tal y como se ha hecho en los casos analizados.

A partir de los resultados obtenidos hasta ahora se puede afirmar que resulta bastante sencillo el cálculo del ELD y, en consecuencia, de la complejidad lineal para muchos casos de funciones de orden dos.

A continuación se plantea el problema del cálculo del ELD para funciones de un orden superior.

4.1.4. Producto de Orden Tres

Se considera ahora el caso de una función de orden 3 de la forma $s_n s_{n+\delta_1} s_{n+\delta_2}$ con $0 < \delta_1 < \delta_2 < 2^L - 1$.

En el ELD, además de las $1 + \lfloor L/2 \rfloor$ componentes correspondientes a los cosets de peso 1 y de peso 2, aparecen las $\lceil \frac{(L-1)(L-2)}{6} \rceil$ correspondientes a los cosets de peso 3.

La expresión del ELD en este caso viene dada por

$$s_n s_{n+\delta_1} s_{n+\delta_2} =$$
$$s_{n+df_0} + s_{(2+1)n+df_1} + \cdots + s_{(2^r+1)n+df_r} + s_{(2^2+2+1)n+df_{r+1}} + \cdots + s_{(2^{j_s}+2^{k_s}+1)n+df_{r+s}}$$
$$\text{con } j_s = \lfloor L/3 \rfloor \text{ y } k_s = \begin{cases} L-2 & si L \neq \dot{3} \\ 2L/3 & si L = \dot{3} \end{cases}$$

A partir de las propiedades de los cosets en el anillo de los enteros $mod(2^L - 1)$ se deducen los valores de los desplazamientos de fase df_i según las expresiones siguientes:

$\alpha^{df_0} \equiv \alpha^{2^{L-2}\delta_1 + 2^{L-2}\delta_2} + \alpha^{2^{L-2}\delta_1 + 2^{L-1}\delta_2} + \alpha^{2^{L-1}\delta_1 + 2^{L-2}\delta_2}$ en GF(2^L),

$\alpha^{df_i} \equiv \alpha^{2^{L-1}\delta_1 + 2^i\delta_2} + \alpha^{2^{L-1}\delta_1 + 2^{L-1}\delta_2} + \alpha^{2^i\delta_1 + 2^{L-1}\delta_2} + \alpha^{\delta_1 + 2^{i-1}\delta_2} + \alpha^{2^{i-1}\delta_1 + 2^{i-1}\delta_2} + \alpha^{2^{i-1}\delta_1 + \delta_2}$ en GF(2^L), $\forall$i=1,2,...,r,

$\alpha^{df_i} \equiv (\alpha^{2^{j_i}\delta_1} + \alpha^{\delta_1})\alpha^{2^{k_i}\delta_2} + (\alpha^{2^{j_i}\delta_2} + \alpha^{\delta_2})\alpha^{2^{k_i}\delta_1} + \alpha^{\delta_1 + 2^{j_i}\delta_2} + \alpha^{2^{j_i}\delta_1 + \delta_2}$ en GF(2^L), $\forall$i=r+1,r+2,...,r+s-1 y

$$\alpha^{df_{r+s}} \equiv$$
$$\begin{cases} (\alpha^{2^{j_{r+s}}\delta_1} + \alpha^{\delta_1})\alpha^{2^{k_{r+s}}\delta_2} + (\alpha^{2^{j_{r+s}}\delta_2} + \alpha^{\delta_2})\alpha^{2^{k_{r+s}}\delta_1} + \alpha^{\delta_1 + 2^{j_{r+s}}\delta_2} + \alpha^{2^{j_{r+s}}\delta_1 + \delta_2} \\ \qquad si\ L \neq \dot{3} \\ \alpha^{\delta_1 + 2^{j_s}\delta_2} + \alpha^{2^{j_s}\delta_1 + \delta_2}\ si\ L = \dot{3} \end{cases}$$

siendo s=$\lceil \frac{(L-1)(L-2)}{6} \rceil$ y 0¡j$_i$¡k$_i$¡L $\forall i$.

De lo anterior se deduce que cuanto mayor es el orden de la función, más complicado resulta el cálculo de los desplazamientos de fase. En particular, para una función de orden 4 de la forma

$$s_n s_{n+\delta_1} s_{n+\delta_2} s_{n+\delta_3} \text{ con } 0 < \delta_1 < \delta_2 < \delta_3 < 2^L - 1,$$

el desplazamiento de fase de la componente correspondiente al coset de peso uno se obtiene a partir de la expresión siguiente

$\alpha^{df_0} \equiv$

$\alpha^{2^{L-2}(\delta_1+\delta_2+\delta_3)} + \alpha^{2^{L-3}(\delta_1+2\delta_2+2^2\delta_3)} + \alpha^{2^{L-3}(\delta_1+2^2\delta_2+2\delta_3)} + \alpha^{2^{L-3}(2\delta_1+\delta_2+2^2\delta_3)} + \alpha^{2^{L-3}(2^2\delta_1+\delta_2+2\delta_3)} + \alpha^{2^{L-3}(2\delta_1+2^2\delta_2+\delta_3)} + \alpha^{2^{L-3}(2^2\delta_1+2\delta_2+\delta_3)} + \alpha^{2^{L-3}(\delta_1+\delta_2+2^2\delta_3)} + \alpha^{2^{L-3}(\delta_1+\delta_2+2\delta_3)} + \alpha^{2^{L-3}(\delta_1+2^2\delta_2+\delta_3)} + \alpha^{2^{L-3}(\delta_1+2\delta_2+\delta_3)} + \alpha^{2^{L-3}(2^2\delta_1+\delta_2+\delta_3)} + \alpha^{2^{L-3}(2\delta_1+\delta_2+\delta_3)}$ en $\mathrm{GF}(2^L)$.

Se concluye por tanto que a medida que aumenta el orden de la función, resulta más difícil el análisis de la degeneración los cosets mediante el cálculo de los desplazamientos de fase de las correspondientes componentes del equivalente. No obstante, hay que destacar que las expresiones pueden simplificarse en algunos casos concretos.

4.1.5. Generalización y Conclusiones

En esta sección se ha desarrollado un modelo de equivalente lineal descompuesto que cumple dos funciones. Por un lado representa una mejora en la implementación electrónica de las funciones no lineales ya que sustituye el producto de secuencias por una suma siendo además los RDRLs que las generan idénticos al RDRL básico. Por otro lado sirve de herramienta de estudio de la complejidad lineal de las secuencias producidas, tal y como se ha manifestado con los sucesivos resultados obtenidos.

Después de analizar las posibilidades de dicha herramienta para cada una de los casos de funciones de orden dos, hemos introducido el problema para las funciones de un orden superior. En este caso ha quedado de manifiesto que las expresiones generales de los desplazamientos de fase de las componentes del equivalente propuesto se complican a medida que aumenta el orden de la función. Se ha abierto por tanto un camino a la deducción de funciones para las que el cálculo de esas expresiones resulte sencillo, pudiéndose así deducir la degeneración o no degeneración de determinados cosets, tal y como se ha hecho aquí para las funciones de orden dos.

4.2. Generalización del Método de Kumar y Scholtz

En esta sección se generalizan los resultados obtenidos por Kumar y Scholtz en [90] expresándolos en términos de los cosets simétricos estudiados en el capítulo 2. Al mismo tiempo se relajan las condiciones del filtrado

no lineal analizado. Al igual que en aquel caso, aquí se imponen condiciones tanto sobre la función no lineal como sobre la longitud del RDRL.

En adelante α representa un elemento primitivo de GF(2^L). En los dos primeros apartados se toma como valor de L un número par y se utiliza un entero m=L/2≥k.

4.2.1. Degeneración de Todos los Cosets Simétricos

En el próximo resultado se garantiza la degeneración de los cosets simétricos de peso menor o igual que k para unas determinadas funciones.

Teorema 4.2.1

Sea f es una función no lineal de orden k aplicada sobre m elementos s_{n+t_1}, s_{n+t_2}, ..., s_{n+t_m} *de una PN-secuencia tales que* $\alpha^{t_i} \in GF(2^m)$ $\forall i = 1, 2, .., m$. *Entonces para el filtrado no lineal resultante se tiene garantizada la degeneración de todos los cosets simétricos E de peso menor o igual que k tales que* $\alpha^E \in GF(2^m)$.

Demostración Se considera l=2^m + 1. El elemento s_{n+t_i} de la PN-secuencia puede escribirse como

$$s_{n+t_i} = Tr_1^L(\alpha^{t_i}\alpha^n) = \sum_{j=0}^{L-1}(\alpha^{t_i})^{2^j}(\alpha^n)^{2^j}$$

Tomemos ahora $\{s_{nl+t_i}\}$ una subsecuencia de la secuencia $\{s_{n+t_i}\}$. Vemos que

$$s_{nl+t_i} = Tr_1^L(\alpha^{t_i}\alpha^{nl}) = \sum_{j=0}^{L-1}\alpha^{nl2^j}\alpha^{t_i 2^j} = 0$$

puesto que $\alpha^{nl2^{m+j}} = \alpha^{nl2^j}$ $\forall j = 0, 1, ..., m-1$ y $\alpha^{t_i 2^j} = \alpha^{t_i 2^{m+j}}$ $\forall j = 0, 1, ..., m$ por ser $\alpha^{t_i} \in GF(2^m)$ $\forall i = 1, 2, ..., m$.

Por tanto los elementos de la traza anterior se anulan dos a dos.

Sea $\{z_n\}$ una secuencia cuyos elementos son de la forma z_n=f(s_{n+t_1},...,s_{n+t_m}) y sea $\{z_{nl}\}$ una subsecuencia de la anterior $\{z_{nl}\} \subset \{z_n\}$, cuyos elementos son de la forma z_{nl}=f(s_{nl+t_1},...,s_{nl+t_m}). Como s_{nl+t_i}=0 $\forall i = 1, ..., m$, tenemos que f(0,0,...,0)=0 ∀n luego z_{nl}=0 ∀n y la subsecuencia $\{z_{nl}\}$ será idénticamente nula.

Por otro lado sabemos que z_n=f(s_{t_i+n}) $= \sum_Q Tr_1^{P_Q}(A_Q \alpha^{Qn})$ $0 \leq n \leq 2^L - 2$ donde el sumatorio se extiende a todos los cosets Q de peso menor o igual que k. Los coeficientes A_Q de dichos cosets se han obtenido a partir de los $\alpha^{t_i} \in GF(2^m)$ y por tanto son elementos de GF(2^m). Al mismo tiempo se puede ver dichos coeficientes también pertenecen a GF(2^{P_Q}) donde P_Q es un entero tal que GF(2^{P_Q}) es el menor cuerpo que contiene a α^Q. Si se define e=m.c.d.(m, P_Q) entonces GF(2^e) es un subcuerpo de GF(2^m) y de GF(2^{P_Q}) a la vez que $A_Q \in GF(2^e)$.

Denotamos por Q^* al grupo de cosets tales que $\alpha^{Q^*} \notin GF(2^m)$ y por Q al grupo de cosets tales que $\alpha^Q \in GF(2^m)$. Para el primer grupo se tiene que P_{Q^*} no es divisor de m, luego e $\neq P_{Q^*}$ por lo que se puede concluir que e=$P_{Q^*}/2$ y consecuentemente $A_{Q^*} \in GF(2^{P_{Q^*}/2})$. Por otro lado es fácil ver que $\alpha^{lQ^*} \in GF(2^{P_{Q^*}})$ y simultáneamente $\alpha^{lQ^*} \in GF(2^m)$ (ya que $\alpha^l \in GF(2^m)$ por ser E_l un coset simétrico de cardinal m), por lo que se deduce que $\alpha^{lQ^*} \in GF(2^{P_{Q^*}/2})$.

Reescribiendo z_{nl} como

$$z_{nl} = \sum_Q Tr_1^{P_Q}(A_Q \alpha^{lQn}) + \sum_{Q^*} Tr_1^{P_{Q^*}}(A_{Q^*} \alpha^{lQ^* n}) = 0$$

tenemos que por ser $A_{Q^*} \in GF(2^{P_{Q^*}/2})$ la contribución del segundo sumatorio es cero por anularse los sumandos dos a dos, luego los elementos z_{nl} quedan expresados únicamente en función de los cosets Q:

$$z_{nl} = \sum_Q Tr_1^{P_Q}(A_Q \alpha^{lnQ}) = \sum_Q Tr_1^{P_Q}(A_Q \alpha^{2nQ}) = 0 \ \forall n$$

Tendríamos un total de $\sum_Q P_Q$ coeficientes A_Q, luego tomando ese mismo número de ecuaciones homogéneas para n=0,1,..., $(\sum_Q P_Q)$-1 planteamos un sistema de ecuaciones lineal homogéneo cuyo determinante es un Vandermonde distinto de cero y las incógnitas los coeficientes A_Q. La única solución es la trivial A_Q=0, que implica la degeneración de todos los cosets referenciados en el enunciado.

Este resultado se traduce en una disminución en el valor de la complejidad lineal de los filtrados implicados. Su repercusión se centra en que proporciona información sobre algunos cosets de peso menor que el orden de la función, cosets que se escapan del control del 'test de presencia de raíces', por lo que son muy difíciles de analizar.

4.2.2. Degeneración de Algunos Cosets Simétricos

En el siguiente resultado se garantiza la degeneración de algunos cosets bajo determinadas circunstancias.

Teorema 4.2.2

Si f es una función de orden k par y en cada uno de los términos de orden máximo se tienen al menos $\frac{k}{2}+1$ *etapas* s_{n+t_i} *(i=1,...,*$\frac{k}{2}+1$*) tales que* $\alpha^{t_i} \in GF(2^m)$*, entonces para el filtrado no lineal resultante se tiene garantizada la degeneración de todos los cosets simétricos de peso k y cardinal m o divisor de m.*

Demostración El coeficiente que acompaña a dichos cosets sólo depende de los términos de orden k. Se considera uno de estos términos, por ejemplo $\prod_{i=1}^{k} s_{n+t_i}$, que se puede escribir como $\prod_{i=1}^{k}\left(\alpha^{t_i}\alpha^{n}+\alpha^{t_i 2}\alpha^{2n}+\cdots+\alpha^{t_i 2^{L-1}}\alpha^{2^{L-1}n}\right)$.

Dado un coset simétrico de peso k Q=$2^{e_0}+\cdots+2^{e_{k-1}}$, el término α^{Qn} aparece en dicha expresión acompañado por un coeficiente de la forma

$$\begin{vmatrix} \alpha^{t_1 2^{e_0}} & \alpha^{t_2 2^{e_0}} & \ldots & \alpha^{t_k 2^{e_0}} \\ . & . & \ldots & . \\ \alpha^{t_1 2^{e_{k/2-1}}} & \alpha^{t_2 2^{e_{k/2-1}}} & \ldots & \alpha^{t_k 2^{e_{k/2-1}}} \\ . & . & \ldots & . \\ \alpha^{t_1 2^{e_0}} & \alpha^{t_2 2^{e_0}} & \ldots & \alpha^{t_k 2^{e_0}} \\ . & . & \ldots & . \\ \alpha^{t_1 2^{e_{k/2-1}}} & \alpha^{t_2 2^{e_{k/2-1}}} & \ldots & \alpha^{t_k 2^{e_{k/2-1}}} \end{vmatrix},$$

determinante que, al desarrollarse por las k-1+k/2 últimas columnas, resulta ser nulo por ser $\alpha^{t_i} \in GF(2^m)$.

Observación 4.2.1

Si L es potencia de dos se tiene que el número de cosets degenerados según el lema anterior aumenta debido a que todos los subcuerpos de GF(2^L*) están contenidos unos dentro de otros. Por tanto, esta restricción, aunque deja menos posibilidad de elección en las etapas, a cambio permite obtener más información sobre la complejidad lineal del filtrado resultante.*

Ejemplo 4.2.1

Para L=2^3, se tiene GF(2^{2^3}) = $\{0, 1, \alpha, \alpha^2, ..., \alpha^{254}\}$ que tiene los subcuerpos GF(2^{2^2}) = $\{0, 1, \alpha^{17}, \alpha^{34}, \alpha^{68}, \alpha^{136}, \alpha^{51}, \alpha^{85}, \alpha^{153}, \alpha^{102}, \alpha^{170}, \alpha^{204}, \alpha^{187}, \alpha^{119}, \alpha^{238}, \alpha^{221}\} \supset$ GF(2^2)=$\{0,1,\alpha^{85}, \alpha^{170}\} \supset GF(2) = \{0, 1\}$.

Los cosets indicados por el teorema son $2^0+2^1+2^4+2^5$ y $2^0+2^2+2^4+2^6$. Para la función producto de cuatro fases tales que tres de ellas corresponden a los tres elementos no nulos de GF(2^4), s_n, s_{n+85} y s_{n+102} se tiene asegurada la degeneración de los dos cosets anteriores.

4.2.3. No Degeneración de Algunos Cosets

Las funciones no lineales a considerar en este apartado, que poseen un único término de orden máximo. Este término $s_{n+t_0}s_{n+t_1}...s_{n+t_{k-1}}$ es tal que $\alpha^{t_i} \in GF(2^m)$, es decir, las etapas que lo forman se corresponden con elementos de dicho subcuerpo.

En el próximo resultado, válido para cualesquiera L y k, se exige que dichas etapas correspondan a elementos de una base normal del subcuerpo GF(2^m) sobre GF(2), $\{\beta, \beta^2, \beta^{2^2}, ..., \beta^{2^{m-1}}\}$.

Teorema 4.2.3

Si f es una función no lineal de orden k de la forma f(s_{n+t_1}, s_{n+t_2}, ..., s_{n+t_L}) = $\prod_{i=1}^{k} s_{n+t_i}$, el conjunto $\{\beta^{2^i}\}_{i=0,1,...,L-1}$ es una base normal de GF(2^L) sobre GF(2) y la secuencia $s_{n+t_i} = Tr_1^L(\alpha^n)^{\alpha_i^t}$ i=1,2,...,L, siendo $\alpha^{t_i} = \alpha^{r2^{r_i}} \in \{\beta^{2^i}\}_{i=0,1,...,L-1}$ donde 1$\leq r \leq 2^L - 2$, $r_i \equiv i \cdot d \ (mod\ L)$ y mcd(d,L)=1, entonces para el filtrado no lineal resultante se tiene garantizada la no degeneración de todos los cosets de peso k.

Demostración El coeficiente que acompaña a cualquier coset Q=$2^{e_0}+\cdots+2^{e_{k-1}}$ en la expresión de f(s_{n+t_1}, ..., s_{n+t_L})es de la forma

$$\begin{vmatrix} \alpha^{r2^{e_0}2^{r_0}} & \alpha^{r2^{e_0}2^{r_1}} & ... & \alpha^{r2^{e_0}2^{r_{k-1}}} \\ \alpha^{r2^{e_1}2^{r_0}} & \alpha^{r2^{e_1}2^{r_1}} & ... & \alpha^{r2^{e_1}2^{r_{k-1}}} \\ . & . & ... & . \\ \alpha^{r2^{e_{k-1}}2^{r_0}} & \alpha^{r2^{e_{k-1}}2^{r_1}} & ... & \alpha^{r2^{e_{k-1}}2^{r_{k-1}}} \end{vmatrix}.$$

Si se denota $\eta_i = \alpha^{r2^{e_i}}$ $\forall i = 0, 1, ..., k-1$ se tiene que el determinante anterior se puede expresar de la siguiente forma

$$\begin{vmatrix} \eta_0^{2^{r_0}} & \eta_1^{2^{r_0}} & \dots & \eta_{k-1}^{2^{r_0}} \\ \eta_0^{2^{r_1}} & \eta_1^{2^{r_1}} & \dots & \eta_{k-1}^{2^{r_1}} \\ . & . & \dots & . \\ \eta_0^{2^{r_{k-1}}} & \eta_1^{2^{r_{k-1}}} & \dots & \eta_{k-1}^{2^{r_{k-1}}} \end{vmatrix},$$

que por el teorema 2.1 se sabe no nulo en GF(2^L).

Para las funciones que cumplen las hipótesis del resultado anterior se tiene garantizada una aportación a la complejidad lineal de valor $\binom{L}{k}$. Si las hipótesis de los teoremas 4.2 y 4.3 se unen se obtiene la siguiente cota inferior a la complejidad lineal.

Corolario 4.2.1

Si L y k son números pares, m=L/2 y k≤m, dada una función con un único término de orden máximo que sea producto de k fases s_{n+t_i} correspondientes a $\alpha^{t_i} = \beta^{2^{r_i}}$ $\forall i = 0, 1, ..., k-1$, $r_i \equiv d \cdot i \ (mod\ L)$, elementos de una base normal de GF(2^m) sobre GF(2), entonces la complejidad lineal del filtrado no lineal resultante está acotada inferiormente según $\Lambda \geq \binom{L}{k} - \binom{L/2}{k/2}$.

Los teoremas 4.1, 4.2 y 4.3 son fruto de la aplicación del método de Kumar y Scholtz [90] a nuevos grupos de cosets y de funciones. En particular, el último teorema representa una generalización de su análogo en [90], ya que en aquél se exigía que las fases fueran elementos seguidos de la base normal. Por tanto, el teorema 4.3 proporciona una mayor versatilidad en la elección de las fases del término de orden máximo.

Capítulo 5

Principales Aportaciones y Conclusiones

Los principales resultados obtenidos pueden resumirse de la siguiente manera:

1. Se ha definido un grupo de cosets, los cosets de distancia fija, y se ha demostrado que son no degenerados para cualquier filtrado no lineal con un único término de orden máximo. De esta forma se ha obtenido una cota inferior de la complejidad lineal válida para todos esos filtrados.

2. Se ha descrito un nuevo grupo de funciones no lineales, los llamados productos de fases $2^{((d))}$-distantes, tales que, utilizados como filtrados no lineales, resultan tener todos los cosets de peso igual al orden de la función, no degenerados. Además se ha ampliado este análisis a las funciones no lineales que se pueden expresar como combinación lineal de dichos productos obteniendo también en este caso un valor alto de complejidad lineal.
   Ambos resultados resultan de mayor utilidad que los encontrados en la Literatura. El primero porque goza de total independencia tanto de la función no lineal como del polinomio característico del RDRL. El segundo sólo es comparable con el resultado sobre 'productos de fases equidistantes' [152], ya que ningún otro autor ha obtenido una cota inferior tan alta para la complejidad lineal de un filtrado no lineal.

3. Se ha introducido y analizado en profundidad un nuevo y amplio grupo de cosets, los llamados cosets simétricos. Mediante ellos se han obtenido

dos resultados duales:
-Una cota superior de la complejidad lineal válida para un amplio grupo de filtrados no lineales.
-Un grupo de filtrados no lineales para los que es válida una cota superior pequeña.
Los resultados referidos se pueden incorporar a la tabla de recapitulación dada en el apartado 1.3.5 del primer capítulo de la forma siguiente:

Condiciones	*Cota*
Único término de orden máximo	$\Lambda \geq \frac{\phi(L)}{2} \cdot L$
$mcd(d, L) = 1$ *Término de orden máximo* : $\sum_{i=0}^{N-1} b_i s_{n+i+1} s_{n+i+2^{((d))}} \cdots s_{n+i+2^{(((k-1)d)}}$	$\Lambda \geq \binom{L}{k} - N + 1$
Único término de orden máximo $s_n s_{n+t_1} \cdots s_{n+t_{k-1}}$: $k \mid L \ \exists i : \alpha^{t_i} \in GF(2^{L/k})$	$\Lambda \leq \sum_{i=1}^{k} \binom{L}{i} - \frac{L}{k}$
$\forall j, \forall i : \alpha^{t_i} \in GF(2^{L/fc_j(L,k)})$	$\Lambda \leq \sum_{i=1}^{k} \binom{L}{i} - \sum_{j:fc_j \ primo} \binom{L/fc_j}{k/fc_j}$
$L = p^{l_1} l_2, \ k = p^{k_1} k_2, \ p \ primo,$ $(l_2, k_2) = 1, \ \forall i = 0, ..., k/p :$ $\alpha^{t_i} \in GF(2^{L/p})$	$\Lambda \leq \sum_{i=1}^{k} \binom{L}{i} - \binom{L/p}{k/p}$

4. Se ha introducido una representación binaria válida para ciertos determinantes. Esta nueva representación ha permitido el desarrollo de una técnica basada en manipulaciones de cadenas binarias que viene a sustituir el cálculo de dichos determinantes, lo que representa un gran ahorro en computación.

5. Se han presentado dos algoritmos de cálculo de cotas inferiores de la complejidad lineal de cualquier filtrado no lineal con un único término de orden máximo. La única entrada necesaria para ambos es la longitud del RDRL, por lo que las cotas obtenidas son válidas para todos los filtrados no lineales aplicados sobre cualquier RDRL de esa longitud.

6. Mediante los resultados obtenidos con el algoritmo 2 se ha acotado inferiormente la complejidad lineal según una curva polinómica, lo que ha permitido estimar unos valores de cota inferior muy altos para muchos casos prácticos.

7. Se ha propuesto un nuevo equivalente lineal descompuesto (ELD) válido para cualquier filtrado no lineal. Este ELD presenta dos ventajas:
   -Su cálculo implica el de la complejidad lineal.
   -Su uso para la generación de la secuencia filtrada permite sustituir el producto de secuencias por la suma, con la consecuente mejora en la implementación electrónica que esto supone.

8. Se ha generalizado el método presentado por Kumar y Scholtz [90] para el estudio de la complejidad lineal, demostrándose aquí que el caso estudiado por ellos no es más que una mera particularización de un caso analizado en este trabajo.

Los problemas que quedan abiertos y que pueden ser objeto de posteriores investigaciones son:

Estudio de nuevos grupos de cosets, similares a los de distancia fija, que sean no degenerados independientemente de la función no lineal y del polinomio característico del RDRL.

Análisis de otros filtrados no lineales parecidos a los productos de fases $2^{((d))}$-distantes, de manera que las complejidades lineales obtenidas tengan un valor alto asegurado.

Determinación de condiciones y restricciones para los filtrados no lineales de manera que se asegure la no degeneración de los cosets simétricos.

Extensión del rango de aplicación de los algoritmos presentados a un conjunto mayor de cosets.

Implementación en ordenadores de gran potencia de cálculo para completar la tabla de resultados numéricos.

Desarrollo de una técnica complementaria que estudie la degeneración de los cosets no analizados en la memoria.

Deducción de filtrados no lineales para los que el cálculo del ELD resulte sencillo.

Análisis de la complejidad lineal desde un punto de vista probabilístico.

Examen de las interrelaciones entre la complejidad lineal, el PCL, y la aleatoriedad de una secuencia binaria.

Capítulo 6

Apéndice

6.1. Algoritmo 1

```
/*Programa realizado por Amparo Fúster y Pino Caballero*/
/*Algoritmo 1 - Caso L=37, k=19*/
/*Objetivo:Obtener Cota Inferior de la Complejidad Lineal*/

#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <io.h>
unsigned comprueba(unsigned long q);
unsigned long CDF[36]; /* NL x 2 */
unsigned long CD[612]; /* tamano x NL x 2 */
unsigned long incremento, prij = 0; char a[32];
FILE *fi;
int i;
int L = 37, tamano, NL, m;
void main() {
unsigned char s, s1, s2;
int K, d[18], tope, paso; /* NL */
int j1, j2, n;
register int j, k;
unsigned long p1;
K = (L + 1) >> 1;
```

```
tamano = L - (K + 1);
NL = (L - 1) >> 1;
incremento = L * NL;
fi = fopen(çota4.sol", "w");
for(i = 0; i < NL*2; i++) /* inicialización CDF(i) */
CDF[i] = 0;
for(i = 0; i < NL; i++) /* inicialización d(i) */
d[i] = i + 1;
for(i = 0; i < tamano ; i++) /* inicialización prij */
prij |= (0x1 << i);
tope = NL * tamano;
for(i = 0; i < tope * 2; i++) /* inicialización CD(i) */
CD[i] = 0;
fprintf(fi, "\nL = %d K = %d tamano = %d NL = %d incremento = %lu\n\n",L,
K, tamano, NL, incremento);
fprintf(fi, "\ntope = %d prij = %lu\n\n", tope, prij);
for(i = 0; i < NL * 2; i = i + 2) { /* bucle generación CDF */
j = 0;
m = K - 1;
s1 = 0;
paso = d[i>>1];
while(m–) { /* generación cosets de distancia fija */
if(j > 31 ) CDF[i + 1] |= (0x1 << (j-32));
else CDF[i] |= (0x1 << j);
j = (j + paso) % L; } /* cierre del while */
if( j > 31 ) { /* ceros especiales */
j1 =j - 32; s1 = 1; }
else j1 = j;
j2= L-paso;
s2=0;
if (j2 > 31) {
j2 -= 32;
s2=1; }
for(j = 0; j < tamano * 2; j = j + 2) {
CD[(tamano * i) + j] = CDF[i]; /* coset base con k-1 1's */
CD[(tamano * i) + j + 1] = CDF[i + 1]; }
j = 0;
for(k = 0; k < 32; k++) /* genera CD[i] */ {
```

```
    if(! (CDF[i] & (0x1 << k)) ) /* encontramos un 0 */ {
    if( ((s1 == 1) || (k != j1)) && ((s2 == 1) || (k != j2)) ) {
    CD[(tamano*i) + j] |= (0x1 << k); j = j + 2; }} }
    for(k = 0; k < L - 32; k++) {
    if(! (CDF[i + 1] & (0x1 << k)) ) {
    if( ((s1 == 0) || (k != j1)) && ((s2 == 0) || (k != j2)) ) {
    CD[(tamano*i) + j + 1] |= (0x1 << k);
    j = j + 2; } } } /* Añade k-ésimo 1 a CDF[i] */
    CDF[i + s1] |= (0x1 << j1); /* escribe CDF[i] */
    ultoa(CDF[i + 1], a, 2);
    fprintf(fi, "\n\nCDF[%02d] =%05s", i >> 1, a);
    ultoa(CDF[i], a, 2);
    fprintf(fi, " %032s\n", a); /* escribe CD[] */
    for(j = 0; j < tamano * 2; j=j+2) {
    ultoa(CD[(tamano*i) + j + 1], a, 2);
    fprintf(fi, "\n %05s", a);
    ultoa(CD[(tamano*i) + j], a, 2);
    fprintf(fi, " %032s", a); } } /* fin for i */
    for(i = 0; i < NL * 2; i = i + 2) /* bucle trabajo con cada grupo */
    { m = tamano; /* m=n. uplas que toma */
    while(m--) {
    j1 = prij; /* recorre las uplas de long tamano */
    while(j1--) {
    n = 0; /* cuenta unos */
    for(k = 0; k < tamano; k++)
    if(j1 & (0x1 << k)) n++;
    if(n == m)
    if(!comprueba(j1)) /* a la subrutina */ {
    incremento += ((tamano - m) * L);
    ultoa(j1, a, 2);
    fprintf(fi, "\nincremento =%lu m =%d j1 =%s n =%d\n\n", incremen-
to, m, a, n);
    m = 0;
    j1 = 0; } } } }
    fclose(fi);} /* fin programa principal */
    unsigned comprueba(unsigned long q) { /* subrutina */
    register int l, p;
    long unsigned OR0, OR1, RCDF0, RCDF1;
```

```
char flag;
OR0 = OR1 = 0;
for(l = 0; l < tamano; l++)
if(q & (0x1 << l)) {
OR0 |= CD[(i * tamano ) + 2*l];
OR1 |= CD[(i * tamano ) + 2*l + 1]; }
for(l = 0; l < NL * 2; l = l + 2) {
p = L;
RCDF0 = CDF[l];
RCDF1 = CDF[l + 1];
while(p–) /* p=n. rotaciones */ {
flag=0; /* uno a pasar de RCDF0 a RCDF1 */
if(RCDF0 & (0x1 << 31)) flag = 1;
RCDF0 <<= 1; RCDF1 <<= 1; /* uno a pasar de RCDF1 a RCDF0 */
if(flag) RCDF1 |= 0x1;
if(RCDF1 & (0x1 << (L- 32))) {
RCDF0 |= 0x1;
RCDF1 &= 0x1F; /* L-32 unos */ }
if( ((OR0 & RCDF0) == RCDF0) && ((OR1 & RCDF1) == RCDF1) )
return (1); /* si hay absurdo */
} /* fin while =rotaciones */
} /* fin for=para cada CDF */
if(i == 18) {
ultoa(OR1, a, 2);
fprintf(fi, "\n OR1 =%05s", a);
ultoa(OR0, a, 2);
fprintf(fi, " OR0 =%032s", a);
ultoa(RCDF1, a, 2);
fprintf(fi, "\nRCDF1 =%05s", a);
ultoa(RCDF0, a, 2);
fprintf(fi, " RCDF0 =%032s l =%d p =%d", a, l, p); }
return (0); } /* fin subrutina */
```

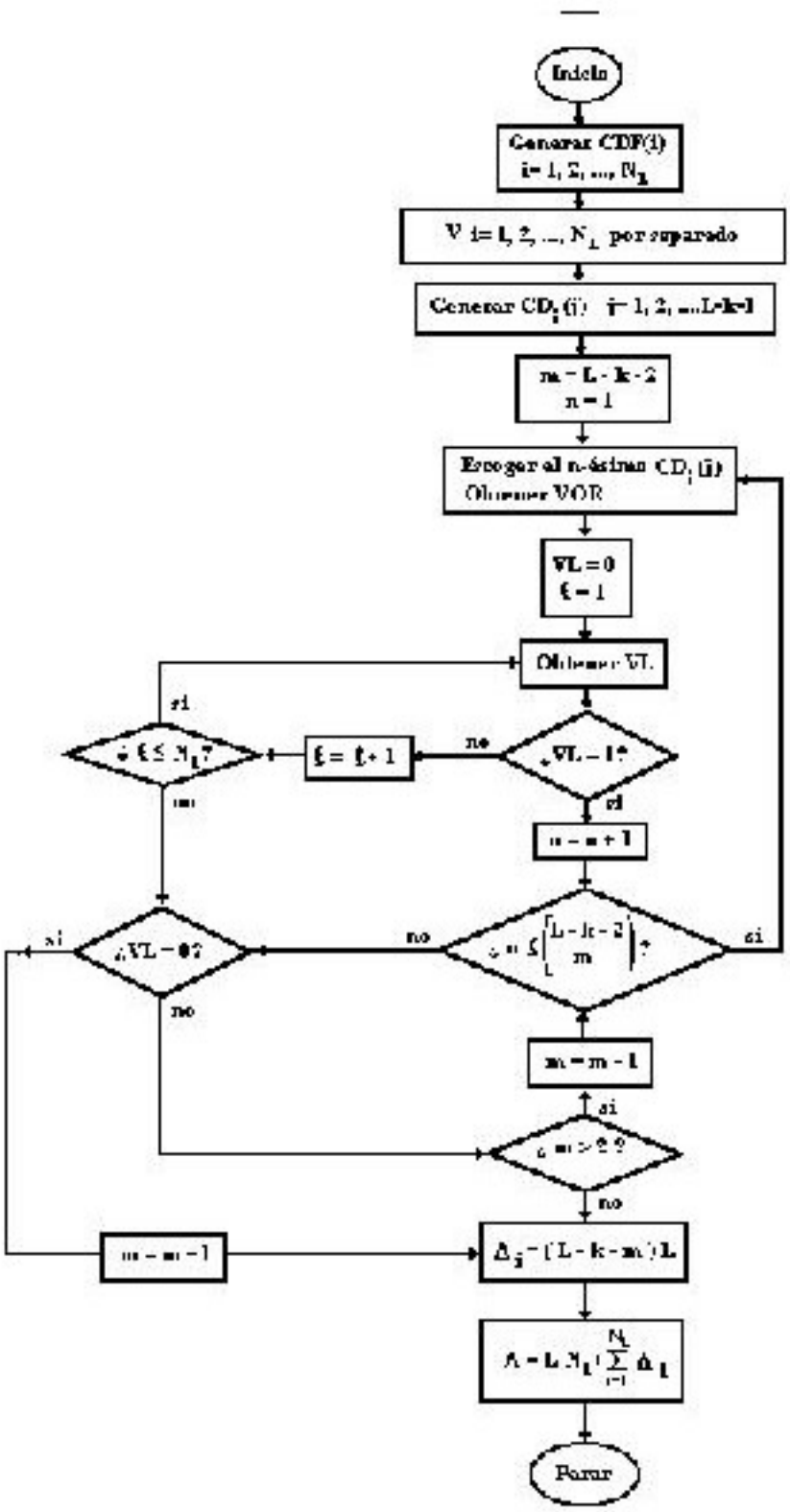


Figura 6.1: Diagrama de Flujo del Algoritmo 1

6.2. Algoritmo 2

{Programa realizado por Antonio Sedeño y Pino Caballero}
{Algoritmo 2 - Programa Principal}
{Objetivo: Obtener Cotas Inferiores de la Complejidad Lineal}

```
program cota; {programa principal}
uses hospital; {utiliza la unidad hospital}
type cd = ^ccdd;
ccdd = record;
valor:code; {code es un string de longitud 15}
sig:cd;
end;
vect = array[1..(nmax*8-1) div 2] of code;
{nmax=15} vect1 = array[1..(nmax*8+1) div 2 - 1] of code;
var primos,conter,L,K,d,tamano,i,j,m,mpri:integer; {tamano:n. char para
L}
delta:longint; {delta es donde se guarda la cota}
n:real; {n es el n. de combinaciones}
copia:code;
mem:word;
CDF:vect;
VCDM:array[1..(nmax*8-1) div 2] of vect1;
CDD:cd;
VL:boolean; {true si hay absurdo para todos false si no hay para alguno}
fichero:text; {fichero donde busca los L}
{***********************************************}
function p6(var a:code):boolean; {hace AND de a y cada CDF y rota-
ciones}
var b,bcop:code; {y devuelve 1 si alguna da absurdo}
i1,i2,j3,codigo,valor:integer;
gm,iguales:boolean;
begin
gm:=false; {true=absurdo}
i1:=1;
while (i1<= d) and not(gm) do {para todos los CDF mientras no haya
alguno}
begin {que d absurdo}
```

```
    for i2:=1 to tamano do {para todos los strings}
    begin
    codigo:=ord(CDF[i1][i2]);
    b[i2]:=chr(codigo); {en b pone el CDF}
    bcop[i2]:=b[i2]; {hace copia de b}
    end;
    iguales:=true;
    andl(bcop,a,tamano); {hace AND de bcop y a y lo pone en bcop}
    j3:=1;
    while (j3<=tamano) and iguales do {para todos los strings mientras sean
ig.}
    begin
    if bcop[j3]<>CDF[i1][j3] then {si un string de bcop es dist. del}
    iguales:=false; {de CDF pone iguales a false}
    j3:=j3+1; {siguiente string}
    end;
    gm:=iguales; {es false si para el CDF(i1) no coinciden}
    i2:=1;
    while (i2<l) and not (gm) do {L-1 rotaciones del CDF mientras sean
dist.}
    begin
    rotar1(b,tamano,l); {rota CDF 1 posición a izda y lo pone en b}
    for j3:=1 to tamano do
    bcop[j3]:=b[j3]; {hace copia de bcop}
    andl(bcop,a,tamano); {AND de bcop y a y lo pone en bcop}
    iguales:=true;
    j3:=1;
    while (j3<=tamano) and iguales do {para todo string mientras iguales}
    begin
    if bcop[j3]<>b[j3] then {si string de bcop es dist. del CDF rotado}
    iguales:=false;
    j3:=j3+1; {siguiente string}
    end;
    gm:=iguales; {false si para la rotación del CDF(i1) no coinciden}
    i2:=i2+1; {otra rotación}
    end;
    i1:=i1+1; {otro CDF}
    end;
```

```
p6:=gm; {variable lógica final que devuelve}
end;
{***********************************************}
procedure DFC(i1,j1:integer); {crea todos los CD para i1,j1, poniéndolos}
var nuevo,ultimo:cd; {en una lista donde cada uno apunta al sgte}
a:code;
uno:boolean;
j2,j3,codigo,position:integer;
begin
for j3:=1 to tamano do {para todo string que hacen falta según L}
begin
codigo:=ord(VCDM[i1][j1][j3]); {inicializa según CDM(i1,j1)}
a[j3]:=chr(codigo);
end;
if cdd<>nil then {inicialización de la lista}
begin
nuevo:=cdd;
while (cdd<>nil) do
begin
cdd:=cddˆ.sig;
dispose(nuevo);
nuevo:=cdd;
end;
end;
cdd:=nil;
ultimo:=cdd;
uno:=true;
for j2:=1 to l-k do {n de CD del grupo}
begin
position:= (i1*(k+j2-1) + 1) mod L; {posición nuevo 1 donde los 0 de
CDF}
if position=0 then
position:=L;
put(a,uno,position); {pone el 1}
new(nuevo);
for j3:=1 to tamano do nuevoˆ.valor[j3]:=a[j3]; {añade el CD creado a la
lista}
nuevoˆ.sig:=nil; {y hace que el último sea nil}
```

```
    if cdd=nil then
    cdd:=nuevo else
    ultimoˆ.sig:=nuevo;
    ultimo:=nuevo;
    for j3:=1 to tamano do
    begin
    codigo:=ord(VCDM[i1][j1][j3]);
    a[j3]:=chr(codigo);
    end;
    end;
    end;
    {************************************************}
    procedure borrar(var h:code); {borra un CD de la lista}
    var anterior,p:cd;
    cod:integer;
    encontrado,igual:boolean;
    begin encontrado:=false;
    anterior:=cdd; {1 de la lista}
    p:=cdd;
    while (p<>nil) and not(encontrado) do {mientras no llegue al final de
la}
    begin {lista y no lo haya encontrado}
    cod:=1;
    igual:=true;
    while (cod<=tamano) and igual do {para cada string siempre que den
ig}
    begin
    if ord(pˆ.valor[cod])<> ord(h[cod]) then {si algún string es dist.}
    igual:=false;
    cod:=cod+1; {siguiente string}
    end;
    if (igual) then {si después de revisar todos los strings fueron ig.}
    encontrado:=true
    else
    begin
    anterior:=p; {pasa al siguiente de la lista}
    p:=pˆ.sig;
    end;
```

```
end;
if encontrado then {si lo encontró}
begin
if anterior=p then {si era el 1 lo borra}
cdd:=cddˆ.sig
else
anteriorˆ.sig:=pˆ.sig; {se lo salta}
dispose(p);
end;
end;
{***********************************************}
procedure or_exc(i1,j1:integer); {para los CD de la lista i,j hace la OR-ex}
var a,b:code; {con los ant. CDF y saca de lista los malos}
codigo,j2,j3,j4,j5,j6:integer;
pp:cd;
borrado:boolean;
begin
pp:=cdd;
borrado:=false;
while pp<>nil do {mientras no llegue al final de la lista}
begin
for j2:=1 to i1 do {para j2 anteriores o iguales a i1}
begin
j3:=1;
j6:=k-1;
if j2=i1 then {en el mismo bloque nos quedamos en el anterior a j1}
j6:=j1-1;
while (j3<=j6) and not(borrado) do {los anteriores mientras no borrado}
begin
if odd(ord(VCDM[j2][j3][1])) then {siempre el 1 string es impar}
begin
for j4:=1 to tamano do
begin
codigo:=ord(VCDM[j2][j3][j4]);
a[j4]:=chr(codigo);
b[j4]:=a[j4]; {mete en a y en b el CDM(j2,j3)}
end;
exorl(a,ppˆ.valor,tamano); {hace OR-ex de a y CD de la lista}
```

```
if (unosolo(a,tamano)) then {si en esa OR-ex hay un 1 solo}
begin
borrar(ppˆ.valor); {lo saca de la lista}
borrado:=true; {para salir del while}
m:=m-1; {uno menos en la lista}
end
else
begin j5:=1;
borrado:=false;
while (j5 <=l-1) and not(borrado) do {L-1 rot. si no borrado}
begin
rotar1(b,tamano,l); {rota 1 a izda el CDM(j2,j3)}
for j4:=1 to tamano do
begin
codigo:=ord(b[j4]);
a[j4]:=chr(codigo); {mete el CDM rotado en a}
end;
exorl(a,ppˆ.valor,tamano); {ORex de a y CD de lista}
if (unosolo(a,tamano)) then {si tiene un 1 solo}
begin
borrar(ppˆ.valor); {lo saca de lista}
borrado:=true; {sale del while}
m:=m-1; {uno menos en la lista}
end;
j5:=j5+1; {otra rotación}
end; {while j5}
end; {else}
end; {if odd}
j3:=j3+1; {otro CDM}
end; {while j3}
end; {for j2}
pp:=ppˆ.sig; {otro CD de la lista}
borrado:=false;
end; {while pp}
end;
{************************************************}
procedure and_dfc; {hace AND de cada CD de la lista i,j y cada CDF y}
var a,b:code; {rotaciones y se queda con los buenos}
```

```
codigo,j3,j4,j5:integer;
pp:cd;
borrado,distinto:boolean;
begin
pp:=cdd;
borrado:=false;
while pp<>nil do {mientras no llegue al final de la lista}
begin
j3:=1;
while (j3<=d) and not(borrado) do {con cada CDF mientras no borrado}
begin
for j4:=1 to tamano do
begin
codigo:=ord(CDF[j3][j4]);
a[j4]:=chr(codigo);
b[j4]:=a[j4];
end;
andl(a,ppˆ.valor,tamano); {hace AND de CDF y CD de la lista}
distinto:=false;
j4:=1;
while (j4<=tamano) and not(distinto) do {mientras ningun string}
begin {sea distinto}
if ord(a[j4]) <> ord(b[j4]) then
distinto:=true;
j4:=j4+1;
end;
if not(distinto) then {si son iguales}
begin
borrar(ppˆ.valor); {lo saca de la lista}
borrado:=true; {para salir del while}
m:=m-1; {uno menos en la lista}
end else {en otro caso a rotar el CDF}
begin
j5:=1;
borrado:=false;
while (j5 <=l-1) and not(borrado) do {L-1 rot si no borrado}
begin
rotar1(b,tamano,l); {rota 1 a izda b}
```

```
for j4:=1 to tamano do {para cada string}
begin
codigo:=ord(b[j4]);
a[j4]:=chr(codigo); {mete el CDF rotado en a}
end;
andl(a,ppˆ.valor,tamano); {hace AND de a y CD}
distinto:=false;
j4:=1;
while (j4<=tamano) and not(distinto) do
begin
if ord(a[j4])<>ord(b[j4]) then
distinto:=true;
j4:=j4+1;
end;
if not(distinto) then {si dan iguales}
begin
borrar(ppˆ.valor); {lo saca de lista}
borrado:=true; {para salir del while}
m:=m-1;
end;
j5:=j5+1; {otra rotación}
end;
j3:=j3+1; {otro CDF}
end;
end; {while j3}
pp:=ppˆ.sig; {otro CD}
borrado:=false;
end;
end;
{************************************************}
procedure CDM(i,j:integer); {crea el CDM(i,j)}
var uno:boolean;
x,codigo,j1:integer;
begin
for j1:=1 to tamano do {para cada string}
begin
codigo:=ord(CDF[i][j1]); {inicializa el CDM(i,j) como el CDF(i)}
VCDM[i][j][j1]:=chr(codigo);
```

```
    end;
    uno:=false;
    x:= (i*j + 1) mod l; {la posición del nuevo 0}
    if x=0 then
    x:=L;
    put(VCDM[i][j],uno,x); {pone un 0 en posición x}
    end;
    procedure FDC(dist:integer); {crea el CDF(dist)}
    var i,j:integer;
    uno:boolean;
    begin
    uno := true;
    j := 1;
    i:=1;
    ponercero(CDF[dist],tamano);
    put(CDF[dist],uno,j); {coloca el primer 1 a dcha}
    if ((i+dist) mod L) <> 0 then {el segundo1 nunca está en posición L}
    i := (i + dist) mod L
    else
    i:=L;
    while (j < k) do {lo mismo para los k-1 unos restantes}
    begin
    put(CDF[dist],uno,i); {coloca j+1-simo 1 en posición i}
    j := j + 1;
    if ((i+dist) mod L) <>0 then
    i := (i + dist) mod L
    else
    i:=L;
    end;
    end;
    {************************************************}
    procedure combina(tamano2:integer;a:code;var bb:code); {haya OR del
grupo}
    var j2,j3:integer; {indicado por la codificación a y la devuelve en bb}
    acop,b,c:code;
    uno:boolean;
    p:cd;
    begin
```

```
ponercero(b,tamano);
j2:=1;
p:=cdd;
uno:=true;
while (p<>nil) do {recorre la lista buscando los que irán a la OR}
begin
ponercero(c,tamano2);
put(c,uno,j2); {c string con un 1 rotante para descubrir los de a}
for j3:=1 to tamano2 do
acop[j3]:=a[j3];
andl(acop,c,tamano2); {para descubrir los 1 de a}
if (j2 mod 8) <> 0 then
j3:=j2 div 8 +1
else
j3:=j2 div 8;
if (ord(acop[j3])=ord(c[j3])) then
orl(b,p^.valor,tamano); {mete en la OR el CD de la lista que toca}
p:=p^.sig; {siguiente CD de la lista}
j2:=j2+1; {para mover el 1 de c}
end;
for j3:=1 to tamano do
bb[j3]:=b[j3];
end;
{************************************************}
procedure p5(var esuno:boolean); {devuelve esuno=true si hay absurdo}
var cop,a:code; {a es la codificación para hacer la OR}
uno:boolean;
j4:integer; {n de string necesarios para una upla de longitud mpri}
procedure patron(i,contador:integer);
var j:integer;
begin
j:=i;
if contador<m then {los m 1 a colocar}
begin
while (j<=mpri) and (esuno) do {recorre las mpri}
begin {posiciones mientras esuno sea true}
uno:=true; {para que coloque un 1}
put(a,uno,j); {coloca un 1 en posición j de a}
```

```
patron(j+1,contador+1); {procedimiento recursivo}
uno:=false; {para que coloque un 0}
put(a,uno,j); {coloca un 0 en posición j de a}
j:=j+1; {movimiento a izda del último 1}
end
end
else
begin combina(j4,a,cop); {con codificación j4 da la OR en cop}
esuno:=p6(cop); {variable que indica absurdo}
end;
end;
begin
if (mpri mod 8) <> 0 then {averigua j4}
j4:= mpri div 8 +1
else
j4:= mpri div 8;
ponercero(a,j4); {pone a 0 todo a[1],...,a[j4]}
patron(1,0); {calcula codificaciones, ORES y var lg que indica absurdo}
end;
{************************************************}
function combi(u,v:integer):real; {haya n combinatorio (u v)}
var h1:real;
g:integer;
begin
h1:=1;
if ((u-v) > u) then
begin
for g:=1 to v do
h1 := trunc(((u-g+1)/g)*h1);
{u/1(u-1)/2...(u-v+1)/v}
end
else
begin
for g:=1 to (u-v) do
h1 := trunc(((u-g+1)/g)*h1); {u/1(u-1)/2...(v+1)/(u-v)}
end;
combi := h1;
end;
```

```
{***********************************************}
begin {comienza el cuerpo del programa principal}
assign(fichero,'primos.dat'); {abre el fichero donde están los primos L}
reset(fichero);
readln(fichero,primos); {lee primera línea donde dice cuántos primos analizar}
for conter:=1 to primos do {para tantos primos con que vayamos a tra-
bajar}
begin
readln(fichero,L); {lee 1 primo como L}
k := (L + 1) div 2;
d := (L - 1) div 2; {n de CDF que hay}
delta := (L - 1) * L div 2; {valor inicial de la cota}
if (L mod 8) <> 0 then {calcula el valor de tamano}
tamano := (L div 8) + 1
else
tamano := (L div 8);
for i:=1 to d do {bucle de creación de los CDF}
FDC(i);
cdd:=nil;
for i:=1 to d do {1 bucle de trabajo, para cada CDF}
for j:=1 to k-1 do {2 bucle de trabajo, para cada CDM}
begin
CDM(i,j); {creación del CDM(i,j)}
m:=L-K; {inicializa el tamaño de la lista}
DFC(i,j); {crea todos los correspondientes CD para i,j}
and_dfc; {limpia lista con las AND con los CDF}
or_exc(i,j); {limpia lista con las ORex con los CDM anteriores}
mpri:=m; {tamaño de la lista que queda}
vl:=true;
while (vl) and (m<>0) do {mientras todos los grupos para un m den}
begin {absurdo y n de uplas de los grupos no sea 0}
p5(vl); {p5 y p6 hasta que algún gm sea 0 todos sean 1}
if (VL) then {si todos los grupos dan absurdo}
begin
m:=m-1; {considerar grupos con una upla menos}
if m < 2 then {si sólo quedan por estudiar grupos de 1 upla}
begin
delta:=delta + (mpri-m)*L; {incrementa cota}
```

```
vl:=false; {para salir del while}
end
end
else {si todavía es m>1}
begin if (mpri=m) then {si esto ocurre con el grupo de mpri uplas}
ponercero(vCDM[i][j],tamano) {elimina el CDM(i,j)}
else
delta:=delta + (mpri-m)*l; {incrementa cota}
end;
end; {while vl}
end; {for j}
writeln(l,' –> ',delta); {escribe valor final de la cota para cada L}
end; {for conter}
end. {cuerpo del programa principal}
```

{Unidad Hospital del Programa del Algoritmo 2}

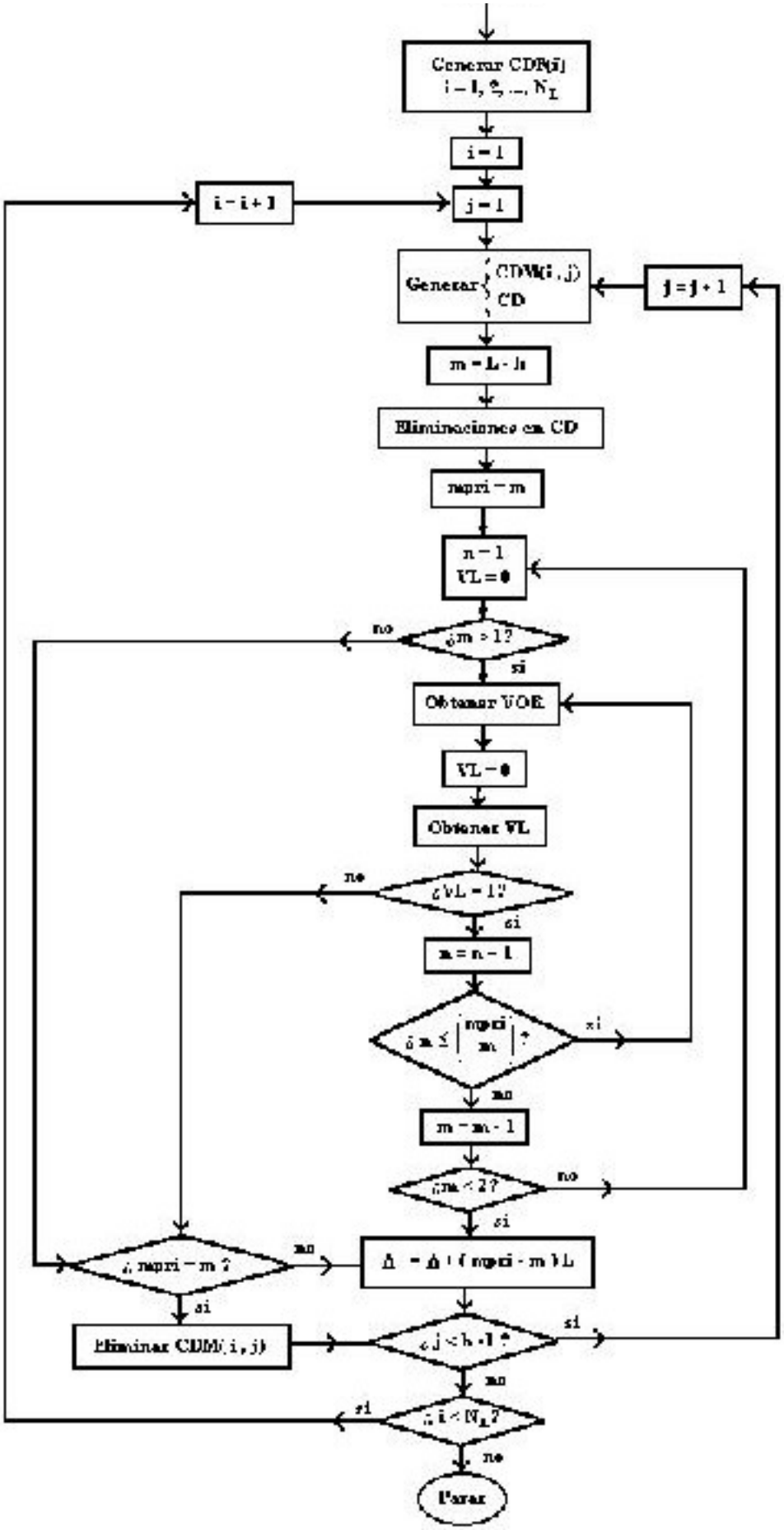
```
unit hospital;
INTERFACE const
nmax=15;
type code=string[nmax];
procedure ponercero(var a:code;ll:integer);
procedure poneruno(var a:code;ll:integer);
function landa (j:integer):integer;
procedure put(var a:code;valor:boolean;position:integer);
procedure exorl(var a,b:code;ll:integer);
procedure andl(var a,b:code;ll:integer);
procedure orl(var a,b:code;ll:integer);
procedure notl(var a:code;ll:integer);
procedure rotar1(var a:code;tamano1,ll:integer);
function unosolo(var a:code;ll:integer):boolean;
IMPLEMENTATION
{************************************************}
procedure ponercero(var a:code;ll:integer); {pone a 0 todos los ll strings
a}
var j:integer;
begin
for j:=1 to ll do
a[j]:=chr(0);
end;
procedure poneruno(var a:code;ll:integer); {pone a 1 todos los ll strings
a}
var j:integer;
begin
for j:=1 to ll do
a[j]:=chr(255);
end;
{************************************************}
function landa (j:integer):integer; {obtiene 2 elevado a j}
var parcial:longint;
i:integer;
begin
parcial:=1;
```

```
    i:=1;
    for i:=1 to j do
    parcial:=2*parcial;
    landa:=parcial;
    end;
    {************************************************}
    procedure put(var a:code;valor:boolean;position:integer); {pone un 1 en}
    var i,j,codigo:integer; {posición de a si valor es true, si no pone un 0}
    begin
    if (position mod 8) <> 0 then {según position se pone en el string corre-
sp.}
    begin
    j:=(position div 8) + 1;
    i:=(position mod 8);
    end
    else
    begin
    j:=(position div 8);
    i:=8;
    end;
    i:=landa(i-1);
    if valor then {poner un 1}
    begin
    codigo:=(ord(a[j]) or i);
    a[j]:=chr(codigo); {pone un 1}
    end
    else {poner un 0}
    begin
    codigo:=(ord(a[j]) and not(i)); {pone un 0}
    a[j]:=chr(codigo);
    end;
    end;
    {************************************************}
    procedure exorl(var a,b:code;ll:integer); {hace ORex de a y b y lo pone
en a}
    var j,codigo:integer;
    begin j:=1;
    while j<=ll do {string por string}
```

```
    begin
    codigo:= ord(a[j]) xor ord(b[j]);
    a[j]:=chr(codigo);
    j:=j+1;
    end;
    end;
    {*************************************************}
    procedure andl(var a,b:code;ll:integer); {hace AND de a y b y lo pone en
a}
    var j,codigo:integer;
    begin
    j:=1;
    while j<=ll do {string por string}
    begin
    codigo:=ord(a[j]) and ord(b[j]);
    a[j]:=chr(codigo); {lo pone en a}
    j:=j+1; {siguiente string}
    end;
    end;
    {*************************************************}
    procedure orl(var a,b:code;ll:integer); {hace OR de a y b y lo pone en a}
    var j,codigo:integer;
    begin
    j:=1;
    while j<=ll do
    begin
    codigo:=ord(a[j]) or ord(b[j]);
    a[j]:=chr(codigo);
    j:=j+1;
    end;
    end;
    {*************************************************}
    procedure notl(var a:code;ll:integer); {complementario de a y lo pone en
a}
    var j,codigo:integer;
    begin
    j:=1;
    while j<=ll do
```

```
    begin
    codigo:=not(ord(a[j])); {complementario de a}
    a[j]:=chr(codigo);
    j:=j+1;
    end;
    end;
    {************************************************}
    procedure rotar1(var a:code;tamano1,ll:integer); {rota 1 a izda a que mide
ll}
    var ii,jj,bdesp,valor,codigo:integer; {y son tamano1 strings}
    begin
    jj:=tamano1;
    ii:=jj;
    while jj>=1 do {para los tamano1 strings}
    begin
    if ii=jj then {para el último string}
    begin
    valor:=landa(8-(8*ii-ll)-1); {1 en última posición izquierda}
    bdesp:= (valor and ord(a[jj]))
    div valor; {el 1 que se sale por izda}
    codigo:= (ord(a[jj])) and not(valor); {lo coloca en string a dcha}
    a[jj]:=chr(codigo);
    end;
    if jj<>1 then {para todos los strings menos el 1}
    begin
    valor:=landa(7);
    codigo := ord(a[jj]) shl 1 or (valor and ord(a[jj-1]) div valor);
    a[jj]:=chr(codigo); {rota string 1 a izda y añade 1 del string anter}
    end
    else {para el 1}
    begin
    codigo:= ord(a[jj]) shl 1 or bdesp; {a izda y añade 1 a dcha si debe}
    a[jj]:=chr(codigo);
    end;
    jj:=jj-1; {string anterior}
    end;
    end;
    {************************************************}
```

```
function unosolo(var a:code;ll:integer):boolean; {ve si hay un 1 solo en a}
var i,j,j1,codigo,valor,contador:integer;
solo:boolean;
begin
i:=ll;
j:=1;
solo:=true;
contador:=0;
while (j<=i) and (solo) do {para cada string mientras haya visto sólo un
1}
begin
codigo:=(ord(a[j]));
j1:=1;
while (j1<=8) and (solo) do {en cada string mientras un 1 solo}
begin
valor:=landa(j1-1);
if ((valor and codigo) = valor) then {si está el 1 en posición j1-1}
begin contador:=contador+1; {aumenta contador}
if contador > 1 then {cuando haya + de 1 solo a false}
solo:=false;
end;
j1:=j1+1;
end;
j:=j+1;
end;
unosolo:=solo;
end;
begin
end.
```


Generar CDR(i)
i = 1, 2, ..., N_L
i = 1
i = i + 1
j = 1
Generar CDW(i, j) CD
j = j + 1
m = L - ls
Eliminaciones en CD
nopri = m
n = 1
VL = ∅
no
¿m > 1?
si
Obtener VOE
VL = ∅
Obtener VL
no
¿VL = 1?
si
n = n - 1
si
no
m = m - 1
no
¿m < 2?
si
no
¿ nopri = m ?
si
Δ = Δ + (nopri - m) L
Eliminar CDW(i, j)
si
¿ j < h - 1 ?
no
si
¿ i < N_L ?
no
Parar

Bibliografía

[1] A.G. Akritas, Elements of Computer Algebra, John Wiley & Sons, 1989.

[2] P. Arnoux, C. Mauduit, I. Shiokawa, J.I. Tamura, Complexity of Sequences Defined by Billiard in the Cube, Bulletin de la Société Mathématique de France, Vol. 122, 1994.

[3] H. Beker, F. Piper, Cipher Systems, the Protection of Communications, Northwood Publications, 1982.

[4] E.R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New York, 1968.

[5] J. Bernasconi, C.G. Günther, Analysis of a Nonlinear Feedforward Logic for Binary Sequences Generators, IEEE Transactions on Information Theory, 1985.

[6] T. Beth, F.C. Piper, The Stop-and-Go Generator, Advances in Cryptology-EUROCRYPT'84, Lecture Notes in Computer Science No. 209, Springer-Verlag, 1985.

[7] R.E. Blahut, Theory and Practice of Error Control Codes, Addison-Wesley, 1983.

[8] G.R. Blakley, One-Time Pads are Key Safeguarding Schemes, Not Cryptosystems, Proceedings of the 1980 Symposium on Security and Privacy, IEEE Computer Society, 1980.

[9] S. Boztas, V. Kumar, Binary Sequences with Gold-Like Correlation but Larger Linear Span, IEEE Transactions on Information Theory, Vol. 40, No. 2, March 1994.

[10] L. Brynielsson, A Short Proof of the Xiao-Massey Lemma, IEEE Transactions on Information Theory, Vol. IT-35, Nov. 1989.

[11] J.O. Brüer, On Pseudo Random Sequences as Crypto Generators, Proceedings of International Zurich Seminary on Digital Communications, Zurich, Switzerland, 1984.

[12] P. Caballero Gil, Criptología Digital: Una Introducción Matemática, Tesina de Licenciatura, Universidad de La Laguna, 1992.

[13] P. Caballero Gil, C. Bruno Castañeda, Uso Didáctico de la Criptografía: La Administración de Secretos, SUMA, No. 19, 1995.

[14] P. Caballero Gil, A. Fúster Sabater, Equivalente Lineal Descompuesto del Filtrado no Lineal y Resultados sobre Complejidad Lineal, Revista de la Academia Canaria de Ciencias, Vol. VI, No. 1, 1994.

[15] P. Caballero Gil, A. Fúster Sabater, Algoritmo de Cálculo de una Cota Inferior de la Complejidad Lineal del Filtrado no Lineal, Libro de Actas de la III Reunión Española sobre Criptología, Barcelona, 1994.

[16] P. Caballero Gil, A. Fúster Sabater, Un Algoritmo para Medir la Seguridad Práctica de Algunos Cifrados en Flujo, Actas de las I Jornadas de Informática, Puerto de la Cruz, 1995.

[17] P. Caballero, A. Fúster, Lower Bounds for the Global Linear Complexity of Certain Running Keys, enviado a IEE Proceedings Computers and Digital Techniques, 1995.

[18] P. Caballero, A. Sedeño, A. Fúster, COTA: Un Algoritmo de Cálculo de la Impredecibilidad de Algunas Secuencias Binarias de Aplicación Criptográfica, enviado a Informática y Automática, 1995.

[19] N.P. Cagigal, S. Bracho, Algorithmic Determination of Linear-Feedback in a Shift Register for Pseudorandom Binary Sequence Generation, IEE Proceedings, Vol. 133, Pt. G, No. 4, August 1986.

[20] G. Carter, Some Conditions on the Linear Complexity Profiles of Certain Binary Sequences, Advances in Cryptology-EUROCRYPT'89, Lecture Notes in Computer Science No. 434, Springer-Verlag, 1990.

[21] G. Carter, Enumeration Results on Linear Complexity Profiles, Collection: Cryptography and Coding, II, Oxford University Press, 1992.

[22] W.G. Chambers, Z.D. Dai, On Binary Sequences from Recursions 'modulo 2^e' Made Non-Linear by the bit-by-bit 'XOR' Function, Advances in Cryptology-EUROCRYPT'91, Lecture Notes in Computer Science No. 547, Springer-Verlag, 1991.

[23] W.G. Chambers, D. Gollman, Generators for Sequences with Near-Maximal Linear Equivalence, IEE Proceedings, Vol. 135, Pt. E, No. 1, Jan. 1988.

[24] W.G. Chambers, S.M. Jennings, Linear Equivalence of Certain BRM Shift-Register Sequences, Electronics Letters, No. 20, Nov. 1984.

[25] A.H. Chan, On the Quadratic Spans of Periodic Sequences, Advances in Cryptology-CRYPTO'88, Lecture Notes in Computer Science No. 403, Springer-Verlag, 1990.

[26] A.H. Chan, M. Goresky, A. Klapper, On the Linear Complexity of Feedback Registers, Advances in Cryptology-EUROCRYPT'89, Lecture Notes in Computer Science No. 434, Springer-Verlag, 1990.

[27] Z.D. Dai, Proof of Rueppel's Linear Complexity Conjecture, IEEE Transactions on Information Theory, Vol. IT-32, No. 3, 1986.

[28] Z.D. Dai, Linear Complexity of Periodically Repeated Random Sequences, Advances in Cryptology-EUROCRYPT'91, Lecture Notes in Computer Science No. 547, Springer-Verlag, 1991.

[29] Z.D. Dai, Binary Sequences Derived from ML-Sequences over Rings.I: Period and Minimal Polynomials, Journal of Cryptology, Vol. 5, No. 3, 1992.

[30] Z.D. Dai, T. Beth, D. Gollmann, Lower Bounds for the Linear Complexity of Sequences over Residue Rings, Advances in Cryptology-EUROCRYPT'90, Lecture Notes in Computer Science No. 473, Springer-Verlag, 1991.

[31] I.B. Damgard (Ed.), Advances in Cryptology-EUROCRYPT'90, Lecture Notes in Computer Science No. 473, Springer-Verlag, 1991.

[32] J.H. Davenport, Y. Siret, E. Tournier, Computer Algebra, Academic Press, 1988.

[33] D.W. Davies (Ed.), Advances in Cryptology-EUROCRYPT'91, Lecture Notes in Computer Science No. 547, Springer-Verlag, 1991.

[34] W.A. Davis, Generation of Delayed Replicas of Maximal Length Linear Binary Sequences, Proceedings of the IEEE, Vol.113, Feb. 1966.

[35] E. Dawson, B. Goldburg, Universal Logic Sequences, Advances in Cryptology-AUSCRYPT'90, Lecture Notes in Computer Science No. 453, Springer-Verlag, 1990.

[36] D.E. Denning, Cryptography and Data Security, Addison-Wesley, 1983.

[37] W. Diffie, M.E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. IT-22, Nov. 1976.

[38] W. Diffie, M.E. Hellman, Privacy and Authentication: An Introduction to Cryptography, Proceedings of the IEEE, Vol. 67, March 1979.

[39] C. Ding, Lower Bounds on the Weight Complexities of Cascaded Binary Sequences, Advances in Cryptology-AUSCRYPT'90, Lecture Notes in Computer Science No. 453, Springer-Verlag, 1990.

[40] P.F: Duvall, Decimation of Periodic Sequences, SIAM Journal of Applied Mathematics, Vol.21, No. 3, Nov. 1971.

[41] P.F. Duvall, R.R. Kibler, On the Parity of the Frequency of Cycle Lengths of Shift Register Sequences, Journal of Combinatorial Theory(A), Vol. 18, No. 3, May 1975.

[42] J. Eichenauer Herrmann, Statistical Independence of a new Class of Inversive Congruential Pseudorandom Numbers, Mathematics of Computation, Vol. 60, No. 201, Jan. 1993.

[43] J. Eichenauer Herrmann, On Generalized Inversive Congruential Pseudorandom Numbers, Mathematics of Computation, Vol. 63, No. 207, July 1994.

[44] M.C. Espinel Febles, P. Caballero Gil, La Matemática que Protege de Errores a los Números de Identificación, a ser publicado en SUMA, 1995.

[45] J. Feigenbaum (Ed.), Advances in Cryptology-CRYPTO'91, Lecture Notes in Computer Science No. 576, Springer-Verlag, 1991.

[46] W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, John Wiley, 1968.

[47] R. Forré, The Strict Avalanche Criterion: Spectral Properties of Boolean Functions and an Extended Definition, Advances in Cryptology-CRYPTO'88, Lecture Notes in Computer Science No. 403, Springer-Verlag, 1990.

[48] R. Forré, A Fast Correlation Attack on Nonlinearly Feedforward Filtered Shift-Register Sequences, Advances in Cryptology-EUROCRYPT'89, Lecture Notes in Computer Science, Springer-Verlag, 1990.

[49] J.N. Franklin, S.W. Golomb, A function-Theoretic Approach to the Study of Nonlinear Recurring Sequences, Pacific Journal of Mathematics, Vol. 56, No. 2, 1975.

[50] S. Fredricsson, Pseudo-Randomness Properties of Binary Shift Register Sequences, IEEE Transactions on Information Theory, Vol. IT-21, 1975.

[51] A. Fúster Sabater, P. Caballero Gil, On the Linear Complexity of Nonlinearly Filtered PN-Sequences, Advances in Cryptology-ASIACRYPT'94, Lecture Notes in Computer Science No. 917, Springer-Verlag, pp.80-90, 1995.

[52] A. Fúster Sabater, P. Caballero Gil, New Bounds for the Equivalent Linear Span of a Class of Nonlinear Sequences, enviado a IEEE Transactions on Information Theory, 1995.

[53] A. Fúster, D. de la Guía, J. Negrillo, F. Montoya, Diseño e Implementación de Algoritmos de Generación de Secuencias Binarias, Libro de Actas de la I Reunión Española sobre Criptología, Palma de Mallorca, 1991.

[54] A. Fúster, F. Montoya, Generador no Lineal de Secuencias Binarias Seudoaleatorias, Patente de Invención No. 8703195, Nov. 1987.

[55] M.R. Garey, D.S. Johnson, Computers and Intractability, a Guide to the Theory of NP-Completeness, W.H. Freeman and Co., San Francisco, California, 1979.

[56] P.R. Geffe, How to Protect Data with Ciphers that are Really Hard to Break, Electronics, Jan. 4, 1973.

[57] B.R. Gelbaum, Linear Algebra, North-Holland, 1989.

[58] K.R. Godfrey, Three-Level m-Sequences, Electronics Letters, Vol. 2, 1966.

[59] T. Goka, An Operator on Binary Sequences, SIAM Review, Vol. 12, 1970.

[60] R. Gold, Characteristic Linear Sequences and Their Coset Functions, SIAM Journal on Applied Mathematics, Vol. 14, 1966.

[61] J.D. Golic, The Number of Output Sequences of a Binary Sequence Generator, Advances in Cryptology-EUROCRYPT'91, Lecture Notes in Computer Science No. 547, Springer-Verlag, 1991.

[62] J.D. Golic, M.J. Mihaljevic, A Noisy Clock-Controlled Shift Register Cryptanalysis Concept Based on Sequence Comparison Approach, Advances in Cryptology-EUROCRYPT'90, Lecture Notes in Computer Science No. 473, Springer-Verlag, 1991.

[63] D. Gollman, Pseudo Random Properties of Cascade Connections of Clock Controlled Shift Registers, Advances in Cryptology-EUROCRYPT'84, Lecture Notes in Computer Science No. 209, Springer-Verlag, 1985.

[64] S.W. Golomb, On Certain Nonlinear Recurring Sequences, American Mathematical Monthly, Vol. 70, No. 4, April, 1963.

[65] S.W. Golomb, Cyclotomic Polynomials and Factorization Theorems, American Mathematical Monthly, Vol. 85, No. 9, Nov. 1978.

[66] S.W. Golomb, Obtaining Specified Irreducible Polynomials Over Finite Fields, SIAM Journal on Algebraic and Discrete Methods, Vol. 1, No. 4, Dec. 1980.

[67] S.W. Golomb, On the Classification of Balanced Binary Sequences of Period 2^n-1, IEEE Transactions on Information Theory, Vol. IT-26, 1980.

[68] S.W. Golomb, Shift Register Sequences, Holden-Day, San Francisco, California, 1967. Revised edition, Aegean Park Press, Laguna Hills, California, 1982.

[69] R. Gottfert, H. Niederreiter, On the Linear Complexity of Products of Shift-Register Sequences, Advances in Cryptology-EUROCRYPT'93, Lecture Notes in Computer Science, Springer-Verlag, 1994.

[70] E. Grosswald, Topics from the Theory of Numbers, 2nd Edition, Birkhäuser,1984.

[71] E.J. Groth, Generation of Binary Sequences with Controllable Complexity, IEEE Transactions on Information Theory, Vol. IT-17, May 1971.

[72] G. Guang, Nonlinear Generators of Binary Sequences with Controllable Complexity and Double Key, Advances in Cryptology-AUSCRYPT'90, Lecture Notes in Computer Science No. 453, Springer-Verlag, 1990.

[73] C.G. Günther, Parallel Generation of Recurring Sequences, Advances in Cryptology-EUROCRYPT'89, Lecture Notes in Computer Science, Springer-Verlag, 1990.

[74] J.T. Harvey, Delay Line in Shift Register Speeds m-Sequence Generation, Electronics, Vol. 48, No. 24, Nov. 1975.

[75] F. Hemmati, D. Costello, An Algebraic Construction for Q-ary Shift Register Sequences, IEEE Transactions on Computers, Vol. C-27, No. 12, Dec. 1978.

[76] T. Herlestam, On the Complexity of Functions of Linear Shift Register Sequences, International Symposium on Information Theory, Les Arc, France, 1982.

[77] C. Hua, G.Z. Xiao, The Linear Complexity of Binary Sequences with Period $(2^n - 1)^k$, IEEE Transactions on Information Theory, Vol. 37, No. 3, May 1991.

[78] C.J.A. Jansen, On the Construction of run Permuted Sequences, Advances in Cryptology-EUROCRYPT'90, Lecture Notes in Computer Science No. 473, Springer-Verlag, 1991.

[79] C.J.A. Jansen, D.E. Boekee, The Shortest Feedback Shift Register that can Generate a given Sequence, Advances in Cryptology-CRYPTO'89, Lecture Notes in Computer Science No. 435, Springer-Verlag, 1990.

[80] C.J.A. Jansen, D.E. Boekee, A Binary Sequence Generator Based on Ziv-Lempel Source Coding, Advances in Cryptology-AUSCRYPT'90, Lecture Notes in Computer Science No. 453, Springer-Verlag, 1990.

[81] H.R. Jordan, D.C.M. Wood, On the Distribution of Sums of Succesive Bits of Shift-Register Sequences, IEEE Transactions on Computers, Vol. C-22, No. 4, April, 1973.

[82] R.J. Kabanagh, Fourier Analysis of Pseudo-Random Binary Sequences, Electronics Letters, Vol. 5, Part I, No. 7, April 1969.

[83] N. Kalouptsidis, M. Manolarakis, Sequences of Linear-Feedback Shift Registers with Nonlinear-Feedforward Logic, IEE Proceedings, Vol. 130, Pt. E, No. 5, Sept. 1983.

[84] C. Kao, J.Y. Wong, Several Extensively Tested Random Number Generators, Computers Operations Research, Vol. 21, No. 9, 1994.

[85] E.L. Key, An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators, IEEE Transactions on Information Theory, Vol. IT-22, Nov. 1976.

[86] K. Kjeldsen, On the Cycle Structure of a Set of Nonlinear Shift Registers with Symmetric Feedback Functions, Journal of Combinatorial Theory, Ser. A, Vol. 20, 1976.

[87] A. Klapper, The Vulnerability of Geometric Sequences Based on Fields of Odd Characteristic, Journal of Cryptology, Vol. 7, No. 1, 1994.

[88] D.E. Knuth, The Art of Computer Programming, Vol. 2: Seminumerical Algorithms, Addison-Wesley, 1981.

[89] H. Krawczyk, How to Predict Congruential Generators, Journal of Algortihms, Vol. 13, 1992.

[90] P.V. Kumar, R.A. Scholtz, Bounds on the Linear Span of Bent Sequences, IEEE Transactions on Information Theory, Vol. IT-29, Nov. 1983.

[91] R.T.C. Kwok, M. Beale, Aperiodic Linear Complexities of de Bruijn Sequences, Advances in Cryptology-CRYPTO'88, Lecture Notes in Computer Science No. 403, Springer-Verlag, 1990.

[92] X. Lai, Condition for the Nonsingularity of a Feedback Shift-Register over a General Finite Field, IEEE Transactions on Information Theory, Vol. IT-33, Sept. 1987.

[93] X. Lai, R.A. Rueppel, J. Woollven, A Fast Cryptographic Checksum Algorithm Based on Stream Ciphers, Advances in Cryptology-AUSCRYPT'92, Lecture Notes in Computer Science, Springer-Verlag, 1993.

[94] S. Lang, Undergraduate Algebra, Springer-Verlag, 1990.

[95] J. Lee, D.R. Smith, Families of Shift-Register Sequences with Impulsive Correlation Properties, IEEE Transactions on Information Theory, Vol. IT-20, 1974.

[96] A. Lempel, On k-Stable Feedback Shift Registers, IEEE Transactions on Computers, Vol. 18, July 1969.

[97] A. Lempel, Analysis and Synthesis of Polynomials and Sequences over GF(2), IEEE Transactions on Information Theory, Vol IT-17, 1971.

[98] A. Lempel, M. Cohn, W.L. Eastman, A Class of Balanced Binary Sequences with Optimal Autocorrelation Properties, IEEE Transactions on Information Theory, Vol. IT-23, 1977.

[99] A. Lempel, W.L. Eastman, High Speed Generation of Maximal Length Sequences, IEEE Transactions on Computers, Vol. C-20, No. 2, Feb. 1971.

[100] A. Lempel, J. Ziv, On the Complexity of Finite Sequences, IEEE Transactions on Information Theory, Vol. IT-22, Nov. 1976.

[101] R. Lidl, H. Niederreiter, Introduction to Finite Fields and Their Applications, Cambridge University Press, 1986.

[102] B.W. Lindgren, Statistical Theory, 3rd Edition, MacMillan, 1976.

[103] J.H. Lindholm, An Analysis of the Pseudo-Randomness Properties of Subsequences of Long m-Sequences, IEEE Transactions on Information Theory, Vol. IT-14, 1968.

[104] C.L. Liu, C. Tseng, Complementary Sets of Sequences, IEEE Transactions on Information Theory, Vol. IT-18, Sept. 1972.

[105] S. Lloyd, Properties of Binary Functions, Advances in Cryptology-EUROCRYPT'90, Lecture Notes in Computer Science No. 473, Springer-Verlag, 1991.

[106] G. Longo, M. Marchi, Geometries, Codes and Cryptography, Springer-Verlag, 1990.

[107] J.H. Loxton, Number Theory and Cryptography, Cambridge University Press, 1990.

[108] F.J. MacWilliams, N.J.A. Sloane, Pseudo-Random Sequences and Arrays, Proceedings of the IEEE, Vol.64, 1976.

[109] F.J. MacWilliams, N.J.A. Sloane, The Theory of Error-Correcting Codes, North-Holland, 1983.

[110] D.G. Maritsas, On the Statistical Properties of a Class of Linear Product Feedback Shift-Register Sequences, IEEE Transactions on Computers, Vol. C-22, No. 10, 1973.

[111] J.L. Massey, Shift-Register Synthesis and BCH Decoding, IEEE Transactions on Information Theory, Vol. IT-15, Nov. 1969.

[112] J.L. Massey, Cryptography: Fundamentals and Applications, ATS Seminar, Zürich, Switzerland, 1994.

[113] J.L. Massey, Where do we Stand Today in Cryptography?, Libro de Actas de la III Reunión Española sobre Criptología, Barcelona, 1994.

[114] J.L. Massey, U. Maurer, M. Wang, Non-Expanding, Key-Minimal, Robustly-Perfect, Linear and Bilinear Ciphers, Advances in Cryptology-EUROCRYPT'87, Lecture Notes in Computer Science No. 304, Springer-Verlag, 1988.

[115] J.L: Massey, R.A. Rueppel, Linear Ciphers and Random Sequence Generators with Multiple Clocks, Advances in Cryptology-EUROCRYPT'84, Lecture Notes in Computer Science No. 209, Springer-Verlag, 1985.

[116] J.L. Massey, S. Serconek, A Fourier Transform Approach to the Linear Complexity of Nonlinearly Filtered Sequences, Advances in Cryptology-CRYPTO'94, Lecture Notes in Computer Science No. 839, Springer-Verlag, 1995.

[117] U.M. Maurer, A Universal Statistical test for Random Bit Generators, Journal of Cryptology, Vol. 5, No. 2, 1992.

[118] U.M. Maurer, J.L. Massey, Perfect Local Randomness in Pseudo-Random Sequences, Advances in Cryptology-CRYPT0'89, Lecture Notes in Computer Science No. 435, Springer-Verlag, 1990.

[119] U.M. Maurer, J.L. Massey, Local Randomness in Pseudorandom Sequences, Journal of Cryptology, Vol. 4, No. 2, 1991.

[120] U.M. Maurer, J.L. Massey, Cascade Ciphers: The Importance of Being First, Journal of Cryptology, Vol. 6, No. 1, 1993.

[121] K.S. McCurley, Irregularities in the Distribution of Irreducible Polynomials, Proceedings of the AMS, Vol. 117, No. 1, Jan. 1993.

[122] R.J. McEliece, On Periodic Sequences from GF(q), Journal on Combinatorial Theory, Series A, Vol. 10, 1971.

[123] W. Meier, O. Staffelbach, Nonlinearity Criteria for Cryptographic Functions, Advances in Cryptology-EUROCRYPT'89, Lecture Notes in Computer Science, Springer-Verlag, 1990.

[124] W. Meier, O. Staffelbach, Correlation Properties of Combiners with Memory in Stream Ciphers, Journal of Cryptology, Vol. 5, No. 1, 1992.

[125] A.J. Menezes (Ed.), Applications of Finite Fields, Kluwer Academic Publishers, 1993.

[126] S. Micali, C.P. Schnorr, Efficient, Perfect Random Number Generators, Advances in Cryptology-CRYPTO'88, Lecture Notes in Computer Science No. 403, Springer-Verlag, 1990.

[127] M.J. Mihaljevic, J.D. Golic, A Fast Iterative Algorithm for a Shift Register Initial State Reconstruction Given the Noisy Output Sequence, Advances in Cryptology-AUSCRYPT'90, Lecture Notes in Computer Science No. 453, Springer-Verlag, 1990.

[128] C. Mitchell, Enumerating Boolean Function of Cryptographic Significance, Journal of Cryptology, Vol. 2, No. 3, 1990.

[129] F. Montoya, Estudios de Diseño de Redes Militares en los Aspectos de Conversión A/D y D/A, Cátedra del Grupo XXII Electroacústica y TV, Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Madrid, 1984.

[130] M. Morii, M. Kasahara, Perfect Staircase Profile of Linear Complexity for Finite Sequences, Information Processing Letters, Vol. 44, No. 2, 1992.

[131] F.J. Mowle, Relations Between PN Cycles and Stable FSR, IEEE Transactions on Electronic Computers, Vol. EC-15, No. 3, June 1966.

[132] F.J. Mowle, An Algorithm for Generating Stable FSR of Order n. Journal of the ACM, Vol. 14, No. 3, July 1967.

[133] S. Mund, Ziv-Lempel Complexity for Periodic Sequences and its Cryptographihc Application, Advances in Cryptology-EUROCRYPT'91, Lecture Notes in Computer Science No. 547, Springer-Verlag, 1991.

[134] M. Nadler, A. Sengupta, Shift Register Code for Indexing Applications, Communications of the ACM, Vol. 2, No. 10, Oct. 1959.

[135] H. Niederreiter, The Probabilistic Theory of Linear Complexity, Advances in Cryptology-EUROCRYPT'88, Lecture Notes in Computer Science, Springer-Verlag, 1989.

[136] H. Niederreiter, Keystream Sequences with a good Linear Complexity Profile for Every Starting Point, Advances in Cryptology-EUROCRYPT'89, Lecture Notes in Computer Science, Springer-Verlag, 1990.

[137] H. Niederreiter, A Combinatorial Approach to Probabilistic Results on the Linear-Complexity Profile of Random Sequences, Journal of Cryptology, Vol. 2, 1990.

[138] H. Niederreiter, C.P. Schnorr, Local Randomness in Polynomial Random Number and Random Function Generators, SIAM Journal of Computers, Vol. 22, No. 4, August 1993.

[139] L. O'Connor, A. Klapper, Algebraic Nonlinearity and its Applications to Cryptography, Journal of Cryptology, Vol. 7, No. 4, 1994.

[140] J.D. Olsen, R.A. Scholtz, L.R. Welch, Bent-Function Sequences, IEEE Transactions on Information Theory, Vol. IT-28, No. 6, Nov. 1982.

[141] O. Ore, Contributions on the Theory of Finite Fields, Transactions of the AMS, Vol. 36, 1934.

[142] J. Pastor, Criptografía: Cifrado, Protocolos y Aplicaciones, Seminario, Escuela de Ingenieros de Telecomunicaciones, Universidad Politécnica de Madrid, 1988.

[143] K.G. Paterson, New Lower Bounds on the Linear Complexity of Nonlinearly Filtered m-Sequences, enviado a IEEE Transactions on Information Theory, 1995.

[144] W.W. Peterson, E.J. Weldon, Error-Correcting Codes, Cambridge, MA:MIT Press, 1972.

[145] F. Piper, M. Walker, Linear Ciphers and Spreads, Journal of Cryptology, Vol. 1, 1989.

[146] S. Prasad, L.C. Quynh, Equivalent Linear Span Analysis of Binary Sequences Having an Interleaved Structure, IEE Proceedings, Vol. 133, Pt. F, No. 3, June 1986.

[147] I.S. Reed, R. Turn, A Generalization of Shift-Register Sequence Generators, Journal of the ACM, Vol. 16, No. 3, July 1969.

[148] K.H. Rosen, Elementary Number Theory and its Applications, Addison-Wesley, 1988.

[149] H.H. Roth, Linear Binary Shift Register Circuits Utilizing a Minimum Number of Mod-2 Adders, IEEE Transactions on Information Theory, Vol. IT-11, April 1965.

[150] F. Rubin, Decrypting a Stream Cipher Based on J-K Flip-Flops, IEEE Transactions on Computers, Vol. C-28, July 1979.

[151] R.A. Rueppel, Correlation Inmunity and the Summation Generator, Advances in Cryptology-CRYPTO'85, Lecture Notes in Computer Science, Springer-Verlag, 1986.

[152] R.A. Rueppel, Analysis and Design of Stream Ciphers, Springer-Verlag, 1986.

[153] R.A. Rueppel, O. Staffelbach, Products of Sequences with Maximum Linear Complexity, IEEE Transactions on Information Theory, Vol. IT-33, Jan. 1987.

[154] R. Safavi Naini, Parallel Generation of Pseudo-Random Sequences, Advances in Cryptology-AUSCRYPT'90, Lecture Notes in Computer Science No. 453, Springer-Verlag, 1990.

[155] C.P. Schnorr, On the Construction of Random Number Generators and Random Function Generators, Advances in Cryptology-EUROCRYPT'88, Lecture Notes in Computer Science No. 330, Springer-Verlag, 1988.

[156] S.R. Searle, Matrix Algebra Useful for Statistics, John Wiley & Sons, 1982.

[157] J.Seberry, J. Pieprzyk (Eds.), Advances in Cryptology-AUSCRYPT'90, Lecture Notes in Computer Science No. 453, Springer-Verlag, 1990.

[158] C.E. Shannon, A Mathematical Theory of Communications, Bell System Technical Journal, Vol. 27, Jul.-Oct. 1948.

[159] C.E. Shannon, Communication Theory of Secrecy Systems, Bell System Technical Journal, Vol. 28, Oct. 1949.

[160] T. Siegenthaler, Decrypting a Class of Stream Ciphers Using Ciphertext only, IEEE Transactions on Computers, Vol. C-33, 1984.

[161] T. Siegenthaler, Correlation-Inmunity of Nonlinear Combining Functions for Cryptographic Applications, IEEE Transactions on Information Theory, Vol. IT-30, Oct. 1984.

[162] T. Siegenthaler, Design of Combiners to Prevent Divide and Conquer Attacks, Advances in Cryptology-CRYPTO'85, Lecture Notes in Computer Science, Springer-Verlag, 1986.

[163] G.J. Simmons (Ed.), Contemporary Cryptology:The Science of Information Integrity, IEEE Press, 1991.

[164] O. Staffelbach, W. Meier, Cryptographic Significance of the Carry for Ciphers Based on Integer Addition, Advances in Cryptology-CRYPT0'90, Lecture Notes in Computer Science No. 537, Springer-Verlag, 1991.

[165] M. Stamp, C.F. Martin, An Algorithm for the k-Error Linear Complexity of Binary Sequences with Period 2^n, IEEE Transactions on Information Theory, Vol. 39, No. 4, July 1993.

[166] N.M. Stephens, The Zero Algorithm for Computing Linear Complexity Profiles, Collection: Cryptography and Coding, II, Oxford University Press, 1992.

[167] R. Turyn , J. Storer, On Binary Sequences, Proceedings of the AMS, Vol. 12, 1961.

[168] H.C.A. Van Tilborg, An Introduction to Cryptology, Kluwer Academic Publishers, 1988.

[169] G. S. Vernam, Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications, Journal of the American Institute of Electrical Engineers, Vol. 55, 1926.

[170] D. Wang, A. Compagner, On the use of Reducible Polynomials as Random Number Generators, Mathematics of Computation, Vol. 60, No. 201, Jan. 1993.

[171] M. Ward, The Arithmetical Theory of Linear Recurring Sequences, Transactions of the AMS, Vol. 35, 1933.

[172] L. Weng, Decomposition of m-Sequences and its Applications, IEEE Transactions on Information Theory, Vol. IT-17, 1971.

[173] M. Willett, The Minimum Polynomial for a Given Solution of a Linear Recursion, Duke Mathematical Journal, Vol. 39, 1972.

[174] M. Willett, The Index of an m-Sequence, SIAM Journal on Applied Mathematics, Vol. 25, 1973.

[175] M. Willett, Characteristic m-Sequences, Mathematics of Computation, Vol. 30, 1976.

[176] G.Z. Xiao, J.L. Massey, A Spectral Approach to Correlation-Inmune Combining Functions, IEEE Transactions on Information Theory, Vol. IT-34, May 1988.

[177] Y.X. Yang, New Binary Sequences with Perfect Staircase Profile of Linear Complexity, Information Processing Letters, Vol. 46, No. 1, 1993.

[178] M. Yoeli, Counting with Nonlinear Binary Feedback Shift Registers, IEEE Transactions on Electronic Computers, Vol. EC-12, August 1962.

[179] N. Zierler, On a Class of Binary Sequences, Proceedings of the AMS, Vol. 7, 1956.

[180] N. Zierler, W.H. Mills, Products of Linear Recurring Sequences, Journal of Algebra, Vol. 27, 1973.